**基于SSM框架的酒店管理系统的设计与实现**

作者姓名：\*\*\* 专业班级：软件工程\*班 指导教师：唐云

**摘 要**

随着计算机技术的高速发展，酒店管理方面从传统的纸质记录到计算机网络系统管理，管理与实时数据融合、业务全贯通、数字化管理成为现实。传统的酒店管理模式和应用、现代信息化建设的需求已经超越了服务系统的能力，因此我们需要开发一种更高效、更直观的现代管理技术来取代传统的管理方式。酒店管理系统是一种能够满足这一需求的理想选择，它具有高利用率、高可用性、低成本等特点，因此被广泛应用于绝大多数酒店。

本文首先介绍了基于 SSM框架、JSP开发的酒店管理系统，其前端框架所采用的是常用的LayUI框架，数据库部分选择了当下盛行的、可靠的MySQL数据库。然后对其进行了需求分析与整体设计，最后利用主流的 Idea集成开发平台完成了该系统的开发。系统实现了管理员登录、房间管理、住客管理、 VIP管理、打印当前信息等功能。在经过测试运行之后，发现该酒店管理系统功能完善、运行稳定，可以较好地满足实际应用的需求。

**关键词：酒店管理系统；SSM；JSP；LayUI；MySQL**

**The Design and Implementation of The Hotel Management System With SSM Framework**

Abstract:With the rapid development of computer technology, hotel management has transitioned from traditional paper-based records to computer network system management. The integration of management and real-time data, seamless business operations, and digitized management have become a reality. Traditional hotel management models, applications, and the current information construction demands have surpassed the capabilities of existing service and system infrastructure.Therefore, developing a more efficient and intuitive modern management technology is urgently needed. Hotel management systems can effectively meet these requirements. The features of high utilization, high availability, low cost, low energy consumption, rapid deployment, and easy scalability make hotel management systems the choice for the majority of hotels.

This article first introduces a hotel management system based on the SSM framework and developed using JSP. The front-end design utilizes the commonly used LayUI, while the database employs the popular MySQL database. The system's requirement analysis and overall design are subsequently presented in the article, followed by the development of the system using the mainstream Integrated Development Environment (IDE) called Idea. The system implements functionalities such as administrator login, room management, guest management, VIP management, and printing of current information. Upon conducting tests and implementing the system, it was discovered that it possesses a multitude of features, exhibits stability, and is proficient in fulfilling the practical application demands.

Key words: Hotel Management System;SSM;JSP;LayUI;MySQL

**目 录**

[第1章 前 言 1](#_Toc145834465)

[1.1 论文研究的目的和意义 1](#_Toc145834466)

[1.2 国内外研究综述 1](#_Toc145834467)

[1.3 论文研究的内容和取得的成果 2](#_Toc145834468)

[1.4 文章组织结构 3](#_Toc145834469)

[第2章 网站开发技术简介 4](#_Toc145834470)

[2.1 JSP动态网站开发技术简介 4](#_Toc145834471)

[2.2 SSM框架简介 5](#_Toc145834472)

[2.2.1 SpringMVC框架 6](#_Toc145834473)

[2.2.2 Spring框架 7](#_Toc145834474)

[2.2.3 Mybatis框架 7](#_Toc145834475)

[2.2.4 SSM框架整合 7](#_Toc145834476)

[2.3 前端LayUI框架简介 8](#_Toc145834477)

[2.4 MySQL数据库技术简介 9](#_Toc145834478)

[第3章 系统分析 10](#_Toc145834479)

[3.1 可行性分析 10](#_Toc145834480)

[3.1.1 技术可行性 10](#_Toc145834481)

[3.1.2 经济可行性 10](#_Toc145834482)

[3.1.3 市场可行性 11](#_Toc145834483)

[3.1.4 法律可行性 11](#_Toc145834484)

[3.2 需求分析 11](#_Toc145834485)

[3.2.1 用户登陆模块 12](#_Toc145834486)

[3.2.2 系统管理模块 12](#_Toc145834487)

[3.2.3 住客管理模块 13](#_Toc145834488)

[3.2.4 房间管理模块 13](#_Toc145834489)

[3.2.5 会员管理模块 13](#_Toc145834490)

[3.2.6 数据表打印模块 13](#_Toc145834491)

[第4章 系统设计 14](#_Toc145834492)

[4.1 概要设计 14](#_Toc145834493)

[4.1.1 用户登录设计 14](#_Toc145834494)

[4.1.2 系统管理设计 15](#_Toc145834495)

[4.1.3 住客管理设计 15](#_Toc145834496)

[4.1.4 房间管理设计 16](#_Toc145834497)

[4.1.5 会员管理设计 16](#_Toc145834498)

[4.1.6 数据表打印设计 17](#_Toc145834499)

[4.2 数据库设计 18](#_Toc145834500)

[第5章 系统实现 21](#_Toc145834501)

[5.1 用户登录功能实现 21](#_Toc145834502)

[5.2 系统管理功能实现 22](#_Toc145834503)

[5.3 住客管理功能实现 24](#_Toc145834504)

[5.4 房间管理功能实现 27](#_Toc145834505)

[5.5 会员管理功能实现 29](#_Toc145834506)

[5.6 数据表打印功能实现 31](#_Toc145834507)

[第6章 系统测试 32](#_Toc145834508)

[6.1 功能性测试 32](#_Toc145834509)

[6.1.1 功能测试 32](#_Toc145834510)

[6.1.2 界面测试 33](#_Toc145834511)

[6.2 安全性测试 33](#_Toc145834512)

[结 论 34](#_Toc145834513)

[参考文献 35](#_Toc145834514)

# 前 言

本章一共包含四段内容，第一段将会介绍本论文的写作背景、目的及意义，而第二段是综合了国内对酒店管理系统的发展情况的研究综述，第三段是主要论述了本论文写作和开发过程运用到的关键技术和取得的主要成果，最后一段则是为读者能够更好地理解本论文而梳理的文章组织结构。

## 论文研究的目的和意义

计算机在酒店管理的应用已扩展至各部门，酒店管理系统已成为酒店现代管理的重要内容。通过使用计算机系统来对酒店的事务进行管理，不仅解决了人工管理存在的弊端，而且节约了人力与物力（赵晓彤，2014）。

近年来，随着计算机事业的迅速发展，利用计算机进行信息管理已成为一个新兴的研究领域。当今，无论是图书馆、学校、银行，还是政府、交通这些部门都在逐渐采用计算机智能化管理。所以推行酒店管理智能化是大势所趋也势在必行。一个优秀的酒店管理系统不仅能有效地提高管理水平，提高管理效率；还能提高决策分析能力，降低管理成本，提升员工和客人的满足感。

本文房间管理模块可以直接更新当前房间的状态，方便客人和工作人员第一时间了解房间情况并安排入住。用户模块可以纪录用户信息并登记VIP信息。在工作交接时还能对当前的住户和房间信息进行打印。不仅提高了酒店员工在平时工作中的效率，减轻了工作负担，更让入住酒店的用户拥有了归属感，能极大地提升回头客的比例，是人性化和理性化的结合。

## 国内外研究综述

随着全球化的加速，国内酒店行业在与国际酒店业的市场竞争中并没有明显的优势。为了与国际酒店业实现互动融合，国内酒店管理者应该积极学习国外酒店管理的优秀理念和模式，结合本土特色，以保持市场竞争的优势。为了能够更好地发展和实现对酒店客房的专业化管理和信息系统运营，酒店客房的管理和信息系统的管理信息化和服务合理化已经逐渐成为了酒店管理和经营者的必然选择（吴官学，2018）。

随着酒店行业的快速发展，管理系统已成为酒店提高自身管理水平，保持持续稳定发展的重要手段。酒店管理系统作为一种现代化管理工具，在整个酒店行业的应用越来越广泛，其发展也呈现出多元化、多样化的趋势。在国外，美国是酒店信息化最早的国家，也是最先进的国家，美国有约4400家酒店，其中大约有80%的酒店拥有自己独立的信息化系统(Alarifi, A.,Al-Surmi, M.,2016)。我国的酒店行业虽然起步较晚，但发展速度却非常快，目前正处于快速发展的阶段。近年来，国内很多高星级酒店引进了国际先进的管理软件和系统平台，如希尔顿、香格里拉等国际知名酒店集团也都相继推出了适合国内市场需求的管理系统(Kim, H.,Lee, D.,2017)。

在国内酒店管理系统的研究起步较晚，主要集中在国外著名酒店管理公司的引进，以及国内一些大型连锁酒店集团的开发应用。目前国内大部分酒店管理系统以国际著名品牌的管理软件为主要学习对象，主要目标是提高酒店的服务水平和经济效益，在功能、界面等方面均有模仿，但在数据库建设、系统开放性和安全性方面却存在一定的不足(Chou, H. H.,Li, Y.,2017)。

随着新一代信息技术在中国的飞速发展，酒店管理系统必将朝着更开放、更灵活、更专业和更具个性化的方向发展。同时随着网络技术和智能设备的不断成熟，以及客户需求不断增加，酒店管理系统也将越来越智能化和人性化。

## 论文研究的内容和取得的成果

论文将着重探讨在SSM框架下，采用JSP技术开发、LayUI框架前端设计的酒店管理系统，对其进行了全面完善的需求分析与整体设计，最后利用主流的 Idea集成开发平台完成了该系统的开发，并采用MySQL数据库进行数据库开发。通过论文研究，本文所取得了以下研究成果如下：

1. 深入认识和剖析了当前时代酒店管理系统的重要性和必要性，完善了其需求分析和总体设计。
2. 进一步了解和掌握了软件开发的严格流程，例如前期的市场调研、需求分析，中期的代码编写和后期的文档整理等。对大部分企业开发的前后端分离流程也有了一定了解。
3. 深入了解和学习了LayUi等优秀的几个前端框架并应用与酒店管理系统中，提高了代码的简便性和易读性。
4. 设计和实现了能够让用户简单、快速完成操作的前端页面，提高了页面交互性，提升了用户体验。
5. 继续深入学习和使用了Java、JavaScript语言开发。
6. 了解和掌握了SSM框架编程思想，对使用SSM框架的四层架构（Dao层、Service层、Controller层和View层）进行编程有了深厚理解。
7. 本文系统主要使用Java和JavaScript语言进行开发，采用JSP动态网站开发技术，在前端框架上选择的则是现在主要面向后端开发者的框架LayUI，数据库采用的是可靠的、流行的MySQL数据库，后端上采取的是传统的SSM框架开发，采用前后端整体集成开发。
8. 本文按照用户需求，准确、完善地完成了各个功能模块的开发。

## 文章组织结构

本文严格参照软件工程思想，分阶段对开发的酒店管理系统进行详细说明。全文共分6章，各章内容简介如下：

第1章的主要内容为本系统的研究目的和意义，国内外研究现状和论文研究的内容与取得的成果。

第2章的主要内容为系统中介绍使用的技术。从技术名称到技术原理详细介绍了技术所运用的知识点和技术优势。其中设计到的技术点有SpringMvc框架、Spring框架、Mybatis框架、前端LayUI框架和MySQL数据库。

第3章主要内容是对本系统进行需求分析和可行性分析，分别从技术、经济、市场、法律几个方面进行了可行性论证，分用户登陆、系统管理、住客管理、房间管理、会员管理、数据表打印六个模块进行需求分析。

第4章主要内容是系统的概要设计和数据库设计方面。同样是划分为六个功能模块进行详细介绍并展示数据库结构。

第5章主要内容是系统的详细设计与实现，并展示了系统运行效果图，且详细描述了功能模块运行原理和流程。

第6章主要内容是进行系统测试。进行了功能测试、界面测试这两个功能性测试并进行了合理的安全性测试。

# 网站开发技术简介

本章将对在开发酒店管理系统中用到的主要技术原理进行介绍。在本系统的设计与开发工作中，主要用到了Java和JavaScript语言，采用JSP动态开发，在前端上选择LayUI框架，数据库采用的是MySQL，后端上采取的是传统的SSM框架开发。

## JSP动态网站开发技术简介

JSP( java server pages) 是在sun microsystems公司主导下，由从多公司一起参与构建的一种动态网页设计技术标准，其目的在于快速、便捷、简易地构建一个基于web的网页应用程序。该技术的脚本语言是java语言，与其他动态网页设计技术相比较，优势极大，尤其是跨平台问题上，更为有效，在windows、linux、soloris等平台下都可以正常运行（方忠，2000）。

JSP是建立在Java Servlet基础之上并进行了优化改进的一种服务，它能够使得在Java运行环境下开发WEB程序更加高效和简便。主要优点有以下几点：

（1）能生成且显示分离内容，运用HTML等语言动态化编辑页面。

（2）能有效适应各种平台并保持高执行效率。

（3）具有可重用组件来处理复杂程序。

（4）封装java组件使页面开发更简化。

（5）可拓展性强。

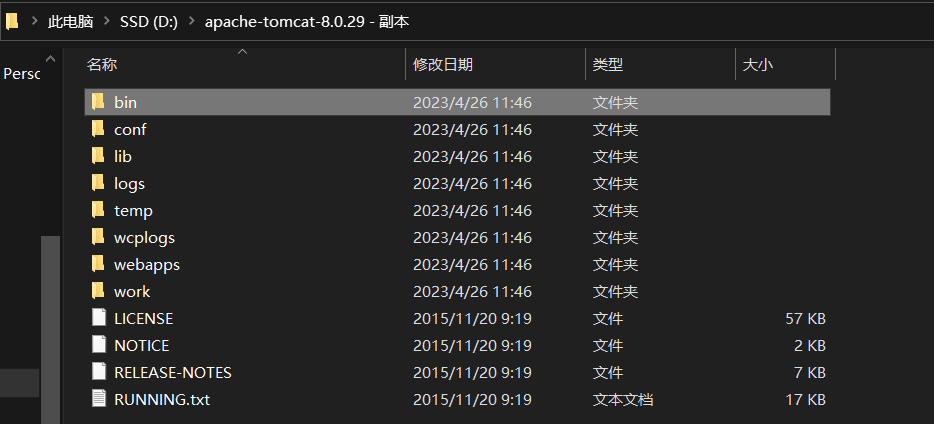
JSP 是一种以java servlet技术为基础，经过大量改进后构建的一种理想的 web 应用开发框架，是一种开放性高、跨平台性强的结构，能够使编译后的动态网页程序在不影响 JSP 基本程序的同时，在WEB服务器、多个操作平台上、其他组件中简单地实现升级或切换，是一种更为适应当前web应用程序需求的技术，基于JSP技术设计的动态网页程序较 ASP、PHP等的执行效率更高，维护管理更为简单，是未来动态网页设计的主流技术（蒋治学，2020）。

而本系统采用Tomcat作为Web应用服务器，Tomcat的优点如下：

1. 易拓展。由于该servlet容器运行在服务器上，无需考虑分布式环境下维护session状态等问题。
2. 安全性高。由于后端服务器被设置为无法从外部访问，黑客只能通过访问前端的web服务器来尝试攻击，而无法直接访问后端的应用服务器。这种设置显然大大提高了系统的安全性。
3. 在兼具拓展性和安全性的同时，Tomcat运行依旧稳定。

Tomcat的使用则先需要配置好前置Java，在官网完成下载后解压如图2-1所示。

图2-1 Tomcat配置图



同时应该配置好相对应的环境变量如图2-2所示。

在配置好以上文件后，我们应该进入到集成开发平台例如Idea内，进行Tomcat和集成开发平台的绑定以方便运行，如图2-3所示。

完成Idea的绑定后，我们就可以进入到项目的部署环节了。Idea给我们提供了非常方便的一键部署功能，在完成Tomcat配置后，点击部署Idea则会自动地将我们的项目以war包的形式部署在Tomcat服务器上。

## SSM框架简介

在后端编写过程中，虽然有许多可供选择的框架，但它们的根源都是按照“MVC”（模型-视图-控制器）的设计模式进行的。这种设计模式中，三个容器各自承担不同的功能。

模型层主要负责业务模型和数据模型的定义和处理，包括对数据库的查询和写入。因此，模型层是框架的核心层。

视图层则根据模型的创建情况，允许用户通过前台界面访问模型。视图层一方面支持用户的输入，另一方面响应用户的请求。

控制层主要是根据用户的请求信息，选择合适的模型来响应，并不直接参与数据的处理过程，仅仅是在模型层和视图层起信息传递的纽带作用（曹华山，2021）。

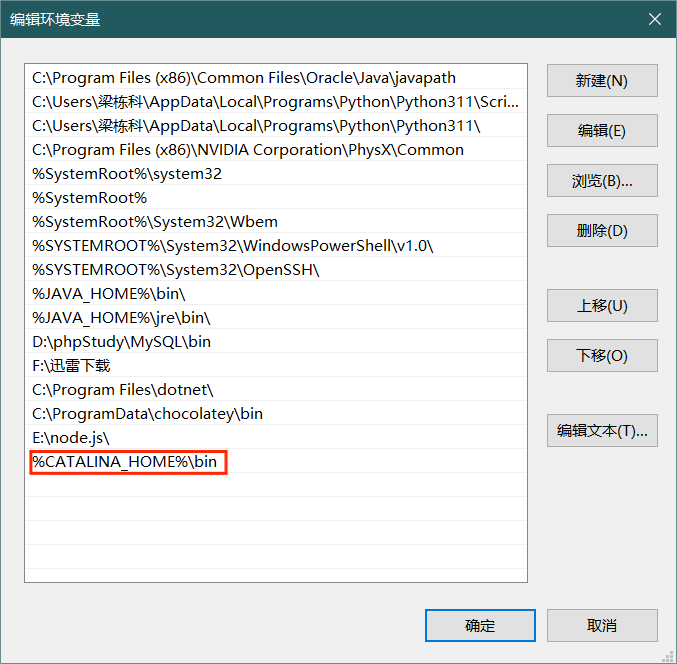
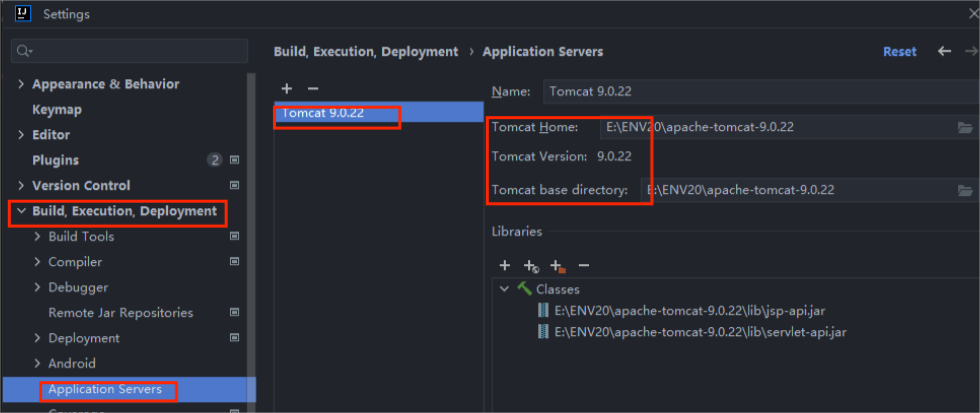


图2-2 环境变量

图2-3 idea设置



目前，SSM框架是最广泛使用的Web开发框架。接下来，本文将详细介绍这三个框架。

### SpringMVC框架

基于Spring框架和MVC设计模式构建的SpringMVC，是一种常用后端框架，主要用于Web应用的开发。

使用该框架可以快速配置Web开发过程中所需的JSP视图。与Struts框架在表面上相似，但实际上它们的实现原理完全不同。

Struts框架主要实现的是“类”级别的拦截，即每个类对应一个请求，而SpringMVC框架则实现了“类中的方法”级别的拦截，即每个方法对应一个请求，并且每个方法对应一个URL。(Chen, C. H.，Liao,Y. C.，2018)。

### Spring框架

开源的Java应用程序框架Spring，为开发人员提供了一系列工具和组件，帮助构建高效、可扩展、易于维护的企业级应用程序。IoC和AOP是Spring框架的核心概念，它们有助于实现松耦合、模块化的应用程序设计。

Spring框架的优点在于灵活、可拓展等。在AOP中主要包含五种增强处理的方法，在业务方法中如果需要植入这些增强方法，都需要在Spring的配置文件中进配置，为了减少配置的工作量，我们可以使用 AspectJ框架中的注解方式来标注增强方法（邱丹萍，2020）。

### Mybatis框架

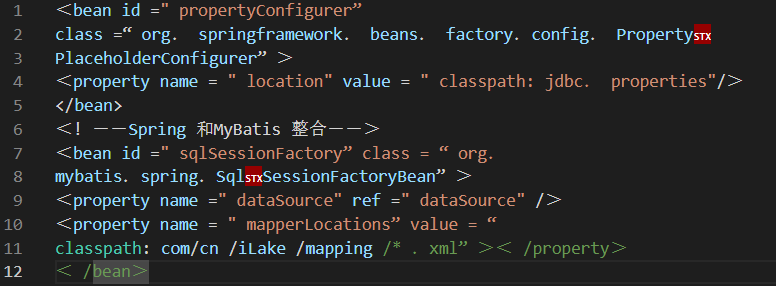
一种基于Java语言的持久层框架，将SQL语句和Java对象之间的映射关系定义在XML文件中，实现数据库操作和Java对象之间的解耦。

运行原理：先解析配置文件中的SQL语句和映射关系，生成MappedStatement对象，包含SQL语句、参数映射关系、结果映射关系等信息。当Java程序需要执行SQL语句时，MyBatis会根据SQL语句的ID从MappedStatement对象中获取信息，将参数传递给SQL语句，执行SQL语句并将结果映射为Java对象返回给程序(Alomari, M., and Altahat, S.,2019)。

### SSM框架整合

以上分析了SSM三个框架的技术特征，但是要实现将这三个框架在系统开发中利用起来，就必须将三个框架有机地整合起来。当我们将Spring和Mybatis整合时，可以将Mybatis核心配置文件中的数据源、事务、映射接口路径等配置打包交给Spring配置文件，而Mybatis仅需完成配置类别名的任务。这个过程中，展示了Spring框架的核心作用。在整合 Mybatis 和 SpringMVC 的过程中，需要在 SpringMVC 的配置文件中配置控制器、适配器、处理器和视图解析器，同时还需要在 web.xml 文件中配置前端控制器和过滤器（耿祥义，张跃平）。Spring 和MyBatis 整合关键代码如图2-4所示。

图2-4 Spring 和MyBatis 整合关键代码图



SpringMVC和Spring整合关键代码如图2-5所示。

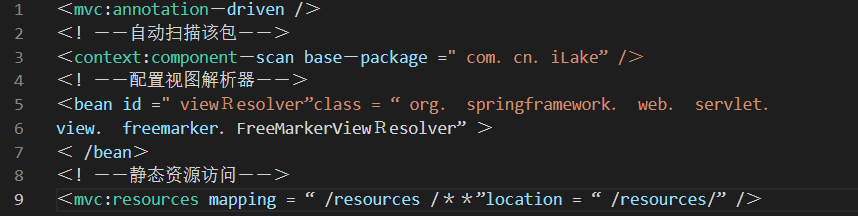


图2-5 SpringMVC和Spring整合关键代码图

## 前端LayUI框架简介

LayUI是用于搭建用户可视化的具体操作界面的框架。该前端框架采用自身轻量级模块化规范而成为了超级轻量级框架，风格轻盈简约且极易上手。虽然LayUI为小众前端开发框架，但是经过近十年来长时间的精雕细琢，LayUI的技术逐渐成熟。在主流标准尚未完全普及的前 ES5 时代，LayUI的开发主题就只有“极简”这种返璞归真的执念，也逐渐成为了LayUI的标志和精神，一直沿用至今（秦不凡，2020）。LayUI普通模块示例如图2-6所示。

LayUI完全可以将其视为像使用普通 API一样管理模块，在此基础上，各种组件的承载也变得十分轻松，使用者完全能够游刃有余地存在以Web为宿主、原生态的HTML/CSS/JavaScript 的开发模式当中。并且LayUI 也不是仅仅只是一个模块加载器，



图2-6 普通模块示例图

它是一套相对完整的 UI 解决方案。区别于 Bootstrap，Layui 的组件更倾向于 JavaScript 的动态化渲染，而不仅仅是 HTML+CSS本身的静态化处理。LayUI还为之提供了相对完善的、统一的 API，只需用户在使用前稍加熟悉，便可轻松应对各种交互。

## MySQL数据库技术简介

MySQL是一款高效、跨平台、安全可靠的关系型数据库管理系统，它与主流程序语言例如 PHP, JAVA有密切的联系。当前，大部分互联网上的中小规模网站，MySQL因其小巧、高速、低成本等优点而备受青睐，而选择使用 MySQL作为后台数据库（胡孔法，2020）。除此之外，开源的特性也让 MySQL数据库受到了众多公司的青睐。除了上面提到的这些好处， MySQL数据库还有一个很有吸引力的好处，那就是它是一个完全免费的工具，不需要用户付费。MySQ具有高安全性、低成本的特性。MySQL会对全过程全密码进行加密，权限管理和密码系统在安全性的同时也兼顾了灵活性。对于企业来说MySQL是一款免费开源程序，对于个人用户而言MySQL更是一款直接的免费软件。

索引也是MySQL数据库一个重要的特征。Mysql底层提供了B+树的数据结构，与以往全表扫描的开发方式不同，B+树索引相对于Hash索引，支持范围匹配和排序等操作。而使用B+树则可以显著地提高MySQL的使用性能。

# 系统分析

本章分为两节，第一节主要从技术可行性等三个方面来研究酒店管理系统的研发可行性，第二节则是项目具体的各个模块间的功能的需求分析。

## 可行性分析

酒店管理系统是一种利用计算机技术对酒店日常运营进行自动化管理的软件系统，其主要功能包括住客管理、系统管理、住房管理、会员管理、数据表打印等。下面是酒店管理系统可行性分析的一些方面。

### 技术可行性

技术实力：酒店管理系统需要运用计算机技术进行开发，因此需要具备相关的技术实力和资源支持。开发者需要具备软件开发、数据库设计等方面的技术能力，并且需要有足够的硬件设备和软件工具支持。

数据库设计：酒店管理系统需要对酒店的客户、房间、订单、财务等数据进行管理和处理，因此需要合适的数据库结构和数据管理方案。

界面设计：需要友好的用户界面，方便用户操作。因此，需要进行合适的界面设计，包括界面布局、颜色搭配、图标设计等方面，以提高用户体验和使用效率。

安全和数据保护：评估酒店管理系统的安全性和数据保护措施。系统应该采取安全措施，例如访问控制、数据加密等，以确保系统的安全性。同时，还需要定期进行安全评估和漏洞修复。

硬件设备和基础设施：需要对酒店管理系统的硬件设备和基础设施进行评估，以确保其符合要求

根据以上五个方面，经研究，从技术可行性方面分析，该系统开发切实可行。

### 经济可行性

酒店管理系统的开发和运营需要包括人力、物力、财力等资源的一定量的投入。因此，需要进行经济可行性分析，确定系统的投资回报率、成本效益等指标，以确保系统的经济效益和可行性。而本系统开发成本在当今的软硬件条件下来说非常低，完成项目开发后，不仅可以提高酒店管理效率，还可以节省一些人工管理费用。经研究，从经济可行性方面分析，该系统值得开发。

### 市场可行性

为了满足酒店管理的实际需求，酒店管理系统必须符合市场需求，具备良好的用户体验和市场竞争力。因此，需要进行市场可行性分析，确定市场规模、市场需求、竞争状况等指标，以确保系统的市场可行性。近年来，随着计算机事业的迅速发展，几乎所有行业都逐渐采用计算机智能化管理。所以研发酒店管理系统也是顺势而行。一个优秀的酒店管理系统不仅能有效地提高管理水平，提高管理效率；还能提高决策分析能力，降低管理成本，提升员工和客人的满足感。经研究，从市场可行性方面分析，该系统值得开发。

### 法律可行性

该系统在设计、实施和运营过程中需要符合相关的法律法规和行业标准。以下是一些与酒店管理系统相关的法律和合规性方面的详细阐述：

数据隐私和保护：酒店管理系统通常会涉及处理客户个人信息，如姓名、联系方式、信用卡信息等。在设计和使用系统时，必须遵守个人数据保护法规。为了防止未经授权的访问、数据泄露或网络攻击，我们可以采取一系列措施，例如使用防火墙、加密通信、安全认证和访问控制等手段。

网络安全：酒店管理系统通常是基于网络的，因此需要确保系统的网络安全性。这包括使用防火墙、加密通信、安全认证和访问控制等措施，以防止未经授权的访问、数据泄露或网络攻击。

地方法规和行业标准：不同地区可能有特定的法规要求和行业标准，酒店管理系统应符合这些特点。经研究，从法律可行性方面分析，该系统开发切实可行。

## 需求分析

为了更好的对系统有一个全局的认识，本文使用了用例图来对业务流程进行解说，图中展示了管理员进入管理系统，管理员通过不同的路径进入相应的目标页面，进行相关的具体操作。在进入管理系统后，管理员可以开始管理工作，选择系统管理、住客管理等不同的管理功能去完成相应的工作详细的项目业务流程使用的用例图表示如图3-1所示。

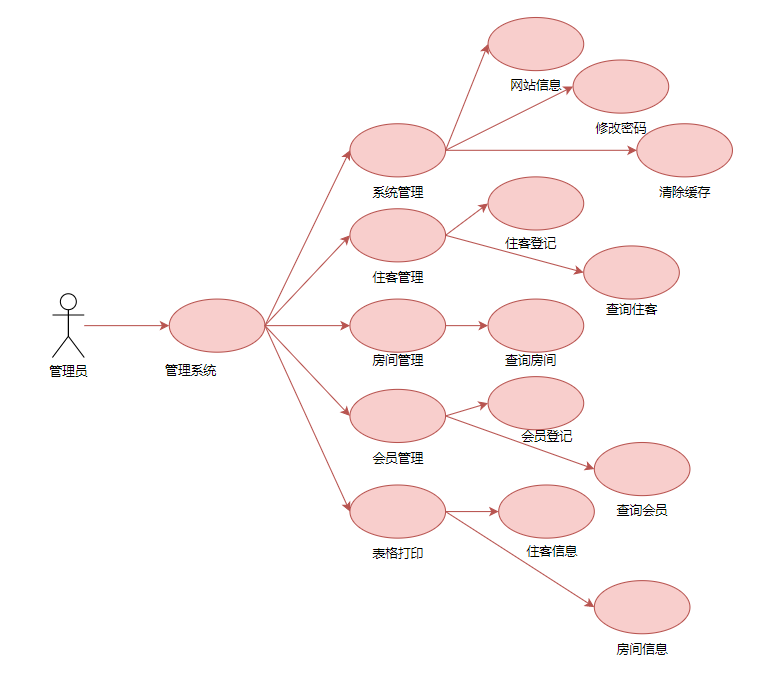


图3-1 用例图

### 用户登陆模块

酒店管理系统中登录功能是本系统中的基础功能模块之一。为了确保酒店管理系统的相对安全，所有工作人员第一次进入本系统都只能由上级告知本系统的账号密码，而不能通过常规的登录注册来进入管理系统。

### 系统管理模块

酒店管理系统中系统管理模块同样是本系统中的基础功能模块之一。该模块主要分为三部分。

第一部分是网站信息，这部分是介绍本系统的名称、标题、版权等信息，由于信息长期固定，所以不允许在界面直接更改。

第二部分是修改密码，这部分是允许用户更改登录系统的账号密码。

第三部分是清除缓存，在系统运行过程中，可能由于几个月长时间地连续运行，导致浏览器cookie爆满而影响系统功能的正常运行，所以添加了这个一键清除浏览器cookie的功能。

### 住客管理模块

酒店管理系统中住客管理是本系统中的重要功能模块之一，该模块主要分为两部分。

第一部分是住客入住时，酒店工作人员通过本系统进行住客入住信息例如名字、身份证号、手机号等的登记入册，且记录好住客入住的时间。

第二部分是住客入住信息的查看。可以通过搜索手机号或者下翻菜单栏的方式浏览住客信息，从而决定删除和修改住客的详细信息。

### 房间管理模块

房间管理同样是本系统中的重要功能模块之一。

该模块主要是对房间信息进行对接管理，进行增删改查。此外，该模块还支持直接修改和更新房间的当前状态，这使得前台工作人员可以快速查看哪些房间处于可入住状态，以便将其分配给住客，从而提高工作效率。

### 会员管理模块

酒店管理系统中会员管理同样是本系统中的重要功能模块之一。该模块主要分为两部分。

第一部分是酒店工作人员通过本系统确认住客同意注册本酒店的会员后进行住客例如名字、身份证号、手机号等的会员信息登记入册，且记录好会员开通的时间和住户会员到期的时间。第二部分是会员信息的查看。可以通过搜索手机号或者下翻菜单栏的方式浏览会员信息，从而决定删除和修改会员的详细信息或者延长会员的持续时间。

### 数据表打印模块

酒店管理系统中数据表打印同样是本系统中的基础功能模块之一。由于酒店管理过程中，经常涉及到各种警察查房、管理层查看入住信息等情况，特设计了本模块以快速地完成将现阶段酒店的情况以Ecexl表单形式打印的功能。该模块主要分为两个部分。

第一部分是房间信息表的打印。第二部分则是住户信息表的打印。

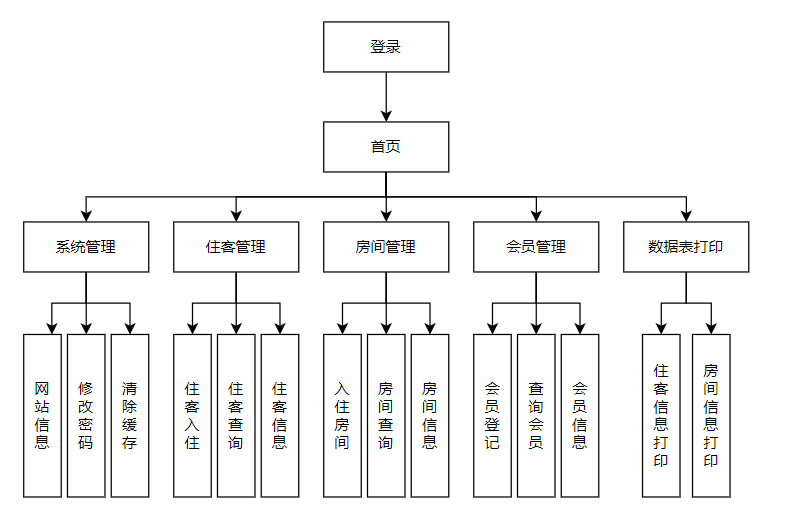
# 系统设计

本章分为两节，第一节主要从各个模块出发阐述酒店管理系统的概要设计，第二节则是项目具体的数据库详细设计阐述。

## 概要设计

本小节对该系统的总体设计进行了介绍，并分模块进行了详细的流程图介绍。本系统系统流程图如图4-1所示。

图4-1 系统结构图



### 用户登录设计

首次进入本系统的工作人员必须由上级告知账号密码，不能通过常规的登录注册进入管理系统。因此，用户需要在主页面填写账号密码进行登录。如果账号密码与数据库中的用户账号密码吻合，则可以进入主页页面。如果不吻合，则会提示账号或密码错误，需要用户重新输入正确的账号密码。用户登录注册流程图分析如图4-2所示。

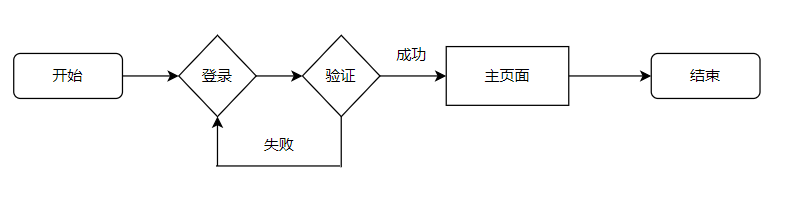
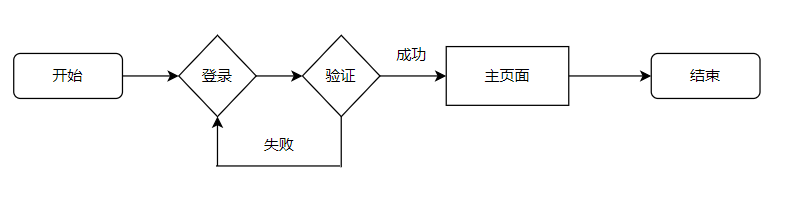


图4-2 用户登陆注册流程图



### 系统管理设计

系统管理的三个功能分别是网站信息、修改密码和清除缓存。用户在选择进入系统管理后，会在此选择进入这三个功能其中一个。网站信息这部分是介绍本系统的名称、标题、版权等信息，由于信息长期固定，所以不允许在界面直接更改。修改密码，这部分是允许用户更改登录系统的账号密码，在修改密码符合条件成功后会自动跳转到网站信息页面。清除缓存是一键清除浏览器cookie的功能，成功清除后会自动跳转到网站信息页面。系统管理流程图分析如图4-3所示。

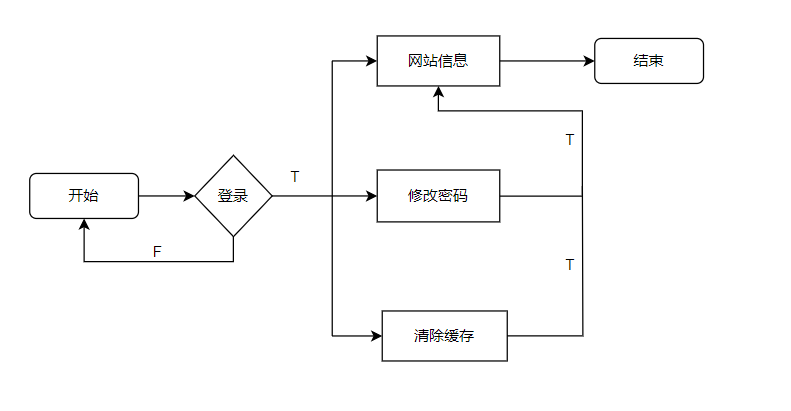
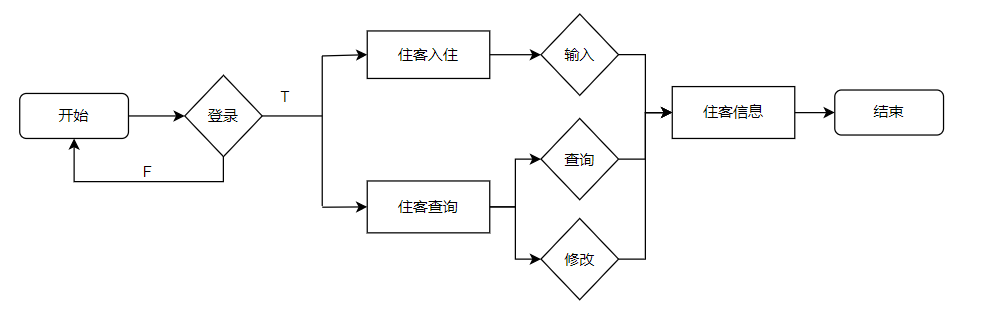


图4-3系统管理流程图

### 住客管理设计

住客管理的两个功能分别是住客入住信息填写和查询住客信息并修改。用户在选择进入住客管理后，会在此选择进入这两个功能其中一个。住客入住信息填写部分是进行例如名字、身份证号、手机号等的登记入册，且记录好住客入住的时间。查询住客信息这部分是允许用户通过手机号对住客信息进行查询匹配，匹配成功后会进入该住客的信息页面，进行增删改查。住客管理流程图分析如图4-4所示。

图4-4住客管理流程图



### 房间管理设计

房间管理的主要功能是房间信息填写和查询房间信息并修改。用户在选择进入住客管理后，会在此选择使用查询功能或者滑动自己查找。房间信息填写部分是进行例如房间号、房间类型、房间图片等，且记录好房间的当前状态以便住客入住。用户可以通过输入房间号查询匹配房间信息，匹配成功后可以进入该房间的信息页面，并且可以进行添加、删除、修改等操作。房间管理流程图分析如图4-5所示。

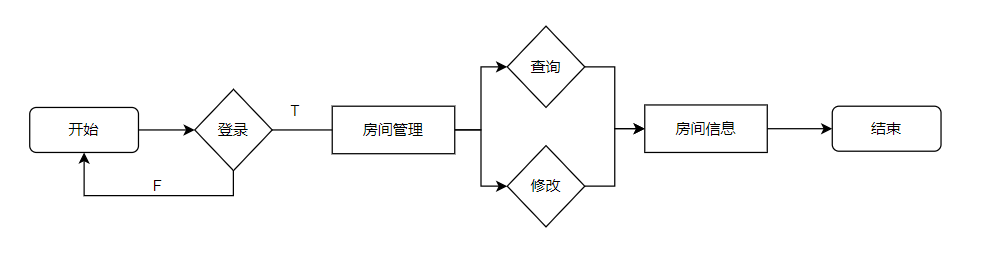


图4-5 房间管理流程图

### 会员管理设计

会员管理的两个功能分别是会员信息填写和查询会员信息并修改。用户在选择进入会员管理后，会在此选择进入这两个功能其中一个。会员信息填写部分需要填写会员的个人信息，包括姓名、身份证号码、手机号码等，并记录会员开通的时间。查询会员信息这部分是允许用户通过手机号对会员信息进行查询匹配，匹配成功后会进入该住客的会员页面，同时可以对住客信息进行添加、删除、修改等操作。会员管理流程图分析如图4-6所示。

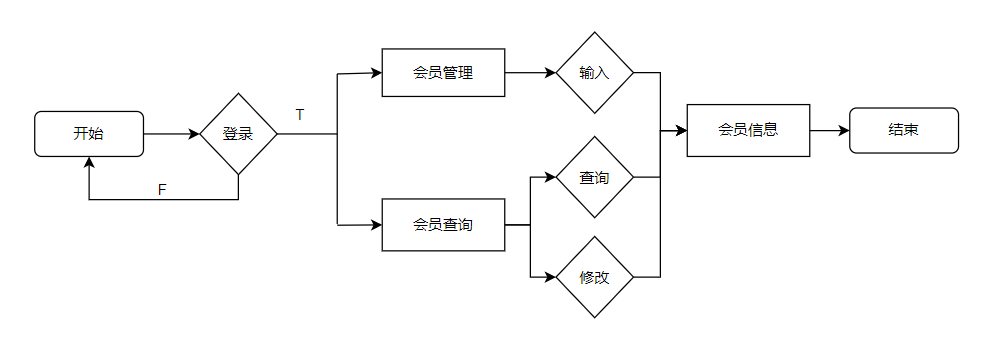


图4-6 会员管理流程图

### 数据表打印设计

数据表打印的具体功能是快速地将现阶段酒店的情况以Ecexl表单形式打印，用户可以选择打印住客信息表或者房间信息表。用户在选择进入数据表打印后，会在此选择打印这两个表中其中一个。点击打印后，数据表将以excel形式下载到本地。数据表打印流程图分析如图4-7所示。

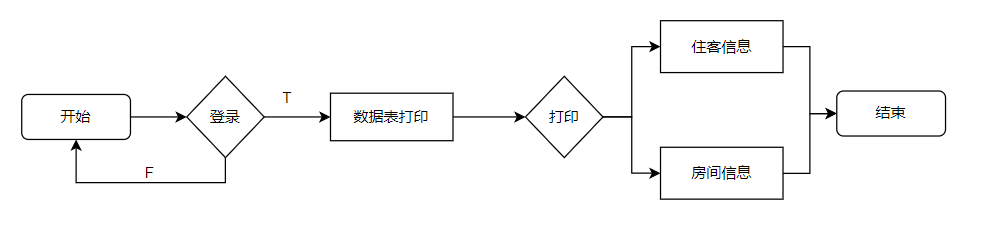


图4-7 数据表打印流程图

## 数据库设计

在当今这个数据时代，数据库设计非常重要，因为它直接关系到系统的数据存储和管理。优秀的数据库设计不仅可以提升系统的性能和安全性。

本系统的数据库的结构和实体说明如下：共创建了管理员实体、会员实体、住客实体、房价实体共四个实体。

管理员实体：用于存储管理员的账号、密码的实体，同时管理员实体可以操作会员实体、住客实体、房价实体这三个实体。管理员可以通过添加、删除、修改和查询会员、住客和房价的属性来执行操作。

会员实体：用于储存会员名、会员手机号、会员身份证以及会员注册、到期时间等的实体。

房间实体：用于储存房间号、房间价格、房间信息以及房间状态、房间类型等的实体。管理员实体和被操作实体之间存在多对多的关系。同样地，住客实体和被操作实体之间也存在多对多的关系。

住客实体：用于储存住客名、住客手机号、住客身份证以及住客订房、退房时间等的实体。住客实体可以通过对房价实体中的属性进行修改来实行操作功能。被管理员实体操作，与管理员实体是多对多的关系。

所有属性在后面的表中列出，得到该在线互动学习平台的E-R图，如图4-8所示。

本酒店管理系统根据系统需求，共设置了四张数据表，分别是：admin、guests、home、vip四张。

1. admin表，储存了酒店管理人员登陆注册的账号（username）、密码（password）的数据表。admin表结构如表4-1所示。
2. guests表，储存了住客序号（id）、住客名（name）、住客性别（sex）、住客手机号（phone）、住客身份证号（card）、住客入住时间（enterTime）、住客退房时间（exitTime）、房间类型（h\_type）、房间号（num）的数据表。guests表的结构如表4-2所示。

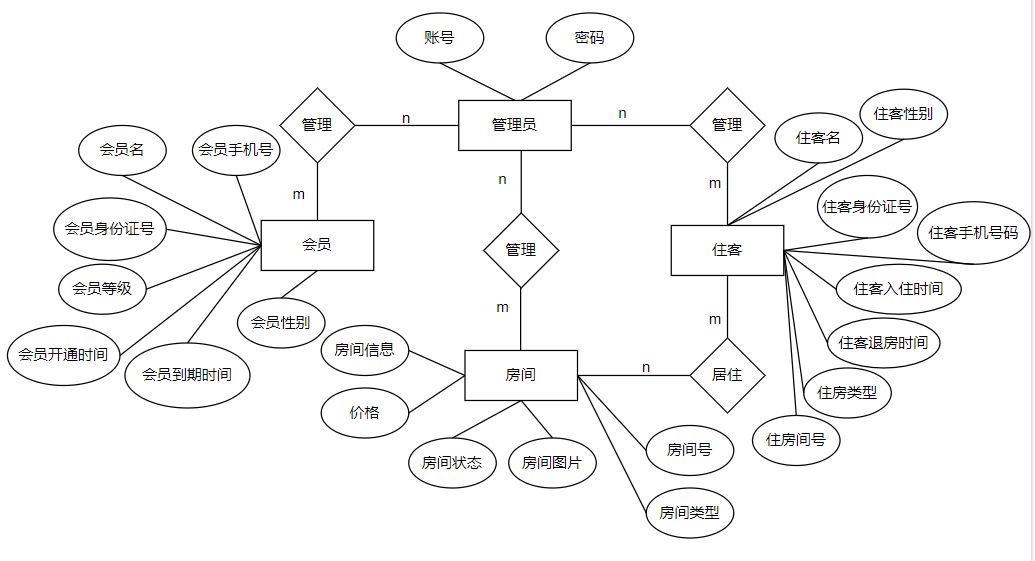


图4-8酒店管理系统 ER图

表4-1 admin表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **主键** | **允许空** | **注释** |
| 1 | username | varchar | 10 | 是 | 否 | 用户名 |
| 2 | password | int | 10 | 否 | 否 | 用户密码 |

表4-2 guests表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **主键** | **允许空** | **注释** |
| 1 | id | int | 10 | 是 | 否 | 住客序号 |
| 2 | name | varchar | 10 | 否 | 否 | 住客名 |
| 3 | sex | varchar | 10 | 否 | 否 | 住客性别 |
| 4 | card | bigint | 20 | 否 | 否 | 住客身份证号 |
| 5 | phone | bigint | 20 | 否 | 否 | 住客手机号 |
| 6 | enterTime | varchar | 30 | 否 | 否 | 住客入住时间 |
| 7 | exitTime | varchar | 30 | 否 | 否 | 住客退房时间 |
| 8 | h\_type | varchar | 10 | 否 | 否 | 房间类型 |
| 9 | num | int | 10 | 否 | 否 | 房间号 |

1. home表，储存了房间序号（id）、房间号（num）、房间类型（h\_type）、房间价格（price）、房间状态（state）、房间图片（img）、房间信息（text）、的数据表。home表的结构如表4-3所示。

表4-3 home表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **主键** | **允许空** | **注释** |
| 1 | id | int | 10 | 是 | 否 | 房间序号 |
| 2 | num | int | 10 | 否 | 否 | 房间号 |
| 3 | h\_type | varchar | 10 | 否 | 否 | 房间类型 |
| 4 | price | varchar | 10 | 否 | 否 | 房间价格 |
| 5 | state | varchar | 10 | 否 | 否 | 房间状态 |
| 6 | img | varchar | 50 | 否 | 否 | 房间图片 |
| 7 | text | varchar | 255 | 否 | 否 | 房间信息 |

1. vip表，储存了会员序号（id）、会员名（name）、会员性别（sex）、会员手机号（phone）、会员身份证号（card）、会员注册时间（startTime）、会员结束时间（endTime）、会员类型（v\_type）的数据表。结构如表4-4所示。

表4-4 vip表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **列名** | **数据类型** | **长度** | **主键** | **允许空** | **注释** |
| 1 | id | int | 10 | 是 | 否 | 会员序号 |
| 2 | name | varchar | 10 | 否 | 否 | 会员名 |
| 3 | sex | varchar | 10 | 否 | 否 | 会员性别 |
| 4 | card | bigint | 20 | 否 | 否 | 会员身份证号 |
| 5 | phone | bigint | 20 | 否 | 否 | 会员手机号 |
| 6 | startTime | varchar | 30 | 否 | 否 | 会员注册时间 |
| 7 | endTime | varchar | 30 | 否 | 否 | 会员结束时间 |
| 8 | v\_type | varchar | 10 | 否 | 否 | 会员类型 |

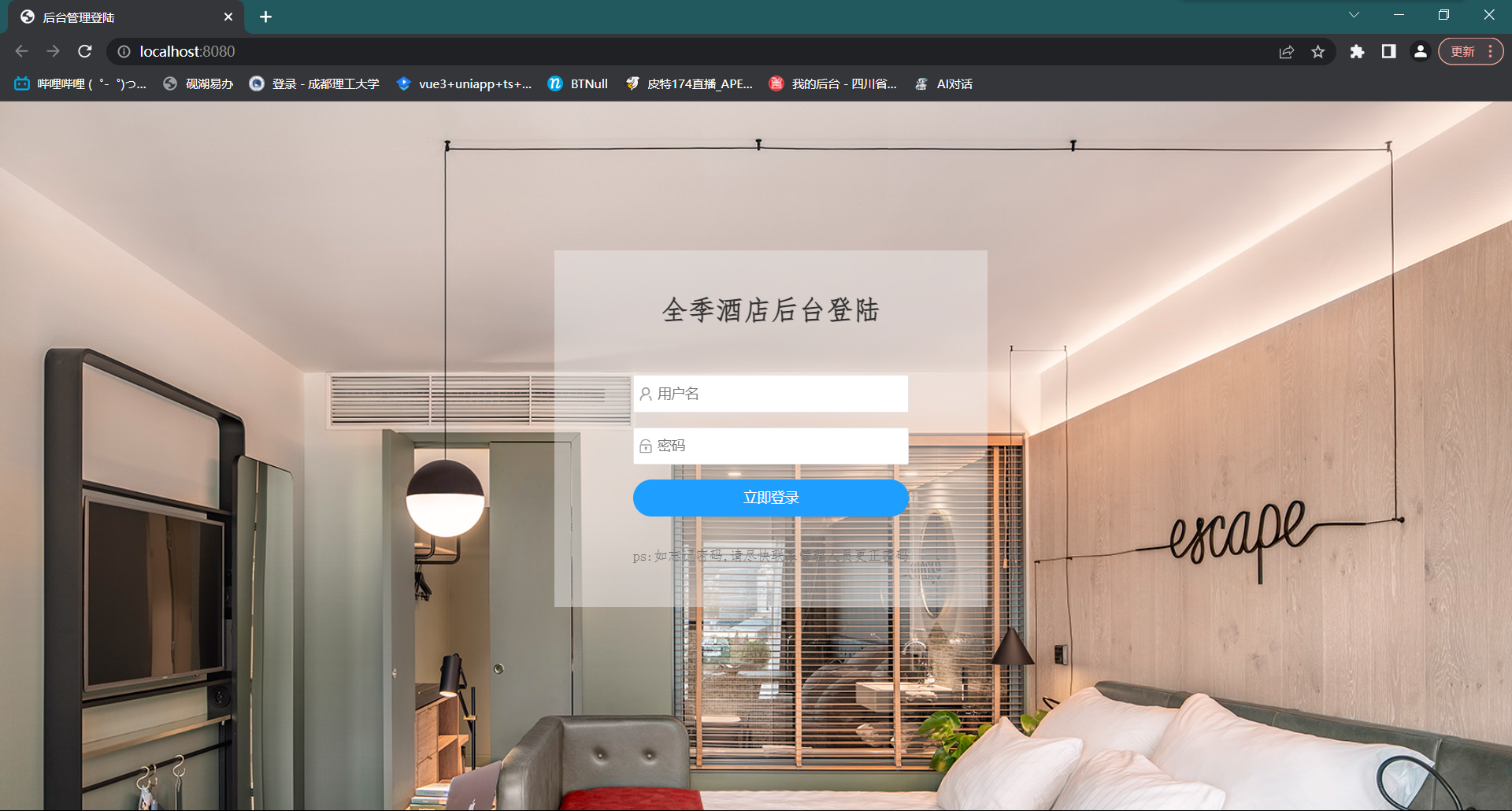
# 系统实现

本章主要以主要代码、实现逻辑两个方面来介绍各个模块的具体实现，同时会展示页面的整体布局和功能效果。

## 用户登录功能实现

用户在通过网址访问本平台时，必须输入账号密码，然后点击登录，在验证成功后才能跳转到首页从而进行之后的进一步操作。实现逻辑为首先在AdminMapper.xml里写好update函数以录入数据库，然后在AdminService里面写AdminServiceImpl接口调用mapper层的update函数为controller层调用做好准备，最后在AdminController里使用RequestMapping返回到前端和Mapper层实现前后端关联。在登录过程中，如果用户名、密码与数据库中管理员用户名、密码不一致，则跳转到报错界面，同时如果在数据库中找不到管理员用户名、密码同样抛出错误页面。登录与注册页面如图5-1所示。

图5-1 登录与注册页面



关键代码如下：

@RequestMapping("/login")

public String login(String username, int password){

Admin admin=adminService.findAdmin(username,password );

// 调用adminService的findAdmin方法，根据用户名和密码查找管理员

if(admin!=null){

if(admin.getPassword()==password){

// 比较管理员的密码和输入的密码是否一致

return "index";

// 如果一致，返回"index"页面

}else {

return "error";

// 如果不一致，返回"error"页面

}

}

return "error";

// 如果找不到管理员账号密码，返回"error"页面

}

## 系统管理功能实现

用户成功登录后，会进入网站的首页。从左侧菜单栏中选择系统管理界面，后即可进入网站信息界面。该界面由前端表单和按钮组成。为了清除浏览器缓存，我们使用了Cache-Control指定的no-cache缓存机制。修改密码的实现逻辑与登录类似。我们在已经编写好的impl函数和updateh函数的基础上，在AdminController中使用RequestMapping()将请求发送到Dao层的AdminMapper，以实现前后端的关联。系统管理页面如图5-2、5-3、5-4所示

图5-2 系统管理页面

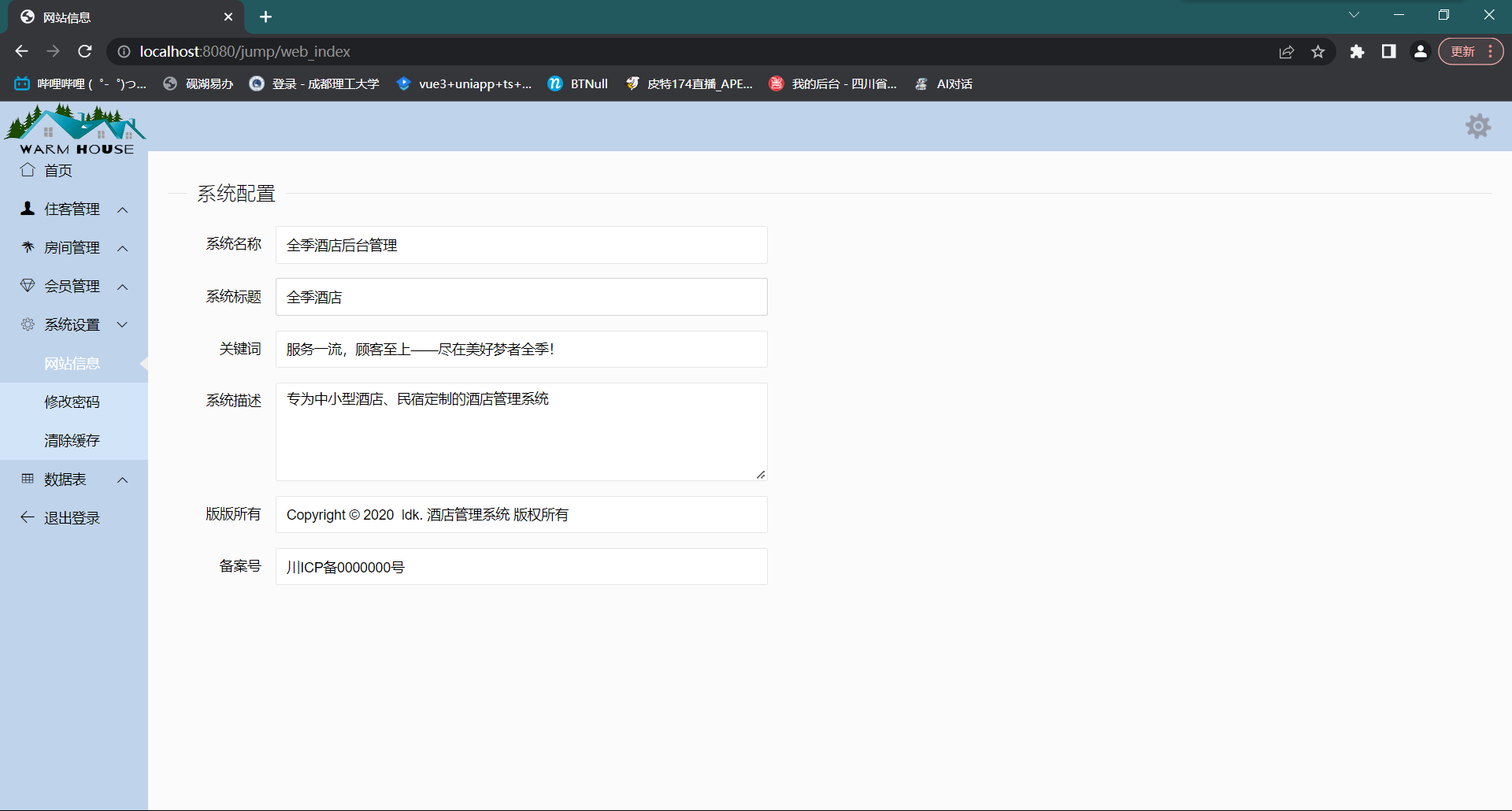


图5-3系统管理页面

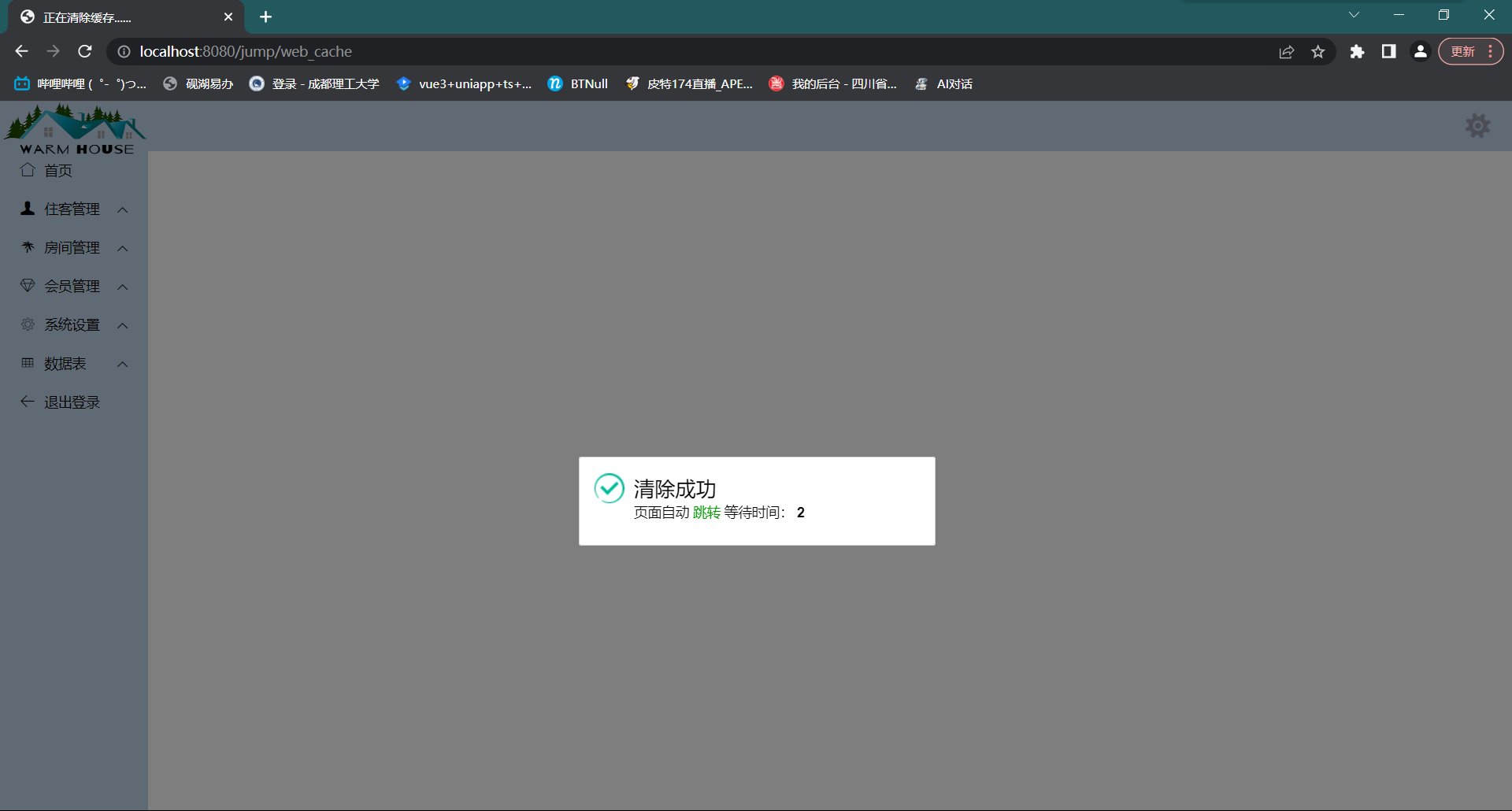
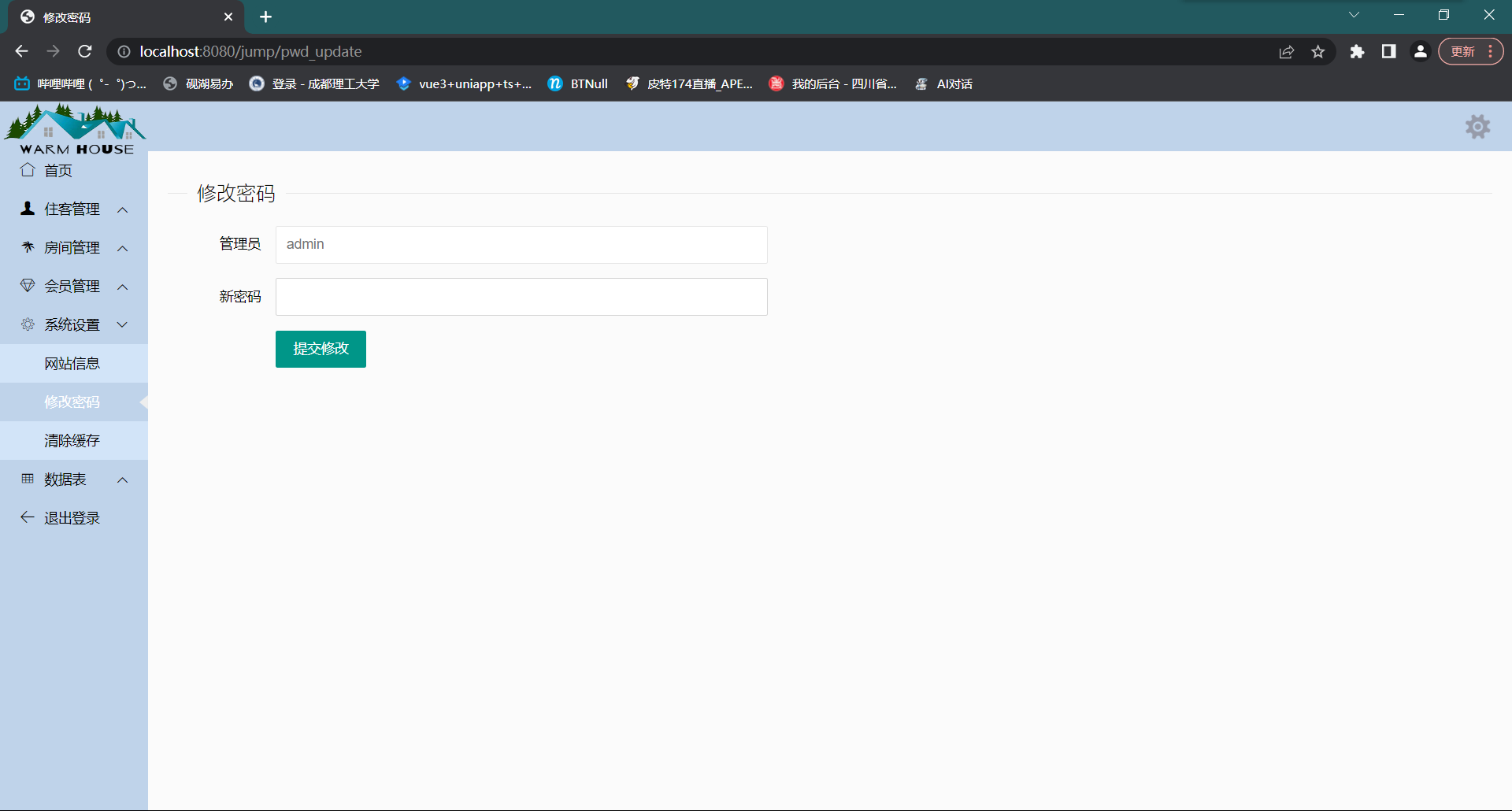


图5-4 系统管理页面

关键代码如下：

//修改密码

@RequestMapping("/updatePwd")

public String update(Admin admin){

adminService.updatePwd(admin);

// 调用adminService的updatePwd方法，更新管理员的密码

System.out.println(admin);

return "suc\_a";

// MyBatis的update语句，用于更新管理员的密码

<update id="updatePwd" parameterType="Admin">

update admin

set password=#{password}

where username='admin';

</update>

//系统信息

<div class="right">

<fieldset class="layui-elem-field layui-field-title">

<legend>系统配置</legend>

</fieldset>

<form class="layui-form " action="">

<div class="layui-form-item">

<label class="layui-form-label">系统名称</label>

<div class="layui-input-block">

<input type="text" name="name" value="全季酒店后台管理" placeholder=" " class="layui-input">

</div>

</div>

//清除缓存

<%

response.setHeader("Pragma","No-cache");

//设置响应头Pragma为No-cache，表示不缓存响应结果

response.addHeader( "Cache-Control", "must-revalidate" );//必须重新验证缓存

response.addHeader( "Cache-Control", "no-cache" );//不缓存响应结果

response.addHeader( "Cache-Control", "no-store" );//不缓存任何内容

//添加响应头Cache-Control，设置缓存控制策略

response.setDateHeader("Expires", 0);

//设置响应头Expires为0，表示缓存已经过期

response.flushBuffer();

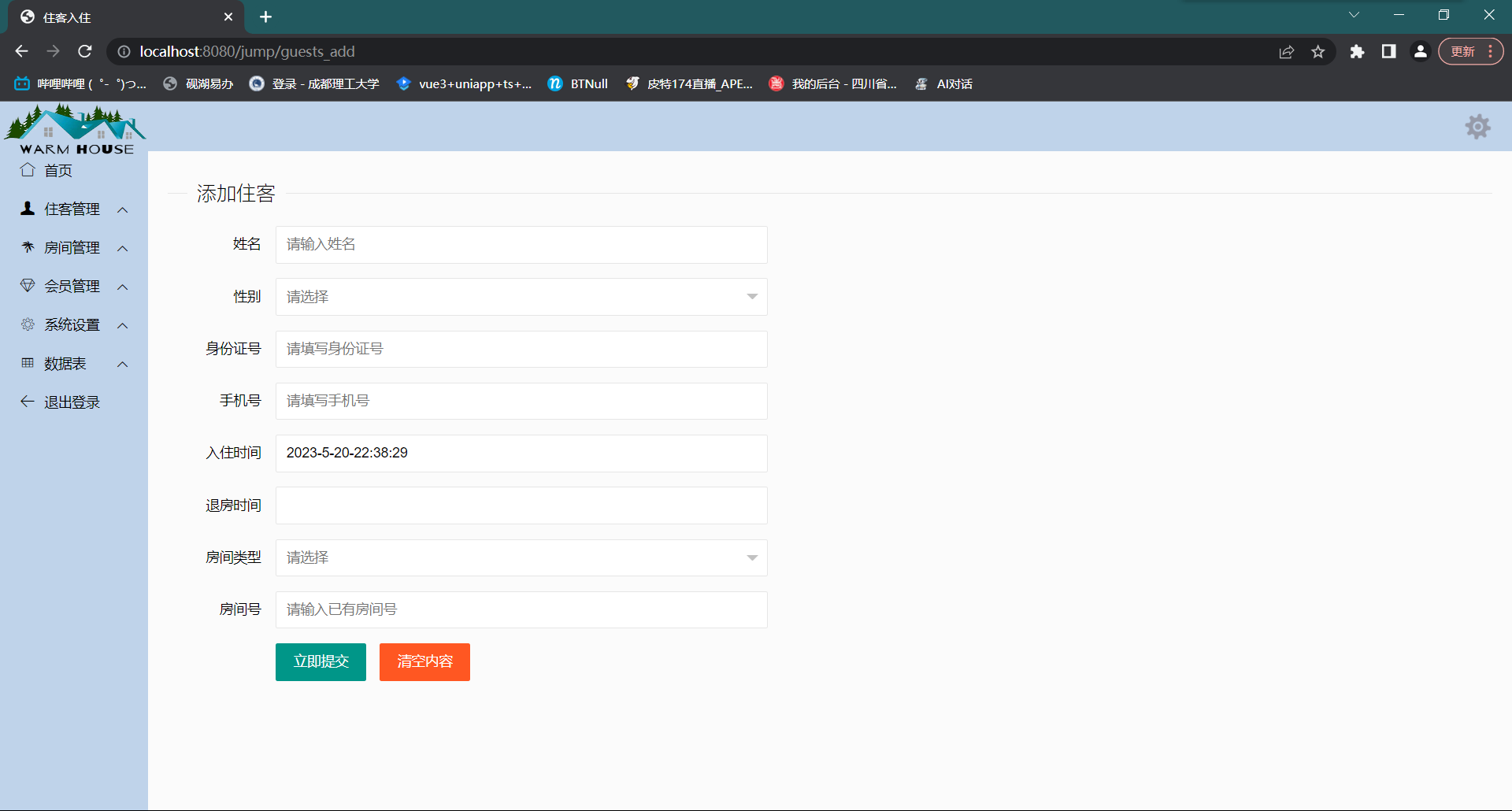
//刷新响应缓冲区，确保响应头信息能够及时发送到客户端

%>

## 住客管理功能实现

用户成功登录后，会进入网站的首页，从左侧菜单栏中选择住客管理界面即可进入。住客信息编辑的实现逻辑是在GuestsMapper.xml里写好update、select等函数以录入数据库，然后在GuestsService里面写GuestsServiceImpl接口调用mapper层的数据库写入函数为controller层调用做好准备，最后在GuestsController里使用RequestMapping来返回到Dao层中的GuestsMapper（add/delect等）来实现住客管理模块的前后端关联。住客管理页面如图5-5所示。

图5-5 住客管理页面



住客管理页面关键代码如下：

// 添加住客

// 通过@RequestMapping注解映射URL路径为"/add"的POST请求到add()方法

// 方法参数为Guests类型的guests对象

@RequestMapping("/add")

public ModelAndView add(Guests guests){

// 创建ModelAndView对象

ModelAndView mv = new ModelAndView();

// 调用guestsService的addGuests()方法添加住客

guestsService.addGuests(guests);

// 设置视图名称为"suc\_g"

mv.setViewName("suc\_g");

// 返回ModelAndView对象

return mv;

}

// 删除住客

// 通过@RequestMapping注解映射URL路径为"/delete"的GET请求到delete()方法

// 方法参数为int类型的id

@RequestMapping("/delete")

public String delete(int id){

// 调用guestsService的deleteGuestsById()方法删除指定id的住客

guestsService.deleteGuestsById(id);

// 重定向到"/guests/list"路径

return "redirect:/guests/list";

}

查询功能则是在通过@RequestMapping注解映射URL路径为"/find"的GET请求到find()方法后，调用guestsService的queryGuestsByPhone()方法查询指定手机号的住客信息，用户输入值与数据库中值不一样则抛错，一样则添加到list。查询页面如图5-6所示，住客信息修改如图5-7所示。

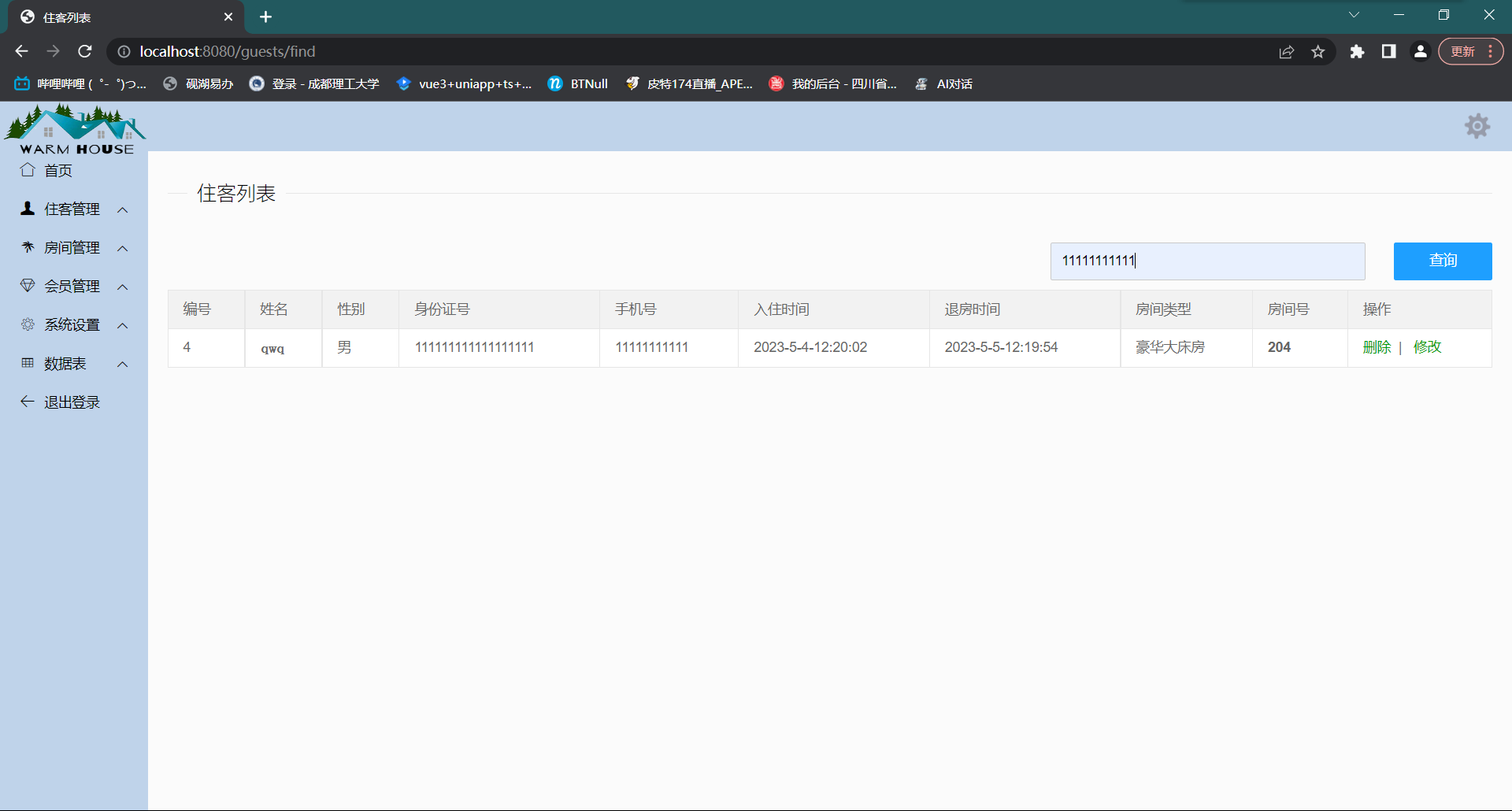


图5-6 住客查询页面

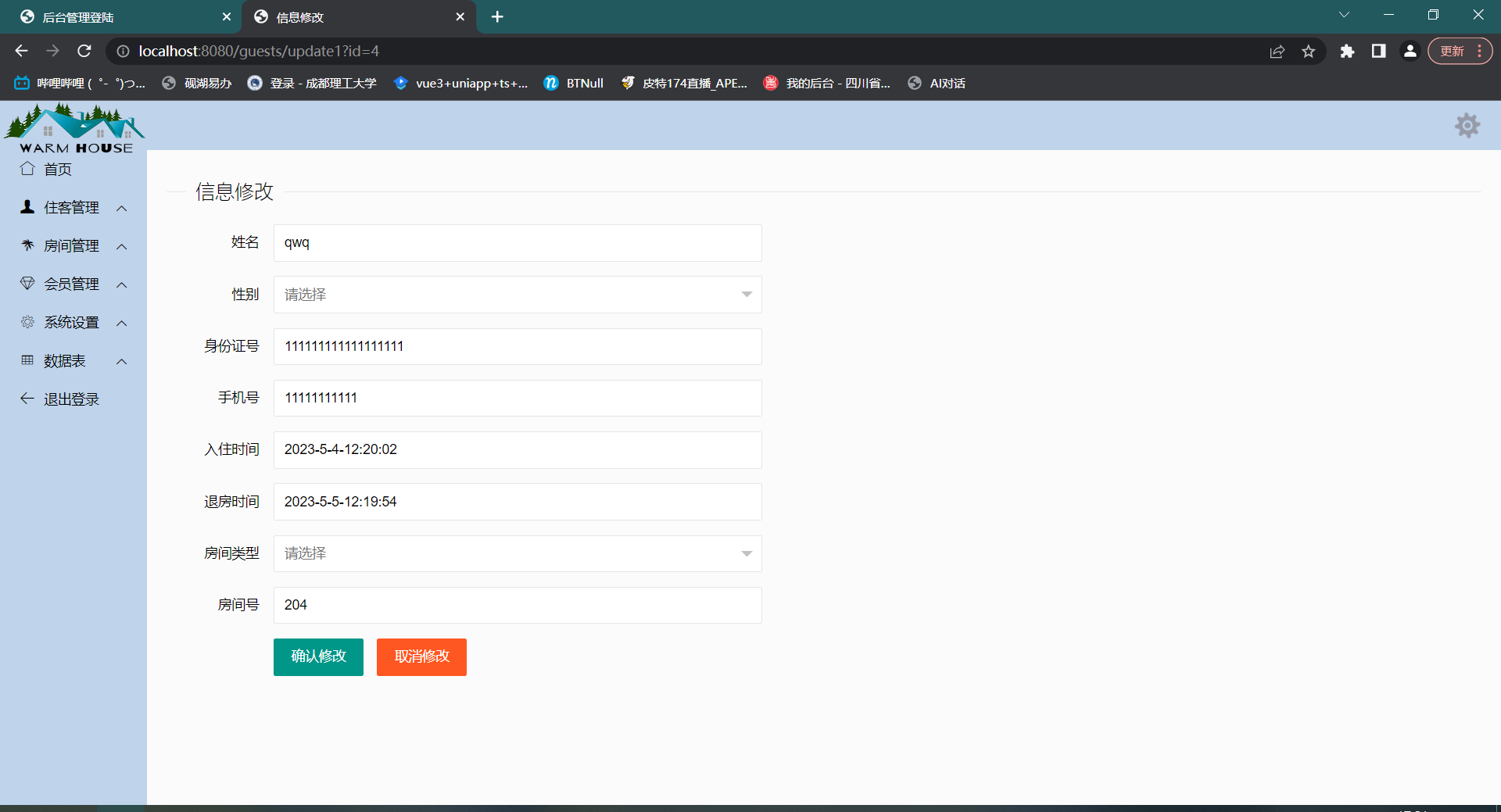


图5-7 住客信息修改页面

查询住客关键代码如下：

// 查询住客

// 通过@RequestMapping注解映射URL路径为"/find"的GET请求到find()方法

// 方法参数为String类型的findByPhone

@RequestMapping("/find")

public ModelAndView find(String findByPhone){

// 创建ModelAndView对象

ModelAndView mv = new ModelAndView();

// 调用guestsService的queryGuestsByPhone()方法查询指定手机号的住客信息

Guests guests = guestsService.queryGuestsByPhone(findByPhone);

// 创建List对象

List<Guests> guestsList=new ArrayList<Guests>();

// 将查询结果添加到List中

guestsList.add(guests);

// 如果查询结果为空

if (guests==null){

// 调用guestsService的queryAllGuests()方法查询所有住客信息

guestsList=guestsService.queryAllGuests();

// 将错误信息添加到ModelAndView对象中

mv.addObject("error","未查询出结果");

}

// 将List对象添加到ModelAndView对象中

mv.addObject("list",guestsList);

// 设置视图名称为"guests\_list"

mv.setViewName("guests\_list");

// 返回ModelAndView对象

return mv;

}

## 房间管理功能实现

用户成功登录后，会进入网站的首页，从左侧菜单栏中选择房间管理界面即可进入。房间信息编辑的实现逻辑是在HomeMapper.xml里写好update、select等函数以录入数据库，然后在HomeService里面写HomeServiceImpl接口调用mapper层的数据库写入函数为controller层调用做好准备，最后在HomeController里使用RequestMapping来返回到Dao层中的HomeMapper（add/delect等）来实现住客管理模块的前后端关联。查询功能同样是@RequestMapping注解映射URL路径为"/find"的GET请求到find()方法后，调用guestsService的queryHomeByNum()方法查询指定房间号的房间信息，用户输入值与数据库中值不一样则抛错，一样则添加到list。房间管理页面如图5-8所示，房间信息修改如图5-9所示。

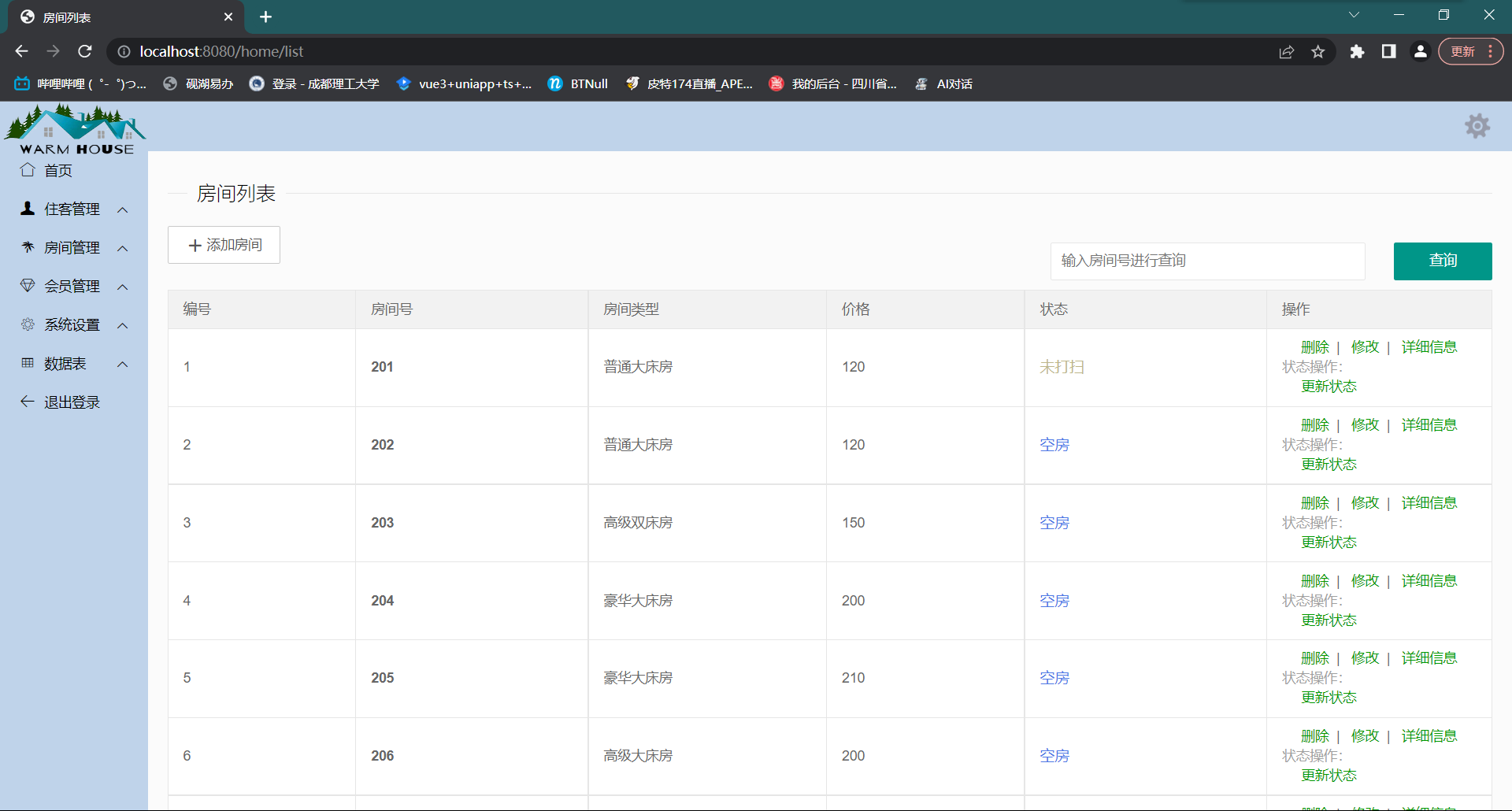


图5-8 房间管理页面

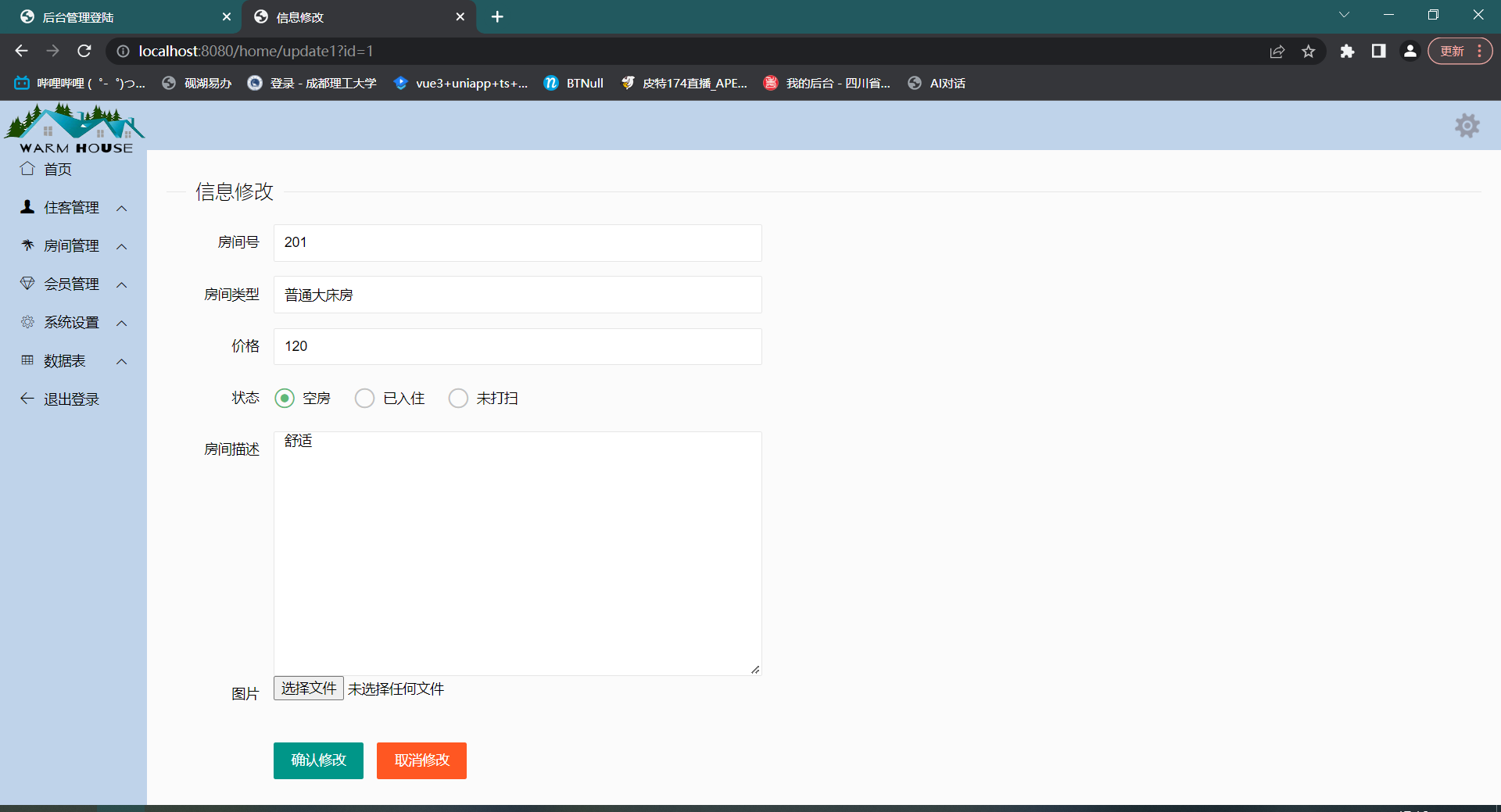


图5-9 房间信息页面

房间信息登记关键代码如下：

@RequestMapping("/add")

public String add(Home home, Model model) throws IOException{

String sqlPath = null;

//定义文件保存的本地路径

String localPath="E:\\Java 项目\\Hotel\_Manage\\src\\main\\webapp\\upload";

//定义 文件名

String filename=null;

if(!home.getFile().isEmpty()){

//生成uuid作为文件名称

String uuid = UUID.randomUUID().toString().replaceAll("-","");

//获得文件类型（可以判断如果不是图片，禁止上传）

String contentType=home.getFile().getContentType();

//获得文件后缀名

String suffixName=contentType.substring(contentType.indexOf("/")+1);

//得到 文件名

filename=uuid+"."+suffixName;

System.out.println(filename);

//文件保存路径

home.getFile().transferTo(new File(localPath+filename));

}

//把图片的相对路径保存至数据库

sqlPath = "/upload/"+filename;

System.out.println(sqlPath);

home.setImg(sqlPath);

homeService.addHome(home);

model.addAttribute("home",home);

return "home\_show";

}

## 会员管理功能实现

用户成功登录后，会进入网站的首页，从左侧菜单栏中选择会员管理界面即可进入。会员信息编辑的实现逻辑是在VipMapper.xml里写好update、select等函数以录入数据库，然后在VipService里面写VipServiceImpl接口调用mapper层的数据库写入函数为controller层调用做好准备，最后在VipController里使用VipMapping来返回到Dao层中的VipMapper（add/delect等）来实现住客管理模块的前后端关联。查询功能同样是@RequestMapping注解映射URL路径为"/find"的GET请求到find()方法后，调用guestsService的queryVipByPhone()方法查询指定手机号的会员信息，用户输入值与数据库中值不一样则抛错，一样则添加到list。会员管理页面如图5-10、图5-11、5-12所示。

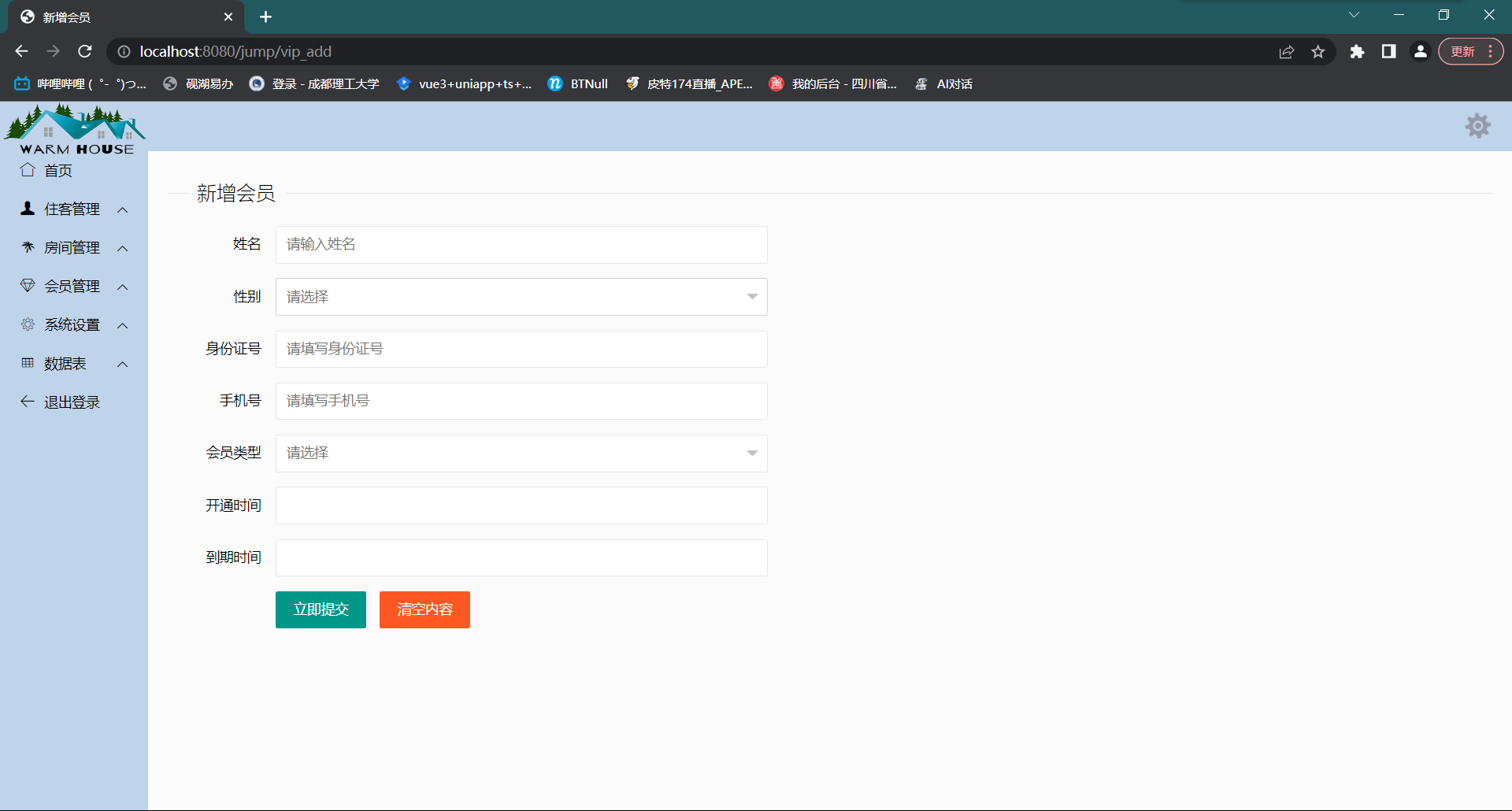


图5-10 会员管理页面

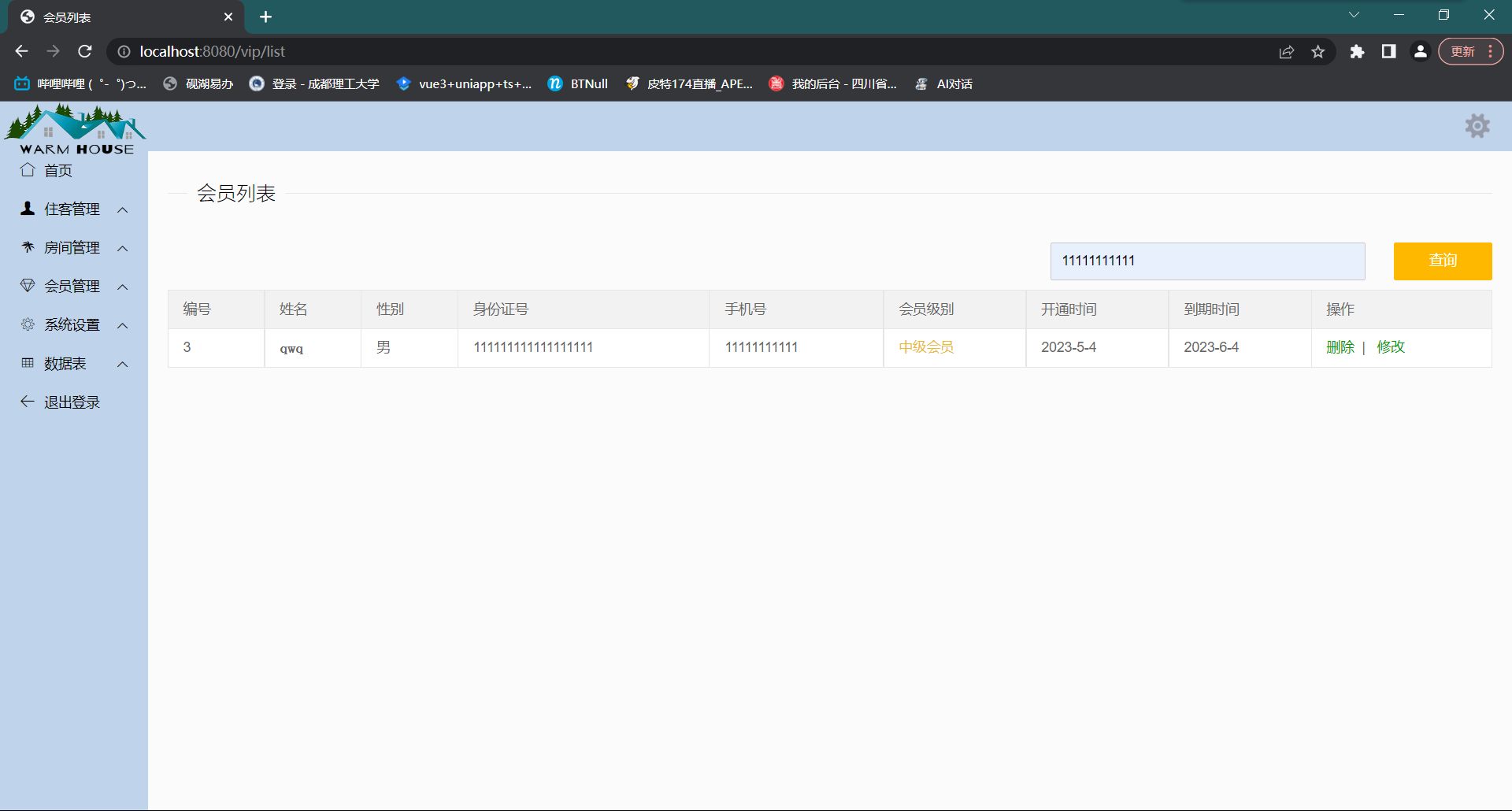


图5-11 会员管理页面

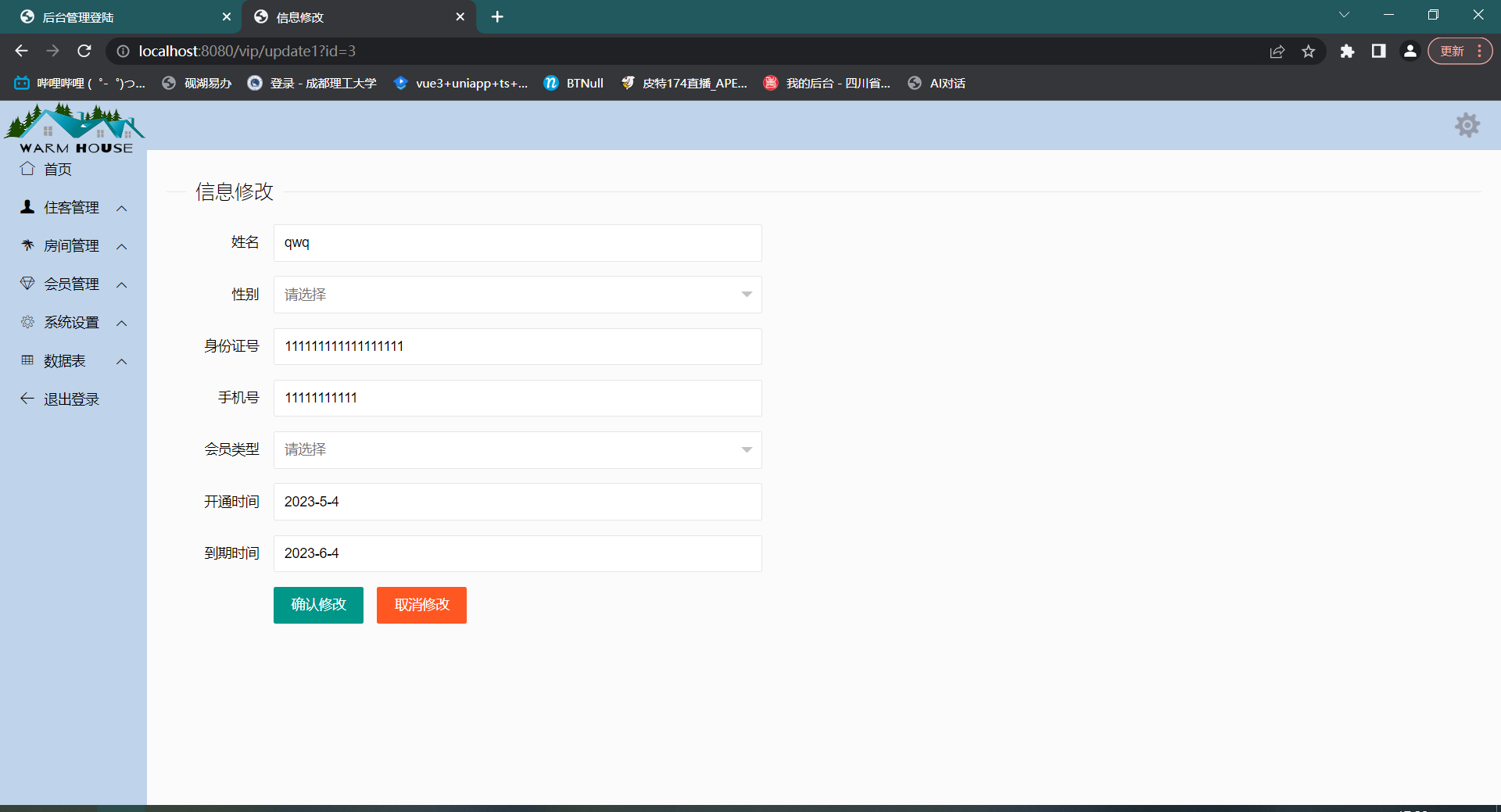


图5-12 会员信息修改页面

## 数据表打印功能实现

用户在成功登录后，首先进入首页，然后从左侧菜单栏选择进入数据表打印。数据表打印的实现原理为利用开源的Apache POI这款API中自带的HSSF方法完成数据库以Excel表单形式打印。实现逻辑为在ExcelController里用HSSFSheet方法手动编辑表头、表单，后用HSSFSheetRow方法完成数据写入后打印。数据表打印页面如图5-13、5-14所示。

图5-13 数据表打印页面

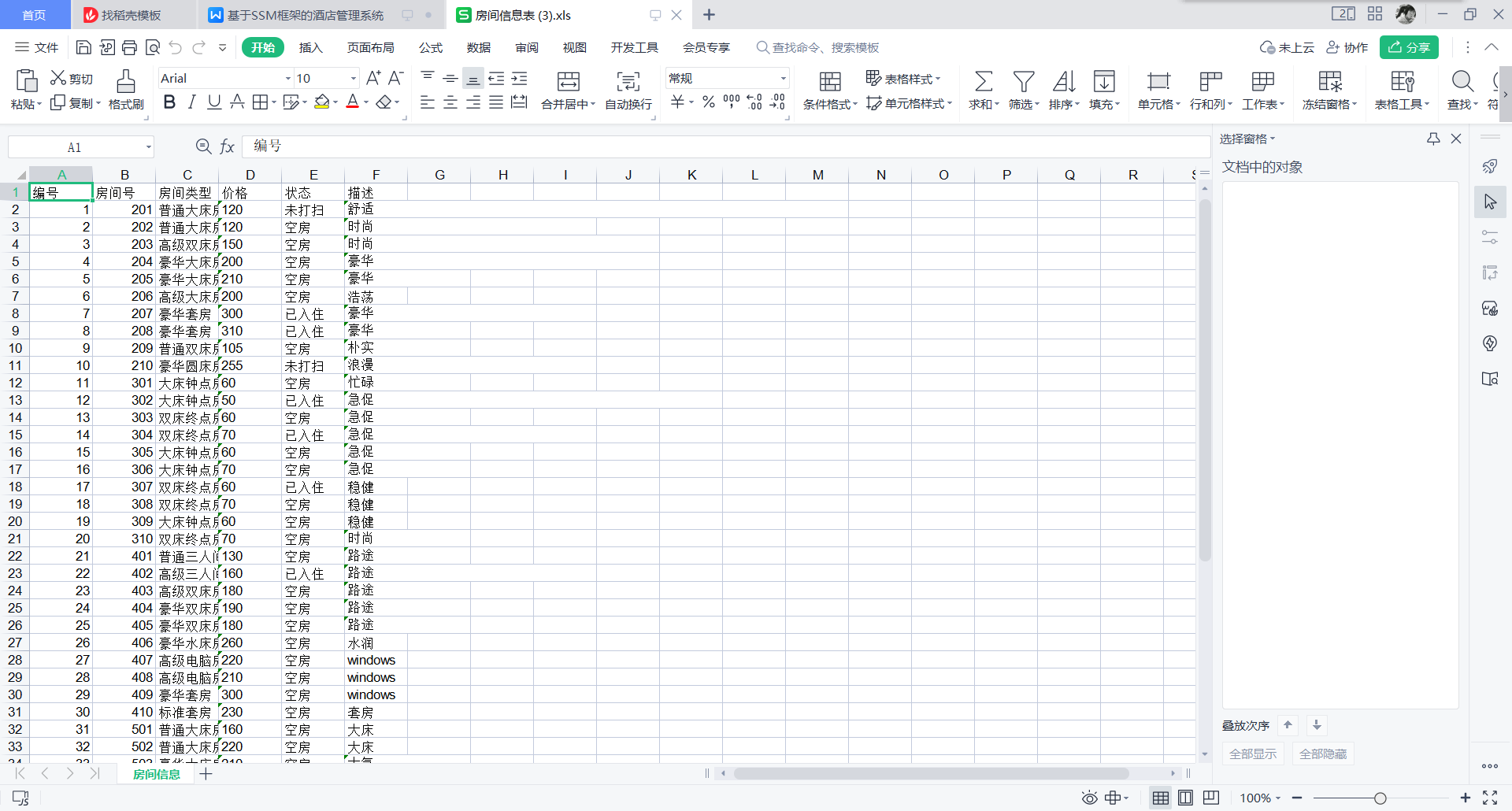
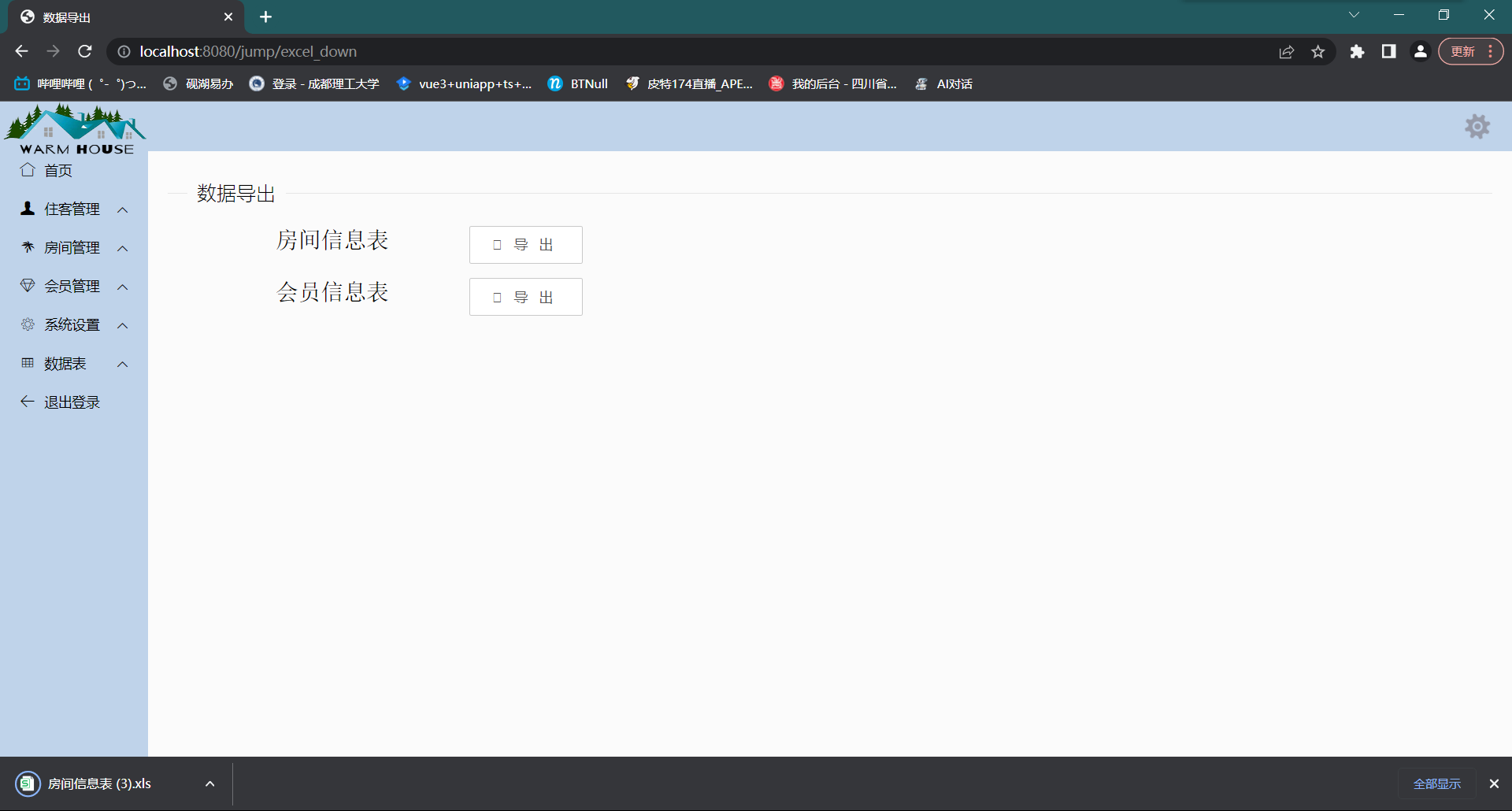


图5-14 数据表打印页面

# 系统测试

系统测试包括客房分配、入住登记算等核心功能的测试。通过系统测试，可以发现和解决潜在的功能缺陷，确保系统满足用户的需求。系统测试可以对酒店管理系统的性能进行评估，包括响应速度、系统负载等方面的测试。通过模拟实际使用场景和负载，可以发现性能瓶颈和优化需求，从而确保系统在高负载和高并发的情况下保持稳定。

## 功能性测试

### 功能测试

采用黑盒测试测试结果如表6-1所示。

表6-1 功能测试表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能组件** | **测试内容** | **测试结果** | **是否有异常** |
| 用户管理 | 用户信息 | 通过 | 无 |
| 登录管理 | 通过 | 无 |
| 系统管理 | 密码修改 | 通过 | 无 |
| 展示 | 通过 | 无 |
| 清除缓存 | 通过 | 无 |
| 住客管理 | 住客添加 | 通过 | 无 |
| 住客删除 | 通过 | 无 |
| 搜索住客 | 通过 | 无 |
| 反馈处理 | 通过 | 无 |
| 房间管理 | 房间添加 | 通过 | 无 |
| 房间删除 | 通过 | 无 |
| 搜索房间 | 通过 | 无 |
| 更新房间信息 | 通过 | 无 |
| 会员管理 | 会员添加 | 通过 | 无 |
| 会员删除 | 通过 | 无 |
| 搜索会员 | 通过 | 无 |
| 数据表打印 | 用户信息表打印 | 通过 | 无 |
| 房间信息表打印 | 通过 | 无 |

### 界面测试

界面测试主要看界面设计是否合理，交互功能是否正常，不通浏览器及不同尺寸屏幕是否显示正常，经过测试，系统在目前主流的浏览器（IE9、谷歌等）都能正常运行。对界面的所有控件和点击事件都进行了测试，测试结果均正常。

## 安全性测试

基本安全测试的内容及测试结果如表6-2所示。

表6-2 基本安全测试表

|  |  |
| --- | --- |
| **测试内容** | **测试结果** |
| 用户权限验证 | 正常 |
| Session有效期 | 时效一天，注销即失效 |
| 敏感数据加密 | 正常 |
| SQL注入 | 无SQL语句字符串拼接，正常 |

权限管理测试:

权限管理测试的主要目的是确保系统在权限控制方面的可靠性和安全性。本次系统测试进行了角色和权限分配测试为验证系统是否正确实施了角色和权限的分配，经测试表明权限分配正常。

文件和目录测试:

文件测试是对文件是否能正常保存进行测试。文件名处理应注意特殊字符，禁止获取敏感目录或文件信息，所有对目录的访问不得打印文件列表，不能越权获取不应获取的文件。经过测试，文件和目录的处理均正常。

# 结 论

对于该项目来说，主要的创新点在于基于SSM框架的酒店管理的高效性能：SSM框架结合了Spring、Spring MVC和MyBatis等优秀框架的特点，能够提供高性能的数据访问和处理能力。同时，本系统的界面设计对于用户操作来说，也是比较方便的。此外，本系统充分考虑了数据安全性、用户体验感、后期是否易于维护等问题。考虑比较全面，无论是从管理员对房间的分配、信息的管理、角色的分配、日志的管理等模块都做得比较详细。是一个较为完善的后台管理系统项目。

在本次系统设计里，我在着重在系统分析上面做了很多功夫。俗话说磨刀不误砍柴工，如果在可行性分析、需求分析这一个阶段出了纰漏，那么在后期想要进行更正甚至于项目重构，那么造成的损失将是无比巨大的。一开始从查阅软件工程教材、查阅论文到查看需求文档，我所了解的需求分析轮廓逐渐清晰。我将功能模块化，然后再从功能模块入手，画出每个模块的业务流程图，逐步地拨开迷雾，走向前方。

当然，在本系统开发过程中，同样遇到了许多问题，由于软件开发的时间问题尚未得到妥善地解决。例如，系统可能存在访问控制不严密的问题，导致未经授权的用户可以访问敏感信息或执行特权操作。建议对系统进行全面的访问控制验证，类似于敏感操作的二次验证、细粒度的权限控制等。这些问题，我也会在以后继续进行完善和修正。而且SSM框架及其相关组件经常发布安全更新和修复漏洞。所以我也会定期更新框架框架和依赖库的版本，并及时应用安全补丁来确保系统安全。

# 参考文献

曹华山.2021. SSM框架在Web应用开发中的设计与实现[J].无线互联科技

胡孔法.2020. 数据库原理及应用[M].机械工业出版社

方 忠.2000. JSP 技术及其在动态网页开发中的应用[J].微型机与应用

郭克华.2019. Java Web课程设计（第3版）[M].清华大学出版社

耿祥义,张跃平.2014. Java程序设计实用教程（第2版）[M].人民邮电出版社

蒋治学.2020. JSP技术及其在动态网页开发中的应用分析[J].浙江水利水电学院学报

邱丹萍.2020. Web开发中SSM框架的分析[J].电脑知识与技术

秦不凡.2020. 大数据时代下软件工程技术的应用[J].农家参谋

吴官学.2018. 基于SpringMVC 酒店信息管理系统[D].吉林大学

赵晓彤.2014. 酒店管理系统设计[J].电子制作

Alomari, M., Altahat, S. 2019. Design and implementation of a hotel management system using ASP.NET MVC. International Journal of Advanced Trends in Computer Science and Engineering, 8(1.3).

Alarifi, A.,Al-Surmi, M. 2016. A comprehensive hotel management system based on Internet of Things (IoT). 2016 IEEE International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, Austria.

Chou, H. H.,Li, Y. 2017. Design and implementation of a hotel management system based on cloud computing. International Journal of Information Engineering and Electronic Business, 9(3).

Chen, C. H., Liao, Y. C. 2018. Design and implementation of a hotel management system based on RFID technology. International Journal of Advanced Computer Science and Applications, 9(6).

Kim, H., Lee, D. 2017. Design and implementation of an intelligent hotel management system based on IoT and cloud computing. Journal of Information Processing Systems.

目 录

第1章 前 言 1

1.1 论文研究的目的和意义 1

1.2 国内外研究综述 1

1.3 论文研究的内容和取得的成果 2

1.4 文章组织结构 3

第2章 网站开发技术简介 4

2.1 JSP动态网站开发技术简介 4

2.2 SSM框架简介 5

2.2.1 SpringMVC框架 6

2.2.2 Spring框架 7

2.2.3 Mybatis框架 7

2.2.4 SSM框架整合 7

2.3 前端LayUI框架简介 8

2.4 MySQL数据库技术简介 9

第3章 系统分析 10

3.1 可行性分析 10

3.1.1 技术可行性 10

3.1.2 经济可行性 10

3.1.3 市场可行性 11

3.1.4 法律可行性 11

3.2 需求分析 11

3.2.1 用户登陆模块 12

3.2.2 系统管理模块 12

3.2.3 住客管理模块 13

3.2.4 房间管理模块 13

3.2.5 会员管理模块 13

3.2.6 数据表打印模块 13

第4章 系统设计 14

4.1 概要设计 14

4.1.1 用户登录设计 14

4.1.2 系统管理设计 15

4.1.3 住客管理设计 15

4.1.4 房间管理设计 16

4.1.5 会员管理设计 16

4.1.6 数据表打印设计 17

4.2 数据库设计 18

第5章 系统实现 21

5.1 用户登录功能实现 21

5.2 系统管理功能实现 22

5.3 住客管理功能实现 24

5.4 房间管理功能实现 27

5.5 会员管理功能实现 29

5.6 数据表打印功能实现 31

第6章 系统测试 32

6.1 功能性测试 32

6.1.1 功能测试 32

6.1.2 界面测试 33

6.2 安全性测试 33

结 论 34

参考文献 35