Final Project Sporty Shoes Specifications

**Sporty Shoes**

Class: Implement Frameworks the DevOps way

Student: Hector Alarcon

Application: sportyshoes.com

Date: 11/12/2021

Contents

[Project Description: 3](#_Toc79327167)

[Java technologies utilized: 3](#_Toc79327168)

[Unique Selling Points 3](#_Toc79327169)

[Sprint breakdown 4](#_Toc79327170)

[Program Details: 5](#_Toc79327171)

[GitHub: 5](#_Toc79327172)

# Project Description:

* This project focused on making a web application using Spring boot and Spring MVC

## Java technologies utilized:

* Spring MVC allows for quick and seamless mapping of multiple pages and the ability to circumvent the need to design multiple servlets by using Controllers that directly communicate with the rest of the project components.
* Spring boot made for the quick delivery of a prototype possible by handle ling most of the dependencies needed for the project.
* Maven framework allowing for quickly gathering and allocation of dependencies.
* Java server pages as well as some JSTLs for easier and less cumbersome acquisition and transmission of data between server pages.
* HTML and multiple tags for user input and data transmission. Mostly using the POST method for security reasons

## Unique Selling Points

This excellent application provides the following features to the customer:

* Ease of use, all the menus have user interaction information displayed on the console describing all the options available to the user.
* Use of CSS to maintain even and consistent formatting through out the entire application.
* Security, all of the class variables are set to private, and resources closed and only accessible to the class itself. Only letting the main method access the multiple resources.
* Bullet proof, sentinel values are used for input validation and making sure the user is aware of possible mistakes during inputting data.

# Sprint breakdown

|  |
| --- |
|  |
| Figure 1 PlantUML Sprint Schedule |

# Program Details:

## GitHub:

<https://github.com/ProgrammedPeinado/PracticeJava/tree/main/Simplilearn/Final%20Projects/sportyshoes>

|  |
| --- |
| Project Folder |
|  |
| AdminController |
| package com.sportyshoes.web.controller;  import java.util.List;  import java.util.ArrayList;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestParam;  import org.springframework.web.servlet.ModelAndView;  import com.sportyshoes.web.model.Admin;  import com.sportyshoes.web.service.AdminService;  import com.sportyshoes.web.service.ShoeService;  @Controller  public class AdminController  {  @Autowired  private AdminService adminService;    @Autowired  private ShoeService shoeService;    @RequestMapping("/adminLogin")  public String getAdminLogin()  {  return "admin\_login";  }    @RequestMapping("/adminPassChangePage")  public String getAdminPassChange()  {  return "adminpasschange";  }    @RequestMapping("/authenticate")  public ModelAndView authentication(@RequestParam("user") String user, @RequestParam("pass") String pass)  {  Admin admin = new Admin();  admin.setUser(user);  admin.setPass(pass);      if (adminService.authenticate(admin)) {  ModelAndView modelAndView = new ModelAndView("adminpage");  modelAndView.addObject("admin", admin);  return modelAndView;  } else {  ModelAndView modelAndView = new ModelAndView("failedadmin");  return modelAndView;  }  }    @RequestMapping("/passwordChange")  public ModelAndView passwordChange(@RequestParam("oldpass") String oldpass, @RequestParam("newpass") String newpass)  {  List<Admin> admins = adminService.getAllAdmins();  Admin a = new Admin();  a.setUser(admins.get(0).getUser());  a.setPass(admins.get(0).getPass());  boolean flag = false;    if(a.getPass().equals(oldpass))  {  ModelAndView modelAndView = new ModelAndView("changepass\_success");  flag = adminService.updateAdmin(a.getUser(), newpass);  if(flag != false)  return modelAndView;  } else {  ModelAndView modelAndView = new ModelAndView("failedadmin");  return modelAndView;  }  ModelAndView modelAndView = new ModelAndView("home");  return modelAndView;  }  } |

|  |
| --- |
| HelloController |
| package com.sportyshoes.web.controller;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.RequestMapping;  @Controller  public class HelloController  {  @RequestMapping("/home")  public String getHomepage()  {  return ("home");  }  } |

|  |
| --- |
| ShoeController |
| package com.sportyshoes.web.controller;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestParam;  import org.springframework.web.servlet.ModelAndView;  import java.io.FileNotFoundException;  import java.io.PrintWriter;  import java.util.ArrayList;  import java.util.List;  import java.util.Map;  import com.sportyshoes.web.model.Shoe;  import com.sportyshoes.web.service.ShoeService;  @Controller  public class ShoeController  {  @Autowired  private ShoeService shoeService;    @RequestMapping("/addShoePage")  public String getAddShoe()  {  return "addshoe";  }  @RequestMapping("/updateShoePage")  public String getUpdateShoe()  {  return "updateshoe";  }  @RequestMapping("/deleteShoePage")  public String getDeleteShoe()  {  return "deleteshoe";  }    @RequestMapping("/getShoes")  public ModelAndView getShoes()  {  Map<String, ArrayList<Shoe>> shoes = shoeService.getShoeTags();  ModelAndView modelAndView = new ModelAndView("displayshoes");  modelAndView.addObject("shoes", shoes);  return modelAndView;  }    @RequestMapping("/addShoe")  public ModelAndView addShoe(@RequestParam("name") String name, @RequestParam("tag") String tag, @RequestParam("thumbnail") String thumbnail)  {  ModelAndView modelAndView = new ModelAndView("addshoesuccessful");  Shoe shoe = new Shoe();  shoe.setShoe\_name(name);  shoe.setTag(tag);  shoe.setThumbnail(thumbnail);    boolean flag = shoeService.addShoe(shoe);    return modelAndView;  }    @RequestMapping("/updateShoe")  public ModelAndView updateShoe(@RequestParam("id") String id, @RequestParam("tag") String tag)  {  ModelAndView modelAndView = new ModelAndView("updateshoesuccessful");  shoeService.updateShoe(Integer.parseInt(id), tag);  return modelAndView;  }  } |

|  |
| --- |
| TransactionController |
| package com.sportyshoes.web.controller;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.servlet.ModelAndView;  import com.sportyshoes.web.model.Transaction;  import com.sportyshoes.web.service.TransactionService;  @Controller  public class TransactionController  {  @Autowired  private TransactionService transactionService;    @RequestMapping("/getTransactions")  public ModelAndView getAllTransactions()  {  List<Transaction> transactions = transactionService.getAllTransactions();  ModelAndView modelAndView = new ModelAndView("displaytransactions");  modelAndView.addObject("transactions", transactions);  return modelAndView;  }  } |

|  |
| --- |
| UserController |
| package com.sportyshoes.web.controller;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.ModelAttribute;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestParam;  import org.springframework.web.servlet.ModelAndView;  import com.sportyshoes.web.model.User;  import com.sportyshoes.web.service.UserService;  @Controller  public class UserController  {  @Autowired  private UserService userService;    @RequestMapping("/addUser")  public String getAddUser()  {  return "adduser";  }    @RequestMapping("/updateUser")  public String getUpdateUser()  {  return "updateUser";  }    @RequestMapping("/searchUserPage")  public String getSearchUser()  {  return "searchuser";  }  @RequestMapping("/searchUser")  public ModelAndView searchUser(@RequestParam("id") String id)  {  ModelAndView modelAndView = new ModelAndView("searchsuccess");  modelAndView.addObject("user", userService.searchUser(Integer.parseInt(id)));  return modelAndView;  }    @RequestMapping("/addNewUser")  public String addUser(@ModelAttribute("user") User user)  {  String viewName = null;  boolean isUserAdded = userService.addUser(user);  if(isUserAdded)  viewName = "addsuccess"; //jsp on add successful  else  viewName = "addfailure"; //jsp on add failed  return "addsuccess";  }        @RequestMapping("/updateNewUser")  public String updateUser(@RequestParam("id") int id, @RequestParam("password") String password)  {  String viewName = null;  boolean isUserUpdated = userService.updateUser(id, password);  if(isUserUpdated)  viewName = "addsuccess";  else  viewName = "addfailure";    return viewName;  }    @RequestMapping("/listUsers")  public ModelAndView getAllUsers()  {  List<User> allUsers = userService.getAllUsers();  ModelAndView modelAndView = new ModelAndView("displayusers");  modelAndView.addObject("users", allUsers);  return modelAndView;    }  } |

|  |
| --- |
| AdminDaoImpl |
| package com.sportyshoes.web.dao;  import java.util.ArrayList;  import java.util.List;  import java.util.Map;  import javax.annotation.PostConstruct;  import javax.sql.DataSource;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.jdbc.core.support.JdbcDaoSupport;  import org.springframework.stereotype.Repository;  import com.sportyshoes.web.model.Admin;  import com.sportyshoes.web.model.Shoe;  @Repository  public class AdminDaoImpl extends JdbcDaoSupport implements AdminDao  {  @Autowired  DataSource dataSource;    @PostConstruct  private void initialize()  {  setDataSource(dataSource);  }    @Override  public boolean updateAdmin(String username, String password)  {  boolean isAdminUpdated = false;  String sql = "update administrators set password=? where username=?";  int val = getJdbcTemplate().update(sql, password, username);  if(val > 0)  isAdminUpdated = true;    return isAdminUpdated;  }  @SuppressWarnings("deprecation")  @Override  public boolean authenticate(Admin admin)  {  boolean flag = false;  String sql = "select username from administrators where password=?";  String check = null;    try  {  check = getJdbcTemplate().queryForObject(sql, new Object[]{admin.getPass()}, String.class);  }  catch(Exception e)  {  System.out.println(e.getLocalizedMessage());  }    if(admin.getUser().equals(check))  {  flag = true;  }    return flag;  }    @Override  public List<Admin> getAllAdmins()  {  List<Admin> admins = new ArrayList<Admin>();  String sql = "select \* from administrators";  List<Map<String, Object>> adminData = getJdbcTemplate().queryForList(sql);    for(Map<String, Object> s: adminData)  {  Admin admin = new Admin();  admin.setUser((String)s.get("username"));  admin.setPass((String)s.get("password"));  admins.add(admin);  }  return admins;  }  } |

|  |
| --- |
| ShoeDaoImpl |
| package com.sportyshoes.web.dao;  import java.util.ArrayList;  import java.util.HashMap;  import java.util.List;  import java.util.Map;  import javax.annotation.PostConstruct;  import javax.sql.DataSource;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.jdbc.core.support.JdbcDaoSupport;  import org.springframework.stereotype.Repository;  import com.sportyshoes.web.model.Shoe;  import com.sportyshoes.web.model.User;  @Repository  public class ShoeDaoImpl extends JdbcDaoSupport implements ShoeDao  {  @Autowired  DataSource dataSource;    @PostConstruct  private void initialize()  {  setDataSource(dataSource);  }    @Override  public boolean addShoe(Shoe shoe) {  boolean flag = false;  String sql = "insert into shoes values(?,?,?,?)";  int val = getJdbcTemplate().update(sql, new Object[] {shoe.getShoe\_id(), shoe.getShoe\_name(), shoe.getTag(), shoe.getThumbnail()});  if(val > 0)  flag = true;    return flag;  }  @Override  public List<Shoe> getAllShoes() {  List<Shoe> shoes = null;  String sql = "select \* from shoes";  List<Map<String, Object>> shoeData = getJdbcTemplate().queryForList(sql);    for(Map<String, Object> s: shoeData)  {  Shoe shoe = new Shoe();  shoe.setShoe\_id((Integer)s.get("shoe\_id"));  shoe.setShoe\_name((String)s.get("shoe\_name"));  shoe.setTag((String)s.get("tag"));  shoe.setThumbnail((String)s.get("thumbnail"));  shoes.add(shoe);  }    return shoes;  }  @Override  public Map<String, ArrayList<Shoe>> getShoeTags() {  Map<String, ArrayList<Shoe>> shoes = new HashMap<>();      String sql = "select \* from shoes";  List<Map<String, Object>> shoeData = getJdbcTemplate().queryForList(sql);    for(Map<String, Object> keys: shoeData)  {  ArrayList<Shoe> values = new ArrayList<Shoe>();    for(Map<String, Object> value : shoeData)  {  if(value.get("tag").equals(keys.get("tag")))  {  Shoe shoe = new Shoe();  shoe.setShoe\_id((Integer)value.get("shoe\_id"));  shoe.setShoe\_name((String)value.get("shoe\_name"));  shoe.setTag((String)value.get("tag"));  shoe.setThumbnail((String)value.get("thumbnail"));  values.add(shoe);  }    }    shoes.put((String)keys.get("tag"), values);  values=null;  }    //Returns a Map of <Tags, Products with the Tag>  return shoes;  }  @Override  public Shoe searchShoe(int id)  {  // TODO Auto-generated method stub  return null;  }  @Override  public boolean updateShoe(int shoe\_id, String tag)  {  boolean isShoeUpdated = false;  String sql = "update shoes set tag=? where shoe\_id=?";  int val = getJdbcTemplate().update(sql, tag, shoe\_id);  if(val > 0)  isShoeUpdated = true;    return isShoeUpdated;  }  @Override  public void deleteShoe(int id)  {  }  } |

|  |
| --- |
| TransactionDaoImpl |
| package com.sportyshoes.web.dao;  import java.time.LocalDateTime;  import java.util.ArrayList;  import java.util.List;  import java.util.Map;  import javax.annotation.PostConstruct;  import javax.sql.DataSource;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.jdbc.core.support.JdbcDaoSupport;  import org.springframework.stereotype.Repository;  import com.sportyshoes.web.model.Transaction;  @Repository  public class TransactionDaoImpl extends JdbcDaoSupport implements TransactionDao  {  @Autowired  DataSource dataSource;    @PostConstruct  private void initialize()  {  setDataSource(dataSource);  }    @Override  public boolean addTransaction(Transaction transaction) {  boolean flag = false;  String sql = "insert into transactions values(?,?,?,?,?)";  int val = getJdbcTemplate().update(sql, new Object[] {transaction.getTransaction\_id(), transaction.getBuyer\_id(), transaction.getProduct\_id(), transaction.getCategory(), transaction.getTransaction\_date()});  if(val > 0)  flag = true;    return flag;  }  @Override  public List<Transaction> getAllTransactions() {  List<Transaction> transactions = new ArrayList<Transaction>();  String sql = "select \* from transactions ORDER BY category,transaction\_date ASC";  List<Map<String, Object>> transactionData = getJdbcTemplate().queryForList(sql);    for(Map<String, Object> t: transactionData)  {  Transaction trans = new Transaction();  trans.setTransaction\_id((Integer)t.get("transaction\_id"));  trans.setBuyer\_id((Integer)t.get("Buyer\_id"));  trans.setProduct\_id((Integer)t.get("product\_id"));  trans.setCategory((String)t.get("category"));  trans.setTransaction\_date((LocalDateTime)t.get("transaction\_date"));    if(trans !=null)  {  transactions.add(trans);  }  }    return transactions;  }  @Override  public Transaction searchTransaction(int id) {  // TODO Auto-generated method stub  return null;  }  @Override  public void deleteTransaction(int id) {  // TODO Auto-generated method stub  }  } |

|  |
| --- |
| UserDaoImpl |
| package com.sportyshoes.web.dao;  import java.util.ArrayList;  import java.util.List;  import java.util.Map;  import javax.annotation.PostConstruct;  import javax.sql.DataSource;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.jdbc.core.support.JdbcDaoSupport;  import org.springframework.stereotype.Repository;  import com.sportyshoes.web.model.User;  @Repository  public class UserDaoImpl extends JdbcDaoSupport implements UserDao  {  @Autowired  DataSource dataSource;    @PostConstruct  private void initialize()  {  setDataSource(dataSource);  }    @Override  public boolean addUser(User user)  {  boolean flag = false;  String sql = "insert into user values(?,?,?,?,?)";  int val = getJdbcTemplate().update(sql, new Object[] {user.getId(), user.getFirstname(), user.getLastname(), user.getPhoneNumber(),user.getEmail()});  if(val > 0)  flag = true;    return flag;  }  @Override  public List<User> getAllUsers()  {  List<User> users = new ArrayList();  String sql = "select \* from user";  List<Map<String, Object>> userData = getJdbcTemplate().queryForList(sql);    for(Map<String, Object> usr: userData)  {  User user = new User();  user.setId((Integer)usr.get("id"));  user.setFirstname((String)usr.get("firstname"));  user.setLastname((String)usr.get("lastname"));  user.setPhoneNumber((String)usr.get("phoneNumber"));  user.setEmail((String)usr.get("email"));    if(user != null)  {  users.add(user);  }    }    return users;  }  @Override  public User searchUser(int id)  {  boolean flag = false;  User user = new User();  List<Map<String, Object>> userData= null;  String sql = "select \* from user where id=?";  String check = null;    try  {  userData = getJdbcTemplate().queryForList(sql, id); //, new Object[] user.getFirstname(), user.getLastname(), user.getPhoneNumber(), user.getEmail(), User.class)  for(Map<String, Object> usr: userData)  {  user.setId((Integer)usr.get("id"));  user.setFirstname((String)usr.get("firstname"));  user.setLastname((String)usr.get("lastname"));  user.setPhoneNumber((String)usr.get("phoneNumber"));  user.setEmail((String)usr.get("email"));  }  }  catch(Exception e)  {  System.out.println(e.getLocalizedMessage());  }  return user;  }  @Override  public boolean updateUser(int id, String password)  {  boolean isUserUpdated = false;  String sql = "update user set password=? where id=?";  int val = getJdbcTemplate().update(sql, password, id);  if(val > 0)  isUserUpdated = true;    return isUserUpdated;  }  @Override  public void deleteUser(int id)  {  }  } |

|  |
| --- |
| Admin |
| package com.sportyshoes.web.model;  import javax.persistence.Column;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.GenerationType;  import javax.persistence.Id;  import javax.persistence.Table;  public class Admin  {  private String user;  private String pass;    public String getUser() {  return user;  }  public void setUser(String user) {  this.user = user;  }  public String getPass() {  return pass;  }  public void setPass(String pass) {  this.pass = pass;  }    @Override  public String toString() {  return "Admin [user=" + user + ", pass=" + pass + "]";  }  } |

|  |
| --- |
| Shoe |
| **package** com.sportyshoes.web.model;  **public** **class** Shoe  {  **private** **int** shoe\_id;  **private** String shoe\_name;  **private** String tag;  **private** String thumbnail;    **public** **int** getShoe\_id() {  **return** shoe\_id;  }  **public** **void** setShoe\_id(**int** shoe\_id) {  **this**.shoe\_id = shoe\_id;  }  **public** String getShoe\_name() {  **return** shoe\_name;  }  **public** **void** setShoe\_name(String shoe\_name) {  **this**.shoe\_name = shoe\_name;  }  **public** String getTag() {  **return** tag;  }  **public** **void** setTag(String tag) {  **this**.tag = tag;  }  **public** String getThumbnail() {  **return** thumbnail;  }  **public** **void** setThumbnail(String thumbnail) {  **this**.thumbnail = thumbnail;  }    @Override  **public** String toString() {  **return** "Shoe [shoe\_id=" + shoe\_id + ", shoe\_name=" + shoe\_name + ", tag=" + tag + ", thumbnail=" + thumbnail  + "]";  }  } |

|  |
| --- |
| Transaction |
| **package** com.sportyshoes.web.model;  **import** java.time.LocalDateTime;  **public** **class** Transaction  {  **private** **int** transaction\_id;  **private** **int** buyer\_id;  **private** **int** product\_id;  **private** String category;  **private** LocalDateTime transaction\_date;  **public** **int** getTransaction\_id() {  **return** transaction\_id;  }  **public** **void** setTransaction\_id(**int** transaction\_id) {  **this**.transaction\_id = transaction\_id;  }  **public** **int** getBuyer\_id() {  **return** buyer\_id;  }  **public** **void** setBuyer\_id(**int** buyer\_id) {  **this**.buyer\_id = buyer\_id;  }  **public** **int** getProduct\_id() {  **return** product\_id;  }  **public** **void** setProduct\_id(**int** product\_id) {  **this**.product\_id = product\_id;  }  **public** String getCategory() {  **return** category;  }  **public** **void** setCategory(String category) {  **this**.category = category;  }  **public** LocalDateTime getTransaction\_date() {  **return** transaction\_date;  }  **public** **void** setTransaction\_date(LocalDateTime transaction\_date) {  **this**.transaction\_date = transaction\_date;  }    @Override  **public** String toString() {  **return** "Transaction [transaction\_id=" + transaction\_id + ", buyer\_id=" + buyer\_id + ", product\_id=" + product\_id  + ", category=" + category + ", transaction\_date=" + transaction\_date + "]";  }  } |

|  |
| --- |
| User |
| **package** com.sportyshoes.web.model;  **public** **class** User  {  **private** **int** id;  **private** String firstname;  **private** String lastname;  **private** String phoneNumber;  **private** String email;    **public** **int** getId() {  **return** id;  }  **public** **void** setId(**int** id) {  **this**.id = id;  }  **public** String getFirstname() {  **return** firstname;  }  **public** **void** setFirstname(String firstname) {  **this**.firstname = firstname;  }  **public** String getLastname() {  **return** lastname;  }  **public** **void** setLastname(String lastname) {  **this**.lastname = lastname;  }  **public** String getPhoneNumber() {  **return** phoneNumber;  }  **public** **void** setPhoneNumber(String phoneNumber) {  **this**.phoneNumber = phoneNumber;  }  **public** String getEmail() {  **return** email;  }  **public** **void** setEmail(String email) {  **this**.email = email;  }    @Override  **public** String toString() {  **return** "User [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname  + ", phoneNumber=" + phoneNumber + ", email=" + email +"]";  }  } |

|  |
| --- |
| AdminServiceImpl |
| package com.sportyshoes.web.service;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.sportyshoes.web.dao.AdminDaoImpl;  import com.sportyshoes.web.model.Admin;  @Service  public class AdminServiceImpl implements AdminService {    @Autowired  private AdminDaoImpl adminDao;    @Override  public boolean updateAdmin(String admin, String pass)  {  return adminDao.updateAdmin(admin, pass);  }  @Override  public boolean authenticate(Admin admin)  {  return adminDao.authenticate(admin);  }    @Override  public List<Admin> getAllAdmins()  {  return adminDao.getAllAdmins();  }  } |

|  |
| --- |
| ShoeServiceImpl |
| package com.sportyshoes.web.service;  import java.util.ArrayList;  import java.util.List;  import java.util.Map;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.sportyshoes.web.dao.ShoeDaoImpl;  import com.sportyshoes.web.model.Shoe;  @Service  public class ShoeServiceImpl implements ShoeService  {  @Autowired  private ShoeDaoImpl shoeDao;    @Override  public boolean addShoe(Shoe shoe)  {  return shoeDao.addShoe(shoe);  }  @Override  public List<Shoe> getAllShoes()  {  return shoeDao.getAllShoes();  }  @Override  public Map<String, ArrayList<Shoe>> getShoeTags()  {  return shoeDao.getShoeTags();  }  @Override  public Shoe searchShoe(int id)  {  return shoeDao.searchShoe(id);  }  @Override  public boolean updateShoe(int shoe\_id, String tag) {  return shoeDao.updateShoe(shoe\_id, tag);  }  @Override  public void deleteShoe(int id)  {  shoeDao.deleteShoe(id);  }  } |

|  |
| --- |
| TransactionServiceImpl |
| package com.sportyshoes.web.service;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.sportyshoes.web.dao.TransactionDaoImpl;  import com.sportyshoes.web.model.Transaction;  @Service  public class TransactionServiceImpl implements TransactionService  {  @Autowired  private TransactionDaoImpl transactionDao;    @Override  public boolean addTransaction(Transaction transaction)  {  return transactionDao.addTransaction(transaction);  }  @Override  public List<Transaction> getAllTransactions()  {  return transactionDao.getAllTransactions();  }  @Override  public Transaction searchTransaction(int id)  {  return transactionDao.searchTransaction(id);  }  @Override  public void deleteTransaction(int id)  {  transactionDao.deleteTransaction(id);  }  } |

|  |
| --- |
| UserServiceImpl |
| package com.sportyshoes.web.service;  import java.util.List;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.sportyshoes.web.dao.UserDaoImpl;  import com.sportyshoes.web.model.User;  @Service  public class UserServiceImpl implements UserService  {  @Autowired  private UserDaoImpl userDao;    @Override  public boolean addUser(User user)  {  return userDao.addUser(user);  }  @Override  public List<User> getAllUsers()  {  return userDao.getAllUsers();  }  @Override  public User searchUser(int id)  {  return userDao.searchUser(id);  }  @Override  public boolean updateUser(int id, String password)  {  return userDao.updateUser(id, password);  }  @Override  public void deleteUser(int id)  {  userDao.deleteUser(id);  }    } |

# Screenshots

|  |
| --- |
| Admin Pages |
| Graphical user interface, text, application  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, text  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, text  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, application  Description automatically generatedGraphical user interface, text, application  Description automatically generated |

|  |
| --- |
| User pages |
| Graphical user interface, text, application  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, text, application  Description automatically generated |