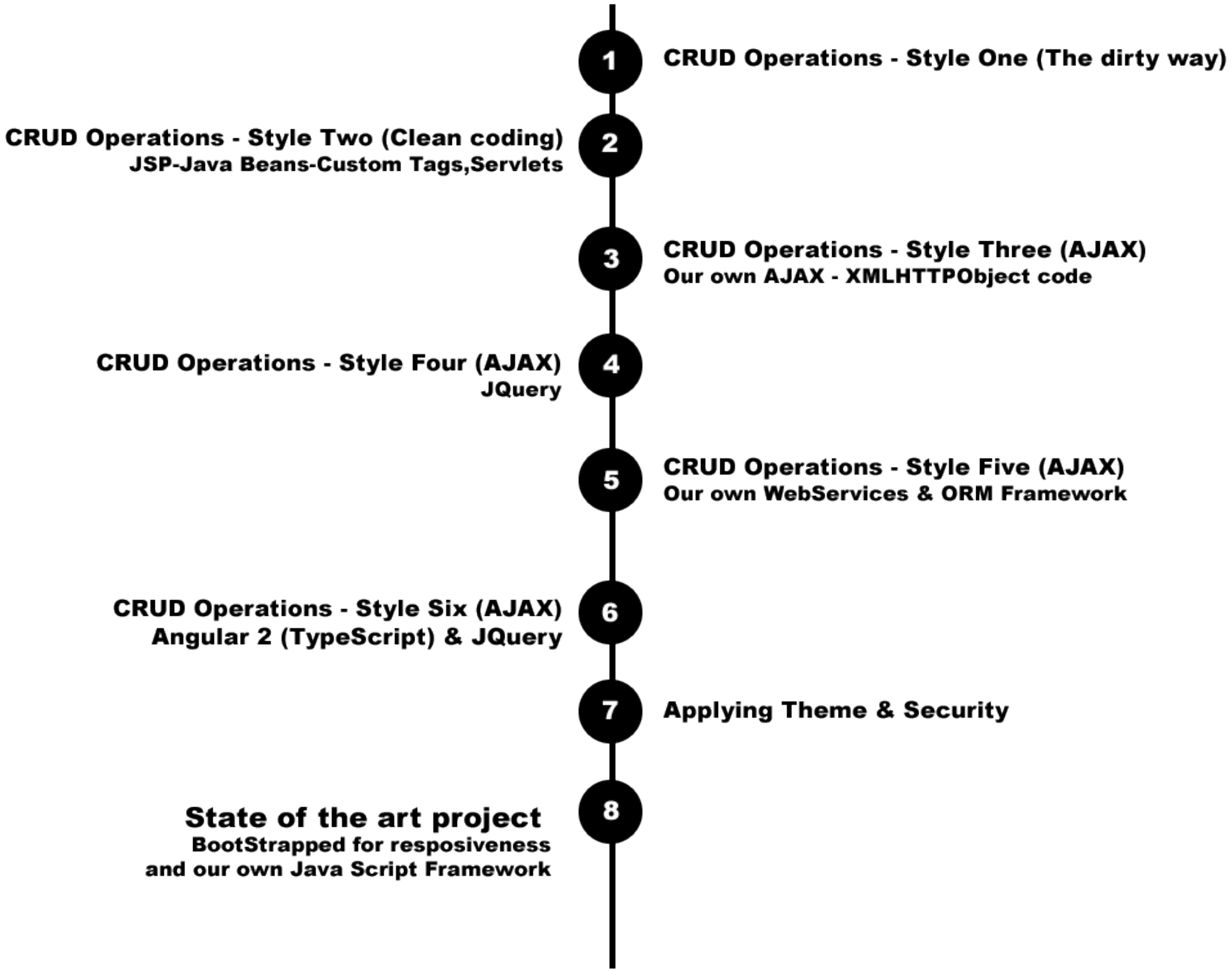
J2EE
Web Application Development

This is the kind of work that most of you will get,
when it comes to Indian Service Industry

I will try to take you beyond that.

The learning curve is going to be very steep.

To create magic,
me
as
well as you
need to work hard



1 CRUD Operations - Style One (The dirty way)

2 CRUD Operations - Style Two (Clean coding)
JSP-Java Beans-Custom Tags,Servlets

3 CRUD Operations - Style Three (AJAX)
Our own AJAX - XMLHTTPObject code

4 CRUD Operations - Style Four (AJAX)
JQuery

5 CRUD Operations - Style Five (AJAX)
Our own WebServices & ORM Framework

6 CRUD Operations - Style Six (AJAX)
Angular 2 (TypeScript) & JQuery

7 Applying Theme & Security

8 State of the art project
BootStrapped for resposiveness
and our own Java Script Framework

Download Apache Tomcat 8 from Apache's site (Download the zip version of binary file)

Unzip it and copy the extracted folder to c:\ and rename it to tomcat8

Note : The tomcat8 folder contains the bin, lib, webapps and some more folders.
The jar files to be included into classpath are in the lib folder and the context name (site folder) should be kept in the webapps folder.

Now edit to c:\tomcat8\bin\startup.bat file and insert the following 2 lines at the top

```
set JAVA_HOME=c:\jdk1.8
set CATALINA_HOME=c:\tomcat8
```

Note : those who are working on LINUX don't need to do anything as we have already configured tomcat8 for linux. You just have to replace ; with : and \ with / when it comes to specifying classpath.

Now create a shorcut to c:\tomcat8\bin\startup.bat named as tomcat8. (For windows users)
Double click the shortcut to start the server
Start the browser and type the url http://localhost:8080 and you should see the Apache Tomcat default home page.

We will be developing the CRUD Modules using various techniques.

I have created a mysql database named as mydb2017 and I will be connecting using the username as root and password as kelkar. I have already taught you to create database, create user and how to grant rights of the database to the use. You can create a database/user account of your choice.

Login and create table using the following SQL Statement

```
create table author
(
code int primary key auto_increment,
name char(35) not null unique
)Engine=InnoDB;
create table book
(
code int primary key auto_increment,
title char(35) not null unique,
author_code int not null,
category char(35) not null,
price int not null,
foreign key (author_code) references author(code)
)Engine=InnoDB;
```

CRUD One

Dirty Programming, but it is very essential to understand the techniques involved.

Issues : Refresh and Back button Problem, at some places I have purposely made the mistake of writing incorrect style.

These will be solved in CRUD Two

Download & See it in Action
http://tm-certificates.com/CRUDOne.wmv

Create the following structure in webapps folder
crudone – WEB-INF - classes
crudone - WEB-INF - lib
crudone - images
The lib folder should contain mysql.jar file
The classes folder will contain the necessary packages with compiled classes
The image folder should contain the following image files

delete_icon.png (16x16)

edit_icon.png (16x16)

logo.png (whatever is your requirement)

You can download zip file with these images - CRUDOneResources.zip

The Data Layer (To compile, use javac *.java)

```
package com.thinking.machines.library.dl;
public class DAOException extends Exception
{
public DAOException(String message)
{
super(message);
}
}
```

```
package com.thinking.machines.library.dl;
import java.sql.*;
public class DAOConnection
{
private DAOConnection()
{
}
public static Connection getConnection() throws DAOException
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection connection;
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb2017","root","kelkar");
return connection;
}catch(ClassNotFoundException classNotFoundException)
{
throw new DAOException("Driver class : com.mysql.jdbc.Driver missing in classpath");
}
catch(SQLException sqlException)
{
throw new DAOException("Unable to connect : "+sqlException.getMessage());
}
}
}
```

```
package com.thinking.machines.library.dl;
public interface AuthorInterface extends java.io.Serializable,Comparable<AuthorInterface>
{
public void setCode(int code);
public int getCode();
public void setName(String name);
public String getName();
}
```

```
package com.thinking.machines.library.dl;
public class Author implements AuthorInterface
{
private int code;
public String name;
public Author()
{
this.code=0;
this.name="";
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
return this.code;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
public boolean equals(Object object)
{
if(!(object instanceof Author)) return false;
Author otherAuthor=(Author)object;
return this.code==otherAuthor.code;
}
public int compareTo(AuthorInterface otherAuthor)
{
return this.code-otherAuthor.getCode();
}
public int hashCode()
{
return this.code;
}
}
```

```
package com.thinking.machines.library.dl;
import java.util.*;
public interface AuthorDAOInterface
{
public void add(AuthorInterface author) throws DAOException;
public void update(AuthorInterface author) throws DAOException;
public void remove(int code) throws DAOException;
public AuthorInterface getByCode(int code) throws DAOException;
public AuthorInterface getByName(String name) throws DAOException;
public LinkedList<AuthorInterface> getAll() throws DAOException;
public long getCount() throws DAOException;
}
```

```
package com.thinking.machines.library.dl;
import java.util.*;
import java.sql.*;
public class AuthorDAO implements AuthorDAOInterface
{
public void add(AuthorInterface author) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from author where name=?");
preparedStatement.setString(1,author.getName());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Author : "+author.getName()+" exists.");
}
preparedStatement=connection.prepareStatement("insert into author (name)
values(?)",Statement.RETURN_GENERATED_KEYS);
preparedStatement.setString(1,author.getName());
preparedStatement.executeUpdate();
resultSet=preparedStatement.getGeneratedKeys();
resultSet.next();
author.setCode(resultSet.getInt(1));
resultSet.close();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to add : "+sqlException.getMessage());
}
}
public void update(AuthorInterface author) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from author where name=? and
code <>?");
preparedStatement.setString(1,author.getName());
preparedStatement.setInt(2,author.getCode());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Author : "+author.getName()+" exists.");
}
preparedStatement=connection.prepareStatement("update author set name=? where code=?");
preparedStatement.setString(1,author.getName());
preparedStatement.setInt(2,author.getCode());
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to update : "+sqlException.getMessage());
}
}

public void remove(int code) throws DAOException
```

```
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from author where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists==false)
{
connection.close();
throw new DAOException("Author code : "+code+"does not exist.");
}
preparedStatement=connection.prepareStatement("select 1 as result from book where author_code=?");
preparedStatement.setInt(1,code);
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Cannot delete author as book exists against it.");
}
preparedStatement=connection.prepareStatement("delete from author where code=?");
preparedStatement.setInt(1,code);
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to delete : "+sqlException.getMessage());
}
}
public AuthorInterface getByCode(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from author where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Author code : "+code+" does not exist.");
}
AuthorInterface author=new Author();
author.setCode(resultSet.getInt("code"));
author.setName(resultSet.getString("name").trim());
resultSet.close();
preparedStatement.close();
connection.close();
return author;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}

public AuthorInterface getByName(String name) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from author where name=?");
preparedStatement.setString(1,name);
ResultSet resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Author : "+name+" does not exist.");
}
AuthorInterface author=new Author();
author.setCode(resultSet.getInt("code"));
author.setName(resultSet.getString("name").trim());
resultSet.close();
preparedStatement.close();
connection.close();
return author;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}
```

```
public LinkedList<AuthorInterface> getAll() throws DAOException
{
LinkedList<AuthorInterface> authors;
try
{
Connection connection=DAOConnection.getConnection();
Statement statement=connection.createStatement();
ResultSet resultSet=statement.executeQuery("select * from author order by name ");
if(resultSet.next()==false)
{
resultSet.close();
statement.close();
connection.close();
throw new DAOException("No authors");
}
authors=new LinkedList<AuthorInterface>();
AuthorInterface author;
do
{
author=new Author();
author.setCode(resultSet.getInt("code"));
author.setName(resultSet.getString("name").trim());
authors.add(author);
}while(resultSet.next());
resultSet.close();
statement.close();
connection.close();
return authors;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
public long getCount() throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
Statement statement=connection.createStatement();
ResultSet resultSet=statement.executeQuery("select count(*) as cnt from author");
resultSet.next();
long count=resultSet.getLong("cnt");
resultSet.close();
statement.close();
connection.close();
return count;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
}
```

```
package com.thinking.machines.library.dl;
public interface BookInterface extends java.io.Serializable,Comparable<BookInterface>
{
public void setCode(int code);
public int getCode();
public void setTitle(String title);
public String getTitle();
public void setAuthorCode(int authorCode);
public int getAuthorCode();
public void setCategory(String category);
public String getCategory();
public void setPrice(int price);
public int getPrice();
}
```

```
package com.thinking.machines.library.dl;
public class Book implements BookInterface
{
private int code;
public String title;
private int authorCode;
private String category;
private int price;
public Book()
{
this.code=0;
this.title="";
this.authorCode=0;
this.category="";
this.price=0;
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
return this.code;
}
public void setTitle(String title)
{
this.title=title;
}
public String getTitle()
{
return this.title;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public void setCategory(String category)
{
this.category=category;
}
public String getCategory()
{
return this.category;
}
public void setPrice(int price)
{
this.price=price;
}
public int getPrice()
{
return this.price;
}
public boolean equals(Object object)
{
if(!(object instanceof Book)) return false;
Book otherBook=(Book)object;
return this.code==otherBook.code;
}
public int compareTo(BookInterface otherBook)
{
return this.code-otherBook.getCode();
}
public int hashCode()
{
return this.code;
}
}
```

```
package com.thinking.machines.library.dl;
import java.util.*;
```

```
public interface BookDAOInterface
{
public void add(BookInterface book) throws DAOException;
public void update(BookInterface book) throws DAOException;
public void remove(int code) throws DAOException;
public BookInterface getByCode(int code) throws DAOException;
public BookInterface getByTitle(String title) throws DAOException;
public LinkedList<BookInterface> getAll() throws DAOException;
public LinkedList<BookInterface> getAllByAuthorCode(int authorCode) throws DAOException;
public boolean containsBookWithAuthorCode(int code) throws DAOException;
public long getCount() throws DAOException;
}
```

```
package com.thinking.machines.library.dl;
import java.util.*;
import java.sql.*;
public class BookDAO  implements BookDAOInterface
{
public void add(BookInterface book) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from book where title=?");
preparedStatement.setString(1,book.getTitle());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Book with title : "+book.getTitle()+" exists.");
}
preparedStatement=connection.prepareStatement("select 1 as result from author where code=?");
preparedStatement.setInt(1,book.getAuthorCode());
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Invalid author code : "+book.getAuthorCode());
}
preparedStatement=connection.prepareStatement("insert into book (title,author_code,category,price)
values(?,?,?,?)",Statement.RETURN_GENERATED_KEYS);
preparedStatement.setString(1,book.getTitle());
preparedStatement.setInt(2,book.getAuthorCode());
preparedStatement.setString(3,book.getCategory());
preparedStatement.setInt(4,book.getPrice());
preparedStatement.executeUpdate();
resultSet=preparedStatement.getGeneratedKeys();
resultSet.next();
book.setCode(resultSet.getInt(1));
resultSet.close();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to add : "+sqlException.getMessage());
}
}
public void update(BookInterface book) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from book where code=?");
preparedStatement.setInt(1,book.getCode());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Book code : "+book.getCode()+" does not exist.");
}
preparedStatement=connection.prepareStatement("select 1 as result from book where title=? and
code<>?");
preparedStatement.setString(1,book.getTitle());
preparedStatement.setInt(2,book.getCode());
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Book title : "+book.getTitle()+" exists.");
}
```

```
preparedStatement=connection.prepareStatement("select 1 as result from author where code=?");
preparedStatement.setInt(1,book.getAuthorCode());
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Author code : "+book.getAuthorCode()+" does not exist.");
}
preparedStatement=connection.prepareStatement("update book set
title=?,author_code=?,category=?,price=? where code=?");
preparedStatement.setString(1,book.getTitle());
preparedStatement.setInt(2,book.getAuthorCode());
preparedStatement.setString(3,book.getCategory());
preparedStatement.setInt(4,book.getPrice());
preparedStatement.setInt(5,book.getCode());
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to update : "+sqlException.getMessage());
}
}

public void remove(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from book where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Book code : "+code+" does not exist.");
}
preparedStatement=connection.prepareStatement("delete from book where code=?");
preparedStatement.setInt(1,code);
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to delete : "+sqlException.getMessage());
}
}
public BookInterface getByCode(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from book where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet;
resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Book code : "+code+" does not exist.");
}
BookInterface book;
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
resultSet.close();
preparedStatement.close();
connection.close();
return book;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}
public BookInterface getByTitle(String title) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from book where title=?");
```

```
preparedStatement.setString(1,title);
ResultSet resultSet;
resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Book title : "+title+" does not exist.");
}
BookInterface book;
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
resultSet.close();
preparedStatement.close();
connection.close();
return book;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}

public LinkedList<BookInterface> getAll() throws DAOException
{
LinkedList<BookInterface> books;
try
{
Connection connection=DAOConnection.getConnection();
Statement statement=connection.createStatement();
ResultSet resultSet=statement.executeQuery("select * from book order by title");
if(resultSet.next()==false)
{
resultSet.close();
statement.close();
connection.close();
throw new DAOException("No books");
}
books=new LinkedList<BookInterface>();
Book book;
do
{
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
books.add(book);
}while(resultSet.next());
resultSet.close();
statement.close();
connection.close();
return books;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
public LinkedList<BookInterface> getAllByAuthorCode(int authorCode) throws DAOException
{
LinkedList<BookInterface> books;
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement=connection.prepareStatement("Select * from book where
author_code=?");
preparedStatement.setInt(1,authorCode);
ResultSet resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("No books");
}
books=new LinkedList<BookInterface>();
Book book;
do
{
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
books.add(book);
}while(resultSet.next());
resultSet.close();
preparedStatement.close();
```

```
connection.close();
return books;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
public boolean containsBookWithAuthorCode(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement=connection.prepareStatement("select 1 as result from book where author_code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
boolean contains=resultSet.next();
resultSet.close();
preparedStatement.close();
connection.close();
return contains;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
public long getCount() throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
Statement statement=connection.createStatement();
ResultSet resultSet=statement.executeQuery("select count(*) as cnt from book");
resultSet.next();
long count=resultSet.getLong("cnt");
resultSet.close();
statement.close();
connection.close();
return count;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
}
```

index.html - The Index Page (Location (crudone) folder)

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
</head>
<body style='background:#F5F5F5'>
<div style='background:#DBDBDB;width:100%;border:2px'>
<img src='/crudone/images/logo.png'><br>
</div>
<br>
<br>
<h3>Author</h3>
<ul>
<li><a href='/crudone/AuthorAddForm.html'>Add</a></li>
<li><a href='/crudone/AuthorEditForm.html'>Edit</a></li>
<li><a href='/crudone/AuthorDeleteForm.html'>Delete</a></li>
<li><a href='/crudone/getAuthors'>List</a></li>
</ul>
<h3>Book</h3>
<ul>
<li><a href='/crudone/BookAddForm.html'>Add</a></li>
<li><a href='/crudone/getBooks'>Edit/Delete/List</a></li>
</ul>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040
</center>
</div>
</body>
</html>
```

The Author Add Module Starts
AuthorAddForm.html (Location – (crudone) folder)

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<script>
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
</script>
</head>
<body style='background:#F5F5F5'>
<div style='background:#DBDBDB;width:100%;border:2px'>
<table width='100%'>
<tr>
<td>
<img src='/crudone/images/logo.png'><br>
</td>
<td align='right' style='padding:5px'>
<a href='/crudone/'>Home</a>
</td>
</table>
</div>
<br>
<br>
<h2>Add Author </h2>
<form action='/crudone/addAuthor' onsubmit='return validateForm(this)' method='post'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
```

```
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' style='color:red'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Add</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040
</center>
</div>
</body>
</html>
```

web.xml (Location – (WEB-INF) folder)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">
  <description>
    CRUD Operations - Style One
  </description>
  <display-name>CRUD Operations - Style One</display-name>

<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditAuthor</servlet-name>
<url-pattern>/editAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateAuthor</servlet-name>
<url-pattern>/updateAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>GetAuthorForDeletion</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthorForDeletion</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthorForDeletion</servlet-name>
<url-pattern>/getAuthorForDeletion</url-pattern>
</servlet-mapping>


<servlet>
<servlet-name>DeleteAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteAuthor</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>DeleteAuthor</servlet-name>
<url-pattern>/deleteAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>GetAuthors</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthors</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthors</servlet-name>
<url-pattern>/getAuthors</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>BookAddForm</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.BookAddForm</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>BookAddForm</servlet-name>
<url-pattern>/BookAddForm.html</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>AddBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddBook</servlet-name>
<url-pattern>/addBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>GetBooks</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetBooks</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetBooks</servlet-name>
<url-pattern>/getBooks</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>DeleteBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteBook</servlet-name>
```

```
<url-pattern>/deleteBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditBook</servlet-name>
<url-pattern>/editBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateBook</servlet-name>
<url-pattern>/updateBook</url-pattern>
</servlet-mapping>
</web-app>
```

To compile the servlets

On Windows

javac -classpath c:\tomcat8\lib\*;c:\tomcat8\webapps\crudone\WEB-INF\classes;. *.java

On Linux

replace c:\tomcat8 with your apache-tomcat folder path, replace all ; with : and \ with /

AddAuthor.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorInterface author=new Author();
author.setName(name);
AuthorDAO authorDAO=new AuthorDAO();
try
{
```

```
authorDAO.add(author);
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Author added successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorAddForm.html'>");
pw.println("<Button type='submit'>Add more authors</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Add Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/addAuthor' onsubmit='return validateForm(this)' method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='36' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Add</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

The Author Edit Module
AuthorEditForm.html (Location – (crudone) folder)

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
```

```
<meta name="author" content="Thinking Machines">
<script>
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
</script>
</head>
<body style='background:#F5F5F5'>
<div style='background:#DBDBDB;width:100%;border:2px'>
<table width='100%'>
<tr>
<td>
<img src='/crudone/images/logo.png'><br>
</td>
<td align='right' style='padding:5px'>
<a href='/crudone/'>Home</a>
</td>
</table>
</div>
<br>
<br>
<h2>Edit Author </h2>
<form action='/crudone/editAuthor' onsubmit='return validateForm(this)' method='post'>
<table border='0'>
<tr>
```

```
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' style='color:red'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Edit</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040
</center>
</div>
</body>
</html>
```

EditAuthor.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class EditAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
```

```
author=authorDAO.getByName(name);
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Author </h2>");
pw.println("<form action='/crudone/updateAuthor' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' name='code' id='code' value='"+author.getCode()+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Code</td>");
pw.println("<td>");
pw.println("<b>"+author.getCode()+"</b>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/editAuthor' onsubmit='return validateForm(this)' method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Edit</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

UpdateAuthor.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
```

```
public class UpdateAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
int code=Integer.parseInt(request.getParameter("code"));
String name=request.getParameter("name");
AuthorInterface author;
AuthorDAO authorDAO=new AuthorDAO();
try
{
author=authorDAO.getByCode(code);
}catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>That author cannot be updated as someone working in parallel deleted the author while
you were trying to update it.</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorEditForm.html'>");
pw.println("<Button type='submit'>Edit other author</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
return;
}
author.setName(name);
try
{
authorDAO.update(author);
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Author updated successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorEditForm.html'>");
pw.println("<Button type='submit'>Edit more authors</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/updateAuthor' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' name='code' id='code' value='"+author.getCode()+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Code</td>");
pw.println("<td>");
pw.println("<b>"+author.getCode()+"</b>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

The Author Delete Module

AuthorDeleteForm.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<script>
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
</script>
</head>
<body style='background:#F5F5F5'>
<div style='background:#DBDBDB;width:100%;border:2px'>
<table width='100%'>
<tr>
<td>
<img src='/crudone/images/logo.png'><br>
</td>
<td align='right' style='padding:5px'>
<a href='/crudone/'>Home</a>
</td>
</table>
</div>
<br>
<br>
<h2>Delete Author </h2>
<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)' method='post'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' style='color:red'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Delete</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040
```

```
</center>
</div>
</body>
</html>
```

GetAuthorForDeletion.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class GetAuthorForDeletion extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
author=authorDAO.getByName(name);
BookDAO bookDAO=new BookDAO();
try
{
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(author.getCode());
if(bookWithAuthorExists)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<div style='color:red'>Cannot delete author as book exists against it.</div>");
pw.println("<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
return;
}
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
```

```
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<form action='/crudone/deleteAuthor' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' name='code' id='code' value='"+author.getCode()+"'>");
pw.println("<input type='hidden' name='name' id='name' value='"+author.getName()+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Code : </td>");
pw.println("<td>");
pw.println("<b>"+author.getCode()+"</b>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Name of author : </td>");
pw.println("<td>");
pw.println(author.getName());
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

DeleteAuthor.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class DeleteAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
int code=Integer.parseInt(request.getParameter("code"));
String name=request.getParameter("name");
BookDAO bookDAO=new BookDAO();
try
{
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(code);
if(bookWithAuthorExists)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<div style='color:red'>Cannot delete author as book exists against it.</div>");
pw.println("<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
return;
}
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.remove(code);
}catch(DAOException daoException)
{
}
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Author deleted successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorDeleteForm.html'>");
pw.println("<Button type='submit'>Delete more authors</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

The Author List Module
GetAuthor.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetAuthors extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Authors </h2>");
pw.println("<table  style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Author</th>");
pw.println("<th>Code</th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(AuthorInterface author:authors)
```

```
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+author.getName()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+author.getCode()+"</td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove this after testing
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Authors </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No authors added</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/'>");
pw.println("<Button type='submit'>Ok</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

The Book Add Module

BookAddForm.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class BookAddForm extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
```

```
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
try
{
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode==-1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
pw.println("}");
pw.println("}");
pw.println("if(category=='none')");
pw.println("{");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.category;");
pw.println("}");
pw.println("}");
pw.println("if(price.length==0)");
pw.println("{");
pw.println("frm.price.value=\"0\";");
pw.println("}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{");
pw.println("if(v.indexOf(price.charAt(i))==-1)");
pw.println("{");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.price;");
pw.println("}");
pw.println("break;");
pw.println("}");
pw.println("i++;");
pw.println("}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Add Book</h2>");
pw.println("<form action='/crudone/addBook'  onsubmit='return validateForm(this)'  method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
pw.println("<option value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
pw.println("<option value='Science fiction'>Science fiction</option>");
pw.println("<option value='Satire'>Satire</option>");
pw.println("<option value='Drama'>Drama</option>");
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
pw.println("<option value='Mystery'>Mystery</option>");
pw.println("<option value='Horror'>Horror</option>");
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Add</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Add Book </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No authors added, cannot add book without adding authors.</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorAddForm.html'>");
pw.println("<Button type='submit'>Add author</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
}
}
}
```

AddBook.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
String title=request.getParameter("title");
int authorCode=Integer.parseInt(request.getParameter("authorCode"));
```

```
String category=request.getParameter("category");
int price=Integer.parseInt(request.getParameter("price"));
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookDAO bookDAO=new BookDAO();
BookInterface book=new Book();
book.setTitle(title);
book.setAuthorCode(authorCode);
book.setCategory(category);
book.setPrice(price);
try
{
bookDAO.add(book);
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Book added successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/BookAddForm.html'>");
pw.println("<Button type='submit'>Add more books</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
try
{
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode==-1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
pw.println("}");
pw.println("}");
pw.println("if(category=='none')");
pw.println("{");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.category;");
pw.println("}");
pw.println("}");
pw.println("if(price.length==0)");
pw.println("{");
pw.println("frm.price.value=\"0\";");
pw.println("}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{");
pw.println("if(v.indexOf(price.charAt(i))==-1)");
pw.println("{");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.price;");
pw.println("}");
pw.println("break;");
pw.println("}");
pw.println("i++;");
pw.println("}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Add Book</h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/addBook'  onsubmit='return validateForm(this)'  method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31' value='"+title+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
if(author.getCode()==authorCode)
{
pw.println("<option selected value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
else
{
pw.println("<option value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
if(category.equals("Science fiction"))
{
pw.println("<option selected value='Science fiction'>Science fiction</option>");
}
else
{
pw.println("<option value='Science fiction'>Science fiction</option>");
}
if(category.equals("Satire"))
{
pw.println("<option selected value='Satire'>Satire</option>");
}
else
{
pw.println("<option value='Satire'>Satire</option>");
}
if(category.equals("Drama"))
{
pw.println("<option selected value='Drama'>Drama</option>");
}
else
{
pw.println("<option value='Drama'>Drama</option>");
}
if(category.equals("Action and Adventure"))
{
pw.println("<option selected value='Action and Adventure'>Action and Adventure</option>");
}
else
{
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
}
if(category.equals("Mystery"))
{
pw.println("<option selected value='Mystery'>Mystery</option>");
}
else
{
pw.println("<option value='Mystery'>Mystery</option>");
}
if(category.equals("Horror"))
{
pw.println("<option selected value='Horror'>Horror</option>");
}
```

```
else
{
pw.println("<option value='Horror'>Horror</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'
value='"+price+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Add</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException2)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Add Book </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No authors added, cannot add book without adding authors.</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorAddForm.html'>");
pw.println("<Button type='submit'>Add author</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}
}catch(Exception exception)
{
}
}
}
```

The Book List With Edit/Delete Options

GetBooks.java

```
package com.thinking.machines.library.servlets;
```

```
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetBooks extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
BookDAO bookDAO=new BookDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
LinkedList<BookInterface> books;
books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<table  style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface book:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+book.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==book.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border:1px solid black'>"+book.getCategory()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+book.getCode()+"'><img src='/crudone/images/edit_icon.png'
style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook("+book.getCode()+",\""+book.getTitle()+"\")'><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='POST'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove this after testing
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No books added </b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/'>");
pw.println("<Button type='submit'>Ok</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

EditBook.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class EditBook extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
int code=Integer.parseInt(request.getParameter("code"));
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookInterface book;
BookDAO bookDAO=new BookDAO();
```

```
try
{
book=bookDAO.getByCode(code);
String title=book.getTitle();
int authorCode=book.getAuthorCode();
String category=book.getCategory();
int price=book.getPrice();
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode==-1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
pw.println("}");
pw.println("}");
pw.println("if(category=='none')");
pw.println("{");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.category;");
pw.println("}");
pw.println("}");
pw.println("if(price.length==0)");
pw.println("{");
pw.println("frm.price.value=\"0\";");
pw.println("}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{");
pw.println("if(v.indexOf(price.charAt(i))==-1)");
pw.println("{");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.price;");
pw.println("}");
pw.println("break;");
pw.println("}");
pw.println("i++;");
pw.println("}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Book</h2>");
pw.println("<form action='/crudone/updateBook'  onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' id='code' name='code' value='"+code+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31' value='"+title+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
if(author.getCode()==authorCode)
{
pw.println("<option selected value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
else
{
pw.println("<option value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
if(category.equals("Science fiction"))
{
pw.println("<option selected value='Science fiction'>Science fiction</option>");
}
else
{
pw.println("<option value='Science fiction'>Science fiction</option>");
}
if(category.equals("Satire"))
{
pw.println("<option selected value='Satire'>Satire</option>");
}
else
{
pw.println("<option value='Satire'>Satire</option>");
}
if(category.equals("Drama"))
{
pw.println("<option selected value='Drama'>Drama</option>");
}
else
{
pw.println("<option value='Drama'>Drama</option>");
}
if(category.equals("Action and Adventure"))
{
pw.println("<option selected value='Action and Adventure'>Action and Adventure</option>");
}
else
{
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
}
if(category.equals("Mystery"))
{
pw.println("<option selected value='Mystery'>Mystery</option>");
}
else
{
pw.println("<option value='Mystery'>Mystery</option>");
}
if(category.equals("Horror"))
{
pw.println("<option selected value='Horror'>Horror</option>");
}
else
{
```

```
pw.println("<option value='Horror'>Horror</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'
value='"+price+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
LinkedList<BookInterface> books;
```

```
books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<table  style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface b:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+b.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+b.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==b.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border:1px solid black'>"+b.getCategory()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+b.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+b.getCode()+"'><img src='/crudone/images/edit_icon.png' style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook("+b.getCode()+",\""+b.getTitle()+"\")'><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='POST'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException2)
{
// this case won't arise
}
}
}catch(Exception exception)
{
}
}
}
```

UpdateBook.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class UpdateBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
int code=Integer.parseInt(request.getParameter("code"));
String title=request.getParameter("title");
int authorCode=Integer.parseInt(request.getParameter("authorCode"));
String category=request.getParameter("category");
int price=Integer.parseInt(request.getParameter("price"));
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookDAO bookDAO=new BookDAO();
BookInterface book=new Book();
book.setCode(code);
book.setTitle(title);
book.setAuthorCode(authorCode);
book.setCategory(category);
book.setPrice(price);
try
{
bookDAO.update(book);
LinkedList<AuthorInterface> authors;
AuthorDAO authorDAO=new AuthorDAO();
```

```
authors=authorDAO.getAll();
LinkedList<BookInterface> books;
books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<h3>Book title : "+title+" updated.</h3>");
pw.println("<table  style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface b:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+b.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+b.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==b.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border:1px solid black'>"+b.getCategory()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+b.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+b.getCode()+"'><img src='/crudone/images/edit_icon.png' style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook("+b.getCode()+",\""+b.getTitle()+"\")'><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='POST'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
try
{
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode==-1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
pw.println("}");
pw.println("}");
pw.println("if(category=='none')");
pw.println("{");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.category;");
pw.println("}");
pw.println("}");
pw.println("if(price.length==0)");
pw.println("{");
pw.println("frm.price.value=\"0\";");
pw.println("}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{");
pw.println("if(v.indexOf(price.charAt(i))==-1)");
pw.println("{");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.price;");
pw.println("}");
pw.println("break;");
pw.println("}");
pw.println("i++;");
pw.println("}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Book</h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/updateBook'  onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' id='code' name='code' value='"+code+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31' value='"+title+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
if(author.getCode()==authorCode)
{
pw.println("<option selected value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
else
{
pw.println("<option value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
if(category.equals("Science fiction"))
{
pw.println("<option selected value='Science fiction'>Science fiction</option>");
}
else
{
pw.println("<option value='Science fiction'>Science fiction</option>");
}
if(category.equals("Satire"))
{
pw.println("<option selected value='Satire'>Satire</option>");
}
else
{
pw.println("<option value='Satire'>Satire</option>");
}
if(category.equals("Drama"))
{
pw.println("<option selected value='Drama'>Drama</option>");
}
else
{
pw.println("<option value='Drama'>Drama</option>");
}
if(category.equals("Action and Adventure"))
{
pw.println("<option selected value='Action and Adventure'>Action and Adventure</option>");
}
else
{
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
}
if(category.equals("Mystery"))
{
pw.println("<option selected value='Mystery'>Mystery</option>");
}
else
{
pw.println("<option value='Mystery'>Mystery</option>");
}
if(category.equals("Horror"))
```

```
{
pw.println("<option selected value='Horror'>Horror</option>");
}
else
{
pw.println("<option value='Horror'>Horror</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'
value='"+price+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException2)
{
// this case won't arise
```

```
}
}
}catch(Exception exception)
{
}
}
}
```

DeleteBook.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class DeleteBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
int code=Integer.parseInt(request.getParameter("code"));
String title=request.getParameter("title");
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookDAO bookDAO=new BookDAO();
try
{
bookDAO.remove(code);
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove this after testing
}
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
LinkedList<BookInterface> books;
books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<h3>Book : "+title+" deleted.</h3>");
pw.println("<table  style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface book:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+book.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==book.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border:1px solid black'>"+book.getCategory()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+book.getCode()+"'><img src='/crudone/images/edit_icon.png'
style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook("+book.getCode()+",\""+book.getTitle()+"\")'><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='post'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException)
{
```

```
System.out.println(daoException); // don't forget to remove this after testing
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No books added </b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/'>");
pw.println("<Button type='submit'>Ok</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}catch(Exception exception)
{
}
}
```

```
}
```

--- CRUD Two Complete ---

CRUD Two

JSP – Java Beans – TagSupport – Servlets

Bridging the gap between a designer and a programmer.

Download & See it in Action
http://tm-certificates.com/CRUDTwo.wmv

Create folder named as crudtwo in webapps

The following folder structures need to be created

crudtwo – WEB-INF
crudtwo – WEB-INF – classes
crudtwo – WEB-INF – lib
crudtwo – WEB-INF – tlds
crudtwo – images
crudtwo – js
crudtwo – css

all the JSP files should be kept in crudtwo folder
copy the images from crudone\images to crudtwo\images

Create the necessary folders for packages

crudtwo – WEB-INF – classes – com – thinking – machines – library – dl
crudtwo – WEB-INF – classes – com – thinking – machines – library – beans
crudtwo – WEB-INF – classes – com – thinking – machines – library – tags
crudtwo – WEB-INF – classes – com – thinking – machines – library – servlets

copy the contents of the crudone\WEB-INF\classes\com\thinking\machines\library\dl to crudtwo\WEB-INF\classes\com\thinking\machines\library\dl
We won't be rewriting the data layer classes, they are going to stay same.

The CustomTags.tld file should be placed in crudtwo\WEB-INF\tlds folder (the second char is l for lion and not digit 1)

The web.xml should be placed in crudtwo\WEB-INF folder
The mysql.jar file should be in the crudtwo\WEB-INF\lib folder

The files with extension js should be placed in the crudtwo\js folder
the files with extension css should be placed in the crudtwo\css folder

The BackBone

We are wearing the shoes of a Programmer and know nothing about designing.

web.xml (Location WEB-INF folder)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

    <description>
      CRUD Operations - Style One
    </description>
    <display-name>CRUD Operations - Style One</display-name>

<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
```

```
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditAuthor</servlet-name>
<url-pattern>/editAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateAuthor</servlet-name>
<url-pattern>/updateAuthor</url-pattern>
</servlet-mapping>


<servlet>
<servlet-name>GetAuthorForDeletion</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthorForDeletion</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthorForDeletion</servlet-name>
<url-pattern>/getAuthorForDeletion</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>DeleteAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteAuthor</servlet-name>
<url-pattern>/deleteAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>AddBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddBook</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>AddBook</servlet-name>
<url-pattern>/addBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>DeleteBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteBook</servlet-name>
<url-pattern>/deleteBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditBook</servlet-name>
<url-pattern>/editBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateBook</servlet-name>
<url-pattern>/updateBook</url-pattern>
</servlet-mapping>

</web-app>
```

CustomTags.tld (location WEB-INF\tlds folder)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!DOCTYPE taglib
        PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
        "http://java.sun.com/j2ee/dtd/web-jsptaglibrary_1_2.dtd">

<taglib>

  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>CustomTags</short-name>
  <description>
   A tag library for this project
  </description>
  <tag>
    <name>SetEnvironmentVariables</name>
    <tag-class>com.thinking.machines.library.tags.EnvironmentVariablesTagHandler</tag-class>
    <description> Environment Variables Tag Handler </description>
  </tag>
  <tag>
    <name>FormId</name>
    <tag-class>com.thinking.machines.library.tags.FormIdTagHandler</tag-class>
    <description> Form Id. Generator Tag </description>
  </tag>
  <tag>
    <name>FormResubmitted</name>
    <tag-class>com.thinking.machines.library.tags.FormResubmittedTagHandler</tag-class>
    <description> Process body if form is resubmitted </description>
  </tag>
  <tag>
    <name>IfModule</name>
    <tag-class>com.thinking.machines.library.tags.IfModuleTagHandler</tag-class>
    <description> Tag to identify module page </description>
  </tag>
  <tag>
    <name>If</name>
    <tag-class>com.thinking.machines.library.tags.IfTagHandler</tag-class>
    <description> Tag to identify module page </description>
    <attribute>
    <name>expression</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    </attribute>
  </tag>
  <tag>
    <name>Author</name>
    <tag-class>com.thinking.machines.library.tags.AuthorTagHandler</tag-class>
    <tei-class>com.thinking.machines.library.tags.AuthorTEI</tei-class>
    <description> Author Tag Handler </description>
  </tag>
  <tag>
    <name>Book</name>
    <tag-class>com.thinking.machines.library.tags.BookTagHandler</tag-class>
    <tei-class>com.thinking.machines.library.tags.BookTEI</tei-class>
    <description> Book Tag Handler </description>
    <attribute>
    <name>authorCode</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    </attribute>
  </tag>
</taglib>
```

To compile the java files

```
javac   -classpath    c:\tomcat8\lib\*;c:\tomcat8\webapps\crudtwo\WEB-INF\classes;.    *.java
```

Note : You know what to do in case of LINUX

Data Layer classes (Same as crudone)

```
package com.thinking.machines.library.beans;
public class NotificationBean implements java.io.Serializable
{
private String message;
private String actionText;
private String actionURL;
public NotificationBean()
{
this.message="";
}
public void setMessage(String message)
{
this.message=message;
}
public String getMessage()
{
return this.message;
}
public void setActionText(String actionText)
{
this.actionText=actionText;
}
public String getActionText()
{
return this.actionText;
}
public void setActionURL(String actionURL)
{
this.actionURL=actionURL;
}
public String getActionURL()
{
return this.actionURL;
}
}
```

```
package com.thinking.machines.library.beans;
public class ErrorBean implements java.io.Serializable
{
private String errorMessage;
public ErrorBean()
{
this.errorMessage="";
}
public void setErrorMessage(String errorMessage)
{
this.errorMessage=errorMessage;
}
public String getErrorMessage()
{
return this.errorMessage;
}
}
```

```
package com.thinking.machines.library.beans;
public class AuthorBean implements java.io.Serializable,Comparable<AuthorBean>
{
private int code;
public String name;
public AuthorBean()
{
this.code=0;
this.name="";
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
```

```
return this.code;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
public boolean equals(Object object)
{
if(!(object instanceof AuthorBean)) return false;
AuthorBean otherAuthorBean=(AuthorBean)object;
return this.code==otherAuthorBean.code;
}
public int compareTo(AuthorBean otherAuthorBean)
{
return this.code-otherAuthorBean.getCode();
}
public int hashCode()
{
return this.code;
}
}
```

```
package com.thinking.machines.library.beans;
public class BookBean implements java.io.Serializable,Comparable<BookBean>
{
private int code;
public String title;
private int authorCode;
private String category;
private int price;
public BookBean()
{
this.code=0;
this.title="";
this.authorCode=0;
this.category="";
this.price=0;
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
```

```
{
return this.code;
}
public void setTitle(String title)
{
this.title=title;
}
public String getTitle()
{
return this.title;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public void setCategory(String category)
{
this.category=category;
}
public String getCategory()
{
return this.category;
}
public void setPrice(int price)
{
this.price=price;
}
public int getPrice()
{
return this.price;
}
public boolean equals(Object object)
{
if(!(object instanceof BookBean)) return false;
BookBean otherBookBean=(BookBean)object;
return this.code==otherBookBean.code;
}
public int compareTo(BookBean otherBookBean)
{
return this.code-otherBookBean.getCode();
}
public int hashCode()
{
```

```
return this.code;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
public class IfTagHandler extends TagSupport
{
private boolean expression;
public void setExpression(boolean expression)
{
this.expression=expression;
}
public boolean getExpression()
{
return this.expression;
}
public int doStartTag()
{
if(expression)  return EVAL_BODY_INCLUDE;
return SKIP_BODY;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
public class IfModuleTagHandler extends TagSupport
{
public int doStartTag()
{
HttpServletRequest request=(HttpServletRequest)pageContext.getRequest();
if(request.getRequestURI().endsWith("/") || request.getRequestURI().endsWith("/index.jsp")) return
SKIP_BODY;
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import javax.servlet.jsp.tagext.*;
public class FormIdTagHandler extends TagSupport
{
public int doStartTag()
{
try
{
String formId=java.util.UUID.randomUUID().toString();
pageContext.setAttribute(formId,formId,pageContext.SESSION_SCOPE);
pageContext.getOut().write("<input type='hidden' id='formId' name='formId' value='"+formId+"'>");
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.beans.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public class FormResubmittedTagHandler extends TagSupport
{
public int doStartTag()
{
try
{
HttpServletRequest request=(HttpServletRequest)pageContext.getRequest();
String formId=request.getParameter("formId");
if(formId==null)
{
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("You should not refresh / resubmit data.");
notificationBean.setActionText("Ok");
notificationBean.setActionURL("/");
pageContext.setAttribute("notificationBean",notificationBean,PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
HttpSession session=request.getSession();
if(session.getAttribute(formId)==null)
{
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("You should not refresh / resubmit data.");
notificationBean.setActionText("Ok");
notificationBean.setActionURL("/");
pageContext.setAttribute("notificationBean",notificationBean,PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
session.removeAttribute(formId);
}catch(Exception exception)
{
System.out.println(exception); // dont' forget to remove after testing
}
return SKIP_BODY;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
public class EnvironmentVariablesTagHandler extends TagSupport
{
public int doStartTag()
{
pageContext.setAttribute("contextName","crudtwo",PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
public class AuthorTagHandler extends BodyTagSupport
{
private LinkedList<AuthorInterface> authors;
private int index;
private int serialNumber;
private void reset()
{
this.index=0;
this.serialNumber=0;
}
public int doStartTag()
{
this.index=0;
this.serialNumber=0;
try
{
AuthorDAO authorDAO=new AuthorDAO();
authors=authorDAO.getAll();
}catch(DAOException daoException)
{
return SKIP_BODY;
}
if(authors.size()==0) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_INCLUDE;
}
public int doAfterBody()
{
index++;
if(index==authors.size()) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_AGAIN;
}
public int doEndTag()
{
reset();
return EVAL_PAGE;
}
private void setScriptingVariables()
{
AuthorInterface author;
author=authors.get(index);
pageContext.setAttribute("serialNumber",new Integer(serialNumber));
pageContext.setAttribute("code",new Integer(author.getCode()));
pageContext.setAttribute("name",author.getName());
pageContext.setAttribute("evenRow",new Boolean(serialNumber%2==0));
pageContext.setAttribute("oddRow",new Boolean(serialNumber%2!=0));
}
}
```

```
package com.thinking.machines.library.tags;
import javax.servlet.jsp.tagext.*;
public class AuthorTEI extends TagExtraInfo
{
public VariableInfo[] getVariableInfo(TagData tagData)
{
VariableInfo[] variables=new VariableInfo[5];
variables[0]=new VariableInfo("serialNumber","java.lang.Integer",true,VariableInfo.NESTED);
variables[1]=new VariableInfo("code","java.lang.Integer",true,VariableInfo.NESTED);
variables[2]=new VariableInfo("name","java.lang.String",true,VariableInfo.NESTED);
variables[3]=new VariableInfo("evenRow","java.lang.Boolean",true,VariableInfo.NESTED);
variables[4]=new VariableInfo("oddRow","java.lang.Boolean",true,VariableInfo.NESTED);
return variables;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
public class BookTagHandler extends BodyTagSupport
{
private LinkedList<BookInterface> books;
private LinkedList<AuthorInterface> authors;
private int index;
private int serialNumber;
private int authorCode;
private void reset()
{
this.index=0;
this.serialNumber=0;
this.authorCode=0;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public int doStartTag()
```

```
{
this.index=0;
this.serialNumber=0;
try
{
AuthorDAO authorDAO=new AuthorDAO();
authors=authorDAO.getAll();
BookDAO bookDAO=new BookDAO();
if(authorCode==0)
{
books=bookDAO.getAll();
}
else
{
books=bookDAO.getAllByAuthorCode(authorCode);
}
}catch(DAOException daoException)
{
return SKIP_BODY;
}
if(books.size()==0) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_INCLUDE;
}
public int doAfterBody()
{
index++;
if(index==books.size()) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_AGAIN;
}
public int doEndTag()
{
reset();
return EVAL_PAGE;
}
private void setScriptingVariables()
{
BookInterface book;
book=books.get(index);
pageContext.setAttribute("serialNumber",new Integer(serialNumber));
pageContext.setAttribute("code",new Integer(book.getCode()));
pageContext.setAttribute("title",book.getTitle());
pageContext.setAttribute("authorCode",new Integer(book.getAuthorCode()));
pageContext.setAttribute("authorName",new String(getAuthor(book.getAuthorCode()).getName()));
pageContext.setAttribute("category",book.getCategory());
pageContext.setAttribute("price",new Integer(book.getPrice()));
pageContext.setAttribute("evenRow",new Boolean(serialNumber%2==0));
pageContext.setAttribute("oddRow",new Boolean(serialNumber%2!=0));
}
private AuthorInterface getAuthor(int code)
{
for(AuthorInterface author:authors)
{
if(author.getCode()==code) return author;
}
return null; // this case won't arise
}
}
```

```
package com.thinking.machines.library.tags;
import javax.servlet.jsp.tagext.*;
public class BookTEI extends TagExtraInfo
{
public VariableInfo[] getVariableInfo(TagData tagData)
{
VariableInfo[] variables=new VariableInfo[9];
variables[0]=new VariableInfo("serialNumber","java.lang.Integer",true,VariableInfo.NESTED);
variables[1]=new VariableInfo("code","java.lang.Integer",true,VariableInfo.NESTED);
variables[2]=new VariableInfo("title","java.lang.String",true,VariableInfo.NESTED);
variables[3]=new VariableInfo("authorCode","java.lang.Integer",true,VariableInfo.NESTED);
variables[4]=new VariableInfo("authorName","java.lang.String",true,VariableInfo.NESTED);
variables[5]=new VariableInfo("category","java.lang.String",true,VariableInfo.NESTED);
variables[6]=new VariableInfo("price","java.lang.Integer",true,VariableInfo.NESTED);
variables[7]=new VariableInfo("evenRow","java.lang.Boolean",true,VariableInfo.NESTED);
variables[8]=new VariableInfo("oddRow","java.lang.Boolean",true,VariableInfo.NESTED);
return variables;
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
AuthorBean authorBean;
authorBean=(AuthorBean)request.getAttribute("authorBean");
AuthorInterface author=new Author();
author.setName(authorBean.getName());
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.add(author);
authorBean.setCode(author.getCode());
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Author : "+authorBean.getName()+" added successfully.");
notificationBean.setActionText("Add more authors");
notificationBean.setActionURL("/AuthorAddForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorAddForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class EditAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
AuthorBean authorBean;
authorBean=(AuthorBean)request.getAttribute("authorBean");
AuthorInterface author;
AuthorDAO authorDAO=new AuthorDAO();
try
{
author=authorDAO.getByName(authorBean.getName());
authorBean.setCode(author.getCode());
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorUpdateForm.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorEditForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class UpdateAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
AuthorBean authorBean;
authorBean=(AuthorBean)request.getAttribute("authorBean");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
author=authorDAO.getByCode(authorBean.getCode());
}catch(DAOException daoException)
{
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("That author cannot be updated as someone working in parallel deleted
the author while you were trying to update it.");
notificationBean.setActionText("Edit other author");
notificationBean.setActionURL("/AuthorEditForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
return;
}
author.setName(authorBean.getName());
try
{
authorDAO.update(author);
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Author : "+authorBean.getName()+" updated successfully.");
notificationBean.setActionText("Edit more authors");
notificationBean.setActionURL("/AuthorEditForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorUpdateForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class DeleteAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
AuthorBean authorBean=(AuthorBean)request.getAttribute("authorBean");
BookDAO bookDAO=new BookDAO();
try
{
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(authorBean.getCode());
if(bookWithAuthorExists)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage("Cannot delete author as book exists against it.");
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorDeleteForm.jsp");
requestDispatcher.forward(request,response);
return;
}
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
try
{
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.remove(authorBean.getCode());
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Author : "+authorBean.getName()+" deleted successfully.");
notificationBean.setActionText("Delete more authors");
notificationBean.setActionURL("/AuthorDeleteForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
```

```
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class GetAuthorForDeletion extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
AuthorBean authorBean;
authorBean=(AuthorBean)request.getAttribute("authorBean");
AuthorInterface author;
AuthorDAO authorDAO=new AuthorDAO();
try
{
author=authorDAO.getByName(authorBean.getName());
authorBean.setCode(author.getCode());
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorDeleteConfirmationForm.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorDeleteForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class AddBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookInterface book=new Book();
book.setTitle(bookBean.getTitle());
book.setAuthorCode(bookBean.getAuthorCode());
book.setCategory(bookBean.getCategory());
book.setPrice(bookBean.getPrice());
BookDAO bookDAO=new BookDAO();
try
{
bookDAO.add(book);
bookBean.setCode(book.getCode());
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Book : "+bookBean.getTitle()+" added successfully.");
notificationBean.setActionText("Add more books");
notificationBean.setActionURL("/BookAddForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/BookAddForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class EditBook extends HttpServlet
{
```

```
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookInterface book;
BookDAO bookDAO=new BookDAO();
try
{
book=bookDAO.getByCode(bookBean.getCode());
bookBean.setTitle(book.getTitle());
bookBean.setAuthorCode(book.getAuthorCode());
bookBean.setCategory(book.getCategory());
bookBean.setPrice(book.getPrice());
NotificationBean notificationBean;
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/BookUpdateForm.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class UpdateBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookInterface book;
BookDAO bookDAO=new BookDAO();
try
{
book=bookDAO.getByCode(bookBean.getCode());
book.setTitle(bookBean.getTitle());
book.setAuthorCode(bookBean.getAuthorCode());
book.setCategory(bookBean.getCategory());
book.setPrice(bookBean.getPrice());
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage("That book cannot be updated as someone working in parallel book while
you were trying to update it.");
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
return;
}
try
{
bookDAO.update(book);
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Book : "+bookBean.getTitle()+" updated successfully.");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/BookUpdateForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class DeleteBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookDAO bookDAO=new BookDAO();
try
{
bookDAO.remove(bookBean.getCode());
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Book : "+bookBean.getTitle()+" deleted successfully.");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

The following Java Script Files should be in crudtwo\js folder

AuthorAddForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

AuthorEditForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

AuthorUpdateForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

AuthorDeleteForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

BookAddForm.js

```
function validateForm(frm)
{
var title=frm.title.value.trim();
var titleErrorSection=document.getElementById('titleErrorSection');
titleErrorSection.innerHTML='&nbsp;';
var authorCode=frm.authorCode.value;
var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');
authorCodeErrorSection.innerHTML='&nbsp;';
var category=frm.category.value;
var categoryErrorSection=document.getElementById('categoryErrorSection');
categoryErrorSection.innerHTML='&nbsp;';
var price=frm.price.value.trim();
var priceErrorSection=document.getElementById('priceErrorSection');
priceErrorSection.innerHTML='&nbsp;';
var errorComponent=null;
var valid=true;
if(title.length==0)
{
titleErrorSection.innerHTML="Required";
valid=false;
errorComponent=frm.title;
}
if(authorCode==-1)
{
authorCodeErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.authorCode;
}
}
if(category=='none')
{
categoryErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.category;
}
}
if(price.length==0)
{
frm.price.value="0";
}
v="0123456789";
var i=0;
while(i<price.length)
{
if(v.indexOf(price.charAt(i))==-1)
{
priceErrorSection.innerHTML="Invalid";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.price;
}
break;
}
i++;
}
if(!valid) errorComponent.focus();
return valid;
}
```

BookUpdateForm.js

```
function validateForm(frm)
{
var title=frm.title.value.trim();
var titleErrorSection=document.getElementById('titleErrorSection');
titleErrorSection.innerHTML='&nbsp;';
var authorCode=frm.authorCode.value;
var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');
authorCodeErrorSection.innerHTML='&nbsp;';
var category=frm.category.value;
var categoryErrorSection=document.getElementById('categoryErrorSection');
categoryErrorSection.innerHTML='&nbsp;';
var price=frm.price.value.trim();
var priceErrorSection=document.getElementById('priceErrorSection');
priceErrorSection.innerHTML='&nbsp;';
var errorComponent=null;
var valid=true;
if(title.length==0)
{
titleErrorSection.innerHTML="Required";
valid=false;
errorComponent=frm.title;
}
if(authorCode==-1)
{
authorCodeErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.authorCode;
}
}
if(category=='none')
{
categoryErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.category;
}
}
if(price.length==0)
{
frm.price.value="0";
}
v="0123456789";
var i=0;
while(i<price.length)
{
if(v.indexOf(price.charAt(i))==-1)
{
priceErrorSection.innerHTML="Invalid";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.price;
}
break;
}
i++;
}
if(!valid) errorComponent.focus();
return valid;
}
```

Books.js

```
function deleteBook(bookCode,bookTitle)
{
var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');
if(c)
{
var deleteBookForm=document.getElementById('deleteBookForm');
deleteBookForm.code.value=bookCode;
deleteBookForm.title.value=bookTitle;
deleteBookForm.submit();
}
}
```

MasterPageTopSection.js

```
function initializeBody()
{
  var history_api = typeof history.pushState !== 'undefined';
  if(location.hash!=='#library') location.hash='#library';
  if ( location.hash === '#library' ) {
    if ( history_api )
      history.pushState(null, '', '#librarian');
    else
      location.hash = '#librarian';
    window.onhashchange = function() {
      if ( location.hash === '#library' ) {
        if ( history_api )
          history.pushState(null, '', '#librarian');
        else
          location.hash = '#librarian';
      }
    };
  }
}

window.addEventListener("load",initializeBody);
```

Back Bone Complete

The UI Part Starts

Now we are wearing the shoes of a person who doesn't know how to write programming logics, but is well aware of html, designing and know how to follow guidelines.

styles.css (This file should be placed in crudtwo\css folder)

```
body {
background: #F5F5F5;
}

.home
```

```
{
padding: 5px;
}

.header {
background: #DBDBDB;
width:100%;
}

.header table
{
width: 100%;
}

.footer {
background: #DBDBDB;
width:100%;
padding:10px;
text-align: center;
}

.error {
color: red;
}

.evenRow
{
background: #DDDDDD;
}

.oddRow
{
background: #F5F5F5;
}

.author-list {
border:1px solid black;
}

.author-list tr td {
border:1px solid black;
text-align: right;
}

.author-list tr td + td{
border:1px solid black;
text-align: left;
```

```
}

.author-list tr td + td + td{
border:1px solid black;
text-align: right;
}

.book-list {
border:1px solid black;
}

.book-list tr td {
border:1px solid black;
text-align: right;
}

.book-list tr td + td{
border:1px solid black;
text-align: left;
}

.book-list tr td + td + td{
border:1px solid black;
text-align: right;
}

.book-list tr td + td + td + td{
border:1px solid black;
text-align: left;
}

.book-list tr td + td + td + td + td{
border:1px solid black;
text-align: left;
}

.book-list tr td + td + td + td + td + td{
border:1px solid black;
text-align: right;
}

.book-list tr td + td + td + td + td + td + td{
border:1px solid black;
text-align: center;
}

.book-list tr td + td + td + td + td + td + td + td{
```

```
border:1px solid black;
text-align: center;
}
```

MasterPageTopSection.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
</head>
<body>
<div class='header'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
<a href='/${contextName}/'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp

```
<div class='footer'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

index.jsp

```
<jsp:include page='/MasterPageTopSection.jsp' />
<h3>Author</h3>
<ul>
<li><a href='/${contextName}/AuthorAddForm.jsp'>Add</a></li>
<li><a href='/${contextName}/AuthorEditForm.jsp'>Edit</a></li>
<li><a href='/${contextName}/AuthorDeleteForm.jsp'>Delete</a></li>
<li><a href='/${contextName}/Authors.jsp'>List</a></li>
</ul>
```

```
<h3>Book</h3>
<ul>
<li><a href='/${contextName}/BookAddForm.jsp'>Add</a></li>
<li><a href='/${contextName}/Books.jsp'>Edit/Delete/List</a></li>
</ul>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

AuthorAddForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorAddForm.js'></script>
<h2>Add Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/AddAuthor.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Add</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

AddAuthor.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='authorBean' scope='request'
```

```
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/addAuthor' />
```

AuthorEditForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorEditForm.js'></script>
<h2>Edit Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/EditAuthor.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Edit</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

EditAuthor.jsp

```
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/editAuthor' />
```

AuthorUpdateForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
```

```
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorUpdateForm.js'></script>
<h2>Edit Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/UpdateAuthor.jsp' onsubmit='return validateForm(this)'
method='post'>
<tm:FormId/>
<input type='hidden' name='code' id='code' value='${authorBean.code}'>
<table border='0'>
<tr>
<td>Code</td>
<td>
<b>${authorBean.code}</b>
</td>
</tr>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Update</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

UpdateAuthor.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
```

```
<jsp:forward page='/updateAuthor' />
```

AuthorDeleteForm.jsp

```
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorDeleteForm.js'></script>
<h2>Delete Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/GetAuthorForDeletion.jsp' onsubmit='return validateForm(this)'
method='post'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Delete</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

GetAuthorForDeletion.jsp

```
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/getAuthorForDeletion' />
```

AuthorDeleteConfirmationForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<h2>Delete Author </h2>
```

```
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/DeleteAuthor.jsp' onsubmit='return validateForm(this)'
method='post'>
<tm:FormId/>
<input type='hidden' id='code' name='code' value='${authorBean.code}' >
<input type='hidden' id='name' name='name' value='${authorBean.name}' >
<table border='0'>
<tr>
<td>Code</td>
<td>
${authorBean.code}
</td>
</tr>
<tr>
<td>Name of author</td>
<td>
${authorBean.name}
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Delete</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

DeleteAuthor.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/deleteAuthor' />
```

Authors.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<h2>Authors</h2>
<table class='author-list'>
<thead>
<tr>
```

```
<th>S.No.</th>
<th>Author</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tm:Author>
<tm:If expression='${evenRow}'>
<tr class='evenRow'>
</tm:If>
<tm:If expression='${oddRow}'>
<tr class='oddRow'>
</tm:If>
<td>${serialNumber}</td>
<td>${name}</td>
<td>${code}</td>
</tr>
</tm:Author>
</tbody>
</table>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

BookAddForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/BookAddForm.js'></script>
<h2>Add Book </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/AddBook.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<table border='0'>
<tr>
<td>Title</td>
<td>
<input type='text' name='title' id='title' maxlength='35' size='31' value='${bookBean.title}'>
</td>
</tr>
<tr>
<td></td><td>
<span id='titleErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
```

```
<tr>
<td>Author</td>
<td>
<select id='authorCode' name='authorCode'>
<option value='-1'>&lt;Select&gt;</option>
<tm:Author>
<tm:If expression='${bookBean.authorCode==code}'>
<option value='${code}' selected>${name}</option>
</tm:If>
<tm:If expression='${bookBean.authorCode!=code}'>
<option value='${code}'>${name}</option>
</tm:If>
</tm:Author>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='authorCodeErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'>
<option value='none'>&lt;Select&gt;</option>
<option value='Science fiction' ${(bookBean.category=="Science fiction")?"selected":""}>Science
fiction</option>
<option value='Satire' ${(bookBean.category=="Satire")?"selected":""}>Satire</option>
<option value='Drama' ${(bookBean.category=="Drama")?"selected":""}>Drama</option>
<option value='Action and Adventure' ${(bookBean.category=="Action and
Adventure")?"selected":""}>Action and Adventure</option>
<option value='Mystery' ${(bookBean.category=="Mystery")?"selected":""}>Mystery</option>
<option value='Horror' ${(bookBean.category=="Horror")?"selected":""}>Horror</option>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='categoryErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Price</td>
<td>
<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right' value='$
{bookBean.price}'>
```

```
</td>
</tr>
<tr>
<td></td><td>
<span id='priceErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Add</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

AddBook.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/addBook' />
```

Books.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:useBean id='notificationBean' scope='request'
class='com.thinking.machines.library.beans.NotificationBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/js/Books.js'></script>
<h2>Books</h2>
<tm:If expression='${notificationBean.message!=""}'>
<h3>${notificationBean.message}</h3>
</tm:If>
<div class='error'>${errorBean.errorMessage}</div>
<table class='book-list'>
<thead>
<tr>
<th>S.No.</th>
<th>Title</th>
<th>Code</th>
<th>Author</th>
<th>Category</th>
```

```
<th>Price</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tm:Book>
<tm:If expression='${evenRow}'>
<tr class='evenRow'>
</tm:If>
<tm:If expression='${oddRow}'>
<tr class='oddRow'>
</tm:If>
<td>${serialNumber}</td>
<td>${title}</td>
<td>${code}</td>
<td>${authorName}</td>
<td>${category}</td>
<td>${price}</td>
<td><a href='/${contextName}/EditBook.jsp?code=${code}'><img src='/$
{contextName}/images/edit_icon.png' style='padding:5px'></a></td>
<td><a href='javascript:deleteBook(${code},"${title}")'><img src='/$
{contextName}/images/delete_icon.png' style='padding:5px'></a></td>
</tr>
</tm:Book>
</tbody>
</table>
<form id='deleteBookForm' action='/${contextName}/DeleteBook.jsp' method='POST'>
<tm:FormId/>
<input type='hidden' name='code' id='code'>
<input type='hidden' name='title' id='title'>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

EditBook.jsp

```
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/editBook' />
```

BookUpdateForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/BookUpdateForm.js'></script>
```

```
<h2>Edit Book </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/UpdateBook.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<input type='hidden' id='code' name='code' value='${bookBean.code}' >
<table border='0'>
<tr>
<td>Title</td>
<td>
<input type='text' name='title' id='title' maxlength='35' size='31' value='${bookBean.title}'>
</td>
</tr>
<tr>
<td></td><td>
<span id='titleErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Author</td>
<td>
<select id='authorCode' name='authorCode'>
<option value='-1'>&lt;Select&gt;</option>
<tm:Author>
<tm:If expression='${bookBean.authorCode==code}'>
<option value='${code}' selected>${name}</option>
</tm:If>
<tm:If expression='${bookBean.authorCode!=code}'>
<option value='${code}'>${name}</option>
</tm:If>
</tm:Author>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='authorCodeErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'>
<option value='none'>&lt;Select&gt;</option>
<option value='Science fiction' ${(bookBean.category=="Science fiction")?"selected":""}>Science
fiction</option>
<option value='Satire' ${(bookBean.category=="Satire")?"selected":""}>Satire</option>
<option value='Drama' ${(bookBean.category=="Drama")?"selected":""}>Drama</option>
```

```
<option value='Action and Adventure' ${(bookBean.category=="Action and
Adventure")?"selected":""}>Action and Adventure</option>
<option value='Mystery' ${(bookBean.category=="Mystery")?"selected":""}>Mystery</option>
<option value='Horror' ${(bookBean.category=="Horror")?"selected":""}>Horror</option>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='categoryErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Price</td>
<td>
<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right' value='$
{bookBean.price}'>
</td>
</tr>
<tr>
<td></td><td>
<span id='priceErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Update</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

UpdateBook.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/updateBook' />
```

DeleteBook.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
```

```
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/deleteBook' />
```

Notification.jsp

```
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='notificationBean' scope='request'
class='com.thinking.machines.library.beans.NotificationBean' />
<h2>Notification</h2>
<div style='font-size:24pt'>
<b>${notificationBean.message}</b>
<br><br>
<form action='/${contextName}${notificationBean.actionURL}'>
<Button type='submit'>${notificationBean.actionText}</button>
</form>
</div>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

Note : We will be changing the Books List to incorporate feature to filter list of books by author in the classroom session.

--- CRUD Two Complete ---

Before going on to CRUD Three, we need to learn how to implement AJAX

We will be learning 3 examples
a) using xmlHttpObject
b) using Jquery
c) building a wrapper like Jquery $.ajax

I will roll all into one web application

create a web app folder named as ajax.com

create the following servlet

```
package thinking.machines;
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class GetSquareAndFactorial extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
        try
        {
            PrintWriter pw=response.getWriter();
            int num=Integer.parseInt(request.getParameter("num"));
            int s=num*num;
            int e,f;
            e=f=1;
            while(e<=num)
            {
            f=f*e;
            e++;
            }
            pw.print(s+","+f);
        }catch(Exception ex)
        {
            System.out.println(ex);
        }
    }
    }
```

Following is the servlet mapping that needs to be done in the deployment descriptor.

```
    <servlet>
    <servlet-name>GetSquareAndFactorial</servlet-name>
    <servlet-class>thinking.machines.GetSquareAndFactorial</servlet-class>
    </servlet>
    <servlet-mapping>
    <servlet-name>GetSquareAndFactorial</servlet-name>
    <url-pattern>/GetSquareAndFactorial</url-pattern>
    </servlet-mapping>
    </web-app>
```

---

index.html

```
<html>
<head>
<title>
</title>
<script language='JavaScript'>
var xmlHttpObject=null;
function setupXMLHttpObject()
{
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlHttpObject=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlHttpObject=new ActiveXObject("Microsoft.XMLHTTP");
  }
}

function computeSquareAndFactorial()
{
if(xmlHttpObject==null)
{
setupXMLHttpObject()
}
var num=document.getElementById("myForm").num.value;
xmlHttpObject.onreadystatechange=jawabAaGaya;
xmlHttpObject.open("GET","GetSquareAndFactorial?num="+num,true);
xmlHttpObject.send();
}

function jawabAaGaya()
{
  if (xmlHttpObject.readyState==4 )
{
if(xmlHttpObject.status==200)
    {
    var res=xmlHttpObject.responseText.split(",");
    document.getElementById("squareDivision").innerHTML="Square : "+res[0];
    document.getElementById("factorialDivision").innerHTML="Factorial : "+res[1];
    }
else
{
alert("Invalid request or server not ready");
}
}
}

</script>
</head>
<body>
<h2>AJAX Example One</h2>
<form id='myForm' name='myForm'>
Enter number
<input type='text' name='num' id='num'>
<input type='button' value='Compute' onclick='computeSquareAndFactorial()'>
</form>
<div id='squareDivision'>
</div>
```

```
</body>
<br/>
<div id='factorialDivision'>
</div>
<br>
<br>
<br>
<a href='jqueryajax.html'>AJAX Example Two</a><br>
<a href='tmjsajax.html'>AJAX Example Three</a><br>
</html>
```

Download jquery as discussed in the classroom session, and place the folder containg the jquery js file in the ajax.com folder, in the following html, replace the jquery js file name with the one that you have. I have highlighted it.

jqueryajax.html

```
<html>
<head>
<title>
</title>
<script src='jquery3.1.1/jquery-3.1.1.min.js'></script>
<script>
function computeSquareAndFactorial()
{
$.ajax({
  type: 'GET',
  url: 'GetSquareAndFactorial',
  data: {
    num : $("#num").val()
  },
  success: function(data) {
   var res=data.split(",");
   document.getElementById("squareDivision").innerHTML="Square : "+res[0];
   $("#factorialDivision").html("Factorial : "+res[1]);
  },
  error: function() {
    alert('Unable to send request, server not ready......');
  }
});
}
</script>
</head>
<body>
<h2>AJAX Example</h2>
<form id='myForm' name='myForm'>
Enter number
<input type='text' name='num' id='num'>
```

```
<input type='button' value='Compute' onclick='computeSquareAndFactorial()'>
</form>
<div id='squareDivision'>
</div>
</body>
<br/>
<div id='factorialDivision'>
</div>
<br><br><br>
<a href='index.html'>AJAX Example One</a><br>
<a href='tmjsajax.html'>AJAX Example Three</a><br>

</html>
```

Now create a folder named as tmjs-1
in it create the following file tmjs-1.js as follows

tmjs-1.js

```
function setupXMLHttpObject()
{
var xmlHttpObject=null;
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlHttpObject=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlHttpObject=new ActiveXObject("Microsoft.XMLHTTP");
  }
return xmlHttpObject;
}
function jsonToQueryString(json)
{
return '?' +
Object.keys(json).map(function(key) {
return encodeURIComponent(key) + '=' +
encodeURIComponent(json[key]);
}).join('&');
}

function AjaxHandler()
{
var xmlHttpObject; // a private property
this.ajax=function(object)
{
if(xmlHttpObject==null)  xmlHttpObject=setupXMLHttpObject();
xmlHttpObject.onreadystatechange=function(){
 if (xmlHttpObject.readyState==4 )
{
if(xmlHttpObject.status==200)
{
object.success(xmlHttpObject.responseText);
}
else
{
if(object.error) object.error();
}
}
};
if(!object.type) object.type='GET';
if(!object.url) return;
if(object.data && object.type=='GET')
{
var qs=jsonToQueryString(object.data);
xmlHttpObject.open(object.type,object.url+qs,true);
}
if(!object.data && object.type=='GET')
{
xmlHttpObject.open(object.type,object.url,true);
}
// many more if conditions to handle other types of request
xmlHttpObject.send();
};
}

var $=new AjaxHandler();
```

---

tmjsajax.html

```
<html>
<head>
<title>
</title>
<script src='tmjs-1/tmjs-1.js'></script>
<script>
function computeSquareAndFactorial()
{
$.ajax({
  type: 'GET',
  url: 'GetSquareAndFactorial',
  data: {
    num : document.getElementById("num").value
  },
  success: function(data) {
   var res=data.split(",");
```

```
    document.getElementById("squareDivision").innerHTML="Square : "+res[0];
    document.getElementById("factorialDivision").innerHTML="Square : "+res[1];
   },
   error: function() {
     alert('Unable to send request, server not ready......');
   }
});
}
</script>
</head>
<body>
<h2>AJAX Example</h2>
<form id='myForm' name='myForm'>
Enter number
<input type='text' name='num' id='num'>
<input type='button' value='Compute' onclick='computeSquareAndFactorial()'>
</form>
<div id='squareDivision'>
</div>
</body>
<br/>
<div id='factorialDivision'>
</div>
</html>
```

CRUD Three

AJAX
Zero libraries
The learning will be incremental fashion

This is the first learning phase, where we will try to understand how to create UI for performing all the CRUD Operations in AJAX Style without using any third party library.

I am assuming that for this example, we know the resolution of the clients monitor and it won't change. As if we are writing the web application to be used in an organization for its process management.

Create a webapplication folder named as crudthree
in it create folders named as css, images, js, tm-js-1,WEB-INF

tm-js-1.css (in tm-js-1 folder)

```
.progressModalMask
{
position:absolute;
left:0;
top:0;
width:100%;
height:100%;
background: url('../tm-js-1/progressMask.png');
visibility:hidden;
opacity: 50%;
text-align:center;
}
.progressIndicator
{
position: absolute;
top:0;
bottom: 0;
left: 0;
right: 0;
margin: auto;
}
.forModal
{
visibility: hidden;
}
.modalMask
{
position:absolute;
left:0;
top:0;
width:100%;
height:100%;
background: url('../tm-js-1/mask.png');
visibility:hidden;
opacity: 50%;
}
.modal
```

```
{
border : 1px solid black;
height: 100px;
width: 100px;
position: absolute;
left:50%;
top: 10%;
z-index: 10000;
background: #FFFFFF;
border-radius: 0px;
box-shadow: 5px 5px 3px grey;
}
.modalHeader
{
background: #2E2E2E;
margin: 1px;
padding: 5px;
color: #FFFFFF;
font-size: 10pt;
height: 18px;
}
.modalClose
{
float: right;
cursor: pointer;
}
.modalTitle
{
float: left;
}
.modalFooter
{
background: #DFDFDF;
margin: 1px;
padding: 6px;
font-size: 8pt;
height: 10px;
text-align: center;
vertical-align: middle;
}
.modalContainer
{
margin: 1px;
padding:10px;
font-size:12pt;
overflow: auto;
}
```

tm-js-1.js (in tm-js-1 folder)

```
function getXMLHttpObject()
{
var xmlHttpObject=null;
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlHttpObject=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlHttpObject=new ActiveXObject("Microsoft.XMLHTTP");
  }
return xmlHttpObject;
}
function jsonToQueryString(json)
{
return '?' +
Object.keys(json).map(function(key) {
return encodeURIComponent(key) + '=' +
encodeURIComponent(json[key]);
}).join('&');
}

function AjaxHandler()
{
var xmlHttpObject; // a private property
this.ajax=function(object)
{
if(xmlHttpObject==null)  xmlHttpObject=getXMLHttpObject();
xmlHttpObject.onreadystatechange=function(){
 if (xmlHttpObject.readyState==4 )
{
if(xmlHttpObject.status==200)
{
object.success(xmlHttpObject.responseText);
}
else
{
if(object.error) object.error();
}
}
};
if(!object.type) object.type=object.type.toUpperCase();
if(!object.type) object.type='GET';
if(!object.url) return;
if(object.data && object.type=='GET')
{
var qs=jsonToQueryString(object.data);
xmlHttpObject.open(object.type,object.url+qs,true);
xmlHttpObject.send();
}else if(!object.data && object.type=='GET')
{
xmlHttpObject.open(object.type,object.url,true);
xmlHttpObject.send();
} else if(!object.data && object.type=='POST')
{
xmlHttpObject.open(object.type,object.url, true);
xmlHttpObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttpObject.send();
}else if(object.data && object.type=='POST')
{
xmlHttpObject.open(object.type,object.url, true);
xmlHttpObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttpObject.send(jsonToQueryString(object.data).substring(1));
}
};
}
function showProgressModal()
{
var pmm=document.getElementsByClassName("progressModalMask")[0];
pmm.style.visibility="visible";
}
function hideProgressModal()
{
var pmm=document.getElementsByClassName("progressModalMask")[0];
pmm.style.visibility="hidden";
}
function showModal(divisionToBeFixedInModal,width,height,title,footer)
{
document.body.scrollTop=document.documentElement.scrollTop=0;
var modalWrapperMask=document.getElementById('modalWrapperMask');
var modalWrapper=document.getElementById("modalWrapper");
modalWrapper.style.width=width;
modalWrapper.style.height=height;
var widthValue=parseInt(modalWrapper.style.width);
var widthUOM=modalWrapper.style.width.substring(modalWrapper.style.width.length-2);
modalWrapper.style.marginLeft=((widthValue/2)*(-1))+"px";
var heightValue=parseInt(modalWrapper.style.height);
var heightUOM=modalWrapper.style.height.substring(modalWrapper.style.height.length-2);
var modalWrapperContainer=document.getElementById("modalWrapperContainer");
modalWrapperContainer.style.height=(heightValue-18-18-38)+heightUOM;  // 18 header, 18 footer, 38
shadow
```

```
var modalWrapperCloseIcon=document.getElementById("modalWrapperCloseIcon");
if(title)
{
var modalWrapperTitle=document.getElementById("modalWrapperTitle");
modalWrapperTitle.innerHTML=title;
}
if(footer)
{
var modalWrapperFooter=document.getElementById("modalWrapperFooter");
modalWrapperFooter.innerHTML=footer;
}
var divToFix=document.getElementById(divisionToBeFixedInModal);
divToFix.remove();
var clone=divToFix.cloneNode(true);
clone.style.visibility="visible";
modalWrapperContainer.append(clone);
modalWrapperContainer.fixedComponent=divToFix;
modalWrapperContainer.clonedComponent=clone;
modalWrapperMask.style.visibility="visible";
}
function disposeModal()
{
var modalWrapperMask=document.getElementById('modalWrapperMask');
var modalWrapperContainer=document.getElementById("modalWrapperContainer");
var divToUnfix=modalWrapperContainer.clonedComponent;
divToUnfix.remove();
modalWrapperMask.style.visibility="hidden";
document.body.appendChild(modalWrapperContainer.fixedComponent);
}
function setupModals()
{
var progressModalWrapperMask=document.createElement("div");
progressModalWrapperMask.className="progressModalMask";
var img=document.createElement("img");
img.className='progressIndicator';
img.src="tm-js-1/loading.gif"
progressModalWrapperMask.appendChild(img);
document.getElementsByTagName('body')[0].appendChild(progressModalWrapperMask);
progressModalWrapperMask.style.zIndex = "2000"
// Following is what we created and appended it to body of the page
/*<div id='progressModalWrapperMask' class='progressModalMask'>
<img src='images/loading.gif' class='progressIndicator'></img>
</div>*/
var modalWrapperMask=document.createElement("div");
modalWrapperMask.className="modalMask";
modalWrapperMask.id="modalWrapperMask";
var modalWrapper=document.createElement("div");
```

```
modalWrapper.id="modalWrapper";
modalWrapper.className="modal";
modalWrapperMask.appendChild(modalWrapper);
var modalWrapperHeader=document.createElement("div");
modalWrapperHeader.className="modalHeader";
modalWrapperHeader.id="modalWrapperHeader";
modalWrapper.appendChild(modalWrapperHeader);
var modalWrapperTitle=document.createElement("div");
modalWrapperTitle.className="modalTitle";
modalWrapperTitle.id="modalWrapperTitle";
modalWrapperHeader.appendChild(modalWrapperTitle);
var modalWrapperCloseIcon=document.createElement("div");
modalWrapperCloseIcon.className="modalClose";
modalWrapperCloseIcon.id="modalWrapperCloseIcon";
modalWrapperCloseIcon.onclick=disposeModal;
var closeIconImage=document.createElement("img");
closeIconImage.src='tm-js-1/close_modal_icon.png';
modalWrapperCloseIcon.appendChild(closeIconImage);
modalWrapperHeader.appendChild(modalWrapperCloseIcon);
var modalWrapperContainer=document.createElement("div");
modalWrapperContainer.className="modalContainer";
modalWrapperContainer.id="modalWrapperContainer";
modalWrapper.appendChild(modalWrapperContainer);
var modalWrapperFooter=document.createElement("div");
modalWrapperFooter.className="modalFooter";
modalWrapperFooter.id="modalWrapperFooter";
modalWrapper.appendChild(modalWrapperFooter);
document.getElementsByTagName('body')[0].appendChild(modalWrapperMask);
modalWrapperMask.style.zIndex = "1000"
// Following is what we created and appended it to body of the page
/*
<div id='modalWrapperMask' class='modalMask'>
<div id='modalWrapper' class='modal'>
<div id='modalWrapperHeader' class='modalHeader'>
<div id='modalWrapperTitle' class='modalTitle'></div>
<div id='modalWrapperCloseIcon' class='modalClose' onClick='disposeModal()'>
<img src='tm-js-1/close_modal_icon.png'></img>
</div>
</div>
<div id='modalWrapperContainer' class='modalContainer'>
</div>
<div id='modalWrapperFooter' class='modalFooter'>
</div>
</div>
</div>*/
}
```

```
// following will setup the function to be called on window load event
window.onload=setupModals;

// following is the declaration of a global varable named as dollar ($)
// The name dollar ($) is just to give you an insight of how JQuery must have done things
// Note : My code is nowhere near to what JQuery creators have done, it is just an introduction

var $=new AjaxHandler();
```

Download tm-js-resources.zip from http://tm-certificates.com/tm-js-resources.zip. Unzip and place the images in tm-js-1 folder

---

Place delete_icon.png,edit_icon.png and logo.png in images folder (Refer crudone)

---

styles.css (place it in css folder)

```
body {
background: #F5F5F5;
}

.home
{
padding: 5px;
}

.header {
background: #DBDBDB;
width:100%;
}

.header table
{
width: 100%;
}

.footer {
background: #DBDBDB;
width:100%;
padding:10px;
text-align: center;
}

.error {
color: red;
}
```

```
.evenRow
{
background: #DDDDDD;
}

.oddRow
{
background: #F5F5F5;
}

.author-list {
border:1px solid black;
}

.author-list tr td {
border:1px solid black;
text-align: right;
}

.author-list tr td + td{
border:1px solid black;
text-align: left;
}

.author-list tr td + td + td{
border:1px solid black;
text-align: right;
}

.book-list {
border:1px solid black;
}

.book-list tr td {
border:1px solid black;
text-align: right;
}

.book-list tr td + td{
border:1px solid black;
text-align: left;
}

.book-list tr td + td + td{
border:1px solid black;
text-align: right;
}

.book-list tr td + td + td + td{
border:1px solid black;
text-align: left;
}

.book-list tr td + td + td + td + td{
border:1px solid black;
text-align: left;
}

.book-list tr td + td + td + td + td + td{
border:1px solid black;
text-align: right;
}

.book-list tr td + td + td + td + td + td + td{
border:1px solid black;
text-align: center;
}

.book-list tr td + td + td + td + td + td + td + td{
border:1px solid black;
text-align: center;
}
```

MasterPageTopSection.js (in js folder)

```
function initializeBody()
{
   var history_api = typeof history.pushState !== 'undefined';
   if(location.hash!=='#library') location.hash='#library';
   if ( location.hash === '#library' ) {
      if ( history_api )
         history.pushState(null, '', '#librarian');
      else
         location.hash = '#librarian';
      window.onhashchange = function() {
         if ( location.hash === '#library' ) {
            if ( history_api )
               history.pushState(null, '', '#librarian');
            else
               location.hash = '#librarian';
         }
      };
   }
}

window.addEventListener("load",initializeBody);
```

MasterPageTopSection.jsp (in crudthree folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/tm-js-1/tm-js-1.css' >
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
</head>
<body>
<div class='header'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
<a href='/${contextName}/'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp (in crudthree folder)

```
<div class='footer'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

index.jsp (in crudthree folder)

```
<jsp:include page='/MasterPageTopSection.jsp' />
<ul>
<li><a href='/${contextName}/Authors.jsp'>Authors</a></li>
<li><a href='/${contextName}/Books.jsp'>Books</a></li>
</ul>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

Authors.jsp (in crudthree folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/tm-js-1/tm-js-1.js'></script>
<script>
function clearAuthors()
{
var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
while(authorsListBody.rows.length>0)
{
authorsListBody.deleteRow(0);
}
var row=authorsListBody.insertRow(authorsListBody.rows.length);
var cell1=row.insertCell(0);
cell1.innerHTML="&nbsp";
var cell2=row.insertCell(1);
cell2.innerHTML="&nbsp";
var cell3=row.insertCell(2);
cell3.innerHTML="&nbsp";
var cell4=row.insertCell(3);
cell4.innerHTML="&nbsp";
}
function loadAuthors()
{
showProgressModal();
clearAuthors();
$.ajax({
  type: 'GET',
  url: 'getAuthors',
  success: function(data) {
  var spl=data.split(",")
  var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
  while(authorsListBody.rows.length>0) authorsListBody.deleteRow(0);
  var i=0;
  var serialNumber=0;
  while(i<spl.length)
  {
  serialNumber++;
  var row=authorsListBody.insertRow(authorsListBody.length);
  var cell1=row.insertCell(0);
  var cellText=document.createTextNode(serialNumber+".");
  cell1.appendChild(cellText);
  var cell2=row.insertCell(1);
  cellText=document.createTextNode(spl[i+1]);
  cell2.appendChild(cellText);
  var cell3=row.insertCell(2);
  cellText=document.createTextNode("Edit");
  cell3.appendChild(cellText);
  var cell4=row.insertCell(3);
  cellText=document.createTextNode("Delete");
  cell4.appendChild(cellText);
  i+=2;
  }
  hideProgressModal();
  },
  error: function() {
  hideProgressModal();
  alert('Unable to send request, server not ready......');
  }
});
}
function addAuthor()
{
var authorAddForm=document.getElementById("authorAddForm");
var name=authorAddForm.name.value.trim();
var nameErrorSection=document.getElementById("authorAddFormNameErrorSection");
nameErrorSection.innerHTML="&nbsp;";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
authorAddForm.name.focus();
return;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
authorAddForm.name.focus();
return;
}
e++;
}
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'addAuthor',
    data: {
    name : name
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
   hideProgressModal();
   disposeModal();
   loadAuthors();
  }
  else
  {
  hideProgressModal();
  var formErrorSection=document.getElementById("authorAddFormErrorSection");
  formErrorSection.innerHTML=spl[1];
  }
  },
  error: function() {
    hideProgressModal();
    alert('Unable to send request, server not ready......');
  }
});
}
window.addEventListener('load',loadAuthors);
</script>
<style>
.actionIcon
{
cursor:pointer;
}
.listContainer
{
text-align:right;
}
.authorsListView
{
border: 1px solid black;
border-spacing: 0px;
border-collapse: collapse;
width: 600px
font-size: 12pt;
}
.authorsListView thead
{
background-color: lightgray;
}
.authorsListView th
{
border: 1px solid black;
}
.authorsListView td
{
```

```
border: 1px solid black;
text-align:right;
width: 100px;
}
.authorsListView td+td
{
border: 1px solid black;
width:300px;
text-align:left;
}
.authorsListView td+td+td
{
border: 1px solid black;
text-align: center;
width:100px;
}
.authorsListView td+td+td+td
{
border: 1px solid black;
width:100px;
}
.authorsListLoader
{
text-align: center;
}
</style>
<h2>Authors</h2>
<table class='listContainer'>
<tr>
<td>
<img src='/${contextName}/images/add_icon.png' class='actionIcon'
onclick='showModal("authorAddFormContainer","400px","200px","Add author","")'></img>
</td>
</tr>
<tr>
<td>
<table id='authorsList' class='authorsListView'>
<thead>
<tr>
<th>S.No.</th>
<th>Author</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
</tbody>
```

```
</table>
</td>
</tr>
</table>
<br>
<br>
<br>
<!-- Module Container For Modal Window Start -->
<div id='authorAddFormContainer' class='forModal'>
<span id='authorAddFormErrorSection' class='error'>&nbsp;</span>
<form id='authorAddForm'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' >
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='authorAddFormNameErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='button' onclick='addAuthor()'>Add</button>
</td>
</tr>
</table>
</form>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

The WEB-INF folder should have tlds, lib and classes folder

the lib folder should contain mysql.jar

CustomTags.tld (in tlds folder)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!DOCTYPE taglib
      PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
      "http://java.sun.com/j2ee/dtd/web-jsptaglibrary_1_2.dtd">

<taglib>

  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>CustomTags</short-name>
  <description>
   A tag library for this project
  </description>
  <tag>
    <name>SetEnvironmentVariables</name>
    <tag-class>com.thinking.machines.library.tags.EnvironmentVariablesTagHandler</tag-class>
    <description> Environment Variables Tag Handler </description>
  </tag>
  <tag>
    <name>FormId</name>
    <tag-class>com.thinking.machines.library.tags.FormIdTagHandler</tag-class>
    <description> Form Id. Generator Tag </description>
  </tag>
  <tag>
    <name>FormResubmitted</name>
    <tag-class>com.thinking.machines.library.tags.FormResubmittedTagHandler</tag-class>
    <description> Process body if form is resubmitted </description>
  </tag>
  <tag>
    <name>IfModule</name>
    <tag-class>com.thinking.machines.library.tags.IfModuleTagHandler</tag-class>
    <description> Tag to identify module page </description>
  </tag>
  <tag>
    <name>If</name>
    <tag-class>com.thinking.machines.library.tags.IfTagHandler</tag-class>
    <description> Tag to identify module page </description>
    <attribute>
    <name>expression</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    </attribute>
  </tag>
  <tag>
    <name>Author</name>
    <tag-class>com.thinking.machines.library.tags.AuthorTagHandler</tag-class>
    <tei-class>com.thinking.machines.library.tags.AuthorTEI</tei-class>
    <description> Author Tag Handler </description>
  </tag>
  <tag>
    <name>Book</name>
    <tag-class>com.thinking.machines.library.tags.BookTagHandler</tag-class>
    <tei-class>com.thinking.machines.library.tags.BookTEI</tei-class>
    <description> Book Tag Handler </description>
    <attribute>
    <name>authorCode</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
    </attribute>
  </tag>
</taglib>
```

web.xml should contain following servlet mappings (in WEB-INF folder)

```
<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>GetAuthors</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthors</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthors</servlet-name>
<url-pattern>/getAuthors</url-pattern>
</servlet-mapping>
```

create the following structure in classes folder com\thinking\machines\library

in classes, create servlets, dl, beans and tags folder
copy all the classes from crudtwo for dl and beans folder
copy all the classes from crudtwo for tags folder and make one small change in the following class

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
public class EnvironmentVariablesTagHandler extends TagSupport
{
public int doStartTag()
{
pageContext.setAttribute("contextName","crudthree",PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

Following are the servlets to be created in the servlets folder

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetAuthors extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
Thread.sleep(1500);  // remove this line after testing, this is just to Stimulate that the server is taking time to fetch records
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
boolean applyComma=false;
for(AuthorInterface author:authors)
{
if(applyComma) pw.print(",");
pw.print(author.getCode()+","+author.getName());
applyComma=true;
}
}catch(DAOException daoException)
{
pw.print("false,No authors");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
Thread.sleep(1500);
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorInterface author=new Author();
author.setName(name);
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.add(author);
pw.print("true,Author added");
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

Problems pending

The table has no scroll bars and it will become endless.

The newly added row doesn't get selected
On every addition operation, the table is being populated from zero

Assignment : Replace the words Delete and Edit with appropriate icons and add the necessary functionality against it as discussed in the classroom session.

Solutions to the pending problems

Note : In my javascript code I have called the remove function and append function and in some cases appendChild. In on some browsers remove/append function won't work.

According to DOM level 4 specs, which is the current version in development, there are some new handy mutation methods available: append(), prepend(), before(), after(), replace(), and remove().

For example in the function (showModal), I have placed the following code.

divToFix.remove();
modalWrapperContainer.append(clone);

If it doesn't work, you will have to replace it with

divToFix.parentNode.removeChild(divToFix);
modalWrapperContainer.appendChild(clone);

Now let us solve the pending problems.

Adding scrollbars to the table & making a row selectable

replace the **style** section with the following **style** section in **Authors.jsp**

```
<style>
.selectedRow
{
background-color:black;
color: white;
}
.actionIcon
{
cursor:pointer;
}
.listContainer
{
text-align:right;
}


.authorsListView
{
width: 600px;
```

```
border-collapse: collapse;
border-spacing: 0;
border: 2px solid black;
}
.authorsListView  thead
{
display: block;
}
.authorsListView thead tr th
{
height: 30px;
width: 50px;
line-height: 30px;
text-align: right;
border-right: 1px solid black;
}
.authorsListView thead tr th+th
{
width: 410px;
text-align: left;
}
.authorsListView thead tr th+th+th
{
width: 50px;
text-align: center;
}
.authorsListView thead tr th+th+th+th
{
width: 50px;
text-align: center;
border-right: none;
}
.authorsListView tbody
{
display: block;
width: 600px;
height: 300px;
overflow-y: auto;
overflow-x: hidden;
border-top: 2px solid black;
cursor: pointer;
}
.authorsListView tbody tr td
{
width: 50px;
text-align: right;
border-right: 1px solid black;
```

```
}
.authorsListView tbody tr td+td
{
width: 410px;
text-align: left;
}
.authorsListView tbody tr td+td+td
{
width: 50px;
text-align: center;
}
.authorsListView tbody tr td+td+td+td
{
width: 50px;
text-align: center;
border-right: 1px solid black;
}
.authorsListLoader
{
text-align: center;
}
</style>
```

Now lets make the row selectable.

In the script section, just above the definition of clearAuthors insert the following

```
var selectedRow=null;
function selectRow(row)
{
if(selectedRow)
{
selectedRow.classList.remove('selectedRow');
}
selectedRow=row;
selectedRow.classList.add('selectedRow');
}
```

Now the table will be of a fixed height with scrollbars to scroll and the user can click and select row.

Now let us change the code in such a way that on every addition, the table won't get repopulated.

All we have to do is just insert a new row at appropriate position, to keep the authors list in sorted order. Then select it and then bring it into the visible area of the table.

CRUD Three Revisited

Download & See it in Action
http://tm-certificates.com/CRUDThree.wmv

Lesson Learnt

We should have to guts to dump everything and start from scratch

The Data Layer and the TagSupport classes will be same, don't change them
The Servlets (Minor changes)

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorInterface author=new Author();
author.setName(name);
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.add(author);
pw.print("true,Author added.,"+author.getCode());
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class UpdateAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
int code=Integer.parseInt(request.getParameter("code"));
String name=request.getParameter("name");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
author=authorDAO.getByCode(code);
}catch(DAOException daoException)
{
pw.print("false,That author cannot be updated as someone working in parallel deleted the author while you were trying to update it.,delete");
return;
}
author.setName(name);
try
{
authorDAO.update(author);
pw.print("true,Author : "+name+" updated.");
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class DeleteAuthor extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
doPost(request,response);
}
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
BookDAO bookDAO=new BookDAO();
PrintWriter pw=null;
int code=Integer.parseInt(request.getParameter("code"));
try
{
```

```
pw=response.getWriter();
response.setContentType("text/html");
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(code);
if(bookWithAuthorExists)
{
pw.print("false,Cannot delete author as book exists against it.");
return;
}
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
try
{
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.getByCode(code);
}catch(DAOException daoException)
{
pw.print("true,Author deleted.");
return;
}
try
{
authorDAO.remove(code);
pw.print("true,Author deleted.");
return;
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetAuthors extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
boolean applyComma=false;
for(AuthorInterface author:authors)
{
if(applyComma) pw.print(",");
pw.print(author.getCode()+","+author.getName());
applyComma=true;
}
}catch(DAOException daoException)
{
pw.print("false,No authors");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class AddBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookInterface book=new Book();
book.setTitle(request.getParameter("title"));
book.setAuthorCode(Integer.parseInt(request.getParameter("authorCode")));
book.setCategory(request.getParameter("category"));
book.setPrice(Integer.parseInt(request.getParameter("price")));
BookDAO bookDAO=new BookDAO();
try
{
bookDAO.add(book);
pw.print("true,Book added.,"+book.getCode());
return;
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
return;
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class UpdateBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw;
pw=response.getWriter();
BookInterface book;
BookDAO bookDAO=new BookDAO();
try
{
book=bookDAO.getByCode(Integer.parseInt(request.getParameter("code")));
book.setTitle(request.getParameter("title"));
book.setAuthorCode(Integer.parseInt(request.getParameter("authorCode")));
book.setCategory(request.getParameter("category"));
book.setPrice(Integer.parseInt(request.getParameter("price")));
}catch(DAOException daoException)
{
pw.print("false,That book cannot be updated as someone working in parallel deleted the book while
you were trying to update it.,delete");
return;
}
try
{
bookDAO.update(book);
pw.print("true,Book updated.");
return;
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
return;
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class DeleteBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
try
{
int code=Integer.parseInt(request.getParameter("code"));
BookDAO bookDAO=new BookDAO();
bookDAO.remove(code);
pw.print("true,Book deleted successfully.");
return;
}catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetBooks extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookDAO bookDAO=new BookDAO();
try
{
LinkedList<BookInterface> books;
books=bookDAO.getAll();
boolean applyComma=false;
for(BookInterface book:books)
{
if(applyComma) pw.print(",");
pw.print(book.getCode()+","+book.getTitle()+","+book.getAuthorCode()+","+book.getCategory()
+","+book.getPrice());
applyComma=true;
}
}catch(DAOException daoException)
{
pw.print("false,No books");
}
}catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

Download CRUDThreeImages.zip from http://tm-certificates.com/CRUDThreeImages.zip
Unzip and copy the extracted images to images folder, overwrite the older images.

tm-js-1.css (tm-js-1 folder)

```
.progressModalMask
{
position:absolute;
```

```
left:0;
top:0;
width:100%;
height:100%;
background: url('../tm-js-1/progressMask.png');
visibility:hidden;
display: none;
opacity: 50%;
text-align:center;
}
.progressIndicator
{
position: absolute;
top:0;
bottom: 0;
left: 0;
right: 0;
margin: auto;
}
.forModal
{
visibility: hidden;
display: none;
}
.modalMask
{
position:absolute;
left:0;
top:0;
width:100%;
height:100%;
background: url('../tm-js-1/mask.png');
visibility:hidden;
display:none;
opacity: 50%;
}
.modal
{
border : 1px solid black;
height: 100px;
width: 100px;
position: absolute;
left:50%;
top: 10%;
z-index: 10000;
background: #FFFFFF;
border-radius: 0px;
```

```
box-shadow: 5px 5px 3px grey;
}
.modalHeader
{
background: #2E2E2E;
margin: 1px;
padding: 5px;
color: #FFFFFF;
font-size: 10pt;
height: 18px;
}
.modalClose
{
float: right;
cursor: pointer;
}
.modalTitle
{
float: left;
}
.modalFooter
{
background: #DFDFDF;
margin: 1px;
padding: 6px;
font-size: 8pt;
height: 10px;
text-align: center;
vertical-align: middle;
}
.modalContainer
{
margin: 1px;
padding:10px;
font-size:12pt;
overflow: auto;
}
```

tm-js-1.js (tm-js-1 folder)

```
function getXMLHttpObject()
{
var xmlHttpObject=null;
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlHttpObject=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlHttpObject=new ActiveXObject("Microsoft.XMLHTTP");
  }
return xmlHttpObject;
}
function jsonToQueryString(json)
{
return '?' +
Object.keys(json).map(function(key) {
return encodeURIComponent(key) + '=' +
encodeURIComponent(json[key]);
}).join('&');
}

function AjaxHandler()
{
var xmlHttpObject; // a private property
this.ajax=function(object)
{
if(xmlHttpObject==null)  xmlHttpObject=getXMLHttpObject();
xmlHttpObject.onreadystatechange=function(){
 if (xmlHttpObject.readyState==4 )
{
if(xmlHttpObject.status==200)
{
object.success(xmlHttpObject.responseText);
}
else
{
if(object.error) object.error();
}
}
};
if(!object.type) object.type=object.type.toUpperCase();
if(!object.type) object.type='GET';
if(!object.url) return;
if(object.data && object.type=='GET')
{
```

```
var qs=jsonToQueryString(object.data);
xmlHttpObject.open(object.type,object.url+qs,true);
xmlHttpObject.send();
}else if(!object.data && object.type=='GET')
{
xmlHttpObject.open(object.type,object.url,true);
xmlHttpObject.send();
} else if(!object.data && object.type=='POST')
{
xmlHttpObject.open(object.type,object.url, true);
xmlHttpObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttpObject.send();
}else if(object.data && object.type=='POST')
{
xmlHttpObject.open(object.type,object.url, true);
xmlHttpObject.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttpObject.send(jsonToQueryString(object.data).substring(1));
}
};
}
function showProgressModal()
{
var pmm=document.getElementsByClassName("progressModalMask")[0];
pmm.style.visibility="visible";
pmm.style.display="inline";
}
function hideProgressModal()
{
var pmm=document.getElementsByClassName("progressModalMask")[0];
pmm.style.visibility="hidden";
pmm.style.display="none";
}
function showModal(divisionToBeFixedInModal,width,height,title,footer)
{
document.body.scrollTop=document.documentElement.scrollTop=0;
var modalWrapperMask=document.getElementById('modalWrapperMask');
var modalWrapper=document.getElementById("modalWrapper");
modalWrapper.style.width=width;
modalWrapper.style.height=height;
var widthValue=parseInt(modalWrapper.style.width);
var widthUOM=modalWrapper.style.width.substring(modalWrapper.style.width.length-2);
modalWrapper.style.marginLeft=((widthValue/2)*(-1))+"px";
var heightValue=parseInt(modalWrapper.style.height);
var heightUOM=modalWrapper.style.height.substring(modalWrapper.style.height.length-2);
var modalWrapperContainer=document.getElementById("modalWrapperContainer");
modalWrapperContainer.style.height=(heightValue-18-18-38)+heightUOM;  // 18 header, 18 footer, 38
shadow
```

```
var modalWrapperCloseIcon=document.getElementById("modalWrapperCloseIcon");
if(title)
{
var modalWrapperTitle=document.getElementById("modalWrapperTitle");
modalWrapperTitle.innerHTML=title;
}
if(footer)
{
var modalWrapperFooter=document.getElementById("modalWrapperFooter");
modalWrapperFooter.innerHTML=footer;
}
var divToFix=document.getElementById(divisionToBeFixedInModal);
divToFix.remove();
var clone=divToFix.cloneNode(true);
clone.style.visibility="visible";
clone.style.display="inline";
modalWrapperContainer.append(clone);
modalWrapperContainer.fixedComponent=divToFix;
modalWrapperContainer.clonedComponent=clone;
modalWrapperMask.style.visibility="visible";
modalWrapperMask.style.display="inline";
}
function disposeModal()
{
var modalWrapperMask=document.getElementById('modalWrapperMask');
var modalWrapperContainer=document.getElementById("modalWrapperContainer");
var divToUnfix=modalWrapperContainer.clonedComponent;
divToUnfix.remove();
modalWrapperMask.style.visibility="hidden";
modalWrapperMask.style.display="none";
document.body.appendChild(modalWrapperContainer.fixedComponent);
}
function setupModals()
{
var progressModalWrapperMask=document.createElement("div");
progressModalWrapperMask.className="progressModalMask";
var img=document.createElement("img");
img.className='progressIndicator';
img.src="tm-js-1/loading.gif"
progressModalWrapperMask.appendChild(img);
document.getElementsByTagName('body')[0].appendChild(progressModalWrapperMask);
progressModalWrapperMask.style.zIndex = "2000"
// Following is what we created and appended it to body of the page
/*<div id='progressModalWrapperMask' class='progressModalMask'>
<img src='images/loading.gif' class='progressIndicator'></img>
</div>*/
var modalWrapperMask=document.createElement("div");
```

```
modalWrapperMask.className="modalMask";
modalWrapperMask.id="modalWrapperMask";
var modalWrapper=document.createElement("div");
modalWrapper.id="modalWrapper";
modalWrapper.className="modal";
modalWrapperMask.appendChild(modalWrapper);
var modalWrapperHeader=document.createElement("div");
modalWrapperHeader.className="modalHeader";
modalWrapperHeader.id="modalWrapperHeader";
modalWrapper.appendChild(modalWrapperHeader);
var modalWrapperTitle=document.createElement("div");
modalWrapperTitle.className="modalTitle";
modalWrapperTitle.id="modalWrapperTitle";
modalWrapperHeader.appendChild(modalWrapperTitle);
var modalWrapperCloseIcon=document.createElement("div");
modalWrapperCloseIcon.className="modalClose";
modalWrapperCloseIcon.id="modalWrapperCloseIcon";
modalWrapperCloseIcon.onclick=disposeModal;
var closeIconImage=document.createElement("img");
closeIconImage.src='tm-js-1/close_modal_icon.png';
modalWrapperCloseIcon.appendChild(closeIconImage);
modalWrapperHeader.appendChild(modalWrapperCloseIcon);
var modalWrapperContainer=document.createElement("div");
modalWrapperContainer.className="modalContainer";
modalWrapperContainer.id="modalWrapperContainer";
modalWrapper.appendChild(modalWrapperContainer);
var modalWrapperFooter=document.createElement("div");
modalWrapperFooter.className="modalFooter";
modalWrapperFooter.id="modalWrapperFooter";
modalWrapper.appendChild(modalWrapperFooter);
document.getElementsByTagName('body')[0].appendChild(modalWrapperMask);
modalWrapperMask.style.zIndex = "1000"
// Following is what we created and appended it to body of the page
/*
<div id='modalWrapperMask' class='modalMask'>
<div id='modalWrapper' class='modal'>
<div id='modalWrapperHeader' class='modalHeader'>
<div id='modalWrapperTitle' class='modalTitle'></div>
<div id='modalWrapperCloseIcon' class='modalClose' onClick='disposeModal()'>
<img src='tm-js-1/close_modal_icon.png'></img>
</div>
</div>
<div id='modalWrapperContainer' class='modalContainer'>
</div>
<div id='modalWrapperFooter' class='modalFooter'>
</div>
</div>
```

```
</div>*/
}

// following will setup the function to be called on window load event
window.onload=setupModals;

// following is the declaration of a global varable named as dollar ($)
// The name dollar ($) is just to give you an insight of how JQuery must have done things
// Note : My code is nowhere near to what JQuery creators have done, it is just an introduction

var $=new AjaxHandler();
```

styles.css (css folder)

```
body {
background: #F5F5F5;
}

.home
{
padding: 5px;
}

.header {
background: #DBDBDB;
width:100%;
}

.header table
{
width: 100%;
}

.footer {
background: #DBDBDB;
width:100%;
padding:10px;
text-align: center;
}

.error {
color: red;
}
.selectedRow
{
background-color: #3B3B3B;
color: white;
}
```

```
.actionIcon
{
cursor:pointer;
}
.listContainer
{
padding:1px;
}
.authorsListToolBar
{
width:600px;
border: 2px solid black;
background-color: #C1C1C1;
height: 28px;
padding-top:5px;
}
.authorSearchComponent
{
float: left;
padding-left:5px;
}
.authorSearchComponentInput
{
border: 2px solid black;
}
.authorSearchComponentInputError
{
border: 2px solid red;
}
.authorsListToolBarButtons
{
float: right;
cursor: pointer;
padding-right:5px;
}
.authorsListView
{
width: 600px;
border-collapse: collapse;
border-spacing: 0;
border: 2px solid black;
}

.authorsListView  thead
{
display: block;
}
```

```
.authorsListView thead tr th
{
height: 30px;
width: 50px;
line-height: 30px;
text-align: right;
border-right: 1px solid black;
}
.authorsListView thead tr th+th
{
width: 410px;
text-align: left;
}
.authorsListView thead tr th+th+th
{
width: 50px;
text-align: center;
}
.authorsListView thead tr th+th+th+th
{
width: 50px;
text-align: center;
border-right: none;
}
.authorsListView tbody
{
display: block;
width: 600px;
height: 300px;
overflow-y: auto;
overflow-x: hidden;
border-top: 2px solid black;
cursor: pointer;
}
.authorsListView tbody tr
{
height: 40px;
border-bottom: 1px solid black;
}
.authorsListView tbody tr td
{
width: 50px;
text-align: right;
border-right: 1px solid black;
}
.authorsListView tbody tr td+td
```

```
{
width: 410px;
text-align: left;
}
.authorsListView tbody tr td+td+td
{
width: 50px;
text-align: center;
}
.authorsListView tbody tr td+td+td+td
{
width: 50px;
text-align: center;
}

input[type=number]
{
text-align: right;
}

.clearFilter
{
visibility: hidden;
}
.booksListView
{
width: 850px;
border-collapse: collapse;
border-spacing: 0;
border: 2px solid black;
}
.booksListView  thead
{
display: block;
}
.booksListView thead tr th
{
height: 30px;
width: 50px;
line-height: 30px;
text-align: right;
border-right: 1px solid black;
}
.booksListView thead tr th+th
{
width: 250px;
text-align: left;
```

```
}
.booksListView thead tr th+th+th
{
width: 150px;
text-align: center;
}
.booksListView thead tr th+th+th+th
{
width: 200px;
text-align: center;
}
.booksListView thead tr th+th+th+th+th
{
width: 50px;
text-align: center;
}
.booksListView thead tr th+th+th+th+th+th
{
width: 50px;
text-align: center;
}
.booksListView thead tr th+th+th+th+th+th+th
{
width: 50px;
text-align: center;
border-right:none;
}
.booksListView tbody
{
display: block;
width: 850px;
height: 300px;
overflow-y: auto;
overflow-x: hidden;
border-top: 2px solid black;
cursor: pointer;
}
.booksListView tbody tr
{
height: 40px;
border-bottom: 1px solid black;
}
.booksListView tbody tr td
{
width: 50px;
text-align: right;
border-right: 1px solid black;
```

```
}
.booksListView tbody tr td+td
{
width: 250px;
text-align: left;
}
.booksListView tbody tr td+td+td
{
width: 150px;
text-align: center;
}
.booksListView tbody tr td+td+td+td
{
width: 200px;
text-align: center;
}
.booksListView tbody tr td+td+td+td+td
{
width: 50px;
text-align: center;
}
.booksListView tbody tr td+td+td+td+td+td
{
width: 50px;
text-align: center;
}
.booksListView tbody tr td+td+td+td+td+td+td
{
width: 50px;
text-align: center;
}
.booksListToolBar
{
width:850px;
border: 2px solid black;
background-color: #C1C1C1;
height: 28px;
padding-top:5px;
}
.bookSearchComponent
{
float: left;
padding-left:5px;
}
.bookSearchComponentInput
{
border: 2px solid black;
```

```
}
.bookSearchComponentInputError
{
border: 2px solid red;
}
.booksListToolBarButtons
{
float: right;
cursor: pointer;
padding-right:5px;
}
.bookFilterComponent
{
float:right;
padding-right:30px;
}
.bookFilterComponent select
{
border: 1px solid black;
}
```

common.js (js folder)

```
function hasClass(element,className)
{
if(element.classList)
return element.classList.contains(className);
else
return !!element.className.match(new RegExp('(\\s|^)'+className+'(\\s|$)'));
}
function addClass(element,className)
{
if(element.classList)
element.classList.add(className)
else if(!hasClass(element, className))element.className+=(" "+className);
}
function removeClass(element,className)
{
if(element.classList)
element.classList.remove(className);
else if(hasClass(element,className))
{
var reg=new RegExp('(\\s|^)'+className+'(\\s|$)');
element.className=el.className.replace(reg,' ');
}
}
function isElementInViewport(parent, element)
{
```

```
var elementRect = element.getBoundingClientRect();
var parentRect = parent.getBoundingClientRect();
return elementRect.top>=parentRect.top && elementRect.left>=parentRect.left &&
elementRect.bottom<=parentRect.bottom && elementRect.right<=parentRect.right;
}
function selectOption(element,value)
{
var i;
for(i=0;i<element.options.length;i++)
{
element.options[i].removeAttribute("selected");
}
for(i=0;i<element.options.length;i++)
{
if(element.options[i].value==value)
{
element.options[i].setAttribute("selected","");
break;
}
}
}
```

MasterPageTopSection.js (As it is – The same old one)

entities.js (js folder)

```
function Author(code,name)
{
this.code=code;
this.name=name;
}
function Book(code,title,author,category,price)
{
this.code=code;
this.title=title;
this.author=author;
this.category=category;
this.price=price;
}
```

Authors.js (js folder)

```
var selectedRow=null;
var searchText=null;
function selectRow(row)
{
var editIcon,deleteIcon;
if(selectedRow)
{
selectedRow.classList.remove('selectedRow');
```

```
editIcon=selectedRow.cells[2].getElementsByTagName("img")[0];
editIcon.src="images/edit_icon.png"
deleteIcon=selectedRow.cells[3].getElementsByTagName("img")[0];
deleteIcon.src="images/delete_icon.png"
}
selectedRow=row;
selectedRow.classList.add('selectedRow');
editIcon=selectedRow.cells[2].getElementsByTagName("img")[0];
editIcon.src="images/selected_row_edit_icon.png"
deleteIcon=selectedRow.cells[3].getElementsByTagName("img")[0];
deleteIcon.src="images/selected_row_delete_icon.png"
}
function clearAuthors()
{
var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
while(authorsListBody.rows.length>0)
{
authorsListBody.deleteRow(0);
}
var row=authorsListBody.insertRow(authorsListBody.length);
var cell1=row.insertCell(0);
cell1.innerHTML="&nbsp";
var cell2=row.insertCell(1);
cell2.innerHTML="&nbsp";
var cell3=row.insertCell(2);
cell3.innerHTML="&nbsp";
var cell4=row.insertCell(3);
cell4.innerHTML="&nbsp";
}
function loadAuthors()
{
showProgressModal();
clearAuthors();
$.ajax({
  type: 'GET',
  url: 'getAuthors',
  data: {
  now: new Date(),
  },
  success: function(data) {
  var spl=data.split(",")
  var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
  while(authorsListBody.rows.length>0) authorsListBody.deleteRow(0);
  if(spl[0]=='false')
  {
  hideProgressModal();
  return;
  }
  var i=0;
  var serialNumber=0;
  while(i<spl.length)
  {
  serialNumber++;
  var row=authorsListBody.insertRow(authorsListBody.rows.length);
  row.setAttribute("onclick","selectRow(this)");
  var cell1=row.insertCell(0);
  var cellText=document.createTextNode(serialNumber+".");
  cell1.appendChild(cellText);
  var cell2=row.insertCell(1);
  cellText=document.createTextNode(spl[i+1]);
  cell2.appendChild(cellText);
  var cell3=row.insertCell(2);
  var editIcon=document.createElement("img");
  editIcon.src="images/edit_icon.png"
  editIcon.setAttribute("onclick","editAuthor("+spl[i]+",'"+spl[i+1]+"')");
  cell3.appendChild(editIcon);
  var cell4=row.insertCell(3);
  var deleteIcon=document.createElement("img");
  deleteIcon.src="images/delete_icon.png"
  deleteIcon.setAttribute("onclick","confirmAuthorDeletion("+spl[i]+",'"+spl[i+1]+"')");
  cell4.appendChild(deleteIcon);
  i+=2;
  }
  hideProgressModal();
  },
  error: function() {
  hideProgressModal();
  alert('Unable to send request, server not ready......');
  }
});
}
function addAuthor()
{
var authorAddForm=document.getElementById("authorAddForm");
var name=authorAddForm.name.value.trim();
var nameErrorSection=document.getElementById("authorAddFormNameErrorSection");
nameErrorSection.innerHTML="&nbsp;";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
authorAddForm.name.focus();
return;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
authorAddForm.name.focus();
return;
}
e++;
}
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'addAuthor',
    data: {
    name : name
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
   hideProgressModal();
   disposeModal();
   var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
   var e;
   var authorOnRow;
   e=0;
   while(e<authorsListBody.rows.length)
   {
   authorOnRow=authorsListBody.rows[e].cells[1].innerHTML;
   if(authorOnRow.localeCompare(name)>0) break;
   e++;
   }
   var row=authorsListBody.insertRow(e);
   row.setAttribute("onclick","selectRow(this)");
   var cell1=row.insertCell(0);
   cell1.innerHTML=(e+1)+".";
   var cell2=row.insertCell(1);
   cell2.innerHTML=name;
   var cell3=row.insertCell(2);
  var editIcon=document.createElement("img");
  editIcon.src="images/edit_icon.png"
  editIcon.setAttribute("onclick","editAuthor("+spl[2]+",'"+name+"')");
  cell3.appendChild(editIcon);
  var cell4=row.insertCell(3);
  var deleteIcon=document.createElement("img");
  deleteIcon.src="images/delete_icon.png"
  deleteIcon.setAttribute("onclick","confirmAuthorDeletion("+spl[2]+",'"+name+"')");
  cell4.appendChild(deleteIcon);
  e++;
  while(e<authorsListBody.rows.length)
  {
  authorsListBody.rows[e].cells[0].innerHTML=(e+1)+".";
  e++;
  }
   selectRow(row);
   if(!isElementInViewport(authorsListBody,row))
   {
   row.scrollIntoView(true);
   document.body.scrollTop=document.documentElement.scrollTop=0;
   }
  clearAuthorSearchComponent();
  }
  else
  {
  hideProgressModal();
  var formErrorSection=document.getElementById("authorAddFormErrorSection");
  formErrorSection.innerHTML=spl[1];
  }
  },
  error: function() {
    hideProgressModal();
    alert('Unable to send request, server not ready......');
  }
});
}
function editAuthor(code,name)
{
var authorUpdateForm=document.getElementById("authorUpdateForm");
authorUpdateForm.code.value=code;
authorUpdateForm.name.value=name;
authorUpdateForm.oldName.value=name;
showModal("authorUpdateFormContainer","400px","200px","Update author","")
}
function updateAuthor()
{
var authorUpdateForm=document.getElementById("authorUpdateForm");
var code=authorUpdateForm.code.value;
var name=authorUpdateForm.name.value.trim();
var oldName=authorUpdateForm.oldName.value;
var nameErrorSection=document.getElementById("authorUpdateFormNameErrorSection");
nameErrorSection.innerHTML="&nbsp;";
if(name.length==0)
```

```
{
nameErrorSection.innerHTML="Required";
authorUpdateForm.name.focus();
return;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
authorUpdateForm.name.focus();
return;
}
e++;
}
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'updateAuthor',
    data: {
    code: code,
    name : name
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
   hideProgressModal();
   disposeModal();
   var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
   var e;
   var authorOnRow;
   e=0;
   while(e<authorsListBody.rows.length)
   {
   authorOnRow=authorsListBody.rows[e].cells[1].innerHTML;
   if(authorOnRow==oldName)
  {
  authorsListBody.deleteRow(e);
  break;
  }
   e++;
   }
  while(e<authorsListBody.rows.length)
  {
```

```
authorsListBody.rows[e].cells[0].innerHTML=(e+1)+".";
e++;
}
 e=0;
 while(e<authorsListBody.rows.length)
 {
 authorOnRow=authorsListBody.rows[e].cells[1].innerHTML;
 if(authorOnRow.localeCompare(name)>0) break;
 e++;
 }
 var row=authorsListBody.insertRow(e);
 row.setAttribute("onclick","selectRow(this)");
 var cell1=row.insertCell(0);
 cell1.innerHTML=(e+1)+".";
 var cell2=row.insertCell(1);
 cell2.innerHTML=name;
 var cell3=row.insertCell(2);
var editIcon=document.createElement("img");
editIcon.src="images/edit_icon.png"
editIcon.setAttribute("onclick","editAuthor("+code+",'"+name+"')");
cell3.appendChild(editIcon);
var cell4=row.insertCell(3);
var deleteIcon=document.createElement("img");
deleteIcon.src="images/delete_icon.png"
deleteIcon.setAttribute("onclick","confirmAuthorDeletion("+code+",'"+name+"')");
cell4.appendChild(deleteIcon);
e++;
while(e<authorsListBody.rows.length)
{
authorsListBody.rows[e].cells[0].innerHTML=(e+1)+".";
e++;
}
 selectRow(row);
 if(!isElementInViewport(authorsListBody,row))
 {
 row.scrollIntoView(true);
 document.body.scrollTop=document.documentElement.scrollTop=0;
 }
clearAuthorSearchComponent();
}
else
{
if(spl.length>2 && spl[2]=='delete')
{
 var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
 var e=0;
 var authorOnRow;
```

```
   while(e<authorsListBody.rows.length)
   {
   authorOnRow=authorsListBody.rows[e].cells[1].innerHTML;
   if(authorOnRow==oldName)
  {
  authorsListBody.deleteRow(e);
  break;
  }
   e++;
   }
  selectedRow=null;
  hideProgressModal();
  disposeModal();
  document.getElementById("messageDialogMessageSection").innerHTML=spl[1];
  showModal("messageDialogContainer","900px","250px","Update author","")
  return;
  }
  hideProgressModal();
  var formErrorSection=document.getElementById("authorUpdateFormErrorSection");
  formErrorSection.innerHTML=spl[1];
  }
  },
  error: function() {
    hideProgressModal();
    alert('Unable to send request, server not ready......');
  }
});
}
function confirmAuthorDeletion(code,name)
{
var authorDeleteForm=document.getElementById("authorDeleteForm");
authorDeleteForm.code.value=code;
authorDeleteForm.name.value=name;
document.getElementById("authorDeleteFormNameSection").innerHTML=name;
showModal("authorDeleteFormContainer","400px","200px","Delete author","")
}
function deleteAuthor()
{
var authorDeleteForm=document.getElementById("authorDeleteForm");
var code=authorDeleteForm.code.value;
var name=authorDeleteForm.name.value;
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'deleteAuthor',
    data: {
    code : code
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
   hideProgressModal();
   disposeModal();
   var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
   var e;
   var authorOnRow;
   e=0;
   while(e<authorsListBody.rows.length)
   {
   authorOnRow=authorsListBody.rows[e].cells[1].innerHTML;
   if(authorOnRow==name)
   {
    authorsListBody.deleteRow(e);
   break;
   }
   e++;
   }
  selectedRow=null;
  document.getElementById("messageDialogMessageSection").innerHTML="Author : "+name+"
deleted.";
  showModal("messageDialogContainer","450px","200px","Delete author","")
  clearAuthorSearchComponent();
  }
  else
  {
  hideProgressModal();
  var formErrorSection=document.getElementById("authorDeleteFormErrorSection");
  formErrorSection.innerHTML=spl[1];
  }
  },
  error: function() {
    hideProgressModal();
    alert('Unable to send request, server not ready......');
  }
});
}
function clearAuthorSearchComponent()
{
var authorToSearch=document.getElementById("authorToSearch");
authorToSearch.value="";
searchText="";
if(hasClass(authorToSearch,"authorSearchComponentInputError"))
{
```

```
removeClass(authorToSearch,"authorSearchComponentInputError");
addClass(authorToSearch,"authorSearchComponentInput");
}
}
function searchAuthor(textToSearch)
{
var authorToSearch=document.getElementById("authorToSearch");
if(textToSearch.trim().length==0)
{
if(hasClass(authorToSearch,"authorSearchComponentInputError"))
{
removeClass(authorToSearch,"authorSearchComponentInputError");
addClass(authorToSearch,"authorSearchComponentInput");
}
searchText="";
return;
}
textToSearch=textToSearch.toLowerCase();
if(searchText==textToSearch) return;
searchText=textToSearch;
var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
var authorOnRow;
var e;
for(e=0;e<authorsListBody.rows.length;e++)
{
authorOnRow=authorsListBody.rows[e].cells[1].innerHTML.toLowerCase();
if(authorOnRow.startsWith(searchText)) break;
}
if(e==authorsListBody.rows.length)
{
removeClass(authorToSearch,"authorSearchComponentInput");
addClass(authorToSearch,"authorSearchComponentInputError");
return;
}
if(hasClass(authorToSearch,"authorSearchComponentInputError"))
{
removeClass(authorToSearch,"authorSearchComponentInputError");
addClass(authorToSearch,"authorSearchComponentInput");
}
var row=authorsListBody.rows[e];
selectRow(row);
if(!isElementInViewport(authorsListBody,row))
{
row.scrollIntoView(true);
document.body.scrollTop=document.documentElement.scrollTop=0;
}
}
```

```
window.addEventListener('load',loadAuthors);
```

Books.js (js folder)

```
function Model()
{
this.authors=null;
this.books=null;
this.selectedRow=null;
this.searchText=null;
this.authorCode=0;
}
var dataModel=null;
function selectRow(row)
{
var editIcon,deleteIcon;
if(dataModel.selectedRow)
{
dataModel.selectedRow.classList.remove('selectedRow');
editIcon=dataModel.selectedRow.cells[5].getElementsByTagName("img")[0];
editIcon.src="images/edit_icon.png"
deleteIcon=dataModel.selectedRow.cells[6].getElementsByTagName("img")[0];
deleteIcon.src="images/delete_icon.png"
}
dataModel.selectedRow=row;
dataModel.selectedRow.classList.add('selectedRow');
editIcon=dataModel.selectedRow.cells[5].getElementsByTagName("img")[0];
editIcon.src="images/selected_row_edit_icon.png"
deleteIcon=dataModel.selectedRow.cells[6].getElementsByTagName("img")[0];
deleteIcon.src="images/selected_row_delete_icon.png"
}
function clearBooksView()
{
var booksListBody=document.getElementById("booksList").getElementsByTagName('tbody')[0];
while(booksListBody.rows.length>0)
{
booksListBody.deleteRow(0);
}
}
function clearFilter()
{
var filterByAuthorCode=document.getElementById("filterByAuthorCode");
filterByAuthorCode.selectedIndex=0;
authorSelectionChanged(0);
}
function authorSelectionChanged(code)
{
var bookAddForm=document.getElementById("bookAddForm");
var bookAddFormClearFilterSpan=document.getElementById("bookAddFormClearFilterSpan");
if(code==0)
{
populateBooksView(0);
selectOption(bookAddForm.authorCode,0);
bookAddForm.authorCode.disabled=false;
bookAddFormClearFilterSpan.style.visibility="hidden";
return;
}
populateBooksView(code);
selectOption(bookAddForm.authorCode,code);
bookAddForm.authorCode.disabled=true;
bookAddFormClearFilterSpan.style.visibility="visible";
}
function initializeDataModel(onInitialized)
{
$.ajax({
  type: 'GET',
  url: 'getAuthors',
  data: {
  now: new Date(),
  },
  success: function(data) {
  var authors=new Array();
  var spl=data.split(",")
  var i;
  if(spl[0]!='false')
  {
  i=0;
  while(i<spl.length)
  {
  authors.push(new Author(parseInt(spl[i]),spl[i+1]));
  i+=2;
  }
  }
  dataModel=new Model();
  dataModel.authors=authors;
  // code to getBooks starts here
  $.ajax({
  type: 'GET',
  url: 'getBooks',
  data: {
  now: new Date(),
  },
  success: function(data) {
  var books=new Array();
  var spl=data.split(",")
  var i;
  if(spl[0]!='false')
  {
  i=0;
  var f;
  var author;
  var book;
  while(i<spl.length)
  {
  author=null;
  for(f=0;f<dataModel.authors.length;f++)
  {
  if(dataModel.authors[f].code==parseInt(spl[i+2]))
  {
  author=dataModel.authors[f];
  break;
  }
  }
  book=new Book(parseInt(spl[i]),spl[i+1],author,spl[i+3],parseInt(spl[i+4]));
  books.push(book);
  i+=5;
  }
  }
  dataModel.books=books;
  // authors and books are ready with data in the dataModel
  onInitialized();
  },
  error: function() {
  hideProgressModal();
  alert('Unable to send request, server not ready......');
  }
  });
  // code to getBooks ends here
  },
  error: function() {
  hideProgressModal();
  alert('Unable to send request, server not ready......');
  }
});
}
function populateAuthorsList()
{
var filterByAuthorCode=document.getElementById("filterByAuthorCode");
var e;
var option;
var author;
for(e=0;e<dataModel.authors.length;e++)
```

```
{
author=dataModel.authors[e];
option=document.createElement("option");
option.text=author.name;
option.value=author.code;
filterByAuthorCode.options.add(option);
}

var bookAddFormAuthorCode=document.getElementById("bookAddForm").authorCode;
for(e=0;e<dataModel.authors.length;e++)
{
author=dataModel.authors[e];
option=document.createElement("option");
option.text=author.name;
option.value=author.code;
bookAddFormAuthorCode.options.add(option);
}

var bookUpdateFormAuthorCode=document.getElementById("bookUpdateForm").authorCode;
for(e=0;e<dataModel.authors.length;e++)
{
author=dataModel.authors[e];
option=document.createElement("option");
option.text=author.name;
option.value=author.code;
bookUpdateFormAuthorCode.options.add(option);
}

}
function populateBooksView(authorCode)
{
  dataModel.authorCode=authorCode;
  clearBooksView();
  var booksListBody=document.getElementById("booksList").getElementsByTagName('tbody')[0];
  while(booksListBody.rows.length>0) booksListBody.deleteRow(0);
  var i=0;
  var serialNumber=0;
  var book=null;
  while(i<dataModel.books.length)
  {
  book=dataModel.books[i];
  if(authorCode!=0 && book.author.code!=authorCode)
  {
  i++;
  continue;
  }
  serialNumber++;
  var row=booksListBody.insertRow(booksListBody.rows.length);
  row.setAttribute("onclick","selectRow(this)");
  var cell1=row.insertCell(0);
  var cellText=document.createTextNode(serialNumber+".");
  cell1.appendChild(cellText);
  var cell2=row.insertCell(1);
  cellText=document.createTextNode(book.title);
  cell2.appendChild(cellText);
  var cell3=row.insertCell(2);
  cellText=document.createTextNode(book.category);
  cell3.appendChild(cellText);
  var cell4=row.insertCell(3);
  cellText=document.createTextNode(book.author.name);
  cell4.appendChild(cellText);
  var cell5=row.insertCell(4);
  cellText=document.createTextNode(book.price);
  cell5.appendChild(cellText);
  var cell6=row.insertCell(5);
  var editIcon=document.createElement("img");
  editIcon.src="images/edit_icon.png"
  editIcon.setAttribute("onclick","editBook("+i+")");
  cell6.appendChild(editIcon);
  var cell7=row.insertCell(6);
  var deleteIcon=document.createElement("img");
  deleteIcon.src="images/delete_icon.png"
  deleteIcon.setAttribute("onclick","confirmBookDeletion("+i+")");
  cell7.appendChild(deleteIcon);
  i++;
  }
}
function initializeView()
{
  showProgressModal();
  initializeDataModel(function(){
  populateBooksView(0);
  populateAuthorsList();
  hideProgressModal();
});
}
function addBook()
{
var errorComponent=null;
var bookAddForm=document.getElementById("bookAddForm");
var title=bookAddForm.title.value.trim();
var titleErrorSection=document.getElementById("bookAddFormTitleErrorSection");
titleErrorSection.innerHTML="&nbsp;";
var valid=true;
```

```
if(title.length==0)
{
titleErrorSection.innerHTML="Required";
errorComponent=bookAddForm.title;
valid=false;
}
if(valid)
{
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .
0123456789";
var e=0;
while(e<title.length)
{
if(validCharacters.indexOf(title.charAt(e))==-1)
{
titleErrorSection.innerHTML="Invalid characters in title of author";
bookAddForm.title.focus();
valid=false;
errorComponent=bookAddForm.title;
break;
}
e++;
}
}
var category=bookAddForm.category.value;
var categoryErrorSection=document.getElementById("bookAddFormCategoryErrorSection");
categoryErrorSection.innerHTML="&nbsp;";
if(category=='none')
{
if(errorComponent==null) errorComponent=bookAddForm.category;
valid=false;
categoryErrorSection.innerHTML="Required";
}
var authorCode=bookAddForm.authorCode.value;
var authorCodeErrorSection=document.getElementById("bookAddFormAuthorCodeErrorSection");
authorCodeErrorSection.innerHTML="&nbsp;";
if(authorCode==0)
{
if(errorComponent==null) errorComponent=bookAddForm.authorCode;
valid=false;
authorCodeErrorSection.innerHTML="Required";
}
var price=bookAddForm.price.value;
var priceErrorSection=document.getElementById("bookAddFormPriceErrorSection");
priceErrorSection.innerHTML="&nbsp;";
if(!price)
{
```

```
if(errorComponent==null) errorComponent=bookAddForm.price;
valid=false;
priceErrorSection.innerHTML="Required";
}
if(price && parseInt(price)<=0)
{
if(errorComponent==null) errorComponent=bookAddForm.price;
valid=false;
priceErrorSection.innerHTML="Invalid price";
}
if(!valid)
{
errorComponent.focus();
return;
}
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'addBook',
    data: {
    title : title,
    category : category,
    authorCode : authorCode,
    price : price
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
   var code=parseInt(spl[2]);
   var author=null;
   var f;
   for(f=0;f<dataModel.authors.length;f++)
   {
   if(dataModel.authors[f].code==authorCode)
   {
   author=dataModel.authors[f];
   break;
   }
   }
   var book=new Book(code,title,author,category,price);
   for(f=0;f<dataModel.books.length;f++)
   {
   if(dataModel.books[f].title.toUpperCase().localeCompare(title.toUpperCase())>0)
   {
   break;
   }
   }
   dataModel.books.splice(f,0,book);
   hideProgressModal();
   disposeModal();
   populateBooksView(dataModel.authorCode);
   var booksListBody=document.getElementById("booksList").getElementsByTagName('tbody')[0];
   var row=booksListBody.rows[f];
   selectRow(row);
   if(!isElementInViewport(booksListBody,row))
   {
   row.scrollIntoView(true);
   document.body.scrollTop=document.documentElement.scrollTop=0;
   }
  clearBookSearchComponent();
  }
  else
  {
  hideProgressModal();
  var formErrorSection=document.getElementById("bookAddFormErrorSection");
  formErrorSection.innerHTML=spl[1];
  }
  },
  error: function() {
    hideProgressModal();
    alert('Unable to send request, server not ready......');
  }
});
}
function editBook(index)
{
var bookUpdateForm=document.getElementById("bookUpdateForm");
bookUpdateForm.index.value=index;
var book=dataModel.books[index];
bookUpdateForm.title.value=book.title;
selectOption(bookUpdateForm.authorCode,book.author.code);
selectOption(bookUpdateForm.category,book.category);
bookUpdateForm.price.value=book.price;
showModal("bookUpdateFormContainer","400px","340px","Update book","");
}
function updateBook()
{
var errorComponent=null;
var bookUpdateForm=document.getElementById("bookUpdateForm");
var index=bookUpdateForm.index.value;
var code=dataModel.books[index].code;
var title=bookUpdateForm.title.value.trim();
var titleErrorSection=document.getElementById("bookUpdateFormTitleErrorSection");
```

```
titleErrorSection.innerHTML="&nbsp;";
var valid=true;
if(title.length==0)
{
titleErrorSection.innerHTML="Required";
errorComponent=bookUpdateForm.title;
valid=false;
}
if(valid)
{
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .
0123456789";
var e=0;
while(e<title.length)
{
if(validCharacters.indexOf(title.charAt(e))==-1)
{
titleErrorSection.innerHTML="Invalid characters in title of author";
bookUpdateForm.title.focus();
valid=false;
errorComponent=bookUpdateForm.title;
break;
}
e++;
}
}
var category=bookUpdateForm.category.value;
var categoryErrorSection=document.getElementById("bookUpdateFormCategoryErrorSection");
categoryErrorSection.innerHTML="&nbsp;";
if(category=='none')
{
if(errorComponent==null) errorComponent=bookUpdateForm.category;
valid=false;
categoryErrorSection.innerHTML="Required";
}
var authorCode=bookUpdateForm.authorCode.value;
var authorCodeErrorSection=document.getElementById("bookUpdateFormAuthorCodeErrorSection");
authorCodeErrorSection.innerHTML="&nbsp;";
if(authorCode==0)
{
if(errorComponent==null) errorComponent=bookUpdateForm.authorCode;
valid=false;
authorCodeErrorSection.innerHTML="Required";
}
var price=bookUpdateForm.price.value;
var priceErrorSection=document.getElementById("bookUpdateFormPriceErrorSection");
priceErrorSection.innerHTML="&nbsp;";
```

```
if(!price)
{
if(errorComponent==null) errorComponent=bookUpdateForm.price;
valid=false;
priceErrorSection.innerHTML="Required";
}
if(price && parseInt(price)<=0)
{
if(errorComponent==null) errorComponent=bookUpdateForm.price;
valid=false;
priceErrorSection.innerHTML="Invalid price";
}
if(!valid)
{
errorComponent.focus();
return;
}
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'updateBook',
    data: {
    code : code,
    title : title,
    category : category,
    authorCode : authorCode,
    price : price
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
   dataModel.books.splice(index,1);
   var author=null;
   var f;
   for(f=0;f<dataModel.authors.length;f++)
   {
   if(dataModel.authors[f].code==authorCode)
   {
   author=dataModel.authors[f];
   break;
   }
   }
   var book=new Book(code,title,author,category,price);
   for(f=0;f<dataModel.books.length;f++)
   {
   if(dataModel.books[f].title.toUpperCase().localeCompare(title.toUpperCase())>0)
```

```
 {
 break;
 }
 }
 dataModel.books.splice(f,0,book);
 hideProgressModal();
 disposeModal();
 populateBooksView(dataModel.authorCode);
 var booksListBody=document.getElementById("booksList").getElementsByTagName('tbody')[0];
 var row=booksListBody.rows[f];
 selectRow(row);
 if(!isElementInViewport(booksListBody,row))
 {
 row.scrollIntoView(true);
 document.body.scrollTop=document.documentElement.scrollTop=0;
 }
clearBookSearchComponent();
}
else
{
if(spl.length>2 && spl[2]=='delete')
{
dataModel.books.splice(index,1);
populateBooksView(dataModel.authorCode);
dataModel.selectedRow=null;
hideProgressModal();
disposeModal();
document.getElementById("messageDialogMessageSection").innerHTML=spl[1];
showModal("messageDialogContainer","900px","250px","Update book","")
return;
}
hideProgressModal();
var formErrorSection=document.getElementById("bookUpdateFormErrorSection");
formErrorSection.innerHTML=spl[1];
}
},
error: function() {
  hideProgressModal();
  alert('Unable to send request, server not ready......');
}
});
}
function confirmBookDeletion(index)
{
var bookDeleteForm=document.getElementById("bookDeleteForm");
bookDeleteForm.index.value=index;
var bookDeleteFormTitleSection=document.getElementById("bookDeleteFormTitleSection");
```

```
bookDeleteFormTitleSection.innerHTML=dataModel.books[index].title;
showModal("bookDeleteFormContainer","400px","200px","Delete book","")
}
function deleteBook()
{
var bookDeleteForm=document.getElementById("bookDeleteForm");
var index=bookDeleteForm.index.value;
var book=dataModel.books[index];
showProgressModal();
$.ajax({
  type: 'POST',
  url: 'deleteBook',
    data: {
    code : book.code
  },
  success: function(data) {
  var spl=data.split(",");
  if(spl[0]==="true")
  {
  hideProgressModal();
  disposeModal();
  populateBooksView(dataModel.authorCode);
  dataModel.selectedRow=null;
  document.getElementById("messageDialogMessageSection").innerHTML="Book : "+book.title+"
deleted.";
  dataModel.books.splice(index,1);
  populateBooksView(dataModel.authorCode);
  showModal("messageDialogContainer","450px","200px","Delete book","")
  clearBookSearchComponent();
  }
  else
  {
  hideProgressModal();
  var formErrorSection=document.getElementById("bookDeleteFormErrorSection");
  formErrorSection.innerHTML=spl[1];
  }
  },
  error: function() {
    hideProgressModal();
    alert('Unable to send request, server not ready......');
  }
});
}
function clearBookSearchComponent()
{
var bookToSearch=document.getElementById("bookToSearch");
bookToSearch.value="";
```

```
dataModel.searchText="";
if(hasClass(bookToSearch,"bookSearchComponentInputError"))
{
removeClass(bookToSearch,"bookSearchComponentInputError");
addClass(bookToSearch,"bookSearchComponentInput");
}
}
function searchBook(textToSearch)
{
var bookToSearch=document.getElementById("bookToSearch");
if(textToSearch.trim().length==0)
{
if(hasClass(bookToSearch,"bookSearchComponentInputError"))
{
removeClass(bookToSearch,"bookSearchComponentInputError");
addClass(bookToSearch,"bookSearchComponentInput");
}
dataModel.searchText="";
return;
}
textToSearch=textToSearch.toLowerCase();
if(dataModel.searchText==textToSearch) return;
// done done
var book;
var e;
for(e=0;e<dataModel.books.length;e++)
{
book=dataModel.books[e];
if(book.title.toLowerCase().startsWith(textToSearch)) break;
}
if(e==dataModel.books.length)
{
removeClass(bookToSearch,"bookSearchComponentInput");
addClass(bookToSearch,"bookSearchComponentInputError");
return;
}
if(hasClass(bookToSearch,"bookSearchComponentInputError"))
{
removeClass(bookToSearch,"bookSearchComponentInputError");
addClass(bookToSearch,"bookSearchComponentInput");
}
var booksListBody=document.getElementById("booksList").getElementsByTagName('tbody')[0];
var row=booksListBody.rows[e];
selectRow(row);
if(!isElementInViewport(booksListBody,row))
{
row.scrollIntoView(true);
document.body.scrollTop=document.documentElement.scrollTop=0;
}
}
window.addEventListener('load',initializeView);
```

index.jsp (crudthree folder)

```
<jsp:include page='/MasterPageTopSection.jsp' />
<ul>
<li><a href='/${contextName}/Authors.jsp'>Authors</a></li>
<li><a href='/${contextName}/Books.jsp'>Books</a></li>
</ul>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

Authors.jsp (crudthree folder)

```
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/tm-js-1/tm-js-1.js'></script>
<script src='/${contextName}/js/common.js'></script>
<script src='/${contextName}/js/Authors.js'></script>
<h2>Authors</h2>
<table class='listContainer'>
<tr>
<td>
<div class='authorsListToolBar'>
<div class='authorSearchComponent'>
Search
<input id='authorToSearch' class='authorSearchComponentInput' type='text' maxlength='35' size='36'
onkeyup='searchAuthor(this.value)'>
</div>
<div class='authorsListToolBarButtons'>
<img src='/${contextName}/images/add_icon.png'
onclick='showModal("authorAddFormContainer","400px","200px","Add author","")'></img>
</div>
</div>
</td>
</tr>
<tr>
<td>
<table id='authorsList' class='authorsListView'>
<thead>
<tr>
<th>S.No.</th>
<th>Author</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
</tbody>
```

```
</table>
</td>
</tr>
</table>
<br>
<br>
<br>
<!-- Module Container For Modal Window Start -->
<div id='authorAddFormContainer' class='forModal'>
<span id='authorAddFormErrorSection' class='error'>&nbsp;</span>
<form id='authorAddForm'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='authorAddFormNameErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='button' onclick='addAuthor()'>Add</button>
</td>
</tr>
</table>
</form>
</div>

<div id='authorUpdateFormContainer' class='forModal'>
<span id='authorUpdateFormErrorSection' class='error'>&nbsp;</span>
<form id='authorUpdateForm'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='hidden' id='code' name='code'>
<input type='hidden' id='oldName' name='oldName'>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='authorUpdateFormNameErrorSection' class='error'>&nbsp;</span>
```

```
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='button' onclick='updateAuthor()'>Update</button>
</td>
</tr>
</table>
</form>
</div>
<div id='authorDeleteFormContainer' class='forModal'>
<center>
<span id='authorDeleteFormErrorSection' class='error'>&nbsp;</span>
<br>
<form id='authorDeleteForm'>
<input type='hidden' id='code' name='code'>
<input type='hidden' id='name' name='name'>
Delete Author : <span id='authorDeleteFormNameSection'></span> ?
<br>
<br>
<button type='button' onclick='deleteAuthor()'>Yes</button>
<button type='button' onclick='disposeModal()'>No</button>
</form>
</center>
</div>
<div id='messageDialogContainer' class='forModal'>
<center>
<br>
<h2><span id='messageDialogMessageSection'></span></h2>
<button type='button' onclick='disposeModal()'>Ok</button>
</center>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

Books.jsp (crudthree folder)

```
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/tm-js-1/tm-js-1.js'></script>
<script src='/${contextName}/js/common.js'></script>
<script src='/${contextName}/js/entities.js'></script>
<script src='/${contextName}/js/Books.js'></script>
<h2>Books</h2>
<table class='listContainer'>
<tr>
<td>
<div class='booksListToolBar'>
<div class='bookSearchComponent'>
Search
```

```
<input id='bookToSearch' class='bookSearchComponentInput' type='text' maxlength='35' size='36'
onkeyup='searchBook(this.value)'>
</div>
<div class='booksListToolBarButtons'>
<img src='/${contextName}/images/add_icon.png'
onclick='showModal("bookAddFormContainer","400px","340px","Add book","")'></img>
</div>
<div class='bookFilterComponent'>
Filter by author :
<select id='filterByAuthorCode' name='filterByAuthorCode'
onchange='authorSelectionChanged(this.value)'>
<option value='0'>&lt;All&gt;</select>
</select>
</div>
</div>
</td>
</tr>
<tr>
<td>
<table id='booksList' class='booksListView'>
<thead>
<tr>
<th>S.No.</th>
<th>Book</th>
<th>Category</th>
<th>Author</th>
<th>Price</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
</tbody>
</table>
</td>
</tr>
</table>
<br>
<br>
<br>
<!-- Module Container For Modal Window Start -->
<div id='bookAddFormContainer' class='forModal'>
<span id='bookAddFormErrorSection' class='error'>&nbsp;</span>
<form id='bookAddForm' >
<table>
<tr>
<td colspan='2'>
```

```
<table>
<tr>
<td>Title</td>
<td>
<input type='text' name='title' id='title' maxlength='35' size='31'
onkeyup="document.getElementById('bookAddFormTitleErrorSection').innerHTML='&nbsp;'">
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookAddFormTitleErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'
onchange="document.getElementById('bookAddFormCategoryErrorSection').innerHTML='&nbsp;'">
<option value='none'>&lt;Select&gt;</option>
<option value='Science fiction'>Science fiction</option>
<option value='Satire'>Satire</option>
<option value='Drama'>Drama</option>
<option value='Action and Adventure'>Action and Adventure</option>
<option value='Mystery'>Mystery</option>
<option value='Horror'>Horror</option>
</select>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookAddFormCategoryErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Author</td>
```

```
<td>
<select name='authorCode' id='authorCode'
onchange="document.getElementById('bookAddFormAuthorCodeErrorSection').innerHTML='&nbsp;'
">
<option value='0'>&lt;Select&gt;</option>
</select>
</td>
<td>
<span id='bookAddFormClearFilterSpan' class='clearFilter'><a href='javascript:clearFilter()'>Clear
filter</a></span>
&nbsp;
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookAddFormAuthorCodeErrorSection' class='error'>&nbsp;</span>
</td>
<td>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Price</td>
<td>
<input type='number' min='0' name='price' id='price'
onkeyup="document.getElementById('bookAddFormPriceErrorSection').innerHTML='&nbsp;'">
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookAddFormPriceErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='button' onclick='addBook()'>Add</button>
</td>
</tr>
</table>
```

```
</form>
</div>

<div id='bookUpdateFormContainer' class='forModal'>
<span id='bookUpdateFormErrorSection' class='error'>&nbsp;</span>
<form id='bookUpdateForm'>
<table>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Title</td>
<td>
<input type='hidden' id='index' name='index'>
<input type='text' name='title' id='title' maxlength='35' size='31'
onkeyup="document.getElementById('bookUpdateFormTitleErrorSection').innerHTML='&nbsp;'">
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookUpdateFormTitleErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'
onchange="document.getElementById('bookUpdateFormCategoryErrorSection').innerHTML='&nbsp;'
">
<option value='none'>&lt;Select&gt;</option>
<option value='Science fiction'}>Science fiction</option>
<option value='Satire'>Satire</option>
<option value='Drama'>Drama</option>
<option value='Action and Adventure'>Action and Adventure</option>
<option value='Mystery'>Mystery</option>
<option value='Horror'>Horror</option>
</select>
</td>
<td>
<span id='bookUpdateFormClearFilterSpan' class='clearFilter'><a href='javascript:clearFilter()'>Clear
filter</a></span>
```

```
&nbsp;
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookUpdateFormCategoryErrorSection' class='error'>&nbsp;</span>
</td>
<td></td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Author</td>
<td>
<select name='authorCode' id='authorCode'
onchange="document.getElementById('bookUpdateFormAuthorCodeErrorSection').innerHTML='&nb
sp;'">
<option value='0'>&lt;Select&gt;</option>
</select>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookUpdateFormAuthorCodeErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2'>
<table>
<tr>
<td>Price</td>
<td>
<input type='number' min='0' name='price' id='price'
onkeyup="document.getElementById('bookUpdateFormPriceErrorSection').innerHTML='&nbsp;'">
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='bookUpdateFormPriceErrorSection' class='error'>&nbsp;</span>
</td>
```

```
</tr>
</table>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='button' onclick='updateBook()'>Update</button>
</td>
</tr>
</table>
</form>
</div>
<div id='bookDeleteFormContainer' class='forModal'>
<center>
<span id='bookDeleteFormErrorSection' class='error'>&nbsp;</span>
<br>
<form id='bookDeleteForm'>
<input type='hidden' id='index' name='name'>
Delete Book : <span id='bookDeleteFormTitleSection'></span> ?
<br>
<br>
<button type='button' onclick='deleteBook()'>Yes</button>
<button type='button' onclick='disposeModal()'>No</button>
</form>
</center>
</div>
<div id='messageDialogContainer' class='forModal'>
<center>
<br>
<h2><span id='messageDialogMessageSection'></span></h2>
<button type='button' onclick='disposeModal()'>Ok</button>
</center>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

MasterPageTopSection.jsp (crudthree folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/tm-js-1/tm-js-1.css' >
```

```
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
</head>
<body>
<div class='header'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
<a href='/${contextName}/'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp (crudthree folder)

```
<div class='footer'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

(Servlet Mappings to be added to deployment descriptor)

```
<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>UpdateAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateAuthor</servlet-name>
<url-pattern>/updateAuthor</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>DeleteAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteAuthor</servlet-name>
```

```
<url-pattern>/deleteAuthor</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>GetAuthors</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthors</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthors</servlet-name>
<url-pattern>/getAuthors</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>GetBooks</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetBooks</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetBooks</servlet-name>
<url-pattern>/getBooks</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>AddBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddBook</servlet-name>
<url-pattern>/addBook</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>DeleteBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteBook</servlet-name>
<url-pattern>/deleteBook</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>UpdateBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateBook</servlet-name>
<url-pattern>/updateBook</url-pattern>
</servlet-mapping>
```

CRUD Three Complete

Parsing JSON

download JSONParsers.zip from http://tm-certificates.com/JSONParsers.zip

Extract the zip file and copy the jackson and the simplejson folder to c:\ or wherever you wish to copy

Library - Simple JSON

json1.java

```
import org.json.simple.*;
class json1
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
JSONObject jsonObject;
jsonObject=new JSONObject();
jsonObject.put("rollNumber",rollNumber);
jsonObject.put("name",name);
jsonObject.put("gender",gender);
System.out.println(jsonObject.toString());
}
}
```

To compile and run you need to include the jar files in the simplejson folder
While executing, pass 3 arguments (roll number, name and gender)

json2.java

```
import org.json.simple.*;
class json2
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
JSONObject jsonObject;
jsonObject=new JSONObject();
jsonObject.put("rollNumber",rollNumber);
jsonObject.put("name",name);
jsonObject.put("gender",gender);
JSONArray jsonArray=new JSONArray();
jsonArray.add(jsonObject);
rollNumber=Integer.parseInt(data[3]);
name=data[4];
gender=data[5];
jsonObject=new JSONObject();
jsonObject.put("rollNumber",rollNumber);
jsonObject.put("name",name);
jsonObject.put("gender",gender);
jsonArray.add(jsonObject);
System.out.println(jsonArray.toString());
}
}
```

To compile and run you need to include the jar files in the simplejson folder
While executing, pass 6 arguments (data of 2 students) (roll number1, name1 and gender1,roll number2,name2 and gender2)

json3.java

```
import org.json.simple.*;
import org.json.simple.parser.*;
class json3
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
JSONObject jsonObject;
jsonObject=new JSONObject();
jsonObject.put("rollNumber",rollNumber);
jsonObject.put("name",name);
jsonObject.put("gender",gender);
String jsonString=jsonObject.toString();
JSONParser jsonParser=new JSONParser();
JSONObject  jsonObject2;
try
{
jsonObject2=(JSONObject)jsonParser.parse(jsonString);
System.out.println(jsonObject2.get("rollNumber"));
System.out.println(jsonObject2.get("name"));
System.out.println(jsonObject2.get("gender"));
}catch(ParseException parseException)
{
System.out.println("Unable to parse");
}
}
}
```

To compile and run you need to include the jar files in the simplejson folder
While executing, pass 3 arguments (roll number, name and gender)

json4.java

```
import org.json.simple.*;
import org.json.simple.parser.*;
class json4
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
JSONObject jsonObject;
jsonObject=new JSONObject();
jsonObject.put("rollNumber",rollNumber);
jsonObject.put("name",name);
jsonObject.put("gender",gender);
JSONArray jsonArray=new JSONArray();
jsonArray.add(jsonObject);
rollNumber=Integer.parseInt(data[3]);
name=data[4];
gender=data[5];
jsonObject=new JSONObject();
jsonObject.put("rollNumber",rollNumber);
jsonObject.put("name",name);
jsonObject.put("gender",gender);
jsonArray.add(jsonObject);
String json=jsonArray.toString();
JSONParser jsonParser=new JSONParser();
try
{
JSONArray jsonArray2=(JSONArray)jsonParser.parse(json);
System.out.println("Size : "+jsonArray2.size());
for(int e=0;e<jsonArray2.size();e++)
{
jsonObject=(JSONObject)jsonArray2.get(e);
System.out.println("Roll number : "+jsonObject.get("rollNumber"));
System.out.println("Name : "+jsonObject.get("name"));
System.out.println("Gender : "+jsonObject.get("gender"));
}
}catch(ParseException pe)
{
System.out.println("Unable to parse JSON Array : "+pe.getMessage());
}
}
}
```

To compile and run you need to include the jar files in the simplejson folder
While executing, pass 6 arguments (data of 2 students) (roll number1, name1 and gender1,roll number2,name2 and gender2)

Library - Jackson

jackson1.java

```
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
class jackson1
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
ObjectMapper objectMapper=new ObjectMapper();
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
System.out.println(objectNode.toString());
}
}
```

To compile and run you need to include the jar files in the jackson folder
While executing, pass 3 arguments (roll number, name and gender)

Now I am not going to specify, what to include in classpath and how many arguments to pass, you need to understand on your own

jackson2.java

```
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
class jackson2
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
ObjectMapper objectMapper=new ObjectMapper();
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
ArrayNode arrayNode=objectMapper.createArrayNode();
arrayNode.add(objectNode);
rollNumber=Integer.parseInt(data[3]);
name=data[4];
gender=data[5];
objectNode=objectMapper.createObjectNode();
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
arrayNode.add(objectNode);
System.out.println(arrayNode.toString());
}
}
```

jackson3.java

```
import java.io.*;
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
class jackson3
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
ObjectMapper objectMapper=new ObjectMapper();
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
String jsonString=objectNode.toString();
ObjectNode objectNode2;
try
{
objectNode2=objectMapper.readValue(jsonString,ObjectNode.class);
System.out.println("Output after parsing");
System.out.println(objectNode2.get("rollNumber"));
System.out.println(objectNode2.get("name").textValue());
System.out.println(objectNode2.get("gender").textValue());
}catch(IOException ioException)
{
System.out.println(ioException);
}
}
}
```

jackson4.java

```
import java.io.*;
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
class jackson4
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
ObjectMapper objectMapper=new ObjectMapper();
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
ArrayNode arrayNode=objectMapper.createArrayNode();
arrayNode.add(objectNode);
rollNumber=Integer.parseInt(data[3]);
name=data[4];
gender=data[5];
objectNode=objectMapper.createObjectNode();
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
arrayNode.add(objectNode);
String json=arrayNode.toString();
try
{
ArrayNode arrayNode2=objectMapper.readValue(json,ArrayNode.class);
System.out.println("Size : "+arrayNode2.size());
for(int e=0;e<arrayNode2.size();e++)
{
objectNode=(ObjectNode)arrayNode.get(e);
System.out.println("Roll number : "+objectNode.get("rollNumber"));
System.out.println("Name : "+objectNode.get("name").textValue());
System.out.println("Gender : "+objectNode.get("gender").textValue());
}
}catch(IOException ioException)
{
System.out.println("Unable to parse JSON Array : "+ioException.getMessage());
}
}
}
```

jackson5.java

```
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
import java.util.*;
class City
{
private int code;
private String name;
public City()
{
this.code=0;
this.name="";
}
public City(int code,String name)
{
this.code=code;
this.name=name;
}
public int getCode()
{
return this.code;
}
public String getName()
{
return this.name;
}
}
class jackson5
{
public static void main(String data[])
{
int rollNumber=Integer.parseInt(data[0]);
String name=data[1];
String gender=data[2];
int cityCode=Integer.parseInt(data[3]);
String cityName=data[4];
City city=new City(cityCode,cityName);
ObjectMapper objectMapper=new ObjectMapper();
ObjectNode objectNode=objectMapper.createObjectNode();
ObjectNode cityNode=objectMapper.createObjectNode();
cityNode.put("code",city.getCode());
cityNode.put("name",city.getName());
objectNode.put("rollNumber",rollNumber);
objectNode.put("name",name);
objectNode.put("gender",gender);
objectNode.put("city",cityNode);
System.out.println("----------------------------------------------------");
System.out.println("Result of analysis");
System.out.println("----------------------------------------------------");
analyze(objectNode);
```

```
}
public static void analyze(JsonNode jsonNode)
{
Iterator<String> iterator=jsonNode.fieldNames();
JsonNode jsonNodeProperty;
String property;
boolean isValue,isContainerNode,isTextual;
while(iterator.hasNext())
{
property=iterator.next();
jsonNodeProperty=jsonNode.get(property);
isValue=jsonNodeProperty.isValueNode();
if(isValue)
{
if(jsonNodeProperty.isTextual())
{
System.out.println(property+" : "+jsonNodeProperty.textValue());
}
else
{
if(jsonNodeProperty.isShort())
{
System.out.println(property+" : "+Short.parseShort(jsonNode.toString()));
}
if(jsonNodeProperty.isInt())
{
System.out.println(property+" : "+Integer.parseInt(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isLong())
{
System.out.println(property+" : "+Long.parseLong(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isFloat())
{
System.out.println(property+" : "+Float.parseFloat(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isDouble())
{
System.out.println(property+" : "+Double.parseDouble(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isBoolean())
{
System.out.println(property+" : "+Boolean.parseBoolean(jsonNodeProperty.toString()));
}
}
}
else
{
System.out.println("Properties of : "+property);
System.out.println("--------------------------------");
analyze(jsonNodeProperty);
}
}
}
}
```

jackson6.java

```
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
import java.util.*;
class Country
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class State
{
private int code;
private String name;
private Country country;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
```

```
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
public void setCountry(Country country)
{
this.country=country;
}
public Country getCountry()
{
return this.country;
}
}

class City
{
private int code;
private String name;
private State state;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
public void setState(State state)
{
this.state=state;
}
public State getState()
{
return this.state;
```

```
}
}
class Category
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class Branch
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class Student
{
private int rollNumber;
private String name;
private City city;
private Branch branch;
private Category category;
private String hobbies[];
private int marks[];
public void setRollNumber(int rollNumber)
{
this.rollNumber=rollNumber;
}
public void setName(String name)
{
this.name=name;
}
public void setCity(City city)
{
this.city=city;
}
public void setBranch(Branch branch)
{
this.branch=branch;
}
public void setCategory(Category category)
{
this.category=category;
}
public void setHobbies(String hobbies[])
{
this.hobbies=hobbies;
}
public void setMarks(int marks[])
{
this.marks=marks;
}
public int getRollNumber()
{
return this.rollNumber;
}
public String getName()
{
return this.name;
}
public City getCity()
{
return this.city;
}
public Branch getBranch()
```

```
{
return this.branch;
}
public Category getCategory()
{
return this.category;
}
public String [] getHobbies()
{
return this.hobbies;
}
public int[] getMarks()
{
return this.marks;
}
}
class jackson6
{
public static void main(String data[])
{
Country country=new Country();
country.setCode(1);
country.setName("India");
State state=new State();
state.setCode(101);
state.setName("Madhya Pradesh");
state.setCountry(country);
City city=new City();
city.setCode(1001);
city.setName("Ujjain");
city.setState(state);
Branch branch=new Branch();
branch.setCode(5001);
branch.setName("Computer Science");
Category category=new Category();
category.setCode(6001);
category.setName("General");
Student student1=new Student();
student1.setRollNumber(10001);
student1.setName("Sameer Gupta");
student1.setHobbies(new String[]{"Reading fiction","Solving Crossword Puzzles","Robotics"});
student1.setMarks(new int[]{91,93,92,85,93});
student1.setCity(city);
student1.setBranch(branch);
student1.setCategory(category);
// assume that we have a object of type Student to which student1
// is pointing
```

```
ObjectMapper objectMapper=new ObjectMapper();
ObjectNode studentNode=objectMapper.createObjectNode();
ObjectNode branchNode=objectMapper.createObjectNode();
ObjectNode categoryNode=objectMapper.createObjectNode();
ObjectNode cityNode=objectMapper.createObjectNode();
ObjectNode stateNode=objectMapper.createObjectNode();
ObjectNode countryNode=objectMapper.createObjectNode();
cityNode.put("code",student1.getCity().getCode());
cityNode.put("name",student1.getCity().getName());
stateNode.put("code",student1.getCity().getState().getCode());
stateNode.put("name",student1.getCity().getState().getName());
countryNode.put("code",student1.getCity().getState().getCountry().getCode());
countryNode.put("name",student1.getCity().getState().getCountry().getName());
stateNode.put("country",countryNode);
cityNode.put("state",stateNode);
branchNode.put("code",student1.getBranch().getCode());
branchNode.put("name",student1.getBranch().getName());
categoryNode.put("code",student1.getCategory().getCode());
categoryNode.put("name",student1.getCategory().getName());
studentNode.put("rollNumber",student1.getRollNumber());
studentNode.put("name",student1.getName());
ArrayNode marksNode=objectMapper.createArrayNode();
for(int i=0;i<student1.getMarks().length;i++)
{
marksNode.add(student1.getMarks()[i]);
}
studentNode.putArray("marks").addAll(marksNode);
ArrayNode hobbiesNode=objectMapper.createArrayNode();
for(int i=0;i<student1.getHobbies().length;i++)
{
hobbiesNode.add(student1.getHobbies()[i]);
}
studentNode.putArray("hobbies").addAll(hobbiesNode);
studentNode.put("category",categoryNode);
studentNode.put("branch",branchNode);
studentNode.put("city",cityNode);
System.out.println("----------------------------------------------------");
System.out.println("Result of analysis");
System.out.println("----------------------------------------------------");
analyze(studentNode);
}
public static void analyze(JsonNode jsonNode)
{
Iterator<String> iterator=jsonNode.fieldNames();
JsonNode jsonNodeProperty;
String property;
boolean isValue,isContainerNode,isTextual,isArray,isObject;
```

```
while(iterator.hasNext())
{
property=iterator.next();
jsonNodeProperty=jsonNode.get(property);
isValue=jsonNodeProperty.isValueNode();
if(isValue)
{
if(jsonNodeProperty.isTextual())
{
System.out.println(property+" : "+jsonNodeProperty.textValue());
}
else
{
if(jsonNodeProperty.isShort())
{
System.out.println(property+" : "+Short.parseShort(jsonNode.toString()));
}
if(jsonNodeProperty.isInt())
{
System.out.println(property+" : "+Integer.parseInt(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isLong())
{
System.out.println(property+" : "+Long.parseLong(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isFloat())
{
System.out.println(property+" : "+Float.parseFloat(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isDouble())
{
System.out.println(property+" : "+Double.parseDouble(jsonNodeProperty.toString()));
}
if(jsonNodeProperty.isBoolean())
{
System.out.println(property+" : "+Boolean.parseBoolean(jsonNodeProperty.toString()));
}
}
}
else
{
isArray=jsonNodeProperty.isArray();
isContainerNode=jsonNodeProperty.isArray();
isObject=jsonNodeProperty.isObject();
if(isArray)
{
System.out.println("--------------------------------");
System.out.println("Values against : "+property);
System.out.println("---------------------------------");

Iterator<JsonNode> arrayIterator=jsonNodeProperty.elements();
JsonNode elementNode;
int i=0;
while(arrayIterator.hasNext())
{
elementNode=arrayIterator.next();
// ideally here also the elementNode needs to be analyzed for type using isType methods
// you do that yourself.
System.out.printf("Element at index %d is %s\n",i,elementNode.toString());
i++;
}
}else if(isObject)
{
System.out.println("---------------------------------");
System.out.println("Properties of : "+property);
System.out.println("---------------------------------");
analyze(jsonNodeProperty);
}
}
}
}
}
```

Bugs in CRUD Three

1) In Authors module if a row is deleted, then the serial numbers in the rows following the row which got deleted are not adjusted.
2) In Authors module, while updating a author, if the author is found to be deleted by someone else working in parallel, then the author gets deleted from the UI but the serial number in the rows following the row which got deleted are not adjusted.
3) In books module, when a filter is applied, the search functionality is not working as desired.
4) In books module, If a row is selected and then filter is set or cleared, if the new result contains the row that was selected before changing the filter, it does not appear as selected after changing filter.
5) In books module, The search function contains a line // done done, determine what is missing and fill the missing code.
6) In books module, in populateBooksView function a line which is not required exists. Determine that and remove it.

Fixing the bugs in CRUD Three

Solution to problem 1 & 2 : Introduce this code at the correct place in js\Authors.js

```
while(e<authorsListBody.rows.length)
{
authorsListBody.rows[e].cells[0].innerHTML=(e+1)+".";
e++;
}
```

Hint : somewhere in the success part of the ajax call (in deleteAuthor() and updateAuthor())

functions)
Solution to problem 3 & 4 :

In js\Books.js do the following

Add the following function just above the definition of the selectRow function

```
function getSelectedBookIndex()
{
if(dataModel.selectedRow)
{
var lookFor=dataModel.selectedRow.cells[1].innerHTML;
var book;
for(var i=0;i<dataModel.books.length;i++)
{
book=dataModel.books[i];
if(book.title.localeCompare(lookFor)==0) return i;
}
}
return -1;
}
```

Modify the definition of **authorSelectionChanged** as follows (just replace the old one with the following one). I have highlighted the changes

```
function authorSelectionChanged(code)
{
var selectedBookIndex=getSelectedBookIndex();
var bookAddForm=document.getElementById("bookAddForm");
var bookAddFormClearFilterSpan=document.getElementById("bookAddFormClearFilterSpan");
if(code==0)
{
populateBooksView(0);
selectOption(bookAddForm.authorCode,0);
bookAddForm.authorCode.disabled=false;
bookAddFormClearFilterSpan.style.visibility="hidden";
}
else
{
populateBooksView(code);
selectOption(bookAddForm.authorCode,code);
bookAddForm.authorCode.disabled=true;
bookAddFormClearFilterSpan.style.visibility="visible";
}
if(selectedBookIndex==-1)
{
dataModel.selectedRow=null;
}
else
{
if(dataModel.authorCode!=0 && dataModel.authorCode!
=dataModel.books[selectedBookIndex].author.code)
{
dataModel.selectedRow=null;
return;
}
var booksListBody=document.getElementById("booksList").getElementsByTagName('tbody')
[0];
var book;
var e,i;
i=0;
for(e=0;e<dataModel.books.length;e++)
{
book=dataModel.books[e];
if(dataModel.books[e].code==dataModel.books[selectedBookIndex].code) break;
if(dataModel.authorCode==book.author.code || dataModel.authorCode==0)
{
i++;
}
}
if(e==dataModel.books.length)
{
dataModel.selectedRow=null;
}
else
{
var row=booksListBody.rows[i];
selectRow(row);
if(!isElementInViewport(booksListBody,row))
{
row.scrollIntoView(true);
document.body.scrollTop=document.documentElement.scrollTop=0;
}
}
}
}
```

Solution to problem 5 : replace // done done with the following in searchAuthor function
searchText=textToSearch;

Solution to problem 6 : The following line
while(booksListBody.rows.length>0) booksListBody.deleteRow(0);
is not required as we have already placed a call to **clearBooksView()** function.

Thats it – Bugs fixed

CRUD Four

JQuery (Core – UI and Validator)

Tabulator – Interactive JQuery Plugin

Observer pattern

The learning will be incremental fashion, and this time the student has to contribute.

Download & See it in Action
http://tm-certificates.com/CRUDFour.wmv

Create webapplication folder structure with context name as crudfour

Download crudfour_resources.zip from http://tm-certificates.com/crudfour_resources.zip
Unzip it and copy the folders to crudfour folder

in crudfour folder create css and js folder
in WEB-INF folder create classes and lib folder
In lib folder copy the mysql.jar and 3 jackson-annotations,jackson-core and jackson-databind (3) jar files
in classs folder create the following structures

com\thinking\machines\library
in library folder, create beans, dl, servlets and tags folder

In the dl folder copy all files from

crudthree\WEB-INF\classes\com\thinking\machines\library\dl

in the tags folder copy all the files from crudthree\WEB-INF\classes\com\thinking\machines\library\tags folder

Make a change in EnvironmentVariablesTagHandler.java (change value against context name to crudfour and compile it

in beans folder create the following files

```
package com.thinking.machines.library.beans;
public class AuthorBean implements java.io.Serializable,Comparable<AuthorBean>
{
private int code;
public String name;
public AuthorBean()
{
this.code=0;
this.name="";
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
return this.code;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
public boolean equals(Object object)
{
if(!(object instanceof AuthorBean)) return false;
AuthorBean otherAuthorBean=(AuthorBean)object;
return this.code==otherAuthorBean.code;
}
public int compareTo(AuthorBean otherAuthorBean)
{
return this.code-otherAuthorBean.getCode();
}
public int hashCode()
{
return this.code;
}
}
```

```
package com.thinking.machines.library.beans;
public class BookBean implements java.io.Serializable,Comparable<BookBean>
{
private int code;
public String title;
private int authorCode;
private String category;
private int price;
public BookBean()
{
this.code=0;
this.title="";
this.authorCode=0;
this.category="";
this.price=0;
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
```

```
return this.code;
}
public void setTitle(String title)
{
this.title=title;
}
public String getTitle()
{
return this.title;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public void setCategory(String category)
{
this.category=category;
}
public String getCategory()
{
return this.category;
}
public void setPrice(int price)
{
this.price=price;
}
public int getPrice()
{
return this.price;
}
public boolean equals(Object object)
{
if(!(object instanceof BookBean)) return false;
BookBean otherBookBean=(BookBean)object;
return this.code==otherBookBean.code;
}
public int compareTo(BookBean otherBookBean)
{
return this.code-otherBookBean.getCode();
}
public int hashCode()
{
return this.code;
```

```
}
}
```

In servlets folder create the following servlets

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import com.thinking.machines.library.beans.*;
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetAuthors extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
PrintWriter pw=null;
ObjectMapper objectMapper=null;
try
{
pw=response.getWriter();
response.setContentType("application/json");
objectMapper=new ObjectMapper();
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
List list=new LinkedList<AuthorBean>();
AuthorBean authorBean;
for(AuthorInterface author:authors)
{
authorBean=new AuthorBean();
authorBean.setCode(author.getCode());
authorBean.setName(author.getName());
list.add(authorBean);
}
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",true);
String jsonString=objectMapper.writeValueAsString(list);
JsonNode jsonNode=objectMapper.readTree(jsonString);
objectNode.putPOJO("authors",jsonNode);
jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}catch(DAOException daoException)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message",daoException.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}catch(Exception exception)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",false);
objectNode.put("error",true);
objectNode.put("message",exception.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}
}
```

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
PrintWriter pw=null;
ObjectMapper objectMapper=null;
try
{
pw=response.getWriter();
response.setContentType("application/json");
objectMapper=new ObjectMapper();
String name=request.getParameter("name");
AuthorInterface author=new Author();
author.setName(name);
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.add(author);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",true);
objectNode.put("code", author.getCode());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}catch(DAOException daoException)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message",daoException.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}catch(Exception exception)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",false);
objectNode.put("error",true);
objectNode.put("message",exception.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}
}
```

Following are the contents of the web.xml file to be kept in the WEB-INF folder

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

    <description>
      CRUD Operations - Style One
    </description>
    <display-name>CRUD Operations - Style One</display-name>
<servlet>
<servlet-name>GetAuthors</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthors</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthors</servlet-name>
<url-pattern>/getAuthors</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>

</web-app>
```

In crudfour\css folder create the following styles.css file

styles.css

```
body {
background: #F5F5F5;
}

.home
{
padding: 5px;
}

.headerComponent {
background: #DBDBDB;
width:100%;
}

.header table
{
width: 100%;
}

.footerComponent {
background: #DBDBDB;
width:100%;
padding:10px;
text-align: center;
margin-top: 20px;
}

.error {
color: red;
}


.viewComponent
{
display1:none;
}

.errorComponent
{
}

.titleComponent
{
}
```

```
.toolBar
{
width:100%;
border: none;
background-color: #C1C1C1;
height: 28px;
padding-top:5px;
}

.toolBarButtons
{
float: right;
cursor: pointer;
padding-right:5px;
}

.toolBarButtons button
{
cursor: pointer;
}

.searchComponent
{
float: left;
padding-left:5px;
}

.searchComponentInput
{
border: 2px solid black;
}

.searchComponentInputError
{
border: 2px solid red;
}

.listComponent
{

}

.addComponent {
display1: none;
}

.editComponent {
```

```
display1: none;
}

.deleteComponent {
display1: none;
}

.emptyListComponent
{
width: 100%;
font-size: 24px;
text-align: center;
}
```

In js folder create MasterPageTopSection.js file (copy it from crudthree\js folder) No changes required.

Following is the MasterPageTopSection.jsp (location crudfour folder)

MasterPageTopSection.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/jquery-ui-1.12.1/jquery-ui.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/tabulator/css/tabulator.min.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/font-awesome-4.7.0/css/font-awesome.min.css' >
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
<script src='/${contextName}/jquery-3.2.1/jquery-3.2.1.min.js'></script>
<script src='/${contextName}/jquery-ui-1.12.1/jquery-ui.min.js'></script>
<script src='/${contextName}/jquery-validation/jquery.validate.min.js'></script>
<script src='/${contextName}/jquery-validation/additional-methods.min.js'></script>
<script src='/${contextName}/tabulator/js/tabulator.min.js'></script>
</head>
<body>
<div id='headerComponent' class='headerComponent'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
```

```
<a href='/${contextName}/'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp (location : crudfour folder)

```
<div id='footerComponent' class='footerComponent'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

Authors.jsp (location : crudfour folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/watch/watch.js'></script>
<style>
.ui-widget-overlay {
opacity: .50;
filter: Alpha(opacity=50);
background: rgb(50,50,50);
}
</style>
<script>
jQuery.fn.insertAt=function(index, element)
{
var lastIndex = this.children().length;
if(index<0)
{
index=Math.max(0, lastIndex+1+index);
}
this.append(element);
if(index<lastIndex)
{
this.children().eq(index).before(this.children().last());
}
return this;
}
var Modes={
NONE: 0,
VIEW_MODE: 1,
ADD_MODE: 2,
EDIT_MODE: 3,
DELETE_MODE: 4,
ERROR_MODE: 7
}
```

```
function Author(code,name)
{
this.code=code;
this.name=name;
}
function AuthorService()
{
this.getAuthors=function(successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: "GET",
url: "getAuthors",
data: {
now: new Date()
},
success: function(data){
if(data.success)
{
successCallBack(data.authors);
}
else
{
exceptionCallBack(data.message);
}
},
error: function(){
errorCallBack();
}
});
}; // getAuthor ends
this.addAuthor=function(author,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'addAuthor',
data: author,
success: function(data){
if(data.success)
{
successCallBack(data.code);
}
if(data.exception)
{
exceptionCallBack(data.message);
}
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // addAuthor ends
}
// service ends
function Component(element)
{
this.inDOM=null;
if(element)
{
this.parent=element.parent();
this.element=element;
this.index=this.element.index();
this.inDOM=true;
}
this.isInDOM=function() {
return this.inDOM;
};
this.find=function(lookFor) {
if(this.element && this.inDOM) return this.element.find(lookFor);
return null;
};
this.add=function(){
if(this.element && !this.inDOM)
{
this.parent.insertAt(this.index,element);
this.inDOM=true;
}
};
this.remove=function(){
if(this.element && this.inDOM)
{
this.element.remove();
this.inDOM=false;
}
};
this.hide=function(){
if(this.element && this.inDOM)  this.element.hide();
};
this.show=function(){
if(this.element && this.inDOM) this.element.show();
};
}
function Model()
{
this.searchError=null;
this.collection=null;
this.selectedAuthor=null;
this.pageSize=10;
this.mode=Modes.NONE;
this.pageNumber=0;
}
function AuthorController()
{
// setting up service
var authorService=new AuthorService();
// setting up model
var model=new Model();
// setting up view
var authorView=new AuthorView(this,model);
this.selectAuthor=function(code) {
for(var i=0;i<model.collection.length;i++)
{
if(model.collection[i].code==code) break;
}
if(i<model.collection.length)
{
model.selectedAuthor=model.collection[i];
}
else
{
model.selectedAuthor=null;
}
}
this.searchAuthor=function(authorNameLeftPart) {
if($.trim(authorNameLeftPart).length==0)
{
model.searchError=null;
return;
}
authorNameLeftPart=authorNameLeftPart.toLowerCase();
var y=model.collection.findIndex(function(author){
return author.name.toLowerCase().startsWith(authorNameLeftPart);
});
if(y==-1)
{
model.searchError=true;
model.selectedAuthor=null;
}
else
{
model.searchError=false;
model.selectedAuthor=model.collection[y];
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
```

```
}
}
this.addAuthor=function() {
};
this.setPageNumber=function(pageNumber)
{
if(model.pageNumber!=pageNumber)
{
model.pageNumber=pageNumber;
}
}
this.getAuthors=function() {
authorService.getAuthors(function(authors){
model.collection=authors;
model.mode=Modes.VIEW_MODE;
},function(exception){
model.collection=[];
model.mode=Modes.VIEW_MODE;
},function(){
alert('problem with server');
});
};
// setupObserver
watch(model, function(property,action,newValue,oldValue){
authorView.updateView(model,property);
});
// loading authors
this.getAuthors();
}
function AuthorView(authorController,model)
{
// following variable is required tabulators getPage is not working
var pageNumber=-1;
// initializing authorsList
var editIcon=function(cell,formatterParams){
var $i=$('<i>');
$i.attr("class","fa fa-edit");
$i.click(function(){
editClicked(cell.getRow().getData());
});
return $i;
};
var deleteIcon=function(cell,formatterParams){
var $i=$('<i>');
$i.attr("class","fa fa-trash");
$i.click(function(){
deleteClicked(cell.getRow().getData());
});
return $i;
};
function intializeTabulator()
{
$("#authorsList").tabulator({
pagination: 'local',
paginationSize: model.pageSize,
columns: [
{title: 'S.No.',formatter:"rownum",width: 60,headerSort: false,align: 'right' },
{title: 'Author',field: 'name',width: 400, headerSort: false},
{title: 'Edit',formatter:editIcon,width:60,headerSort: false,align: 'center'},
{title: 'Delete',formatter:deleteIcon,width:60,headerSort: false,align: 'center'}],
index: 'code',
selectable: 1,
resizableColumns:false,
rowClick:function (e,row){
authorController.selectAuthor(row.getIndex());
},
pageLoaded:function(pageNumber){
authorController.setPageNumber(pageNumber);
}
});
}
// Initialization of authorsList complete
// setup view DS
var components={
};
components.viewComponent=new Component($("#viewComponent"));
components.titleComponent=new
Component(components.viewComponent.find("div[id='titleComponent']"));
components.listComponent=new
Component(components.viewComponent.find("div[id='listComponent']"));
components.addComponent=new
Component(components.viewComponent.find("div[id='addComponent']"));
//components.editComponent=new
Component(components.viewComponent.find("div[id='editComponent']"));
//components.deleteComponent=new
Component(components.viewComponent.find("div[id='deleteComponent']"));
components.emptyListComponent=new
Component(components.viewComponent.find("div[id='emptyListComponent']"));
components.errorComponent=new
Component(components.viewComponent.find("div[id='errorComponent']"));
components.listComponent.remove();
components.emptyListComponent.remove();
components.errorComponent.remove();
components.addComponent.hide();
components.viewComponent.show();
// the function responsible for updating the view
this.updateView=function(authorModel,property)
{
if(property==='collection')
{
if(components.listComponent.isInDOM()==false)
{
components.listComponent.add();
}
intializeTabulator();
$("#authorsList").tabulator('setData',model.collection);
return;
} // collection ends
if(property=='searchError')
{
if(model.searchError)
{
$
("#authorToSearch").removeClass("searchComponentInput").addClass("searchComponentInputError")
;
}
else
{
$
("#authorToSearch").removeClass("searchComponentInputError").addClass("searchComponentInput")
;
}
}
if(property=='selectedAuthor')
{
if(model.selectedAuthor)
{
$("#authorsList").tabulator('selectRow',model.selectedAuthor.code);
}
else
{
$("#authorsList").tabulator('deselectRow');
}
}// selectedAuthor ends
if(property=='pageSize')
{
// we are not enabling the user to change page size, hence
// let us just ignore this notification right now
return;
}// pageSize ends
if(property=='pageNumber')
{
if(pageNumber!=model.pageNumber)
{
$("#authorsList").tabulator('setPage',model.pageNumber)
pageNumber=model.pageNumber;
}
return;
}// pageNumber ends
if(property=='mode')
{
if(authorModel.mode==Modes.VIEW_MODE)
{
if(model.collection.length>0)
{
components.addComponent.hide();
components.emptyListComponent.remove();
components.listComponent.add();
components.listComponent.show();
}
else
{
components.addComponent.hide();
components.listComponent.remove();
components.emptyListComponent.add();
components.emptyListComponent.show();
}
return;
} // VIEW_MODE ends
if(authorModel.mode==Modes.ADD_MODE)
{
} // ADD_MODE ends
if(authorModel.mode==Modes.ERROR_MODE)
{
} //  ERROR_MODE ends
} // mode ends
};
}
var authorController=null;
function initialize()
{
authorController=new AuthorController();
}
$(document).ready(initialize);
</script>
<div id='viewComponent' class='viewComponent'>
<div id='titleComponent' class='titleComponent'>
<h1>Authors</h1>
```

```
</div>
<div id='listComponent' class='listComponent'>
<div id='toolBar' class='toolBar'>
<div class='searchComponent'>
Search
<input id='authorToSearch' class='searchComponentInput' type='text' maxlength='35' size='36' onkeyup='authorController.searchAuthor(this.value)'>
</div>
<div class='toolBarButtons'>
<button type='button' onclick='model.mode=ADD_MODE'><i class='fa fa-plus'></i></button>
</div>
</div>
<div id='authorsList'>
</div>
</div>

<div id='addComponent' class='addComponent'>
<form id='authorAddForm'>
<div id='formErrorSection' class='error'></div>
<table>
<tr>
<td>Name</td>
<td><input type='text' name='name' id='name'></td>
</tr>
<tr>
<td colspan='2' align='center'><button type='button' onclick='addAuthor()'><i class='fa fa-floppy-o'></i></button></td>
</tr>
</table>
</form>
</div>

<div id='emptyListComponent' class='emptyListComponent'>
No authors<br>
<button type='button' onclick='model.mode=ADD_MODE'><i class='fa fa-plus'></i></button>
</div>

<div id='errorComponent' class='errorComponent'>
<center>
<img src='images/error.jpg'>
</center>
</div>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

For now, we have not created index.jsp, hence to test our code you will have to specify URL to Authors.jsp as **http://localhost:8080/crudfour/Authors.jsp**

Minor changes to css and complete code for Author Module
(This time the javascript code is in different files)

Add the following servlets

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
public class UpdateAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
PrintWriter pw=null;
ObjectMapper objectMapper=null;
try
{
pw=response.getWriter();
response.setContentType("application/json");
objectMapper=new ObjectMapper();
int code=Integer.parseInt(request.getParameter("code"));
String name=request.getParameter("name");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
author=authorDAO.getByCode(code);
}catch(DAOException daoException)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message","That author cannot be updated as someone working in parallel deleted the
author while you were trying to update it.");
objectNode.put("delete",true);
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}
author.setName(name);
try
{
authorDAO.update(author);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",true);
objectNode.put("message","Author : "+name+" updated.");
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}catch(DAOException daoException)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message",daoException.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}catch(Exception exception)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",false);
objectNode.put("error",true);
objectNode.put("message",exception.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}
}
```

```
package com.thinking.machines.library.servlets;
import java.io.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
import com.fasterxml.jackson.core.*;
import com.fasterxml.jackson.databind.*;
import com.fasterxml.jackson.databind.node.*;
public class DeleteAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
```

```
{
PrintWriter pw=null;
ObjectMapper objectMapper=null;
int code;
try
{
pw=response.getWriter();
response.setContentType("application/json");
objectMapper=new ObjectMapper();
code=Integer.parseInt(request.getParameter("code"));
BookDAO bookDAO=new BookDAO();
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(code);
if(bookWithAuthorExists)
{
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message","Cannot delete author as book exists against it.");
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}
}catch(DAOException daoException)
{
System.out.println(daoException);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message",daoException.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}
catch(Exception exception)
{
System.out.println(exception);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",false);
objectNode.put("error",true);
objectNode.put("message",exception.getMessage());
```

```
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}
try
{
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.getByCode(code);
}catch(DAOException daoException)
{
System.out.println(daoException);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",true);
objectNode.put("message","Author deleted.");
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}
try
{
authorDAO.remove(code);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",true);
objectNode.put("message","Author deleted.");
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}catch(DAOException daoException)
{
System.out.println(daoException);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",true);
objectNode.put("error",false);
objectNode.put("message",daoException.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
return;
}
}catch(Exception exception)
{
System.out.println(exception);
ObjectNode objectNode=objectMapper.createObjectNode();
objectNode.put("success",false);
objectNode.put("exception",false);
objectNode.put("error",true);
objectNode.put("message",exception.getMessage());
String jsonString=objectNode.toString();
pw.print(jsonString);
pw.flush();
pw.close();
}
}
}
```

Add the following servlet mappings to web.xml

```
<servlet>
<servlet-name>UpdateAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateAuthor</servlet-name>
<url-pattern>/updateAuthor</url-pattern>
</servlet-mapping>
<servlet>
<servlet-name>DeleteAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteAuthor</servlet-name>
<url-pattern>/deleteAuthor</url-pattern>
</servlet-mapping>
```

custom-additions.js (jquery-3.2.1 folder)

```
jQuery.fn.insertAt=function(index, element)
{
var lastIndex=this.children().length;
if(index<0)
{
index=Math.max(0,lastIndex+1+index);
}
this.append(element);
if(index<lastIndex)
{
```

```
this.children().eq(index).before(this.children().last());
}
return this;
}
```

entities.js (js folder)

```
function Author(code,name)
{
this.code=code;
this.name=name;
}
function Book(code,title,author,category,price)
{
this.code=code;
this.title=title;
this.author=author;
this.category=category;
this.price=price;
}
```

common.js (js folder)

```
function hasClass(element,className)
{
if(element.classList)
return element.classList.contains(className);
else
return !!element.className.match(new RegExp('(\\s|^)'+className+'(\\s|$)'));
}
function addClass(element,className)
{
if(element.classList)
element.classList.add(className)
else if(!hasClass(element, className))element.className+=(" "+className);
}
function removeClass(element,className)
{
if(element.classList)
element.classList.remove(className);
else if(hasClass(element,className))
{
var reg=new RegExp('(\\s|^)'+className+'(\\s|$)');
element.className=el.className.replace(reg,' ');
}
}
function isElementInViewport(parent, element)
{
var elementRect = element.getBoundingClientRect();
var parentRect = parent.getBoundingClientRect();
```

```
return elementRect.top>=parentRect.top && elementRect.left>=parentRect.left &&
elementRect.bottom<=parentRect.bottom && elementRect.right<=parentRect.right;
}
function selectOption(element,value)
{
var i;
for(i=0;i<element.options.length;i++)
{
element.options[i].removeAttribute("selected");
}
for(i=0;i<element.options.length;i++)
{
if(element.options[i].value==value)
{
element.options[i].setAttribute("selected","");
break;
}
}
}
var Modes={
NONE: 0,
VIEW_MODE: 1,
ADD_MODE: 2,
EDIT_MODE: 3,
DELETE_MODE: 4,
ERROR_MODE: 7
}

function Component(element)
{
this.inDOM=null;
if(element)
{
this.parent=element.parent();
this.element=element;
this.index=this.element.index();
this.inDOM=true;
}
this.isInDOM=function() {
return this.inDOM;
};
this.find=function(lookFor) {
if(this.element && this.inDOM) return this.element.find(lookFor);
return null;
};
this.add=function(){
if(this.element && !this.inDOM)
{
this.parent.insertAt(this.index,element);
this.inDOM=true;
}
};
this.remove=function(){
if(this.element && this.inDOM)
{
this.element.remove();
this.inDOM=false;
}
};
this.hide=function(){
if(this.element && this.inDOM)  this.element.hide();
};
this.show=function(){
if(this.element && this.inDOM) this.element.show();
};
}

function Model()
{
this.searchError=null;
this.collection=null;
this.selected=null;
this.pageSize=10;
this.mode=Modes.NONE;
this.pageNumber=0;
this.exception="";
this.deletedBySomeoneElse=false;
}
```

in js folder create a folder named as author

service.js (js\author folder)

```
function AuthorService()
{
this.getAuthors=function(successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: "GET",
url: "getAuthors",
data: {
now: new Date()
},
success: function(data){
if(data.success)
{
successCallBack(data.authors);
}
else
{
exceptionCallBack(data.message);
}
},
error: function(){
errorCallBack();
}
});
}; // getAuthor ends
this.addAuthor=function(author,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'addAuthor',
data: author,
success: function(data){
if(data.success)
{
successCallBack(data.code);
}
if(data.exception)
{
exceptionCallBack(data.message);
}
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // addAuthor ends
this.updateAuthor=function(author,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'updateAuthor',
data: author,
success: function(data){
if(data.success)
{
successCallBack(data.message);
}
if(data.exception)
{
if(data.delete) exceptionCallBack(data.message,data.delete);
else
exceptionCallBack(data.message)
}
```

```
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // updateAuthor ends
this.deleteAuthor=function(code,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'deleteAuthor',
data: { "code" : code },
success: function(data){
if(data.success)
{
successCallBack(data.message);
}
else
{
if(data.exception)
{
exceptionCallBack(data.message)
}
}
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // deleteAuthor ends
} // service ends
```

controller.js (js\author folder)

```
function AuthorController()
{
// setting up service
var authorService=new AuthorService();
// setting up model
var model=new Model();
// setting up view
var authorView=new AuthorView(this,model);
this.selectAuthor=function(code) {
for(var i=0;i<model.collection.length;i++)
{
if(model.collection[i].code==code) break;
}
if(i<model.collection.length)
{
```

```
model.selected=model.collection[i];
}
else
{
model.selected=null;
}
}
this.searchAuthor=function(authorNameLeftPart) {
if($.trim(authorNameLeftPart).length==0)
{
model.searchError=null;
return;
}
authorNameLeftPart=authorNameLeftPart.toLowerCase();
var y=model.collection.findIndex(function(author){
return author.name.toLowerCase().startsWith(authorNameLeftPart);
});
if(y==-1)
{
model.searchError=true;
model.selected=null;
}
else
{
model.searchError=false;
model.selected=model.collection[y];
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
}
}
this.setPageNumber=function(pageNumber)
{
if(model.pageNumber!=pageNumber)
{
model.pageNumber=pageNumber;
}
}
this.getAuthors=function() {
authorService.getAuthors(function(authors){
model.collection=authors;
model.mode=Modes.VIEW_MODE;
},function(exception){
model.collection=[];
model.mode=Modes.VIEW_MODE;
},function(){
model.mode=Modes.ERROR_MODE;
});
```

```
};
this.setViewMode=function()
{
model.mode=Modes.VIEW_MODE;
};
this.setAddMode=function()
{
model.mode=Modes.ADD_MODE;
};
this.setEditMode=function()
{
model.mode=Modes.EDIT_MODE;
}
this.setDeleteMode=function()
{
model.mode=Modes.DELETE_MODE;
}
this.addAuthor=function(author)
{
authorService.addAuthor(author,function(code){
author.code=code;
var locator=function(a) {
if(a.name.localeCompare(author.name)>0)
{
return true;
}
return false;
};
var y=model.collection.findIndex(locator);
if(y!=-1)
{
model.collection.splice(y,0,author);
}
else
{
model.collection.push(author);
y=model.collection.length-1;
}
model.mode=Modes.VIEW_MODE;
model.selected=author;
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
},function(exception){
model.exception=exception;
},function(){
model.mode=Modes.ERROR_MODE;
});
```

```
};
this.updateAuthor=function(author)
{
authorService.updateAuthor(author,function(message){
var locatorByCode=function(a) {
return a.code==author.code;
};
var f=model.collection.findIndex(locatorByCode);
if(f!=-1)
{
model.collection.splice(f,1);
}
var locator=function(a) {
if(a.name.localeCompare(author.name)>0)
{
return true;
}
return false;
};
var y=model.collection.findIndex(locator);
if(y!=-1)
{
model.collection.splice( y,0,author);
}
else
{
model.collection.push(author);
y=model.collection.length-1;
}
model.mode=Modes.VIEW_MODE;
model.selected=author;
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
},function(exception,authorDeleted){
if(authorDeleted)
{
model.selected=null;
model.exception=exception;
model.deletedBySomeoneElse=true;
var locatorByCode=function(a) {
return a.code==author.code;
};
var f=model.collection.findIndex(locatorByCode);
if(f!=-1)
{
model.collection.splice(f,1);
}
}
else
{
model.exception=exception;
}
},function(){
model.mode=Modes.ERROR_MODE;
});
};
this.deleteAuthor=function()
{
authorService.deleteAuthor(model.selected.code,function(message){
var locatorByCode=function(a) {
return a.code==model.selected.code;
};
var f=model.collection.findIndex(locatorByCode);
if(f!=-1)
{
model.collection.splice(f,1);
}
model.mode=Modes.VIEW_MODE;
model.selected=null;
var numberOfPages=model.collection.length/model.pageSize;
if(model.collection.length%model.pageSize!=0) numberOfPages++;
if(model.pageNumber>numberOfPages) model.pageNumber=model.pageNumber-1;
},function(exception){
model.exception=exception;
},function(){
model.mode=Modes.ERROR_MODE;
});
};
// setupObserver
watch(model, function(property,action,newValue,oldValue){
authorView.updateView(model,property);
});
// loading authors
this.getAuthors();
}
```

view.js (js\author folder)

```
function AuthorView(authorController,model)
{
var dialog=null;
// following variable is required tabulators getPage is not working
var pageNumber=-1;
// initializing authorsList
var editIcon=function(cell,formatterParams){
```

```
var $i=$('<i>');
$i.attr("class","fa fa-edit");
$i.click(function(){
authorController.setEditMode();
});
return $i;
};
var deleteIcon=function(cell,formatterParams){
var $i=$('<i>');
$i.attr("class","fa fa-trash");
$i.click(function(){
authorController.setDeleteMode();
});
return $i;
};
function initializeTabulator()
{
$("#authorsList").tabulator({
pagination: 'local',
paginationSize: model.pageSize,
columns: [
{title: 'S.No.',formatter:"rownum",width: 60,headerSort: false,align: 'right' },
{title: 'Author',field: 'name',width: 400, headerSort: false},
{title: 'Edit',formatter:editIcon,width:60,headerSort: false,align: 'center'},
{title: 'Delete',formatter:deleteIcon,width:60,headerSort: false,align: 'center'}],
index: 'code',
selectable: 1,
resizableColumns:false,
rowClick:function (e,row){
authorController.selectAuthor(row.getIndex());
},
pageLoaded:function(pageNumber){
authorController.setPageNumber(pageNumber);
}
});
}
// Initialization of authorsList complete
// setup view DS
var components={
};
components.headerComponent=new Component($("#headerComponent"));
components.footerComponent=new Component($("#footerComponent"));
components.viewComponent=new Component($("#viewComponent"));
components.titleComponent=new
Component(components.viewComponent.find("div[id='titleComponent']"));
components.listComponent=new
Component(components.viewComponent.find("div[id='listComponent']"));
```

```
components.addComponent=new
Component(components.viewComponent.find("div[id='addComponent']"));
components.editComponent=new
Component(components.viewComponent.find("div[id='editComponent']"));
components.deleteComponent=new
Component(components.viewComponent.find("div[id='deleteComponent']"));
components.emptyListComponent=new
Component(components.viewComponent.find("div[id='emptyListComponent']"));
components.errorComponent=new
Component(components.viewComponent.find("div[id='errorComponent']"));
components.listComponent.remove();
components.emptyListComponent.remove();
components.errorComponent.remove();
components.addComponent.hide();
components.editComponent.hide();
components.deleteComponent.hide();
components.viewComponent.show();
// the function responsible for updating the view
this.updateView=function(authorModel,property)
{
if(property==='collection')
{
property='mode';
} // collection ends
if(property=='searchError')
{
if(model.searchError)
{
$
("#authorToSearch").removeClass("searchComponentInput").addClass("searchComponentInputError")
;
}
else
{
$
("#authorToSearch").removeClass("searchComponentInputError").addClass("searchComponentInput")
;
}
}
if(property=='selected')
{
if(model.selected)
{
$("#authorsList").tabulator('selectRow',model.selected.code);
}
else
{
```

```
$("#authorsList").tabulator('deselectRow');
}
}// selected ends
if(property=='pageSize')
{
// we are not enabling the user to change page size, hence
// let us just ignore this notification right now
return;
}// pageSize ends
if(property=='pageNumber')
{
if(pageNumber!=model.pageNumber)
{
$("#authorsList").tabulator('setPage',model.pageNumber)
pageNumber=model.pageNumber;
}
return;
}// pageNumber ends
if(property=='exception')
{
if(model.mode==Modes.ADD_MODE)
{
$("#authorAddForm").find("div[id='formErrorSection']").html(model.exception);
}
if(model.mode==Modes.EDIT_MODE)
{
$("#authorEditForm").find("div[id='formErrorSection']").html(model.exception);
}
if(model.mode==Modes.DELETE_MODE)
{
$("#authorDeleteForm").find("div[id='formErrorSection']").html(model.exception);
}
return;
}
if(property=='deletedBySomeoneElse')
{
if(model.deletedBySomeoneElse)
{
$("#authorEditForm").find("table[id='authorEditFormTable']").hide();
}
return;
}
if(property=='mode')
{
if(authorModel.mode==Modes.ADD_MODE)
{
showModal("addComponent",300,150,"Add author","",clearAuthorAddForm);
```

```
return;
}
if(authorModel.mode==Modes.EDIT_MODE)
{
var authorEditForm=$("#authorEditForm");
authorEditForm.find("input[id='code']").val(model.selected.code);
authorEditForm.find("input[id='name']").val(model.selected.name);
showModal("editComponent",300,150,"Edit author","",clearAuthorEditForm);
return;
}
if(authorModel.mode==Modes.DELETE_MODE)
{
var authorDeleteForm=$("#authorDeleteForm");
authorDeleteForm.find("input[id='code']").val(model.selected.code);
authorDeleteForm.find("span[id='name']").html(model.selected.name);
showModal("deleteComponent",400,160,"Delete author","",clearAuthorDeleteForm);
return;
}
if(authorModel.mode==Modes.VIEW_MODE)
{
if(model.collection.length>0)
{
components.addComponent.hide();
components.emptyListComponent.remove();
components.listComponent.add();
components.listComponent.show();
}
else
{
components.addComponent.hide();
components.listComponent.remove();
components.emptyListComponent.add();
components.emptyListComponent.show();
}
if(dialog) dialog.dialog('close');
// the following is just a trick to determine if tabulator has been initialized
if($("#authorsList").html()!==undefined)
{
if($("#authorsList").html().length<=1) initializeTabulator();
}
$("#authorsList").tabulator('setData',model.collection);
$("#authorsList").tabulator('setPage',model.pageNumber)
if(model.selected)
{
$("#authorsList").tabulator('selectRow',model.selected.code);
}
else
```

```
{
$("#authorsList").tabulator('deselectRow');
}
return;
} // VIEW_MODE ends
if(authorModel.mode==Modes.ERROR_MODE)
{
components.headerComponent.remove();
components.footerComponent.remove();
components.titleComponent.remove();
components.listComponent.remove();
components.addComponent.remove();
components.editComponent.remove();
components.deleteComponent.remove();
components.emptyListComponent.remove();
components.errorComponent.add();
components.errorComponent.show();
} //  ERROR_MODE ends
} // mode ends
};
function showModal(d,w,h,t,f,cleaner)
{
dialog=$("#"+d);
$("#"+d).dialog({
modal:true,
width: w,
height: h,
title: t,
footer: f,
closeOnEscape: false,
close: function(){
dialog=null;
cleaner();
authorController.setViewMode();
}
});
}
function clearAuthorAddForm()
{
var authorAddForm=$("#authorAddForm")
authorAddForm.trigger("reset");
authorAddForm.find("div[id='formErrorSection']").html("");
authorAddFormValidator.resetForm();
}
function clearAuthorEditForm()
{
var authorEditForm=$("#authorEditForm");
```

```
authorEditForm.trigger("reset");
authorEditForm.find("div[id='formErrorSection']").html("");
authorEditForm.find("table[id='authorEditFormTable']").show();
authorEditFormValidator.resetForm();
}
function clearAuthorDeleteForm()
{
var authorDeleteForm=$("#authorDeleteForm");
authorDeleteForm.find("div[id='formErrorSection']").html("");
}
} // View ends
```

Authors.js (js\author folder)

```
// setting up add form validator
var authorAddFormValidator=null;
var authorEditFormValidator=null;
$(function(){
authorAddFormValidator=$("#authorAddForm").validate({
rules: {
name: {
required: true,
maxlength: 35
}
},
messages: {
name: {
required: "Required",
maxlength: "Cannot exceed 35"
}
}
});
authorEditFormValidator=$("#authorEditForm").validate({
rules: {
name: {
required: true,
maxlength: 35
}
},
messages: {
name: {
required: "Required",
maxlength: "Cannot exceed 35"
}
}
});
});
```

```
function addAuthor()
{
if(!$("#authorAddForm").valid()) return;
var name=$("#authorAddForm").find("input[name='name']").val();
var author={
code: 0,
name: name
};
authorController.addAuthor(author);
}


function updateAuthor()
{
var authorEditForm=$("#authorEditForm");
if(!authorEditForm.valid()) return;
var code=authorEditForm.find("input[name='code']").val();
var name=authorEditForm.find("input[name='name']").val();
var author={
code: code,
name: name
};
authorController.updateAuthor(author);
}
var authorController=null;
function initialize()
{
authorController=new AuthorController();
}
$(document).ready(initialize);
```

styles.css (css folder)

```
body {
background: #F5F5F5;
}

.home
{
padding: 5px;
}

.headerComponent {
background: #DBDBDB;
width:100%;
}
```

```
.header table
{
width: 100%;
}

.footerComponent {
background: #DBDBDB;
width:100%;
padding:10px;
text-align: center;
margin-top: 20px;
}

.error {
color: red;
}
.ui-widget-overlay {
opacity: .50;
filter: Alpha(opacity=50);
background: rgb(50,50,50);
}
.viewComponent
{
}
.errorComponent
{
display: none;
}
.titleComponent
{
}
.toolBar
{
width:100%;
border: none;
background-color: #C1C1C1;
height: 28px;
padding-top:5px;
}

.toolBarButtons
{
float: right;
cursor: pointer;
padding-right:5px;
}
```

```
.toolBarButtons button
{
cursor: pointer;
}

.searchComponent
{
float: left;
padding-left:5px;
}

.searchComponentInput
{
border: 2px solid black;
}

.searchComponentInputError
{
border: 2px solid red;
}

.listComponent
{

}
.addComponent {
}
.editComponent {
}
.deleteComponent {
}
.emptyListComponent
{
width: 100%;
font-size: 24px;
text-align: center;
}
```

MasterPageTopSection.jsp (crudfour folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
```

```
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/jquery-ui-1.12.1/jquery-ui.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/tabulator/css/tabulator.min.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/font-awesome-4.7.0/css/font-awesome.min.css' >
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
<script src='/${contextName}/jquery-3.2.1/jquery-3.2.1.min.js'></script>
<script src='/${contextName}/jquery-3.2.1/custom-additions.js'></script>
<script src='/${contextName}/jquery-ui-1.12.1/jquery-ui.min.js'></script>
<script src='/${contextName}/jquery-validation/jquery.validate.min.js'></script>
<script src='/${contextName}/jquery-validation/additional-methods.min.js'></script>
<script src='/${contextName}/tabulator/js/tabulator.min.js'></script>
<script src='/${contextName}/watch/watch.js'></script>
<script src='/${contextName}/js/common.js'></script>
</head>
<body>
<div id='headerComponent' class='headerComponent'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
<a href='/${contextName}/'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp

```
<div id='footerComponent' class='footerComponent'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

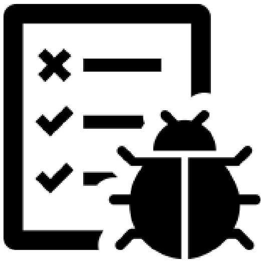
Authors.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/js/entities.js'></script>
<script src='/${contextName}/js/author/service.js'></script>
<script src='/${contextName}/js/author/controller.js'></script>
<script src='/${contextName}/js/author/view.js'></script>
<script src='/${contextName}/js/author/Authors.js'></script>
```

```
<div id='viewComponent' class='viewComponent'>
<div id='titleComponent' class='titleComponent'>
<h1>Authors</h1>
</div>
<div id='listComponent' class='listComponent'>
<div id='toolBar' class='toolBar'>
<div class='searchComponent'>
Search
<input id='authorToSearch' class='searchComponentInput' type='text' maxlength='35' size='36'
onkeyup='authorController.searchAuthor(this.value)'>
</div>
<div class='toolBarButtons'>
<button type='button' onclick='authorController.setAddMode()'><i class='fa fa-plus'></i></button>
</div>
</div>
<div id='authorsList'></div>
</div>

<div id='addComponent' class='addComponent'>
<form id='authorAddForm'>
<div id='formErrorSection' class='error'></div>
<table>
<tr>
<td>Name</td>
<td><input type='text' name='name' id='name'></td>
</tr>
<tr>
<td colspan='2' align='center'><button type='button' onclick='addAuthor()'><i class='fa fa-floppy-
o'></i></button></td>
</tr>
</table>
</form>
</div>

<div id='editComponent' class='editComponent'>
<form id='authorEditForm'>
<div id='formErrorSection' class='error'></div>
<input type='hidden' id='code' name='code'>
<table id='authorEditFormTable'>
<tr>
<td>Name</td>
<td><input type='text' name='name' id='name'></td>
</tr>
<tr>
<td colspan='2' align='center'><button type='button' onclick='updateAuthor()'><i class='fa fa-floppy-
o'></i></button></td>
</tr>
```

```
</table>
</form>
</div>

<div id='deleteComponent' class='deleteComponent'>
<form id='authorDeleteForm'>
<center>
<div id='formErrorSection' class='error'></div>
<input type='hidden' id='code' name='code'>
Delete author <span id='name'></span> ?<br>
<br>
<button type='button' onclick='authorController.deleteAuthor()'>Yes</i></button>
<button type='button' onclick='authorController.setViewMode()'>No</i></button>
</center>
</form>
</div>
<div id='emptyListComponent' class='emptyListComponent'>
No authors<br>
<button type='button' onclick='authorController.setAddMode()'><i class='fa fa-plus'></i></button>
</div>
<div id='errorComponent' class='errorComponent'>
<center>
<img src='images/error.jpg'>
</center>
</div>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

Bugs in CRUD Four (Author Module)



1) If a record is being updated and it has been deleted by someone working in parallel, then the module notifies that such kind of thing has happened and everything works well, but immediately one more record is being updated and that record also has been deleted by someone working in parallel, then the module gets stucck.
2) If a record is selected in the grid and the page is changed, the one that was selected does not get deselected. Right now it doesn't seem like a bug but it will be problematic in master/detail panel type module like the (book module).
3) The Home Hyperlink doesn't appear at the right side.
4) When the add and update modals appear, the background doesn't look disabled / blur.

Fixing the bugs in CRUD Four (Author Module)

Solution to problem 1 : Change the setEditMode function of the controller as follows

```
this.setEditMode=function()
{
model.mode=Modes.EDIT_MODE;
model.deletedBySomeoneElse=false;
model.exception="";
}
```

Solution to problem 2 : Change the setPageNumber function of the controller as follows

```
this.setPageNumber=function(pageNumber)
{
if(model.pageNumber!=pageNumber)
{
model.pageNumber=pageNumber;
model.selected=null;
}
}
```

Solution to problem 2 : In the styles.css file just after the headerComponent class is the header class with table tag as follows

```
.header table
{
width: 100%;
}
```

change the class name to headerComponent as follows

```
.headerComponent table
{
width: 100%;
}
```

Solution to problem 4 : It is a silly mistake, in MasterPageTopSection.jsp I have included the styles.css before the jquery files, just place the line to include styles.css in the end, just before the </head> tag

<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >

Books.jsp

Note : I have not separated the javascript code in js\book\service.js, js\book\controller.js, js\book\view.js and js\book\Books.js, you are supposed to do that after removing the bugs and completing the assignment given after Books.jsp

I have also not separated the css from Books.jsp, you do that and put it in css\styles.css

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/js/entities.js'></script>
<script src='/${contextName}/js/author/service.js'></script>
<style>
.bookDetails
{
font-size:14pt;
padding: 10px;
border: 1px solid #AAAAAA;
}
.bookDetails .title
{
font-size:14pt;
font-weight: bold;
background: #AAAAAA;
padding: 5px;
}
.bookDetails .key
```

```
{
font-size:14pt;
font-weight: bold;
}
.bookDetails table td
{
padding : 7px;
}
.deleteComponent table
{
margin-top: 10px;
}
.deleteComponent table td
{
padding: 5px;
}
</style>
<script>
// service starts
function BookService()
{
this.getBooks=function(successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: "GET",
url: "getBooks",
data: {
now: new Date()
},
success: function(data){
if(data.success)
{
successCallBack(data.books);
}
else
{
exceptionCallBack(data.message);
}
},
error: function(){
errorCallBack();
}
});
}; // getBooks ends
this.addBook=function(book,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'addBook',
data: {
code: 0,
title: book.title,
category: book.category,
price: book.price,
authorCode: book.author.code
},
success: function(data){
if(data.success)
{
successCallBack(data.code);
}
if(data.exception)
{
exceptionCallBack(data.message);
}
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // addBook ends
this.updateBook=function(book,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'updateBook',
data: {
code: book.code,
title: book.title,
category: book.category,
price: book.price,
authorCode: book.author.code
},
success: function(data){
if(data.success)
{
successCallBack(data.message);
}
if(data.exception)
{
if(data.delete) exceptionCallBack(data.message,data.delete);
else
exceptionCallBack(data.message)
}
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // updateBook ends
this.deleteBook=function(code,successCallBack,exceptionCallBack,errorCallBack){
$.ajax({
type: 'POST',
url: 'deleteBook',
data: { "code" : code },
success: function(data){
if(data.success)
{
successCallBack(data.message);
}
else
{
if(data.exception)
{
exceptionCallBack(data.message)
}
}
},
error: function(xhr,status,error){
errorCallBack();
}
});
}; // deleteBook ends*/
} // service ends
// controller starts
function BookController()
{
// setting up service
var bookService=new BookService();
var authorService=new AuthorService();
// setting up model
var model=new Model();
model.pageSize=4;
// setting up view
var bookView=new BookView(this,model);
this.selectBook=function(code) {
for(var i=0;i<model.collection.length;i++)
{
if(model.collection[i].code==code) break;
}
if(i<model.collection.length)
{
model.selected=model.collection[i];
}
```

```
else
{
model.selected=null;
}
}
this.searchBook=function(bookTitleLeftPart) {
if($.trim(bookTitleLeftPart).length==0)
{
model.searchError=null;
return;
}
bookTitleLeftPart=bookTitleLeftPart.toLowerCase();
var y=model.collection.findIndex(function(book){
return book.title.toLowerCase().startsWith(bookTitleLeftPart);
});
if(y==-1)
{
model.searchError=true;
model.selected=null;
}
else
{
model.searchError=false;
model.selected=model.collection[y];
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
}
}
this.setPageNumber=function(pageNumber)
{
if(model.pageNumber!=pageNumber)
{
model.pageNumber=pageNumber;
model.selected=null;
}
}
// dynamically adding property before setting up the observer
model.authors=null;
model.allBooks=null;
model.filterByAuthor=null;
function getAuthors()
{
authorService.getAuthors(function(authors){
model.authors=authors;
},function(exception){
model.authors=[];
},function(){
```

```
model.mode=Modes.ERROR_MODE;
});
};
function getBooks()
{
bookService.getBooks(function(books){
model.allBooks=books;
var books=[];
for(var i=0;i<model.allBooks.length;i++)
{
books.push(model.allBooks[i]);
}
model.collection=books;
model.mode=Modes.VIEW_MODE;
},function(exception){
model.collection=[];
model.allBooks=[];
model.mode=Modes.VIEW_MODE;
},function(){
model.mode=Modes.ERROR_MODE;
});
};
this.applyFilter=function(authorCode)
{
model.filterByAuthor=null;
if(authorCode>0)
{
for(var i=0;i<model.authors.length;i++)
{
if(model.authors[i].code==authorCode)
{
model.filterByAuthor=model.authors[i];
break;
}
}
}
var selected=model.selected;
model.selected=null;
model.pageNumber=0;
var collection=[];
if(authorCode==0)
{
for(var i=0;i<model.allBooks.length;i++)
{
collection.push(model.allBooks[i]);
}
}
```

```
else
{
for(var i=0;i<model.allBooks.length;i++)
{
if(model.allBooks[i].author.code==authorCode)
{
collection.push(model.allBooks[i]);
}
}
}
model.collection=collection;
if(selected!=null && (authorCode==0 || selected.author.code==authorCode))
{
y=model.collection.findIndex(function(book){
return book.code==selected.code;
});
if(y!=-1)
{
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
model.selected=selected;
}
}
}
this.setViewMode=function()
{
model.mode=Modes.VIEW_MODE;
};
this.setAddMode=function()
{
model.mode=Modes.ADD_MODE;
};
this.setEditMode=function()
{
model.mode=Modes.EDIT_MODE;
}
this.setDeleteMode=function()
{
model.mode=Modes.DELETE_MODE;
}
this.clearFilter=function(){
this.applyFilter(0);
}
this.addBook=function(book)
{
bookService.addBook(book,function(code){
book.code=code;
```

```
var locator=function(a) {
if(a.title.localeCompare(book.title)>0)
{
return true;
}
return false;
};
var y=model.allBooks.findIndex(locator);
if(y!=-1)
{
model.allBooks.splice(y,0,book);
}
else
{
model.allBooks.push(book);
}
y=model.collection.findIndex(locator);
if(y!=-1)
{
model.collection.splice(y,0,book);
}
else
{
model.collection.push(book);
y=model.collection.length-1;
}
model.mode=Modes.VIEW_MODE;
model.selected=book;
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
},function(exception){
model.exception=exception;
},function(){
model.mode=Modes.ERROR_MODE;
});
};
this.updateBook=function(book)
{
bookService.updateBook(book,function(message){
var locatorByCode=function(a) {
return a.code==book.code;
};
var f=model.allBooks.findIndex(locatorByCode);
if(f!=-1)
{
model.allBooks.splice(f,1);
}
```

```
f=model.collection.findIndex(locatorByCode);
if(f!=-1)
{
model.collection.splice(f,1);
}
var locator=function(a) {
if(a.title.localeCompare(book.title)>0)
{
return true;
}
return false;
};
var y=model.allBooks.findIndex(locator);
if(y!=-1)
{
model.allBooks.splice( y,0,book);
}
else
{
model.allBooks.push(book);
}
y=model.collection.findIndex(locator);
if(y!=-1)
{
model.collection.splice( y,0,book);
}
else
{
model.collection.push(book);
y=model.collection.length-1;
}
model.mode=Modes.VIEW_MODE;
model.selected=book;
var foundOnPage=Math.floor(y/model.pageSize)+1;
if(model.pageNumber!=foundOnPage) model.pageNumber=foundOnPage;
},function(exception,bookDeleted){
if(bookDeleted)
{
model.selected=null;
model.exception=exception;
model.deletedBySomeoneElse=true;
var locatorByCode=function(a) {
return a.code==book.code;
};
var f=model.allBooks.findIndex(locatorByCode);
if(f!=-1)
{
```

```
model.allBooks.splice(f,1);
}
f=model.collection.findIndex(locatorByCode);
if(f!=-1)
{
model.collection.splice(f,1);
}
}
else
{
model.exception=exception;
}
},function(){
model.mode=Modes.ERROR_MODE;
});
};
this.deleteBook=function()
{
bookService.deleteBook(model.selected.code,function(message){
var locatorByCode=function(a) {
return a.code==model.selected.code;
};
var f=model.allBooks.findIndex(locatorByCode);
if(f!=-1)
{
model.allBooks.splice(f,1);
}
f=model.collection.findIndex(locatorByCode);
if(f!=-1)
{
model.collection.splice(f,1);
}
model.mode=Modes.VIEW_MODE;
model.selected=null;
var numberOfPages=model.collection.length/model.pageSize;
if(model.collection.length%model.pageSize!=0) numberOfPages++;
if(model.pageNumber>numberOfPages) model.pageNumber=model.pageNumber-1;
},function(exception){
model.exception=exception;
},function(){
model.mode=Modes.ERROR_MODE;
});
};
// setupObserver
watch(model, function(property,action,newValue,oldValue){
bookView.updateView(model,property);
});
```

```
// loading authors
getAuthors();
// loading books
getBooks();
}
// view starts
function BookView(bookController,model)
{
var dialog=null;
// following variable is required tabulators getPage is not working
var pageNumber=-1;
// initializing booksList
var editIcon=function(cell,formatterParams){
var $i=$('<i>');
$i.attr("class","fa fa-edit");
$i.click(function(){
bookController.setEditMode();
});
return $i;
};
var deleteIcon=function(cell,formatterParams){
var $i=$('<i>');
$i.attr("class","fa fa-trash");
$i.click(function(){
bookController.setDeleteMode();
});
return $i;
};
function initializeTabulator()
{
$("#booksList").tabulator({
pagination: 'local',
paginationSize: model.pageSize,
columns: [
{title: 'S.No.',formatter:"rownum",width: 60,headerSort: false,align: 'right' },
{title: 'Book',field: 'title',width: 400, headerSort: false},
{title: 'Edit',formatter:editIcon,width:60,headerSort: false,align: 'center'},
{title: 'Delete',formatter:deleteIcon,width:60,headerSort: false,align: 'center'}],
index: 'code',
selectable: 1,
resizableColumns:false,
rowClick:function (e,row){
bookController.selectBook(row.getIndex());
},
pageLoaded:function(pageNumber){
bookController.setPageNumber(pageNumber);
}
});
}
// Initialization of booksList complete
// setup view DS
var components={
};
components.headerComponent=new Component($("#headerComponent"));
components.footerComponent=new Component($("#footerComponent"));
components.viewComponent=new Component($("#viewComponent"));
components.titleComponent=new
Component(components.viewComponent.find("div[id='titleComponent']"));
components.listComponent=new
Component(components.viewComponent.find("div[id='listComponent']"));
components.addComponent=new
Component(components.viewComponent.find("div[id='addComponent']"));
components.editComponent=new
Component(components.viewComponent.find("div[id='editComponent']"));
components.deleteComponent=new
Component(components.viewComponent.find("div[id='deleteComponent']"));
components.emptyListComponent=new
Component(components.viewComponent.find("div[id='emptyListComponent']"));
components.errorComponent=new
Component(components.viewComponent.find("div[id='errorComponent']"));
components.listComponent.remove();
components.emptyListComponent.remove();
components.errorComponent.remove();
components.addComponent.hide();
components.editComponent.hide();
components.deleteComponent.hide();
components.viewComponent.show();
// the function responsible for updating the view
this.updateView=function(bookModel,property)
{
if(property==='collection')
{
property='mode';
} // collection ends
if(property=='searchError')
{
if(model.searchError)
{
$("#bookToSearch").removeClass("searchComponentInput").addClass("searchComponentInputError");
}
else
{
$("#bookToSearch").removeClass("searchComponentInputError").addClass("searchComponentInput");
}
```

```
}
if(property=='authors')
{
property='mode';
}
if(property=='selected')
{
if(model.selected)
{
$("#booksList").tabulator('selectRow',model.selected.code);
}
else
{
$("#booksList").tabulator('deselectRow');
}
property='mode';
}// selected ends
if(property=='pageSize')
{
// we are not enabling the user to change page size, hence
// let us just ignore this notification right now
return;
}// pageSize ends
if(property=='pageNumber')
{
if(model.pageNumber>0 && pageNumber!=model.pageNumber)
{
$("#booksList").tabulator('setPage',model.pageNumber)
pageNumber=model.pageNumber;
}
if(model.selected==null)
{
$("#bookToSearch").val("");
}
return;
}// pageNumber ends
if(property=='exception')
{
if(model.mode==Modes.ADD_MODE)
{
$("#bookAddForm").find("div[id='formErrorSection']").html(model.exception);
}
if(model.mode==Modes.EDIT_MODE)
{
$("#bookEditForm").find("div[id='formErrorSection']").html(model.exception);
}
if(model.mode==Modes.DELETE_MODE)
```

```
{
$("#bookDeleteForm").find("div[id='formErrorSection']").html(model.exception);
}
return;
}
if(property=='deletedBySomeoneElse')
{
if(model.deletedBySomeoneElse)
{
$("#bookEditForm").find("table[id='bookEditFormTable']").hide();
}
return;
}
if(property=='filterByAuthor')
{
if(bookModel.filterByAuthor==null)
{
$("#filterByAuthor").val(0);
$("#booksList").tabulator('setData',model.collection);
if(model.pageNumber>0)
{
$("#booksList").tabulator('setPage',model.pageNumber)
}
var bookDetails=$("#bookDetails");
var bookDetailsTitle=bookDetails.find("span[id='title']");
var bookDetailsAuthor=bookDetails.find("span[id='author']");
var bookDetailsPrice=bookDetails.find("span[id='price']");
if(model.selected)
{
$("#booksList").tabulator('selectRow',model.selected.code);
bookDetailsTitle.html(model.selected.title+"("+model.selected.category+")");
bookDetailsAuthor.html(model.selected.author.name);
bookDetailsPrice.html(model.selected.price);
}
else
{
$("#booksList").tabulator('deselectRow');
bookDetailsTitle.html("");
bookDetailsAuthor.html("");
bookDetailsPrice.html("");
}
}
else
{
$("#filterByAuthor").val(bookModel.filterByAuthor.code);
}
if(bookModel.mode==Modes.ADD_MODE)
```

```
{
var bookAddForm=$("#bookAddForm");
var authorComboBox=bookAddForm.find("span[id='authorComboBox']");
var authorLabel=bookAddForm.find("span[id='authorLabel']");
var clearFilterHyperlink=bookAddForm.find("span[id='clearFilterHyperlink']");
if(bookModel.filterByAuthor==null)
{
authorLabel.hide();
clearFilterHyperlink.hide();
authorComboBox.show();
}
else
{
authorComboBox.hide();
authorLabel.html(bookModel.filterByAuthor.name);
authorLabel.show();
clearFilterHyperlink.show();
}
}
if(bookModel.mode==Modes.EDIT_MODE)
{
var bookEditForm=$("#bookEditForm");
var authorComboBox=bookEditForm.find("span[id='authorComboBox']");
var authorLabel=bookEditForm.find("span[id='authorLabel']");
if(bookModel.filterByAuthor==null)
{
authorLabel.hide();
authorComboBox.show();
}
else
{
authorComboBox.hide();
authorLabel.html(bookModel.filterByAuthor.name);
authorLabel.show();
}
}
return;
}
if(property=='mode')
{
if(bookModel.mode==Modes.ADD_MODE)
{
var bookAddForm=$("#bookAddForm");
var authorComboBox=bookAddForm.find("span[id='authorComboBox']");
var authorLabel=bookAddForm.find("span[id='authorLabel']");
var clearFilterHyperlink=bookAddForm.find("span[id='clearFilterHyperlink']");
if(bookModel.filterByAuthor==null)
```

```
{
authorLabel.hide();
clearFilterHyperlink.hide();
authorComboBox.show();
}
else
{
bookAddForm.find("select[id='authorCode']").val(model.filterByAuthor.code);
authorComboBox.hide();
authorLabel.html(bookModel.filterByAuthor.name);
authorLabel.show();
clearFilterHyperlink.show();
}
showModal("addComponent",500,250,"Add book","",clearBookAddForm);
return;
}
if(bookModel.mode==Modes.EDIT_MODE)
{
var bookDetails=$("#bookDetails");
var bookDetailsTitle=bookDetails.find("span[id='title']");
var bookDetailsAuthor=bookDetails.find("span[id='author']");
var bookDetailsPrice=bookDetails.find("span[id='price']");
bookDetailsTitle.html(model.selected.title+"("+model.selected.category+")");
bookDetailsAuthor.html(model.selected.author.name);
bookDetailsPrice.html(model.selected.price);
var bookEditForm=$("#bookEditForm");
bookEditForm.find("input[id='code']").val(model.selected.code);
bookEditForm.find("input[id='title']").val(model.selected.title);
bookEditForm.find("select[id='category']").val(model.selected.category);
bookEditForm.find("select[id='authorCode']").val(model.selected.author.code);
bookEditForm.find("input[id='price']").val(model.selected.price);
var bookEditForm=$("#bookEditForm");
var authorComboBox=bookEditForm.find("span[id='authorComboBox']");
var authorLabel=bookEditForm.find("span[id='authorLabel']");
var clearFilterHyperlink=bookEditForm.find("span[id='clearFilterHyperlink']");
if(bookModel.filterByAuthor==null)
{
authorLabel.hide();
clearFilterHyperlink.hide();
authorComboBox.show();
}
else
{
authorComboBox.hide();
authorLabel.html(bookModel.filterByAuthor.name);
authorLabel.show();
clearFilterHyperlink.show();
```

```
}
showModal("editComponent",500,250,"Edit book","",clearBookEditForm);
return;
}
if(bookModel.mode==Modes.DELETE_MODE)
{
var bookDetails=$("#bookDetails");
var bookDetailsTitle=bookDetails.find("span[id='title']");
var bookDetailsAuthor=bookDetails.find("span[id='author']");
var bookDetailsPrice=bookDetails.find("span[id='price']");
bookDetailsTitle.html(model.selected.title+"("+model.selected.category+")");
bookDetailsAuthor.html(model.selected.author.name);
bookDetailsPrice.html(model.selected.price);
var bookDeleteForm=$("#bookDeleteForm");
bookDeleteForm.find("input[id='code']").val(model.selected.code);
bookDeleteForm.find("span[id='title']").html(model.selected.title);
bookDeleteForm.find("span[id='author']").html(model.selected.author.name);
bookDeleteForm.find("span[id='category']").html(model.selected.category);
bookDeleteForm.find("span[id='price']").html(model.selected.price);
showModal("deleteComponent",400,240,"Delete book","",clearBookDeleteForm);
return;
}
if(bookModel.mode==Modes.VIEW_MODE)
{

var bookAddFormAuthor=$("#bookAddForm").find("select[id='authorCode']");
if(bookAddFormAuthor.children("option").length==0)
{
bookAddFormAuthor.empty();
bookAddFormAuthor.append($('<option>', {
   value: "",
   text: '<Select>'
}));
for(var i=0;i<model.authors.length;i++)
{
bookAddFormAuthor.append($('<option>', {
   value: model.authors[i].code,
   text: model.authors[i].name
}));
}
var bookEditFormAuthor=$("#bookEditForm").find("select[id='authorCode']");
bookEditFormAuthor.empty();
bookEditFormAuthor.append($('<option>', {
   value: "",
   text: '<Select>'
}));
for(var i=0;i<model.authors.length;i++)
```

```
{
bookEditFormAuthor.append($('<option>', {
   value: model.authors[i].code,
   text: model.authors[i].name
}));
}
} // condition to populate author combobox in add and edit form ends
if(model.collection.length>0)
{
components.addComponent.hide();
components.emptyListComponent.remove();
components.listComponent.add();
components.listComponent.show();
var filterByAuthor=$("#filterByAuthor");
if(filterByAuthor.children("option").length==0)
{
filterByAuthor.empty();
filterByAuthor.append($('<option>', {
   value: 0,
   text: '<All>'
}));
for(var i=0;i<model.authors.length;i++)
{
filterByAuthor.append($('<option>', {
   value: model.authors[i].code,
   text: model.authors[i].name
}));
}
}// populating authors comboboxes condition ends
}
else
{
components.addComponent.hide();
components.listComponent.remove();
components.emptyListComponent.add();
components.emptyListComponent.show();
}
if(dialog) dialog.dialog('close');
// the following is just a trick to determine if tabulator has been initialized
if($("#booksList").html()!==undefined)
{
if($("#booksList").html().length<=1) initializeTabulator();
}
$("#booksList").tabulator('setData',model.collection);
if(model.pageNumber>0)
{
$("#booksList").tabulator('setPage',model.pageNumber)
```

```
}
var bookDetails=$("#bookDetails");
var bookDetailsTitle=bookDetails.find("span[id='title']");
var bookDetailsAuthor=bookDetails.find("span[id='author']");
var bookDetailsPrice=bookDetails.find("span[id='price']");
if(model.selected)
{
$("#booksList").tabulator('selectRow',model.selected.code);
bookDetailsTitle.html(model.selected.title+"("+model.selected.category+")");
bookDetailsAuthor.html(model.selected.author.name);
bookDetailsPrice.html(model.selected.price);
}
else
{
$("#booksList").tabulator('deselectRow');
bookDetailsTitle.html("");
bookDetailsAuthor.html("");
bookDetailsPrice.html("");
}
return;
} // VIEW_MODE ends
if(bookModel.mode==Modes.ERROR_MODE)
{
components.headerComponent.remove();
components.footerComponent.remove();
components.titleComponent.remove();
components.listComponent.remove();
components.addComponent.remove();
components.editComponent.remove();
components.deleteComponent.remove();
components.emptyListComponent.remove();
components.errorComponent.add();
components.errorComponent.show();
} //  ERROR_MODE ends
} // mode ends
};
function showModal(d,w,h,t,f,cleaner)
{
dialog=$("#"+d);
$("#"+d).dialog({
modal:true,
width: w,
height: h,
title: t,
footer: f,
closeOnEscape: false,
close: function(){
```

```
dialog=null;
cleaner();
bookController.setViewMode();
}
});
}
function clearBookAddForm()
{
var bookAddForm=$("#bookAddForm")
bookAddForm.trigger("reset");
bookAddForm.find("div[id='formErrorSection']").html("");
bookAddFormValidator.resetForm();
}
function clearBookEditForm()
{
var bookEditForm=$("#bookEditForm");
bookEditForm.trigger("reset");
bookEditForm.find("div[id='formErrorSection']").html("");
bookEditForm.find("table[id='bookEditFormTable']").show();
bookEditFormValidator.resetForm();
}
function clearBookDeleteForm()
{
var bookDeleteForm=$("#bookDeleteForm");
bookDeleteForm.find("div[id='formErrorSection']").html("");
}
} // View ends
// general starts
// setting up add form validator
var bookAddFormValidator=null;
var bookEditFormValidator=null;
$(function(){
bookAddFormValidator=$("#bookAddForm").validate({
rules: {
title: {
required: true,
maxlength: 35
},
category: {
required: true
},
authorCode: {
required: true
},
price: {
required: true,
number:true,
min: 0
}
},
messages: {
title: {
required: "Required",
maxlength: "Cannot exceed 35"
},
category : {
required: "Required"
},
authorCode: {
required: "Required"
},
price: {
required: "Required",
number: "Number required",
min: "Invalid"
}
}
});
bookEditFormValidator=$("#bookEditForm").validate({
rules: {
title: {
required: true,
maxlength: 35
}
},
messages: {
title: {
required: "Required",
maxlength: "Cannot exceed 35"
}
}
});
});
function addBook()
{
var bookAddForm=$("#bookAddForm");
if(!bookAddForm.valid()) return;
var title=bookAddForm.find("input[name='title']").val();
var category=bookAddForm.find("select[name='category']").val();
var authorCode=bookAddForm.find("select[name='authorCode']").val();
var authorName=bookAddForm.find("select[name='authorCode']").find("option:selected").text();
var price=bookAddForm.find("input[name='price']").val();
var book={
code: 0,
title: title,
category: category,
author: {
code: authorCode,
name: authorName
},
price: price
};
bookController.addBook(book);
}
function updateBook()
{
var bookEditForm=$("#bookEditForm");
if(!bookEditForm.valid()) return;
var code=bookEditForm.find("input[name='code']").val();
var title=bookEditForm.find("input[name='title']").val();
var category=bookEditForm.find("select[name='category']").val();
var authorCode=bookEditForm.find("select[name='authorCode']").val();
var authorName=bookEditForm.find("select[name='authorCode']").find("option:selected").text();
var price=bookEditForm.find("input[name='price']").val();
var book={
code: code,
title: title,
category: category,
author: {
code: authorCode,
name: authorName
},
price: price
};
bookController.updateBook(book);
}
var bookController=null;
function initialize()
{
bookController=new BookController();
}
$(document).ready(initialize);
</script>
<div id='viewComponent' class='viewComponent'>
<div id='titleComponent' class='titleComponent'>
<h1>Books</h1>
</div>
<div id='listComponent' class='listComponent'>
<div id='toolBar' class='toolBar'>
<div class='searchComponent'>
Search
```

```
<input id='bookToSearch' class='searchComponentInput' type='text' maxlength='35' size='36'
onkeyup='bookController.searchBook(this.value)'>
</div>
<div class='toolBarButtons'>
Author
<select id='filterByAuthor' onchange='bookController.applyFilter(this.value)'></select>
&nbsp;&nbsp;&nbsp;
<button type='button' onclick='bookController.setAddMode()'><i class='fa fa-plus'></i></button>
</div>
</div>
<div id='booksList'></div>
<div id='bookDetails' class='bookDetails'>
<div class='title'>Details</div>
<table>
<tr>
<td class='key'>
Book
</td>
<td>
<span id='title'></span>
</td>
</tr>
<tr>
<td class='key'>
Author
</td>
<td>
<span id='author'></span>
</td>
</tr>
<tr>
<td class='key'>
Price
</td>
<td>
<span id='price'></span>
</td>
</tr>
</table>
</div>
</div>

<div id='addComponent' class='addComponent'>
<form id='bookAddForm'>
<div id='formErrorSection' class='error'></div>
<table>
<tr>
```

```
<td>Title</td>
<td><input type='text' name='title' id='title'></td>
</tr>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'>
<option value="">&lt;Select&gt;</option>
<option value='Science fiction'>Science fiction</option>
<option value='Satire'>Satire</option>
<option value='Drama'>Drama</option>
<option value='Action and Adventure'>Action and Adventure</option>
<option value='Mystery'>Mystery</option>
<option value='Horror'>Horror</option>
</select>
</td>
</tr>
<tr>
<td>Author</td>
<td>
<span id='authorComboBox'><select name='authorCode' id='authorCode'></select></span>
<span id='authorLabel'></span>
<span id='clearFilterHyperlink'><Button type='button' onclick='bookController.clearFilter()'>Clear
filter</button></span>
</td>
</tr>
<tr>
<td>Price</td>
<td>
<input type='number' min='0' name='price' id='price'>
</td>
</tr>
<tr>
<td colspan='2' align='center'><button type='button' onclick='addBook()'><i class='fa fa-floppy-
o'></i></button></td>
</tr>
</table>
</form>
</div>
<div id='editComponent' class='editComponent'>
<form id='bookEditForm'>
<div id='formErrorSection' class='error'></div>
<input type='hidden' id='code' name='code'>
<table id='bookEditFormTable'>
<tr>
<td>Title</td>
<td><input type='text' name='title' id='title'></td>
```

```
</tr>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'>
<option value="">&lt;Select&gt;</option>
<option value='Science fiction'>Science fiction</option>
<option value='Satire'>Satire</option>
<option value='Drama'>Drama</option>
<option value='Action and Adventure'>Action and Adventure</option>
<option value='Mystery'>Mystery</option>
<option value='Horror'>Horror</option>
</select>
</td>
</tr>
<tr>
<td>Author</td>
<td>
<span id='authorComboBox'><select name='authorCode' id='authorCode'></select></span>
<span id='authorLabel'></span>
<span id='clearFilterHyperlink'><Button type='button' onclick='bookController.clearFilter()'>Clear filter</button></span>
</td>
</tr>
<tr>
<td>Price</td>
<td>
<input type='number' min='0' name='price' id='price'>
</td>
</tr>
<tr>
<td colspan='2' align='center'><button type='button' onclick='updateBook()'><i class='fa fa-floppy-o'></i></button></td>
</tr>
</table>
</form>
</div>

<div id='deleteComponent' class='deleteComponent'>
<form id='bookDeleteForm'>
<div id='formErrorSection' class='error'></div>
<input type='hidden' id='code' name='code'>
<b>Delete book <span id='title'></span> ?</b><br>
<table>
<tr>
<td>Author</td><td><span id='author'></span></td>
</tr>
```

```
<tr>
<td>Category</td><td><span id='category'></span></td>
</tr>
<tr>
<td>Price</td><td><span id='price'></span></td>
</tr>
</table>
<br>
<button type='button' onclick='bookController.deleteBook()'>Yes</i></button>
<button type='button' onclick='bookController.setViewMode()'>No</i></button>
</form>
</div>
<div id='emptyListComponent' class='emptyListComponent'>
No books<br>
<button type='button' onclick='bookController.setAddMode()'><i class='fa fa-plus'></i></button>
</div>
<div id='errorComponent' class='errorComponent'>
<center>
<img src='images/error.jpg'>
</center>
</div>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

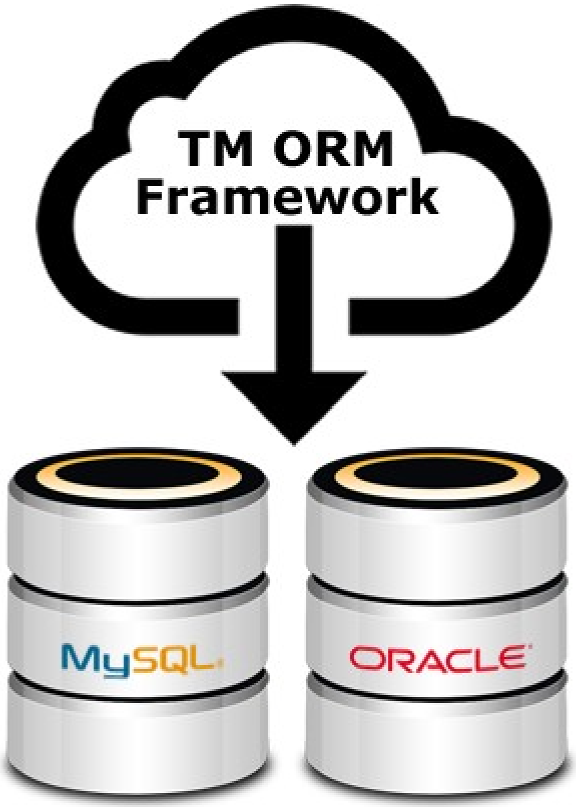
Bugs in CRUD Four (Book Module)

Watch the video carefully, the end part, If a filter is applied and no books exists against that Author (for eg. Filter by Dan Brown) , then the message No Books appear whereas the message should be **No Books of Dan Brown exist**. Then the user can add book, Now when the user is presented the add dialog, if the user clicks the clear filter button then all books don't appear in background (Only in case of filter applied and no books exist of that author).

This time I won't provide the solutions to fix the bug, that you need to do on your own.

Enhancement required : Provide a combo box to the user. It should enable the user to change page size.

Our own framework for data storage



Download TMORMFramework.zip from tm-certificates.com

Unzip it, it contains a folder named as TMORMFramework with src\main\java folder containing all the sourrce code

in TMORMFramework folder is the build.gradle file
in TMORMFramework\lib folder are all the required dependeencies

Install gradle, set GRADLE_HOME environment variable pointing to gradle folder

Now while staying in the TMORMFramework folder, type

```
gradle    build
```

The code will be compiled and build\libs folder should contain the TMORMFramework.jar file.

Now in TMORMFramework folder create a folder named as testcases, it testcases folder create the following testcases.
Create the entity classes as follows in the testcases folder

```
import com.thinking.machines.dmframework.annotations.*;
@Display(value="Author")
@Table(name="author")
public class Author implements java.io.Serializable,Comparable<Author>
{
@Display(value="code")
@Column(name="code")
private Integer code;
@Sort(priority=1)
@Display(value="name")
@Column(name="name")
private String name;
public Author()
{
this.code=null;
this.name=null;
}
public void setCode(Integer code)
{
this.code=code;
}
public Integer getCode()
{
return this.code;
}
public void setName(String name)
{
```

```
this.name=name;
}
public String getName()
{
return this.name;
}
public boolean equals(Object object)
{
if(object==null) return false;
if(!(object instanceof Author)) return false;
Author anotherAuthor=(Author)object;
if(this.code==null && anotherAuthor.code==null) return true;
if(this.code==null || anotherAuthor.code==null) return false;
return this.code.equals(anotherAuthor.code);
}
public int compareTo(Author anotherAuthor)
{
if(anotherAuthor==null) return 1;
if(this.name==null && anotherAuthor.name==null) return 0;
int difference;
if(this.name==null && anotherAuthor.name!=null) return 1;
if(this.name!=null && anotherAuthor.name==null) return -1;
difference=this.name.compareTo(anotherAuthor.name);
return difference;
}
public int hashCode()
{
if(this.code==null) return 0;
return this.code.hashCode();
}
}
```

```
import com.thinking.machines.dmframework.annotations.*;
@Display(value="Book")
@Table(name="book")
public class Book implements java.io.Serializable,Comparable<Book>
{
@Display(value="code")
@Column(name="code")
private Integer code;
@Sort(priority=1)
@Display(value="title")
@Column(name="title")
private String title;
@Display(value="author code")
@Column(name="author_code")
private Integer authorCode;
```

```
@Display(value="category")
@Column(name="category")
private String category;
@Display(value="price")
@Column(name="price")
private Integer price;
public Book()
{
this.code=null;
this.title=null;
this.authorCode=null;
this.category=null;
this.price=null;
}
public void setCode(Integer code)
{
this.code=code;
}
public Integer getCode()
{
return this.code;
}
public void setTitle(String title)
{
this.title=title;
}
public String getTitle()
{
return this.title;
}
public void setAuthorCode(Integer authorCode)
{
this.authorCode=authorCode;
}
public Integer getAuthorCode()
{
return this.authorCode;
}
public void setCategory(String category)
{
this.category=category;
}
public String getCategory()
{
return this.category;
}
public void setPrice(Integer price)
```

```
{
this.price=price;
}
public Integer getPrice()
{
return this.price;
}
public boolean equals(Object object)
{
if(object==null) return false;
if(!(object instanceof Book)) return false;
Book anotherBook=(Book)object;
if(this.code==null && anotherBook.code==null) return true;
if(this.code==null || anotherBook.code==null) return false;
return this.code.equals(anotherBook.code);
}
public int compareTo(Book anotherBook)
{
if(anotherBook==null) return 1;
if(this.title==null && anotherBook.title==null) return 0;
int difference;
if(this.title==null && anotherBook.title!=null) return 1;
if(this.title!=null && anotherBook.title==null) return -1;
difference=this.title.compareTo(anotherBook.title);
return difference;
}
public int hashCode()
{
if(this.code==null) return 0;
return this.code.hashCode();
}
}
```

For compiling incude build\libs and lib folder in classpath

While staying in testcases folder

```
javac -classpath  ..\build\libs\*;..\lib\*;.   *.java
```

To run the testcases do the same

```
import com.thinking.machines.dmframework.*;
import com.thinking.machines.dmframework.exceptions.*;
class AuthorInsertTestCase
{
public static void main(String data[])
{
```

```
String name=data[0];
Author author=new Author();
author.setName(name);
DataManager dm=new DataManager();
try
{
dm.begin();
dm.insert(author);
dm.end();
System.out.println(name+" inserted and was alloted code : "+author.getCode());
}catch(DMFrameworkException dmFrameworkException)
{
System.out.println(dmFrameworkException.getMessage());
}
catch(ValidatorException validatorException)
{
ExceptionsIterator exceptionsIterator=validatorException.getIterator();
ExceptionIterator exceptionIterator;
while(exceptionsIterator.hasNext())
{
exceptionIterator=exceptionsIterator.next();
while(exceptionIterator.hasNext())
{
exceptionIterator.next();
System.out.println(exceptionIterator.index()+","+exceptionIterator.property()
+","+exceptionIterator.exception());
}
}
}

}
}
```

```
import com.thinking.machines.dmframework.*;
import com.thinking.machines.dmframework.exceptions.*;
class AuthorUpdateTestCase
{
public static void main(String data[])
{
Integer code=Integer.parseInt(data[0]);
String name=data[1];
Author author=new Author();
author.setCode(code);
author.setName(name);
DataManager dm=new DataManager();
try
{
```

```
dm.begin();
dm.update(author);
dm.end();
System.out.println("Author updated");
}catch(DMFrameworkException dmFrameworkException)
{
System.out.println(dmFrameworkException.getMessage());
}
catch(ValidatorException validatorException)
{
ExceptionsIterator exceptionsIterator=validatorException.getIterator();
ExceptionIterator exceptionIterator;
while(exceptionsIterator.hasNext())
{
exceptionIterator=exceptionsIterator.next();
while(exceptionIterator.hasNext())
{
exceptionIterator.next();
System.out.println(exceptionIterator.index()+","+exceptionIterator.property()
+","+exceptionIterator.exception());
}
}
}

}
}
```

```
import com.thinking.machines.dmframework.*;
import com.thinking.machines.dmframework.exceptions.*;
class AuthorDeleteTestCase
{
public static void main(String data[])
{
Integer code=Integer.parseInt(data[0]);
DataManager dm=new DataManager();
try
{
dm.begin();
dm.delete(Author.class,code);
dm.end();
System.out.println("Author Deleted");
}catch(DMFrameworkException dmFrameworkException)
{
System.out.println(dmFrameworkException.getMessage());
}
catch(ValidatorException validatorException)
{
```

```
ExceptionsIterator exceptionsIterator=validatorException.getIterator();
ExceptionIterator exceptionIterator;
while(exceptionsIterator.hasNext())
{
exceptionIterator=exceptionsIterator.next();
while(exceptionIterator.hasNext())
{
exceptionIterator.next();
System.out.println(exceptionIterator.index()+","+exceptionIterator.property()
+","+exceptionIterator.exception());
}
}
}

}
}
```

```
import java.util.*;
import com.thinking.machines.dmframework.*;
import com.thinking.machines.dmframework.exceptions.*;
class AuthorSelectTestCase
{
public static void main(String data[])
{
DataManager dm=new DataManager();
List<Author> authors;
System.out.println("All authors test case");
try
{
dm.begin();
authors=dm.select(Author.class).query();
for(Author author:authors)
{
System.out.println(author.getCode()+","+author.getName());
}
dm.end();
System.console().readLine("Press any key...");
System.out.println("Total number of authors");
dm.begin();
int count=(int)dm.select(Author.class).count("code").query().intValue();
dm.end();
System.out.println("Total number of authors : "+count);
System.console().readLine("Press any key...");
System.out.println("Max code");
dm.begin();
int code=(int)dm.select(Author.class).max("code").query().intValue();
dm.end();
System.out.println("Largest alloted code : "+code);
System.console().readLine("Press any key...");
System.out.println("Min code");
dm.begin();
code=(int)dm.select(Author.class).min("code").query().intValue();
dm.end();
System.out.println("Smallest alloted code : "+code);
System.console().readLine("Press any key...");
System.out.println("Like clause");
dm.begin();
authors=dm.select(Author.class).where("name").like("A%").query();
for(Author author:authors)
{
System.out.println(author.getCode()+","+author.getName());
}
dm.end();
System.console().readLine("Press any key...");
}catch(DMFrameworkException dmFrameworkException)
{
System.out.println(dmFrameworkException.getMessage());
}
}
}
```

Before running the testcases, in testcases folder create the following files

log4j.properties

```
log4j.rootLogger=error,stdout,file

# Redirect log messages to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m
%n

# Redirect log messages to a log file, support file rolling.
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=log4j-application.log
log4j.appender.file.MaxFileSize=5MB
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m
%n
```

TMDMFramework.xml

```
<database architecture="MySQL" connectionString="jdbc:mysql://localhost:3306/mydb2017"
username="mydb2017" password="mydb2017" driver="com.mysql.jdbc.Driver" />
<entity class="Author" />
```

```
<entity class="Book" />
```

Assignment :

Design testcases for Book module.
Design a view in database, book_view aginst which provide an SQL Statement to join Book and Author table.
Now add the entry for View in the configuration file TMDMFramework.xml, then create BookView class with annotations to map to book_view

Then write BookViewSelectTestCase

Web Services Frameworks

The over hyped server side technology.

List of some available frameworks

Apache Axis, Apache Axis2, Apache CXF, Skytells,
Codeigniter, gSOAP, Jello, Jersey, Yii, Zend, Smart, WSO2
WSF, WSI, WCF,
Glassfish WSDP, ...

We won't be **bogged down** by the **big names**, instead we will be

creating our own web services framework

Create a folder named as tmwstest1, in it create **WEB-INF\classes, WEB-INF\lib** and **WEB-INF\classes\com\thinking\machines** folders

Download **tmws.zip** from http://tm-certificates.com
unzip it and copy the **jar files** in **WEB-INF\lib** and copy the **tmws** folder from **src** folder that contains the java files to the **WEB-INF\classes\com\thinking\machines** folder

compile the java files in the tmws folder that we copied in the classes\com\thinking\machines folder (do the same in annotations and plugin folder).

add the following servlet-mappings to web.xml (to be created in WEB-INF folder)

```
  <servlet>
  <servlet-name>TMWebService</servlet-name>
  <servlet-class>com.thinking.machines.tmws.TMWebService</servlet-class>
  <load-on-startup>1</load-on-startup>
  <init-param>
  <param-name>configuration</param-name>
  <param-value>/WEB-INF/ourconf/kuchbhi.xml</param-value>
  </init-param>
```

```
<init-param>
<param-name>development-mode</param-name>
<param-value>true</param-value>
</init-param>
<init-param>
<param-name>uploadPath</param-name>
<param-value>WEB-INF/whatever</param-value>
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>TMWebService</servlet-name>
<url-pattern>/wswsws/*</url-pattern>
</servlet-mapping>
```

in WEB-INF folder create two folders named as whatever and ourconf

in ourcconf folder create the following kuchbhi.xml file

kuchbhi.xml

```
<?xml version='1.0' encoding='utf-8' ?>
<packages>
<package>com.cool.one</package>
<package>com.cool.two</package>
<package>com.cool.three</package>
</packages>
```

in WEB-INF\classes folder create the following structure com\cool\one in com\cool\one folder create the following class

```
package com.cool.one;
import com.thinking.machines.tmws.annotations.*;
import java.util.*;
@Path("/sch")
public class SchoolWhatever
{
@Path("count")
public int getStudentCount()
{
return 500;
}
@Path("add")
public void addStudent(int rollNumber,String name) throws
Exception
{
if(rollNumber<1001 || rollNumber>5001)
{
throw new Exception("Invalid roll number");
}
}
@Path("update")
public Object updateStudent(int rollNumber,String name)
throws Exception
{
if(rollNumber<1001 || rollNumber>5001)
{
return new Exception("Invalid roll number");
}
return "Student updated";
}
@Path("list")
public ArrayList<String> getListOfStudents(String cityName)
{
if(cityName.equals("Pune")) throw new
RuntimeException("Invalid city");
ArrayList<String> k=new ArrayList<String>();
k.add("Sameer");
k.add("Lokesh");
k.add("Suresh");
return k;
}
}
```

// you can create more classes and the class should not be annotated with

@Path("/sch")

use something else in place of sch

somewhere else for eg. c:\del or wherever you want, create a folder structure com\cool\two, in it create the following classes

```
package com.cool.two;
public class Employee implements java.io.Serializable
{
private int code;
private String name;
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
return this.code;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
}
```

```
package com.cool.two;
import java.util.*;
import com.thinking.machines.tmws.annotations.*;
@Path("/emp")
public class EmployeeManager
{
@Path("aaa")
public Object add(Employee e)
{
if(e.getName().trim().length()==0) return new
Exception("Invalid employee name : "+e.getName());
e.setCode(101);
return e;
}
@Path("bbb")
public List<Employee> getList()
{
List<Employee> l=new LinkedList<Employee>();
Employee e;
e=new Employee();
e.setCode(101);
e.setName("Rakesh");
l.add(e);
e=new Employee();
e.setCode(102);
e.setName("Rahul");
l.add(e);
e=new Employee();
e.setCode(103);
e.setName("Sameer");
l.add(e);
return l;
}
}
```

stay in the folder that contains the com folder. In it create a folder named as classes and then ask the compiled to compile the java files in com\cool\two folder and ask him to create the compiled code in classes folder using the -d option.

Now create a folder named as dist and in dist create a jar file named as ujjain.jar (it should contain the com folder from the classes folder)

Now copy the ujjain.jar file to tmwstest1\WEB-INF\lib folder

On your own create a jar named as indore.jar containing com.cool.three package with some classes containing methods annotated properly.

Install google chrome canary.

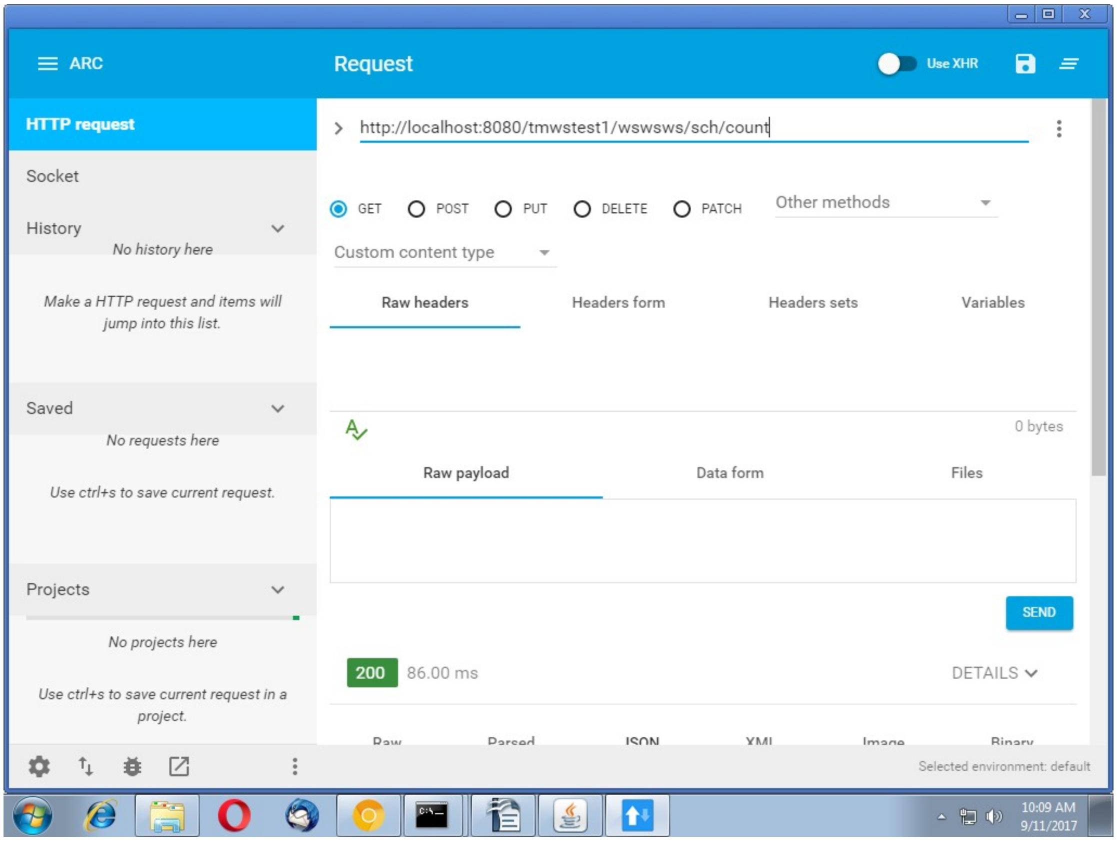
Install Adavanced REST client for chrome

Start server

Start google chrome canary (or whatever browser you are using with REST client plugin installed)

launch the REST client plugin app. I am place the screen shots according to ARC

In the request section, type the following URL and click the send button as follows
http://localhost:8080/tmwstest1/wswsws/sch/count



http://localhost:8080/tmwstest1/wswsws/sch/count

● GET ○ POST ○ PUT ○ DELETE ○ PATCH

200 86.00 ms

You should see the following response in the response section.

```
{
    "success": true,
    "isReturningSomething": true,
    "isResultNull": false,
    "result": 500
}
```

Now change the url to
http://localhost:8080/tmwstest1/wswsws/sch/add

In the raw payload (the title might vary according to the plugin that you are using) type the following JSON

```
{
 "rollNumber":1002,
 "name": "Sameer"
}
```

and click the send button, this time the response should be as follows

```
{
"success": false,
"isException": false,
"isObject": false,
"error": "Expected : 2 arguments, found : 0"
}
```

Now change the JSON in raw payload section as follows

```
{
 "argument-1":1002,
 "argument-2": "Sameer"
}
```

and select the POST option

and click the send button, the response should be as follows

```
{
"success": true,
"isReturningSomething": false
}
```

Now change the URL to

http://localhost:8080/tmwstest1/wswsws/emp/aaa

Keep the post option selected and type the following JSON in raw payload section

```
{
 "argument-1": {
  "name": ""
 }
}
```

and click the send button and following should be the output

```
{
"success": false,
"isException": true,
"isObject": true,
"exception":
{
   "cause": null,
   "stackTrace": [

{
         "methodName": "add",
         "fileName": "EmployeeManager.java",
         "lineNumber": 10,
         "className": "com.cool.two.EmployeeManager",
         "nativeMethod": false
}
,

{
         "methodName": "invoke0",
         "fileName": "NativeMethodAccessorImpl.java",
         "lineNumber": -2,
         "className": "sun.reflect.NativeMethodAccessorImpl",
         "nativeMethod": true
}
,

{
         "methodName": "invoke",
         "fileName": "NativeMethodAccessorImpl.java",
         "lineNumber": 62,
         "className": "sun.reflect.NativeMethodAccessorImpl",
         "nativeMethod": false
}
,

{
         "methodName": "invoke",
         "fileName": "DelegatingMethodAccessorImpl.java",
         "lineNumber": 43,
         "className": "sun.reflect.DelegatingMethodAccessorImpl",
         "nativeMethod": false
}
,

{
        "methodName": "invoke",
        "fileName": "Method.java",
        "lineNumber": 483,
        "className": "java.lang.reflect.Method",
        "nativeMethod": false
}
,

{
        "methodName": "process",
        "fileName": "RequestProcessor.java",
        "lineNumber": 248,
        "className": "com.thinking.machines.tmws.RequestProcessor",
        "nativeMethod": false
}
,

{
        "methodName": "processRequest",
        "fileName": "TMWebService.java",
        "lineNumber": 123,
        "className": "com.thinking.machines.tmws.TMWebService",
        "nativeMethod": false
}
,

{
        "methodName": "doPost",
        "fileName": "TMWebService.java",
        "lineNumber": 85,
        "className": "com.thinking.machines.tmws.TMWebService",
        "nativeMethod": false
}
,

{
        "methodName": "service",
        "fileName": "HttpServlet.java",
        "lineNumber": 661,
        "className": "javax.servlet.http.HttpServlet",
        "nativeMethod": false
}
,
{
        "methodName": "service",
        "fileName": "HttpServlet.java",
        "lineNumber": 742,
        "className": "javax.servlet.http.HttpServlet",
        "nativeMethod": false
}
,

{
        "methodName": "internalDoFilter",
        "fileName": "ApplicationFilterChain.java",
        "lineNumber": 231,
        "className": "org.apache.catalina.core.ApplicationFilterChain",
        "nativeMethod": false
}
,

{
        "methodName": "doFilter",
        "fileName": "ApplicationFilterChain.java",
        "lineNumber": 166,
        "className": "org.apache.catalina.core.ApplicationFilterChain",
        "nativeMethod": false
}
,

{
        "methodName": "doFilter",
        "fileName": "WsFilter.java",
        "lineNumber": 52,
        "className": "org.apache.tomcat.websocket.server.WsFilter",
        "nativeMethod": false
}
,

{
        "methodName": "internalDoFilter",
        "fileName": "ApplicationFilterChain.java",
        "lineNumber": 193,
        "className": "org.apache.catalina.core.ApplicationFilterChain",
        "nativeMethod": false
```

```
}
,

{
         "methodName": "doFilter",
         "fileName": "ApplicationFilterChain.java",
         "lineNumber": 166,
         "className": "org.apache.catalina.core.ApplicationFilterChain",
         "nativeMethod": false
}
,

{
         "methodName": "invoke",
         "fileName": "StandardWrapperValve.java",
         "lineNumber": 198,
         "className": "org.apache.catalina.core.StandardWrapperValve",
         "nativeMethod": false
}
,

{
         "methodName": "invoke",
         "fileName": "StandardContextValve.java",
         "lineNumber": 96,
         "className": "org.apache.catalina.core.StandardContextValve",
         "nativeMethod": false
}
,

{
         "methodName": "invoke",
         "fileName": "StandardHostValve.java",
         "lineNumber": 140,
         "className": "org.apache.catalina.core.StandardHostValve",
         "nativeMethod": false
}
,

{
         "methodName": "invoke",
         "fileName": "ErrorReportValve.java",
         "lineNumber": 80,
        "className": "org.apache.catalina.valves.ErrorReportValve",
        "nativeMethod": false
}
,

{
        "methodName": "invoke",
        "fileName": "AbstractAccessLogValve.java",
        "lineNumber": 624,
        "className": "org.apache.catalina.valves.AbstractAccessLogValve",
        "nativeMethod": false
}
,

{
        "methodName": "invoke",
        "fileName": "StandardEngineValve.java",
        "lineNumber": 87,
        "className": "org.apache.catalina.core.StandardEngineValve",
        "nativeMethod": false
}
,

{
        "methodName": "service",
        "fileName": "CoyoteAdapter.java",
        "lineNumber": 342,
        "className": "org.apache.catalina.connector.CoyoteAdapter",
        "nativeMethod": false
}
,

{
        "methodName": "service",
        "fileName": "Http11Processor.java",
        "lineNumber": 799,
        "className": "org.apache.coyote.http11.Http11Processor",
        "nativeMethod": false
}
,

{
        "methodName": "process",
        "fileName": "AbstractProcessorLight.java",
        "lineNumber": 66,
        "className": "org.apache.coyote.AbstractProcessorLight",
        "nativeMethod": false
}
,

{
        "methodName": "process",
        "fileName": "AbstractProtocol.java",
        "lineNumber": 861,
        "className": "org.apache.coyote.AbstractProtocol$ConnectionHandler",
        "nativeMethod": false
}
,

{
        "methodName": "doRun",
        "fileName": "NioEndpoint.java",
        "lineNumber": 1455,
        "className": "org.apache.tomcat.util.net.NioEndpoint$SocketProcessor",
        "nativeMethod": false
}
,

{
        "methodName": "run",
        "fileName": "SocketProcessorBase.java",
        "lineNumber": 49,
        "className": "org.apache.tomcat.util.net.SocketProcessorBase",
        "nativeMethod": false
}
,

{
        "methodName": "runWorker",
        "fileName": "ThreadPoolExecutor.java",
        "lineNumber": 1142,
        "className": "java.util.concurrent.ThreadPoolExecutor",
        "nativeMethod": false
}
,
{
         "methodName": "run",
         "fileName": "ThreadPoolExecutor.java",
         "lineNumber": 617,
         "className": "java.util.concurrent.ThreadPoolExecutor$Worker",
         "nativeMethod": false
}
,

{
         "methodName": "run",
         "fileName": "TaskThread.java",
         "lineNumber": 61,
         "className": "org.apache.tomcat.util.threads.TaskThread$WrappingRunnable",
         "nativeMethod": false
}
,

{
         "methodName": "run",
         "fileName": "Thread.java",
         "lineNumber": 745,
         "className": "java.lang.Thread",
         "nativeMethod": false
}
],
   "localizedMessage": "Invalid employee name : ",
   "message": "Invalid employee name : ",
   "suppressed": [],
}
}
```

Now change the JSON to the following and again click the send button

```
{
 "argument-1": {
  "name": "Sameer"
 }
}
```

and this time the output should be as follows

```
{
"success": true,
"isReturningSomething": true,
"isResultNull": false,
"result":
{
   "code": 101,
   "name": "Sameer"
}
}
```

Now change the url to

http://localhost:8080/tmwstest1/wswsws/emp/ccc

change the request type to GET

http://localhost:8080/tmwstest1/wswsws/emp/bbb

and click the send button and you should see the following output

```
{
"success": true,
"isReturningSomething": true,
"isResultNull": false,
"result": [

{
      "code": 101,
      "name": "Rakesh"
}
,

{
      "code": 102,
        "name": "Rahul"
}
,

{
        "code": 103,
        "name": "Sameer"
}
],
}
```

Now make the necessary changes in the URL and test it your self.

Now close the plugin and in the address bar of the browser type the following URL

http://localhost:8080/tmwstest1/wswsws/webservice-report

and you should see a PDF (either the browser will open it or download it). That PDF should contain information about the deployed services.