…………………………………………………………… 装 订 线 ……………………………………………………………

**基于Struts+Spring+Hibernate**

**的网上书城系统**

**专 业： 计算机科学与技术**

**班 级： 2014级3班**

**学 号： 20143562**

**姓 名： 刘 磊**

**指导教师： 张继军**

**二〇一七年一月**

1. **JAVA的网络功能与编程**

1-1 JAVA语言简介

Java是一种简单易用、完全面向对象、具有平台无关性且安全可靠的主要面向Internet的开发工具。自从1995年正式问世以来，Java的快速发展已经让整个Web世界发生了翻天覆地的变化。在早期，Java比较多的用在浏览器上，插入到网页中(即是Java Applet程序)，成为最灵活、最强大的网页多媒体的载体，但由于Java虚拟机相对于个人电脑而言，有占用资源大，安全性相对较差等缺点，Applet逐渐的被后起之秀Flash所替代，但随着Java Servlet的推出，Java在电子商务方面开始崭露头角，最新的JSP(Java Server Page)技术的推出，更是让Java成为基于Web的应用程序的首选开发工具，目前的Java技术已成为所有大型电子商务项目的必然选择。

1-2 JAVA语言在网络上的应用

Java程序可以获取网络上结点的图象、声音、HTML文档及文本等资源，并可以对获得的资源进行处理。例如Java程序可以每隔一定时间读取某结点提供的最新数据，并以图表的形式显示出来。在编程处理上，一般先生成一个URL类型的对象，然后用 Java中相应的方法（method）获取该对象所代表的资源。

1. **SSH框架概述**

2-1 SSH框架的基本情况

SSH为Struts+Spring+Hibernate的一个集成框架，是目前较流行的一种Web应用程序开源框架。

集成SSH框架的系统从职责上分为四层：表示层、业务逻辑层、数据持久层和域模块层，以帮助开发人员在短期内搭建结构清晰、可复用性好、维护方便的Web应用程序。其中使用Struts作为系统的整体基础架构，负责MVC的分离，在Struts框架的模型部分，控制业务跳转，利用Hibernate框架对持久层提供支持，Spring做管理，管理Struts和Hibernate。具体做法是：用面向对象的分析方法根据需求提出一些模型，将这些模型实现为基本的Java对象，然后编写基本的DAO(Data Access Objects)接口，并给出Hibernate的DAO实现，采用Hibernate架构实现的DAO类来实现Java类与数据库之间的转换和访问，最后由Spring做管理，管理Struts和Hibernate。

2-2业务流程

系统的基本业务流程是： 在表示层中，首先通过JSP页面实现交互界面，负责接收请求(Request)和传送响应(Response)，然后Struts根据配置文件(struts-config.xml)将ActionServlet接收到的Request委派给相应的Action处理。在业务层中，管理服务组件的Spring IoC容器负责向Action提供业务模型(Model)组件和该组件的协作对象数据处理(DAO)组件完成业务逻辑，并提供事务处理、缓冲池等容器组件以提升系统性能和保证数据的完整性。而在持久层中，则依赖于Hibernate的对象化映射和数据库交互，处理DAO组件请求的数据，并返回处理结果。

采用上述开发模型，不仅实现了视图、控制器与模型的彻底分离，而且还实现了业务逻辑层与持久层的分离。这样无论前端如何变化，模型层只需很少的改动，并且数据库的变化也不会对前端有所影响，大大提高了系统的可复用性。而且由于不同层之间耦合度小，有利于团队成员并行工作，大大提高了开发效率。

1. **网上书城系统的详细设计**

3-1 项目基本架构

3-1-1 设计思想

1.该项目主要有对用户、图书以及订单信息的管理，以及用户购买图书和购物车的操作，注重实现以上几个功能。

2.前台有大量的动态数据使用，固多采用jsp页面，后台采用MVC模式对系统进行统一的管理，便于接口的实现和维护。

3-1-2 主要运用的知识点

1.Struts2的基本语法，包括拦截器，文件配置，上传下载等。

2.Hibernate的模板对数据库的操作。

3.Spring对Hibernate及Struts2的事务管理以及配置；

4.前端开发的一些知识。

3-2 需求分析

网上书城系统主要是实现网上选书、购书、产生订单等功能的系统。一个典型的网上商城一般都需要实现商品信息的动态提示、购物车管理、客户信息登录注册管理、订单处理等模块。

根据网上书城的基本需求，本系统需要完成的具体任务如下：

A.图书查询：当客户进入网上书店时，应该在主页面中分类显示最新的书目信息，畅销书目信息，以供客户选择所需图书，同时也应该提供按照图书名称，或者作者信息快速查询所需数目信息的功能。

B.购物车管理：当客户选择购买某图书产品时，应该能够将对应图书信息，如：价格、数量记录到对应的购物车中，并允许客户返回数目查询页面，选择其他商品，并添加到购物车中，当对应的购物订单生成后，应该能够自动清除以生成订单的购物车中的信息。

C.订单处理：对应客户购买图书商品西溪的需求，在确定了所购图书商品的价格、数量等信息后，提示用户选择对应的送货地址和联系方式，最终生成对应的订单记录，以便于网站配货人员一句订单信息进行后续的出货、送货的处理。

D.会员注册：为了能够实现图书商品的购买，需要管理客户相关的联系方式、送货地点等相关的信息。

E.商品管理：为了能够实现图书商品的管理，特价信息的修改等，添加管理员账户对图书、订单、优惠信息的管理。

3-3 开发环境

|  |  |  |
| --- | --- | --- |
| 项目 | 名称 | 版本 |
| 操作系统 | Microsoft Windows 10 Home Edition | 1607(x64) |
| 服务器 | Apache Tomcat | 9.0.0(x64) |
| JDK | - | 1.8.0.1110.14(x64) |
| SSH框架 | Struts2 | 2.3.20 |
| Hibernate | 5.2.3 |
| Spring | 4.3.4 |
| 数据库 | MySQL | 5.7(x86) |
| IDE | JetBrains IntillJ IDEA | Ultimate 2016.3.2(x64) |

3-4 总体设计

下表中给出了构成网上出城系统的各个组件，通过下表，可以从宏观上了解各个组件的功能。

|  |  |  |  |
| --- | --- | --- | --- |
| 用户表示层（视图） | 控制处理层（控制器） | | 业务逻辑层（模型） |
| ActionForm | Action |
| allBook.jso  allOeders.jsp  bottom.html  enroll.jsp  error.html  fisrtPage.jsp  index.jsp  left.jsp  login.jsp  managerLogin.jsp  oneType.jsp  personalInfomation.jsp  shoppingCart.jsp  singleBook.jsp  singleOrders.jsp  top.jsp  updatePassword.jsp  addBook.jsp  manageAllOrders.jsp  manageBook.jsp  manageLeft.jsp  manageWelcome.jsp  updateBook.jsp  userInfomation.jsp | BookManage  OrdersManage  PersonManage  ManagerLoginCheck  UserLoginCheck | AllbookAciton  AllOrdersAction  BookAction  DealOrdersAction  EnrollAction  LeftAction  LoginAction  OneTypeAciton  OrdersAction  OrdersManageAction  ShoppingCartAction  SingleBookAction  SingleOrdersAction  UpdateBookAction  UpdateCartAciton  UserExitAction  UserInfomationAction  UserManageAction | Bargain  Book  Manager  Orders  Ordersbook  Recommended  Sex  Type  User |

3-5 功能设计

1.图书信息管理：

该模块实现图书数目信息的分类显示，也是提供图书信息依据书名及作者信息快速搜索的功能。菜外，在用户选择了对应书目信息后，还可以显示出有关该图书的详细信息，以便于客户了解所购图书。

2.购物车管理：

用于委会每一个进入网上书店的客户对应的购物车。即将客户所选购的图书商品信息记录到对应的购物车中，以便于到收银台进行结账处理。

3.会员注册：

实现网上书店客户信息的注册及身份验证。由于目前网上商城普遍采用的方式为送货上门或者邮寄，因此需要收集与客户相关的联系方式、通讯地址等信息。

4.订单处理：

根据客户购物车中的信息以及客户所选择的送货方式和付款方式，连同客户对应的个人信息生成订单，以便于后续进行送货处理。

5.管理员管理：

管理员通过登录管理员账户对图书分类、图书信息、订单维护以及特价商品维护进行管理。

3-6 数据库的设计与实现

3-6-1 数据库的需求分析

依据网上书城的处理需求，对就数据表的设计及功能如下：

A.图书基本信息表：存放网上书城所销售图书的进本信息。

B.图书分类基本信息表：存放网上书城所提供图书分类信息。

C.客户基本信息表：存放客户的基本信息。

D.优惠信息表：存放图书的优惠价格。

E.管理员表：存放管理员登录的账号和密码。

F.订单表：存放用户订单信息。

G.订单图书表：存放订单中图书的信息。

H.需求图书表：存放订单所需要图书信息。

I.性别表：存放性别。

3-6-2创建数据表的脚本

A.图书基本信息表

|  |
| --- |
| CREATE TABLE `book` (  `bookId` int(11) NOT NULL AUTO\_INCREMENT,  `bookNumber` varchar(21) DEFAULT NULL,  `bookName` varchar(20) NOT NULL,  `bookAuthor` varchar(20) NOT NULL,  `bookPress` varchar(20) NOT NULL,  `bookPicture` varchar(100) NOT NULL,  `bookAmount` int(11) NOT NULL,  `typeId` int(11) NOT NULL,  `bookShelveTime` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  `bookPrice` double(10,2) NOT NULL,  `bookRemark` varchar(200) DEFAULT NULL,  `bookSales` int(11) NOT NULL,  PRIMARY KEY (`bookId`),  KEY `FK2E3AE91674045A` (`typeId`),  CONSTRAINT `FK2E3AE91674045A` FOREIGN KEY (`typeId`) REFERENCES `type` (`typeId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=gb2312 |

B.图书分类基本信息表

|  |
| --- |
| CREATE TABLE `type` (  `typeId` int(11) NOT NULL AUTO\_INCREMENT,  `typeName` varchar(16) NOT NULL,  PRIMARY KEY (`typeId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=gb2312 |

C.客户信息基本表

|  |
| --- |
| CREATE TABLE `user` (  `userId` int(11) NOT NULL AUTO\_INCREMENT,  `userName` varchar(16) NOT NULL,  `userPassword` varchar(12) NOT NULL,  `userEmail` varchar(100) NOT NULL,  `userNickname` varchar(10) DEFAULT NULL,  `sexId` int(11) NOT NULL,  `userAddress` varchar(200) DEFAULT NULL,  `userPhone` varchar(24) DEFAULT NULL,  `userRemark` varchar(200) DEFAULT NULL,  PRIMARY KEY (`userId`),  KEY `FK36EBCBAE0DAFBC` (`sexId`),  CONSTRAINT `FK36EBCBAE0DAFBC` FOREIGN KEY (`sexId`) REFERENCES `sex` (`sexId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=gb2312 |

D.优惠信息表

|  |
| --- |
| CREATE TABLE `bargain` (  `bargainId` int(11) NOT NULL AUTO\_INCREMENT,  `bookId` int(11) NOT NULL,  `bookNewPrice` double(10,2) NOT NULL,  PRIMARY KEY (`bargainId`),  KEY `FKEC1F8612F7272BF8` (`bookId`),  CONSTRAINT `FKEC1F8612F7272BF8` FOREIGN KEY (`bookId`) REFERENCES `book` (`bookId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=gb2312 |

E.管理员表

|  |
| --- |
| CREATE TABLE `manager` (  `managerId` int(11) NOT NULL AUTO\_INCREMENT,  `managerName` varchar(16) NOT NULL,  `managerPassword` varchar(12) NOT NULL,  PRIMARY KEY (`managerId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=gb2312 |

F.订单表

|  |
| --- |
| CREATE TABLE `orders` (  `ordersId` int(11) NOT NULL AUTO\_INCREMENT,  `userId` int(11) NOT NULL,  `ordersTime` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  `isDeal` char(1) NOT NULL,  `ordersNumber` varchar(21) DEFAULT NULL,  `totalMoney` double(10,2) NOT NULL,  PRIMARY KEY (`ordersId`),  KEY `FKC3DF62E517CFDD3C` (`userId`),  CONSTRAINT `FKC3DF62E517CFDD3C` FOREIGN KEY (`userId`) REFERENCES `user` (`userId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=19 DEFAULT CHARSET=gb2312 |

G.订单图书表

|  |
| --- |
| CREATE TABLE `ordersbook` (  `ordersBookId` int(11) NOT NULL AUTO\_INCREMENT,  `ordersId` int(11) NOT NULL,  `bookId` int(11) NOT NULL,  `bookAmount` int(11) NOT NULL,  PRIMARY KEY (`ordersBookId`),  KEY `FK2E27A34EF7272BF8` (`bookId`),  CONSTRAINT `FK2E27A34EF7272BF8` FOREIGN KEY (`bookId`) REFERENCES `book` (`bookId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=29 DEFAULT CHARSET=gb2312 |

H.需求图书表

|  |
| --- |
| CREATE TABLE `recommended` (  `recommendedId` int(11) NOT NULL AUTO\_INCREMENT,  `bookId` int(11) NOT NULL,  PRIMARY KEY (`recommendedId`),  KEY `FK55B4DE5BF7272BF8` (`bookId`),  CONSTRAINT `FK55B4DE5BF7272BF8` FOREIGN KEY (`bookId`) REFERENCES `book` (`bookId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=gb2312 |

I.性别表

|  |
| --- |
| CREATE TABLE `sex` (  `sexId` int(11) NOT NULL AUTO\_INCREMENT,  `sexType` varchar(4) NOT NULL,  PRIMARY KEY (`sexId`)  ) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=gb2312 |

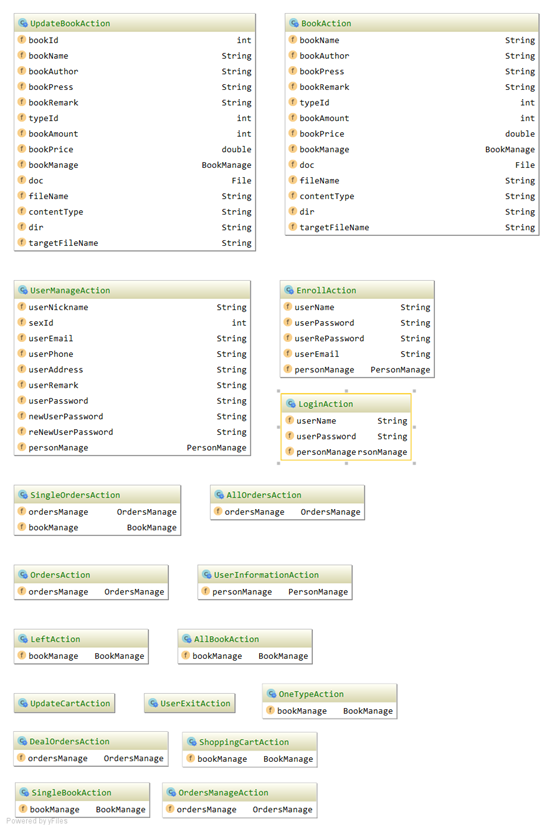
H.数据库内容项

详见附件。

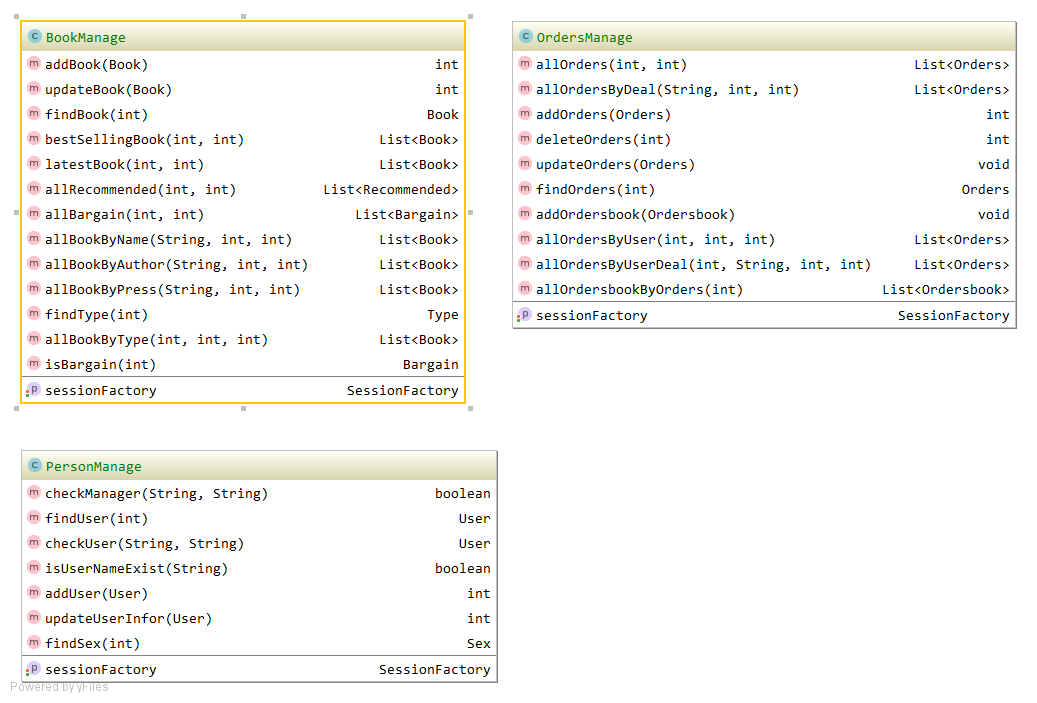
3-7 网上书城各个页面的功能

|  |  |
| --- | --- |
| 页面 | 功能 |
| firstPage.jsp | 包括四块，top.jsp,left.jsp,allBook.jsp,bottom.html，展示网站的主要内容。其中bottom.html包含网站的版权声明等信息，以下不做具体描述。 |
| top.jsp | 包含用户登录、用户注册、购物车、图书搜索等功能，点击后分别跳转到对应的页面 |
| left.jsp | 用于展示图书分类信息等内容 |
| login.jsp | 账号登录 |
| AllBook.jsp | 列出全部图书 |
| AllOrders.jsp | 列出全部订单 |
| Entoll.jsp | 用户注册 |
| Error.html | 出错信息页 |
| manangerLogin.jsp | 管理员登录页 |
| oneType.jsp | 单个类型书目列表 |
| personanlInformation.jsp | 个人信息详情页 |
| ShoppingCart.jsp | 购物车页 |
| singleBook.jsp | 单个书目详情页 |
| singleOreders.jsp | 单个订单表 |
| updatePassword.jsp | 更换登录密码页 |
| Manage/addBook.jsp | 添加书籍 |
| Manage/manageAllOrders.jsp | 管理全部订单 |
| Manage/manageBook.jsp | 管理具体某一本书目的属性 |
| Manage/manageLeft.jsp | 管理书目目录 |
| Manage/manageWelcome.jsp | 网上书城管理端欢迎页面 |
| Manage/oneType.jsp | 单个类型书目列表 |
| Manage/updateBook.jsp | 修改图书信息 |
| Manage/userInfomation.jsp | 个人信息详情页 |

3-8 Action的类图



3-9 DAO控制层的类图



3-10 Struts2配置文件struts.xml核心内容

|  |
| --- |
| <struts>  <package name="default" extends="struts-default">  <interceptors>  <interceptor name="userLoginCheck" class="com.bkos.interceptor.UserLoginCheck"/>  </interceptors>  <global-results>  <result name="error" type="redirect">/error.html</result>  </global-results>  <action name="enrollAction" class="com.bkos.action.EnrollAction">  <result name="input">/enroll.jsp</result>  <result name="success">/login.jsp</result>  </action>  下略...  <action name="bookAction" class="com.bkos.action.BookAction">  <interceptor-ref name="fileUpload">  <param name="allowedTypes">image/jpeg,image/gif,image/bmp,image/png</param>  <param name="maximumSize">20000</param>  </interceptor-ref>  <interceptor-ref name="defaultStack"/>  <result name="input">/manage/addBook.jsp</result>  <result name="success" type="redirect">/manage/addBook.jsp</result>  </action>  下略...  <interceptor-ref name="fileUpload">  <param name="allowedTypes">image/jpeg,image/gif,image/bmp,image/png</param>  <param name="maximumSize">20000</param>  </interceptor-ref>  <interceptor-ref name="defaultStack"/>  <result name="input">/manage/updateBook.jsp?bookId=${bookId}</result>  <result name="success" type="redirect">/manage/updateBook.jsp?bookId=${bookId}</result>  </action>  </package>  </struts> |

3-11 Hibernate配置文件hibernate.cfg.xml

|  |
| --- |
| <hibernate-configuration>  <session-factory>  <property name="connection.url">  jdbc:mysql://localhost:3306/bookstore?useUnicode=true&amp;characterEncoding=utf8  </property>  <property name="connection.driver\_class">com.mysql.jdbc.Driver</property>  <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>  <property name="connection.username">root</property>  <property name="connection.password">root</property>  <property name="hbm2ddl.auto">update</property>  <mapping class="com.bkos.entity.Bargain"/>  <mapping resource="com/bkos/entity/Bargain.hbm.xml"/>  下略...  </session-factory>  </hibernate-configuration> |

3-12 Spring核心配置文件applicationContext.xml

3-12-1 Spring的Hibernate事务管理配置

|  |
| --- |
| <bean id="sessionFactory"  class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">  <property name="dataSource">  <ref bean="candy"/>  </property>  <property name="hibernateProperties">  <props>  <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>  </props>  </property>  <property name="mappingResources">  <list>  <value>com/bkos/entity/Bargain.hbm.xml</value>  <value>com/bkos/entity/Book.hbm.xml</value>  <value>com/bkos/entity/Manager.hbm.xml</value>  <value>com/bkos/entity/Orders.hbm.xml</value>  <value>com/bkos/entity/Recommended.hbm.xml</value>  <value>com/bkos/entity/Sex.hbm.xml</value>  <value>com/bkos/entity/Type.hbm.xml</value>  <value>com/bkos/entity/User.hbm.xml</value>  <value>com/bkos/entity/Ordersbook.hbm.xml</value>  </list>  </property>  </bean> |

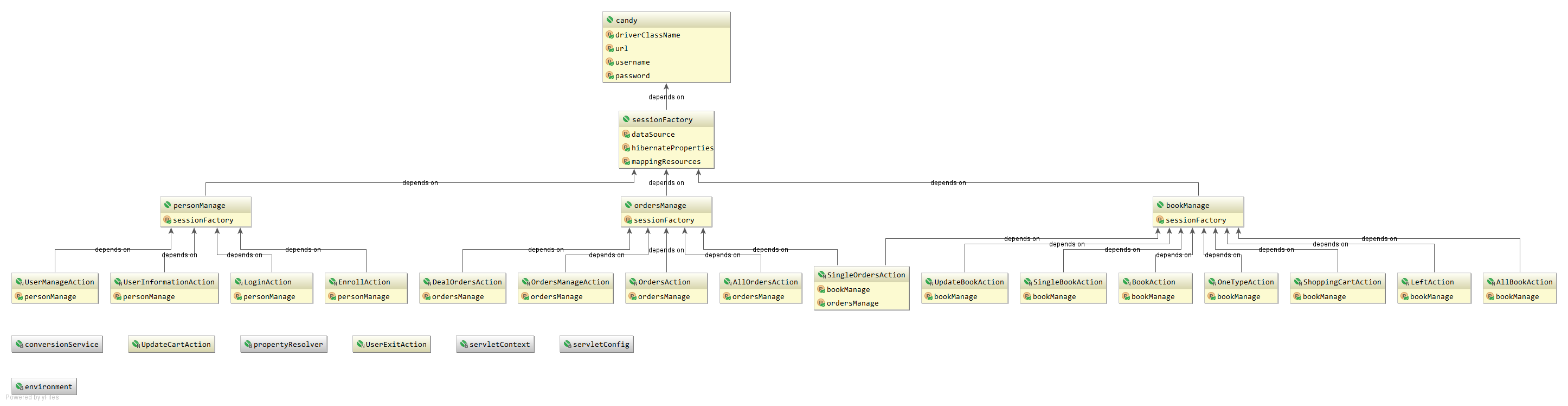
3-12-2 Spring对Struts2拦截器的配置

|  |
| --- |
| <bean id="personManage" class="com.bkos.dao.PersonManage" abstract="false"  lazy-init="default" autowire="default">  <property name="sessionFactory">  <ref local="sessionFactory"/>  </property> |

3-12-3 Spring对Struts2中Action的管理配置

|  |
| --- |
| <bean id="LoginAction" class="com.bkos.action.LoginAction" scope="prototype">  <property name="personManage">  <ref bean="personManage"/>  </property>  </bean>  下略... |

3-12-4 项目的Spring与各个DAO的映射关系



**第四章 设计中遇到的问题以及反思**

4-1 设计中遇到的问题

1.由于项目采用的框架JAR包比较多，包括第三方JAR包的存在，使得文件的框架过于冗杂。调试过程中大部分错误均出现再这一问题上。

2.未采用Hibernate中的C3P0包，在遇到较大的客户流量时会造成系统的卡顿甚至瘫痪。

3.对SSH框架掌握的不够透彻，特别是在配置Struts文件和Spring文件中，由于采用了注释配置和文件配置两种方法，导致配置文件的管理方式过度混乱，查错困难。

4.其他第三方JAVA库与使用版本的SSH框架难兼容，在使用较新版本的SSH架构是，部分老版本的方法过时，在现在的参考资料上难以与其匹配。

4-2改进与反思

1.对于包内容过于庞大问题，开始学习Ant、Maven、Gradle等Java包管理工具。

2.基于现阶段所学知识，继续深入学习SSH框架，对所学的知识熟练掌握。

3.写好注释以及相关出错抛出的文本，便于未来查错和再开发。