****

Southwest University of Science and Technology

本科毕业设计（论文）

题目名称：酒店预订管理系统

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术学院 |
| 专业名称 | 信息安全 |
| 学生姓名 | 李雨翔 |
| 学号 | 5120144823 |
| 指导教师 | 赵建峰 讲师 |

二〇一八年五月

**酒店预订管理系统**

摘要：改革开放以来，随着社会经济的发展，人们出行或旅游，或出差的频率越来越高，对酒店客房的需求量自然随之增加，同时，人们也越来越追求质量与效率，因此，对于酒店而言，一个高质量的预订管理系统是必不可少的。

网络技术日新月异的今天，相对于传统C/S架构的系统，B/S架构更利于开发、维护和升级。因此，酒店预定管理系统基于B/S架构进行开发，能大大提高系统的效率。同时系统整体主要是采用Java语言，利用Apache的一些开源组件进行整合开发，例如Spring、SpringMVC、Mybatis、Freemarker等。系统设计主要是针对客房预订方面进行管理。系统在角色上主要分为用户和管理员角色，管理员由于在职责方面的差异同时又分为普通管理员和超级管理员。超级管理员主要在后台对注册用户、普通管理员、客房信息以及酒店活动信息进行管理，普通管理员对订单进行处理，办理客户入住、退房等业务。用户主要在前台进行一系列的操作，个人信息管理、客房信息浏览、活动信息浏览、评论等、系统的用户界面设计友好，操作合理，有很强的实用性。

关键词：酒店； 预定管理系统； B/S架构；开源组件

**Hotel Reservation Management System**

**Abstract:**Since the reform and opening up, with the development of social economy, people travel or travel, or travel more frequently, the demand for hotel rooms naturally increases. At the same time, people are increasingly pursuing quality and efficiency. Therefore, For hotels, a high-quality booking management system is essential.

With the ever-changing network technology today, the B/S architecture is more conducive to the development, maintenance, and upgrade than the traditional C/S architecture. Therefore, the hotel reservation management system is developed based on the B/S architecture, which can greatly improve the efficiency of the system. At the same time, the system as a whole is mainly using Java language, using some of Apache's open source components for integrated development, such as Spring, SpringMVC, Mybatis, Freemarker and so on. The system design is mainly for the management of room reservations. The system is mainly divided into users and administrator roles in roles. Administrators are divided into ordinary administrators and super administrators because of differences in their responsibilities. The super administrator mainly manages registered users, general administrators, room information, and hotel event information in the background. The ordinary administrator processes orders and handles check-in and check-out operations. The user mainly performs a series of operations at the front desk, personal information management, room information browsing, activity information browsing, comments, etc. The user interface of the system is friendly in design, reasonable in operation, and highly practical.

**Key words:**hotel, management, B/S, components

目 录

[第1章 绪 论 1](#_Toc17809)

[1.1 研究的目的和意义 1](#_Toc27887)

[1.2酒店预订管理系统的作用 1](#_Toc30023)

[1.3研究的内容 2](#_Toc13145)

[1.4本章小结 2](#_Toc15493)

[第2章 系统开发关键技术介绍 3](#_Toc15566)

[2.1开发语言 3](#_Toc30392)

[2.2开发工具 3](#_Toc11798)

[2.3 B/S架构 4](#_Toc22618)

[2.4服务器技术 4](#_Toc10602)

[2.5数据库技术 4](#_Toc17360)

[2.6 后台Web服务的实现技术 5](#_Toc5192)

[2.6.1 实现模式 5](#_Toc10900)

[2.6.2 SSM框架 5](#_Toc23689)

[2.7 前端页面的实现技术 6](#_Toc1101)

[2.8 本章小结 7](#_Toc16262)

[第3章 需求分析 8](#_Toc29696)

[3.1 可行性分析 8](#_Toc27410)

[3.2 业务需求分析 8](#_Toc13192)

[3.3 用户需求分析 9](#_Toc20369)

[3.4 系统整体功能需求分析 9](#_Toc24554)

[3.4.1前台用户功能 9](#_Toc29554)

[3.4.2 后台管理员功能 10](#_Toc21836)

[3.5 非功能性需求分析 12](#_Toc23513)

[3.6 安全性需求 13](#_Toc32657)

[3.6.1系统的安全性 13](#_Toc31550)

[3.6.2数据的安全性 13](#_Toc27880)

[3.7 软硬件需求 13](#_Toc24131)

[3.8 本章小结 14](#_Toc6616)

[第4章 系统概要设计 15](#_Toc20506)

[4.1系统设计思想 15](#_Toc27267)

[4.1.1 前台用户模块 15](#_Toc15146)

[4.1.2 后台管理模块 15](#_Toc9022)

[4.2总体功能结构设计 15](#_Toc28633)

[4.3 数据库设计 17](#_Toc688)

[4.3.1 数据库环境 17](#_Toc2724)

[4.3.2 数据库命名规范 17](#_Toc29200)

[4.3.3 数据库概念设计 17](#_Toc15334)

[4.3.4 数据库物理设计 21](#_Toc9283)

[第5章 系统详细设计与实现 26](#_Toc1798)

[5.1前台子系统功能流程图 26](#_Toc734)

[5.2后台子系统功能流程图 26](#_Toc31714)

[5.3系统功能模块设计与实现 27](#_Toc98)

[5.3.1 用户注册功能 27](#_Toc17578)

[5.3.2 客房信息浏览功能 28](#_Toc24875)

[5.3.3 预订客房功能 29](#_Toc29582)

[5.3.4 个人信息维护功能 31](#_Toc22751)

[5.3.5 评论发布 32](#_Toc5871)

[5.3.6 处理预定订单功能 33](#_Toc2152)

[5.3.7 客房管理功能 34](#_Toc4076)

[5.3.8 用户管理功能 35](#_Toc2245)

[5.4 本章小结 35](#_Toc18050)

[第6章 系统相关功能测试 36](#_Toc20772)

[6.1测试概述 36](#_Toc28577)

[6.2 测试方法 36](#_Toc31536)

[6.3测试用例 36](#_Toc31964)

[6.3.1 用户注册功能模块测试 36](#_Toc15997)

[6.3.2登录模块功能测试 37](#_Toc19102)

[6.3.3 用户预订客房模块功能测试 38](#_Toc1122)

[6.3.4客房相关信息管理模块功能测试 38](#_Toc2316)

[6.3.5 客房信息浏览模块功能测试 39](#_Toc16346)

[6.4 本章小结 40](#_Toc31999)

[结论 41](#_Toc20756)

[致谢 42](#_Toc25015)

[参考文献 43](#_Toc15068)

# 第1章 绪 论

## 1.1 研究的目的和意义

近几年随着社会的发展，各地旅客数量不断增加，酒店数量也随之增加，导致酒店客房的管理方面的各种信息量也在不断成倍增长，因此，传统的酒店信息管理方式暴露出越来越多的问题，耗费人力，效率低下。这时，酒店就急需各种各样能使酒店工作人员对庞大的信息量进行更快更方便更准确进行管理的系统。

在互联网技术无处不在的今天，人们的生活已经离不开计算机，利用其工作、学习乃至购物。例如：人们去外地旅游或出差，通常会根据自己的需求选择在网上事先预订好客房，大大节约了找房的时间。此时，酒店就需要相应的客房预定管理系统对客房和预订信息进行管理。

对于客房数量较多的酒店来说，利用一套酒店预定管理系统来管理客房信息和预订信息显得尤为重要，这个系统不仅方便了客户进行预订，同时也方便了工作人员，大大节省了时间。所以开发一个酒店预定管理系统是很有必要的。

## 1.2酒店预订管理系统的作用

酒店预定管理系统从表现形式看是对酒店大量的客房信息进行信息化管理，方便处理客户的入住、退房，同时提供的客房预订功能也为客户提供便利。其作用主要表现在以下几个方面：

1. 提高酒店的管理效应

酒店预定管理系统能节省大量的人力物力，增加酒店的服务项目，提高酒店的服务档次，减少酒店管理上的漏洞，从而整体上减少了管理的成本，提高了酒店的经济效应。完善的预订功能可有效防止有房不能租或客房重订的现象，从而提高客房的出租率。及时的面向客户发布酒店的活动公告信息，提高客房收入。

1. 提高服务质量

计算机本身处理信息的速度很快，当可客人办理入住、退房、结账时，能大大节省客人等待的时间。完善的预订功能能让用户快速浏览酒店所有的客房信息，方便用户对酒店能有一个全面的了解，根据自己的需求选择满意的客房。从整体上使客户满意。

1. 提高工作效率

信息化管理能大大提高业务运作的速度和准确性。运用数据库系统保存数据避免了纸质订单填写的低效工作，其严格的数据检查更能有效的避免手工操作的疏忽而造成的错误。

## 1.3研究的内容

课题研究的是酒店预订管理系统。管理员可以在后台对客房信息，评论信息，预订订单信息，活动信息进行一系列的管理操作。用户可以在前台浏览客房信息，预订客房，评论留言等。

酒店预定管理系统整体实现了以下主要功能：

（1）非注册用户可以在线浏览客房，查看相关客房的评论，以及酒店活动公告信息。

（2）注册用户可以管理自己的个人信息，预订客房，查看历史订单以及撤销订单，同时可以对入住过的客房进行评论。

（3）普通管理员可以对预定订单进行处理，办理客人入住或退房。

（4）超级管理员可以对普通管理员进行管理，对注册用户进行管理，同时对酒店的客房信息、活动信息进行管理，同时对客房评论进行管理。

（5）系统整体要求界面美观，易于操作，可以稳定运行。

## 1.4本章小结

本章主要结合酒店的管理需求和互联网技术蓬勃发展的状态这两方面作为背景，详细阐述了现在信息化管理酒店的必要性，同时大致叙述了课题所研究的酒店预订管理系统大致需要实现的功能。

# 第2章 系统开发关键技术介绍

## 2.1开发语言

系统主要采用Java编程语言进行开发。在软件开发中Java的应用最为广泛，相对于其他编程语言来说，Java有着许多优势。

（1）Java语法不复杂，语言各方面比较均衡，拥有值得信赖的GC，能有效的避免很多编程时的低级错误。且Java是纯面向对象的语言。《Java编程思想》中提到“Everthing is object”，它能直接抽象出现实世界中对象，易于模块化开发，这在大项目的协作、维护方面很有优势。

（2）平台无关性。Java语言可以一次编译，到处运行。无论是什么平台，Java程序编译后都可以在该平台上运行，因为Java是解释性语言，编译器将其编译成字节码，然后再jvm中解释执行。所以只有在该平台上安装了对应的jvm，那么就可以在该平台上运行。

（3）Java本身有着丰富的内置的类库，通过这些类库，大大提高了开发效率。同时Java本身是一门开源的编程语言，拥有着大量的高质量的第三方库，正如一句所说：“不要重复的发明轮子”。这些优秀的第三方库拿来就可以用，如Java的很多杀手级应用：Spring、SpringBoot、Mybatis等等。

（4）提供了对Web开发的支持，例如jsp，Applet，尤其是现在广泛应用的servlet都可以用来进行Web应用程序开发；socket、rmi可以用来开发分布式应用程序。

（5）为了加强程序的安全性，Java语言提供了可以防止恶意代码攻击的机制，例如数据边界检测等。因此Java语言体系具有很好的健壮性和安全性，经常被用在网络环境中。

基于Java的种种优势，因此系统采用Java语言进行开发，配置环境为JDK1.8版本。

## 2.2开发工具

为了提高开发效率，系统开发过程中，采用IntelliJ IDEA平台作为编程环境并在其环境下完成编码工作。IDEA是一款专注于Java的强大的集成开发环境，使得编程变得享受。它每个方面的设计都专注于最大限度的提高开发人员的生产力，强大的静态代码分析和人体工程学设计是开发不仅高效，而且另人愉快。

IntelliJ IDEA的强大之处有很多。深度智能，将源代码编入索引之后，在各种环境中提供相关建议，即时和聪明的代码完成，以及即时的代码分析和可靠的重构工具；智能代码完成，基本完成了提示可见性范围内的类，方法，字段和关键字的名称，并支持“禅意代码”。当然少不了各种内置的工具和集成支持的各种框架，例如，版本控制工具，Git，Subversion等，企业级框架SpringBoot、Hibernate等，更有Web开发javascript的HTML/CSS等等。

## 2.3 B/S架构

B/S架构总体分为三层：第一层是表现层，通常表现形式为浏览器，只提供简单的输入输出功能，处理极少的部分的逻辑，由于不需要安装独立的客户端，所以能面向的用户范围很广；第二层是功能层，即Web服务器，扮演者传递和处理消息的角色，囊括了绝大多数的业务逻辑，当用户想要访问数据库时，会首先通过浏览器发送请求到Web服务器，然后经过Web服务器的处理后向数据库服务器发出请求；第三层是数据层，即数据库服务器，它扮演者最重要的角色，系统的大部分数据存在于其中，当收到Web服务器的请求后会更具相应的SQL语句进行查询并处理结果，最终将结果返回给Web服务器。

总的说来，B/S架构的安全性高、一致性好、实用性快，整个架构搭建下来界面呈现美观，并且使用方法简单，整体灵活性强，升级简单，开发和维护成本低等优点。因此，系统的架构设计区别于传统的B/S架构，采用B/S架构设计。

## 2.4服务器技术

系统采用Tomcat服务器作为servlet容器进行部署。Tomcat是Apache基金会的一个开源项目，它是一个servlet容器。由于它在部署在服务器上运行的过程中消耗的系统资源比较小，并且安全性高，扩展性能好。Tomcat服务器能作为目前市场上比较风靡的Web应用服务器，还因为它有很多扩展功能，例如：支持负载均衡、邮件服务等开发应用系统常用的功能。最后，Tomcat服务器在部署上简单，易于操作，且集成便利。

## 2.5数据库技术

系统数据库的设计采用MySQL。

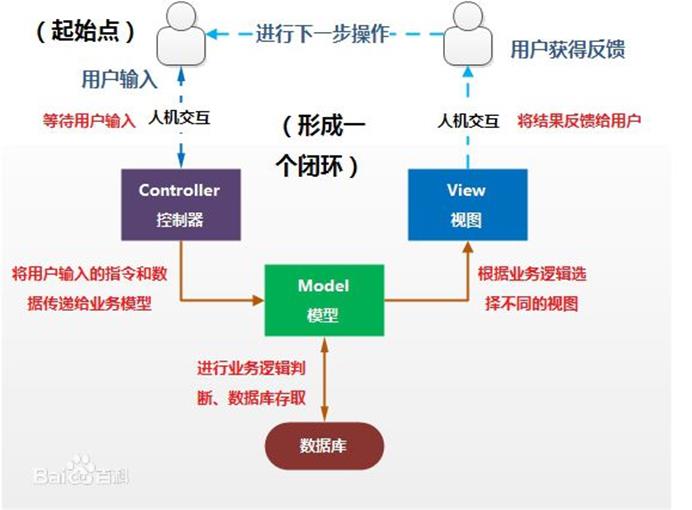
MySQL数据库作为当下最热门的数据库之一，具有许多优势。其一，它完全开源，开源的特点基于用户可以根据自己的需要修改DBMS的自由。其二，可移植性高，几乎可以在所有平台上运行。虽然在每个平台的实现方面有所不同，但其它物理体系是相同的。其三，数据库与实例一一对应的。其四，是支持多处理器，因为它的运行方式是单进程多线程的。同时，为了应对现在庞大的数据信息量，MySQL更是支持读写分离和集群。

## 2.6 后台Web服务的实现技术

### 2.6.1 实现模式

Web服务器开发采用经典的MVC模式，主要分为三个部分：模型（Model）、视图（View）和控制器（Controller）。其中，模型部分负责应用程序的业务逻辑和业务的抽象数据；视图部分负责封装应用程序的输出形式；最后控制器则负责协调模型和视图之间调用关系。

总的说来，MVC模式具有低耦合性、开发成本低、可维护性高等优点。模型、视图、控制器三者之间的交互关系如图2-1所示。



**图2-1 MVC模型**

### 2.6.2 SSM框架

在使用MVC模式中为了更高效的开发，同时整合了Apache的开源框架SSM，即分别是Spring、SpringMVC、Mybatis。

Spring是为了解决企业应用程序开发复杂性而创造的。Spring是一个分层框架，由七个定义良好的模块组成：Spring Core，即通常所说的核心容器，提供了Spring框架的基本功能；Spring context，即Spring上下文，是一个配置文件，向Spring框架提供上下信息；Spring AOP，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中；Spring DAO，制定了一个通用的异常体系，实现屏蔽了具体的持久化技术的异常；Spring ORM，Spring 框架插入了若干个 ORM 框架，都遵从 Spring 的通用事务和 DAO 异常层次结构；Spring Web 模块，Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文；Spring MVC 框架：MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。

Spring MVC是一种基于Java的实现了Web MVC设计模式的请求驱动类型的轻量级Web框架。将web层进行职责解耦，基于请求驱动指的就是使用请求-响应模型。Spring MVC 实现了即用的 MVC 的核心概念。它为控制器和处理程序提供了大量与此模式相关的功能。并且当向 MVC 添加反转控制（Inversion of Control，IoC）时，它使应用程序高度解耦。

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java对象)映射成数据库中的记录。

## 2.7 前端页面的实现技术

由于个人能力和资源有限，系统整体开发是没有采用前后端分离模式的。采用前后端一起开发的模式，相对而言系统耦合性较高。

前端页面的编写采用html5和CSS以及JavaScript。html5和CSS负责页面的整体呈现，JavaScript则负责前端的处理逻辑，例如一些前端数据校验、Ajax请求等。为了提高开发效率，运用了前端框架BootStrap，它的源码是基于最流行的CSS预处理脚本-Less和Sass开发的。你可以采用预编译的CSS文件快速开发，也可以从源码定制自己需要的样式。BootStrap优势之一是响应式布局，同一套代码可以有限的适配PC、手机平板等；其二，包含了丰富的Web组件，例如，下拉菜单、按钮组、按钮下拉菜单、导航、导航条、路径导航、分页、排版、缩略图、警告对话框、进度条、媒体对象等，你可以快速利用这些组件搭建一个美观且功能完备的页面。

页面整体的渲染是交由后台实现的。后台整合了模板渲染引擎Freemarker，它是用Java语言编写的，它不仅可以用作于变现层的实现，更可以用于生成word文档、XML文档、Java文件等。使用FreeMarker模板引擎使得开发过程中人员分工明确，提高开发效率，与此同时，在技术上，可以彻底分离变现层和业务逻辑。

## 2.8 本章小结

本章主要阐述了开发所使用的编程语言和工具，在具体实现上所采用的软件架构模式，在编程实现中使用的具体技术、方案即相关组件，以及最后系统部署时所使用的服务器和数据库系统。综合这些方面的优势，最终能使开发效率更高，产品质量更硬。

# 第3章 需求分析

## 3.1 可行性分析

技术可行性：技术可行性是指分析系统开发的技术条件是否满足项目开发的需要。课题将实现基于B/S架构的酒店预订管理系统，开发过程中采用的技术包括HTML、CSS、JavaScript、BootStrap、Java、Spring、SpringMVC、Mybatis、Freemarker等，目前这些相关技术都已发展成熟，可以在网上查阅到大量的学习资料。客户端运行在Windows平台下浏览器中，服务器端主要运行在Linux系统下，部署在servlet容器Tomcat中。因此，系统整体的实现在技术上是可行的。

经济可行性：系统的开发人员主要是学生本人，开发过程中老师义务指导，开发所使用的主要框架技术都是开源免费的，因此，系统开发成本低廉。并且系统的实现可以帮助酒店大大节省人力资源成本，提高业务效率，同时能够及时、准确、迅速的满足客户的预订需求，提高酒店的服务质量。

操作可行性：系统主要服务于酒店的管理和客户预订客房的操作。客户端主要是浏览器网页的形式，前端开发中使用的BootStrap响应式布局框架能够很好的使网页适配PC、手机、平板等设备，且页面美观简洁。使用者只需要在网页上进行鼠标操作和一些简单的键盘输入，使用者很快便能熟练的使用。因此，系统具有操作可行性。

社会可行性：系统的研发可以简化酒店的管理工作，使酒店的管理工作不在依赖于纸质文档，极大减少了酒店管理方面的漏洞和错误。同时方便酒店前台接待人员处理订单，减少了处理时间。预订功能方便客户出行前能够预订好自己满意的客房。

基于以上四方面的可行性分析可知酒店预订管系统的开发是可行的。

## 3.2 业务需求分析

酒店预定管理系统的整体目标以信息化的管理方式取代传统的手工纸质管理方式，从整体上提高管理效率，从根本上杜绝传统的手工纸质管理方式所带来的漏洞。同时系统要求能够及时、准确的反映酒店的经营状况，从而提高酒店的管理质量，获得更好的经济效益。因此，系统应做到如下要求：

（1）能快速办理客人入住、退房。

（2）能准确有效的记录客人的入住消费情况。

（3）能快捷、高效的管理酒店客房相关信息、活动公告信息。

（4）能及时的发布酒店的客房动态。

## 3.3 用户需求分析

从用户角度出发，用户能够注册系统账号并对个人信息进行维护，浏览酒店的客房相关信息，并了解其他用户对酒店客房的评论意见，根据的自己的需要对满意的客房进行预订，同时若因为某些原因能够对预定订单做出撤销操作。再者，用户需要及时了解酒店的活动信息，并对酒店提出自己的意见。因此，系统应做到如下要求：

（1）能美观的展示客房信息。

（2）能够使用户快速便捷的进行客房的预订。

（3）能够撤销预定订单。

（4）能做到让用户查看评论，并给出自己的意见以提高酒店服务质量。

（5）能及时向用户推送活动信息。

## 3.4 系统整体功能需求分析

客房预定管理系统的功能主要分为前台用户根据自己的需求进行注册登录，浏览客房信息并对选中的客房进行预订操作。后台系统管理员因职责的不同，分为普通管理员和超级管理员，普通管理员主要对预定订单进行处理，办理客人入住或退房，而超级管理员主要对注册用户，活动公告，客房信息详细信息进行管理。

### 3.4.1前台用户功能

前台用户可分为未注册用户需求和以注册用户需求。

未注册用户的功能如下：

（1）注册账号：用户填写个人信息，并验证手机号码。

（2）浏览活动公告：用户可以浏览主页面的活动公告信息来了解酒店的最新活动。

（3）浏览客房：用户可以根据客房类型浏览客房的信息，并选中某间客房查看详情，例如：价格、面积、展示图片等。

（4）浏览评论信息：选中某间客房可查看其用户的评论。

已注册用户的功能如下：

（1）登录：根据账号密码进行登录操作。

（2）维护个人信息：用户因个人信息的变更可以随时修改自己注册信息。

（3）浏览活动公告：用户想要了解酒店的最新活动，例如一些节假日优惠信息，可以浏览主页面的活动公告信息。

（4）浏览客房：用户可以根据客房类型浏览客房的信息，并选中某间客房查看详情，例如：价格、面积、展示图片等。

（5）浏览评论信息：选中某间客房可查看其用户的评论。

（6）预订客房：用户根据自己的需求选中某间客房进行预订操作。

（7）查看历史订单：用户可以查看自己以往预订的信息。

（8）评论：用户可以根据自己的入住体验对入住过的客房进行评论。

用户的用例图如图3-1所示。

### 3.4.2 后台管理员功能

后台管理员可分为普通管理员和超级管理员。

普通管理员功能如下。

（1）修改密码：管理员可以随时修改自己进入系统的登录密码，以保证系统的安全性。

（2）对预订订单信息进行处理：查看预定订单信息，办理客人办理入住、退房等。

超级管理功能如下：

（1）修改密码：管理员可以随时修改自己进入系统的登录密码，以保证系统的安全性。

（2）对预订订单信息进行处理：办理客人办理入住、退房等。

（3）管理普通管理员：对普通管理员进行管理。当酒店工作人员调动示可以添加或删除普通管理员。

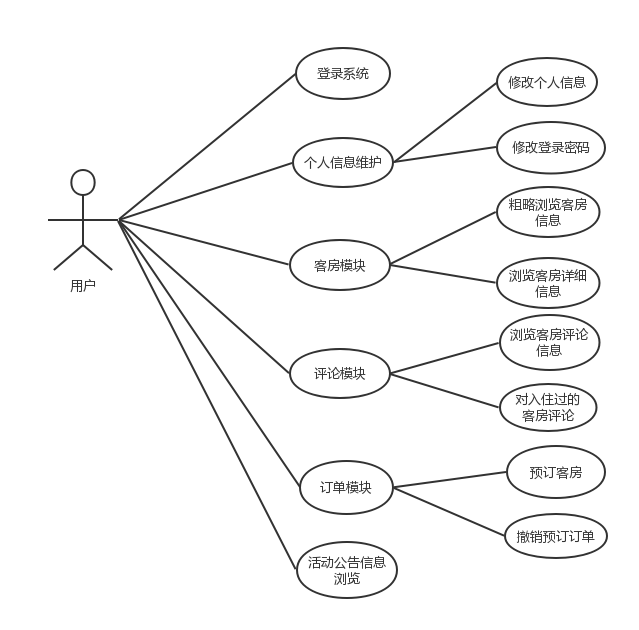
（3）客房信息管理：当酒店的客房信息变动时可以对客房类别进行添加或删除，对客房信息进行维护，添加、删除、修改信息。

（4）活动公告信息管理：发布、删除酒店的活动公告信息。

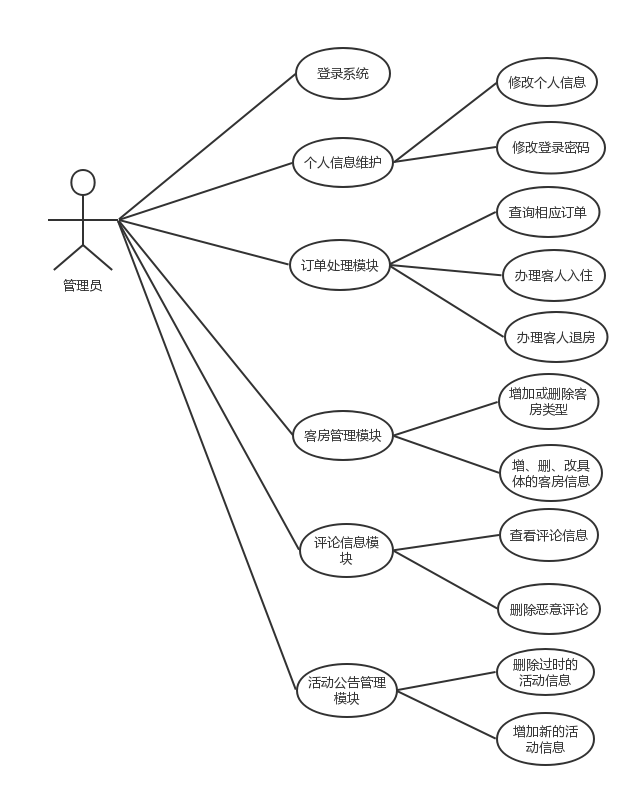
（5）评论信息管理：可以查看用户的评论，及时了解用户对酒店的建议。

（6）用户管理，可以查看注册用户的信息，并对其进行管理。

管理员的用例图如图3-2所示。



**图3-1 前台用户用例图**



**图3-2 后台管理员用例图**

## 3.5 非功能性需求分析

非功能性需求是除去功能需求之外，系统应满足的其它需求条件。这是开发过程中必须考虑的因素，系统非功能性需求分析如下。

性能方面：系统要求健壮性高，支持高并发，有一定的并发处理机制，防止出现脏数据。相应请求的速度应足够快，系统相应的服务要做到足够的性能优化。

扩展性：一个优秀的系统不仅专注于需求，更要着眼于未来，具有良好的可扩展性。因此，服务方面要做到尽可能单一化，代码方面要做到模块化插件开发，同时为之后可能发生的各种业务预留相关接口。

兼容性：系统服务器端的部署要做到尽可能做到兼容多种操作系统，客户端要做到兼容各种浏览器，同时适配各种设备。

经济性：在满足相同或相近系统需求和用户需求的前提下，应当尽可能的去节约开发成本。采用成本较低的技术手段，尽可能对现有资源进行整合升级，减少重新开发的难度。

规范性：系统开发过程中使用的各种协议，开发标准，行业规定，都应符合国家安全标准和系统开发管理规范，以确保系统功能的正常接入和使用。

## 3.6 安全性需求

### 3.6.1系统的安全性

安全性对每一个系统来说都是非常重要的。安全性很好的系统可以保护企业的信息和用户的信息不被窃取。提高系统的安全性不仅是对用户的负责，更是对企业的负责。尤其针对于酒店预订管理系统来说，必须要有很好的安全性来保障整个系统。

系统具有对使用者有权限控制，针对角色的不通限制使用者的权限，以此来确保系统的安全性。

### 3.6.2数据的安全性

数据库中的数据是从外界输入的，当数据的输入时，由于种种原因，输入的数据会无效，或者是脏数据。因此，怎样保证输入的数据符合规定，成为了数据库系统，尤其是多用户的关系数据库系统首要关注的问题。

因此，在写入数据库时，要保证数据完整性、正确性和一致性。

## 3.7 软硬件需求

开发环境：windows平台，ItelliJ IDEA作为集成开发环境，Java环境为JDK1.8。

服务器配置：linux本身是网络操作系统，相对于windows系统而言，硬件需求小，大部分版本免费，成本低。且linux是多用户多进程的系统，用户管理严格，难以被病毒攻击，还支持模块化，可裁剪。因此整套系统部署在linux服务器上，由Tomcat作为容器启动运行。因此配置一台linux主机：带宽1M，CPU2核，内存4G，系统盘40G。

数据库系统服务配置：1核2G，最高1000QPS支持。

浏览器：任意支持html5即可。

## 3.8 本章小结

本章主要结合酒店和用户的实际需求对系统做出了整体构思，分析了系统整体应该要完成的功能，应该有怎样的性能，以及数据方面的设计。

# 第4章 系统概要设计

## 4.1系统设计思想

### 4.1.1 前台用户模块

前台预订系统主要是面向客户的。登录到前台系统的用户可以通过系统提供的注册功能进行注册成为会员。客户登录到前台系统可以在线浏览酒店所有的客房信息以及酒店的活动公告信息，然后选中自己喜欢满意的的房进行预订。在浏览客房信息的时候可以看到以往客户对客房的评价，自己入住完成后也可以根据自己入住的体验对酒店的客房做出评价。前台系统的一切信息在后台都可以浏览到。管理员可以及时了解到前台系统反馈的评论信息，做到随时跟踪用户对酒店的体验，以此为依据提高酒店的服务水平。

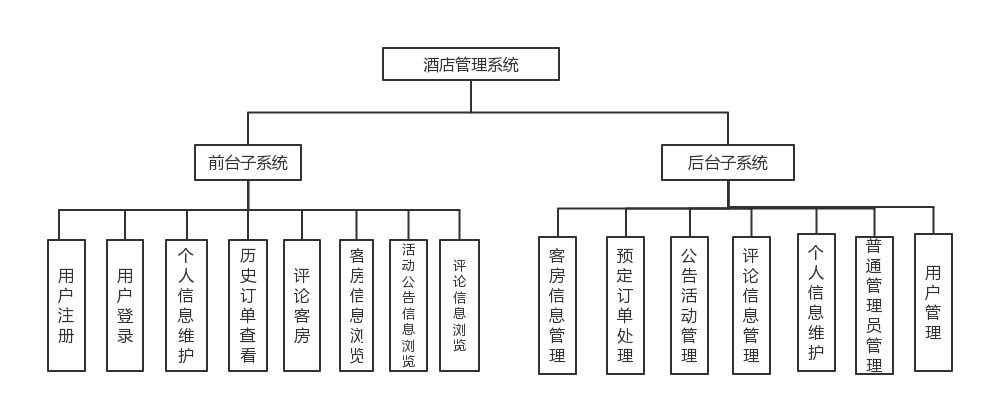
### 4.1.2 后台管理模块

后台管理系统是面向管理员的，而根据酒店工作人员职责的不同，管理员分为普通官员和超级管理员。普通管理员主要是提供给酒店的前台接待人员的，在客人办理入住或退房时，能及时对客人的相关订单作出处理，并将客房的状态改变至应该的状态，防止出现有房不能定或者重复订房的情况。超级管理员主要面向酒店的管理人员，拥有系统的最高权限，管理着酒店所有的业务。管理普通管理员，当酒店前台工作人员职位调动时，及时的添加或删除相应的普通管理员记录。管理酒店客房信息，添加、删除或修改相关的客房信息，然后发放到前台去，客户就可以在前台网页中浏览，方便客户选购。不但如此，管理员还可以通过后台系统发布酒店的活动公告信息，以达到酒店服务的一目了然。

## 4.2总体功能结构设计

酒店预订管理系统总体分为前台用户模块和后台管理员模块。

两个模块表现上是分别独立存在，但是访问的数据库是一样的。每一个模块的功能都是根据先前完成的需求分析，并查阅相关资料后整理制作的。综上所述，整个系统功能模块图如图4-1所示。



**图4-1 系统功能模块**

前台用户的主要功能模块如下：

（1）用户注册：用户填写个人信息进行注册。

（2）用户登录：注册的用户才能登录到系统进行预订客房的一系列操作。

（3）个人信息维护：用户查看自己的个人信息，并修改相应的信息。

（4）客房类型：用户根据自己的需要选择不同的客房类型浏览相应的客房信息。

（5）客房详细信息查看：用户根据自己的需要选择自己喜欢的客房查看客房的详细信息。

（6）酒店活动公告信息浏览：用户可以在首页浏览酒店最近发布的活动公告信息。

（7）评论浏览：用户可以在查看客房详细的同时，浏览该客房的评论信息。

（8）预订客房：用户选择好自己喜欢的客房后，可以进行预订该客房的操作。

（9）撤销订单：用户预订好客房之后，如果临时有别的事或者想换一间客房，可以撤销该订单。

（10）查看历史订单：用户可以查看以往的预订订单信息。

（11）评论客房：用户可以对入住过的客房进行评论，给酒店提出相应的建议。

后台管理员的主要功能模块如下：

（1）预定订单处理：根据相应预定订单信息给客人办理入住或退房。

（2）修改密码：修改登录系统的密码，以此保障系统的安全性。

（3）管理普通管理员：超级管理员对普通管理员进行管理操作，添加或删除普通管理员。

（4）酒店活动公告管理：发布新的酒店活动公告或删除过时的酒店公告。

（5）客房信息管理：对酒店客房类别以及具体的客房信息进行管理。

（6）评论信息管理：管理客房相应的评论信息。

（7）用户管理：对注册用户进行管理，可以锁定账号，或解锁账号。

## 4.3 数据库设计

数据设计是整个系统实现的基础，系统信息的存储绝大部分都依赖于数据库系统。因此，数据库设计是重中之重。

### 4.3.1 数据库环境

数据库版本：MySQL 5.7.17

数据库可视化工具：Navicat

### 4.3.2 数据库命名规范

（1）表命名规则

数据库表命名均以小写英文字母命名，且以("tb\_")开头，如tb\_user、tb\_admin、tb\_promotion等。若表明由几个英文单词组成，则单词之间均以("\_")分隔。表名的长度不宜超过20个字符。

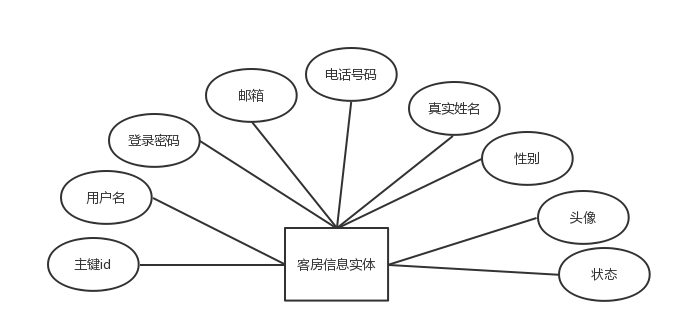
1. 字段命名规则

字段名均以小写英文字母命名。若字段包含几个单词，则单词之间均以("\_")分隔，如room\_info\_id，room\_info\_num等。字段长度一般不超过20字符。

### 4.3.3 数据库概念设计

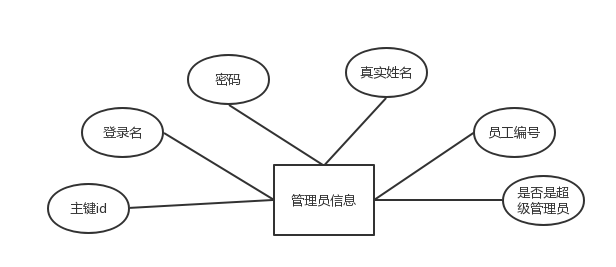
在数据库概念设计阶段我们需要对现实世界的数据对象进行抽象产生出一个能够体现出整体数据结构的抽象概念，然后将抽象概念转换为逻辑结构，最后转换为相应的数据表。在对现实世界数据对象抽象时一般采用E-R模型进行数据结构分析。系统的数据分析和E-R模型如下：

（1）用户要进入系统就必须要先登录相应的账号，在因此要存储户的个人信息，这时就需要一张存储用户信息的数据表。数据信息包括主键id，用户名，密码，邮箱，电话号码，真实姓名，性别，头像，状态。实体属性图如图4-2所示。



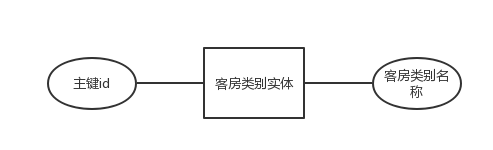
**图4-2 用户实体属性图**

（2）进入系统后台时需要特定的具有管理员身份的账号才能登入，也就是说管理员进入后台也需要登录名和密码，这时就需要建立存储管理员登录名和密码的数据表。数据信息包括主键id，登录名，登录密码，真实姓名，员工编号，是否是超级管理员。实体属性图如图4-3所示。



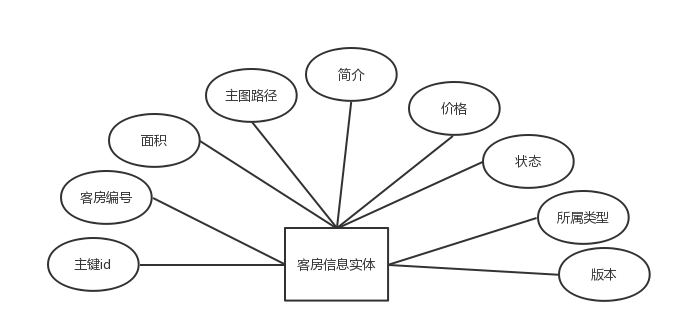
**图4-3 管理员实体属性图**

（3）酒店的客房有多种类型，这时就需要建立存储客房类型的数据基表。数据信息包括主键id，类别名称。实体属性图如4-4所示。



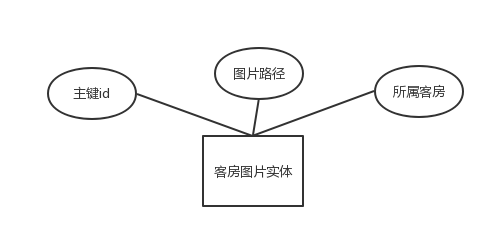
**图4-4 客房类别实体属性图**

（4）系统需要管理庞大的客房信息，此时就需要建立一张存储客房信息的数据表。数据信息包括客房id，客房编号，面积，主图路径，简介，价格，价格，所属类别，客房状态以及版本。实体属性图如图4-5所示。



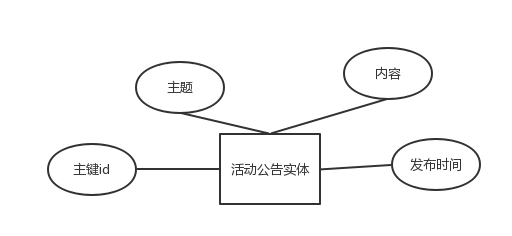
**图4-5 客房信息实体属性图**

（5）客房信息的直观展示需要图片，客房对应的图片路径信息需要一张数据表进行存储。数据信息包括主键id，图片路径，所属客房。实体属性图4-6所示。



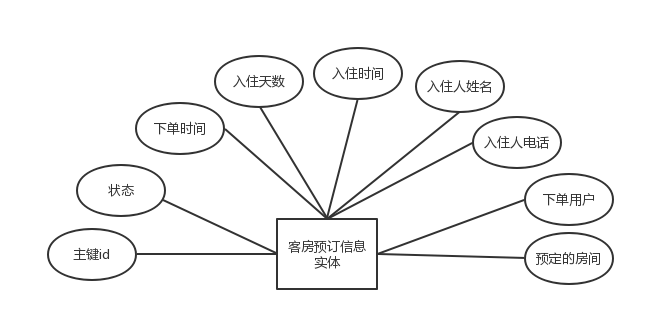
**图 4-6 客房展示图片实体属性图**

（6）酒店要发布活动公告信息，需要建立对应的数据表进行存储。数据信息包括主键id，主题，内容，发布时间。实体属性图如图4-7所示。



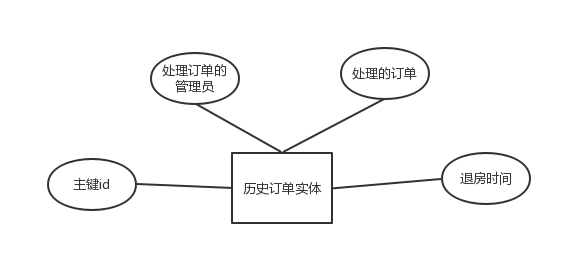
**图4-7 活动公告实体属性图**

（7）用户要对选中的客房进行预订操作，需要建立一张预订信息数据表存储用户的预定订单。数据信息包括主键id，状态，下单时间，预订天数，入住时间，入住人姓名，入住人电话，下单用户，预订的房间。实体属性图如图4-8所示。



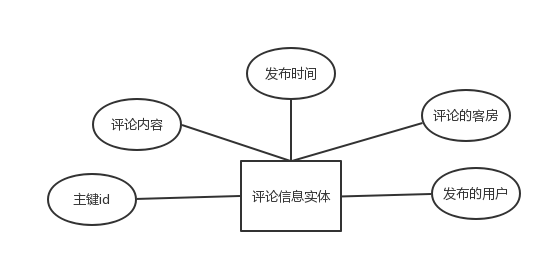
**图4-8 客房预订信息实体属性图**

（8）用户成功入住后，预订订单就变为了历史订单，需要一张对应的数据表。数据信息包括主键id，处理订单的管理员，退房时间，处理的订单。实体属性图如图4-9所示。



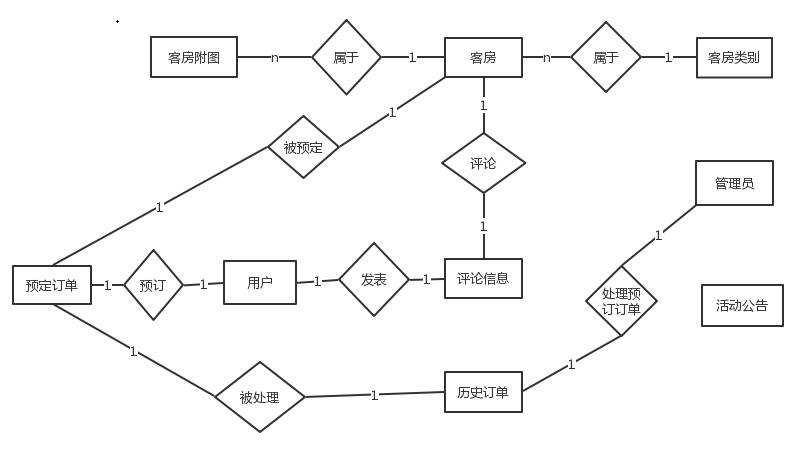
**图4-9 历史订单实体属性图**

（9）用户要对入住过的客房进行评论，需要一张数据表存储评论的相关信息。数据信息包括主键id，评论内容，评论时间，评论的客房，发布评论的用户。实体属性图如图4-10所示。



**图4-10 评论信息实体属性图**

综上所述，系统整体的E-R模型如图4-11所示。



**图4-11 系统整体E-R模型**

### 4.3.4 数据库物理设计

（1）用户信息表。根据E-R模型表结构如表4-1所示。

**表4-1 用户信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| user\_id | int | 4 | 是 | 否 | 编号 |
| username | varchar | 18 | 否 | 否 | 用户名 |
| userpass | varchar | 32 | 否 | 否 | 密码 |
| email | varchar | 255 | 否 | 否 | 邮箱 |
| phone | char | 11 | 否 | 否 | 电话 |
| realname | varchar | 32 | 否 | 否 | 真实姓名 |
| gender | char | 3 | 否 | 否 | 性别 |
| photo\_path | varchar | 255 | 否 | 否 | 头像路径 |
| status | tinyint | 2 | 否 | 否 | 帐号状态 |

（2）管理员信息表。根据E-R模型建立表结构如表4-2所示。

**表4-2 管理员信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| admin\_id | int | 4 | 是 | 否 | 编号 |
| username | varchar | 18 | 否 | 否 | 登录名 |
| userpass | varchar | 32 | 否 | 否 | 密码 |
| realname | varchar | 32 | 否 | 否 | 真实姓名 |
| empnum | char | 12 | 否 | 否 | 员工编号 |
| super\_admin | varchar | 1 | 否 | 否 | 超级管理员 |

1. 客房类别实体表。根据E-R模型建立表结构如表4-3所示。

**表4-3 客房类别实体表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| room\_catalog\_id | int | 4 | 是 | 否 | 编号 |
| catalog | varchar | 32 | 否 | 否 | 类别名称 |

（4）客房信息实体表。根据E-R模型建立表结构如表4-4所示。

**表4-4 客房信息表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| room\_info\_id | int | 4 | 是 | 否 | 编号 |
| rom\_num | varchar | 32 | 否 | 否 | 客房号 |
| area | int | 4 | 否 | 否 | 面积 |
| photo\_path | varchar | 255 | 否 | 否 | 主图路径 |
| price | decimal | 7 | 否 | 否 | 价格 |
| room\_catalog\_id | int | 4 | 否 | 否 | 所属类别 |
| status | int | 1 | 否 | 否 | 状态 |
| version | bigint | 8 | 否 | 否 | 版本号（乐观锁） |

（5）客房展示图片实体表。根据E-R模型建立表结构如表4-5所示。

**表4-5客房展示图片实体表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| room\_photo\_id | int | 4 | 是 | 否 | 编号 |
| photo\_path | varchar | 255 | 否 | 否 | 图片路径 |
| rom\_info\_id | int | 4 | 否 | 否 | 所属的客房 |

（6）客房预订信息实体表。根据E-R模型建立表结构如表4-6所示。

**表4-6客房预订信息实体表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| reservation\_id | int | 4 | 是 | 否 | 编号 |
| status | tinyint | 1 | 否 | 否 | 状态 |
| order\_time | datetime | —— | 否 | 否 | 下单时间 |
| days | int | 4 | 否 | 否 | 预订天数 |
| check\_in\_time | datetime | —— | 否 | 否 | 入住时间 |
| check\_in\_name | varchar | 32 | 否 | 否 | 入住人姓名 |
| check\_in\_phone | char | 11 | 否 | 否 | 入住人电话 |
| user\_id | int | 4 | 否 | 否 | 预订的客房 |
| room\_info\_id | int | 4 | 否 | 否 | 下单由用户 |

（7）历史订单信息实体表。根据E-R模型建立表结构如表4-7所示。

**表4-7历史订单信息实体表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| history\_id | int | 4 | 是 | 否 | 编号 |
| admin\_id | int | 4 | 否 | 否 | 管理员 |
| check\_out\_time | datetime | —— | 否 | 否 | 退房时间 |
| reservation\_id | int | 4 | 否 | 否 | 预定订单 |

（8）评论实体表。根据E-R模型建立表结构如表4-8所示。

**表4-8评论实体表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| comment\_id | int | 4 | 是 | 否 | 编号 |

**续表4-8**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| content | text | —— | 否 | 否 | 内容 |
| pub\_date | datetime | —— | 否 | 否 | 评论时间 |
| room\_info\_id | int | 4 | 否 | 否 | 客房 |
| user\_id | int | 4 | 否 | 否 | 用户 |

（9）活动公告实体表。根据E-R模型建立表结构如表4-9所示。

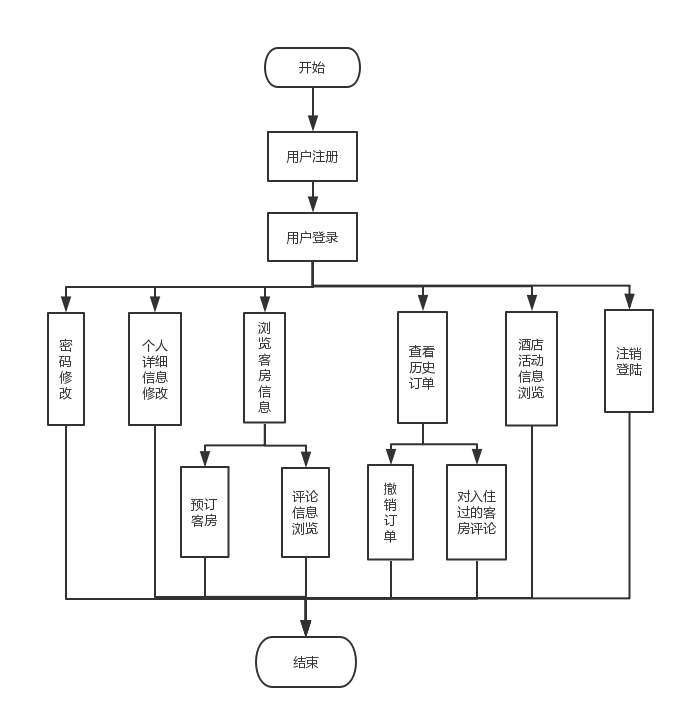
**表4-9活动公告实体表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 主键 | 允许为空 | 含义 |
| promotion\_id | int | 4 | 是 | 否 | 编号 |
| title | varchar | 255 | 否 | 否 | 主题 |
| context | text | —— | 否 | 否 | 内容 |
| pub\_date | datetime | —— | 否 | 否 | 发布时间 |

# 第5章 系统详细设计与实现

## 5.1前台子系统功能流程图

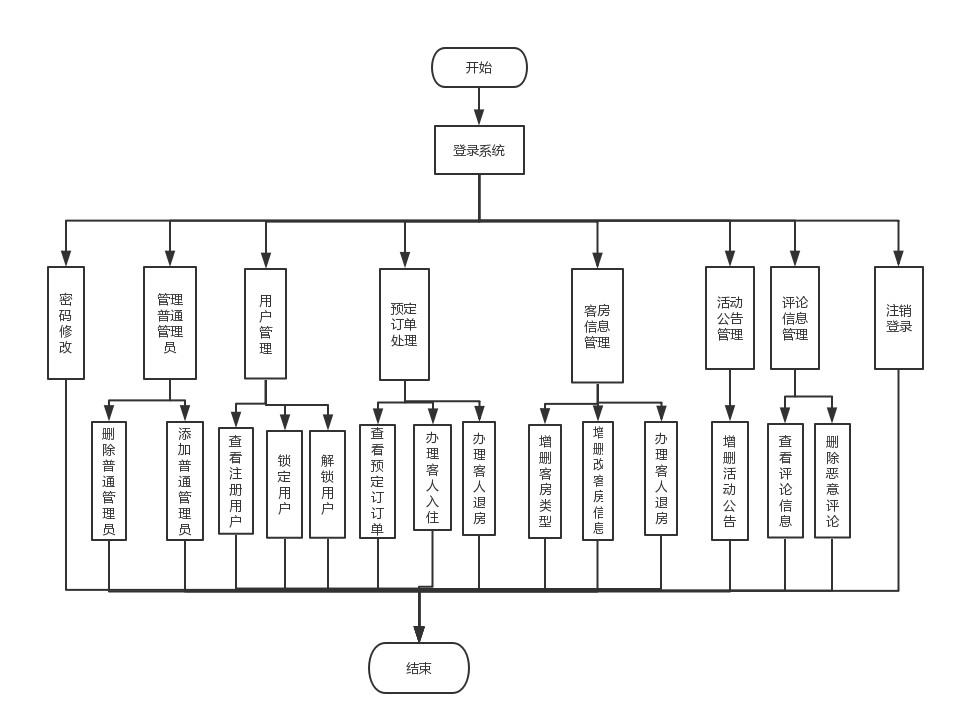
根据系统概要设计中对前台用户模块的功能分析设计，前台子系统的功能流程图如图5-1所示。



**图5-1 前台用户整体功能流程图**

## 5.2后台子系统功能流程图

根据系统概要设计中后台管理模块的功能分析设计，后台子系统的功能流程图如图5-2所示。

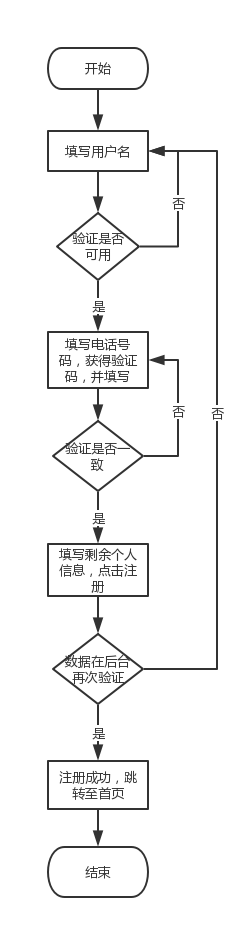


**图5-2 后台管理整体功能流程图**

## 5.3系统功能模块设计与实现

### 5.3.1 用户注册功能

用户进入系统时默认首页为index.html，通过调用controller包中IndexController.java的同toIndex()方法加载一些客房图片以及酒店活动公告信息在页面上，同时在页面右上角有注册和登录链接。用户点击注册链接，会跳转至注册页面。填写个人信息时前端会进行数据合法性的校验，除此此之外，针对用户名会通过Ajax请求后台服务校验唯一性，通过嗲用UserController.java的CheckUsername(String username)实现，同时针对用户的电话号码也会通过Ajax请求后台UserController.java的doSendCode( String realName,String phone,HttpSession session)，通过HttpClient组件调用第三方短信接口给用户发送验证，并且对用户填入的验证码进行核对，最后当点击注册按钮后会将信息提交至后台系统，这时为了防止恶意禁用前端的js验证，后台会通过Validate验证框架对提交的数据进行再次验证，验证通过才会进行注册，这里通过调用UserController.java的都doRegister()实现。注册功能的流程图如图5-3所示。

****

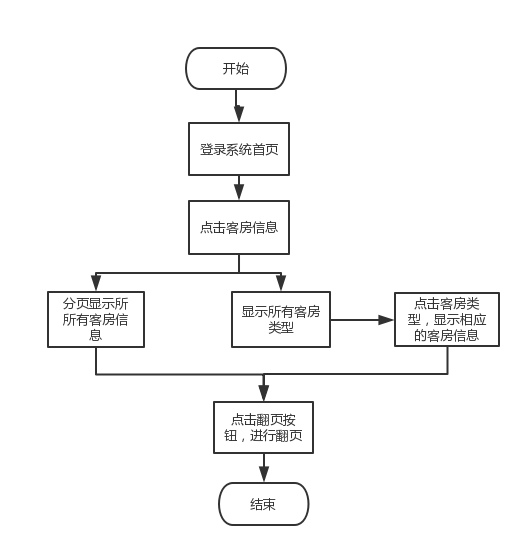
**图5-3 注册功能流程图**

### 5.3.2 客房信息浏览功能

系统首页会展示少量客房的图片，在页面下方会显示酒店最近的活动公告信息，方便用户及时了解相关的优惠信息。当点击页面上方的客房信息链接时，通过RoomController.java的showRoomByCatalog()默认分页加载所有的客房信息，可以点击上一页、下一页按钮进行翻页浏览，同时会在页面头部加载所有的客房类型，这时点击不同的客房类型，会分页显示该类型下的客房信息。

当点击任一具体的客房时，会跳转页面，通过RoomController.java的detailed()加载选中的客房的详细信息，同时在页面下方呈现出用户对该客房的评论信息，方便用户了解大众的意见。

客房信息浏览功能流程图如图5-4所示。



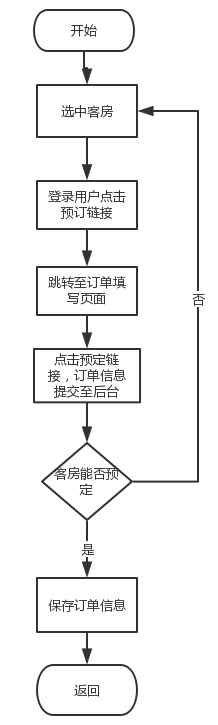
**图5-4 客房信息浏览功能流程图**

### 5.3.3 预订客房功能

预订客房功能需要考虑高并发，防止出现客房重复预订、客房状态显示出错等情况，特对客房这一共享数据增加锁机制。在乐观锁、悲观锁以及线程锁中，综合考虑性能效率和错误的可接受性选择了乐观锁机制。乐观锁的实现方式是使用版本标识来确定读到的数据与提交时的数据是否一致，提交后修改版本标识，不一致时可以采取丢弃和再次尝试的策略。在数据库表tb\_room\_info（对应客房实体）设计中增加了version字段，每次数据提交时（更改客房状态）会判断version是否匹配，若不匹配停止本次提交，若匹配则提交成功并增加version的值。具体持久层（使用的Mybatis）的实现如下：

|  |
| --- |
| //更改客房状态  <update id="modifyStatus" parameterType="roomInfo">  update tb\_room\_info  set status=#{roomStatus}, version=version+1  where version=#{version} and room\_Info\_id=#{id};  </update> |

预订客房功能整体流程：用户浏览客房信息时，同时会显示客房的状态，系统会在其显示详细信息的页面时便会判断客房的状态，若客房状态为空闲，则会显示预订的连接按钮（后台动态渲染页面时通过RoomController.java的detailed()的得到的客房详细信息实体RoomInfo.java的roomStatus属性进行判定）。在用户点击预订按钮时，会先通过UserInterceptor.java拦截器判断用户是否登录，若未登录，会跳转至登陆页面，提示客户先登录，若为登录用户就会跳转至填写预订信息的页面，填写好预订信息之后，点击提交按钮，信息提交至后台ReservationController.java的addReservationByUser()，调用ReservationService.java的saveReservation()，然后通过RoomMapper.java的modifyStatus()判断客房状态是否成功改变为预定状态，若成功则调用reservationMapper.save(reservation)保存预定订单。预订成功之后返回提示信息，告知用户预订成功。预定客房功能流程图如图5-5所示。

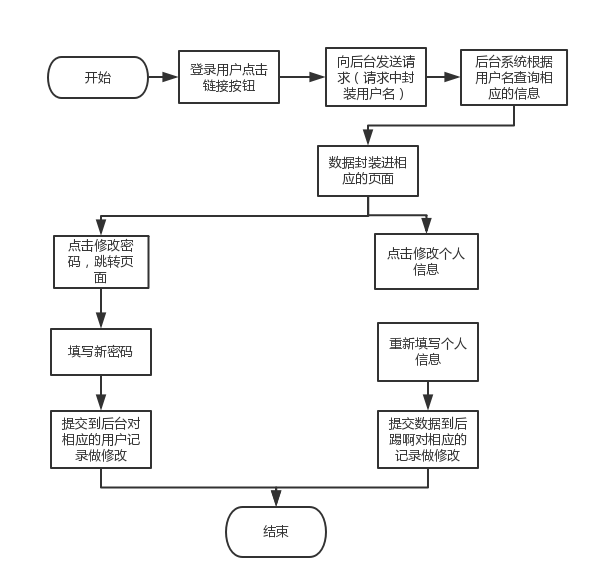


**图5-5 预定功能流程图**

### 5.3.4 个人信息维护功能

系统首页右上角登录、注册链接按钮，点击登录填写账号信息登录后，会切换内容为真实姓名和历史订单，并给出注销链接。当用户登录成功后会将个人信息保存在session作用域中，点击自己的真实姓名时，会跳转到个人详细信息页面，由后台通过Freemarker取出session作用域中的用户信息进行动态渲染，例如，邮箱、电话号码、用户名等等。同时页面上会显示修改个人信息和修改密码的按钮，这时客户可以修改自己的登录密码以保障账号的安全性，防止被人窃取账号，通过UserController.java的updatePassword()实现，同时也可以根据自己的个人信息是否变动做出相应的修改已保障酒店能更好更快的服务客户，通过updateUserInfo()实现。

个人信息维护功能流程图如图5-6所示。

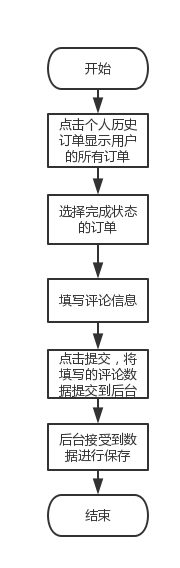
****

**图5-6 个人信息维护功能流程图**

### 5.3.5 评论发布

当登录的用户点击历史订单链接按钮时，后台通过ReservationController.java的toShowReservationByUser()查询的到用户的历史订单，然后跳转页面显示用户的历史订单，这时后台使用Freemarker根据订单的状态进行页面动态渲染，若为完成状态，会给出评论的链接按钮，此时，用户便可以对入住过的客房做出评论。点击按钮之后会跳转至评论填写页面，填写评论提交之后，后台通过CommentController.java的publish()将相关信息保存到数据库系统。

评论模块流程图如图5-7所示。

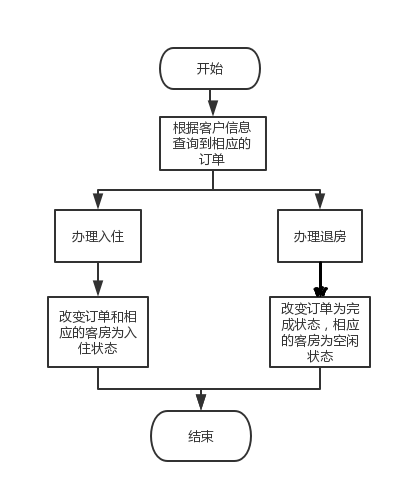
****

**图5-7评论发布功能流程图**

### 5.3.6 处理预定订单功能

处理预定订单的功能主要提供给酒店的前台接待人员。当客人来到酒店准备办理入住时，由于酒店前台接待人员具有系统普通管理员的权限，这时便可以根据客户预订的房间编号或用户账号名称查看相应的订单，同时也可以按照下单时间进行排序来显示所有的有效预定订单，后台通过调用admin模块下的ReservationController.java实现。找到相应预订订单后，后台Freemarker根据订单状态进行动态渲染判断是否给出办理入住的按钮。点击办理入住链接按钮，通过ReservationController.java的dealProcessed()即可将订单状态改变为入住状态，表明该用户已成功入住到酒店，同时改变客房状态为相应的入住状态。当客人入住结束时，前台接待工作人员找到客人相关的订单信息，通过dealCleared()将其改变为完成状态，同时更改对应的客房的状态为空闲状态以及保存相应的完成历史订单。

处理预订订单功能模块流程图如图5-8所示。

****

**图5-8 预定订单处理功能流程图**

### 5.3.7 客房管理功能

客房管理功能需要超级管理员权限的，主要是提供给酒店管理人员的，因此当普通管理员想要点击相关链接进入功能页面时，系统会不予进入，提示没有权限。酒店客房管理主要分为主要为客房类型管理和客房信息管理。

超级管理员根据酒店的需要可对客房类型进行增减操作。添加客房类型时，进入信息填写页面，填写客房类型名称时，页面会发送Ajax请求调用后台服务查询是否已存在。填写好类型名称后，通过点击链接按钮将数据提交至后台调用相关服务保存到数据库。

同样根据酒店的需求，需要对客房进行添加、删除或修改详情信息。删除或修改客房时，系统根据客房的状态判定为空闲状态时，才会给出删除和修改链接，点击删除链接按钮时，请求到达后台，还会先查询客房状态再次做出判定能否删除。点击修改链接按钮时，会跳转到修改信息的页面，重新填写好数据后，数据提交到后台会对数据库中相应的记录做出修改。

添加客房时，会给出数据填写的页面，该页面根据填写好的客房编号同样会事先发送Ajax请求查询编号是否已存在，数据填写好之后提交到后台，会调用相关服务在数据库中插入记录。

### 5.3.8 用户管理功能

超级管理员登录之后，可以分页查看所有注册的用户信息，也可以根据用户名或用户的电话号码查询相应的用户信息。根据酒店的需要可以对用户的账号进行锁定或解锁操作。

## 5.4 本章小结

本章主要详细描述了前台系统、后台系统的总体功能流程。同时对主要模块的功能进行了详细设计的描述，阐述了具体实现细节。

# 第6章 系统相关功能测试

## 6.1测试概述

针对软件的测试就是通过各种渠道来找出软件在设计、业务逻辑以及编码上存在的问题。只有通过测试这个必要的流程，才能使得软件测试员在整个软件投入生产之前尽可能的发现其中存在的漏洞和缺陷，从而对软件进行逐步修改使其功能更加完善更加合理，推动软件质量的提升。

## 6.2 测试方法

黑盒测试是最重要的一类软件测试方法。黑盒测试时，测试人员是不需要了解其实现的细节的，例如，程序的实现逻辑如何、源代码如何撰写等，仅需知道被测对象的输入和预期输出即可完成测试。

因此，黑盒测试方法最大的优势在于如下两方面：

（1）黑盒测试方法对测试人员的技术要求相对较低，测试人员甚至可以是对软件开发完全不懂的非计算机专业人员，只要对照SRS或用户手册，按照文档中描述的软件操作步骤和特性执行软件，观察输出结果就可以了。

（2）黑盒测试不需要了解程序实现的细节，因此，测试团队与开发团队可以并行完成各自的任务，从而提高团队开发的进度。

黑盒测试主要能发现以下问题：功能遗漏或者错误、界面（GUI）问题、初始化问题及终止问题、数据库问题等。

## 6.3测试用例

在软件的测试过程中，通常测试人员需要针对不同的功能模块设计多种测试用例。通过测试用例能够及时发现代码业务逻辑上是否与真实的业务逻辑相对应，及时发现代码上或逻辑上的缺陷，以此来来完善系统，提高软件产品的质量，使软件具有良好的用户体验。

### 6.3.1 用户注册功能模块测试

用户注册功能模块测试用例如表6-1所示。

**表6-1 用户注册功能模块测试用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试性能 | | 新用户注册 | |
| 用例目的 | | 测试系统新用户个人信息注册功能的功能和安全性 | |
| 前提条件 | | 进入注册页面填写个人信息 | |
| 输入条件 | 预期输出 | | 实际情况 |
| 各项基本信息输入不完整 | 不允许注册，无法点击注册按钮 | | 一致 |
| 填写已存在的用户名 | 系统显示出提示信息，要求重新填写 | | 一致 |
| 两次密码输入不一致 | 系统显示出提示信息，要求重新填写 | | 一致 |
| 填写的各项信息没有符合提示的长度和字符要求 | 系统显示出提示信息，要求重新填写 | | 一致 |
| 胡乱填写电话号码 | 收不到验证码 | | 一致 |
| 填写验证码与收到的不一致 | 系统显示提示信息告知用户验证码错误，不予注册 | | 一致 |

### 6.3.2登录模块功能测试

登录模块功能测试用例如表6-2所示。

**表6-2 登录模块功能测试用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试性能 | | 用户或管理员登录系统 | |
| 用例目的 | | 测试用户或管理员登录系统时功能是否正常 | |
| 前提条件 | | 进入用户登录页面或管理员登录页面 | |
| 输入条件 | 预期输出 | | 实际情况 |
| 各项信息不予填写，直接点击登陆按钮 | 不允许登录，提示填写账号相关信息 | | 一致 |

**续表6-2**

|  |  |  |
| --- | --- | --- |
| 输入条件 | 预期输出 | 实际情况 |
| 填写错误的登录名或密码后点击登录系统 | 提示用户名或密码错误，要求重新填写进行登录 | 一致 |
| 填写与验证码信息不一致的信息 | 系统显示出提示信息，表明验证码错误，要求重新填写 | 一致 |

### 6.3.3 用户预订客房模块功能测试

用例如表6-3所示。

**表6-3 用户预订客房模块功能测试用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试性能 | | 用户进行预订客房的操作 | |
| 用例目的 | | 测试用户进行预订客房操作时，系统的功能是否正常 | |
| 前提条件 | | 用户进入客房详情页，且该客房没有被预订或入住，即该客房能够被预定 | |
| 输入条件 | 预期输出 | | 实际情况 |
| 用户未登录，点击预订按钮 | 跳转至登录页面，提示用户先进性登录 | | 一致 |
| 正常的登录用户，点击预订按钮 | 系统弹出提示框，提示用户预订成功，同时该客房的状态更改为已预订状态 | | 一致 |

### 6.3.4客房相关信息管理模块功能测试

客房相关信息管理模块功能用例如表6-4所示。

**表6-4 客房相关信息管理模块功能测试用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试性能 | | 客房相关信息管理功能 | |
| 用例目的 | | 测试系统后台管理员对客房相关信息进行管理的功能是否正常 | |
| 前提条件 | | 登录系统后台进入相关管理页面 | |
| 输入条件 | 预期输出 | | 实际情况 |
| 普通管理员登录，点击进入客房相关信息管理的页面 | 提示非超级用户，权限不够，无法进入相关页面 | | 一致 |
| 超级管理员登入，进行删除某有客房的类别 | 该类别下有客房存在。提示删除失败 | | 一致 |
| 超级管理员登入，删除某间状态为入住或已预订的客房 | 提示该客房删除失败 | | 一致 |
| 点击增加客房按钮 | 页面正常跳转 | | 一致 |
| 增加客房是，填写已存在的客房编号 | 提示该编号已存在 | | 一致 |

### 6.3.5 客房信息浏览模块功能测试

客房信浏览息模块功能用例如表6-5所示。

**表6-5客房信息浏览模块功能测试用例**

|  |  |
| --- | --- |
| 测试性能 | 客房信息浏览功能 |
| 用例目的 | 测试系统浏览客房信息的功能是否正常 |
| 前提条件 | 进入系统首页 |

**续表6-5**

|  |  |  |
| --- | --- | --- |
| 输入条件 | 预期输出 | 实际情况 |
| 用户进入首页点击客房按钮 | 分页显示所有客房信息，并显示出客房类别信息 | 一致 |
| 点击不同的客房类型 | 显示出点击选中的客房类型下的客房信息 | 一致 |

## 6.4 本章小结

本章主要详细描述了在开发完成后进行的一系列功能测试实例。只有通过这些具体的测试实例验证，才能保证系统的安全性和可靠性。

# 结论

随着计算机互联网技术的迅猛发展，各行各业都已经实现采用计算机相关技术对日益放大的数据进行管理。该课题是酒店预订管理系统为核心展开的，主要是为了实现酒店信息化管理和用户在线预订客房的需求。

酒店预订管理系统分为了后台管理子系统和前台预订子系统。后台子系统面向酒店管理人员，实现了人员管理、用户管理、订单处理、活动信息管理、客房动态信息管理等等；前台子系统是面向用户的，实现了用户注册登录、个人信息维护、客房信息浏览、预订客房、撤销订单、历史订单查看、评论浏览等功能。

酒店预订管理系统的开发是以Java编程语言作为基础，在intellij idea平台上完成编码工作，系统整体为B/S架构，数据库系统使用MySQL。论文中详细分析了酒店预订管理系统的研究背景、研究目的和意义、开发工具和相关技术以及系统需求、系统详细设计和系统测试等等一系列内容。系统实现了酒店预定管理系统所需的一些基本功能，并通过测试对这些实现的功能进行了完善，进而提高了系统整体的实用性。整个系统的开发过程中大量使用了Java相关的知识以及前端开发使用的html和javascript等，同时涉及到了很多开源框架和组件，例如后台系统中运用的Mybatis框架、Spring框架、Spring MVC框架、Freemarker模板引擎等，前端运用的BootStrap框架等。

系统投入运行时，各功能均运行正常。用户可以正常登录和注册，正常维护个人信息，随意浏览客房相关信息，活动公告信息，评论信息等，并正常进行客房预订操作。系统的每个界面的操作符合常规逻辑，对使用者来说操作简单，界面友好。整个系统的各个功能设计合理，体现了人性化。

但是由于自己在系统开发过程中对一些用到的相关知识和技术掌握不够牢固，再加上自身开发经验欠缺，因此系统在有些方面的功能还不够完善，考虑的不够全面，因此整个系统还有待日后逐步完善。

# 致谢

伴随着毕业设计的完成，大学生涯也随之即将结束。大学四年是我最珍惜的时光，四年时光中学会了很多，也成长了很多，这段时光中每一段回忆都刻在脑海中。感谢一起学习，一起成长同学们，和成长过程悉心教导的老师们，非常感激有你们的陪伴。

首先感谢我的毕业指导老师赵建峰赵老师，毕业设计的完成离不开赵老师的一系列指导。在毕业设计的完成过程中，赵老师给出了很多中肯的建议，正是由于老师一丝不苟的工作态度，我的毕业设计才能顺利的完成。

最后，感谢在大学生涯中每一位教导我的老师，是你们教给了我丰富的知识，更教会了我遇到问题时，如何去应对并解决。谢谢你们的帮助与支持。

# 参考文献

[1] 孙小焰，酒店客房管理系统[J].信息与电脑(理论版),2012,(08):134-135.

[2] 施荣连，论酒店客房部的管理[J].怀化学院学报,2011,(11):34-35.

[3] 悠虎；如何向，MySQL数据库的表中录入数据[J].网络与信息,2010,(06):39-40

[4] 李洋，SSM框架在Web应用开发中的设计与实现[J].网络与信息,2016,(07):30-79

[5] Bruce Rich;Theoder J.Shrader;Athony J.Nadalin, Java and Internet[J].Intelligent Information Management.,2010,(10):210-230

[6] Elizabeth A Williamson,David K Harrison,Mike Jordan. Information systems development within supply chain management[J]. International Journal of Information Management,2004,(5):25-32.

[7]Matt Bishop. Trends in academic research: vulnerabilities analysis and intrusion detection[J]. Computers & Security,2002,(7)56-78.

[8] 唐建国，MySQL数据库系统安全分析[J].数据库与信息管理,2010,(23):6418~6420.

[9] T Kim;WH Ahn;J Oh, MVC Architecture Driven Restructuring to AchieveClient-Side Web Page Composition[J]Software Engineering and Service Science (ICSESS). 2016 ,Vol.45 ,47-53.

[10] Gordana Lvankovlc, Mateja Jerman.，The use of Decision Making Infomation: A Comparative Expolatory Study of Slovene Hotels[J]Managing Global Transitions. 2011,Vol.8(3),pp.307 DOAJ.

[11] 任钢，iBATIS框架源码剖析[M].卓越科技出版，2010：135-181.

[12] 拉德，深入解析Spring MVC与Web Flow[M].人民邮电出版社出版，2008：135-196.

[13] 约翰逊，Spring框架高级编程[M].机械工业出版社，2006，80-121.