Project Summary

Name of the project: prandium.com

Domain: Food review website.

Summary:

This project aims at helping food critics around the world to unite and chat about different restaurants and bars. The restaurant owners can upload their restaurant location, photos, etc. Food critics can sign up on the website and talk about different cuisines and review different restaurants. This website will also help new users scout for good restaurants in their neighbourhood.

Roles:

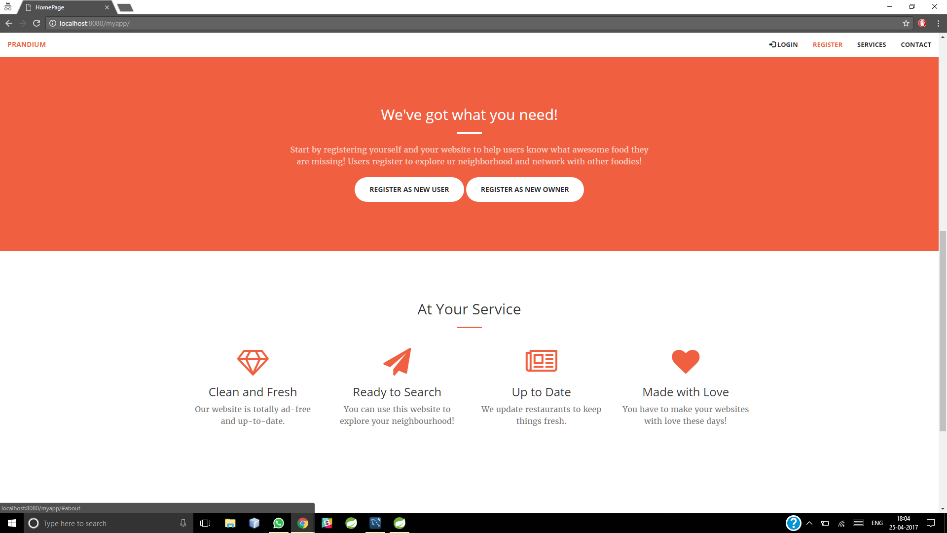
1. Food critics (Users).
2. Restaurant Owners.
3. Website Admin.

Communication between the roles:

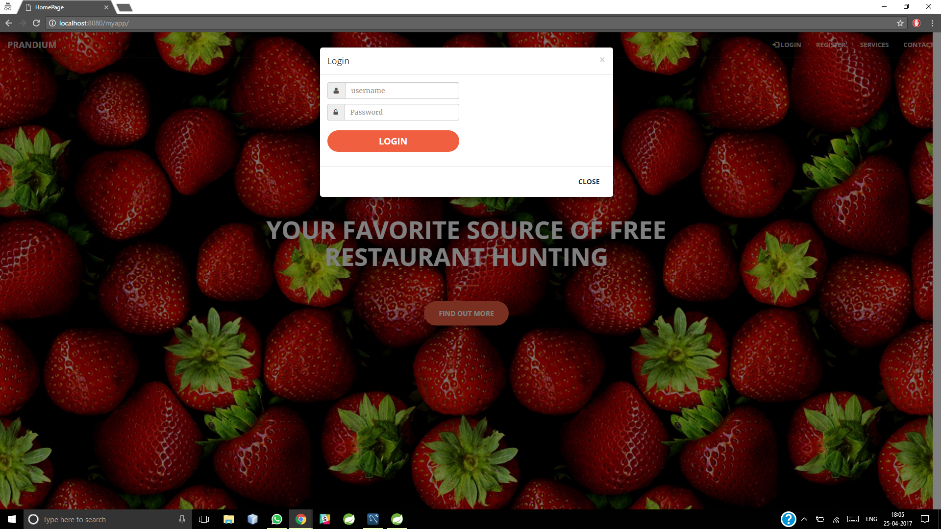
The users will register on the website and they will search and review the restaurants that were posted by the restaurant owners. The restaurant owners will also register on the website and then register their restaurants. The admin will delete the malicious accounts and restaurants and communicate with the users and restaurant owners. The users can also chat with each other and restaurant owners.

Screenshot of key screens:

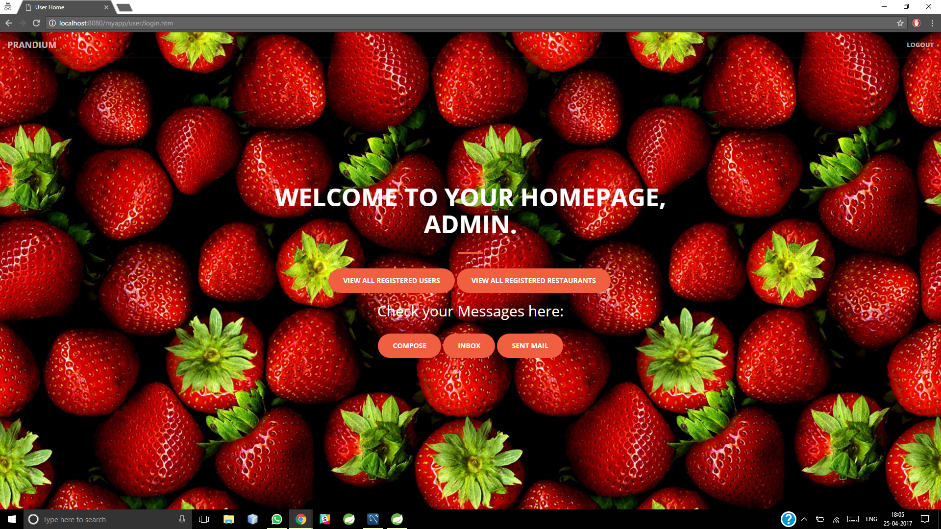
Registration link on home page:



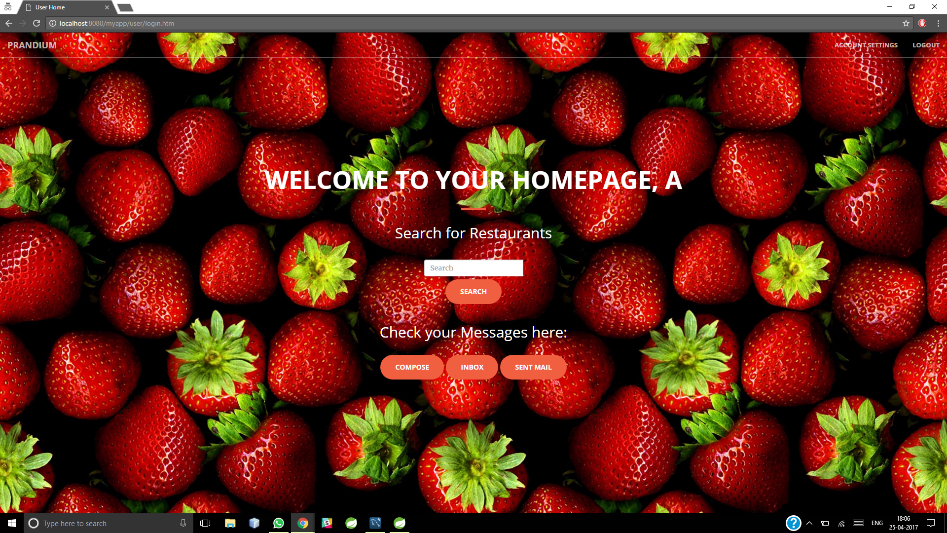
Login modal on home page:



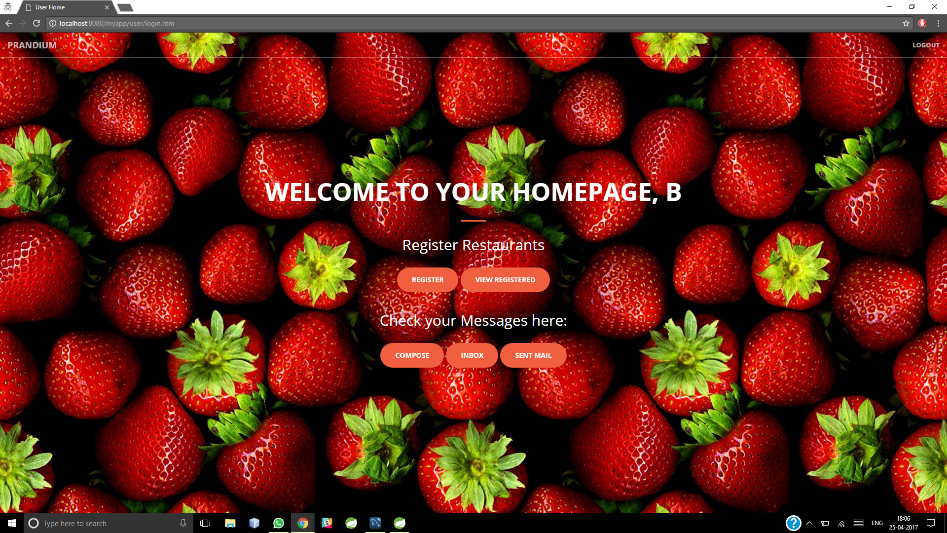
Admin Home Page:



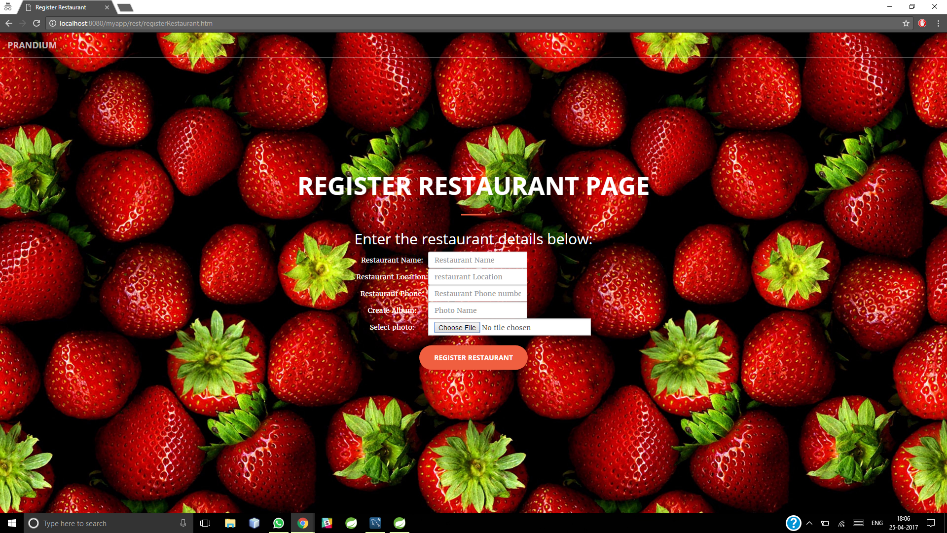
Registered user home page: (User First Name is A)



Restaurant Owner home page (Owner First Name is B)



Register Restaurant Page (Owners):



Controllers:

Home Controller:

package com.prandium.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class HomeController {

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

protected String goToUserHome(HttpServletRequest request) throws Exception {

return "home";

}

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

protected String logOut(HttpServletRequest request) throws Exception {

HttpSession session = request.getSession();

session.invalidate();

System.out.println("Log Out Successful");

return "home";

}

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

protected String denyAccess(HttpServletRequest request) throws Exception {

return "access";

}

}

Message Controller:

package com.prandium.controller;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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.prandium.dao.MessageDAO;

import com.prandium.exception.MessageException;

import com.prandium.filter.XSSFilter;

import com.prandium.pojo.Messages;

import com.prandium.pojo.User;

import com.prandium.validator.MessagesValidator;

@Controller

@RequestMapping("/message/\*")

public class MessageController {

@Autowired

@Qualifier("messageDao")

MessageDAO messageDao;

@Autowired

@Qualifier("messagesValidator")

MessagesValidator messagesValidator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(messagesValidator);

}

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

protected ModelAndView getMessageHome(HttpServletRequest request) throws Exception {

System.out.print("Message Home(GET)");

HttpSession session = (HttpSession) request.getSession();

String username = (String)session.getAttribute("username");

List<Messages> messageList = messageDao.getMessagesFromUser(username);

return new ModelAndView("message-home", "messageList", messageList);

}

@RequestMapping(value = "/message/inbox", method = RequestMethod.POST)

protected ModelAndView setMessageHome(HttpServletRequest request) throws Exception {

System.out.print("Message Home(POST)");

HttpSession session = (HttpSession) request.getSession();

String username = (String)session.getAttribute("username");

List<Messages> messageList = messageDao.getMessagesFromUser(username);

return new ModelAndView("message-home", "messageList", messageList);

}

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

protected ModelAndView getReplyMessage() throws Exception{

return new ModelAndView("message-home", "messages", new Messages());

}

@RequestMapping(value = "/message/reply", method = RequestMethod.POST)

protected ModelAndView replyMessage(HttpServletRequest request, @ModelAttribute("messages") Messages messages, BindingResult result) throws Exception {

System.out.print("New Message Reply");

messagesValidator.validate(messages, result);

if (result.hasErrors()) {

return new ModelAndView("message-home", "messages", messages);

}

try {

System.out.print("registerNewRestaurant");

String message = messages.getMessage();

String finalMessage = XSSFilter.removeXSS(message);

messages.setMessage(finalMessage);

HttpSession session = request.getSession();

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

String recipient = request.getParameter("to");

Date d = new Date();

SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd");

String messageDate = format.format(d);

messages.setMessageDate(messageDate);

messages.setUserName(user.getUsername());

messages.setFromUser(recipient);

Messages m = messageDao.insertMessage(messages);

request.getSession().setAttribute("messages", m);

return new ModelAndView("message-success", "messages", m);

} catch (MessageException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error adding restaurant");

}

}

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

protected ModelAndView getComposeMessage() throws Exception{

return new ModelAndView("message-compose", "messages", new Messages());

}

@RequestMapping(value="/message/compose", method= RequestMethod.POST)

protected ModelAndView composeMessage() throws Exception{

//write code to insert message in database

return new ModelAndView("message-compose", "messages", new Messages());

}

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

protected ModelAndView getSentMessage(HttpServletRequest request) throws Exception{

HttpSession session = (HttpSession) request.getSession();

String username = (String)session.getAttribute("username");

List<Messages> messageList = messageDao.getMessagesForUser(username);

return new ModelAndView("message-sent", "messageList", messageList);

}

@RequestMapping(value="/message/sentMail", method= RequestMethod.POST)

protected ModelAndView setSentMessage(HttpServletRequest request) throws Exception{

HttpSession session = (HttpSession) request.getSession();

String username = (String)session.getAttribute("username");

List<Messages> messageList = messageDao.getMessagesForUser(username);

return new ModelAndView("message-sent", "messageList", messageList);

}

}

Restaurant Controller:

package com.prandium.controller;

import java.io.File;

import java.io.IOException;

import java.util.List;

import javax.servlet.ServletContext;

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.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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 org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.prandium.dao.RestaurantDAO;

import com.prandium.exception.RestaurantException;

import com.prandium.pojo.Restaurant;

import com.prandium.pojo.Review;

import com.prandium.pojo.User;

import com.prandium.validator.RestaurantValidator;

import com.prandium.filter.\*;

@Controller

@RequestMapping("/rest/\*")

public class RestaurantController {

@Autowired

@Qualifier("restaurantDao")

RestaurantDAO restaurantDao;

@Autowired

@Qualifier("restaurantValidator")

RestaurantValidator restaurantValidator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(restaurantValidator);

}

@Autowired

ServletContext servletContext;

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

protected ModelAndView registerRestaurant() throws Exception {

System.out.print("registerRestaurant(GET)");

return new ModelAndView("register-restaurant", "restaurant", new Restaurant());

}

@RequestMapping(value = "/rest/registerRestaurant", method = RequestMethod.POST)

protected ModelAndView registerNewRestaurant(HttpServletRequest request,

@ModelAttribute("restaurant") Restaurant restaurant, BindingResult result) throws Exception {

restaurantValidator.validate(restaurant, result);

if (result.hasErrors()) {

return new ModelAndView("register-restaurant", "restaurant", restaurant);

}

try {

String restaurantName = restaurant.getRestaurantName();

String finalRestaurantName = XSSFilter.removeXSS(restaurantName);

String restaurantLocation = restaurant.getRestaurantLocation();

String finalRestaurantLocation = XSSFilter.removeXSS(restaurantLocation);

String restaurantPhone = restaurant.getRestaurantPhone();

String finalRestaurantPhone = XSSFilter.removeXSS(restaurantPhone);

restaurant.setRestaurantName(finalRestaurantName);

restaurant.setRestaurantLocation(finalRestaurantLocation);

restaurant.setRestaurantPhone(finalRestaurantPhone);

System.out.print("registerNewRestaurant(POST)");

String registrar = (String) request.getSession().getAttribute("username");

restaurant.setRegistrar(registrar);

if (restaurant.getFilename().trim() != "" || restaurant.getFilename() != null) {

File directory;

String check = File.separator;

String path = null;

if (check.equalsIgnoreCase("\\")) {

path = servletContext.getRealPath("").replace("build\\", "");

}

if (check.equalsIgnoreCase("/")) {

path = servletContext.getRealPath("").replace("build/", "");

path += "/"; // Adding trailing slash for Mac systems.

}

directory = new File(path + "\\" + restaurant.getFilename());

boolean temp = directory.exists();

if (!temp) {

temp = directory.mkdir();

}

if (temp) {

// We need to transfer to a file

CommonsMultipartFile photoInMemory = restaurant.getPhoto();

String fileName = photoInMemory.getOriginalFilename();

// could generate file names as well

File localFile = new File(directory.getPath(), fileName);

// move the file from memory to the file

photoInMemory.transferTo(localFile);

restaurant.setFilename(localFile.getPath());

System.out.println("File is stored at" + localFile.getPath());

Restaurant r = restaurantDao.register(restaurant);

request.getSession().setAttribute("restaurant", r);

return new ModelAndView("restaurant-success", "restaurant", r);

} else {

System.out.println("Failed to create directory!");

return new ModelAndView("error", "errorMessage", "error adding restaurant");

}

}

} catch (RestaurantException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error adding restaurant");

} catch (IllegalStateException e) {

System.out.println("\*\*\* IllegalStateException: " + e.getMessage());

} catch (IOException e) {

// TODO Auto-generated catch block

System.out.println("\*\*\* IOException: " + e.getMessage());

}

return new ModelAndView("restaurant-success");

}

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

protected ModelAndView showSearchRestaurant() throws Exception {

System.out.print("searchRestaurant");

return new ModelAndView("user-home", "restaurant", new Restaurant());

}

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

protected ModelAndView searchRestaurant(HttpServletRequest request, HttpServletResponse response) throws Exception {

System.out.print("searchStarted");

try {

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

List<Restaurant> restaurantList = restaurantDao.get(searchQuery);

request.setAttribute("searchQuery", searchQuery);

return new ModelAndView("restaurant-search-results", "restaurantList", restaurantList);

} catch (RestaurantException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected ModelAndView showRestaurantProfilePage(HttpServletRequest request) throws Exception {

System.out.println("Restaurant Profile");

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

HttpSession session = request.getSession();

session.setAttribute("restaurantName", restaurantName);

System.out.println(restaurantName);

Restaurant restaurant = restaurantDao.getRestaurant(restaurantName);

List<Review> listOfReviews = restaurantDao.getReviews(restaurantName);

session.setAttribute("listOfReviews", listOfReviews);

System.out.println(restaurant);

return new ModelAndView("restaurant-profile", "restaurant", restaurant);

}

@RequestMapping(value = "/rest/profile", method = RequestMethod.POST)

protected ModelAndView setRestaurantProfilePage(HttpServletRequest request) throws Exception {

System.out.print("Restaurant Profile");

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

Restaurant restaurant = restaurantDao.getRestaurant(restaurantName);

System.out.println(restaurant);

return new ModelAndView("restaurant-profile", "restaurant", restaurant);

}

@RequestMapping(value = "/rest/viewRegistered", method = RequestMethod.POST)

protected ModelAndView viewRegistered(HttpServletRequest request, HttpServletResponse response) throws Exception {

System.out.print("searchStarted for all registered");

try {

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

List<Restaurant> restaurantList = restaurantDao.getRegistered(username);

return new ModelAndView("restaurant-search-results", "restaurantList", restaurantList);

} catch (RestaurantException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected ModelAndView getViewRegistered(HttpServletRequest request, HttpServletResponse response)

throws Exception {

System.out.print("searchStarted for all registered");

return new ModelAndView("restaurant-search-results", "restaurant", new Restaurant());

}

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

protected ModelAndView getDeleteRestaurant(HttpServletRequest request, HttpServletResponse response)

throws Exception {

System.out.print("delete restaurant(GET)");

User user = (User) request.getSession().getAttribute("user");

if (user.getType().equalsIgnoreCase("A")) {

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

Restaurant rest = restaurantDao.getRestaurant(restaurantName);

restaurantDao.deleteRestaurant(rest);

List<Restaurant> restaurantList = restaurantDao.getAllRestaurants();

return new ModelAndView("view-all-rest", "restaurantList", restaurantList);

} else {

return new ModelAndView("access");

}

}

}

Review Controller:

package com.prandium.controller;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

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

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

import org.springframework.stereotype.Controller;

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

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

import org.springframework.web.servlet.ModelAndView;

import com.prandium.dao.RestaurantDAO;

import com.prandium.exception.RestaurantException;

import com.prandium.filter.XSSFilter;

import com.prandium.pojo.Restaurant;

import com.prandium.pojo.Review;

@Controller

@RequestMapping("/review/\*")

public class ReviewController {

@Autowired

@Qualifier("restaurantDao")

RestaurantDAO restaurantDao;

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

protected ModelAndView getReviews(HttpServletRequest request) throws Exception {

System.out.print("Review Restaurant");

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

List<Review> listOfReviews = restaurantDao.getReviews(restaurantName);

return new ModelAndView("restaurant-profile", "listOfReviews", listOfReviews);

}

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

protected ModelAndView setReviews(HttpServletRequest request) throws Exception {

System.out.print("Review Restaurant");

try {

System.out.println("Inside the review post method");

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

String finalReviewMessage = XSSFilter.removeXSS(reviewMessage);

String restaurantName = (String)request.getSession().getAttribute("restaurantName");

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

System.out.println(username +" " + restaurantName);

Date d = new Date();

SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd");

String reviewDate = format.format(d);

Review review = new Review(finalReviewMessage, reviewDate, username);

review.setRestaurantName(restaurantName);

Review r = restaurantDao.addReview(review);

r.setReviewDate(reviewDate);

List<Review> listOfReviews = restaurantDao.getReviews(restaurantName);

Restaurant restaurant = restaurantDao.getRestaurant(restaurantName);

request.getSession().setAttribute("restaurant", restaurant);

return new ModelAndView("restaurant-profile", ("listOfReviews"), listOfReviews);

} catch (RestaurantException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error adding review");

}

}

}

User Controller:

package com.prandium.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

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

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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.prandium.dao.RestaurantDAO;

import com.prandium.dao.UserDAO;

import com.prandium.exception.UserException;

import com.prandium.filter.XSSFilter;

import com.prandium.pojo.Restaurant;

import com.prandium.pojo.User;

import com.prandium.validator.UserValidator;

/\*\*

\* Handles requests for the application home page.

\*/

@Controller

@RequestMapping("/user/\*")

public class UserController {

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@Autowired

@Qualifier("restaurantDao")

RestaurantDAO restaurantDao;

@Autowired

@Qualifier("userValidator")

UserValidator validator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

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

public ModelAndView goHome() {

return new ModelAndView("home");

}

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

protected String getloginUser(HttpServletRequest request) throws Exception {

return "";

}

@RequestMapping(value = "/user/login", method = RequestMethod.POST)

protected String loginUser(HttpServletRequest request) throws Exception {

HttpSession session = (HttpSession) request.getSession();

try {

System.out.print("loginUser");

String tempUsername = request.getParameter("username");

String tempPassword = request.getParameter("password");

String username = XSSFilter.removeXSS(tempUsername);

String password = XSSFilter.removeXSS(tempPassword);

User u = userDao.get(username, password);

if (u == null) {

System.out.println("UserName/Password does not exist");

session.setAttribute("errorMessage", "UserName/Password does not exist");

return "error";

}

List<Restaurant> restaurantList = restaurantDao.getAllRestaurants();

session.setAttribute("restaurantList", restaurantList);

session.setAttribute("user", u);

session.setAttribute("username", u.getUsername());

if (u.getType().equalsIgnoreCase("U")) {

return "user-home";

} else if (u.getType().equalsIgnoreCase("A")) {

return "admin-home";

} else {

return "owner-home";

}

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

session.setAttribute("errorMessage", "error while login");

return "error";

}

}

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

protected ModelAndView registerUser() throws Exception {

System.out.print("registerUser");

return new ModelAndView("register-user", "user", new User());

}

@RequestMapping(value = "/user/register", method = RequestMethod.POST)

protected ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("user") User user,

BindingResult result) throws Exception {

validator.validate(user, result);

if (result.hasErrors()) {

return new ModelAndView("register-user", "user", user);

}

try {

String tempUsername = user.getUsername();

String tempPassword = user.getPassword();

String tempFirstName = user.getFirstName();

String tempLastName = user.getLastName();

String username = XSSFilter.removeXSS(tempUsername);

String password = XSSFilter.removeXSS(tempPassword);

String firstName = XSSFilter.removeXSS(tempFirstName);

String lastName = XSSFilter.removeXSS(tempLastName);

user.setUsername(username);

user.setPassword(password);

user.setFirstName(firstName);

user.setLastName(lastName);

System.out.print("registerNewUser");

user.setType("U");

User u = userDao.register(user);

request.getSession().setAttribute("user", u);

return new ModelAndView("home", "user", u);

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

@RequestMapping(value = "/user/register-owners", method = RequestMethod.GET)

protected ModelAndView registerOwner() throws Exception {

System.out.print("registerOwner");

return new ModelAndView("register-owners", "user", new User());

}

@RequestMapping(value = "/user/register-owners", method = RequestMethod.POST)

protected ModelAndView registerNewOwner(HttpServletRequest request, @ModelAttribute("user") User user,

BindingResult result) throws Exception {

validator.validate(user, result);

if (result.hasErrors()) {

return new ModelAndView("register-owners", "user", user);

}

try {

String tempUsername = user.getUsername();

String tempPassword = user.getPassword();

String tempFirstName = user.getFirstName();

String tempLastName = user.getLastName();

String username = XSSFilter.removeXSS(tempUsername);

String password = XSSFilter.removeXSS(tempPassword);

String firstName = XSSFilter.removeXSS(tempFirstName);

String lastName = XSSFilter.removeXSS(tempLastName);

user.setUsername(username);

user.setPassword(password);

user.setFirstName(firstName);

user.setLastName(lastName);

System.out.print("registerNewUser");

user.setType("O");

User u = userDao.register(user);

request.getSession().setAttribute("user", u);

return new ModelAndView("home", "user", u);

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected ModelAndView getSettings(HttpServletRequest request) throws Exception {

System.out.print("Account settings (GET)");

User user = (User) request.getSession().getAttribute("user");

return new ModelAndView("edit-profile", "user", user);

}

@RequestMapping(value = "/user/settings", method = RequestMethod.POST)

protected ModelAndView updateSettings(HttpServletRequest request, @ModelAttribute("user") User user,

BindingResult result) throws Exception {

System.out.print("Account settings (POST)");

validator.validate(user, result);

// if (result.hasErrors()) {

// System.out.println("Got Errors");

// return new ModelAndView("edit-profile", "user", user);

// }

//

try {

User u = userDao.updateSettings(user);

request.getSession().setAttribute("user", u);

return new ModelAndView("account-success");

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

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

protected ModelAndView getAllUsers(HttpServletRequest request) throws Exception {

System.out.print("Get all Users (GET)");

User user = (User) request.getSession().getAttribute("user");

if (user.getType().equalsIgnoreCase("A")) {

List<User> userList = userDao.getAll();

return new ModelAndView("view-all-users", "userList", userList);

} else {

return new ModelAndView("access");

}

}

@RequestMapping(value = "/user/viewUsers", method = RequestMethod.POST)

protected ModelAndView setAllUsers(HttpServletRequest request) throws Exception {

System.out.print("Get all Users (POST)");

User user = (User) request.getSession().getAttribute("user");

if (user.getType().equalsIgnoreCase("A")) {

List<User> userList = userDao.getAll();

return new ModelAndView("view-all-users", "userList", userList);

} else {

return new ModelAndView("access");

}

}

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

protected ModelAndView getAllRestaurants(HttpServletRequest request) throws Exception {

System.out.print("Get all restaurants (GET)");

User user = (User) request.getSession().getAttribute("user");

if (user.getType().equalsIgnoreCase("A")) {

List<Restaurant> restaurantList = restaurantDao.getAllRestaurants();

return new ModelAndView("view-all-rest", "restaurantList", restaurantList);

} else {

return new ModelAndView("access");

}

}

@RequestMapping(value = "/user/viewRestaurants", method = RequestMethod.POST)

protected ModelAndView setAllRestaurants(HttpServletRequest request) throws Exception {

System.out.print("Get all restaurants (POST)");

User user = (User) request.getSession().getAttribute("user");

if (user.getType().equalsIgnoreCase("A")) {

List<Restaurant> restaurantList = restaurantDao.getAllRestaurants();

return new ModelAndView("view-all-rest", "restaurantList", restaurantList);

} else {

return new ModelAndView("access");

}

}

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

protected ModelAndView getDeleteUser(HttpServletRequest request) throws Exception {

System.out.print("Delete user (GET)");

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

User user = (User) request.getSession().getAttribute("user");

if (user.getType().equalsIgnoreCase("A")) {

User u = userDao.getUser(username);

userDao.delete(u);

List<User> userList = userDao.getAll();

return new ModelAndView("view-all-users", "userList", userList);

} else {

return new ModelAndView("access");

}

}

}