



杭州电子科技大学 2017-2018 学年第 1 学期

智能火车管理与查询系统

指导教师: 徐争前

小组成员:

15051516高伟

15051335汪品高

15051537徐明潘

15051706温昊然

15051306项雨

本系统由 HDU 软件系统架构小组成员搭建, 若转载请注明出处

2018 年 1 月 8 日

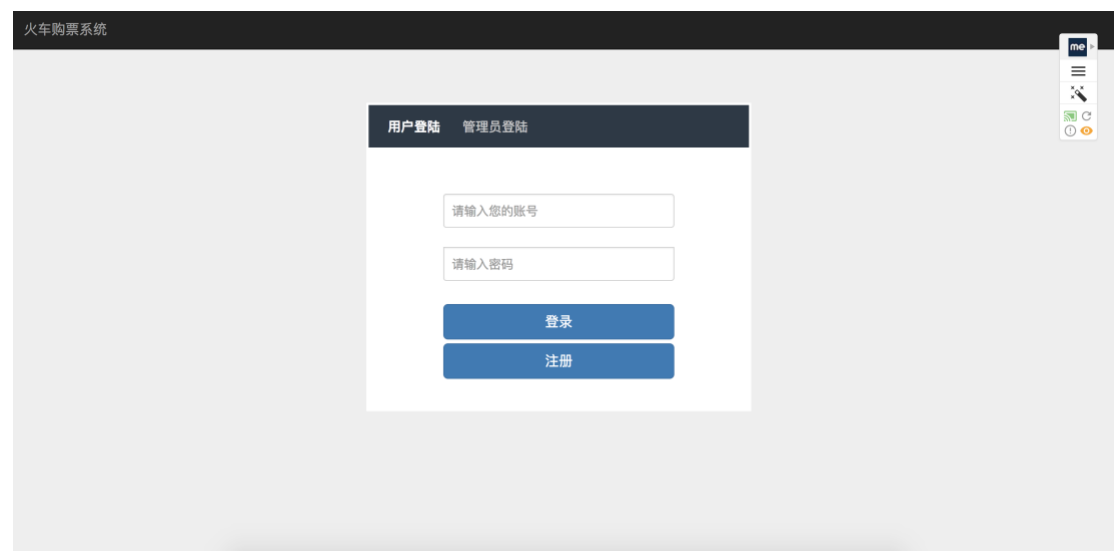
一、摘要

随着软件行业的快速发展,给我们的日常生活带来了非常多的益处，比如我们可以在网上进行订票、购物等。本次毕设就是为方便人们的出行即能够网上订购火车票所设计的，在满足火车票信息管理的基础上，建立火车票信息数据库，采用 MyEclipse 作为开发工具，以面向对象程序设计思想为基础，利用 UML 进行数据逻辑和物理模型设计。

JSP 是一门很优秀的编程语言，具有面向对象、与平台无关、安全、稳定和多线程等特点，所以用 JSP 编程为我们本次程序的编写提供了方便。另外，能够利用 Hibernate 框架连接到 mysql 数据库后，向数据库发送 SQL 语句实现程序与数据库中表间的消息通信，从而实现了相应的火车票管理的功能。

火车票订购系统由管理员和用户二部分组成。在本系统中，主要完成的对火车票信息的添加、修改、删除和查询以及对火车票订购的功能。

二、主要项目界面



火车票系统

me

站站搜索

车次搜索

起始站:

保定

终点站:

保定

立即搜索

火车票系统

车次:	例: D3832	起始站:	保定	终点站:	保定	过站详情
车次	类型	发站/到站	发/到时间	运行时间	参考票价	剩余票量
D101	动车	<div>北京</div> <div>沈阳</div>	13时 29分 24秒	4时 59分 24秒	特等座 ¥ 330	129 张
			16时 28分 46秒		一等座 ¥ 430	51 张
					二等座 ¥ 630	344 张
					硬卧 ¥ 730	334 张
					软卧 ¥ 530	63 张
D11	动车	<div>北京</div> <div>沈阳</div>	12时 22分 12秒	5时 1分 46秒	特等座 ¥ 330	298 张
			17时 24分 0秒		一等座 ¥ 430	326 张
					二等座 ¥ 630	149 张
					硬卧 ¥ 730	281 张
					软卧 ¥ 530	101 张
D17	动车	<div>北京</div>	9时 9分 0秒	4时 59分 24秒	特等座 ¥ 330	102 张
					一等座 ¥ 430	384 张
					二等座 ¥ 630	339 张

火车票系统

起始站:

终点站:

搜索

车次列表

添加车次

站点列表

添加站点

退出

序号	车次	类型	起始站	终点站	出发时间	到达时间	历时	操作
1	1148	普客火车	宝鸡	连云港	16时 0分 0秒	9时 31分 12秒	1	删除 编辑
2	1461	普客火车	北京	上海	11时 32分 24秒	7时 10分 12秒	1	删除 编辑
3	D101	动车	北京	哈尔滨	13时 29分 24秒	21时 23分 24秒	0	删除 编辑
4	D11	动车	北京	沈阳	12时 22分 12秒	17时 24分 0秒	0	删除 编辑
5	D17	动车	北京	沈阳	9时 9分 0秒	14时 8分 24秒	0	删除 编辑
6	D19	动车	北京	长春	14时 10分 48秒	20时 28分 48秒	0	删除 编辑
7	D44	动车	北京	大连	6时 33分 0秒	13时 4分 48秒	0	删除 编辑
8	D6657	动车	北京	保定	13时 0分 0秒	14时 34分 48秒	0	删除 编辑

火车购票系统

车次列表
添加车次
站点列表
添加站点
退出

添加车次信息
添加车次信息

车次:

类型:

起始站:

终点站:

行驶天数:




























出发时间:






















到达时间:

确认添加

三、项目结构

- > Deployment Descriptor: finaltest
- > JAX-WS Web Services
- > Java Resources
 - src
 - com
 - action
 - AddPassAction.java
 - AddTrainAction.java
 - DeletePassAction.java
 - DeleteTrainAction.java
 - LoginAction.java
 - RegistAction.java
 - ShowByStationAction.java
 - ShowPassAction.java
 - ShowPassByNumber.java
 - ShowTrainAction.java
 - StationAction.java
 - UpdatePassAction.java
 - UpdatePPass.java
 - UpdatePTrain.java
 - UpdateTrainAction.java
 - dao
 - Impl
 - AdminDAOImpl.java
 - IPassDAOImpl.java
 - MiddlePassDaoImpl.java
 - PassDAOImpl.java
 - TrainDAOImpl.java
 - AdminDAO.java
 - IMiddlePassDao.java

- >  IMiddlePassDao.java
- >  IPassDao.java
- >  PassDAO.java
- >  TrainDAO.java
- ✚ filter
 - >  UserFilter.java
- ✚ model
 - >  Admin.java
 - >  HibernateSessionFactory.java
 - >  MiddlePass.java
 - >  Pass.java
 - >  Train.java
 - >  Admin.hbm.xml
 - >  MiddlePass.hbm.xml
 - >  Pass.hbm.xml
 - >  Train.hbm.xml
- ✚ service
 - ✚ Impl
 - >  AdminServiceImpl.java
 - >  IPassServiceImpl.java
 - >  MiddlePassServiceImpl.java
 - >  PassServiceImpl.java
 - >  TrainServiceImpl.java
 - >  AdminService.java
 - >  IMiddlePassService.java
 - >  IPassService.java
 - >  PassService.java
 - >  TrainService.java
-  db.properties
-  hibernate.cfg.xml
-  struts.xml

- ✚ WebRoot
 - ✚ image
 -  始.png
 -  终.png
 -  huoche.png
 - ✚ META-INF
 -  MANIFEST.MF
 - ✚ script
 - ✚ laydate
 - ✚ theme
 - ✚ default
 - ✚ font
 -  iconfont.eot
 -  iconfont.svg
 -  iconfont.ttf
 -  iconfont.woff
 -  laydate.css
 - >  laydate.js
 - ✚ style
 - ✚ layui
 - >  css
 - >  font
 - >  images
 - >  lay
 - >  layui.all.js
 - >  layui.js
 -  add-train.css
 -  admin-update.css
 -  admin-user.css
 -  bootstrap.min.css
 -  login_2.css

- WEB-INF
 - lib
 - applicationContext.xml
 - web.xml
 - addpass.jsp
 - addtrain.jsp
 - changepass.jsp
 - error.jsp
 - input.jsp
 - login_2.jsp
 - login.jsp
 - querynumber.jsp
 - querystation.jsp
 - regist.jsp
 - search-by-number.jsp
 - search-result.jsp
 - showPasses.jsp
 - train-detail.jsp
 - update-page.jsp
 - updatePass.jsp
 - updateTrain.jsp

action 是跟前端页面实现交互的

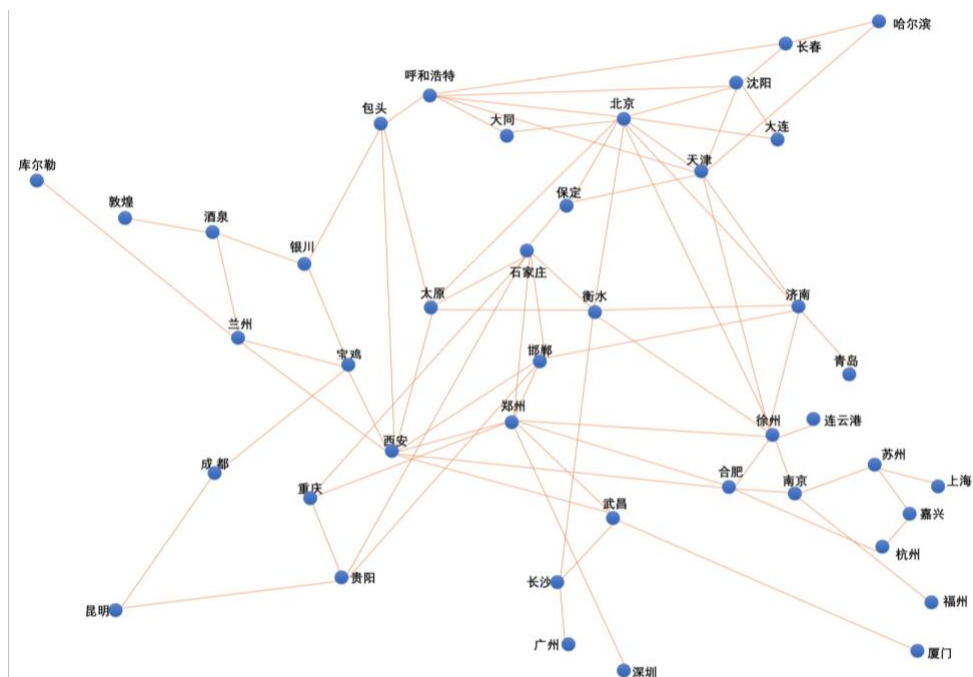
service 是中间层

dao 是数据层

model 是 javabean

filter 是过滤器，为了实现未登录无法查看其他页面

三、车次图



四、数据库结构

```
mysql> use train
Database changed
mysql> show tables
+-----+
| Tables_in_train |
+-----+
| admin            |
| seat             |
| station          |
| tnumber          |
+-----+
4 rows in set (0.01 sec)

mysql>
```

seat:

id	train_number	pseat	fseat	sseat	hsleeper	ssheeper
1	1148	300	400	600	700	500
2	1461	300	400	600	700	500
3	D101	300	400	600	700	500
4	D11	300	400	600	700	500
5	D17	300	400	600	700	500
6	D19	300	400	600	700	500
7	D44	300	400	600	700	500
8	D6657	300	400	600	700	500
9	D6752	300	400	600	700	500
10	K1025	300	400	600	700	500
11	K1118	300	400	600	700	500
12	K1177	300	400	600	700	500
13	K1306	300	400	600	700	500
14	K132	300	400	600	700	500
15	K1364	300	400	600	700	500
16	K1382	300	400	600	700	500
17	K1526	300	400	600	700	500
18	K1566	300	400	600	700	500
19	K1612	300	400	600	700	500
20	K1616	300	400	600	700	500

station:

id	train_number	train_station	train_length	start_time	end_time
1	1148	宝鸡	1	57600	0
2	1148	西安	2	66096	65520
3	1148	郑州	3	7704	7344
4	1148	徐州	4	23724	22320
5	1148	连云港	5	0	34272
6	1461	北京	1	41544	0
7	1461	济南	2	65736	65136
8	1461	徐州	3	81252	81108
9	1461	南京	4	11124	10800
10	1461	苏州	5	19980	19800
11	1461	上海	6	0	25812
12	D101	北京	1	48564	0
13	D101	沈阳	2	66816	66528
14	D101	长春	3	72972	72720
15	D101	哈尔滨	4	0	77004
16	D11	北京	1	44532	0
17	D11	沈阳	2	0	62640

tnumber:

id	train_number	train_type	start_station	end_station	days	start_time	end_time
1	1148	普客火车	宝鸡	连云港	1	57600	34272
2	1461	普客火车	北京	上海	1	41544	25812
3	D101	动车	北京	哈尔滨	0	48564	77004
4	D11	动车	北京	沈阳	0	44532	62640
5	D17	动车	北京	沈阳	0	32940	50904
6	D19	动车	北京	长春	0	51048	73728
7	D44	动车	北京	大连	0	23580	47088
8	D6657	动车	北京	保定	0	46800	52488
9	D6752	动车	包头	呼和浩特	0	28800	33300
10	K1025	快速火车	兰州	青岛	2	84816	26918
11	K1118	快速火车	北京	包头	1	45072	21672
12	K1177	快速火车	北京	银川	1	48060	21816

五、相关技术

下面是相关技术特点及内容介绍：

(1) J2EE 相关知识介绍

J2EE 是一种基于 Java 组件技术的系统开发规范，它是由 Sun 公司推出的一种企业应用，不同厂商都是以此为标准创建平台产品的，这使的不同厂商生产的产品之间的无障碍交互成为可能。

J2EE 规范提供了一个多层次企业级信息系统的标准平台，这个平台是基于组件技术，目的是规范和简化企业级应用系统的部署和开发。它不仅能够在因特网的应用中保护数据的安全模式，而且能够方便的存取数据库的 JDBC API 等等，

同时它还能够对很多技术提供全面的支持, 比如 XML、JSP(Java Serv-er Page)、Java Servlets API 以及 EJB(Enterprise JavaBean)技术, 它最终的目标是成为一个能够让企业开发者的投放市场时间大幅缩短的优化体系结构。J2EE 组成技术一般分为三层或多层的体系架构,这已经成为了开发信息系统的首要选择。最典型也是最经典的 J2EE 体系一般分为三层, 即持久层、业务逻辑层和表现层。

持久层 (Persistence Layer) 主要是负责实体对象持久化的, 也就是对数据库进行各种操作的层次, 在系统设计中我们一般用 Dao 层表示。

业务逻辑层是决定架构是否成功的关键 ,它的主要功能是给表现层提供业务逻辑使用, 它容纳了很多的接口。在一般情况下, 一个好的业务逻辑层主要具备以下几个特点: 面向对象设计, 独立于表现层存在, 容易管理和测试, 对事物能更好的处理, 功能完善,用户使用较简单,同时有非常好的 OO 实施等等。业务层主要是通过以下技术实现的 :EJB, POJO 等等。在系统设计中我们一般用 Service 层表示。

表现层 (Presentation Layer) 主要是实现显示功能的, 它是远程调用的门面或者是客户端界面, 它主要用到的技术有 : MVC 框架技术,SERVLET 和页面技术 JSP, 其三层结构和各层之间的关系, 如图 1-1 所示

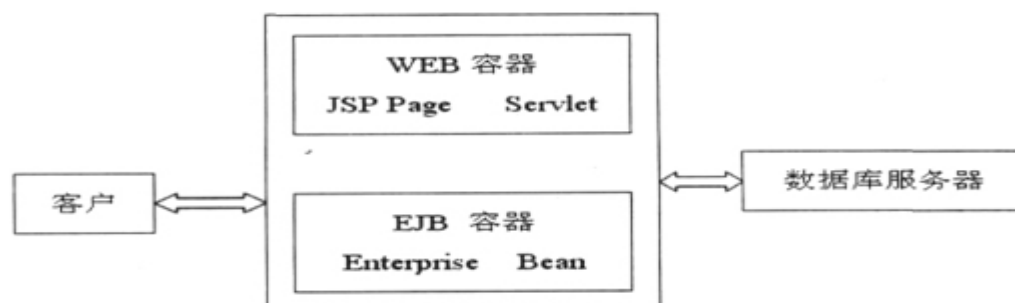


图 1-1 三层体系结构

(2) Struts 和 MVC 的共性和区别

Struts 和 MVC 相互联系, 共存发展, 但又有各自的特点。MVC 应用框架最典型的是 Struts,它使得系统的表示层开发更加简化,更加利于重用,同时还提供了效率。而 MVC(Model/View/Controller)模式是目前广泛流行的一种设计模式, 主要包括 3 类对象。分别为 Model、View、Controller。下面分别对其介绍 :Model 是一种应用对象, View 是显示在屏幕上, Controller 规定了用户的输入界面的响应方式。

在 MVC 模式中，担任模型的角色的是 JavaBean，它在 JSP 和 Servlet 之间的交互起到了桥梁的作用，在完成 Servlet 处理后，就设置实体相对应的属性，JSP 读取所设置的实体属性后进行显示。在该模式更加清晰的将显示与逻辑分离开来，使得代码比模式(JSP+JavaBean)更容易理解了，适合于设计和开发大型的项目。

而 Struts，它是 Apache 的一个组织项目，和 Apache 所组织的其它类型的项目一致。

Struts 是开放源代码的一个项目，它是一个非常优质的 MVC 框架，它把用作实现的部分应 Servlet 和 JSP 标记(属于 J2EE 规范)。MVC 的各项特性基本都被 Struts 继承了，并依据 J2EE 的特征做出了相应的扩展和更新。Struts 的工作原理如图 1-2 所示。

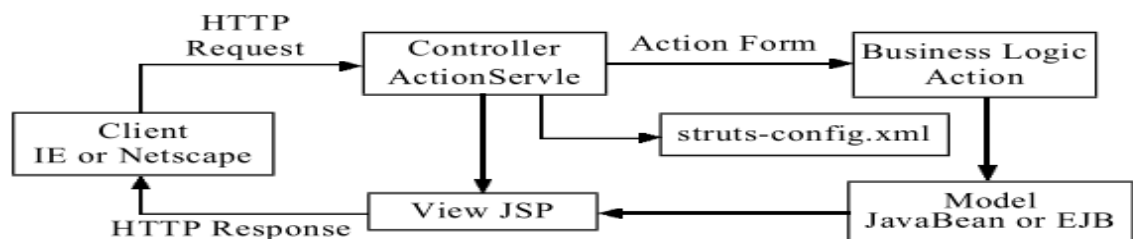


图 1-2 Struts 工作原理

下面按照层次进行详细介绍：第一，在 Struts 控制中，由 ActionServlet 类来充当控制器。ActionServlet 是一个非常通用的控制组件，它提供了很多入口点，主要用来处理发送到 Struts 的所有 HTTP 请求。一般都是依据配置文件 struts-config.xml 的配置信息，把收到的请求转发给合适的 Action 对象，配置信息一般为以下形式：
`<action name="login" method="login" class="com.web.action.AdministratorAction">
 <result name="success" type="chain">afterLogin</result>
 <result name="error">/adminlogin.jsp</result>
</action>`。如果这个 Action 对象不存在，则会先创建一个 Action 对象。第二，在 Struts 视图中，页面的输出主要是由 JSP 控制的。它将接收到的数据（来自 ActionForm），应用 logic, taglib, html, bean 等来显示这些数据。第三，在 Struts 模型中，最主要的实体类有三种，分别为：JavaBean or EJB、ActionForm、Action。ActionForm 主要是用来封装客户的请求信，而 Action 是用来获取 Action-Form 中的数据的，之后再交由 JavaBean 或者 EJB 对数据进行处理。

(3) 对数据库用到的 Hibernate 框架介绍

在之前做项目时，对数据库进行连接和访问操作时，并未使用过 Hibernate 框架，都是用的 JDBC、数据池等。在接触 Hibernate 后，感觉它更加简单，不要编写繁杂的 SQL 语句，利用配置文件自动识别数据库的名称等，不用再手动设置数据库驱动、数据库连接的用户名等，它主要利用到了对象关系的映射文件。Hibernate 由一个开放的源代码构成的，它能够对 JDBC 进行有效的对象封装，这样可以使程序员更加随心所欲地对数据库进行控制，而这一切是基于编程思想的。

在之前没有 Hibernate 框架的应用时，java 语言数据库的主要链接口是 JDBC，而且在当时来看，它是具有比较好性能和比较高的效率的。然而，随着数据量的不断增加和应用系统的不断扩展，数据和数据表之间的关系越来越变的复杂，同时使得 JDBC 操作数据库的代码变得非常复杂，使得整个系统的性能越来越低。而 Hibernate 的出现，解决了这一大难题，它利用对象关系映射的方式大大的简化了数据库的操作，提高了系统的性能。另外，程序对 Hibernate API 进行访问也主要通过持久化对象，这使得对 Hibernate 所用到的数据库资源有了更加有效的配置。

Hibernate 不仅为程序员提供高效的数据查询，还提供了 JAVA 类与数据表之间的有效映射。它相对于手动操作数据库的技术，比如 JDBC 或 SQL 等，更具有一定的自动化，这也是我在学会应用 Hibernate 之后最大的感受，而且有效减少数据库操作的工作量。

除此之外，因为 Hibernate 工作时主要使用了代理模式进行，因而有效的简化了加载类的过程和时间，而且减少了代码提取时所需的编写量，最终使得程序开发的效率和时间得到了大大的提升。目前看来，Web 服务器和 Hibernate 框架已经非常成功实现了良好的集成与融合，而且大部分数据库服务器都是支持的，因此这就使它具有了一定的通用性。

综上所述，Hibernate 为我们的应用程序数据库设计提供了多种运行方式，而且在一些轻量级系统中，使得应用程序可以实现直接与 JDBC 相连接，并进一步执行事务的管理。在全面优化应用程序的体系过程中，处于底层的所有 JDBC 都会被抽象化。然而对于 Hibernate，它能够自己规定查询语言（HQL），同时还能对最初的 SQL 语法兼容，所以它拥有非常高的效率、非常强的灵活性，这也是它能够与多种 Web 服务器集成，而且支持大部分数据库服务器的主要原因。

既然 Hibernate 在数据库操作方面有这么多一处，那么我们要利用 Hibernate

来设计开发应用，提高程序的性能和效率以及为编程人员省去很多麻烦，就要对 Hibernate 有效配置有非常深刻的认识和非常熟悉它的基本配置。以下是对 Hibernate 有效配置的方法做一个简单介绍：

首先是映射文档的配置工作，它的主要作用是有效定义持久化对象，所以对于以 class_name 来命名的文件或者任何一个持久化对象来说，创建映射文件是至关重要的一步。如果它的设置是非常正确无误的，那么 Hibernate 持久化服务就能够正常有效的运行，而且系统数据库的性能也会非常好；但是如果它的设置非常糟糕。那么 Hibernate 持久化服务就不能正常运行，而且数据库性能也会非常低。数据库和持久化类之间的关系是通过映射文档来体现的，它主要包括相关字段和映射字段。另外，它还可以表明持久化类之间的关系，比如：乘客和火车票之间存在着一对多或多对一的关系，由此我们便可以得出关系型数据库表间的关系。

之后，对 Hibernate 框架进行基本的配置工作。创建命名为 hibernate.cfg.xml 的配置文件，位于 src 目录下，并且要把它路径配置到实体类的路径中，最后完善对 H 配置文件的配置工作。Src 目录下的配置文件主要由三个方面组成，包括有特定数据库的信息、连接数据库以及数据库的生成模式等。所以，当程序员要使数据库内的变化与应用当中一致时，就可对配置文件内所设置的属性值进行相应的修改。下面对这三个方面进行详细介绍：数据库模式的生成，它主要是通过 Hibernate 来实现，它所要执行的详细命令主要是：Java-cpclasspath net.sf.hibernate.t001.hbm2ddoptions mapping_files，这条命令可以有效提供合适恰当的数值，并根据这些数值生成的模式；

对数据库的操作，主要是依据 Hibernate 完成数据库的连接，而且能够同时提供对外接口。每当 Hibernate 获取核心接口（SessionFactory）后，就能进行有效的会话（session），并进行删除（delete）、加载（load）、查询（query）、插入（insert）等操作。

任何代码的运行都有其执行的顺序，有可能是程序员所设定的逻辑顺序，还可能是其它，那么 Hibernate 代码在初始化运行时，所执行的步骤是怎样的呢？接下来将详细介绍，来更加深入的了解 Hibernate 的工作流程。

首先，在启动时应该对 net.sf.hibernate 进行获取，用到了 net.sf.hibernate.cfg 对象；然后，调用会话工厂（SessionFactory）对象上的 openSession（）方法，打开 net.sf.hibernate.Session 会话；之后，当有些对象需要保存时，应该关闭会话

(session)；最后，要开启 net.sf.hibernate 时，程序员就能够利用 Hibernate 的核心接口对象上的 OpenSessionInView 的方法打开。值得注意的是，在对数据库报表进行编程时，要考虑到程序运行的科学性以及性能，包括：数据字典中的数据不得随便修改；在对数据进行更新时，要保证系统的数据逻辑不会发生变化；设计报表涉及非常多的数据，保障程序的运行效率是非常重要的，所以程序的逻辑必定要符合科学性；对标准程序进行动态使用可以用附加技术实现，从而避免影响对标准系统的升级。当你选择的数据表编程是有效的时候，就要进一步想到程序的科学性与逻辑性，而且要学会换位思考一些问题，进而找出更优良的解决方案。

在数据持久化对象中，Hibernate 只是其中的一种，而且它还是一个轻量级的映射框架。在程序设计中，数据库用到 Hibernate 数据持久层设计会有很多优势，主要有：第一，它能有效的改善数据访问的效率；第二，因为它属于轻量级映射框架，而且有开放性的特点。程序员在进行开发时，可以根据自身的需求修改代码，拥有非常强的灵活性；第三，应用它进行程序设计，可以有效减少 SQL 语句的用量，从而减少了代码的数量，使系统的性能得到明显的提高。

总之，它可以有效地促进 java 对象模型的发展，程序人员可以使用 Hibernate 的数据持久层的解决方法，不用对数据库的逻辑关系做过多考虑，进而有效的使业务逻辑更加清晰并且实现，使模块的划分更加地科学化、合理化。

六、总结

总的来说这次学习 jsp、Struts 框架等是非常有用的。在这次的学习我把以前学过的知识运用在了实际操作上，使我真正懂得了怎么去运用它。通过毕设使我进一步掌握 jsp 设计的有关知识，提高了运用 jsp 的能力，加深了对它的相关知识的理解。通过自己去查阅资料、阅读程序及动手编写程序提高了我们的设计能力，使我们在设计上有一定飞跃，能用 jsp 的功能来解决一些实际问题。

在做项目设计和程序编写的时间里，从理论到实践，遇到的问题数不胜数，不过从中学到很多很多东西，不仅可以巩固了大学所学过的知识，还提高了解决问题的能力。通过这次完完整整由小组成员完成项目的经验，使我再次懂得了理论与实践相结合的重要性，只有理论知识是远远不够的，只有把所学的理论知

与实践相结合，从理论中得出结论，从实践中得到验证，才能真正提高自己的实际动手能力和独立思考的能力。

课程设计是对我们自身的一种测试，也可能是一种挑战，通过大学三年的学习，你可以做出什么？通过这次经验，相信每个人都能给出自己的答案。因为这是一次考验我们自身实践能力的机会，一次把学习到的知识付诸实践的机会，所以我们必须全力以赴。我们要学的是知识，是解决问题的方法和思想，要端正学习态度，就是在这样的过程中我们才能不断提高。通过这次的经验，我相信在以后的工作和学习中，可以做到一步一个脚印，踏踏实实的走好每一步，相信我们的人生会更加精彩！

从总体上来说，这次的大作业对我们来说是非常难得的一次锻炼的机会，可以使所学知识融会贯通。对于我来说，这次的大作业，最大的收获不是对工具的使用更加熟练，不是对编程语言的熟练运用，而是对整体框架的掌握和解决问题的能力的提升。因为很多问题要我们小组成员一起面对，很多问题要想尽各种办法解决，克服困难，在解决这些困难的过程中提高了我们的解决能力和实践能力。