重庆第二师范学院

**2018 届全日制本科生毕业论文**

**题 目： 基于Java SSM框架的酒**

**店管理系统的设计与实现**

**教学学院： 数学与信息工程学院**

**专业年级： 2014级物联网工程**

**学生姓名： 张百全 学号 1410504147**

**指导教师： 胡 飞 职称 副 教 授**

**2018 年 5 月**

基于Java SSM框架的酒店管理系统的设计与实现

数学与信息工程学院　物联网工程专业2014级　张百全

**摘 要：**随着科技与经济的不断发展，尤其是互联网以及移动互联网的广泛普及，企业对公司业务的管理也越来越走向网络化、智能化。在本文中，根据酒店对业务管理的需求，设计并着手实施了可供酒店管理者工作人员使用的酒店管理系统。此平台基于Java EE技术，使用了开源且企业级开发中流行的Java SSM（Spring，SpringMVC，MyBatis）框架，SSM框架是一个当今流行的MVC架构，旨在打造一个低耦合、高重用，可靠性高，方便维护，易根据酒店业务修改功能的酒店管理系统。本系统前端界面由jsp来实现，后台功能使用Java语言实现，使用MySQL数据库，Tomcat作为整个系统的后台服务器构建。

**关键词：**Java EE； SSM开源框架；MVC架构；低耦合；MySql；Tomcat

**Abstract:**  With the continuous development of science and technology and economy, especially the widespread use of the Internet and mobile Internet, the management of the company's business to the company is increasingly becoming networked and intelligent. In this article, according to the hotel's demand for business management, the hotel management system that can be used by hotel managers is designed and implemented. This platform is based on Java EE technology and uses the Java SSM (Spring, Spring MVC, MyBatis) framework popular in open source and enterprise-level development. The SSM framework is a popular MVC architecture that aims to create a low-coupling, high-reuse, reliability. High, easy to maintain, easy to modify the hotel management system based on hotel business. The front-end interface of this system is implemented by jsp. The back-end function is implemented in Java language. Using MySQL database, Tomcat is built as the background server of the entire system.

**Keywords:** Java EE; SSM o pen source framework; MVC Architecture; Low coupling; MySQL; Tomcat。

目 录

[摘 要： I](#_Toc512382646)

[Abstract: I](#_Toc512382647)

[1 绪论 1](#_Toc512382648)

[1.1 开发背景 1](#_Toc512382649)

[1.2 选课目的 1](#_Toc512382650)

[1.3 研究内容 2](#_Toc512382651)

[1.4 研究方法 2](#_Toc512382652)

[2 开发环境与技术 2](#_Toc512382653)

[2.1 开发环境 2](#_Toc512382654)

[2.1.1 Eclipse简介 2](#_Toc512382655)

[2.1.2 Tomcat简介 3](#_Toc512382656)

[2.1.3 Mysql数据库简介 3](#_Toc512382657)

[2.2 开发技术 4](#_Toc512382658)

[2.2.1 Java EE 简介 4](#_Toc512382659)

[2.2.1 Java SSM(Spring,SPringMVC,Mybatis)简介 5](#_Toc512382660)

[3 系统概要设计 5](#_Toc512382661)

[3.1 酒店管理系统模块设计 6](#_Toc512382662)

[3.2 设计思想以及架构 12](#_Toc512382663)

[4 基于Java SSM框架的酒店管理系统的总设计 12](#_Toc512382664)

[4.1 系统相关技术介绍 13](#_Toc512382665)

[4.2 可行性分析 13](#_Toc512382666)

[4.2.1 业务管理上的可行性 13](#_Toc512382667)

[4.2.2 技术上的可行性 13](#_Toc512382668)

[4.3 功能需求 13](#_Toc512382669)

[4.3.1 酒店住房业务的基本流程主要包括 13](#_Toc512382670)

[4.3.2 系统框架结构示意流程图 15](#_Toc512382671)

[4.3.3 数据库设计概述 15](#_Toc512382672)

[4.3.4 数据字典说明 17](#_Toc512382673)

[4.4 系统测试、运行与维护 20](#_Toc512382674)

[4.4.1 系统测试 20](#_Toc512382675)

[4.4.2 系统运行与维护 20](#_Toc512382676)

[结 论 21](#_Toc512382677)

[致 谢 21](#_Toc512382678)

[参考文献 22](#_Toc512382679)

# 1 绪论

## 1.1 开发背景

酒店行业是一个前景辽阔而又竞争激烈的行业。自改革开放以来，我国的酒店业发展迅速，已经成为了一个具有相当规模的产业。由于我国的综合国力与人民生活水平的快速提高，通过调查，中国将在2020年将成为世界上位居第四的旅游国家；届时中国的酒店业也将得到同步发展，同时我国加入世界贸易组织，旅游业和酒店业将会更加开放，那个时候，我国的酒店行业将面临更多更大的机遇和挑战。但是，至今仍有为数众多的酒店依然停留在由人工来进行管理和操作的阶段，基于这种情况的考虑已经不能很好的适应当前的经济发展趋势。所以说要想让现代的酒店工作质量和效率得到提高，采用先进的数字化酒店管理系统来改变酒店的业务方式，实现酒店业务的自动化管理已经成为一种必然。本酒店管理系统采用B/S架构。按照标准化业务程序流式设计，将采用主流开发工具来开发，搭建比较稳定可靠安全的服务端来帮助客户管理，并且还采用MySQL数据库，会通过测试用例来达到测试效果。

## 1.2 选课目的

对与任何的一家酒店来讲，决定酒店经营水平好坏的就是酒店的服务水平和管理水平。尤其在现今酒店行业空前发展，业务规模越来越大的情况下，传统的人工管理与操作的经营方式显然早已经达不到现在酒店对于其业务支撑的要求。所以采用酒店信息化管理系统，利用科技水平和手段来提高酒店的业务效率与服务水平保障已是现代酒店走向高效化与规范化的一条必经之路。虽然使用信息化管理系统并不一定是酒店管理走向成功的决定性因素，但是足以在酒店的正常经营中发挥出高效、快捷的巨大作用，助理酒店的业务管理与服务水平达到一个新高度。因此，在酒店管理和运营中使用现代化信息化的酒店管理系统，已经成为酒店提升自身工作管理效率，发挥最大价值的必要手段。酒店信息化管理系统在达到节省人力物力成本的同时，提高了酒店的经营效率，并做到了及时、高效的满足顾客的需求，从而使酒店获得更好的口碑与认可。

## 1.3 研究内容

系统为酒店的管理者提供了基于B/S架构的web交互的管理平台。系统提供以下四个一级菜单及二级子菜单：

1. 客房管理：新增房间，实时房态，实时房态列表。
2. 客人入住：入住登记查询，预定管理，预定管理查询，换房，换房查询。
3. 系统设置：用户管理，部门管理，角色管理，功能树管理。
4. 安全管理：应急事件，留言板。

## 1.4 研究方法

1.通过指导老师胡飞的指导帮助。

2.通过网络查阅相关文档资料，以及运用相关开发文档。

3.用Eclipse、Navicat Premium、Mysql、Maven、Notepad++等工具进行开发。

4.数据库采用Mysql数据库。

5.和同学们之间的交流和讨论

6.用Java语言编写代码，反复调试程序，以实现该系统的功能

# 2 开发环境与技术

## 2.1 开发环境

### 2.1.1 Eclipse简介

Eclipse是当今世界上Java开发者最喜欢的IDE之一，它诞生于2001年。作为一款最受欢迎的Java IDE，它丰富的插件功能是其广受欢迎的很大一部分原因。尤其它是功能丰富的Java EE集成开发环境，包括了完备的编码、Debug调试、测试和打包发布等功能，完它整支持HTML,CSS，Spring，SpringMVC，Mybatis,JSP,Javascript,SQL等主流技术和框架。

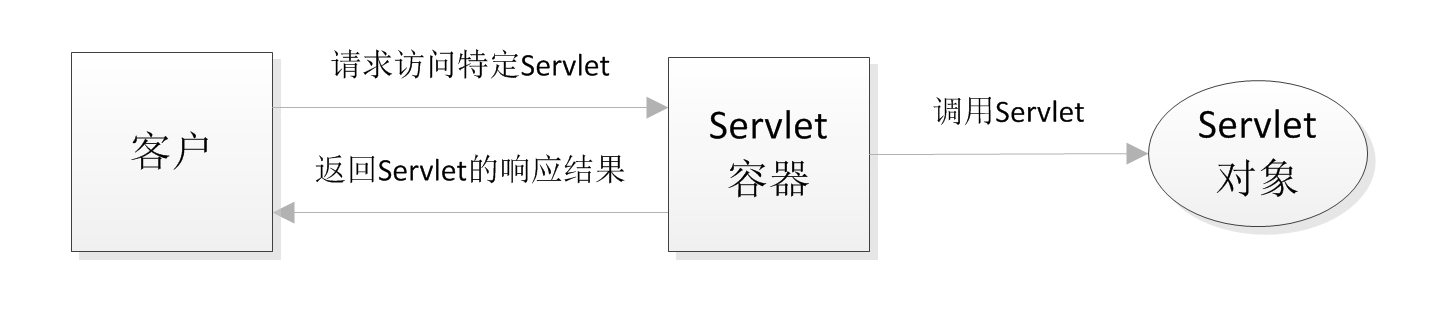
### 2.1.2 Tomcat简介

Tomcat 是由SUN 公司开发的小型轻量级Web服务器，目前由开源组织Apache负责维护的一种Servlet容器，Tomcat由java语言来编写。

Servlet是一种运行在服务器端上的组件，它具有如下的几个特点：

1. 轻量级Web服务器，很好的扩展性，稳定性，易用性。
2. 占用系统资源较小，对服务器机器性能配置要求不高。
3. 使用简单，操作方便，只需把Tomcat解压在服务器主机文件系统中，启动Tomcat就可使用。
4. 免费而且开放源代码，且具有良好的扩展性及安全性。

Tomcat服务器做为Servlet组件的容器的基本原理如下图

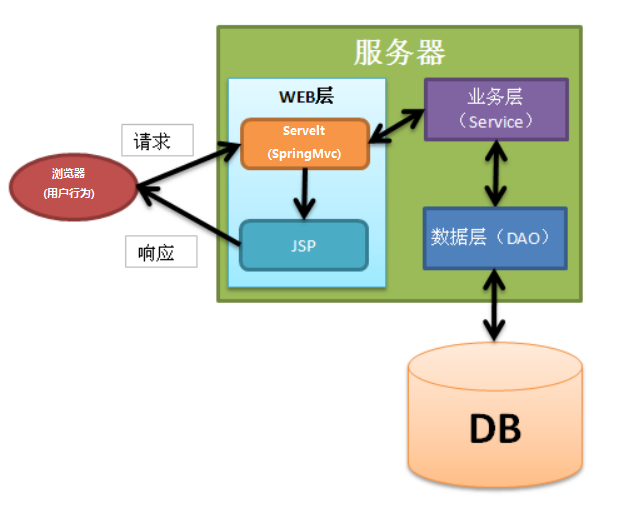


（1）

图2-1 Servlet原理图

### 2.1.3 Mysql数据库简介

MySQL目前是Oracle公司旗下的一款轻量级关系型数据库，其前身是由瑞典Mysql AB软件公司开发的一款数据库软件。Mysql是当下特别受欢迎和特别流行的关系型数据库之一。Mysql数据库操作极其简便，软件安装包特别轻量。支持完整的SQL语言，目前最新的软件版本为Mysql8.0。最新版数据库新增了一系列更加强大的新特性，运行效率较上一版本提升了2倍。在以前的版本中Mysql的默认字符集为latin1,该字符集对中文的支持极不友好，会出现乱码等情况。在刚刚发布的最新版Mysql8.0中，将utf8mb4作为Mysql数据库的默认字符集，这将对中文开发者提供更多的便利。



（2）

图2-1 Mysql数据库构架图

## 2.2 开发技术

### 2.2.1 Java EE 简介

讲到Java EE,首先不得不讲Java语言，Java是目前Github网站上受欢迎程度和使用人数均稳定排名前三的编程语言。Java语言有Java ME，Java SE，Java EE三个分支，其都属于Java语言，不同的地方是它们所主要应用的地方不同。Java ME是为移动设备，如机顶盒，移动电话等嵌入式电子设备提供的Java语言平台，目前已基本被Android替代。Java SE是Java 语言和Java平台的统称，使用Java语言开发了数以亿计的计算机程序，Java SE也是目前最受欢迎的编程语言之一。Java EE是基于Java SE的运行于服务器端的编程语言，Java EE为企业级的Web服务端开发提供了一系列的支持和技术方案，提供了包括Servlet，EJB，JDBC等多种组件的标准规范。

### 2.2.1 Java SSM(Spring,SPringMVC,Mybatis)简介

SSM框架，是Spring + Spring MVC + MyBatis的缩写，这个是继SSH之后，目前比较主流的Java EE企业级框架，适用于搭建各种大型的企业级应用系统。Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。Spring MVC的架构是一个实现代码间低耦合很流行的软件设计的架构。Mybatis是一个数据持久层框架。下面这个是Spring MVC框架主要的工作原理图：



图2-2 Spring MVC原理图

# 3 系统概要设计

概要设计将本酒店管理系统的功能与需求进行进一步的细分，划分出每个单独的功能和模块，包括使用的技术，架构，思想，组件，框架，数据库数据表和数据字典设计等。

## 3.1 酒店管理系统模块设计

客房管理模块：新增房间，实时房态，实时房态列表。

酒店管理系统中最重要的功能就是客房管理，客房管理模块的功能决定了一个酒店管理系统的实用性与便捷性程度。在本系统的客房管理模块中，图形化交互的客房实时房态，清楚明白的反映出了酒店当前所有客房的概要信息，所有房间大体状况，剩余何种类型，位于什么楼层，剩余多少空净房的状况。当酒店前台工作人员登陆进系统，左边侧边栏为系统菜单列表，右边为使用iframe进行单独分割的内容主页，这样在系统使用人员操作的过程中可以方便的查看选择其它功能菜单。

图3-1 （1）实时房间状态图形化视图

该界面为本系统登陆后的主页面，以一种创新的图形化交互方式实现了酒店管理系统中常见的和最基本的功能操作需求。 不同状态的房间以不同的颜色进行显示标注和区分，酒店前台工作人员使用本系统可以非常直观的了解酒店所有房间的实时状态。从而根据房间实时状态进行个性化的操作如点击空房进行入住登记，点击已入住房进行入住信息修改，换房，退房等操作，点击已退房可将脏房重新置为空净房。整个系统的房间状态图形化操作实现了从入住到退房到重置空房的一个完整的酒店业务场景流程。为酒店管理人员和工作人员提供了一个直观有效的操作平台，从而实现了酒店业务的流程化，智能化操作和运营。

本系统核心功能之一图形化交互相关前端jsp代码片段：

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8" isELIgnored="false"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>实时房态列表</title>

<meta name="description" content="实时房态列表">

</head>

<body>

<div class="admin-biaogelist" style="position: static;">

<form id="search"

action="<c:url value='/roommanagement/realtime\_data.action'/>"

method="post">

<input type="hidden" name="page" value="" />

<div class="am-btn-toolbars am-btn-toolbar am-kg am-cf">

<ul>

<li><input class="gjz-quq" name="roomnum" value="${roomnum }"

class="am-form-field am-input-sm am-input-xm" placeholder="房间号查询" />

</li>

<li><select class="xlk-quq" name="floor">

<option value="">房间楼层</option>

<option value="2楼">2楼</option>

<option value="3楼">3楼</option>

<option value="4楼">4楼</option>

<option value="5楼">5楼</option>

<option value="6楼">6楼</option>

<option value="7楼">7楼</option>

<option value="8楼">8楼</option>

<option value="9楼">9楼</option>

</select></li>

<li><select class="xlk-quq" name="state">

<option value="">房间状态</option>

<option value="空净房">空净房</option>

<option value="已入住">已入住</option>

<option value="脏房">脏房</option>

</select></li>

<li><select class="xlk-quq" name="type">

<option value="">房间类型</option>

<option value="标准单人间">标准单人间</option>

<option value="标准双人间">标准双人间</option>

<option value="特惠大床房">特惠大床房</option>

<option value="豪华大床房">豪华大床房</option>

</select></li>

<li><button onclick="compaire()"

class="am-btn am-radius am-btn-xs am-btn-success find-quq">

<span class="find-quq find">搜索</span>

</button></li>

</ul>

</div>

</form>

<div class="box">

<div class="divHeight">

<c:forEach items="${paging.list}" var="room" varStatus="status">

<c:choose>

<c:when test="${room.type eq '标准单人间' }">

<div id="${room.roomid }" class="one" onclick="method(this)">

<p class="room\_num">

<font>${room.roomnum }</font>

</p>

<p class="roon\_info">

<font>${room.type }</font>

</p>

</div>

</c:when>

<c:when test="${room.type eq '标准双人间' }">

<div id="${room.roomid }" class="two" onclick="method(this)">

<p class="room\_num">

<font>${room.roomnum }</font>

</p>

<p class="roon\_info">

<font>${room.type }</font>

</p>

</div>

</c:when>

<c:when test="${room.type eq '特惠大床房' }">

<div id="${room.roomid }" class="three" onclick="method(this)">

<p class="room\_num">

<font>${room.roomnum }</font>

</p>

<p class="roon\_info">

<font>${room.type }</font>

</p>

</div>

</c:when>

<c:when test="${room.type eq '豪华大床房' }">

<div id="${room.roomid }" class="four" onclick="method(this)">

<p class="room\_num">

<font>${room.roomnum }</font>

</p>

<p class="roon\_info">

<font>${room.type }</font>

</p>

</div>

</c:when>

</c:choose>

<c:choose>

<c:when test="${room.state eq '脏房' }">

<script type="text/javascript">

var roomid = $

{

room.roomid

};

var d = document.getElementById(roomid);

d.style.backgroundColor = "red";

</script>

</c:when>

</c:choose>

<c:choose>

<c:when test="${room.state eq '已入住' }">

<script type="text/javascript">

var roomid = $

{

room.roomid

};

var d = document.getElementById(roomid);

d.style.backgroundColor = "blue";

</script>

</c:when>

</c:choose>

</c:forEach>

</div>

<div class="room\_instruct">

<div class="emptynetroom">

<p class="roon\_info">

<font>空净房</font>

</p>

</div>

<div class="livein">

<p class="roon\_info">

<font>已入住</font>

</p>

</div>

<div class="dirty">

<p class="roon\_info">

<font>脏房</font>

</p>

</div>

</div>

</div>

<br /> <br />

</div>

</body>

<script src="<c:url value='/assets/js/amazeui.min.js'/>"></script>

<script>

$(function() {

$("button[look]").click(function() {

$("#modalLook").modal("open");

});

});

var method = function(o) {

roomid = o.id;

var url = "${pageContext.request.contextPath}/roommanagement/room.action?roomid="

+ roomid;

alert(roomid);

window.location.href = url;

}

</script>

</html>

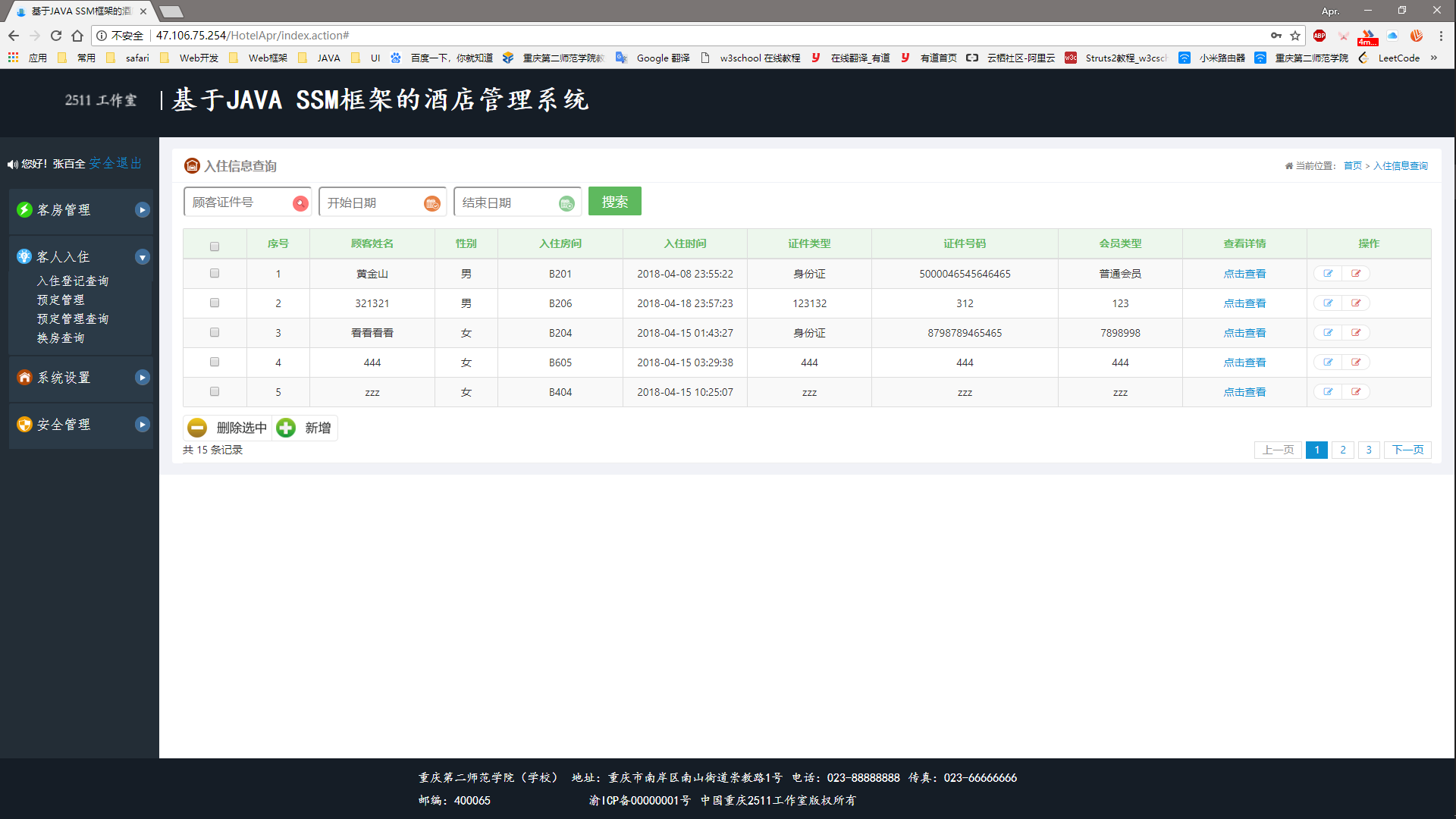
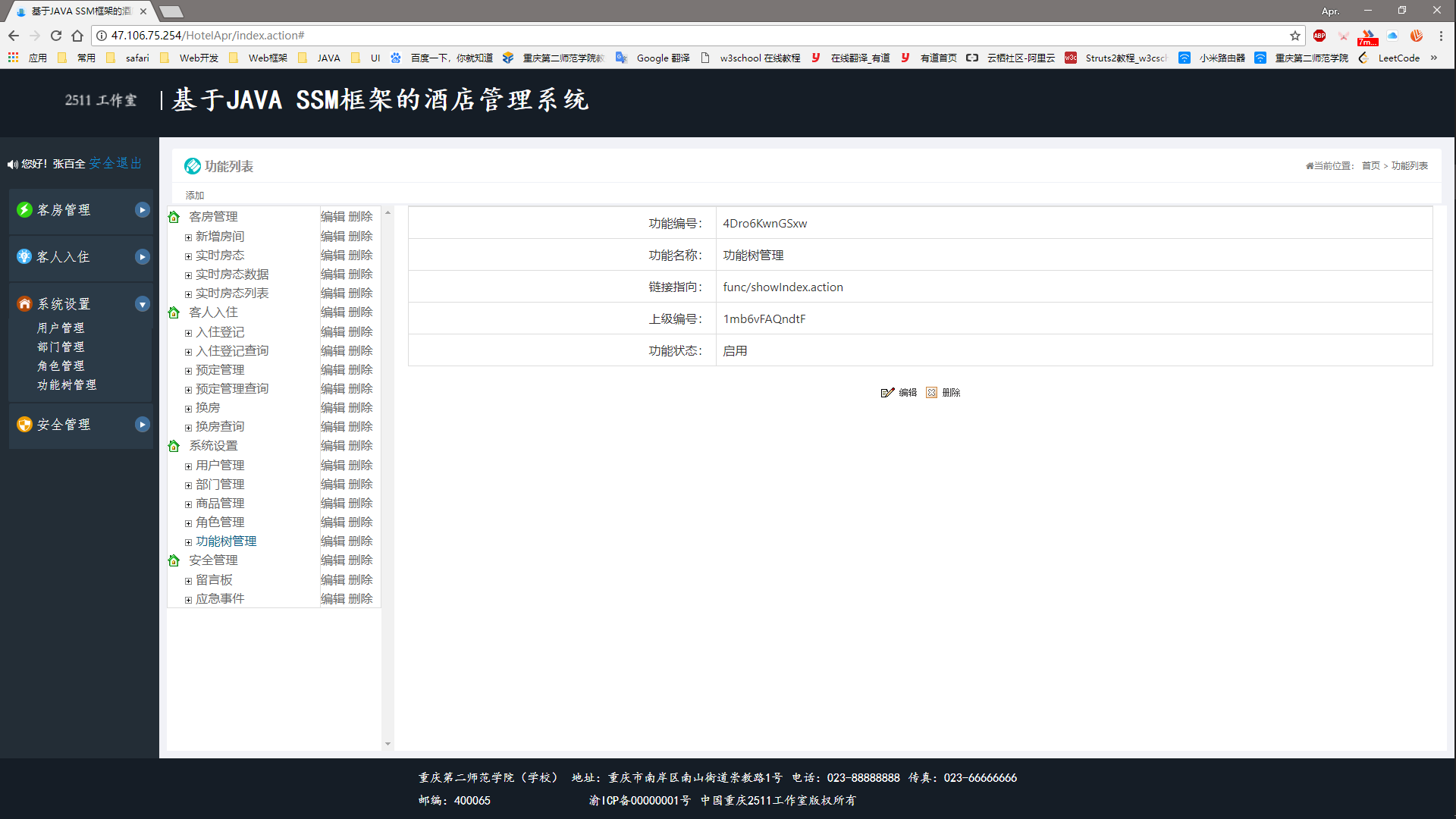
客人入住模块：客人入住模块包含入住登记查询，预定管理，预定管理查询，换房查询等子菜单。

图3-1 （2）入住信息查询菜单列表页面

在该模块中酒店管理人员可以很方便的进行所有预定查询，查看所有入住客人的详细信息，还可以修改客人入住信息，删除客人入住信息，输入入住登记的身份证号，根据入住时间段进行查询等。

系统设置模块：该模块包含了用户管理，部门管理，角色管理，功能树管理等子菜单。系统设置模块是对酒店内部管理单独设置的模块，其主要功能将只对系统管理员和酒店管理人员开放。在角色管理子菜单中可以新增删除角色，设置每个角色的权限，这也实现了一套系统，不同用户不同权限登陆系统将会有不同的个性化功能菜单列表。

图3-1 （3）系统功能菜单列表页面

## 3.2 设计思想以及架构

将整个系统按模块进行划分，采用多种设计模式，提高代码的可读性以便对代码进行修改和对功能的完善，降低类之间的耦合性。按照分层的思想来实现，是程序模块化。

本系统采用的是MVC架构，模型层（Model）、视图层（View）、控制层（Controller）三层结构。

1）、模型(Model)：通常Action类中主要处理控制功能，尽量不包括业务逻辑，而是把业务逻辑的处理委托给其他组件处理。

2)、视图(View)视图层就是前台终端界面，我们把数据通过Action返回到视图层，显示在前台终端上。

3)、控制器(Controller)：MVC架构中，用一个总控制器，然后再根据请求的URL转到相应的业务Action里面去，在由业务Action去访问service层, service层调用DAO层。

# 4 基于Java SSM框架的酒店管理系统的总设计

## 4.1 系统相关技术介绍

本管理信息系统将采用技术成熟的Java语言作为前台开发工具。后台数据库采用MySQL数据库，因为它的灵活性、安全性和易用性为数据库编程提供了良好的条件。因此，系统的软件开发平台已成熟可行。硬件方面，科技的飞速发展使硬件的更新速度越来越快，容量越来越大，可靠性越来越高，价格越来越低，其硬件平台完全能满足此系统的需要。[8]系统采用的是原型法的开发方法。本着开发人员对用户需求的理解，通过简单的系统反洗、系统设计，快速实现一个原型系统，然后通过反复修改来实现管理信息系统。

## 4.2 可行性分析

### 4.2.1 业务管理上的可行性

酒店管理系统可以实时显示酒店所有客房的状态，包含空净房，已入住房，脏房等房间状态，酒店工作人员可以很方便的通过图形化的实时房态进行客人入住登记，已入住客人的查找搜索，并且可以实现按身份证号搜索，按入住时间，入住楼层房间等条件进行个性化精确搜索。同时酒店工作人员可以在图形化的实时房态视图上进行客人退房和换房等操作，完全符合酒店的业务需求。

### 4.2.2 技术上的可行性

技术采用java SSM后端企业级开发框架。后台数据库采用MySQL数据库，搭配Druid数据库连接池，整个系统的稳定性、耦合性、安全性和易用性都有良好的保障。

## 4.3 功能需求

### 4.3.1 酒店住房业务的基本流程主要包括

客房预定，顾客通过拨打酒店前台电话或者其他方式向酒店进行客房预定，酒店前台工作人员查询实时房态后确定有空房就接受顾客预定，并对顾客预定的客房状态做出相应的操作，客房状态改变。

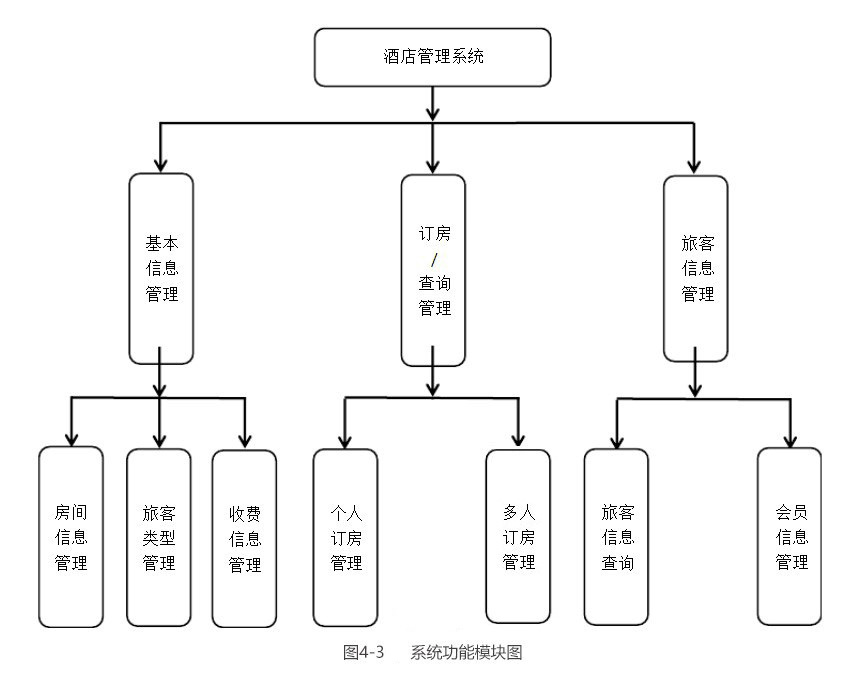
顾客入住，如果顾客已经预定了房间，就根据预定信息直接进行入住操作，此时客房状态由“已预定”变为“已入住”；如果顾客没有预定房间，则酒店前台工作人员根据空房状态和客户需求直接进行顾客入住操作，该房间状态由“空净房”变为“已入住”。

顾客换房，如果顾客对入住的客房条件不满意，则可进行顾客换房操作。酒店前台工作人员将顾客的房间进行更换，原来住的房间状态变更为“脏房”，新更换的房间状态同时转变为“已入住”。

顾客退房，顾客退房时，酒店前台工作人员根据顾客的房卡，进行退房操作。计算顾客入住期间的房费，等各种消费和押金总额。对顾客收取费用或者退回押金。顾客信息做假删除保存到入住表中。客房状态由“已入住”转为“脏房”。

清洁打扫工作人员在打扫完“脏房”后通知酒店前台工作人员，操作人员将房间状态转为“空净房”。

### 4.3.2 系统框架结构示意流程图



4-3 系统框架结构示意流

### 4.3.3 数据库设计概述

该课题旨在实现本项目核心的权限管理的功能，就是在同一个登陆界面不同角色的用户拥有不同的权限。在登陆本系统后会显示有不同的菜单选项。经查阅相关资料和文献后我清楚了实现这样的一个功能需求，需要建立这样的五张数据表，分别为User表，Role表，Permissions表，User role relational关系表，Role permissions relational关系表。

由于与数据库建立连接是一个非常耗时耗费计算机性能资源的过程，并且一个数据库服务器也只能建立有限个连接，一般安装在普通性能机器上的Mysql数据库只能建立20个左右稳定的数据库连接。所以说当一个用户量比较大的Web应用在运营使用中仅仅使用传统JDBC进行连接数据库连接，每个用户发送请求都要建立一个数据库连接的话，系统的性能将会大幅度的降低，很快将导致数据库系统瘫痪，造成非常不好的用户体验。所以为了提升数据库的利用效率，本设计使用了数据库连接池技术，数据库连接池的意义在于稳定与数据库服务器建立一定数量的数据库连接。当一个用户的连接完成后，释放连接到连接池，当下一个用户再发起请求时，该连接仍存在于连接池中，这样便不会是每个用户都直接与数据库服务器建立连接，而是通过数据库连接池，有了数据库连接池，用户所需要的数据库连接服务将得到十分有效且高效的管理，从而保障数据库连接的可靠性，高效性与稳定性。本管理系统使用了阿里巴巴Druid开源数据库连接池来保障系统数据库连接的性能。

applicationContext.xml中关于配置Druid数据库连接池相关代码：

<!—配置Druid数据库连接池 -->

<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource">

<property name="url" value="${jdbc.url}" />

<property name="driverClassName" value="${jdbc.driver}"></property>

<property name="username" value="${jdbc.username}" />

<property name="password" value="${jdbc.password}" />

<property name="initialSize" value="${initialSize}" />

<property name="maxActive" value="${maxActive}" />

<property name="minIdle" value="${minIdle}" />

<property name="await" value="${await}" />

<property name="removeAbandoned" value="${removeAbandoned}" />

<property name="removeAbandonedTimeout" value="${removeAbandonedTimeout}" />

<property name="logAbandoned" value="${logAbandoned}" />

</bean>

dataSource.properties数据库连接配置文件代码：

<!—数据库连接配置文件代码 -->

#数据库

jdbc.url=jdbc:mysql://localhost:3306/hotel\_zhang?useUnicode=true&characterEncoding=UTF-8&allowMultiQueries=true&useSSL=true

jdbc.driver = com.mysql.jdbc.Driver

jdbc.username = root

jdbc.password = root

#连接池

maxActive = 60

initialSize = 10

maxWait = 10000

minIdle = 10

#对于长时间不使用的连接强制关闭

removeAbandoned = true

#超过10分钟开始关闭空闲连接

removeAbandonedTimeout = 600

#将当前关闭动作记录到日志

logAbandoned = true

### 4.3.4 数据字典说明

员工信息表（employee）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| emplyeeId | 员工id | Int | 11 | 主键 |
| usename | 员工名称 | varchar | 255 |  |
| realname | 员工姓名 | varchar | 255 |  |
| phone | 手机号 | varchar | 255 |  |
| titleId | 职务 | varchar | 255 |  |
| deptId | 部门编号 | varchar | 255 |  |
| deptName | 部门名称 | varchar | 255 |  |
| groupId | 分组类别 | varchar | 255 |  |
| Isdelete | 删除标识符 | Int | 1 | 0未删除，1删除 |
| Ex1 | 备用字段 | varchar | 255 | 预留字段1 |
| Ex2 | 备用字段 | varchar | 255 | 预留字段2 |
| Ex3 | 备用字段 | varchar | 255 | 预留字段3 |

角色表（userclassinfo）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| UserClassID | 用户类别id | Int | 11 | 主键 |
| UserClassName | 用户类别名称 | varchar | 255 |  |
| ClassNotes | 类别备注 | longtext | 0 |  |

职务信息表(title)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| titleId | 职务ID | Varchar | 50 | 主键 |
| titleName | 职务名称 | Varchar | 50 |  |
| EX1 | 备用字段 | Varchar | 50 |  |
| EX2 | 备用字段 | Varchar | 50 |  |
| EX3 | 备用字段 | Varchar | 50 |  |

角色权限关系表（m\_functioncontrol）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| ID | 功能编号 | varchar | 11 | 主键 |
| PXID | 功能排序号 | Int | 11 |  |
| NAME | 功能名称 | varchar | 255 |  |
| URL | 功能链接 | varchar | 55 |  |
| PID | 上级编号 | varchar | 255 |  |
| ACTIVATE | 是否启用 | Int | 1 | 0：未启用  1：启用 |

用户角色关系表（m\_sys\_funcsecurity）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| FID | 编号 | int | 11 | 主键 |
| MIS\_USERID | 用户分组 | int | 50 |  |
| FUNCTION\_ID | 功能编号 | varchar | 12 |  |

部门信息表（department）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| deptId | 部门ID | varchar | 50 | 主键 |
| deptName | 部门名称 | varchar | 50 |  |
| company | 公司 | varchar | 50 |  |
| parentId | 上级部门 | varchar | 50 |  |
| Isdelete | 删除标识符 | int | 1 | 0：未删除  1：已删除 |
| EX1 | 备用字段 | varchar | 50 | 预留字段1 |
| EX2 | 备用字段 | varchar | 50 | 预留字段2 |
| EX3 | 备用字段 | varchar | 50 | 预留字段3 |

客房信息表（room）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| roomid | 主键id | int | 11 | 主键 |
| roomnum | 房间号 | varchar | 255 |  |
| customerid | 顾客id | int | 11 |  |
| floor | 房间楼层 | varchar | 255 |  |
| type | 房间类型 | varchar | 255 |  |
| price | 房间价格 | decimal | 10 |  |
| state | 房间状态 | varchar | 255 |  |
| isreservation | 是否有预定 | varchar | 255 |  |
| reservationtime | 预定时间 | datetime | 0 |  |
| arrive | 到达时间 | datetime | 0 |  |
| deposit | 押金 | varchar | 255 |  |
| instructions | 房间说明 | varchar | 255 |  |
| EX1 | 备用字段1 | varchar | 255 |  |
| EX2 | 备用字段2 | varchar | 255 |  |
| EX3 | 备用字段3 | varchar | 255 |  |
| EX4 | 备用字段3 | varchar | 255 |  |
| EX5 | 备用字段5 | varchar | 255 |  |

顾客入住信息表（livein）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 字段类型 | 长度 | 备注 |
| customerid | 主键id | int | 11 | 主键 |
| customernum | 顾客编号 | varchar | 255 |  |
| name | 顾客姓名 | int | 11 |  |
| gender | 顾客性别 | varchar | 255 |  |
| certificatetype | 证件类型 | varchar | 255 |  |
| roomnum | 入住房间号 | varchar | 255 |  |
| checkintime | 入住时间 | datetime | 0 |  |
| leavetime | 退订时间 | datetime | 0 |  |
| days | 入住天数 | varchar | 255 |  |
| deposit | 押金 | varchar | 55 |  |
| certificatenum | 证件号 | varchar | 255 |  |
| membertype | 会员类型 | varchar | 255 |  |
| note | 备注 | varchar | 255 |  |
| EX1 | 备用字段1 | varchar | 255 | 0：删除  1：未删除 |
| EX2 | 备用字段2 | varchar | 255 |  |
| EX3 | 备用字段3 | varchar | 255 |  |
| EX4 | 备用字段3 | varchar | 255 |  |
| EX5 | 备用字段5 | varchar | 255 |  |

## 4.4 系统测试、运行与维护

### 4.4.1 系统测试

(1)检查本系统各操作界面的显示是否正常，各按钮链接跳转是否正确有效。

(2)检查本系统的查询、修改和增加、删除、批量删除功能是否正常实现。

(3)检查各输入框输入内容是否验证，极端值会不会导致系统出现异常错误。

(4)检查系统录入信息是否准确,提示信息是否合理。

(5)检查用户界面是否合理美观，界面操作是否简洁易用。

### 4.4.2 系统运行与维护

本管理系统的维护是针对系统的各种功能，按照不同模块的维护，系统维护的内容大致可分为以下几类：

1. 系统应用维护。系统应用维护是本系统需要维护的主要对象。系统维护对应酒店的具体业务需求，一旦酒店业务发生变化，应该同时对系统的相应模块和功能进行调整和变化。
2. 系统数据维护。数据库是支持酒店业务正常运营的基础。系统正常运行中，系统管理员应经常注意数据库数据是否出现异常，对冗余数据，垃圾数据进行及时清理和整理，保持系统数据的准确有效。

（3）代码维护。代码维护是指更改扩展源系统代码，根据酒店现行业务添加或删除特定模块源代码。随着系统应用范围的不断扩大和应用环境的不断改变，系统开发者对源代码进行适当的增加、更新和删除代码。

（4）硬件维护。主要是指对部署该系统的计算机服务器进行日常维护和酒店前台客户端机器的维护，如机器的清洁，通风，维修设备故障，工作应由专职负责人定期进行，以确保正常工作有效。

# 结论

2018年1月，我开始了毕业论文和毕业设计的工作，到目前为止，该系统预期功能已基本完成。 从最初的如同一张空白的纸，到慢慢进入研究开发状态，然后逐渐理清思路，明确方向。整个的过程很难用语言说清楚。 经过几个月的努力，紧张和实质的毕业项目终于结束了。回想起一路来的经历和感受，虽然很辛苦，越到很多困难和问题，但最终还是圆满的完成了毕业设计，在这整个过程中，真的是有太多令人难忘的记忆和收获。

通过这个项目的开发我学习和掌握了很多的知识和技术，也收获了很多的经验与道理。现在回望整个项目的开发过程，我觉得对我的开发工作起到很大帮助的有在项目开始之初就选择使用Maven包管理工具，很好的解决了java jar包各个版本之间的依赖，减少了很多不必要的麻烦。还有一点我觉得对我的项目完成起到关键作用的是使用了SVN版本控制工具，在开发之初我便将我的项目托管在了我本地的SVN服务器上，我的每一次代码更新操作都被提交到了SVN服务器上。期间在我本地项目出现问题的时候很好的使用了SVN服务器上的代码来进行本地的覆盖更新，这样我便能够大胆的进行新功能的实现与代码的更新改变，完全没有会导致本地项目杂乱不清的后顾之忧。

本课题设计并实现基于了Java SSM（Spring，SpringMVC，Mybatis）和MySQL数据库组合的酒店管理系统。

# 致谢

历时两个多月，我最终完成了论文的写作和毕业设计。从最初的拿到论文题目到系统的设计实现，再到论文文章的写作完成。每一个阶段都充满的很多的挑战，但正是由于存在挑战，使得我完成了我整个大学阶段最大也是最得意的一个项目。通过这个项目的实现，我也发掘了我的潜力，也学会了很多以前不了解不熟悉的知识，更多的是学会了去尝试解决问题的勇气和方法。同时也收获了很多人的关心与帮助，很感谢我的指导老师在我的项目设计和论文写作上对我的建议与指导，还有我的同学朋友们甚至当我遇到问题难以解决时论坛上一些不知名的对我的帮助。

我的论文和设计仍有很多不成熟，待完善的地方，这也是和我目前的知识技术阅历有关，但是此次论文写作的过程已经让我受益终身了。我感受到了知识只有通过实践才能表达发挥出来它的价值，同时也能检验自己的掌握程度。学习同时也是一需要个持续的过程，只有通过不断的学习，才能掌握更深的技术，解决问题的方法和思路也会更加广泛。希望这次的经历能引领我在以后的工作与学习中继续努力，不断进步。

参考文献

[1] 王洪波. 基于Java的小型酒店管理信息系统的设计与实现[J]. 贺州学院计算机科学与信息工程学院,2016.

[2] 贾蓓,镇明敏,杜磊. Java Web整合开发实战:基于Struts 2+Hibernate+Spring [M] 清华大学出版社2013.7

[3] 杨旭. J2EE企业级开发.[M].北京:清华大学出版社2016.3.

[4] 刘美华. 西域文化博览园资源库平台的设计与实现[J] 2017

[5] 许侠. 基于门禁开发与实现非接触式IC卡制作及整合管理系统[J],2004

[6] 李鹏. 基于.NET的酒店管理系统的设计与应用.[J] 2013

[7] 萨师煊,王珊. 数据库系统概论.北京:高等教育出版社[M], 2005.7.

[8] 王冰燕. 酒店管理系统的设计与实现.[D] 2012

[9] 朱福喜,余振坤. J2EE 经典实例详解[M]. 北京:人民邮电出版社, 2006, 26-183.

[10] 张承琳. 酒店管理系统的设计与实现[J] 2014

[11] 杨开振. Java EE 互联网轻量级框架整合开发—SSM框架(Spring MVC+Spring+MyBatis)和Redis实现[M]2017.7.1