类别： 软（硬）件开发类

**南京城市职业学院**

**毕业作业**

“Memories”青年旅社的

设计与实现

|  |  |
| --- | --- |
| **姓 名：** | **童洁** |
| **学 号：** | **16302017** |
| **年 级：** | **2016级** |
| **专 业：** | **计算机应用技术** |
| **学生类别：** | **三年高职** |
| **指导教师：** | **李霞** |
| **教学单位：** | **工程与信息学院** |

2019年03月 20日

**南京城市职业学院**

**毕业作业（设计、论文）诚信承诺书**

本人郑重承诺：

1、本毕业作业（设计、论文）是在指导教师的指导下，查阅相关文献，进行分析研究，独立自主完成。

2、本毕业作业（设计、论文）中，所有实验、数据和有关材料均是真实的。

3、本毕业作业（设计、论文）中，除引文和致谢内容外，不包含其他人或机构已经撰写发表过的研究成果。

4、本毕业作业（设计、论文）中如有剽窃他人研究成果的情况，一切后果自负。

5、本毕业作业（设计、论文）中所取得的成果归学校所有。

学生（签名）： 童洁

2019年03月20日

摘 要

在快速的物质化生活下，人们对于精神上的需求逐渐增加，而旅游可以带来富足的精神需要，逐渐成为人民生活中不可或缺的组成部分。本次网站是关于青年旅社的旅游信息网站。采用了JSP+MySQL技术创建，以Java作为编程语言，以MySQL作为后台数据库。本网站为旅游爱好者提供景点背景、酒店住宿等服务，给您带来便捷服务的同时也能提高旅游业的发展。

从开发环境、技术简介、系统需求分析与设计、系统实现、系统测试这五大模块来具体阐述我的平台。

关键词：旅游 Java MySQL

**目 录**

[第一章 绪 论 5](#_Toc4250545)

[1.1引言 5](#_Toc4250546)

[1.2项目背景 5](#_Toc4250547)

[1.3项目目标 6](#_Toc4250548)

[1.4项目意义 6](#_Toc4250549)

[第二章 开发环境与技术简介 7](#_Toc4250550)

[2.1系统开发环境 7](#_Toc4250551)

[2.2技术介绍 7](#_Toc4250552)

[2.3可行性分析 8](#_Toc4250553)

[2.3.1技术可行性 8](#_Toc4250554)

[2.3.2资源可行性 8](#_Toc4250555)

[2.3.3经济可行性 8](#_Toc4250556)

[第三章 系统需求分析与设计 9](#_Toc4250557)

[3.1 系统的需求分析 9](#_Toc4250558)

[3.1.1 功能需求 9](#_Toc4250559)

[3.1.2 非功能需求 9](#_Toc4250560)

[3.2系统架构设计 10](#_Toc4250561)

[3.3系统数据库设计 10](#_Toc4250562)

[第四章 系统的实现 15](#_Toc4250563)

[4.2.1 用户注册模块的详细设计 15](#_Toc4250564)

[4.2.2 用户登录模块的详细设计 19](#_Toc4250565)

[4.2.3 预约模块的详细设计 22](#_Toc4250566)

[4.3 系统其它模块的详细设计 25](#_Toc4250567)

[第五章 系统测试 28](#_Toc4250568)

[5.1测试环境和方法 28](#_Toc4250569)

[5.2测试用例设计 28](#_Toc4250570)

[5.3测试问题及解决方法 29](#_Toc4250571)

[结束语 29](#_Toc4250572)

[致 谢 30](#_Toc4250573)

[参考文献 31](#_Toc4250574)

# 第一章 绪 论

## 1.1引言

青年旅舍在西方国家已经经历了近80多年的发展，已经成为世界上最大的旅游住宿连锁组织之一，作为一种“安全、经济、卫生、隐私”的新型旅舍，虽然国外青年旅舍已兴起近百年之久，青年旅舍的发展相对于国内是相当成熟了，在对青年旅舍的市场定位、市场空间的研究也是很深入，将着力点放在青年旅舍的环境、可持续发展、特色化管理以及完善创新等方面的研究，我国可以借鉴参考。

随着近年来我国青少年旅游人数的不断增多，我国引入了国际上久负盛名的国际青年旅馆，针对此现象，我们应该充分考虑青年旅馆对中国现行发展的特殊意义，提出相应的举措，以促使中国青年旅舍走向良性、快速发展道路。

## 1.2项目背景

随着人民生活水平日益提高，人民需求已从基本的物质需求，逐渐增加精神需求，如艺术需求，以及对世界上其他优美的地方的追求，对世界上其他地区的人文景观，风土人情，自然风光的好奇。都是当今人类生活所不可或缺的一部分。纵观互联网的发展，电子商务的兴起，使得传统商业模式发生了天翻地覆的变化，而旅游业也随着科技的发展，从以前的实体旅行社，逐渐发展成为一个拥有在线系统的，可以满足更多用户需求同时也可减少更多人力开支的在线旅游网站。小型旅游网站就是一个为用户提供信息的在线旅行社，用户可以在网站中浏览景点信息，景点周边的酒店、宾馆信息，浏览网站中提供的旅游线路，如果是注册会员还可预订各条旅游线路。整个网站相当于一家旅行社，以组织旅行团旅游，收取旅行线路费用，介绍景点周边酒店和宾馆信息，收取广告费为主要盈利模式。所有操作都可以由用户在客户端浏览器中完成，而服务器端程序会按用户的要求来完成对系统数据的操作，并将结果传给Web服务器，再由Web服务器处理成HTML文件后发送到客户端浏览器。这就是所谓的B/S结构应用系统，B/S结构即Browser/Server(浏览器/服务器)结构，是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。

## 1.3项目目标

该网站使用B/S结构，主要实现在服务器端进行旅游景点的展示，旅游景点周边的食宿信息展示，旅游网站推出的旅游路线线的展示，路线详细信息的查询、预订等功能。

## 1.4项目意义

1998年青年旅舍进入中国大陆，得到了广大学者和业内人士的关注。青年旅舍在国外兴起已近百年，如今青年旅舍的服务对象已不再仅仅局限于青年人，众多认同环保、经济、卫生的背包客们就将散布各地的青年旅舍作为自己旅途中移动的家。青年旅舍的出现，不仅有力地启动和丰富了中国旅游市场，为家庭旅游带来更多的选择，同时让中国旅游业发展到世界，很多国际背包客都是中国青年旅舍的常客，为中国新兴的青年旅舍而来。由于青年旅舍的交流、自助、实惠等优势，一经创建就受到了社会大众的关注和热捧，特别受到国际背包客的欢迎，从而有力地带动了入境游市场。

有关人士指出，青年旅舍在我国的兴起发展，一方面说明我国旅游业日趋成熟的大好前景，另一方面说明多元化的住宿设施被迫切适应于多元化的旅游需求，从而吸引着更多的人都有兴趣和条件参与到这项活动中去，培育更多更为广泛的旅游市场。随着青年旅舍网络化的进一步形成和完善，以及家庭旅游、老年旅游等多元化群体的加入，中国将会成为拥有最多青年旅舍的国家。

# 第二章 开发环境与技术简介

## 2.1系统开发环境

硬件环境：

华硕笔记本。

软件环境：

1、MySQL 8.0.12

Mysql是最受欢迎的开源数据库管理系统，它有快捷、可靠、易于使用的优点。

2、Navicat Premium 12

Navicat Premium数据库管理工具64位版是一个可多重连接的数据库管理工具，Navicat 支持以单一程序同時连接到目前世面上所有版本的主流数据库并进行管理和操作，支持的数据库有： MySQL、SQL Server、SQLite、Oracle 及 PostgreSQL。让管理不同类型的数据库更加方便。

3、1.8Jdk

J2SE的Jdk是任何java系统都必不可少的工具包。

4、Eclipse

Eclipse是最流行的java集成开发工具。一经出现，就收到了广大的欢迎，它能很好的与mysql进行集成。

## 2.2技术介绍

Spring、Spring MVC、MyBatis

Spring

Spring是一个开源框架，它由[Rod Johnson](https://baike.baidu.com/item/Rod%20Johnson)创建。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。

Spring MVC

典型教科书式的MVC框架，适用于初学者了解mvc，简化了web开发，将Web层进行职责解耦。

MyBatis

MyBatis 是一款优秀的持久层框架，它支持高级映射和存储过程。避免了几乎所有的 JDBC 代码和手动设置数据库参数。可以使用简单的 XML 代码或注解来配置和映射信息，将接口和Java对象映射成数据库中的记录。

## 2.3可行性分析

从目前的技术、资源和经济方面都满足开发条件。

### 2.3.1技术可行性

可如下表：

表1 技术可行性分析

|  |  |  |  |
| --- | --- | --- | --- |
| 关键技术项 | 要求 | 现有水平 | 对照分析 |
| Java语言 | 熟悉 | 熟悉 | 可以满足要求 |
| MySQL数据库 | 熟练 | 熟练 | 可以满足需求 |

### 2.3.2资源可行性

开发人员：本人

软件资源：mysql、navicat、eclipse 都是开源的软件

时间资源：有半年左右的开发时间，时间充足。

### 2.3.3经济可行性

|  |  |  |  |
| --- | --- | --- | --- |
| 所需的软件或硬件 | 要求 | 现有水平 | 对照分析 |
| 电脑 | 一台 | 已拥有 | 满足要求 |
| MySQL数据库 | 能够存储数据 | 已拥有 | 满足需求 |
| Eclipse | 能够进行代码编写 | 已拥有 | 满足需求 |

# 第三章 系统需求分析与设计

给出系统/网站的总体描述

## 3.1 系统的需求分析

### 3.1.1 功能需求

本网站是以展示旅游信息、旅游线路、景区图片和景点周边酒店、宾馆信息等功能为核心的小型旅游网站。通过联系景区周边的饭馆和酒店，提供旅游景区居住、餐饮等信息，为其做广告，给用户带来方便的同时也为本网站创造利益的网站。

表3-1 功能需求描述表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **需求编号** | **需求描述** | **类型** | **优先级** | **版本** | **备 注** |
| S001 | 旅游景点浏览 | 所有 | 一般 | V1.0 |  |
| S002 | 旅游景点详细信息 | 所有 | 一般 | V1.0 |  |
| S003 | 旅游路线浏览 | 所有 | 优先 | V1.0 |  |
| S004 | 旅游路线详细信息 | 所有 | 最后 | V1.0 |  |
| S005 | 旅游路线预约 | 所有 | 一般 | V1.0 |  |
| S006 | 用户可以分别查询国内的景点或国外的景点 | 所有 | 一般 | V1.0 |  |
| S007 | 酒店浏览页面 | 所有 | 一般 | V1.0 |  |
| S008 | 酒店详细信息 | 所有 | 一般 | V1.0 | 管理员 |
| S009 | 友情链接 | 所有 | 一般 | V1.0 |  |

### 3.1.2 非功能需求

1.性能

采用专用服务器存放数据库，以保证不断增长的数据需求。

2.安全性

要求保证数据的安全，采用Windows身份认证方式登录数据库，系统合法用户才能使用本系统。

3.易用性

在系统完成后的部署过程中，给出部署说明文件，方便用户使用

4.可扩展性

系统开发过程中，注意保证链接位的预留，以方便今后新功能的加入，同时每个功能以模块方式处理，预留足够的冗余空间，方便平台的扩展。

5.可重用性

本系统应该可以借鉴给其他相同或类似的师资管理平台，因此在开发系统中要考虑到可重用性，即以文件方式出现的地方多些，以纯HTML文档出现的少些。

## 3.2系统架构设计

给出系统用户权限表（要有一定的文字说明），可如下表：

表3-2 用户权限分析表

|  |  |
| --- | --- |
| **人员类型** | **权限描述** |
| 管理员 | 新增、查看、修改、删除用户  新增、查看、修改、删除旅社信息 |
| 普通用户 | 查询旅社信息  修改登录用户个人信息 |

## 3.3系统数据库设计

在数据库中建表时，根据ER模型与建表规则，对各个实体实现建表。对于1:N联系，在游客表中加入登记者ID和参加的旅行团外键；在旅行团表中加入选择的旅游线路外键；对于M:N联系，将旅游线路和景点的关系之间进行单独建表，将景点和酒店的关系之间进行单独建表，将景点和宾馆的关系之间进行单独建表。

按如上方法，在数据库中共建立总计10个表，这10个表分别是：user表、tourist表、tourgroup表、tourroute表、scenicspot表、restaurant表、hotel表这七个实体表，以及负责说明实体间M:N的联系，减少实体表之间的冗余，三个实体之间的关系表，旅展示游线路和景点之间的关系的spotofroute表、展示旅游线路和景点之间的关系的restaurantofspot表和展示景点和宾馆之间的关系的hotelofspot表。这10个表的介绍如下。

User表用于存放系统所有注册用户的详细信息，包含用户名、密码、注册时间、用户的email地址等内容，结构如表3.1所示。表中UserID为主键并被设置为自增，这样当插入一条记录且id字段的数据为空（null）时，新记录的id值将由系统自动给出，且给出的值将是表中曾经存在的最大的id值加1（若是一张新表，则从1开始）。这样可以保证整个表中的id字段在其数据类型允许的范围之内没有重复的值。注册时间设置为自动填充，默认值为now()，这样将会让数据库系统对用户的创建时间进行自动记录。这样，UserID以及用户的注册时间在用户注册时不由用户填充，是由数据库自动生成，使得用户在使用注册功能是更加方便，更加人性化。

**表3.1 user表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| UserID | INT |  | 是 | 用户ID | 主键 |
| UserName | VARCHAR | 20 | 是 | 用户名 |  |
| Password | VARCHAR | 20 | 是 | 密码 |  |
| Email | VARCHAR | 45 | 是 | 用户邮箱 |  |
| UserTime | TIMESTAMP |  | 是 | 注册时间 |  |

Tourist表用于存放用户进行登记的游客信息，包含游客ID、身份证号、姓名、性别、联系电话、登记者的ID、参加的旅行团ID等内容，如表3.2所示。其中TouristID为自动填充的自增整型数据。登记者的ID为外键，对应user表中的UserID，参加的旅行团ID为外键，对应tourgroup表中的TourgroupID。

**表3.2 tourist表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| TouristID | INT |  | 是 | 游客ID | 主键 |
| UserID | INT |  | 是 | 登记者ID | 外键 |
| TourgroupID | INT |  | 是 | 旅行团ID | 外键 |
| IDNumber | VARCHAR | 20 | 是 | 身份证号 |  |
| TouristName | VARCHAR | 45 | 是 | 游客姓名 |  |
| TouristGender | VARCHAR | 3 | 是 | 游客性别 |  |
| TouristPhone | VARCHAR | 15 | 否 | 游客电话 |  |

Tourgroup表用于存放开设的旅行团信息，包含旅行团ID、出发时间、导游名字、导游电话以及选择的旅游线路ID等信息，如表3.3所示。其中TourgroupID为主键，是自动填充的自增整型数据。选择的旅游线路ID为外键，对应tourroute表中的TourrouteID。Tourgroup代表的是旅行团实体，该实体与旅游线路实体为N:1联系，与游客实体为1:N联系，在tourgroup表中插入外键TourrouteID，使之与tourroute表产生关系，在日后对tourroute表进行处理时，如删除数据，同时就会对tourgroup表中的相应数据进行删除。

**表3.3 tourgroup表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| TourgroupID | INT |  | 是 | 旅行团ID | 主键 |
| TourgroupTime | DATETIME |  | 是 | 出发时间 |  |
| TourgroupName | VARCHAR | 20 | 是 | 导游姓名 |  |
| TourgroupPhone | VARCHAR | 20 | 是 | 导游电话 |  |
| TourrouteID | INT |  | 是 | 旅游线路ID | 外键 |

Tourroute表用于存放开设的旅游线路信息，包含线路ID、路线信息、路线标题、路线时长、路线价格以及区分路线地域等信息，如表3.4所示。其中TourrouteID为主键，它是自动填充的自增整型数据，这意味着该表中的每一个旅游线路的ID均为唯一值，为接下来的操作奠定基础，在接下来的开发中将会使用到TourrouteID进行操作，如预约、浏览路线详细信息等。TourroutePlace为识别旅游线路为国际线路或国内线路的标识，目的是区分旅游线路的地域特性，在网页中以地域方式查询时，可以跟别查出。当该值等于0时，该条线路为国内线路，当该值等于1时，该条线路为国际线路。

**表3.4 tourroute表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| TourrouteID | INT |  | 是 | 线路ID | 主键 |
| TourrouteInformation | VARCHAR | 200 | 是 | 线路简介 |  |
| TourrouteTime | INT |  | 是 | 路线时长 |  |
| TourroutePrice | DOUBLE |  | 是 | 路线价格 |  |
| TourrouteTitle | VARCHAR | 45 | 是 | 线路标题 |  |
| TourroutePlace | INT |  | 是 | 地域标识 |  |

Scenicspot表用于存放在网站中已登记的旅游景点信息，包含景点ID、景点名称、景点简介、景点价格、景点地址、景点电话、景区图片以及景点地域标识等信息，如表3.5所示。其中ScenicspotID为主键，是自动填充的自增整型数据。ScenicsspotPlace为识别景点为国内景点或国际景点的标识，当该值等于0时，该景点为国内的景点，当该值等于1时，该景点为国外的景点。ScenicspotPicture为景区图片，保存图片的地址，然后进行展示。

**表3.5 scenicspot表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| ScenicspotID | INT |  | 是 | 景点ID | 主键 |
| ScenicspotName | VARCHAR | 45 | 是 | 景点名称 |  |
| ScenicspotPrice | DOUBLE |  | 否 | 景点价格 |  |
| ScenicspotAddress | VARCHAR | 100 | 否 | 景点地址 |  |
| ScenicspotPhone | VARCHAR | 20 | 否 | 联系电话 |  |
| ScenicspotInformation | VARCHAR | 200 | 否 | 景点简介 |  |
| ScenicspotPicture | VARCHAR | 45 | 否 | 景区图片 |  |
| ScenicspotPlace | INT |  | 是 | 地域标识 |  |

Spotofroute表用于存放已登记的景点和开设的路线之间的关系信息，因为景点和旅游线路是M:N联系，旅游线路是由数据库中的旅游景点组成，所以将它们之间的关系单独建表，这样做的好处是消除了冗余，同时也方便了对该数据库中的表的数据的操作。表中包含线路ID，景点ID以及景点在该条线路中的位置，如表3.6所示。其中TourrouteID和ScenicspotID共为该表的主键，而两键又是外键，TourrouteID对应tourroute表中的TourrouteID，ScenicspotID对应scenicspot表中的ScenicspotID。

**表3.6 spotofroute表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| TourrouteID | INT |  | 是 | 线路ID | 主键、外键 |
| ScenicspotID | INT |  | 是 | 景点ID | 主键、外键 |
| SpotInRoute | INT |  | 是 | 景点在线路的位置 |  |

Restaurant表用于存放已登记的酒店信息，其中包含酒店ID、酒店名称、酒店价格、酒店简介、地址以及电话等内容，如表3.7所示。其中RestaurantID为主键，是自动填充的自增整型数据。

**表3.7 restaurant表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| RestaurantID | INT |  | 是 | 酒店ID | 主键 |
| RestaurantName | VARCHAR | 45 | 是 | 酒店名称 |  |
| RestaurantPrice | DOUBLE |  | 否 | 酒店价格 |  |
| RestaurantAddress | VARCHAR | 100 | 否 | 地址 |  |
| RestaurantPhone | VARCHAR | 20 | 否 | 联系电话 |  |
| RestaurantInformation | VARCHAR | 200 | 否 | 酒店简介 |  |

Restaurantofspot表用于存放已登记的景点和已登记的酒店之间的关系信息，即景点周边酒店信息。因为景点和酒店之间是M:N联系，所以将其单独建表。表中包含景点ID、酒店ID以及相对位置等信息，如表3.8所示。其中ScenicspotID和RestaurantID共为该表的主键而又同为外键，ScenicspotID对应scenicspot表中的ScenicspotID，RestaurantID对应restaurant表中的RestaurantID。RestaurantAtSpot为酒店相对于景点的位置。

**表3.8 restaurantofspot表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| ScenicspotID | INT |  | 是 | 景点ID | 主键、外键 |
| RestaurantID | INT |  | 是 | 酒店ID | 主键、外键 |
| RestaurantAtSpot | VARCHAR | 100 | 是 | 相对位置 |  |

Hotel表用于存放已登记的宾馆信息，其中包含宾馆ID、宾馆名称、宾馆价格、宾馆简介、地址以及电话等内容，如表3.9所示。其中HotelID为主键，是自动填充的自增整型数据。该表中的数据大致与restaurant表中的数据相似，故在此不再重复进行介绍。

**表3.9 hotel表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| HotelID | INT |  | 是 | 宾馆ID | 主键 |
| HotelName | VARCHAR | 45 | 是 | 宾馆名称 |  |
| HotelPrice | DOUBLE |  | 否 | 宾馆价格 |  |
| HotelAddress | VARCHAR | 100 | 否 | 地址 |  |
| HotelPhone | VARCHAR | 20 | 否 | 联系电话 |  |
| HotelInformation | VARCHAR | 200 | 否 | 宾馆简介 |  |

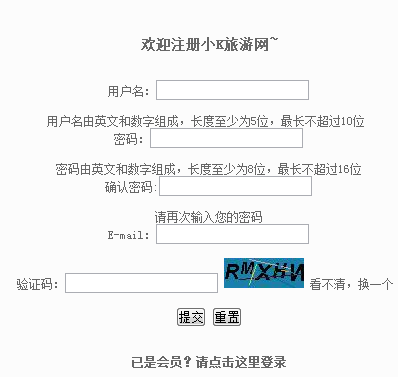
Hotelofspot表用于存放已登记的景点和已登记的宾馆之间的关系信息，即景点周边宾馆信息。因为景点和宾馆之间是M:N联系，所以将其单独建表。表中包含景点ID、宾馆ID以及相对位置等信息，如表3.10所示。其中ScenicspotID和HotelID共为该表的主键而又同为外键，ScenicspotID对应scenicspot表中的ScenicspotID，HotelID对应hotel表中的HotelID。HotelAtSpot为酒店相对于景点的位置。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 数据长度 | 非空 | 说明 | 备注 |
| ScenicspotID | INT |  | 是 | 景点ID | 主键、外键 |
| HotelID | INT |  | 是 | 宾馆ID | 主键、外键 |
| HotelAtSpot | VARCHAR | 100 | 是 | 相对位置 |  |

# 第四章 系统的实现

### 4.2.1 用户注册模块的详细设计

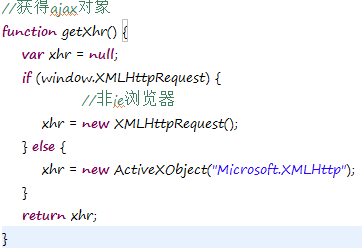
注册模块负责处理用户申请成为注册用户的工作，用户需要输入自己的用户名，密码，邮箱等数据录入到数据库中的user表。同时，在填写注册表单时，还需输入密码的重复密码，防止用户密码输入错误，还需输入网页中自动生成的验证码，以防止有不法分子恶意注册破坏网站的正常运行。注册页面为register.jsp，如图4.4所示。在输入用户名之后，系统会调用JavaScript中的Ajax，采用异步的方式联建数据库，对用户所输入的用户名进行判定，判断该用户名在数据库中是否存在。当用户输入的所有注册信息全部合法时，即可调用ActionServlet.java中的register.do方法，将用户输入的数据写入到数据库中相应的表，并且跳转到登陆页面login.jsp进行登录。如果用户输入的信息不合法系统将会在界面中进行提示，并且会在服务器端进行验证，并返回错误信息。



**图4.4 注册页面**

在系统中的Ajax技术主要就应用于异步连接数据库进行判断用户名是否重复以及判断验证码两方面。下面就详细介绍下Ajax技术对用户名是否存在进行判定的过程。

首先要初始化XMLHttpRequest对象，因为IE核心与非IE核心的浏览器对初始化对象的方法不同，所以要根据不同情况进行跨浏览器的初始化过程。图4.5即为初始化对象的代码。



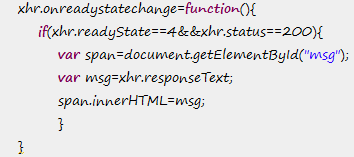
**图4.5 XMLHttpRequest对象初始化**

这段代码为使用Ajax技术的初始化通用代码，位于本项目的js文件夹中，项目中的任何使用到Ajax技术的JavaScript执行时都会首先调用初始化对象，然后再执行相应的逻辑运算。

在初始化XMLHttpRequest对象后，要在运行时向服务器发送请求，如异步方式发送GET或POST方式的请求，然后就可在浏览器与服务器之间进入到连接状态，同时要通过XMLHttpRequest对象的onreadystatechange属性指定一个回调函数，该回调函数的作用是负责处理服务器响应，这就是在向服务器发送请求之后的下一步。

在处理服务器响应的过程中，回调函数起着重要的作用。首先回调函数要判断服务器的请求状态，确定服务器端的请求已完成，接着要根据服务器的HTTP状态码，来判断服务器对请求的响应是否成功，当经过判断发现服务器的响应成功时，就会获取服务器的响应反馈给客户端浏览器。这样就完成一次异步与服务器进行交互的过程。

以注册模块中的判断用户名为例，该处理为处理字符串响应，过程较为简单，如图4.6即为JavaScript中的回调函数的具体代码。

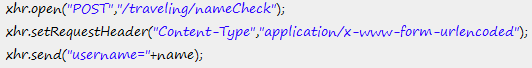


**图4.6 JavaScript回调函数详细代码**



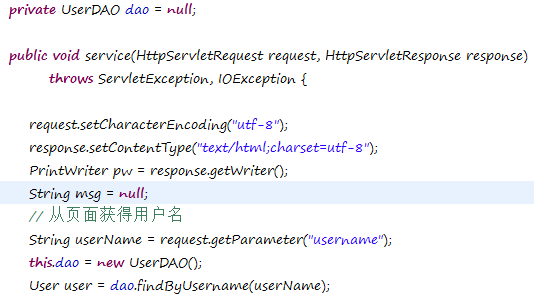
**图4.7 JavaScript获取输入的用户名**

首先将用户在页面的用户名文本框中输入的数据获得，定义该值为userName，如图4.7，然后判断浏览器是否为IE核心，选择不同的初始化方法，接下来将会向服务器端发送请求，调用特定的服务器端业务类，如图4.8所示。



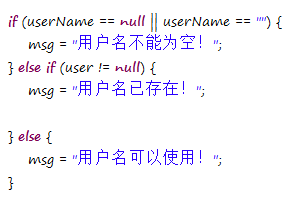
**图4.8 向服务器端发送请求以及获得服务器端响应**

在服务器端，系统将会在获得发送请求后进行一套复杂的处理，执行UserNameCheck的业务类。首先服务器会创建UserDAO接口类的对象dao，然后调用该接口类与数据库进行交互，获得查询结果，如图4.9所示。



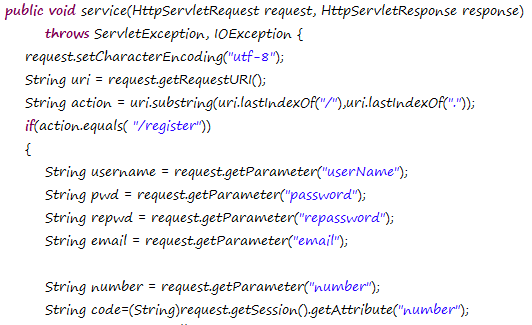
**图4.9 服务器端进行查询并获得查询结果**

在获得了查询结果之后，服务器对查询结果进行判定，如果用户名为空值，即为用户没输入用户名，将提示用户输入用户名，如果在数据库中查到该用户名，即为用户名已存在，将提示用户该用户名已存在，如果从浏览器端获得的用户名既不为空有没有在数据库中查询出，则该用户名可用，将会提示用户该用户名可用。相应的判定代码如图4.10所示。



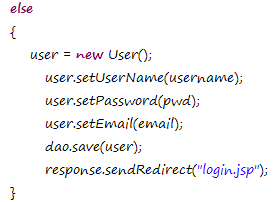
**图4.10 判断用户名合法性的代码**

当注册页面中的各种信息判定均合法后，系统就会调用register.do业务类对数据进行录入处理，首先服务器端获得浏览器端表单内的主要数据，相应代码如图4.11所示。



**图4.11 服务器端获得表单内各类数据**

在获得数据后，服务器端就会进行服务器端判定，这是为了防止不法分子在关闭JavaScript的情况下进行恶意注册，系统会判断用户名、密码、重复密码、邮箱以及验证码的合法性。在判定合法之后，服务器端就会调用user的接口类UserDAO对注册信息进行录入，如图4.12所示。

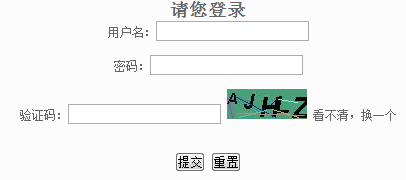


**图4.12 服务器端进行数据录入**

录入完成后，业务类会将用户的页面重定向至登陆页面进行登录，而注册模块的全部流程则已结束，注册完成。

### 4.2.2 用户登录模块的详细设计

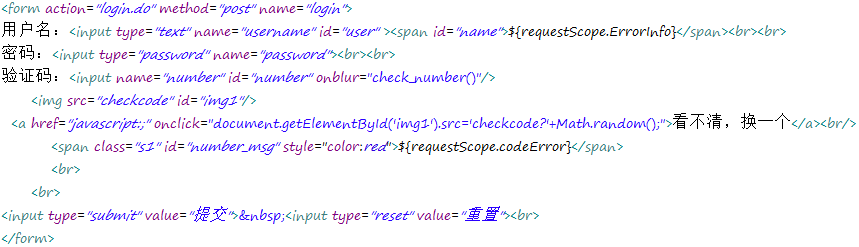
登录模块负责处理用户进行登录请求的工作，用户在登陆页面中进行登录，需要输入用户名和密码，还输入为了防止不法分子恶意登录和暴力破解而设置的，由系统随机生成的验证码。登陆页面为login.jsp，如图4.13所示。



**图4.13 登录页面**

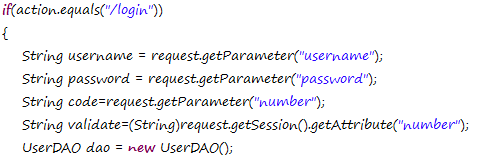
当用户输入用户名，密码和验证码之后点击提交按钮，系统将会调用ActionServlet中的login.do方法，首先验证用户输入的注册码与系统随机生成的验证码是否一致，如果不一致将返回错误信息，回到登陆页面，如果一致将会获取用户输入的用户名和密码，并与数据库中的user表中的数据进行匹配，如果没有搜索到相同的数据，则不能登录，返回到登陆页面并提示错误信息，如果发现用户名和密码均相同且相互对应时，则可以进行登录，登录的用户名保存为session对象并跳转至主页。

下面将详细介绍注册模块的工作流程。首先用户进入到登陆页面后，会有一个登录表单供用户填写。当填写完毕后，系统服务器端的业务类就会获取用户填写的表单信息，为了保护系统以及用户信息的安全性，该表单采用POST的方法发送数据。Login表单代码如图4.14所示。



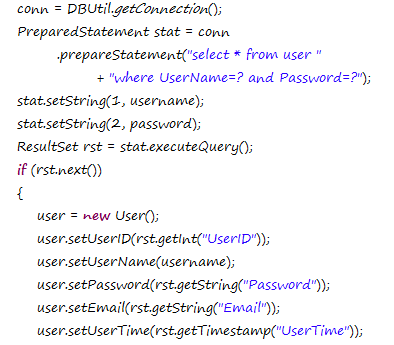
**图4.14 登录页面代码**

服务器端收到浏览器发送的请求之后，业务类将获取表单中的数据，然后将会调用接口类与数据库之间进行交互。详细代码如图4.15表示。



**图4.15 获取表单并与接口类连接数据库**

将表单通过接口类连接数据库，执行相应的数据库语句，查询数据库表中是否有所查询的数据，即查询该用户是否存在以及查询该用户名及密码是否正确。详细代码如图4.16所示。



**图4.16 登录模块接口类主要代码**

服务器端通过接口类获得查询结果之后，将会根据查询结果判断是否允许用户进行登录，首先判断验证码是否正确，若不正确将返回登录页面并提示错误信息。若正确将判断查询结果，若未查询出结果，证明该用户不存在或密码错误，返回到登陆页面并提示错误信息，若查询出结果，证明该用户存在切密码正确，登陆成功，并且将用户名设为session对象，为其它业务所用。主要代码如图4.17所示。

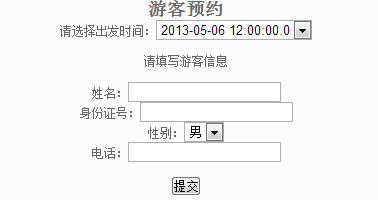


**图4.17 登录模块业务类主要代码**

而后系统将会跳转至主页，表示整个登录过程结束，登陆完成。

### 4.2.3 预约模块的详细设计

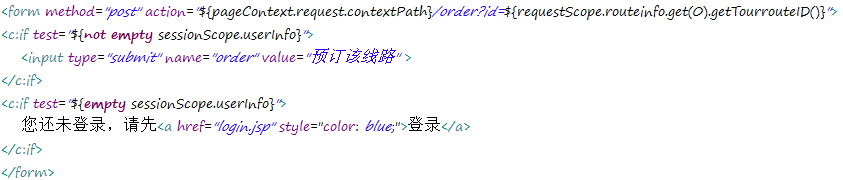
预约模块是负责处理用户浏览旅游线路后的预约工作，是网站中最重要的业务。当用户在浏览路线详细信息时，系统会先判定浏览者是否已经登录，如果浏览者没有登录，该页面将会提示用户登录而没有预约按钮，如果该浏览者已登录，则可以进行预约，然后会跳转至预约页面order.jsp，如图4.18所示。



**图4.18 预约页面**

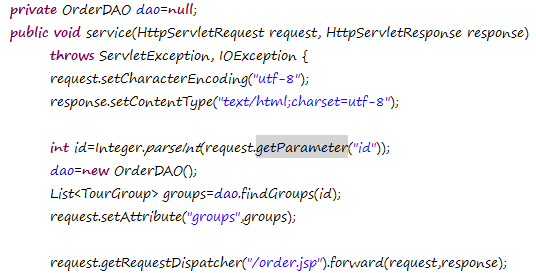
在预约页面中，系统会根据数据库中拥有的旅行团显示全部的出发时间，然后用户可以选择自己中意的出发时间，输入自己的游客信息，然后进行提交。系统就会将用户的信息录入到数据库中即可完成整个预约流程。

下面将详细介绍预约模块的工作流程。在静态页面的建设方面，首先要区分用户是否已经登录，当用户位于阅览旅游路线详细信息页面时，如果用户未登录，页面内将不会显示预约按钮，而会显示提示用户登录的信息，只有当用户已登录，即之前在登录过程中创建的session对象中的用户名存在，才会在页面中显示预约按钮。我使用JSTL标签中的<c:if>标签实现了该功能。网页的实现代码如图4.19所示。



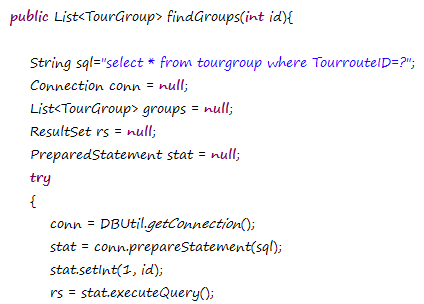
**图4.19 在页面中对用户登录状态的判定**

已登录的用户点击“预订该线路”按钮后，就会执行转入预约的业务类，同时附带id信息来确定用户将要预约的线路是哪一条，然后就会转入到预约页面。执行跳转的业务类的代码如图4.20所示。



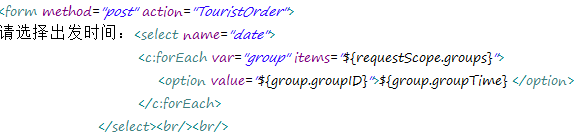
**图4.20 转入预约页面的业务类**

在该业务类中，会根据附带的id信息，通过接口类OrderDAO，查询数据库中的表，获得所需的信息，即属于该id的旅游路线所开设的旅行团，然后再附带着旅行团的信息，跳转至order.jsp页面，登记预约信息。接口类OrderDAO的主要代码如图4.21所示。



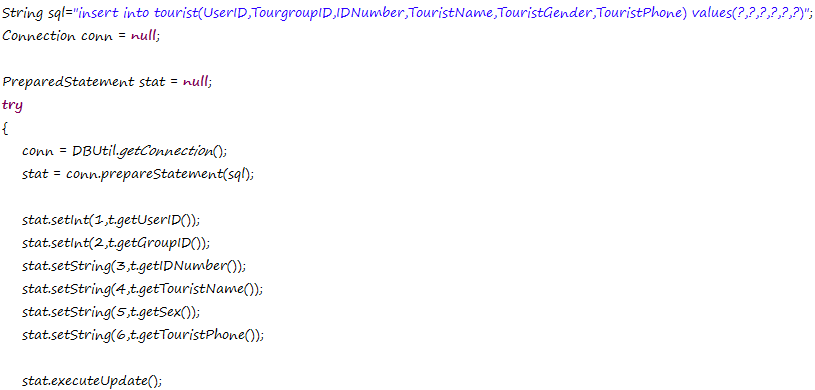
**图4.21 接口类OrderDAO主要代码**

业务类携带着所需的信息跳转至order.jsp页面后，用户即可登记信息，进行预约。在预约表单中，用户将要进行选择的出发时间，是根据之前所预约的路线开设的旅行团而显示的动态的信息，在这里我使用JSTL标签中的<c:forRach>标签和JSP中的<select>标签以及EL表达式代替时间和旅行团ID，这几个技术组合在一起共同实现。详细代码如图4.22所示。



**图4.22 select标签中展示不同的出发时间**

在用户填写好预约信息后，点击提交按钮，发送到服务器端进行响应，这时就会执行TouristOrder业务类，录入预约信息到数据库。业务类首先会获得表单中的数据，如出发时间所对应的旅行团ID，游客姓名，游客身份证号，游客的性别以及游客的联系电话。因业务类的代码与之前几个业务类的代码相似，故在此不再展示。然后将会通过连接接口类TouristDAO，将数据录入到数据库中。TouristDAO的主要代码如图4.23所示。



**图4.22 接口类TouristDAO**

在录入完毕后，就跳转至ordersuccess.jsp预约成功页面。当跳转至预约成功页面，提示预约成功，预约流程即技术，预约完成。

## 4.3 系统其它模块的详细设计

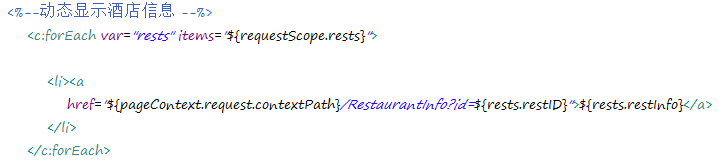
除了之前详细介绍的三个重要模块——登录模块、注册模块和预约模块之外，还有若干个负责显示信息的次要模块，如景点展示，路线展示，酒店展示等，这些模块展示信息，同时和三个重要模块之间进行连接。

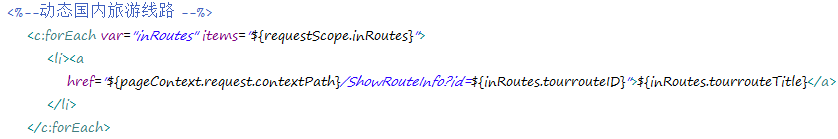
主页模块是负责显示主页信息的一个模块，该模块是一个业务类，共联结调用三个接口类，酒店接口，国际线路接口，国内线路接口。负责展示主页的动态信息，读取数据库中的信息然后传递给主页的业务类负责展示，如图4.23所示。

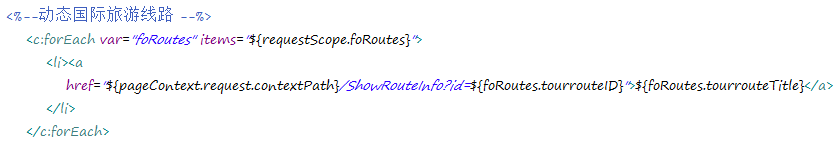


**图4.23 主页界面**

在主页中，界面中央的图片，专题因在设计之初并无建设计划，所以为静态数据酒店信息，路线信息为动态数据，将数据库中的数据读出，然后在主页中以EL表达式语言进行显示。主页中的动态信息显示代码如图4.24所示。

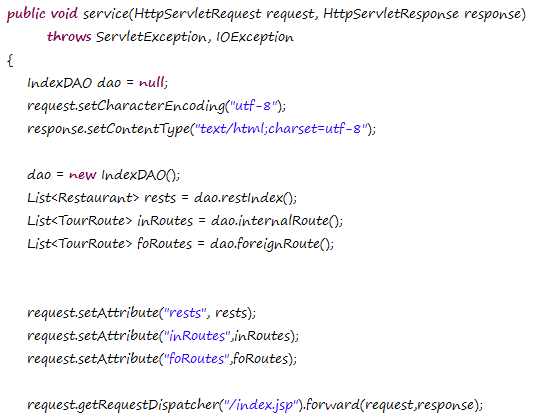






**图4.24 动态展示主页中的内容**

主页中展示的酒店信息、国内路线、国际路线是先由主页的业务类Index通过接口类进行调用，然后再反馈到主页中来，业务类的详细代码如图4.25所示。



**图4.25 业务类Index**

由上图可以看出，业务类调用了接口类IndexDAO中的三个方法：restIndex、internalRoute和foreignRoute。然后再将获得的值赋给不同的元素，然后携带这些元素跳转到主页index.jsp，实现了动态显示数据的功能。接口类IndexDAO中的方法和之前使用的接口类功能相近，代码相似，故不再一一展示出。

因为个人能力有限，并不会使用复杂的技术对网站界面进行美化，所以网站的主题静态模板是从网上搜索的资源，应用到本网站中，而其它设计均由自己完成。将网页的上部的站内链接设计为top.jsp，将下部的版权信息设计为copyright.jsp，然后再每个页面中使用include标签包含这两个页面，提高了代码的重用性。调用top和copyright的代码如图4.26所示。



**图4.26 使用include标签包含页面**

显示信息页面，景点展示，路线展示，酒店展示为整个网站中的主要交互页面，负责向用户展示信息，对于这三个页面，它们的主要结构很相似，均为在获得了用户请求之后，进行各自的工作，业务类连接接口类，然后从数据库中获得相应的数据，与之前在主页中显示信息类似，故在此不进行重复介绍。



**图4.27 旅游景点详细信息展示**

# 第五章 系统测试

## 5.1测试环境和方法

给出测试环境（即程序运行的环境），简介测试方法（如：白盒测试法、黑盒测试法，具体的有边界值测试法、区间测试法、错误测试法等，可参考相同的测试用书。

## 5.2测试用例设计

简介测试用例设计原则；

给出主要测试用例，测试用例表可如下：

表5-1 系统/网站的测试用例表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **CaseNo** | **测试观点** | **测试机能** | **测试步骤** | **预期结果** | **测试结果** | **测试日期** | **对应的问题编号** | **测试者** |
|
| 1 | Case1 | 对登录系统测试 | 正常用例 | 1、输入网址"http://localhost:8080/login"，进入系统 2、输入正确的用户名称“test” 3、输入正确密码“test” 4、点击登陆按钮 | 通过验证，系统登录成功 | OK | 2019-03-20 |  | 童洁 |
| 2 | Case2 | 对用户登陆名称登录测试 | 异常用例 | 1、输入网址"http://localhost:8080/login"，进入系统 2、输入不存在的用户名称“noneTet” 3、输入正确密码“test” 4、点击“登录”按钮 | 登录失败，显示用户名或密码错误 | OK | 2019-03-20 |  | 童洁 |
| 3 | Case3 | 对用户密码登录测试 | 异常用例 | 1、输入网址"http://localhost:8080/login"，进入系统 2、输入正确的用户名称“test” 3、输入错误密码“wrongTest” 4、点击“登录”按钮 | 登录失败，显示用户名或密码错误 | OK | 2019-03-20 |  | 童洁 |
| 4 | Case4 | 对用户功能进行测试 | 正常用例 | 1、登陆系统后，输入网址"http://localhost:8080/userInfo"， 2、单击查看用户具体信息按钮 | 显示用户具体信息 | OK | 2019-03-20 |  | 童洁 |
| 5 | Case5 | 对旅店信息功能进行测试 | 正常用例 | 1、登陆系统后，输入网址"http://localhost:8080/restaurantsList" 2、单击查看旅店列表按钮 | 显示当前数据库中的所有旅社信息 | OK | 2019-03-20 |  | 童洁 |
| 6 | Case6 | 对景点详情进行测试 | 正常用例 | 1、登陆系统后，输入网址"http://localhost:8080/viveArea" 2、在出现的菜单中点击想要查看的景点 3、点击景点名称 | 弹出一个新的页面显示景点详细信息 | OK | 2019-03-20 |  | 童洁 |
| 7 | Case7 | 登出系统试 | 正常用例 | 1、登陆系统后，输入网址"http://localhost:8080/loginOut" 2、在弹出的确认框中点击确认按钮 | 系统退出登陆，返回登陆页面 | OK | 2019-03-20 |  | 童洁 |

## 5.3测试问题及解决方法

给出测试过程中遇到的问题并简述解决的方法，以条目方式体现，也可以下表的形式给出：

表5-2 问题描述表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **提出人** | **发现日期** | **问题描述** | **状态** | **负责人** | **解决方案** | **解决日期** | **备注** |
| 1 | 童洁 | 2018/11/02 | 登陆系统输入错误密码提示有误 | 已关闭 | 童洁 | 修改代码中的提示信息 | 2018/12/06 |  |
| 2 | 童洁 | 2018/12/17 | 新增旅社出现故障，无法新增旅社 | 已关闭 | 童洁 | 数据库字段写错导致无法添加，修改字段后可修复 | 2018/12/22 |  |
| 3 | 童洁 | 2019/02/11 | 订单信息存入数据库后乱码 | 已关闭 | 童洁 | 数据库使用字符集与代码所用字符集不一致，修改代码字符集后存储功能恢复 | 2019/03/12 |  |

# 结束语

通过这次毕业设计，我认识到了自身的不足。也许这些知识点，在我学习的那个学期里我掌握的很好，但是当我在实际运用，将其糅合起来却是一件困难的事情，我发现我已经遗忘了很多，而我所记得也寥寥无几，当我把这些重新拾起来，我不仅能像以前一样熟练运用，我还能通过查阅资料学习到更多的知识。

这次的毕业设计让我学到了很多，不仅仅是知识上的，还有一些别的，比如：好记性不如烂笔头，磨刀不误砍柴工、合理安排时间的重要性等等。所以，可以说这次毕业设计从各个方面锻炼了我，我从这次的经历里得到了成长。

# 致 谢

给出对毕业实训过程中给予个人帮助的个人或单位表示感谢；

注意：每个感谢一段。

# 参考文献

[1] Eric.Jav编程思想[M].第4版.机械工业出版社,机械工业出版社

[2]杜波依斯MySQL技术内幕[M].第4版.人民邮电出版社，2011年07月

[3] 林信良JSP & Servl学习笔记[M].清华大学出版社。2012年05月

[4] 张峋，杨三成关键技术: JSP与JDBC应用详解[M].中国铁道出版社，2010年11月

[5]张银鹤点石成金: JSP+Ajax 网站开发典型实例[M].电子工业出版社，2009年01月

[6] Weiss, MA数据结构与算法分析: Java 语言描述[M].第2版机械工业出版社，2009年01月

[7] MetskerSJ.Java设计模式[M].第2版.电子工业出版社。2012年09月

[8]孙卫琴. Tomcat与Java Web开发技术详解[M].第2版.电子工业出版社，2009年 01月

[9]刘京华等Java Web整合开发王者归来[M].清华大学出版社，2010年01月

[10]贺松平，基于MVC模式的B/S架构的研究及应用[D].武汉:华中科技大学，2006年4月

[11]毕建信.基于MVC设计模式的Web应用研究与实现[D].武汉:武汉理工大学，2006年5月

[12]陈刚，Eclipse 从入门到精通[M].北京:清华大学出版社，2005年4月

[13]清宏计算机工作室,《JSP编程技巧》机械工业出版社，2004年5月

[14]萨师煊，王珊《数据库系统概论》高等教育出版社2002年2月

[15]温谦，HTMI+CSS网页设计与布局从入门到精通[M].人民邮电出版社。2008年08月

[16]刘晓义，XML技术及应用[M].哈尓演工程大学出版社. 2011年06月

[17]赵俊峰，JavaWeb应用开发案例教程:基于MVC模式的JSP+Servlet-JDBC 和AJAX[M].清华大学出版社，2012年01月