

毕业论文
基于 SSM 框架的房源租赁系统

专业：

学号：

姓名：

武汉纺织大学
数学与计算机学院
2019 年 5 月

摘 要

一直以来，住房问题都是人们在社会生活中不可忽视的问题，特别是对于刚进入社会的诸如大学生等的年轻群体而言，以自己的经济实力买房，甚至是仅仅支付房屋的首付都是不太现实的。那么在年轻群体拥有足够的经济实力去买房之前，他们的住房问题只能用其他方式来解决，而其中最主要的方式就是利用房屋租赁的网站或是 App 在网上查找适合自己的房源。

房屋租赁系统是一个 O2O 商业模式的中介网站系统，它通过在网上发布房源的信息让有租房需求的人能够在众多的房源中筛选出自己心仪的房源，然后提供房东与租客之间的沟通渠道，让用户自己与房东线下见面，陪同看房，促成最后的租房成交。

本文基于自身租房的经历为例，结合对租客租房过程的需求分析，以 Java 为后端开发语言，使用 SSM 框架，以 MySQL 为关系型数据库平台，以 Bootstrap 为前端辅助框架，设计了房屋租赁系统的 Web 端和管理后台两个平台。该系统针对租客和房东拥有查看房源，筛选房源，收藏房源和上下架房源等功能。

本文分析了租客在租房过程中可能的需求，重点对系统的前台房源显示功能、后台管理功能、所需的数据库和租客的租赁流程进行了分析，并给出了系统主要功能的实现方法，展示了最终的实现效果和测试结果。

目前该房屋租赁系统的功能已基本完成，租客租房的逻辑流程初步得到实现。

关键词：O2O，房屋租赁，Java，SSM框架，MySQL

ABSTRACT

All the time, people's housing problem has always been a problem that people can not ignore in social life, especially for young people who have just entered society, such as college students, buy a house with your own financial strength, or just paying down on the house is unrealistic. Before young people have enough financial power to buy a house, the housing problem can be solved in other ways, the most important of which is to use a rental website or app to find a suitable home on the Internet. Housing rental system is a O2O business model intermediary website system, it through the online release of housing source information so that people with rental needs in a large number of houses to screen out their favorite source of housing, and then provide a communication channel between landlords and tenants, so that users themselves meet with the landlord line, to see the house, resulting in the final rental transaction.

This paper, based on its own rental experience as an example, combined with the demand analysis of the tenant rental process, takes java as the back-end development language, uses the SSM framework, takes the MySQL as the relational database platform, takes the bootstrap as the front-end auxiliary frame, designs the housing rental system web end and the management backstage two platforms. The system is designed for tenants and landlords to view the source of housing, screening the source of housing, collection of housing and upper and lower shelf housing and other functions.

This paper analyzes the possible needs of tenants in the process of renting, focusing on the system's front house source display function, background management function, the required database and tenant's rental process, and gives the implementation method of the main functions of the system, showing the implementation effect and test results.

At present, the function of the rental system has been basically completed, and the logical process of tenant rental has been initially realized.

Keywords: O2O, Housing rental, Java, SSM framework, MySQL

目录

1 引言	1
1. 1 房屋租赁系统开发背景	3
1. 2 房屋租赁行业发展状况	3
1. 3 O2O 商业模式.....	4
1. 4 系统开发的目的与意义	5
2 系统技术概括	6
2. 1 系统前端技术概括	6
2. 1. 1 Bootstrap 框架	6
2. 1. 2 Jquery 及 Ajax	7
2. 2 系统后端技术概括	8
2. 2. 1 MySQL 数据库	8
2. 2. 2 Tomcat 服务器	9
2. 2. 3 MVC 设计模式	10
2. 2. 4 Spring 框架.....	11
2. 2. 5 SpringMVC 框架.....	11
2. 2. 6 MyBatis 框架.....	12
3 系统分析与设计	13
3. 1 房屋租赁系统概述	13
3. 2 房屋租赁系统需求分析	13
3. 2. 1 租客需求分析.....	13

3.2.2 房东需求分析.....	14
3.2.3 管理员需求分析.....	15
3.3 系统设计	15
3.3.1 数据库设计.....	15
3.3.2 租客功能设计.....	17
3.3.3 房东功能设计.....	19
3.3.4 管理员功能设计.....	21
4 系统实现	22
4.1 数据库实现	22
4.2 租客功能实现	25
4.3 房东功能实现	32
4.4 管理员功能实现	35
5 总结与展望	38
5.1 总结.....	错误!未定义书签。
5.2 展望.....	错误!未定义书签。
参考文献	38
附录 A:英文原文	37
附录 B:中文译文	43
致谢	47

1 引言

1.1 房屋租赁系统开发背景

现今我国较为发达的地区房价高昂，绝大多数年轻人不具备买房能力，即使通过分期付款的方式，年轻群体中能拥有自己居所的人群占比也并不算高，所以不少年轻人会在拥有买房的经济实力之前选择租赁房屋解决自己的住处问题。而对于外来务工的流动人员，通过租赁房屋来解决居住问题则更为常见。可以说房屋租赁仍是许多人解决住宿问题的主要选择，需求量十分巨大，越为发达的地区流动人口越多，房屋租赁的市场也随之更甚。

在这种情况下，人们急需一个安全可靠的平台能够根据所在地不同，筛选出与之相对应的房源，并能详细让租客了解这些房源的信息例如房源房间的大小，房源租赁所需的价格，房屋的租赁方式等等。通过用户的搜索条件筛选出最适合该租客居住的房间，最后与房东线下交流，达成交易。而我设计的房屋租赁系统就能够初步解决这些问题。

论文所做的工作具有一定的社会意义，能够解决实际问题。

1.2 房屋租赁行业发展状况

房屋租赁是当今房地产市场中一种重要的交易方式，是一种出租人通过以口头或订阅局面合同的形式将房屋交付承租人占有、使用与收益，由承租人向出租人支付租金的行为^[1]。而按照房屋的所有权的性质，房屋租赁分为公有房屋的租赁和私有房屋的租赁，其中公有房屋的所有权人是国家，私有房屋的所有权人是指持有完全的房屋所有权证的个人。一般我们所接触到的更多为私有房屋的租赁，包括所设计的房屋租赁系统中的房源也以私有房屋为主。

近年来，房屋租赁行业发展迅速。以 2015 年的数据为例：2015 年我国流动人口 2.3 亿人，对应租房比例为 67%，租房市场总规模约为 2 万亿，14 个热点城市 2015 年时市场规模 8344 亿元，年均复合增长率达到可观的 10.5%。由此可见，房屋租赁的市场十分巨大。而且我国现在处于高速发展的状态，各地政策都鼓励更多年轻人能来热点城市发展。流动人口的增多，相应的租房的需求也继续加大。

不仅现在房屋租赁市场十分可观，其发展的趋势也十分迅猛。一方面更多商业巨头入驻，共同打造房屋租赁市场，必将使房屋租赁市场更加规范，发展势头良好。另一方面，大数据、人工智能、信用评价系统等先进技术层出不穷，房屋租赁市场会更加透明，租住双方的矛盾也在减少。

总体来说，由于热门城市快速发展需要巨大的人力资源，导致流动人口增加，而热门城市居高不下的房价使得年轻人难以快速买房，最终导致房屋租赁行业快速发展，极具潜力。非热点城市流动人口较少，租赁市场相应较小，但仍旧十分可观。另外我国租房比例较低、自住率高一定程度上造成了租房-购房市场之间的失衡，热点城市的租房市场仍在继续发掘。

1. 3 O2O 商业模式

O2O 是 Online To Offline（线上到线下）的缩写，是指将线下的商务机会与互联网结合，让互联网成为线下交易的前台。O2O 的概念非常广泛，只要产业链中既可涉及到线上，又可涉及到线下，就可通称为 O2O^[2]；也有观点认为，O2O 是 B2C（Business To Customers）的一种特殊形式。

自 O2O 商业模式出现发展以来，其一共经历了三个阶段。

早期第一阶段，O2O 线上线下初步对接，主要是利用线上推广的便捷性等把相关的用户集中起来，然后把线上的流量倒到线下，主要领域集中在以美团为代表的线上团购和促销等领域。在这个过程中，存在着主要是单向性、粘性较低等特点。平台和用户的互动较少，基本上以交易的完成为终点。用户更多是受价格等因素驱动，购买和消费频率等也相对较低。

发展到第二阶段后，O2O 基本上已经具备了大家所理解的要素。这个阶段最主要的特色就是升级为了服务性电商模式：包括商品、下单、支付等流程，把之前简单的电商模块，转移到更加高频和生活化场景中来。在新模式的推动和资本的催化下，出现了 O2O 的狂欢热潮，上门按摩、上门送餐、上门生鲜、上门化妆、滴滴打车等各种 O2O 模式开始层出不穷。而在这个阶段，移动终端、微信支付、数据算法等环节的成熟，加上资本的催化，用户数量激增，使

用频率和依赖性开始上升，O2O 开始和用户的日常生活开始融合，成为生活中密不可分的一部分。

到了第三阶段，开始了明显的分化，一个是真正的垂直细分领域的一些公司开始凸现出来。比如专注于快递物流的速递易，专注于高端餐厅排队的美味不用等，专注于白领快速取餐的速位。另外一个就是垂直细分领域的平台化模式发展。由原来的细分领域的解决某个痛点的模式开始横向扩张，覆盖到整个行业。

O2O 其有四种线上线下组合模式。

第一类为“线上社区+线下消费/社区”。这是一种必须到线下进行消费的 O2O 模式，线上主要是在线交流互动，并开展一定的优惠或促销活动。这种模式适应于必须亲自到现场消费的宾馆、餐饮及其他领域。

第二类为“线上消费/社区+线下社区”。这是一种只在线上销售的 O2O 模式，线上还有在线交流或开展促销作用，线下主要是面对面的交流互动或现场展示。这种模式适应于无线下门店仅有线上网店的纯线上电商领域。

第三类为“线上消费/社区+线下消费/社区”。这是一种线上与线下同时进行销售的 O2O 模式，线上、线下都有交流互动的需求，线上与线下还可以分别开展优惠或促销活动。这种模式适应于线上有网店且线下有门店的领域。

第四类为“线上社区+线下社区”。这是一种无销售业务的 O2O 模式，适用于同时需要线下及线下交流互动的社交平台^[3]。

1.4 系统开发的目与意义

出于个人实际的生活经历，在不使用租房网站及 App 的情况下，人们很难在有租房需求时快速找到可租赁的房源，即使通过其他途径找到的房源缺少相关手续的情况下存在不小的安全风险，而因为非正规渠道的租房而出现的意外事件比比皆是。相比之下通过正规租房网站达成租房交易的渠道则更为可靠。

设计一个房屋租赁系统，了解怎样以这种方式解决人们的租房问题同时锻炼自身的能力，这是我系统开发的目的。而该系统开发投入社会，能方便人们筛选出适宜的房源，快速安身。特别是对于年轻群体，能有效地缓解他们的租房压力，

具有一定的社会意义。

2 系统技术概括

2.1 系统前端技术概括

2.1.1 Bootstrap 框架

Bootstrap 是基于 HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使用该框架使得 Web 开发更加快捷。它提供了优雅的 HTML 和由动态 CSS 语言 Less 写成的 CSS 规范^[3]。

Bootstrap 框架拥有以下特色系统：

（1）栅格系统

栅格系统将网页内容以固定大小单元格分隔开来，使用者通过设置容纳标签的单元格的数量，就可以调整该标签的大小和大概位置，达到页面整体布局的效果。而且 Bootstrap 中的栅格系统还根据浏览器窗口大小设定了不同的单元格单位，同时使用多个单元格可以使网页随着窗口的大小动态改变，达到流式布局的目的。

（2）预设定 HTML 组件

Bootstrap 框架拥有预先设定完成的组件。在不使用框架的情况下，HTML 初始设定的按钮、表格等组件十分单调，没有较好的感官效果，需要开发者自己为其添加 CSS 样式表，十分麻烦。而 Bootstrap 内设的按钮、表格已经事先改变了 CSS 属性，看起来更加美观，还设定有不同的类别供用户选择，如按钮的颜色、表格的边框等等。开发者只需改变标签的 class 属性就可以进行设置，十分方便。

（3）特殊组件

Bootstrap 为用户设定了常用的具有特殊作用的组件，如下拉菜单，导航栏，模态框及轮播图等。这些组件效果原本要通过 HTML，CSS，JavaScript 实现起来非常麻烦，而使用 Bootstrap 框架后仅需引用，改变部分属性就可以按 Bootstrap 预设定的 JQuery 插件实现期望的效果，极大地减少了开发者的开发量，加快了开发速度。

通过 Bootstrap 框架和 CSS，JavaScript 的组合使用，可以在确保美观的同时又节约开发的时间，对于一些轻量级的前端开发来说是十分推荐的选择。

2.1.2 JQuery 及 Ajax

Jquery 是一个快速、简洁的 JavaScript 框架，其倡导写更少的代码，做更多的事情。它封装 JavaScript 常用的功能代码，提供一种简便的 JavaScript 设计模式，优化 HTML 文档操作、事件处理、动画设计和 Ajax 交互^[4]。

Jquery 的核心特性可以总结为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的 CSS 选择器，并且可对 CSS 选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。

一般而言，选择使用 JQuery 是因为它的以下特点：

（1）更方便的选择器

Jquery 的选择机制构建于 CSS 的选择器，它提供了快速查询 DOM 文档中元素的能力，而且大大强化了 JavaScript 中获取页面元素的方式。相比于原本的选择器，Jquery 使用更少的语句即能达到相同的效果，同时选择的精度也要更高。

（2）提供页面动态效果

Jquery 中内置了一系列的动画效果，比如淡入淡出、元素移除等动态特效。这些特效增强了用户在浏览网页时的体验感，许多的网站都有使用。

（3）创建 Ajax 无刷新网页

Ajax 是异步的 JavaScript 和 XML 的简称，可以协助开发出非常灵敏无刷新的网页

（4）对 JavaScript 语言的增强

Jquery 提供了对基本 JavaScript 结构的增强，比如元素迭代和数组处理等操作。

（5）增强的事件处理

Jquery 提供了各种页面事件，它可以避免程序员在 HTML 中添加太多事件处理代码，同时它的事件处理器消除了各种浏览器兼容性问题。

Ajax 是 JQuery 框架的一部分功能简称，它在网页制作中应用广泛，其具体功能在上面叙述 JQuery 的特点中也有提到，即可以使我们发送请求时进行异步操作，在不刷新页面的情况下对后台的处理产生响应。

不借助 Ajax 的话，我们在页面逻辑操作时，必须要进行从当前页面到控制视图层再返回当前页面的多次跳转，很影响用户的操作体验。而借助 Ajax，后端在处理请求时用户依然可以在前端页面继续进行浏览，直到响应返回前端用户再进行操作，整个过程更加连贯，用户的操作体验自然也会更好。

2. 2 系统后端技术概括

2. 2. 1 MySQL 数据库

数据库分为关系型数据库和非关系型数据库。

关系型数据库指的是采用了关系模型来组织数据的数据库，拥有关系，元组，属性，域，关键字，关系模式等概念。关系型数据库比较直观，更容易被理解，使用起来较为方便，易于维护。但关系型数据库性能欠佳，在处理大量关系复杂的数据时，效率很低。我们常用的关系型数据库有 Oracle, Microsoft SQL Server, MySQL, PostgreSQL, DB2, Microsoft Access, SQLite 等^[5]。

非关系数据库指的是指与关系型数据库相对的非关系型的，分布式的数据库。它以键值对存储，且结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，不局限于固定的结构。非关系型的数据库结构简单，处理数据时可以节省一点的时间和开销。但它仅限处理一些简单的没有复杂关联关系的数据库，不适合持久存储海量数据。由于非关系型数据库都是针对某些特定的应用需求出现的，因此，不同非关系数据库的特长也有所不同，主要分为以下几类：面向高性能并发读写的 key-value 数据库，主流代表为 Redis, Amazon DynamoDB；面向海量数据访问的面向文档数据库，主流代表为 MongoDB；面向搜索数据内容的搜索引擎，主流代表为 Splunk, Solr；面向可扩展性的分布式数据库，主流代表为 Cassandra, HBase^[6]。

MySQL 是典型的关系型数据库，它由瑞典 MySQL AB 公司开发，在被收购之后属于 Oracle 旗下产品。与其他关系型数据库相同，MySQL 将数据保存在不同的表中，确保效率并提高了灵活性。MySQL 使用 C 和 C++ 编写，并使用了多种编译器进行测试，保证了源代码的可移植性，使得它能够支持大多数的操作系统。

MySQL 所使用的操作语言 是用于访问数据库的最常用标准化语言 SQL 语言。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。虽然 MySQL 与其他的大型数据库比有所不足，但对于个人开发者，MySQL 数据库包含了个人开发所需的全部功能，并且它是免费开放的，降低了学习成本。也正是出于上述原因，我在房屋租赁系统中选择了 MySQL 数据库为项目的数据库。

2.2.2 Tomcat 服务器

搭建 Web 项目，服务器是必不可少的。而 Tomcat 技术先进、性能稳定，而且免费，是个人 JavaWeb 开发的首选。

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，由 Apache、Sun 和其他一些公司及个人共同开发而成，属于轻量级应用服务器。它更多在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。在 Tomcat 配置完成后，开发者就可以在服务器上运行 JSP 页面和 Servlet。它具有处理 HTML 页面请求的功能，同时还是一个 Servlet 和 JSP 容器。

我在项目中搭建的是 Tomcat 7.0 的版本，个人使用时经过一些简单的配置配合 Idea、Eclipse 等 Java 开发软件就能较为方便地使用。

2.2.3 MVC 设计模式

MVC 是软件工程中的一种架构模式，它的全称是 Model View Controller，其实即是 Model（模型），View（视图），Controller（控制器）的组合。它将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

在 Web 应用程序中，MVC 设计模式的三个部分有着各自的业务逻辑：

Model（模型层）：Model（模型）是应用程序中用于处理应用程序数据逻辑的单位对象，通常模型对象负责在数据库中存取数据。举例来说，数据库中每一个数据表都可以对应一个模型，该模型与数据表拥有相同的属性。在我们需要在该数据表中获取数据时，该模型就会被使用。

View（视图层）：View（视图）是应用程序处理数据后进行显示的部分，它提供一个可视化的部分，来较为直观的在用户数据处理完成后产生响应。

Controller（控制层）：Controller（控制器）是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。

我在项目中使用的 SSM 框架也是基于 MVC 设计模式的。SSM 是 Spring、SpringMVC、MyBatis 的首字母缩写，因为 SpringMVC 是 Spring 中的部分内容，所以 SSM 框架可以说由 Spring、MyBatis 两个开源框架整合而成。SSM 框架经常作为数据源较简单的 Web 项目的框架，在下面我也将对 SSM 框架进行一个大概的描述。

2.2.4 Spring 框架

Spring 框架用于简化软件开发而创建，是 JavaWeb 开发的常用框架。它的主要功能是使用 JavaBean 代替 EJB（Enterprise Java Beans 的技术简称，是之前的应用开发组件）。

在 Spring 框架之前，开发过程中编写的代码经常出现类似或者重复的情况，这导致开发过程复杂，开发效率和性能都不太理想。Spring 的出现就是为了解决早期这种应用开发的情况，它致力于提供一个以统一的、高效的方式构造整个应用，并且可以将单层框架以最佳的组合揉和在一起建立一个连贯的体系。

Spring 作为一个容器框架，其最大的两个特性是控制反转（IOC）和面向切面（AOP）。

控制反转：控制反转是 Spring 框架作为容器框架的主要特性。原本我们在 Java 开发中，我们要使用对象时都需要自身创建对象再进行操作。而使用 Spring 框架后，所需对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象，方便开发者使用对象的同时节省空间，促进了松耦合。

面向切面：面向切面是函数式编程的一种衍生范型。利用面向切面可以对业务逻辑的各个部分进行隔离，从而使得业务逻辑各部分之间的耦合度降低，提高程序的可重用性，同时提高了开发的效率。

在项目中使用 Spring 框架需要在项目中配置部分的配置文件，一般在项目中会配置文件搭配注解使用，简化配置并方便进行管理。

2.2.5 SpringMVC 框架

SpringMVC 是 Spring 框架的后续产品，它在 SSM 框架中的主要作用作为

Controller（控制器）部分处理 View（视图层）。SpringMVC 分离了控制器、模型对象、过滤器以及处理程序对象的角色，这种分离让它们更容易进行定制。

在实际的使用中，它的作用类似于 Servlet。配置文件配置完成后，创建对应的文件，并一一添加注解，通过这些注解的引用，前端请求就能转到对应的方法进行处理。但不同的是，SpringMVC 使用注解，一个文件可以处理多个请求，在处理整个项目的请求时，更加轻松，管理起来也更加实用。

2.2.6 MyBatis 框架

MyBatis 框架是基于 Java 的持久层框架，用于定制化 sql、存储过程以及高级映射。

原本在 JavaWeb 开发中需要借助 JDBC 工具实现对数据库的 sql 语句操作，但这样设置 sql 的参数并不方便。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索，使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的模型映射成数据库中的记录。

相比传统的持久层操作，MyBatis 框架下的持久层操作更为简单易学。只需要几个 jar 包和几个配置文件将可以成功搭建出所需的环境。搭建完环境后，开发者就可以开始 sql 映射文件的编写。由于 MyBatis 把 sql 语句统一在这些 sql 映射文件中，实际操作中的编写和管理起来也更为方便。

MyBatis 框架的框架架构流程如下：

（1）加载配置：在开启项目后，服务器会读取我们实现配置的文件以便在后面进行启用。

（2）sql 解析：当前台发送请求调用了事前设定的持久层对象的方法时，会根据方法名称在 sql 映射文件中找出 ID 与之相同的部分，之后传入参数，进行解析，得到最终要执行的 sql 语句和参数。

（3）sql 执行：将之前 sql 解析得到的参数整合到 sql 语句中，然后执行 sql 语句

(4) 结果映射：将操作数据库的结果按照映射的配置进行转换，可以转换成 HashMap、JavaBean 或者基本数据类型，并将最终结果返回。

3 系统分析与设计

3.1 房屋租赁系统概述

自如房屋租赁系统是基于 SSM 框架，以 Java 为主要开发语言，以 MySQL 为数据库集合 Bootstrap, JQuery 等前端框架而成的网站应用。

该系统希望为时下年轻人提供了一个便利，快捷的租房渠道，为广大的租房需求的群体提供有效的帮助。在这里，租客们可以直接浏览上架到该网站上所有的房源以及该房源的相关信息，依次点击寻找到其中自己感兴趣的房源。也可以直接搜索，按照房源的位置，租金，类型进行层层地筛选，在综合比较之后选择自己心仪的房源。而房东们则可以将自己需要出租的房源信息发布到系统上供租客查看，或者对之前上架的房源进行管理。在等待有租客提出预约看房的请求后再进行线下联系，促成最后租房的交易。

3.2 房屋租赁系统需求分析

3.2.1 租客需求分析

租客需求是房屋租赁系统是该系统需要解决的主要需求之一，租客可能的需求如下：

(1) 登录注册：基本上所有的网站在使用时都有登录注册功能，房屋租赁系统也不例外。租客在查看房源信息时可以不进行登录，以游客模式进行使用，只有当需要房东的相关功能时在进行登录。另外注册时可以先不进行实名操作，当需要成为房东上架房源时再实名。

(2) 查看房源：租客使用房屋租赁系统首先需要的功能即是查看周边房源，租客只有在浏览周边房源并进行对比后才能判断选择出最适合自己的房源进行交易。

(3) 房源筛选及房源排序：租客在浏览房源时，房源的数量过多会使得用户查看起来很不方便，所以该系统需要在查看房源页面中添加筛选功能和排序

功能。用户可以以房源位置，面积，类型，租金等等属性为自己筛选房源的标准，排除掉不合条件的优先级较低的房源或以这些属性来排序，让租客只需在剩余的优先级高的房源中进行挑选，节省用户的时间。

（4）收藏房源：有时租客在浏览到自己感兴趣的房源后可能不会第一时间选择该房源来进行租赁，而是期望浏览更多的房源之后再确定心仪的房源。此时系统需要为房源设置收藏按钮，当用户收藏该房源后，可以随时进入自己的个人主页查看这些收藏的房源，以避免后续需要重新从众多房源中再去找之前感兴趣的房源，导致浪费时间让租客产生不好的使用体验。

3.2.2 房东需求分析

房东需求同样是房屋租赁系统是该系统需要解决的主要需求，房东可能的需求如下：

（1）用户实名：房东在上架房源信息的同时也需要将自己的个人信息与房源绑定以方便对房源感兴趣的租客进行联系，所有要想使用房东的功能应该先进行实名操作，填写自己的真实信息后才能使用。

（2）上架房源：上架房源是房屋租赁系统最基本的功能需求，租赁平台需要房东提供房源及房源的例如位置，租金，面积等基本信息，而房东则期望租赁平台展示该房源并吸引租客达成出租房源的目的最终因此盈利。

（3）下架房源：出于各种原因，房东需要更改自己房屋的租赁状态即上架下架自己的房源。例如当用户的房源出现问题不能继续出租时，房东需要房屋租赁系统能够下架该房源，当房源已经被租赁后，房东也期望系统能及时下架房源避免后续出现有租客询问房源信息但实际上该房源已被出租的情况。

（4）重新上架房源：除开房源已被租赁的情况，房东有时也会应其他原因下架自己的房源，而当这些问题被解决后，房东会出现重新上架该房源的需求，所以系统需要有重新上架房源这一功能。重新上架房源可以方便房东进行操作，避免房东重复输入之前已经上架过的房源的信息。但如果房东需要在重新上架房源时又更改房源的信息，则需要再使用上架房源功能。

(5) 房源管理页面：房东一般情况下不会只上架一个房源，当房东上架多个房源时可能需要一个页面把自己上架的这些房源的信息罗列起来方便查看管理，了解自己上架的房源的租赁情况并及时清除已被租赁的房源。

3.2.3 管理员需求分析

基本上所有的系统都需要管理员来对系统进行管理，对房源租赁系统同样如此。管理员的功能需求分析如下：

(1) 个人信息管理：管理员需要查看当前系统中所有的注册的用户的信息，并以实名和未实名两种类型进行区分方便后续的管理，例如当注册用户出现了违规的个人信息时管理员应该可以警告该用户甚至删除该用户的账号信息。

(2) 房东上架请求管理：房东想要上架房屋时不应该直接在用户输入了房源的信息后就直接上架该房源，而是后续交给管理员进行操作。当管理员审核了房源上架的请求后再由管理员进行决策，确定最后该房源是否上架。

(3) 房东下架请求管理：房东下架请求管理的需求与上架请求管理相同，房东在下架自己的房源时需要给出自己下架房源的理由，由管理员进行最后的下架房源操作。

3.3 系统设计

3.3.1 数据库设计

数据库设计是房屋租赁系统中的重要设计部分，需要根据系统中所需的功能来设定数据库中具体的数据表数量及其属性，整个系统中的数据交互都需要借助数据库实现。

房屋租赁系统数据库总共设定了六个数据表，这些数据表的名称、作用及具体属性如下：

(1) User 表（用户账号信息表）：

User 表记录了包括租客和房东在内所有用户账号的信息，其具体的属性有 id（表序号）、telephone（用户注册所用的电话号码）、email（用户注册所用

的邮箱)、**name** (用户注册的用户名, 未实名是该属性随机生成, 实名后则由用户进行设定)、**sex** (用户性别, 用户实名的同时需要补充的属性)、**birth** (用户的出生日期, 用户实名的同时需要补充的属性)、**status** (用户的实名状态, 当为 'true' 时表示已实名, 为 'false' 时表示未实名)。

(2) House 表 (房源信息表)

房东上架的房源的信息都会被储存在该数据表中, 其具体的属性有 **id** (表序号)、**location** (房源的位置信息)、**type** (房源租赁类型)、**rent** (房源月租金)、**area** (房源房间面积)、**lease date** (租赁日期)、**tips** (房源信息备注)、**picture** (展示房源样式的图片名称)、**status** (房源的租赁状态, 总共有上架中、暂停上架、已下架三种状态)、**owner** (房东的用户 **id**, 与 **User** 表外键链接)。

(3) Upper Commit 表 (房东上架请求表)

房东进行上架请求时, 对应的要上架的房源的信息被储存在该数据表中, 其具体的属性有 **id** (表序号)、**location** (要上架的房源的位置信息)、**type** (要上架的房源的租赁类型)、**rent** (要上架的房源的月租金)、**area** (要上架的房源的面积)、**date** (提出上架申请的时间)、**tips** (要上架的房源的备注)、**picture** (展示房源样式的图片名称)、**owner** (上架者即房东的 **id**, 与 **User** 表外键链接)。

(4) Lower Commit 表 (房东下架请求表)

房东进行下架请求操作的相关信息被储存在该数据表中, 其具体的属性有 **id** (表序号)、**cause** (下架房源的原因)、**houseId** (下架的房源的 **id**, 与 **House** 表外键外键链接)。

(5) Collect House 表 (用户收藏房源表)

租客查看房源时收藏的房源的信息被储存在该数据表中, 其具体的属性有 **id** (表序号)、**houseId** (收藏的房源的 **id**, 与 **House** 表外键链接)、**UserId** (收藏该房源的用户的 **id**, 与 **User** 表外键链接)、**date** (收藏该房源的时间)。

(6) Root 表 (管理员账号信息表)

管理员登录后台管理系统时所使用的账号信息表, 该表只有 **id** (序号)、

name（用户名）、password（登录密码）三个属性。

3.3.2 租客功能设计

基于租客可能的需求，进行租客的功能设计。

租客在使用该房源管理系统的流程图如下：

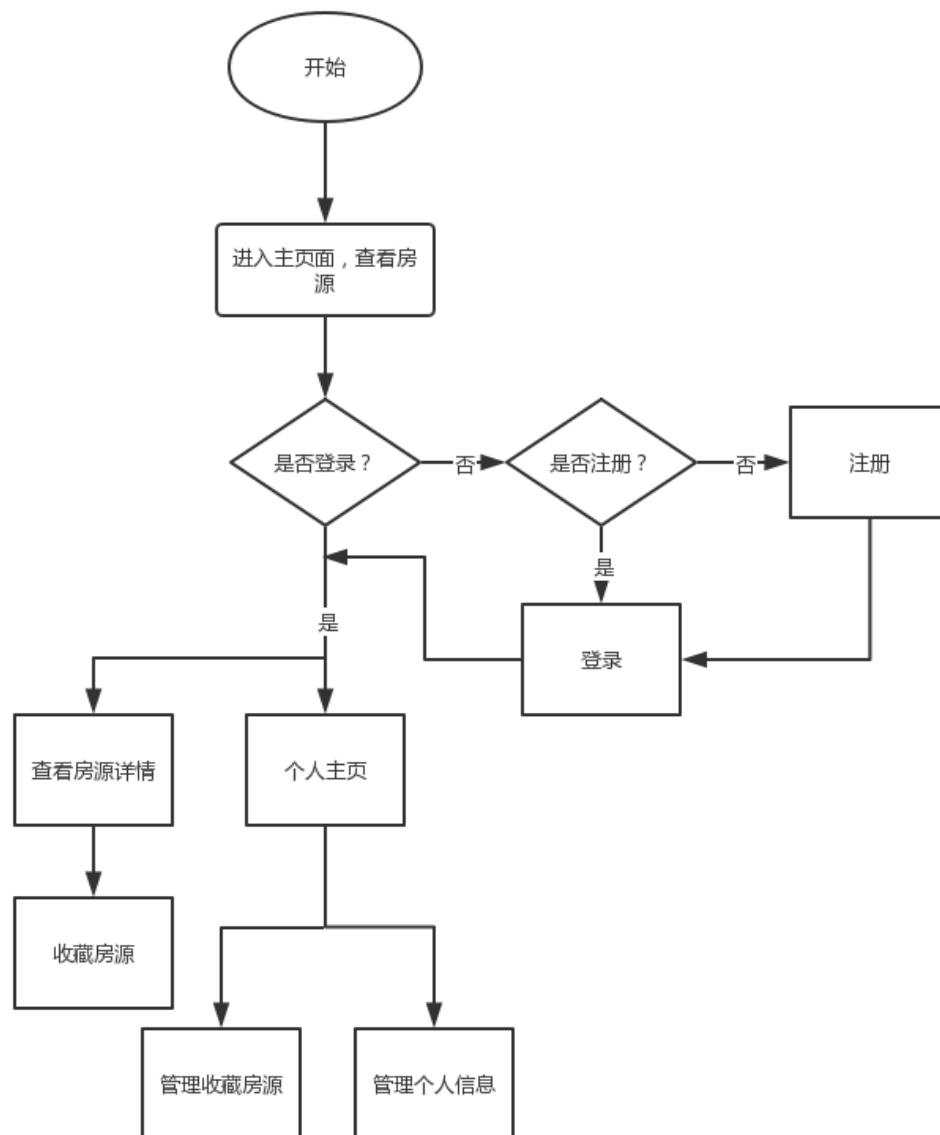


图 3.3.1 租客功能设计流程图

首先系统设定用户在未进行登录时仍然可以以游客模式在主页面查看房源信息，而当用户想要查看房源的详情信息确定是否租赁该房源或想要使用个人主页的功能时则必须进行登录操作。

进行登录操作时，首先用户需要之前在房屋租赁系统进行过注册操作。初次注册时需要用户输入自己的电话、邮箱、及之后登录所用的密码再进行确

认，进行注册后才可以使用登录操作。而用户在注册拥有账号之后，在登录页面输入电话号码、预设的登录密码及验证码即可以进行登录操作。

租客在登录之后，可以继续查看浏览房屋信息，但此时用户可以查看到房源的详细信息特别是房源房东的信息，并可以进行收藏房源操作。租客还可以使用个人主页功能，对自己的个人信息进行管理或管理自己的收藏房源。个人主页的个人信息页面会显示用户注册时所输入的信息，对于实名用户还会额外显示用户的真实信息及性别。租客可以在自己的个人信息页面更改自己的个人信息，或是更改自己的登录密码。另外前面提到用户可以在房源的详情页面收藏该房源，而这些收藏的房源会显示在用户个人主页的房屋预约信息页面，租客可以继续由该页面查看这些房源的详情信息或是取消对部分房源的收藏。

3.3.3 房东功能设计

房东实际上也是先作为租客进行注册，在个人主页进行实名操作后才能成为房东并使用房东可用的相关功能。

房东在使用该房源管理系统的流程图如下：

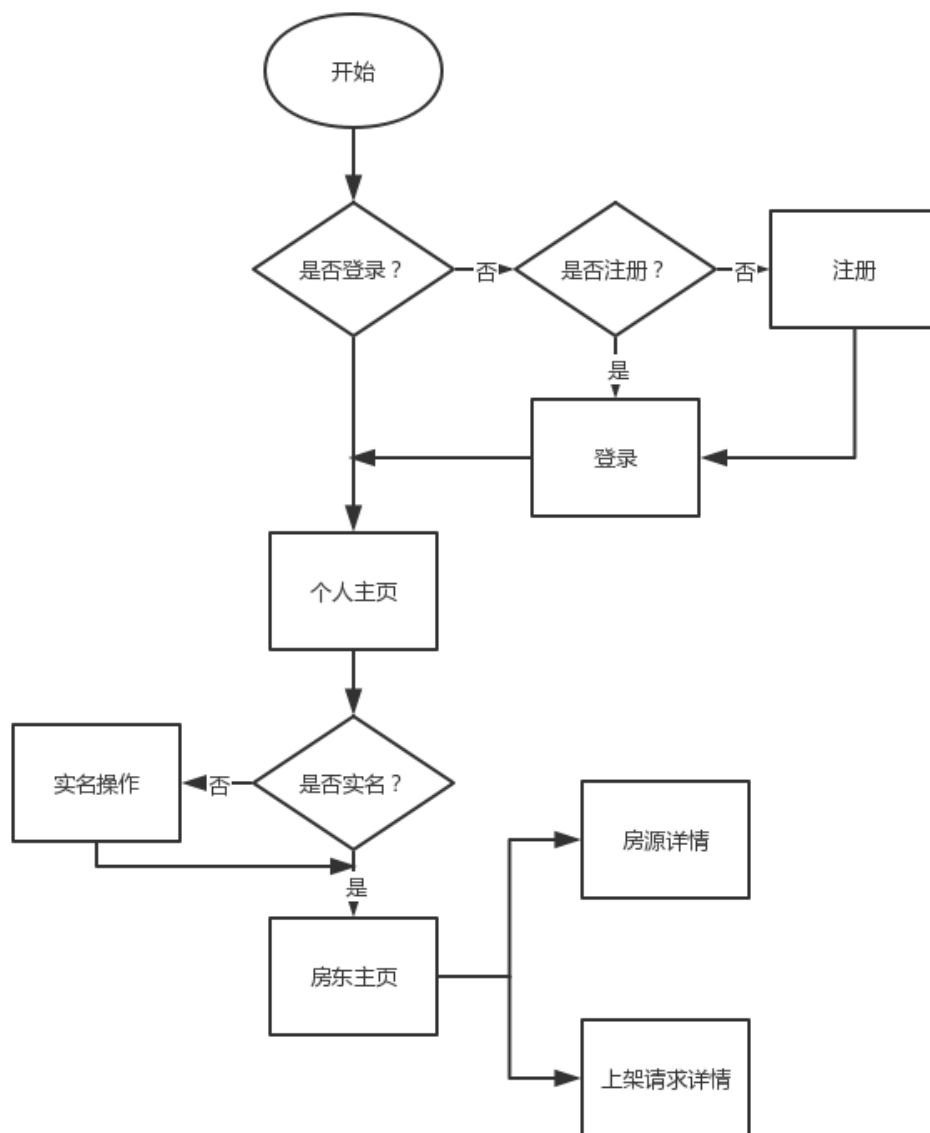


图 3.3.2 房东功能设计流程图

房东可使用的功能必须优先进行登录操作后才能使用，但与租客不同的是房东在进行注册登录一系列操作后还有在个人主页进行实名操作才能真正使用房东的特有功能。

用户实名需要用户补充输入自己的真实姓名信息、性别信息及年龄信息，用户实名后可以使用个人主页的房东主页的查看房源详情功能、查看上架请求

详情功能和上架房源的功能。

房东最首要的功能是上架房源的功能，该功能在系统中设定位于查看上架请求详情页面。房东想要上架房源需要向管理员提出上架房源申请并补充输入房屋的位置、租赁类型、租金、面积等必要信息，由管理员决定是否上架该房源。而在这些上架请求未被管理员处理时，这些请求会罗列在同处的上架请求详情页面，当然房东也可以在管理员处理请求前取消自己的上架请求。

在管理员对房东的上架请求进行批准后，该房源会正式上架显示在系统主页，此时房东可以进行到查看房源页面。查看房源页面会向房东显示房东当前正在上架的房源、暂停上架的房源以及下架的房源，之后房东可以对这些不同状态的房源进行管理。对于当前正在上架的房源，房东可以选择暂停上架该房源；对于当前暂停上架的房源，房东可以选择直接下架该房源或是取消该房源的暂停上架状态重新上架；对于当前出于下架状态的房源，房东也可以重新上架该房源或是直接彻底删除该房源。

3.3.4 管理员功能设计

管理员的功能设计目标分为三类，一类是对用户的个人信息进行管理，一类是对房源的信息进行管理，最后一类是对房东的上架下架请求进行管理，总体的功能设计图如下图：

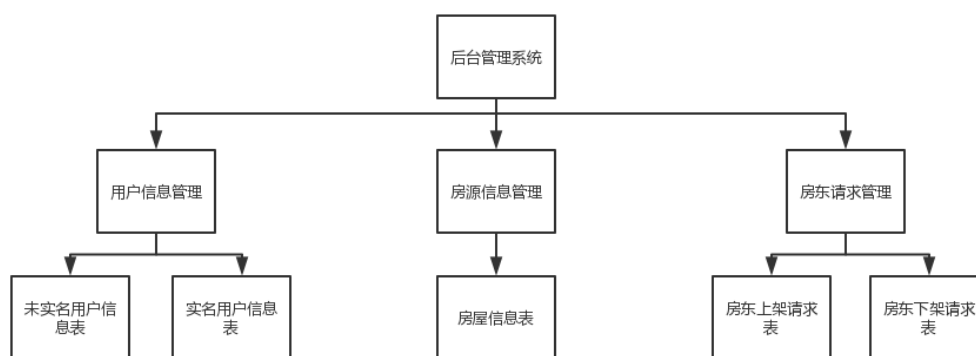


图 3.3.3 管理员功能设计流程图

对于管理员而言，用户信息管理和房源信息管理更多是查看这些数据及排除其中不合理的数据，最主要的是处理房东的请求。房东的上架请求及下架请求会以单元格的形式显示给管理员，管理员一一查看这些请求并决定是点击批准按钮批准这些请求或是拒绝这些请求。

4 系统实现

4.1 数据库实现

在确定了数据库的设计方案后，可以实际开始创建数据库。数据库的创建有两种方式，一种是不借助数据库工具直接调用控制台输入 sql 语句创建数据库，一种是借助数据库工具使用可视化页面创建数据库。这里为了后续方便操作使用

的是 Navicat 数据库工具。

具体的操作图如下：

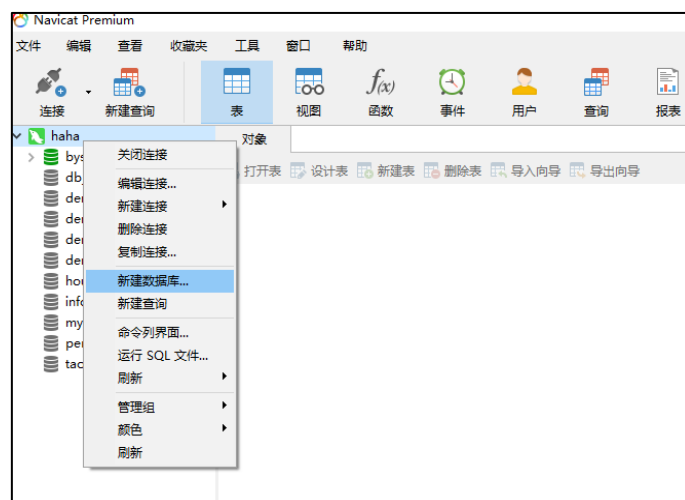


图 4.1.1 创建数据库操作图

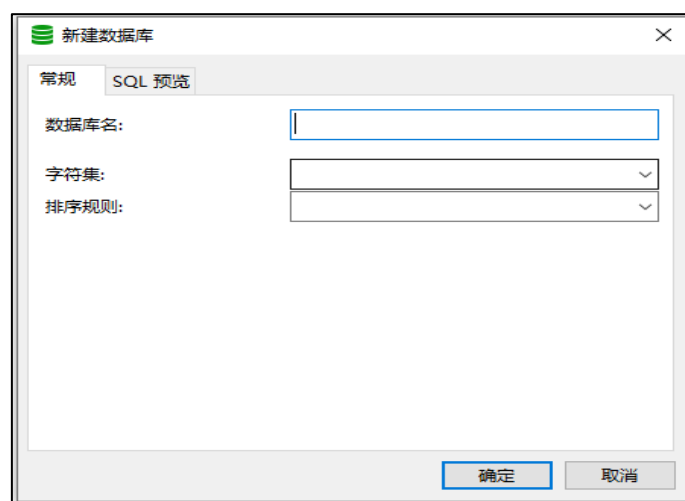


图 4.1.2 创建数据库操作图

在输入了数据库的名称、字符集、排序规则后数据库即可创建完成，也可以使用 sql 语句 ‘create database 数据库名’ 创建数据库。

数据库创建完成后，接下来就是创建数据表的操作。创建数据表继续可以用数据库工具 Navicat 来实现，

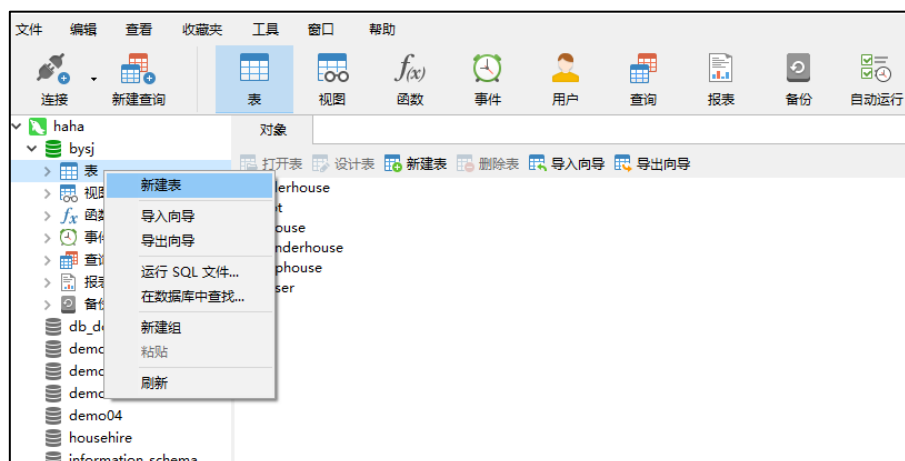


图 4.1.3 创建数据表操作图

点击新建表按钮，补充完成字段后即可完成数据表的创建



图 4.1.4 创建数据表操作图

同样也可以使用 sql 语句（以下方为例）

‘CREATE table User(id int auto_increment,name VARCHAR(20), password VARCHAR(20),PRIMARY KEY(id)); ’ 完成数据表的创建。

注意要先启用该数据库才能进行创建数据表操作。

所有的数据表创建完成后效果如下：



图 4.1.5 系统数据表

通过对这些数据表的修改，整个房屋租赁系统的数据控制得以实现。

4.2 租客功能实现

租客、房东、管理员的功能实现采用了 MVC 设计模式，都是在 SSM 框架的帮助下完成这些功能。

大体的功能实现思路按照以下步骤：

- 1) 完成 Model 层，由预先设计的数据库创建对应的模型。
- 2) 编写 MyBatis 的配置文件及其调用的接口，编写之后要使用的 sql 语句。
- 3) 完成 View 层，编写前端页面，为对应的按钮设置向后端的请求操作。
- 4) 完成 Controller 层，处理前端传过来的请求，获取前端传过来的数据并根据需求调用实现写好的 sql 语句。

系统按照租客功能设计图来一一实现租客功能，具体的实现图如下：

- (1) 登录页面功能实现：



The image shows a login page with a white background and a black border. At the top center is the title "登录" (Login). Below it is a link "没有账号? 可以去注册" (No account? You can go to register). The form contains three input fields: "手机号码:" (Mobile Number) with placeholder "请输入手机号码", "密码:" (Password) with placeholder "请输入密码", and "验证码:" (Verification Code) with placeholder "请输入验证码". The verification code field has a small image showing the code "2 F H D". Below the input fields is a blue button labeled "登录". At the bottom, there is a link "登录即代表同意《自如用户协议》及《自如隐私政策》".

图 4.2.1 系统登录页面

已经注册的用户输入手机号码、密码、验证码后即可进行登录，未注册的用户可以点击上方的注册按钮，这样就会跳转到注册页面。

(2) 注册页面功能实现：



The image shows a registration page with a white background and a black border. At the top center is the title "注册" (Register). Below it is a link "已有账号? 现在去登录" (Already have an account? Go to login now). The form contains four input fields: "电子邮箱" (Email) with placeholder "请输入电子邮箱", "手机号码" (Mobile Number) with placeholder "请输入手机号码", "密码" (Password) with placeholder "请输入密码", and "确认密码" (Confirm Password) with placeholder "请再输入一次密码". There are red asterisks next to the email, mobile number, and password fields. A blue link "邮箱验证" (Email Verification) is next to the email field. Below the input fields is a blue button labeled "注册". At the bottom, there is a link "我已同意《自如用户协议》及《自如隐私政策》".

图 4.2.2 系统注册页面

用户输入电子邮箱、手机号码、登录密码进行确认并且使用的电子邮箱、电

话号码及登录密码符合规则后可以完成注册，如果输入的电子邮箱、电话号码、登录密码不符合规则，系统会给与提示告诉用户当前输入的信息不符合规则不能完成注册。

错误的注册演示：



图 4.2.3 系统错误的注册演示

(3) 系统主页房源显示：

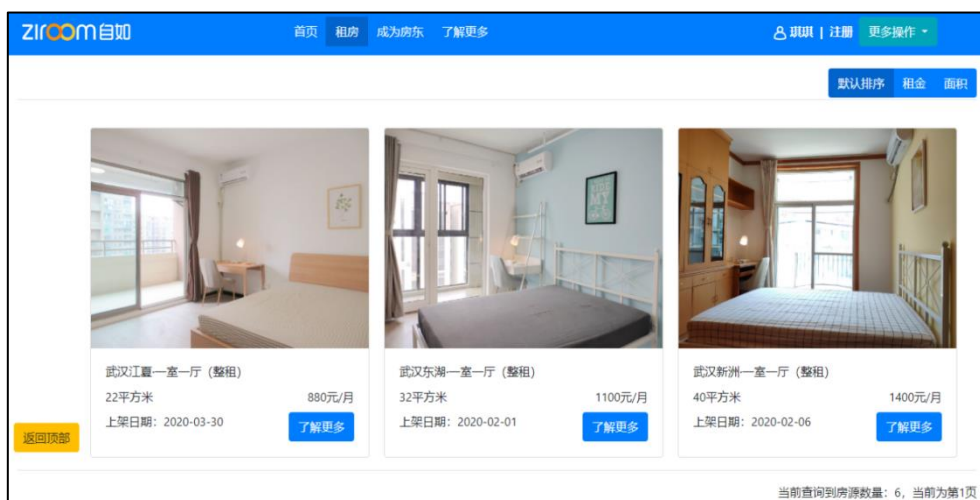


图 4.2.4 系统主页房源显示

房源信息会集中显示在该页面供租客浏览，用户可以查看到房源的图片及其

基本信息。所有房源的数量也会显示在下方供租客参考，页面支持分页功能。如果房源数量过多会隐藏起来不予显示，点击到下一个页面才会继续显示。

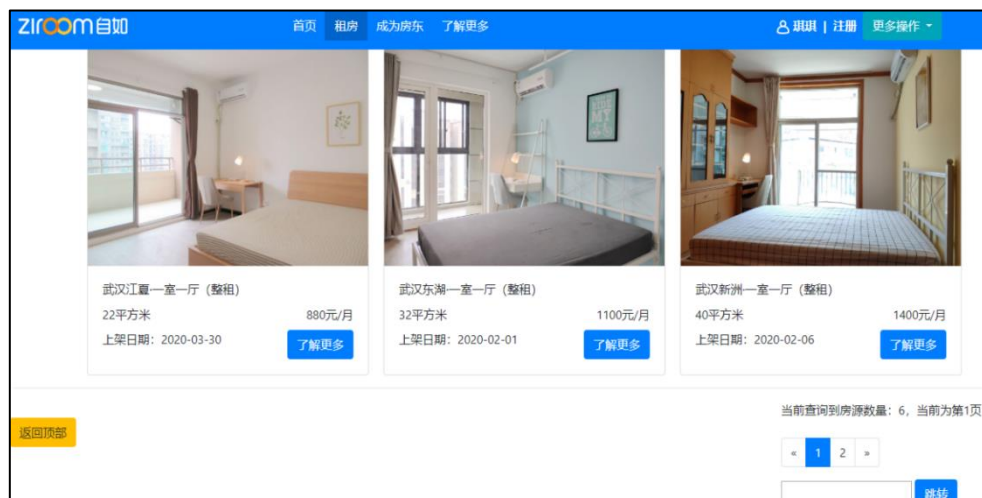


图 4.2.5 系统主页分页功能

(4) 房源的筛选及排序功能:



图 4.2.6 房源的筛选及排序功能

根据房源的基本属性对房源进行筛选以及排序，可以方便用户进行房源的浏览操作。

(5) 房源详情页面:



图 4.2.7 房源详情页面

租客点击主页房源的‘了解更多’按钮可以进入到房源详情页面，房源详情会显示该房源房东的个人信息，根据房东的个人信息租客可以与房东联系达成租房交易，租客也可以先不和房东沟通，点击‘预约看房’按钮可以收藏房源。

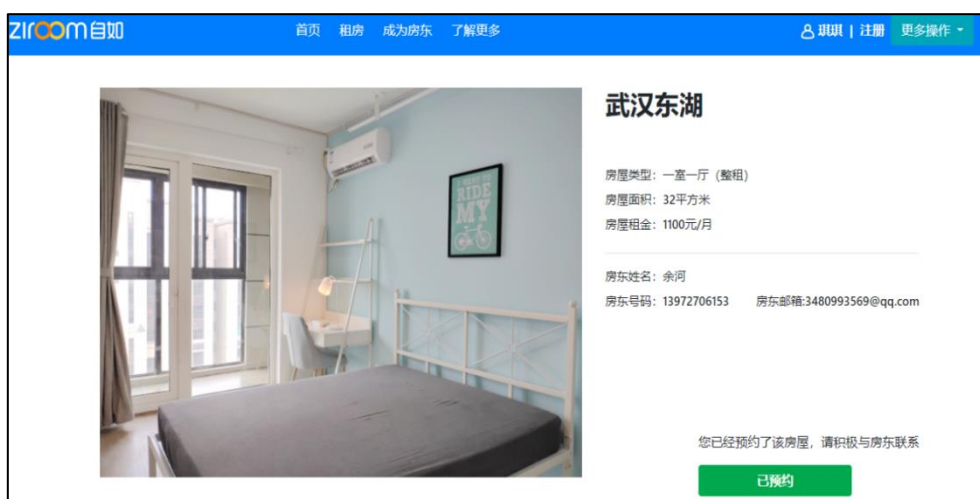


图 4.2.8 收藏房源

(6) 个人信息主页:

点击更多操作，选择个人主页可以进入到用户的个人操作主页，对于未进行实名操作的用户，只有个人信息页面和房屋预约页面可以使用。

个人信息页面会显示当前用户的所有信息，对于未实名用户如下：



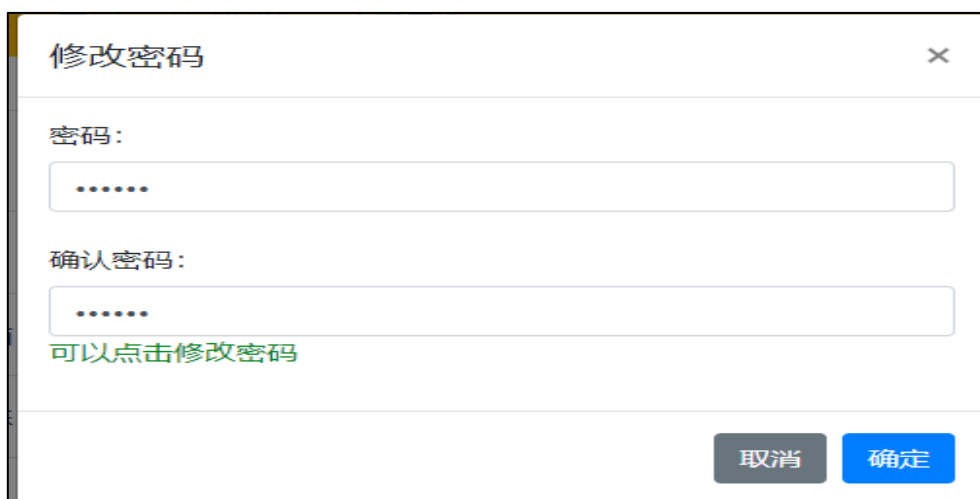
用户名	gh6777	密码	rt7777
电子邮箱	1555544464@qq.com	手机号码	13555689498
实名状态	未实名		

点击以编辑个性头像

用户实名 更改密码

图 4.2.9 未实名用户个人信息

此时用户可以进行用户实名及更改密码操作。



修改密码

密码:

确认密码:

可以点击修改密码

取消 确定

图 4.2.10 更改密码操作

对于实名用户，页面会显示更多的个人信息，



用户名	琪琪	密码	root
性别	女	出生日期	2001-08-08
电子邮箱	3480993569@qq.com	手机号码	root
实名状态	已实名		

点击以编辑个性头像

更改个人信息 更改密码

图 4.2.11 实名用户个人信息

此时用户可以进行更改个人信息操作和更改密码操作。



图 4.2.12 更改个人信息

输入新的信息确认无误后，点击确认即可完成更改个人信息操作。

(7) 房源收藏页面：



图 4.2.13 房源收藏页面

房源收藏页面会显示租客之前在房源详情页面收藏的房源，租客可以点击房源的详情页面快速跳转到该房源的详情页面，也可以点击取消收藏按钮取消对该房源的收藏。

4.3 房东功能实现

房东功能的实现比较特殊，设定需要租客先进行实名操作后才能进行房东的功能操作，房东即上架了房源的实名用户，其特有的功能页面会在下方进行描述。

(1) 用户实名功能：



The image shows a modal window titled "用户实名" (User Real-name Verification). It contains the following fields and controls:

- 姓名:** A text input field with the placeholder "请输入您的真实姓名" (Please enter your real name).
- 性别:** Radio buttons for "男" (Male) and "女" (Female).
- 出生日期:** A date picker showing "2020年04月" (April 2020) with a calendar grid. The selected date is the 23rd.
- Buttons:** "取消" (Cancel) and "确定" (Confirm) buttons.

图 4.3.1 用户实名

用户想要成为房东，需要用户进行实名操作即补充输入性别、出生日期并将用户名更改为真实姓名。在该页面输入了这些信息后点击确认即可完成用户实名认证后面的功能。

(2) 上架房源详情页面：



The image shows a web page for a landlord's listings. The header includes the logo "自如" (ZiRu) and navigation links: "首页" (Home), "租房" (Rent), "成为房东" (Become Landlord), and "了解更多" (Learn More). The user is logged in as "琪琪的个人主页" (QiQi's Personal Homepage).

The page displays two listings under the "上架中" (On Shelf) status:

- Listing 1:** "武汉江夏·一室一厅（整租）" (Wuhan Jiangxia · One Bedroom One Hall (Whole House Rent)). Rent: 880元/月. Listing time: 2020-03-30. Buttons: "房源详情" (Listing Details), "取消上架" (Cancel Listing).
- Listing 2:** "武汉洪山区花山软件新城·一室三厅（合租）" (Wuhan Hongshan District Huashan Software New City · One Bedroom Three Halls (Shared)). Rent: 900元/月. Listing time: 2020-01-15. Buttons: "房源详情" (Listing Details), "取消上架" (Cancel Listing).

图 4.3.2 房东上架房源详情

房东点击房东主页再点击房源详情即可进入上架房源详情页面，该页面会显示房东之前上架的房源。房东可以选择查看房源详情查看自己上架的房源的详细信息，也可以点击取消上架生成房源下架请求暂停该房源的上架，在管理员进行处理后该房源会下架。



图 4.3.3 房东上架房源详情

房源详情页面还会显示当前申请下架的房源及已经下架的房源，房东还可以对这些房源进行后续操作，如对申请下架中的房源进行取消下架申请操作暂停该房源的下架请求。

(3) 上架房源功能:

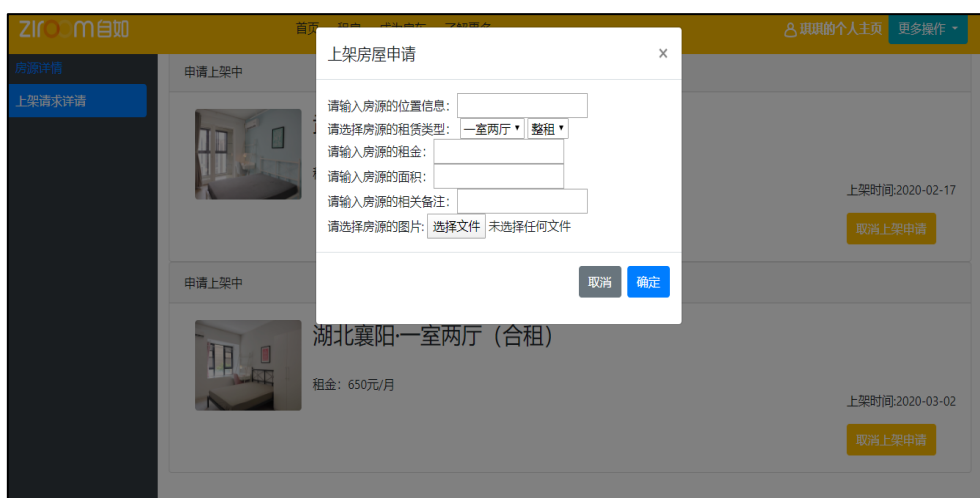


图 4.3.4 上架房源功能

房东点击房东主页再点击上架请求详情页面最后点击上架房源即可使用上架房源功能，出现对应的模态框后补充输入要上架的房源的位置、租赁类型等信息即可生成对应的上架房源请求，管理员批准该请求后该房源会上架到系统主页供租客浏览。

(4) 上架请求详情页面：



图 4.3.5 上架请求详情页面

上架请求详情页面可以进行上架房源操作生成房源上架请求也会显示当前用户提出的但未被处理的房源上架请求。除了浏览这些房源外，房东也可以中途取消该房源的上架申请。

4. 4 管理员功能实现

管理员的功能实现被设置在了另外的后台管理系统页面，管理员已实现的功能效果如下：

(1) 未实名用户表：



用户名	密码	电话	邮箱	操作
gh6777	rt7777	13555689498	1555544464@qq.com	修改

图 4. 4. 1 未实名用户表

该页面显示了所有未实名用户的信息。

(2) 实名用户表：



真实姓名	出生日期	性别	电话	密码	邮箱
琪琪	2001-08-08	女	root	root	3480993569@qq.com
余河	1996-05-24	男	13972706153	rt8888	3480993569@qq.com
杜可星	1999-08-15	女	13977777777	rt7777	2333333333@qq.com

图 4. 4. 2 实名用户表

该页面显示了所有实名用户的信息。

(3) 房源信息表:

房屋地址	租赁类型	租金	面积	上架时间	租赁状态	房东	操作
武汉江夏	一室一厅 (整租)	880	22	2020-03-30	上架中	琪琪	修改
武汉洪山区花山软件新城	一室三厅 (合租)	900	27	2020-01-15	上架中	琪琪	修改
武汉新洲区阳逻	两室两厅 (合租)	1150	30	2020-02-20	申请下架中	琪琪	修改
武汉东湖	一室一厅 (整租)	1100	32	2020-02-01	上架中	余河	修改

图 4.4.3 房源信息一览表

该页面显示了所有房源的信息。

(4) 房东上架请求管理:

房屋上架申请	
申请人:琪琪 租赁类型:一室一厅 (整租) 面积:28平方米	房屋地址:武汉蔡甸区 租金:900元/月 上架时间:2020-02-17 [批准] [拒绝]
房屋上架申请	
申请人:琪琪 租赁类型:一室两厅 (合租) 面积:30平方米	房屋地址:湖北襄阳 租金:650元/月 上架时间:2020-03-02 [批准] [拒绝]

图 4.4.4 房屋上架请求表

该页面显示了所有房东的上架房源申请的信息,管理员可以在这个页面对这些申请进行处理,点击批准按钮该房源会正式上架,点击拒绝按钮该上架申请会删除。

(5) 房东下架请求管理：



图 4.4.5 房屋下架请求表

该页面显示了所有房东的下架房源申请的信息，管理员可以在这个页面对这些申请进行处理，点击批准按钮该房源会进入下架状态，点击拒绝按钮该房源不会下架且房源会回到上架状态。

5 总结与展望

参考文献

- [1]中国报告大厅. 房屋租赁行业定义及分类, 2015-12-08.
<http://m.chinabgao.com/k/fangwuzulin/21100.html>
- [2]百度百科. O2O 模式, 2020-02-10.
<https://baike.baidu.com/item/O2O/8564117?fr=aladdin>
- [3]Casillas J C. O2O Business Patterns and Developing Strategies [J]. International Business Review, 2015, 5(3): 465-479.
- [4]百度百科. bootstrap, 2020-04-10.
<https://baike.baidu.com/item/Bootstrap/8301528>
- [5]百度百科. jquery, 2019-10-18.
<https://baike.baidu.com/item/jQuery/5385065?fr=aladdin>
- [6]简书. 2018-08-30. <https://www.jianshu.com/p/fd7b422d5f93>
- [7]简书. 2018-08-30. <https://www.jianshu.com/p/fd7b422d5f93>

