Manorath Bajaj

001462954

Web Tools Report.

Porject Summary:

A Web Application that lets a User search for Movies and look at reviews, Post Reviews and Follow Critic Reviews online. The aim of this project is to develop an ‘Movie Database and Rating Website’ using Spring MVC and Hibernate. The motivation behind this project is to make a website where People can browse through movies and look at it’s reviews/

The Website will have three different type of users:

• System admin

• Users

• Critics

Key Functionalities:

• System Admins will be able to add movies to the database

• Users will be able to search for movies in the database

• Users will be able to comment and rate on movies

• Users will be able to follow Critics reviews

• Critics will also be able to rate movies(2 different rating systems)

• Users will be shown the average ratings and comments on the movie database

Technologies Used:

• Spring MVC

• Hibernate

• Maven

Roles:

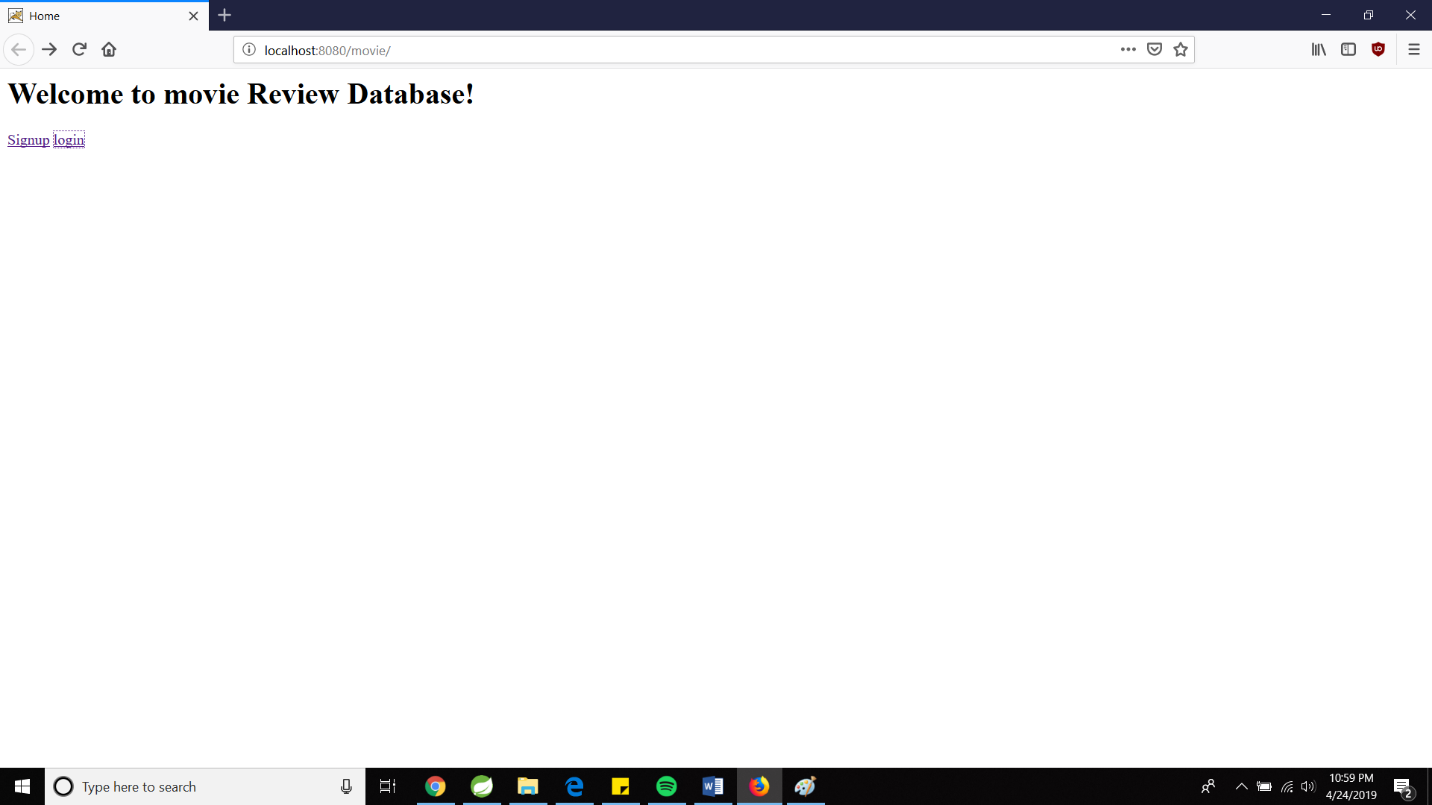
Admins: Admins will be able to add movies to the website, delete users from the website or change their type.

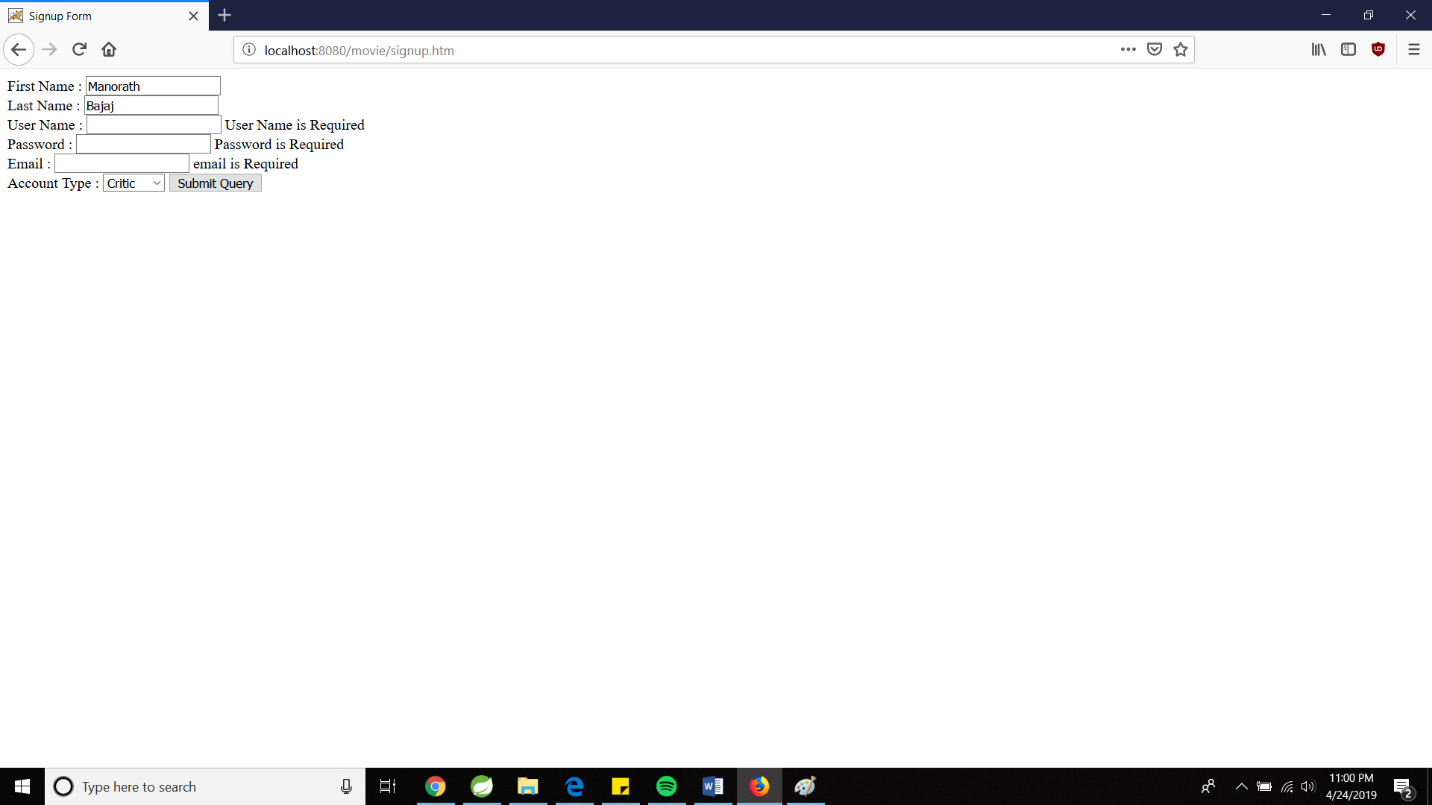
Critics: Critics will be able to post Reviews on movies. They can also do whatever normal users can do, their Reviews also be viewable on a separate page.

Users: Users can search form Movies, Check the Average rating and post their own Review.

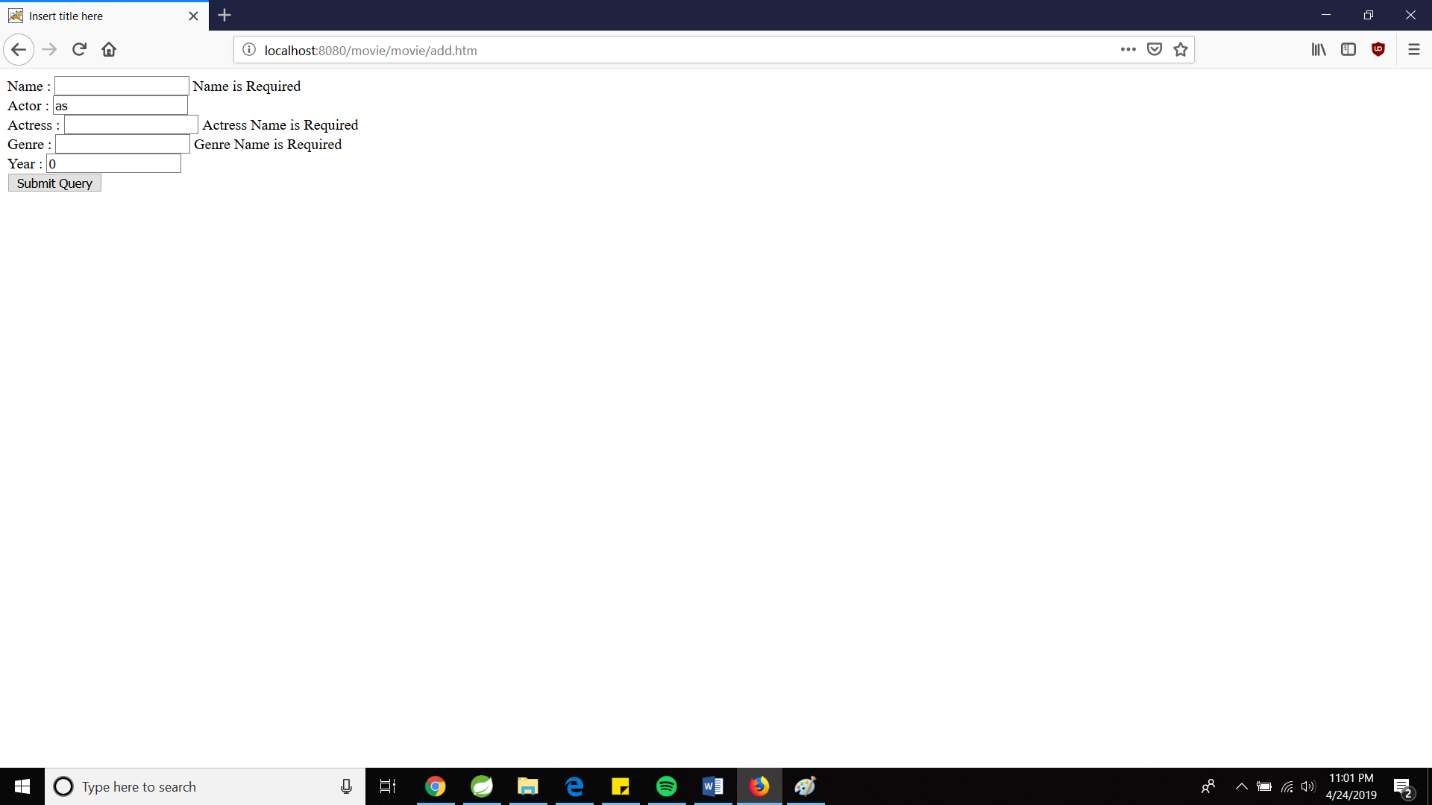
ScreenShots:

homepage

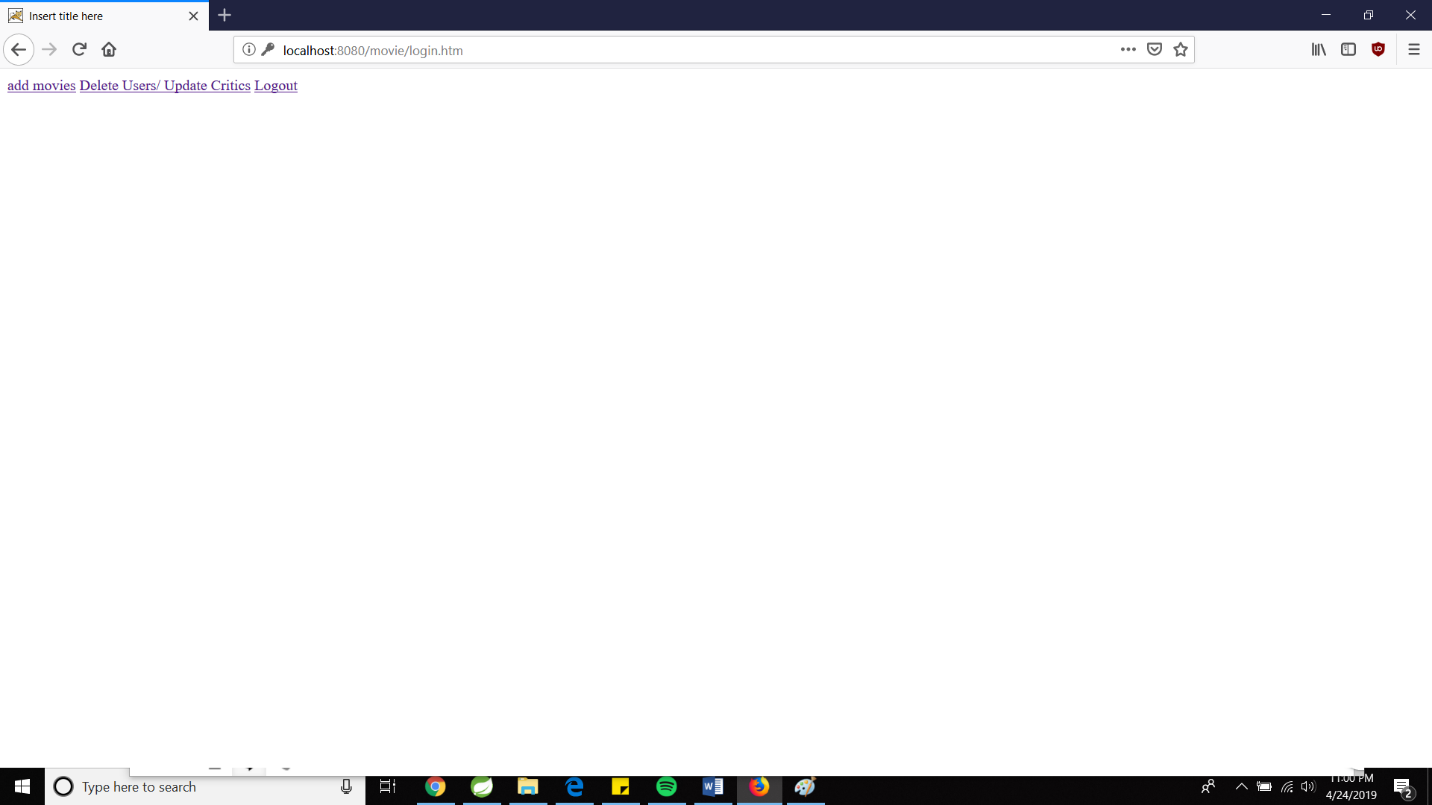


Signup page

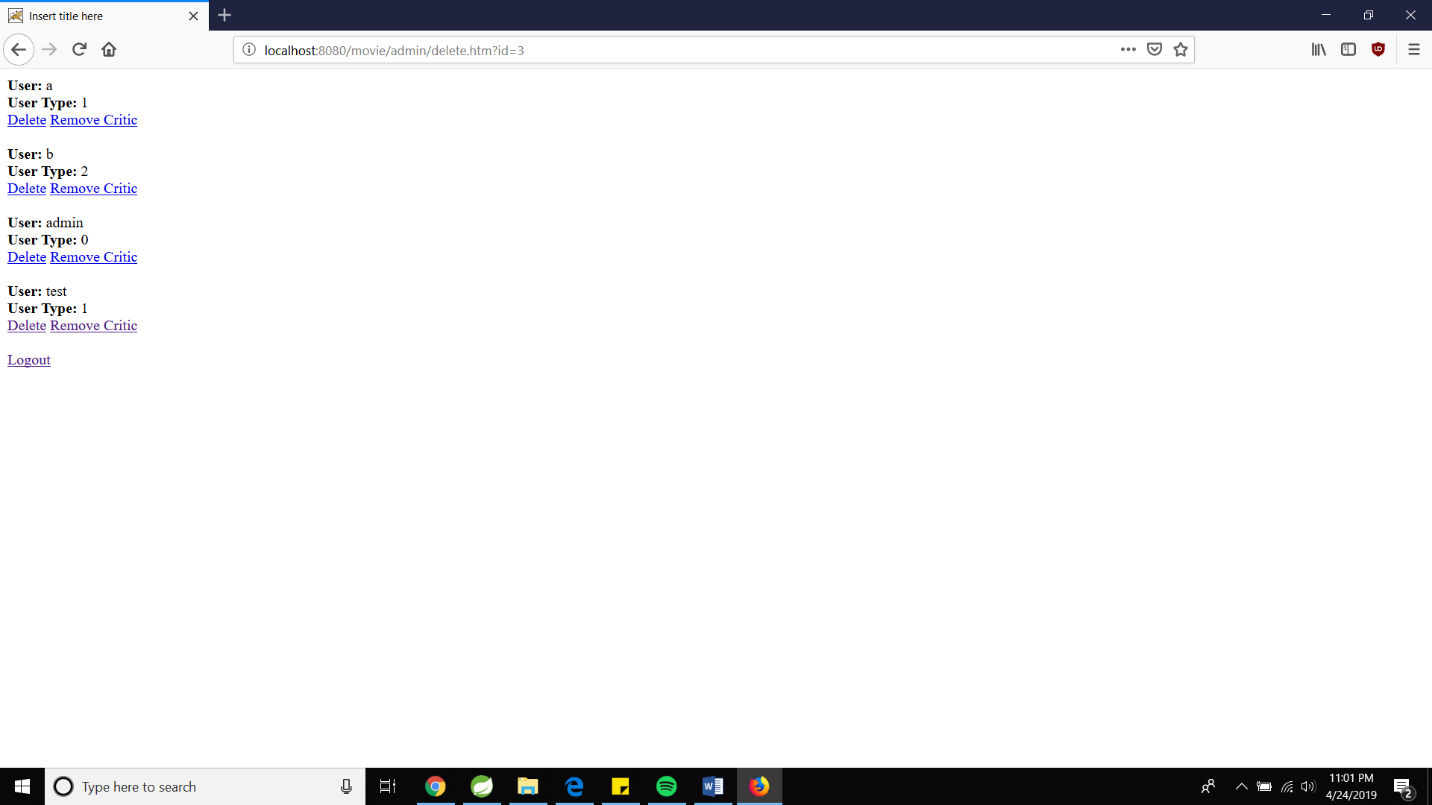
Add Movie page:



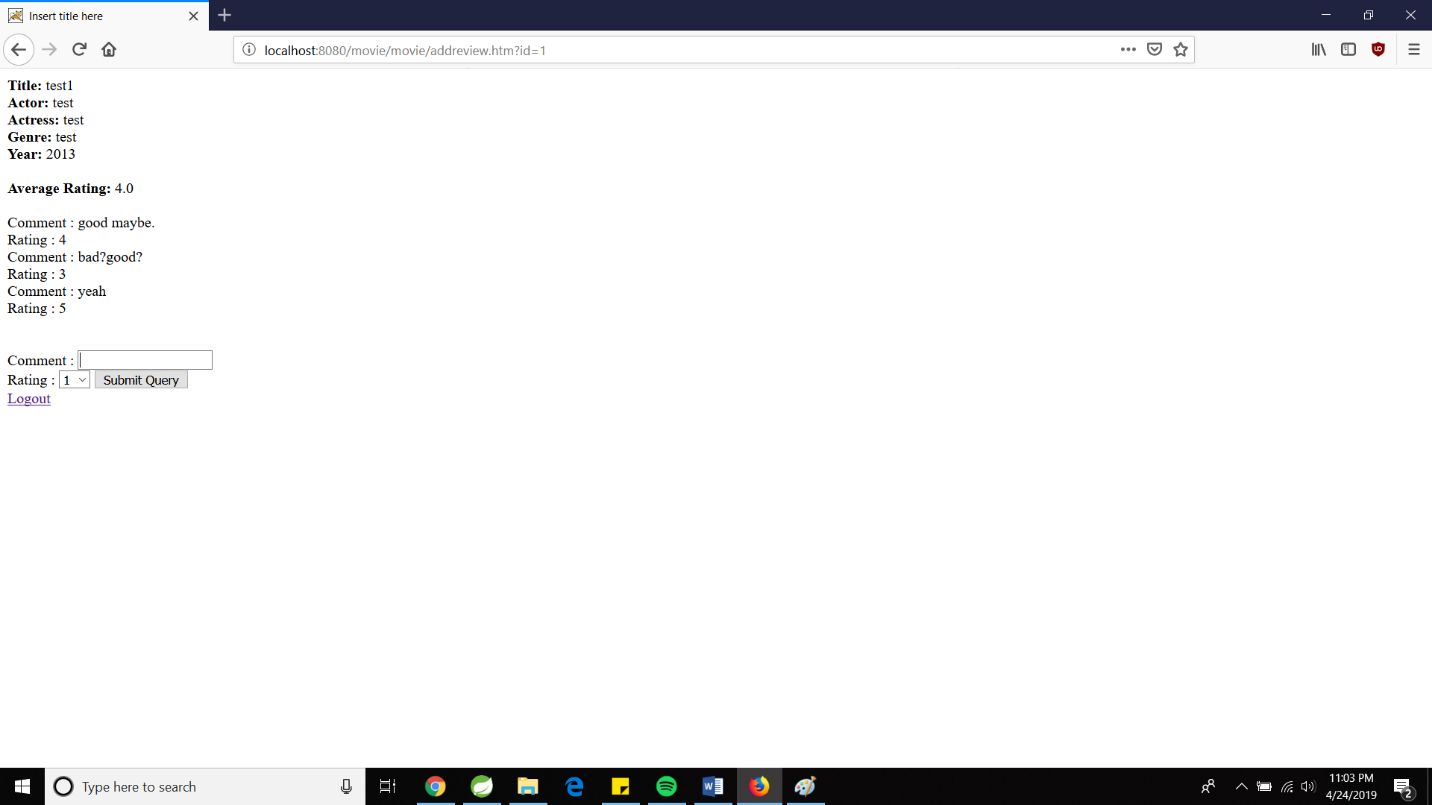
Admin Home Page:



Delete User Page



Movie Review Page



Appendix:

Admin Controller:

**package** com.me.movie;

**import** java.util.List;

**import** javax.servlet.http.HttpServletRequest;

**import** org.apache.taglibs.standard.lang.jstl.test.beans.PublicBean1;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** com.me.dao.UserDao;

**import** com.me.exception.UserException;

**import** com.me.pojo.User;

@Controller

**public** **class** AdminController {

@Autowired

UserDao userDao;

@RequestMapping(value = "/admin/delete.htm", method = RequestMethod.***GET***)

**public** String showUsers(Model model) **throws** UserException {

List<User> u = userDao.getAllUsers();

model.addAttribute("u", u);

**return** "delete-user";

}

@RequestMapping(value = "/admin/deleteUser.htm", method = RequestMethod.***GET***)

**public** String deleteUser(Model model,HttpServletRequest request) **throws** UserException

{ **try** {

**long** id = Long.*valueOf*(request.getParameter("id"));

System.***out***.println("user id " + id);

userDao.delete(userDao.getUserById(id));

**return** showUsers(model);

}**catch**(Exception e) {

e.printStackTrace();

}

**return** "home";

}

@RequestMapping(value = "/admin/remove.htm", method = RequestMethod.***GET***)

**public** String updateUser(Model model,HttpServletRequest request) **throws** UserException

{

**long** id = Long.*valueOf*(request.getParameter("id"));

userDao.changeUserType(id);

**return** showUsers(model);

}

}

Home Controller:

package com.me.movie;

import java.text.DateFormat;

import java.util.Date;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

/\*\*

\* Handles requests for the application home page.

\*/

@Controller

public class HomeController {

private static final Logger logger = LoggerFactory.getLogger(HomeController.class);

/\*\*

\* Simply selects the home view to render by returning its name.

\*/

@RequestMapping(value = "/", method = RequestMethod.GET)

public String home(Locale locale, Model model) {

logger.info("Welcome home! The client locale is {}.", locale);

Date date = new Date();

DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG, locale);

String formattedDate = dateFormat.format(date);

model.addAttribute("serverTime", formattedDate );

return "home";

}

@RequestMapping(value = "/logout.htm", method = RequestMethod.GET)

public String logout(HttpServletRequest request) {

HttpSession session = request.getSession();

session.invalidate();

return "home";

}

}

Login Controller

package com.me.movie;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.validation.annotation.Validated;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import com.me.dao.UserDao;

import com.me.exception.UserException;

import com.me.pojo.User;

import com.me.validator.UserValidator;

@Controller

@RequestMapping(value = "/login.htm")

public class LoginController {

@Autowired

UserValidator uservalidator;

@Autowired

UserDao userDao;

@Autowired

User user;

@InitBinder

protected void initBinder(WebDataBinder binder) {

binder.setValidator(uservalidator);

}

@RequestMapping(method = RequestMethod.GET)

public String home(Locale locale, Model model, User user,HttpServletRequest request, HttpServletResponse response) {

HttpSession session = request.getSession();

session.invalidate();

model.addAttribute("user", user);

return "login-form";

}

@RequestMapping(method = RequestMethod.POST)

public String successView(HttpServletRequest request) throws UserException {

HttpSession session = request.getSession();

User u = userDao.getUser((String)request.getParameter("uName"),(String)request.getParameter("password"));

if(u == null) {

System.out.print("null-back");

return "login-form";

}

session.setAttribute("user", u);

if(u.getuType().equals("0")) {

return "admin-home";

}

else {

return "welcome";

}

}

}

Movie Controller

package com.me.movie;

import java.util.List;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.hibernate.HibernateException;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.validation.annotation.Validated;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.me.dao.MovieDao;

import com.me.exception.MovieException;

import com.me.pojo.Movie;

import com.me.pojo.User;

import com.me.validator.MovieValidator;

@Controller

public class MovieController {

@Autowired

MovieValidator movievalidator;

@Autowired

MovieDao movieDao;

@InitBinder

protected void initBinder(WebDataBinder binder) {

binder.setValidator(movievalidator);

}

@RequestMapping(value = "/movie/add.htm", method = RequestMethod.GET)

public String addForm(Model model, Movie movie, HttpServletRequest request) {

HttpSession session = request.getSession();

User u = (User)session.getAttribute("user");

if(u.getuType().equals("0")) {

model.addAttribute("movie",movie);

return "addmovie-form";

}

else {

return "home";

}

}

@RequestMapping(value = "/movie/add.htm", method = RequestMethod.POST)

public String readForm(@Validated@ModelAttribute("movie") Movie movie,BindingResult bindingResult,ModelMap model) {

if (bindingResult.hasErrors()) {

return "addmovie-form";

}

movieDao.addMovie(movie);

return "admin-home";

}

@RequestMapping(value = "/movie/search.htm",method = RequestMethod.GET)

public String searchMovie() {

return "search-page";

}

@RequestMapping(value = "/movie/search.htm",method = RequestMethod.POST)

public ModelAndView resultPage(HttpServletRequest request) throws MovieException {

String criteria = request.getParameter("criteria");

ModelAndView mv = null;

try {

String keyword = request.getParameter("keyword");

List<Movie> movieList = movieDao.getMovies(criteria, keyword);

mv=new ModelAndView("movie-search-result","movieList",movieList);

}catch(HibernateException e) {

e.printStackTrace();

}

return mv;

}

// @RequestMapping(value = "/movie/movie/addReview.htm")

// public String addReviewFrom( ) {

//

// }

}

Review Controller

package com.me.movie;

import java.util.List;

import java.util.Set;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.me.dao.MovieDao;

import com.me.dao.ReviewDao;

import com.me.dao.UserDao;

import com.me.exception.MovieException;

import com.me.exception.ReviewException;

import com.me.exception.UserException;

import com.me.pojo.Movie;

import com.me.pojo.Review;

import com.me.pojo.User;

@Controller

public class ReviewController {

@Autowired

MovieDao movieDao;

@Autowired

UserDao userDao;

@Autowired

ReviewDao reviewDao;

@Autowired

Review review;

@RequestMapping(value = "/movie/addreview.htm",method = RequestMethod.GET)

public String addResult(HttpServletRequest request,Model model,Review review) throws MovieException, ReviewException {

HttpSession session = request.getSession();

if(session.getAttribute("user") == null)

{

return "home";

}

long id = Integer.valueOf((String)request.getParameter("id"));

Movie m = movieDao.getMovieById(id);

// model.addAttribute("movie", m);

Set<Review> r = m.getReviews();

int total = 0;

int stars = 0;

float avgRating;

for(Review re : r ) {

stars = stars + re.getStar();

total++;

}

avgRating = stars/total;

model.addAttribute("avgRating", avgRating);

model.addAttribute("movie",m);

return "review-page";

}

@RequestMapping(value = "/movie/addreview.htm",method = RequestMethod.POST)

public String showPage(HttpServletRequest request,Review review) throws MovieException, ReviewException {

HttpSession session = request.getSession();

long id = Integer.valueOf((String)request.getParameter("id"));

Movie m = movieDao.getMovieById(id);

User u = (User)session.getAttribute("user");

review.setMovie(m);

review.setPostedBy(u);

review.setStar(Integer.valueOf(request.getParameter("star")));

review.setComment(request.getParameter("comment"));

review = reviewDao.create(review);

return "welcome";

}

@RequestMapping(value = "/movie/critics.htm",method = RequestMethod.GET)

public String showCritics(HttpServletRequest request, Model model) throws UserException {

List<User> u = userDao.getCritics();

model.addAttribute("critics", u);

return "critic";

}

}

Signup Controller:

package com.me.movie;

import java.text.DateFormat;

import java.util.Date;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

//import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.validation.annotation.Validated;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import com.me.dao.UserDao;

import com.me.pojo.User;

import com.me.validator.UserValidator;

@Controller

@RequestMapping(value = "/signup.htm")

public class SignupController {

@Autowired

UserValidator uservalidator;

@Autowired

UserDao userDao;

@InitBinder

protected void initBinder(WebDataBinder binder) {

binder.setValidator(uservalidator);

}

@RequestMapping(method = RequestMethod.GET)

public String home(Locale locale, Model model, User user, HttpServletRequest request) {

HttpSession session = request.getSession();

session.invalidate();

Date date = new Date();

DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG, locale);

String formattedDate = dateFormat.format(date);

model.addAttribute("serverTime", formattedDate );

model.addAttribute("user", user);

return "signup-form";

}

@RequestMapping(method = RequestMethod.POST)

public String successView(@Validated@ModelAttribute("user") User user,BindingResult bindingResult,ModelMap model) {

if (bindingResult.hasErrors()) {

return "signup-form"; //the are validation errors, go to the form view

}

//no errors, so go to the success view

userDao.addUser(user);

return "home";

}

}

Home.jsp

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"* %>

<%@ page session=*"false"* %>

<html>

<head>

<title>Home</title>

</head>

<body>

<h1> Welcome to movie Review Database!</h1>

<a href=*"/movie/signup.htm"*>Signup</a> <a href=*"/movie/login.htm"*>login</a><br/>

</body>

</html>

Delete-user.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<c:forEach var=*"user"* items=*"*${u}*"*>

<b>User:</b> <c:out value=*"*${user.getuName()}*"*/><br>

<b>User Type:</b> <c:out value=*"*${user.getuType()}*"*/><br>

<!-- <b>Id:</b> <c:out value="${movie.getMovieId()}"/><br> -->

<a href=*'/movie/admin/deleteUser.htm?id=*${user.getId()}*'* >Delete</a>

<a href=*'/movie/admin/remove.htm?id=*${user.getId()}*'* >Remove Critic</a>

<br><br>

</c:forEach>

<a href=*"/movie/logout.htm"*>Logout</a>

</body>

</html>

Add-movie-from.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<body>

<form:form commandName=*"movie"* >

Name : <form:input path=*"name"*/> <form:errors path=*"name"*/><br/>

Actor : <form:input path=*"actor"*/> <form:errors path=*"actor"*/> <br/>

Actress : <form:input path=*"actress"*/> <form:errors path=*"actress"*/> <br/>

Genre : <form:password path=*"genre"*/> <form:errors path=*"genre"*/> <br/>

Year : <form:input path=*"year"*/> <form:errors path=*"year"*/> <br/>

<input type=*"submit"*/>

</form:form>

</body>

</html>

Admin-hom.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"movie/add.htm"*>add movies</a>

<a href=*'/movie/admin/delete.htm?id=*${user.getId()}*'* >Delete Users/ Update Critics</a>

<a href=*"/movie/logout.htm"*>Logout</a>

</body>

</html>

Movie-search-result.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<c:forEach var=*"movie"* items=*"*${requestScope.movieList}*"*>

<b>Title:</b> <c:out value=*"*${movie.getName()}*"*/><br>

<b>Actor:</b> <c:out value=*"*${movie.getActor()}*"*/><br>

<b>Actress:</b> <c:out value=*"*${movie.getActress()}*"*/><br>

<b>Genre:</b> <c:out value=*"*${movie.getGenre()}*"*/><br>

<b>Year:</b> <c:out value=*"*${movie.getYear()}*"*/><br>

<!-- <b>Id:</b> <c:out value="${movie.getMovieId()}"/><br> -->

<a href=*'/movie/movie/addreview.htm?id=*${movie.getMovieId()}*'* >Reviews</a>

<br><br>

</c:forEach>

<a href=*"/movie/logout.htm"*>Logout</a>

</body>

</html>

Welcome Jsp:

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"movie/search.htm"*> Search For Movie </a><br>

<a href=*"movie/critics.htm"*>Look at Critic Reviews</a><br>

<a href=*"/movie/logout.htm"*>Logout</a>

</body>

</html>

Signup form :

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Signup Form</title>

</head>

<body>

<form:form commandName=*"user"*>

First Name : <form:input path=*"fName"*/> <form:errors path=*"fName"*/><br/>

Last Name : <form:input path=*"lName"*/> <form:errors path=*"lName"*/> <br/>

User Name : <form:input path=*"uName"*/> <form:errors path=*"uName"*/> <br/>

Password : <form:password path=*"password"*/> <form:errors path=*"password"*/> <br/>

Email : <form:password path=*"email"*/> <form:errors path=*"email"*/> <br/>

Account Type : <form:select path=*"uType"*>

<form:option value=*"1"*>Viewer</form:option>

<form:option value=*"2"*>Critic</form:option>

</form:select>

<input type=*"submit"*/>

</form:form>

</body>

</html>

Search page:

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

Search :

<form:form action=*"/movie/movie/search.htm"* method=*"Post"*>

<label for=*"keyword"*><b>Keyword</b></label>

<input type=*"text"* placeholder=*"Keyword"* name=*"keyword"* required><br><br>

<label for=*"criteria"*></label>

<input type=*"radio"* name=*"criteria"* value =*"name"* required>Search by Title<br>

<input type=*"radio"* name=*"criteria"* value =*"actor"*>Search by Actor<br>

<input type=*"radio"* name=*"criteria"* value =*"actress"*>Search by Actress<br>

<input type=*"radio"* name=*"criteria"* value =*"genre"*>Search by Actress<br>

<input type=*"submit"* name=*"submit"* value=*"Search Movies"*>

</form:form>

</body>

</html>

Review Page:

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<b>Title:</b> <c:out value=*"*${movie.getName()}*"*/><br>

<b>Actor:</b> <c:out value=*"*${movie.getActor()}*"*/><br>

<b>Actress:</b> <c:out value=*"*${movie.getActress()}*"*/><br>

<b>Genre:</b> <c:out value=*"*${movie.getGenre()}*"*/><br>

<b>Year:</b> <c:out value=*"*${movie.getYear()}*"*/><br><br>

<b>Average Rating:</b> <c:out value=*"*${avgRating}*"*/><br><br>

<c:forEach var=*"review"* items=*"*${movie.getReviews()}*"*>

Comment : <c:out value=*"*${review.getComment()}*"*/><br>

Rating : <c:out value=*"*${review.getStar()}*"*/><br>

</c:forEach>

<br><br>

<form action=*'/movie/movie/addreview.htm?id=*${movie.getMovieId()}*'* method=*"post"*>

Comment : <input type=*"text"* name=*"comment"* required=*"required"*/> <br/>

Rating : <select name=*"star"*>

<option value=*"1"*>1</option>

<option value=*"2"*>2</option>

<option value=*"3"*>3</option>

<option value=*"4"*>4</option>

<option value=*"5"*>5</option>

</select>

<input type=*"submit"* name=*"Post"*/>

</form>

<a href=*"/movie/logout.htm"*>Logout</a>

</body>

</html>

DAO:

package com.me.dao;

import java.util.logging.Level;

import java.util.logging.Logger;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class DAO {

private static final Logger log = Logger.getAnonymousLogger();

private static final ThreadLocal sessionThread = new ThreadLocal();

private static final SessionFactory sessionFactory = new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

protected DAO() {

}

public static Session getSession()

{

Session session = (Session) DAO.sessionThread.get();

if (session == null)

{

session = sessionFactory.openSession();

DAO.sessionThread.set(session);

}

return session;

}

protected void begin() {

getSession().beginTransaction();

}

protected void commit() {

getSession().getTransaction().commit();

}

protected void rollback() {

try {

getSession().getTransaction().rollback();

} catch (HibernateException e) {

log.log(Level.WARNING, "Cannot rollback", e);

}

try {

getSession().close();

} catch (HibernateException e) {

log.log(Level.WARNING, "Cannot close", e);

}

DAO.sessionThread.set(null);

}

public static void close() {

getSession().close();

DAO.sessionThread.set(null);

}

}

MovieDao:

package com.me.dao;

import org.hibernate.Criteria;

import org.hibernate.HibernateException;

import org.hibernate.criterion.Criterion;

import org.hibernate.criterion.Restrictions;

import java.util.List;

import com.me.pojo.Movie;

import com.me.exception.MovieException;

//import antlr.collections.List;

public class MovieDao extends DAO{

public Movie addMovie(Movie movie) {

try {

begin();

System.out.println("inside DAO");

getSession().save(movie);

commit();

close();

}

catch(HibernateException e) {

e.printStackTrace();

}

return movie;

}

public List<Movie> getMovies(String criteria,String keyword) throws com.me.exception.MovieException{

try {

begin();

@SuppressWarnings("deprecation")

Criteria c = getSession().createCriteria(Movie.class);

Criterion c1 = Restrictions.ilike(criteria, "%"+keyword+"%");

c.add(c1);

// List results = c.list();

List<Movie> movies = c.list();

close();

return movies;

}

catch(HibernateException e) {

e.printStackTrace();

rollback();

throw new com.me.exception.MovieException("Could not List movie", e);

}

}

public Movie getMovieById(long id) throws com.me.exception.MovieException{

try {

begin();

@SuppressWarnings("deprecation")

Criteria c = getSession().createCriteria(Movie.class);

Criterion c1 = Restrictions.eq("movieId", id);

c.add(c1);

Movie movie = (Movie) c.uniqueResult();

close();

return movie;

}

catch(HibernateException e) {

e.printStackTrace();

rollback();

}

return null;

}

}

ReviewDao

package com.me.dao;

import java.util.List;

import org.hibernate.Criteria;

import org.hibernate.HibernateException;

import org.hibernate.criterion.Criterion;

import org.hibernate.criterion.Restrictions;

import com.me.exception.ReviewException;

import com.me.pojo.Movie;

import com.me.pojo.Review;

public class ReviewDao extends DAO {

public Review create(Review review) throws ReviewException {

try {

begin();

System.out.println(review.getMovie());

getSession().save(review);

commit();

close();

return review;

} catch (HibernateException e) {

rollback();

throw new ReviewException("Exception while creating review: " + e.getMessage());

}

}

public List<Review> getReviewsById(long movieId) throws ReviewException{

try {

begin();

@SuppressWarnings("deprecation")

Criteria c = getSession().createCriteria(Review.class);

Criterion c1 = Restrictions.eq("movieId", movieId);

c.add(c1);

// List results = c.list();

List<Review> movie = (List<Review>) c.list();

close();

return movie;

}

catch(HibernateException e) {

e.printStackTrace();

rollback();

throw new com.me.exception.ReviewException("Could not List movie", e);

}

}

}

UserDao:

package com.me.dao;

import java.util.List;

import org.hibernate.Criteria;

import org.hibernate.HibernateException;

import org.hibernate.Query;

import org.hibernate.criterion.Criterion;

import org.hibernate.criterion.Restrictions;

import com.me.pojo.Movie;

import com.me.pojo.Review;

import com.me.pojo.User;

import com.me.exception.UserException;

@SuppressWarnings("deprecation")

public class UserDao extends DAO {

public User addUser(User user) {

try {

begin();

System.out.println("inside DAO");

getSession().save(user);

commit();

}

catch(HibernateException e) {

e.printStackTrace();

}

close();

return user;

}

public User getUser(String username,String password) throws UserException {

try {

System.out.print("in getuser");

begin();

Query q = getSession().createQuery("from User where uName = :username and password = :password");

q.setString("username", username);

q.setString("password", password);

User user = (User) q.uniqueResult();

commit();

close();

return user;

} catch (HibernateException e) {

System.out.print("prolly a problem");

rollback();

throw new UserException("no user Found");

}

}

public List<User> getCritics() throws UserException {

try {

begin();

Query q = getSession().createQuery("from User where uType = 2");

List<User> user = (List<User>) q.list();

commit();

close();

return user;

}

catch (HibernateException e) {

System.out.print("prolly a problem");

rollback();

throw new UserException("no user Found");

}

}

public void delete(User u) {

try {

begin();

getSession().delete(u);

System.out.println("Is this here");

commit();

} catch (HibernateException e) {

rollback();

System.out.println(" Or here");

e.printStackTrace();

}

close();

}

public User getUserById(long id) {

try {

begin();

Criteria c = getSession().createCriteria(User.class);

Criterion c1 = Restrictions.eq("userID", id);

c.add(c1);

User u = (User)c;

close();

return u;

}

catch(HibernateException e) {

e.printStackTrace();

rollback();

}

return null;

}

public void changeUserType(long id) {

User u = getUserById(id);

delete(u);

u.setuType("1");

addUser(u);

}

public List<User> getAllUsers() throws UserException{

try {

begin();

Query q = getSession().createQuery("from User");

List<User> user = (List<User>) q.list();

commit();

close();

return user;

}

catch (HibernateException e) {

System.out.print("prolly a problem");

rollback();

throw new UserException("no user Found");

}

}

}

User

package com.me.pojo;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Transient;

import org.springframework.web.multipart.MultipartFile;

@Entity

public class Movie {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "movieId", unique = true, nullable = false)

private long movie;

@Column

private String ne;

@Column

private String ctor;

@Column

private String aress;

@Column

private String gere;

@Column

private int yer;

@Column

private String smary;

@Transient

private MultipartFile poster;

public MultipartFile getPoster() {

return poster;

}

public void setPoster(MultipartFile poster) {

this.poster = poster;

}

@OneToMany(fetch = FetchType.EAGER,mappedBy = "movie", cascade = CascadeType.ALL, orphanRemoval = true)

private Set<Review> reviews= new HashSet<Review>();

public String getSummary() {

return summary;

}

public Set<Review> getReviews() {

return reviews;

}

public void setReviews(Set<Review> reviews) {

this.reviews = reviews;

}

public void setSummary(String summary) {

this.summary = summary;

}

public long getMovieId() {

return movieId;

}

public void setMovieId(long movieId) {

this.movieId = movieId;

}

public int getYear() {

return year;

}

public void setYear(int year) {

this.year = year;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getActor() {

return actor;

}

public void setActor(String actor) {

this.actor = actor;

}

public String getActress() {

return actress;

}

public void setActress(String actress) {

this.actress = actress;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

}

Review

package com.me.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Transient;

User:

package com.me.pojo;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "userID", unique = true, nullable = false)

private long id;

public Set<Review> getReviews() {

return reviews;

}

public void setReviews(Set<Review> reviews) {

this.reviews = reviews;

}

@Column

private String fName;

@Column

private String lName;

@Column

private String uName;

@Column

private String password;

@Column

private String email;

@Column

private String uType;

@OneToMany(fetch = FetchType.EAGER,mappedBy = "postedBy", cascade = CascadeType.ALL, orphanRemoval = true)

private Set<Review> reviews;

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getfName() {

return fName;

}

public void setfName(String fName) {

this.fName = fName;

}

public String getlName() {

return lName;

}

public void setlName(String lName) {

this.lName = lName;

}

public String getuName() {

return uName;

}

public void setuName(String uName) {

this.uName = uName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getuType() {

return uType;

}

public void setuType(String uType) {

this.uType = uType;

}

}