|  |
| --- |
| 江 西 理 工 大 学  本 科 毕 业 设 计（论文）  题 目：基于Flask的房屋出租租赁平台  学 院：信息工程学院  专 业：网络工程  班 级：151班  学 生：程资源  学 号：1520153330  指导教师：毛伊敏 职称：教授 |

摘 要

房屋出租租赁平台可以使用户发布自己需要出租的房源，也可以寻找自己需要的房源，简化房东与住户交易的复杂度。

本文按照软件开发流程思想，对系统进行了需求分析，在此基础上给出了系统设计方案，并最后完成了系统实现。主要工作如下：

1.系统技术开发方面，采用MVC（模型视图控制器）开发模式和B/S（浏览器服务器）开发结构。系统开发语言主要使用Python语言，前台视图用嵌入HTML、JS和CSS的JSP页面进行展示。

2.系统数据库采用MySQL数据库，对数据库进行概念设计和逻辑设计，用并绘制出数据库的实体关系。

3.根据需求分析要求，将系统划分为用户登录、房屋发布、订单交易、房屋搜素和实名认证等模块。

4.本次开发工具采用PyCharm + ubuntu，并给出重要部分的实现代码和部分界面，同时利用关键技术解决了循环导入、CSRF防护机制等问题。

**关键词：**出租租赁；Python; MySQL

**ABSTRACT**

Rental rental platform allows users to release their own rental housing, but also to find their own housing needs, to simplify the complexity of transactions between landlords and tenants.

In this paper, according to the idea of software development process, the system requirements analysis, on this basis, the system design scheme, and finally completed the system implementation. The main work is as follows:

1. In terms of system technology development, MVC (model view controller) development mode and B/S (browser server) development structure are adopted. System development language is mainly Python language, foreground view with embedded HTML, JS and CSS JSP pages for display.

2. The system database adopts MySQL database, carries out the conceptual design and logical design of the database, and draws the entity relationship of the database.

3. According to the requirements of demand analysis, the system is divided into user login, house release, order transaction, house search, real-name authentication and other modules.

4. This development tool uses PyCharm + ubuntu, and gives the important part of the implementation code and part of the interface, and uses the key technology to solve the problems of cyclic import, CSRF protection mechanism and so on.

**Key words**: leasing; Python; MySQL.

目 录

[第一章 前 言 1](#_Toc7874123)

[1.1系统研究的背景、现状及意义 1](#_Toc7874124)

[1.2论文结构 1](#_Toc7874125)

[第二章 相关技术介绍 3](#_Toc7874126)

[2.1 python开发语言 3](#_Toc7874127)

[2.2 Python 2.7编译环境 3](#_Toc7874128)

[2.3 Ubuntu子环境 3](#_Toc7874129)

[2.4 MySQL 4](#_Toc7874130)

[2.5 PyCharm 4](#_Toc7874131)

[2.6 硬件参数 4](#_Toc7874132)

[第三章 需求分析 5](#_Toc7874133)

[3.1系统需求介绍 5](#_Toc7874134)

[3.2系统建模 7](#_Toc7874135)

[3.2.1数据流分析 7](#_Toc7874136)

[3.2.2系统数据流图设计 8](#_Toc7874137)

[3.3系统框架设计 10](#_Toc7874138)

[3.3.1管理员模块结构设计 10](#_Toc7874139)

[3.3.2房东模块结构设计 11](#_Toc7874140)

[3.3.3房客模块结构设计 12](#_Toc7874141)

[3.4系统可行性分析 13](#_Toc7874142)

[3.4.1技术可行性 13](#_Toc7874143)

[3.4.2经济可行性 14](#_Toc7874144)

[3.4.3操作可行性 14](#_Toc7874145)

[3.5系统性能需求 14](#_Toc7874146)

[3.6技术故障要求 15](#_Toc7874147)

[3.7本章小结 15](#_Toc7874148)

[第四章 系统概要设计 16](#_Toc7874149)

[4.1技术路线 16](#_Toc7874150)

[4.2系统功能概述 16](#_Toc7874151)

[4.3系统功能模块设计 17](#_Toc7874152)

[4.3.1管理员模块功能设计 17](#_Toc7874153)

[4.3.2房东用户模块功能设计 18](#_Toc7874154)

[4.3.3房客模块功能设计 19](#_Toc7874155)

[4.4系统界面设计 21](#_Toc7874156)

[4.4.1输出设计 21](#_Toc7874157)

[4.4.2输入设计 21](#_Toc7874158)

[第五章 数据库设计 22](#_Toc7874159)

[5.1数据库的意义和功能 22](#_Toc7874160)

[5.2数据库的需求分析 22](#_Toc7874161)

[5.3数据库的概念结构设计 22](#_Toc7874162)

[5.4数据库的逻辑结构设计 26](#_Toc7874163)

[**5.**5 数据库表关系图 29](#_Toc7874164)

[第六章 系统详细式设计 30](#_Toc7874165)

[6.1系统开发架构 30](#_Toc7874166)

[6.2基础信息模块实现 32](#_Toc7874167)

[6.2.1主页 33](#_Toc7874168)

[6.2.2注册页面 33](#_Toc7874169)

[6.2.3登录页面 35](#_Toc7874170)

[6.2.4用户详情页面 36](#_Toc7874171)

[6.2.5用户搜索页面 40](#_Toc7874172)

[6.3系统关键现技术实现 41](#_Toc7874173)

[6.3.1图片验证控制 41](#_Toc7874174)

[6.3.2缓存数据同步问题 42](#_Toc7874175)

[第七章 系统测试 43](#_Toc7874176)

[7.1引言 43](#_Toc7874177)

[7.2测试计划 43](#_Toc7874178)

[7.3测试项目 43](#_Toc7874179)

[7.3.1注册登录模块 44](#_Toc7874180)

[7.3.2房屋管理模块 45](#_Toc7874181)

[7.3.3订单管理模块 45](#_Toc7874182)

[7.4测试结论 46](#_Toc7874183)

[第八章 总结与展望 47](#_Toc7874184)

[参考文献 48](#_Toc7874185)

[致谢 49](#_Toc7874186)

[附录 50](#_Toc7874187)

# 第一章 前 言

## 1.1系统研究的背景、现状及意义

随着近年来房价的不断升高，房屋的价格让现在的年轻人高山仰止，而互联网行业的飞速发展，让人们找到了另一种解决办法，利用手机在网上寻求价格类型合适的租房成为人们的更好选择。

据统计，全国房屋租赁交易每年约5000万次。但租房APP占比并不算高，不过也在快速增长中。2013年针对2000多人的调查显示，其中租户通过房产中介租房占比65%，通过专业网站租房的占比6%;而到2015年初的同类调查中，通过中介租房的比例下降到49%，而使用专业网络、APP平台占比达20%。

可见，由于交易成本降低、交易效率提高，移动互联网带来的人与人、人与信息间的对接便利性在租房市场已经发挥出优势。

目前市场上的租房APP种类繁多。有一大部分都是房东或中介入驻，平台提供房东与租客之间联系的纽带。在发展方向上又有以合租与整组房的用户群体为主，和以房产买卖和房屋租赁等综合性服务为主。还有一种是平台先付费把房子租下来，再统一装修，统一出租。

另外还有其他细分垂直方向的短租APP、社交型租房APP、旅行短租APP、共享经济类租房APP等。这些APP中，一些除了租房、签约交租服务，还延伸到了搬家、保洁和维修等服务，基本满足所有租房住的场景化需求。

从运营模式来讲租房和买房流程比较相似，其中收房、带看房再到签约，流程基本相同，不同的是租房的单价和利润均低于买房，利润直接导致租房APP发展很容易出现发展瓶颈。用户对于租房的需求多种多样，每个人都有不同的需求。除了价格便宜、交通方便、还需要考虑卫生间、厨房环境、采光问题、人数限制等一系列的问题，经过各种对比后用户才能定下来。

网上租房可以迅速具体的将房屋信息展示给用户，方便用户进行选择，极大简化了租房交易的流程。也使房屋拥有者能放心的将自己的空余房更容易的转租出去，提升效率。

## 1.2论文结构

本论文总共有七个章节：

1. 前言： 主要说明一下该项目的研究来源、背景和所拥有的实际意义；简述了如今租房行业的发展现状，以及计算机技术在租房行业的使用方式。
2. 相关技术介绍：主要介绍本次项目需要的开发环境和开发工具，包括：Python、JSP、Ubuntu、MySQL、PyCharm。
3. 需求分析：对用户进行调查，分析系统性能，完成系统的数据流图。
4. 系统设计：依据上章的需求分析，设计系统的数据库，实现功能，并画出类图。
5. 系统实现：根据上章的系统具体设计，将这些功能进行编码实现、接口实现、界面实现等，将系统的功能真正的做出来。
6. 系统测试：对系统进行测试；包括整体测试和单个功能测试。

# 第二章 相关技术介绍

房屋出租租赁平台主要使用python编程语言的flask框架进行开发，前端使用JSP技术，开发环境是在windows中的Ubuntu子系统下的python2.7编译环境，数据库使用的是Ubuntu下的MySQL，开发时使用本地服务器，项目完成后部署到腾讯云。下面我会对本项目所用到的相关技术、开发语言、开发环境和开发所使用的硬件、软件进行详细的介绍。

## 2.1 python开发语言

在应用广泛的各种语言中，Python或许是最容易学习和最好使用的。Python代码很容易阅读和编写，并且非常清晰，没有什么隐秘的。Python是一种表达能力非常强的语言，所以，在实现同一个功能时，Python所需要的代码量远小于使用其他语言（C语言、java等）的代码量。

Python的强大功能之一是具有一个非常完整的标准库，通过该标准库，我们可以方便的实现很多功能，比如从Internet下载一个文件，对压缩的文件进行解压，或创建一个web服务器，而这些看似很复杂的功能，只需要几行Python代码就可以实现；除了标准库外，Python还有很多第三方库，其中一些提供了比标准库功能更强大的功能。例如，Twisted网络库和NumPy数值型库。大多数第三方库都可以通过Python Package Index，网址为<http://pypi.python.org/pypi>进行访问。

## 2.2 Python 2.7编译环境

在Ubuntu里执行语句：

sudo apt-get install python2.7

进行Python环境安装，安装完成后可以输入语句:

Python

确认。

## 2.3 Ubuntu子环境

Ubuntu系统使用的为windows应用商店下的Ubuntu 18.04 LTS.

打开windows的开发者选项，启用WSL，然后在应用商店下载Ubuntu，等待安装完成后即可。

Ubuntu系统下软件安装较为简单，对于项目代码的编译更为有利。

## 2.4 MySQL

MySQL是最流行的关系型数据库管理系统，在WEB应用方面MySQL是最好的RDBMS(Relational Database Management System: 关系数据库管理系统)应用之一。由瑞典的MySQL AB公司开发，目前属于Oracle公司。它把是数据保存到不同的表中，提升了速度并提升了灵活性。

MySQL是开源的，可以免费使用。

MySQL支持大型数据库，可以处理很大型的数据库。

MySQL采用标准的SQL数据语言形式。

MySQL支持多种语言，多种系统。

MySQL可以定制。

因此使用MySQL可以很好的帮助我们建立所需要的数据库。

## 2.5 PyCharm

PyCharm是由捷克公司JetBrains开发的，具有代码分析、图形化调试、集成测试等功能的计算机编程集成开发环境(IDE)，主要用来开发Python。

主要功能如下：

代码分析与辅助功能，拥有补全代码、高亮语法和错误提示；

项目和代码导航：专门的项目视图，文件结构视图和和文件、类、方法和用例的快速跳转；

重构：包括重新命名，提取方法，引入变量，引入常量、pull，push等；

支持网络框架： Django, web2py 和 Flask；

集成Python 调试器；

集成单元测试，按行覆盖代码；

Google App Engine下的Python开发；

集成版本控制系统：为Mercurial， Git， Subversion, Perforce 和 CVS提供统一的用户界面，拥有修改以及合并功能。

## 2.6 硬件参数

操作系统：Windows下的Ubuntu子系统

处理器： Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70Ghz

RAM： 8.00GB

# 第三章 需求分析

## 3.1系统需求介绍

整个界面设计一定要简洁明了，使用户易于操作，主要需要实现以下功能：

1. 主页

1.1 最多5个房屋logo图片展示，点击可跳转至房屋详情页面

1.2 提供登陆/注册入口，登陆后显示用户名，点击可跳转至个人中心

1.3 用户可以选择城区、入住时间、离开时间等条件进行搜索

1.4 城区的区域信息需动态加载

2. 注册

2.1 用户账号默认为手机号

2.2 图片验证码正确后才能发送短信验证码

2.3 短信验证码每60秒可发送一次

2.4 每个条件出错时有相应错误提示

3. 登陆

3.1 用手机号与密码登陆

3.2 错误时有相应提示

4. 房屋列表页

4.1 可根据入住离开时间、区域进行筛选，并可进行排序

4.2 房屋信息分页加载

4.3 区域信息动态加载

4.4 筛选条件更新后，页面立即刷新

5. 房屋详情页

5.1 需展示的详细信息参考设计图

5.2 提供预定入口

5.3 若是房东本人查看房屋信息时，预定入口不显示

6. 房屋预定

6.1 由用户确定入住时间

6.2 根据用户确定的入住离开时间实时显示合计天数与总金额

7. 我的爱家

7.1 显示个人头像、手机号、用户名（用户名未设置时为用户手机号）

7.2 提供修改个人信息的入口

7.3 提供作为房客下单的查询入口

7.4 提供成为房东所需实名认证的入口

7.5 提供作为房东发布房屋信息的入口

7.6 提供作为房东查询客户订单的入口

7.7 提供退出的入口

8. 个人信息修改

8.1 可以修改个人头像

8.2 可以修改用户名

8.3 登陆手机号不能修改

8.4 上传头像与用户名分开保存

8.5 上传新头像后页面理解显示新头像

9. 我的订单（房客）

9.1 按时间倒序显示订单信息

9.2 订单完成后提供评价功能

9.3 已评价的订单能看到评价信息

9.4 被拒绝的订单能看到拒单原因

10. 实名认证

10.1 实名认证只可进行一次

10.2 提交认证信息后再次进入只能查看信息，不能修改

10.3 认证信息包含姓名与身份证号

11. 我的房源

11.1 未实名认证的用户不能发布新房源信息，需引导到实名认证页面

11.2 按时间倒序显示已经发布的房屋信息

11.3 点击房屋可以进入详情页面

11.4 对实名认证的用户提供发布新房屋的入口

12. 发布新房源

12.1 需要用户填写全部房屋信息

12.2 房屋的文字信息与图片分开操作

13. 客户订单（房东）

13.1 按时间倒序显示用户下的订单

13.2 对于新订单提供接单与拒单的功能

13.3 拒单必须填写拒单原因

13.4 若客户进行了订单评价，需显示

14. 退出

14.1 提供退出功能

## 3.2系统建模

系统建模是根据项目需求分析，建立起相关的模型，用图像的方式将客户的需求清晰的呈现出来。可以采用多种方式来进行建模分析：① 基于场景的分析方法，如使用例图；② 基于数据的分析方法，如数据流图和E-R图；③ 基于类的分析方法，如类图和协作图；④ 基于行为的分析方法，如状态图和顺序图。

本次我采用的是基于数据的建模分析方法，利用数据流图来直观的反应系统对事件的处理过程，采用自上而下的方式来对项目的各个功能进行描述。

数据流图需要用到的基本符号如下3-1图：

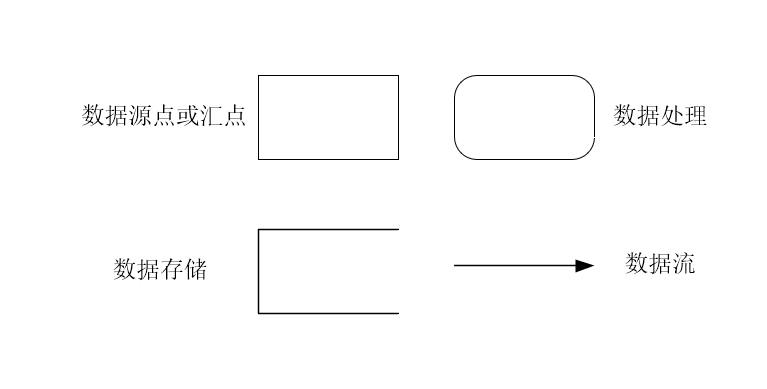


图3-1 数据流图的基本符号

### 3.2.1数据流分析

数据源点或汇点：

* 系统管理员(主要管理用户信息、房屋信息、订单信息)
* 房东(主要管理个人信息、房屋发布、订单信息)
* 房客(主要管理个人信息、发起订单、订单支付)

数据流：

搜索、修改、预定、接单、拒单、支付、评论、退出。

数据存储：

房屋信息、订单信息、评论信息、用户个人信息。

数据处理：

提交订单、发布房源、评论提交。

### 3.2.2系统数据流图设计

1. 顶层数据流图

在该系统中，系统管理员管理系统后台的数据，房东管理房屋的信息，房客提交订单，其数据流图如下图3-2：

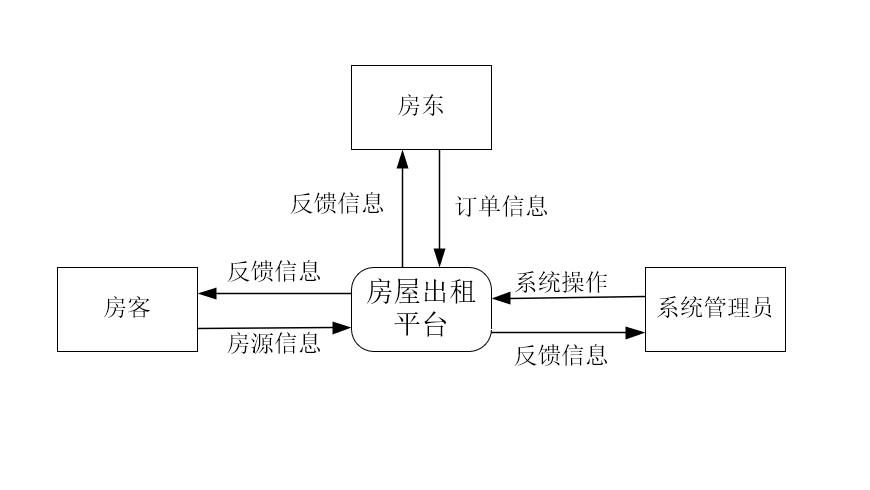


图3-2 顶级数据流图

1. 系统数据流图第一步细化

对系统的功能进行一步细化，可以将房屋出租平台分为“发布房源”、“处理订单”、“完成订单”三个功能；同时增加“订单信息”、“房屋信息”两个数据流；数据流图如下图3-3：

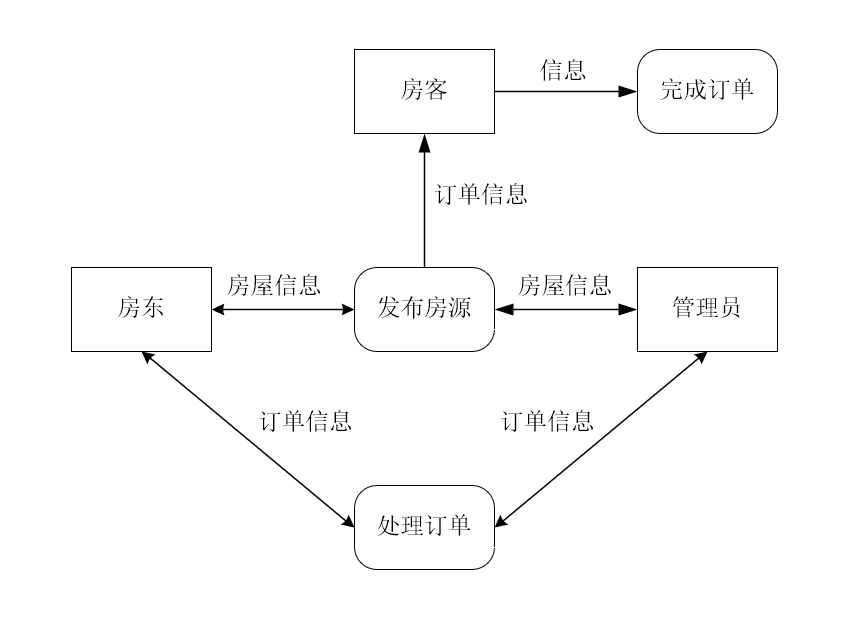


图3-3 房屋出租系统第一步细化数据流图

1. 系统数据流图第二步细化

对系统的功能再进一步细化，添加“订单支付”、“订单评价”、“受理订单”、“拒绝订单”；数据流添加“支付信息”和“拒绝信息”；其数据流图如下图3-4所示：

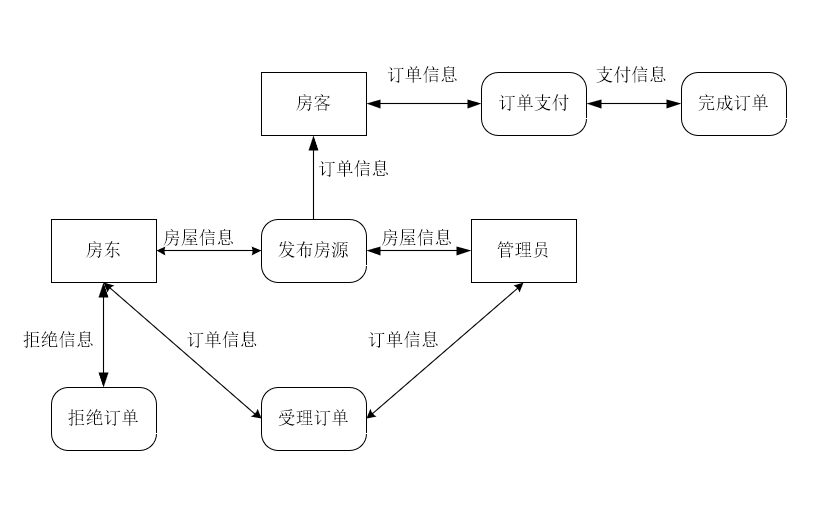


图3-4 房屋出租系统第二步细化数据流图

## 3.3系统框架设计

通过数据流图可知，该系统可分为三个模块，分别为：管理院模块、房东模块、房客模块；管理模块可以管理后台的数据；房东模块可以发布房源和处理订单；房客模块可以搜索房源、预定房源和订单支付，系统总的框架图如下图3-5所示：

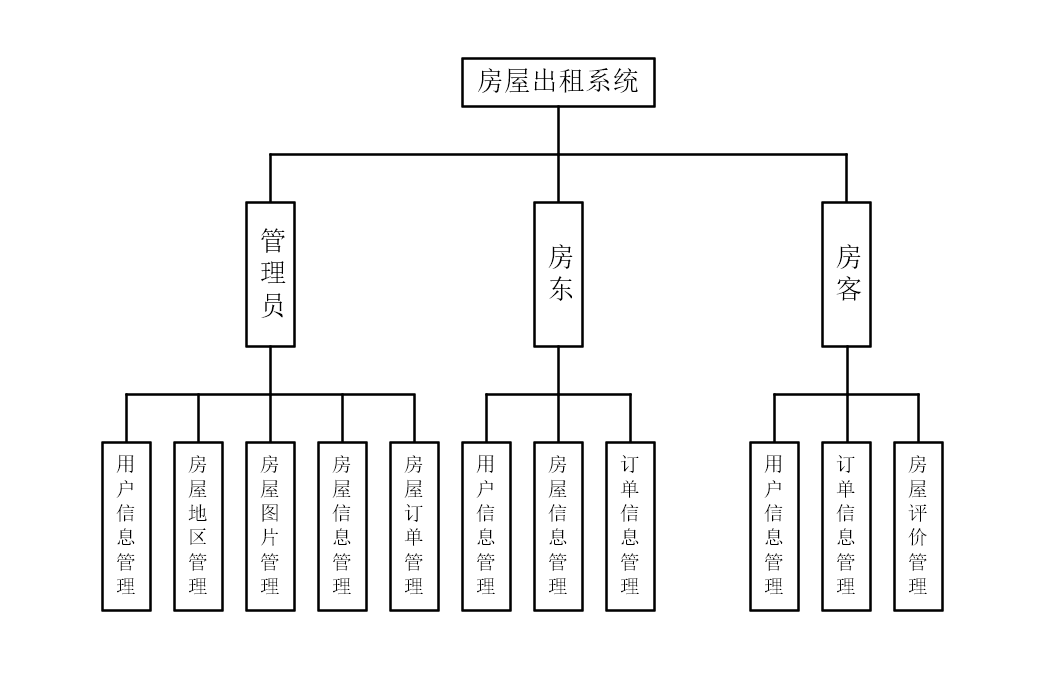


图3-5 系统总框架图

### 3.3.1管理员模块结构设计

管理院模块主要用来实现管理员对系统后台数据（房屋地区、用户信息、房屋信息、房屋图片和订单信息）的管理。

1. 修改登录密码：管理员登录后台管理页面后可以修改密码。
2. 房屋地区管理：添加、删除或修改房屋所在地区。
3. 用户信息管理：管理员可以查看所有房东和房客的个人信息，并可以进行修改删除。
4. 房屋信息管理：管理员可以对房东发布的房源信息进行查看删除和修改。
5. 房屋图片管理：管理员可以查看房屋图片的编号并进行删除。
6. 房屋订单管理：管理员可以查看订单状态和订单的详细信息，并可以对订单进行修改订单状态、删除订单等操作。

管理员模块的结构图如图3-6所示：

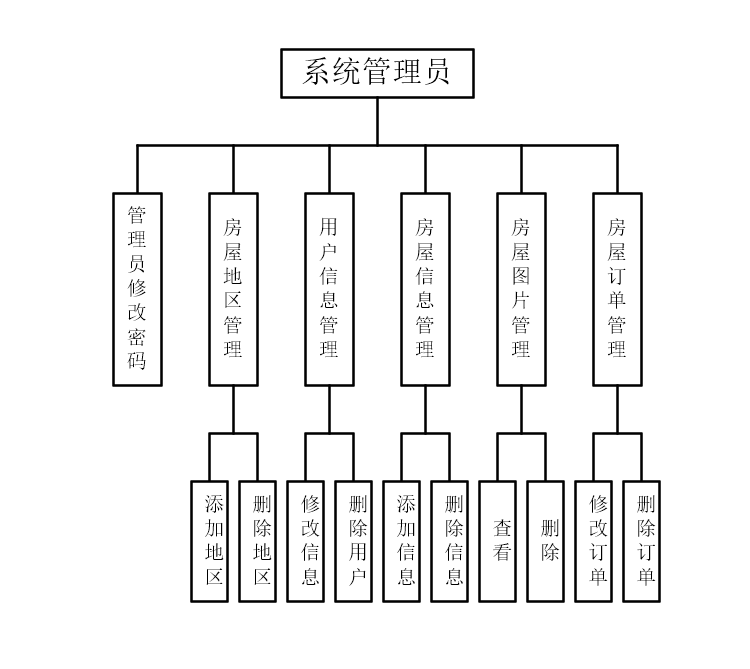


图3-6 系统管理员模块结构图

### 3.3.2房东模块结构设计

房东管理模块主要包括用户注册登录、自身个人信息的修改、房源详细信息查看和订单管理；可以在拒绝订单后填写拒单信息。

1. 个人信息管理：需要注册登录，可以修改用户名和用户头像，发布房源必须要进行实名认证。
2. 房源信息管理：房东可以查看自己发布的房源的详细信息，也可以发布新的房源。
3. 订单管理：可以查看客户订单信息，对等待接单的订单可以接单或拒单，拒单要填写拒单理由。

房东模块结构如图3-7所示：

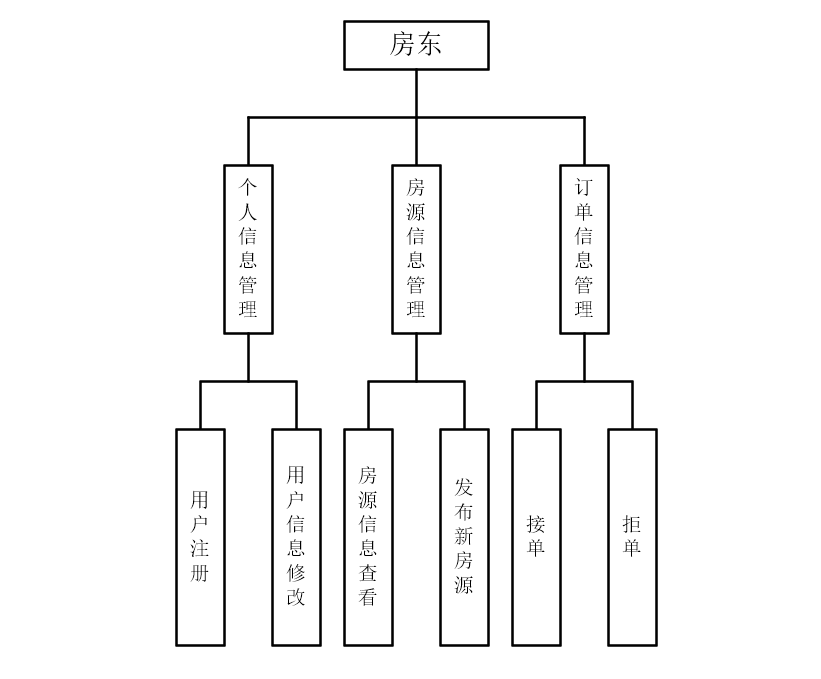


图3-7 房东模块结构图

### 3.3.3房客模块结构设计

房客通过手机号码注册登录后可以使用该系统，可以搜索查看已发布的房源信息，历史用户对该房源的评价，可以在“我的订单”里查看自己已完成订单和对已完成订单的评价，也能查看预定的房源房东是否接单，对房东已接单的个人订单可以进行支付，来完成订单。

1. 用户信息管理：用户注册登录，可以修改自己的用户信息和实名认证。
2. 房屋搜索管理：可以根据房屋地区和租住时间来搜索房源，查看房屋详细信息和历史评价。
3. 我的订单管理：房客可以在“我的订单”页面查看历史租房记录，也可以查看预定房屋进度，对已完成订单可以添加评论信息。

房客模块结构设计如图3-8所示：

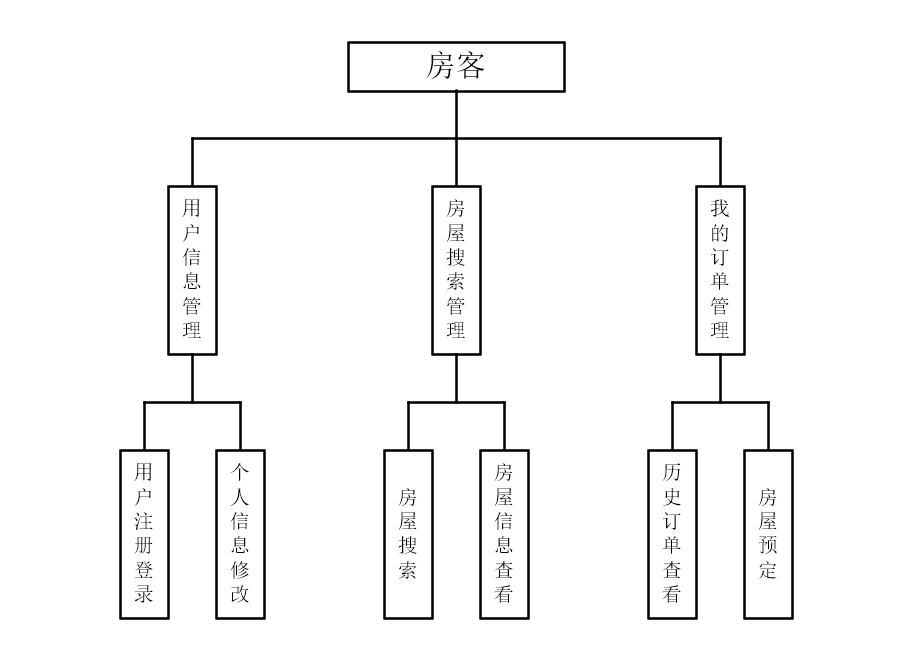


图3-8 房客模块结构图

## 3.4系统可行性分析

系统的可行性分析主要是从经济可行性、技术可行性和 操作可行性三个方面进行分析；通过利用系统分析得到的结果，来对系统的设计开发进行合理的安排，制定可行的方案，对系统开发所要使用的资源、时间进行合理的安排，让整个系统的开发流程满足软件工程所要求的科学性，使整个系统能够顺利的开发完成。该体统采用前后端分离的方式，前端控制页面展示效果，后端提供数据，控制数据的增删改查，与前端页面无关，因此可以更有利与项目的开发和以后的升级维护。

系统开发的一个很重要的环节就是系统的可行性分析，通过从系统的用户需求、资源供应、操作要求等几个方面的研究，从技术、经济、工程等角度对项目进行调查研究和分析比较，判断以我们现在的技术、资源是否有能力来完成项目，是否可以实现项目需要的功能，做出的系统是否可以满足用户的需求，是否可以在大众中得到广泛的推广。

下面我们分别从技术可行性、经济可行性和操作可行性三个方面分析该项目

### 3.4.1技术可行性

本项目在Windows10 64位的操作系统下开发，并在Ubuntu18.04 LTS 子系统下的Python2.7环境下编译，数据库使用MySQL，使用Python2作为开发语言，主要使用Python的flask框架来开发。Python是如今最为火爆的语言之一，对比其他开发语言，Python的语言更加清晰简洁使得它调试起来简单得多，作为动态语言可以让初学者更快的上手。Python可以让初学者把精力放在编程对象和思维方法上，而不用去担心语法、类型等等外在因素。因为Python的火爆，之前学习了一些基础的Python，对这种语言有了一定的了解，觉得可以使用这种语言做一些项目，正好也可以借这次项目，对自己的能力进行一下提高。之前也学习过软件工程这门课程，对一个项目的生命周期有所了解。因此该项目在技术方面的实现是可行的。

### 3.4.2经济可行性

随着计算机技术的不断发展和进步，计算机在其他行业的应用越来越广泛，极大提高了企业的竞争力。作为租客，相比在线下一家接一家的找出租房，用手机在租房网站上搜索更为方便，更加节约成本和时间；房东利用互联网的传播性也能提升房屋出租交易的速度，而只需要花费很少的开支。因此房屋出租系统在经济方面也是可行的。

### 3.4.3操作可行性

在这个信息化的时代，人们对于电脑和手机的使用早已不再陌生，各类型软件的操作方式也都十分熟悉。本系统的操作方式符合用户普遍的操作习惯，十分简单易学，用户只需通过手机浏览器进入该系统，利用手机号码注册登录后就可以在首页预览到其他房客发布的房源，也能按照自己的需求来搜索房屋，找到满意的房子，房东在个人中心里“我的房源”可以查看已发布的房源或发布新房源，和以往使用过的软件没有什么太大区别，所以本系统在操作可行性方面也是满足要求的。

由上可得，本系统的开发从可行方面分析是可行的。

## 3.5系统性能需求

根据需求分析，房屋出租租赁系统的性能方面要满足以下规定：

* 系统要运行流畅，操作简单直接；
* 可以满足同时多个用户的访问，并不会发生异常；
* 系统要易于后期的升级维护；
* 在于数据库进行交互时，响应速度要快。

## 3.6技术故障要求

无论是在来发测试的时候，还是在上线系统在运行中难免会出现一些异常、错误，例如页面获取不到数据库数据，数据提交时发生异常，用户登录出错等问题，这时就需要系统根据不同的问题给我们相应的提示，以便我们及时更正错误，以免浪费不必要的时间。

## 3.7本章小结

本章主要对系统进行详细的分析，来对系统的性能和需求有一个详细的规划，让我们对系统开发有一个合理的任务安排，能更有效率的开发出系统，让我们对系统的开发周期有一个明确的依据。

# 第四章 系统概要设计

## 4.1技术路线

在这个信息急剧膨胀的社会，我们不得不说人类正进入一个崭新的时代——信息时代。信息时代最显著的特征就是计算机网络的应用。计算机网络从最开始的集中式计算，经历了C/S阶段，现在已经发展到目前最流行的B/S计算模式。如今颇具争论和影响力的就是C/S和B/S架构，C/S是一种历史悠久并且技术成熟的架构技术，而B/S是从C/S中演化而来的，具有一些创新，尤其是在web信息时代更是应用广泛。

C/S一般是建立在局域网基础上的，范围较小，局域网之间在通过专门的服务器提供连接和数据交换服务；而B/S是建立在广域网的基础上，不必是专门的网络硬件环境，拥有更好的适应范围，一般只要有浏览器和操作系统就可以。但也是由于B/S是建立在广域网上，对安全的控制里相对较弱，其面向的是不可知的用户群体。我们这个系统面向的用户并不只是集中在一个地区的用户，因此需要选用适用范围更广泛的B/S架构软件更为合适。

## 4.2系统功能概述

房屋出租租赁系统主要实现功能：房东通过本系统发布房源详细信息，顾客通过本系统查找房源，提升了房东与房客之间进行交易的效率和体验，本系统大体上可分为三个功能模块：

管理员模块：系统管理员可以通过后台管理系统查看用户信息、房屋信息、订单信息、地区信息等，也可以对这些信息进行一些操作处理。

房东模块：房东可以在系统网页上注册登录，实名认证后发布房源的相关信息，方便房客搜索查看，然后选择预定。

房客模块：房客注册登录完成后，可以按照地区和入住时间来搜索房源信息，选择预定，待房东确定接单后，支付完成订单。

当然，除了这些主要功能模块外，还有一些比较小的子系统模块。例如，修改头像、修改用户名、订单支付、查看评价信息等功能，有助于用户与其他用户之间的区分，通过查看评价信息，房客可以对房源有更加充分的了解，以做出更加准确的判断，房东也能通过这些房客的体验评价总结自己房子的优缺点，对不足之处加以改正，营造更好的口碑，提升租户的入住体验。

## 4.3系统功能模块设计

### 4.3.1管理员模块功能设计

管理员登录后台管理页面后，能够完成的操作应有：

1. 修改管理员密码；
2. 管理用户信息，并可以对相关数据进行相应的操作；
3. 管理订单信息，可以查看用户的订单详情并对其做出修改；
4. 管理房源信息，可以对其进行相应的操作；
5. 管理地区信息，可以增加或删除地区；
6. 管理房源图片信息，对不合规的图片进行删除操作；

其具体的流程图如下图4-1：



图4-1 管理员流程图

### 4.3.2房东用户模块功能设计

房东用户进入系统并登录后，可以完成的操作有：

1. 管理个人信息，可以修改用户头像，用户名，实名认证；
2. 管理房源信息，可以查看房源详细信息或发布新房源；
3. 管理订单信息，可以对用户提交的订单进行相关的操作。

具体的流程图如图4-1所示：



图4-2 房东模块流程图

### 4.3.3房客模块功能设计

房客注册登录系统后，可以完成的操作有：

1. 管理用户信息，包括修改用户名、头像、实名认证；
2. 管理订单信息，可以对订单进行支付操作；
3. 管理评价信息，对已完成订单进行评价。

具体流程图如图4-3所示：



图4-3 房客模块流程图

系统整个流程图如图4-4所示：



图4-4 系统总流程图

## 4.4系统界面设计

界面美观、操作易用性，维护成本低时一个系统界面设计的关键。因此，用户进入系统界面的时候，系统的主要界面一定要把系统的主要功能展示出来，方便用户的使用，例如房屋预览、注册、登录、和地区、时间搜索房源等。主要使用HTML和JSP来实现前端页面的展示效果，其地区、房屋预览图片信息从后端数据库通过json传到前端展示。

### 4.4.1输出设计

输出设计主要是使管理信息系统输出满足用户需求的信息，将我们现需要的信息从数据库中传到系统的前端页面上，供用户查看，方便了用户的使用。传回的信息能够满足用户需求，直接体现了系统的成功与否，因此输出设计是项目系统设计的核心。

### 4.4.2输入设计

输入设计是用户与信息系统之间的纽带。输入数据的正确性直接影响到处理结果的正确性，如果输入的数据有误，那即便计算机的处理过程正确，也不能得到正确的信息。同时，输入设计决定着人机交互的效率。

输入设计为了提高效率和减少错误应当遵循以下原则：

1. 控制人工输入量。数据录入一般都要有人的参与，其数据输入速都和计算机相比十分缓慢。因此，要尽量控制人工输入数据的总量，人只需输入基本数据，其他数据通过计算机由系统自动产生。
2. 减少输入延迟。采用尽可能高传输效率的方法，提高数据传输效率。
3. 减少错误输入。输入设计中应采取多种输入校验方式和有效性验证技术，减少错输入错误。
4. 避免额外步骤。尽量避免不必要的输入步骤。
5. 简化输入过程。输入设计在为用户提供纠错和输入校验的同时，保证输入过程简单易用。

# 第五章 数据库设计

## 5.1数据库的意义和功能

数据库是存储和管理所有数据的仓库，系统各类数据的都要保存在数据库之中，数据库的主要功能由增加、删除、修改、查询和统计等操作。系统所用到的数据都以一定的格式存放在数据库之中，方便了用户对其数据进行相应的操作。所以，数据库的设计在系统开发中是至关重要的。

## 5.2数据库的需求分析

数据库的设计结构应该要满足系统需要实现的功能，用户登陆时的账号密码验证，房屋的详细信息，个人用户的订单记录，房客对已完成订单的评价，房东发布过的房子，订单的状态等都需要通过数据库来记录。因此，数据库的需求分析是十分必要的。

## 5.3数据库的概念结构设计

将需求分析得到的用户需求抽象为信息结构既为数据库的概念结构设计，能够将事物与事物之间联系清晰明了的展现出来，让人容易理解。通过E-R图，可以概念结构较好的描述出来。根据需求分析，系统数据库的概念结构如下：

1. ih\_admin\_info(管理员)信息的实体包括的属性有：id(管理员编号)、admin (管理员账号)、password(管理员密码)，具体E-R图5-1如下所示：

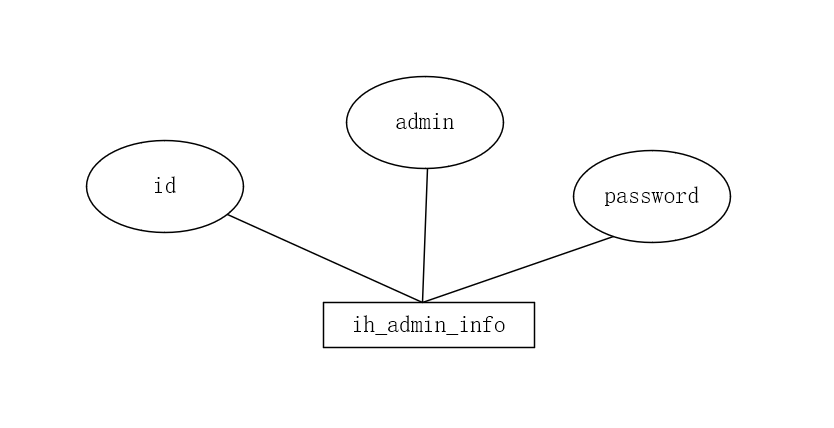


图5-1 管理员信息实体图

1. ih\_user\_profile(用户)信息的实体图包括的属性有：id(编号)、name(昵称)、password\_hash(加密后的密码)、mobile(手机号)、real\_name(真实姓名)、id\_card(身份证号)、avatar\_url(用户头像路径)，具体E-R图如下图5-2所示：

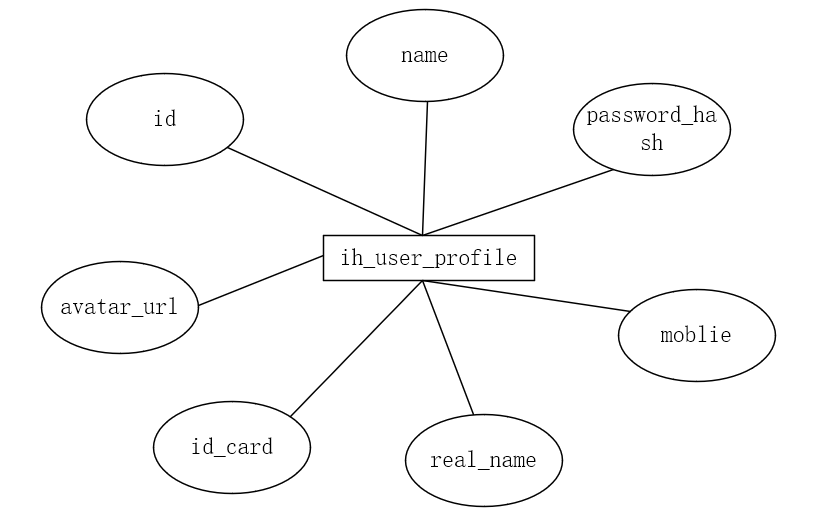


图5-2 用户信息实体图

1. ih\_area\_info(地区)信息的实体图包括的属性有：id(编号)、name(地区)具体E-R图如下图5-3所示：

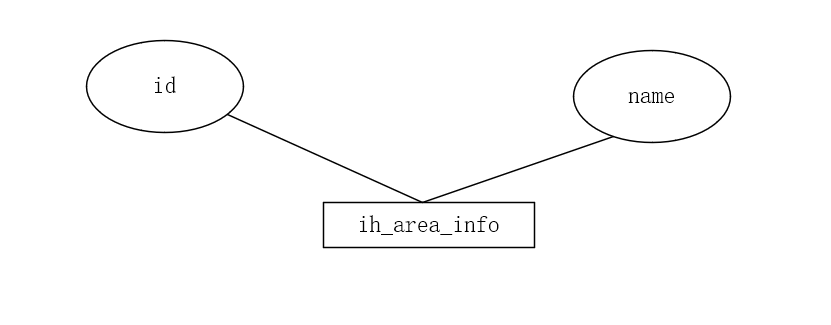


图5-3 地区信息实体图

1. ih\_house\_info(房屋)信息的实体图包括的属性有：id(编号)、user\_id(用户id)、area\_id(地区id)、title(房屋标题)、price(定价)、address(详细地址)、room\_count(房屋数量)、acreage(房屋面积)、unit(户型)、capacity(宜住人数)、beds(床的尺寸)、deposit(定金)、min\_days(最少入住天数)、max\_days(最大入住天数)、order\_count(订单数)、index\_image\_url(主页显示图片路径)，具体流程图如下图5-4所示：

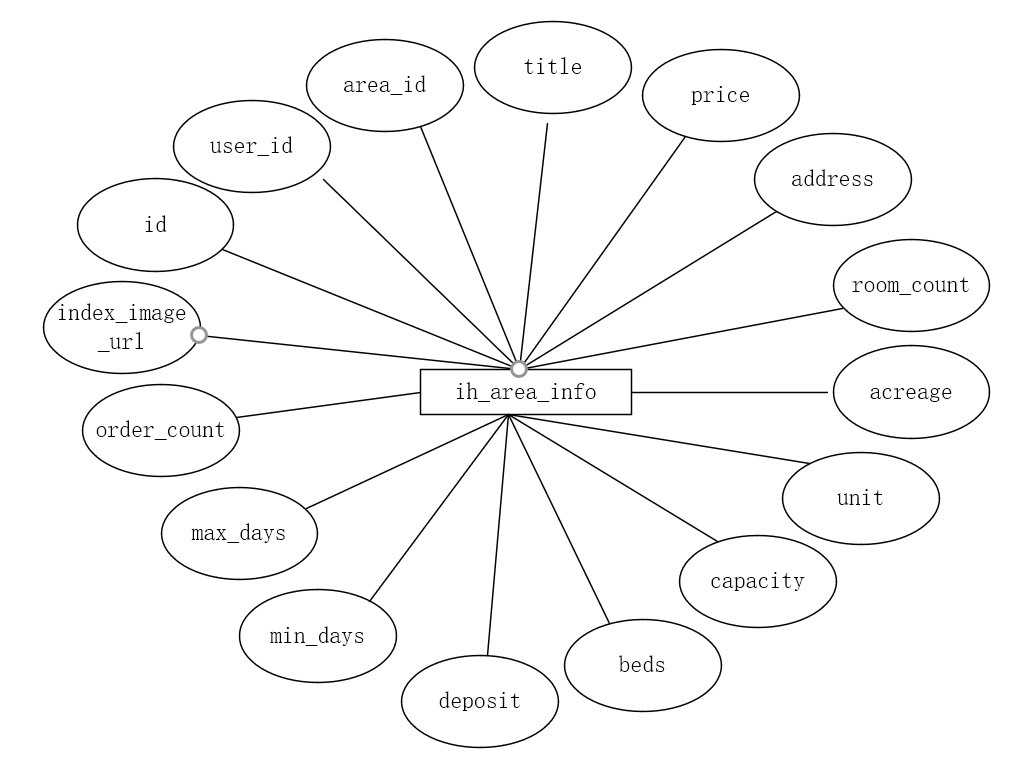


图5-4 房屋信息实体图

1. ih\_house\_image(房屋图片)信息的实体图的属性有：image\_id(图片id)、url(图片路径)、house\_id(对应房屋id)，具体E-R图如下图5-5所示：

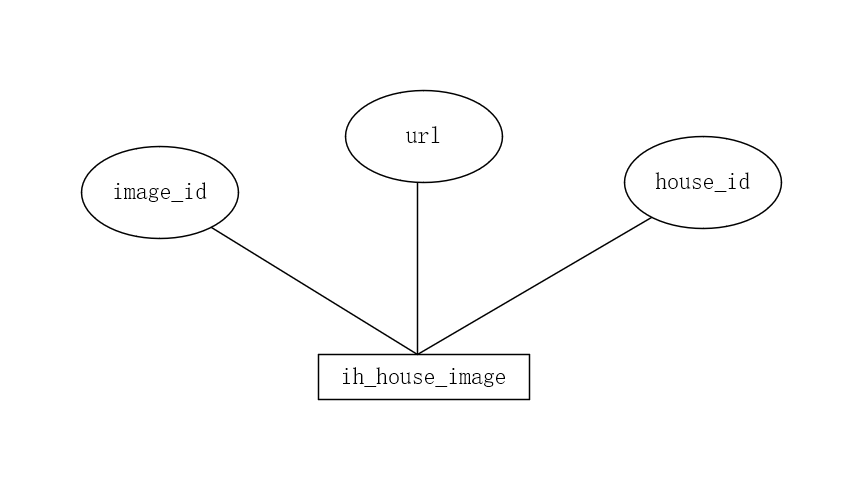


图5-5 房屋图片信息实体图

1. ih\_house\_facility(房屋已选设施表)信息实体的属性有：house\_id(房屋id)、facility\_id(设施id)，具体E-R图如下图5-6所示：

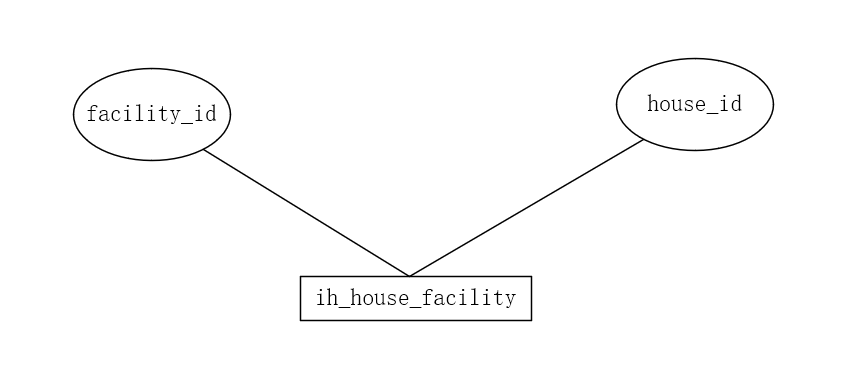


图5-6 房屋设施实体图

1. ih\_facility\_info(房屋可选设施)信息实体的属性有：facility\_id(设施id)、name(设施名称)，具体E-R图如下图5-7所示：

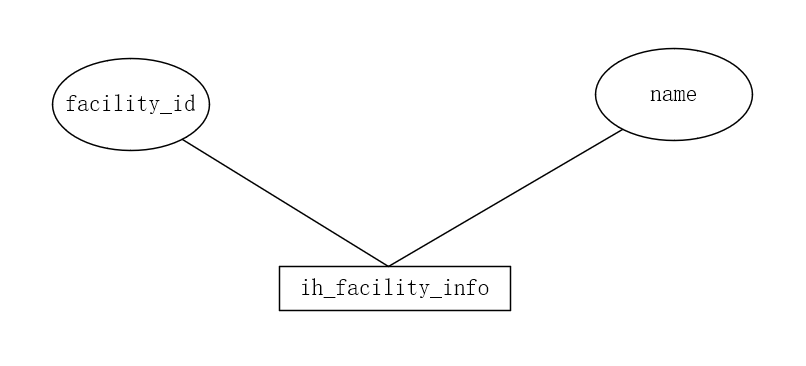


图5-7 房屋可选设施实体图

1. ih\_order\_info(订单详情)信息实体的属性有： id(订单编号)、user\_id(用户编号)、house\_id(房屋编号)、begin\_date(开始日期)、end\_date(结束日期)、house\_price(价格)、amount(总价)、days(入住天数)、status(订单状态)、comment(订单评价)、trade\_no(支付编号)，具体E-R图如图5-8所示：

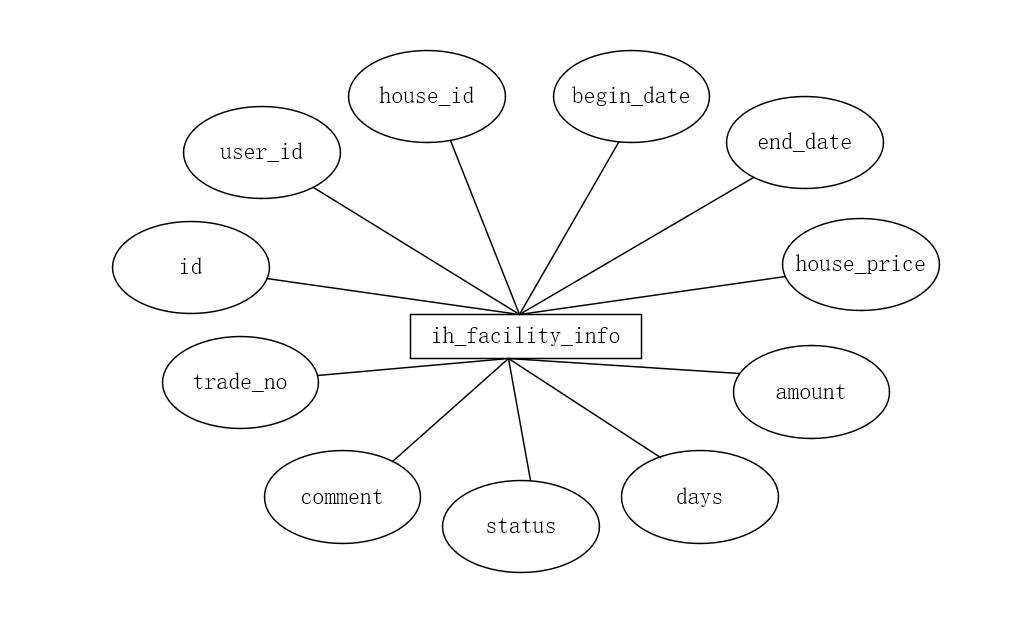


图5-8 订单详情实体图

根据以上各个E-R图，可以画出总的系统E-R图5-9：



图5-9 系统总的E-R图

## 5.4数据库的逻辑结构设计

数据库的逻辑结构设计阶段是把数据库概念1结构的基本E-R图转换成为MySQL数据库可支持的数据模型逻辑结构。E-R图转化为关系模型就是将实体类型和实体之间的关系转化为关系模型，以及确定这些关系模型的属性和码。系统的数据库E-R图转变为关系数据模式如下：

1. 用户信息转变为关系数据模型：

用户表Ih\_user\_profile (id, *name*, password\_hash, *mobile*, real\_name, id\_card, avatar\_url) 下划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| id | 用户编号 | int | 11 | 否 | - | 主键 |  |
| name | 昵称 | varchar | 32 | 否 | - | 外键 |  |
| password\_hash | 密码 | varchar | 128 | 否 | - | - |  |
| mobile | 手机 | varchar | 11 | 否 | - | 外键 |  |
| real\_name | 真实姓名 | varchar | 32 | 是 | - | - |  |
| id\_card | 身份证号 | varchar | 20 | 是 | - | - |  |
| avatar\_url | 头像 | Varchar | 128 | 是 | - | - |  |

1. 地区信息转换为关系模型：

地区信息表ih\_area\_info(id, name) 下划线标注的属性为主键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| id | 地区编号 | Int | 11 | 否 | - | 主键 |  |
| name | 地区名称 | varchar | 32 | 否 | - | - |  |

1. 房屋详细信息转换为关系模型：

房屋详细信息表ih\_house\_info(id, *user\_id*, *area\_id*, title, price, address, room\_count, acreage, unit, capacity, beds, deposit, min\_days, max\_days, order\_count, index\_image\_url) 下划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| id | 房屋编号 | int | 11 | 否 | - | 主键 |  |
| user\_id | 用户id | int | 11 | 否 | - | 外键 |  |
| area\_id | 地区id | int | 11 | 否 | - | 外键 |  |
| title | 房屋标题 | varchar | 64 | 否 | - | - |  |
| price | 单价 | int | 11 | 是 | - | - |  |
| address | 详细地址 | varchar | 512 | 是 | - | - |  |
| room\_count | 房间数 | int | 11 | 是 | - | - |  |
| acreage | 房间大小 | int | 11 | 是 | - | - |  |
| unit | 户型 | varchar | 32 | 是 | - | - |  |
| capacity | 宜住人数 | int | 11 | 是 | - | - |  |
| beds | 床的尺寸 | varchar | 64 | 是 | - | - |  |
| deposit | 定金 | int | 11 | 是 | - | - |  |
| min\_days | 最少入住 | int | 11 | 是 | - | - |  |
| max\_dayd | 最多入住 | int | 11 | 是 | - | - |  |
| order\_count | 订单数 | int | 11 | 是 | - | - |  |
| index\_image\_url | 主页图片 | vatchar | 256 | 是 | - | - |  |

1. 订单详情信息转化为关系模型：

订单详情信息表 ih\_order\_info(id, *user\_id*, *house\_id*, begin\_date, end\_date, days, price, amount, status, comment, trade\_no) 下划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| id | 订单编号 | int | 11 | 否 | - | 主键 |  |
| user\_id | 用户id | int | 11 | 否 | - | 外键 |  |
| house\_id | 房屋id | int | 11 | 否 | - | 外键 |  |
| begin\_date | 开始日期 | datetime | - | 否 | - | - |  |
| end\_date | 结束日期 | datetime | 11 | 否 | - | - |  |
| house\_price | 价格 | int | 11 | 否 | - | - |  |
| amount | 总价 | int | 11 | 否 | - | - |  |
| days | 租住天数 | int | 11 | 是 | - | - |  |
| staus | 订单状态 | enum | 11 | 是 | - | 外键 |  |
| comment | 评论 | text | - | 是 | - | - |  |
| trade\_no | 支付编号 | varchar | 80 | 是 | - | - |  |

1. 房屋可选设施信息转化为关系模型：

房屋可选设施信息表ih\_facility\_info(id, name) 下划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| id | 编号 | int | 11 | 否 | - | 主键 |  |
| nane | 设施名称 | varchar | 32 | 否 | - | - |  |

1. 房屋已选设施信息转化为关系模型：

房屋已选设施信息表ih\_house\_facility(house\_id, facility\_id) 下划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| house\_id | 房屋id | int | 11 | 否 | - | 主键 |  |
| facility\_id | 设施id | int | 11 | 否 | - | 主键 |  |

1. 房屋图片信息转化为关系模型：

房屋图片表ih\_house\_image(id, *house\_id*, url) 划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| id | 图片编号 | int | 11 | 否 | - | 主键 |  |
| house\_id | 房屋编号 | int | 11 | 否 | - | 外键 |  |
| url | 图片路径 | varchar | 256 | 否 | - | - |  |

1. 管理员信息转化为关系模型：

管理员信息表ih\_admin\_info(id, admin, password) 划线标注为表的主键，斜体为外键。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | 长度 | 允许空值 | 默认值 | 主键或外键 | 备注 |
| Id | 管理员编号 | id | 11 | 否 | - | 主键 |  |
| admin | 管理员账号 | varchar | 32 | 否 | - | - |  |
| password | 密码 | varchar | 32 | 否 | - | - |  |

## **5.**5 数据库表关系图

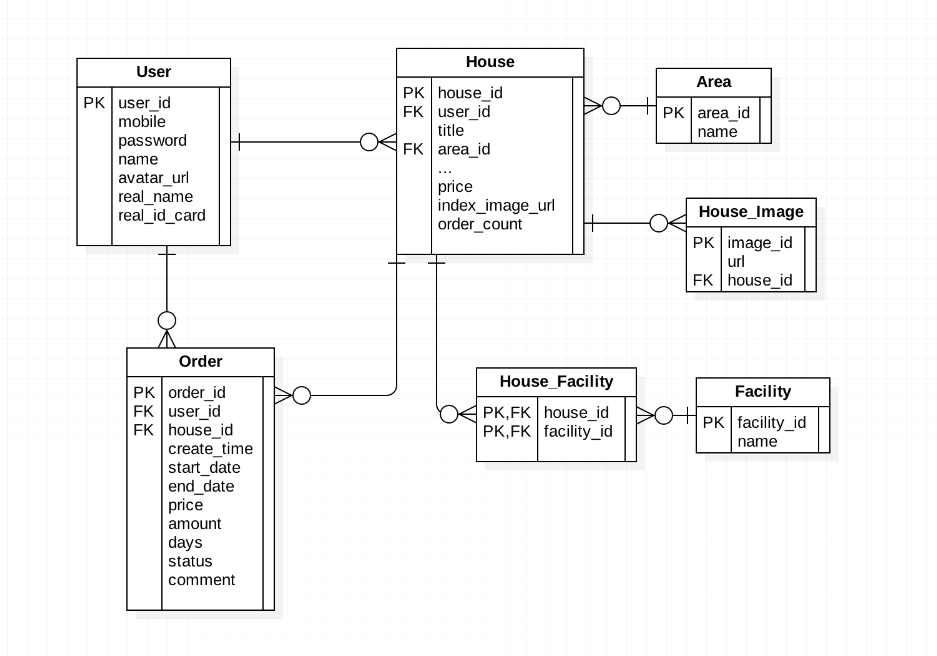


图5-10 数据库表关系图

# 第六章 系统详细式设计

## 6.1系统开发架构

1. 本系统主要使用Python作为主要开发技术，使用MVC开发模式，这些技术前面的章节已经讲过了，便不再重复叙述。

2. 系统源程序目录结构，如图6-1所示。

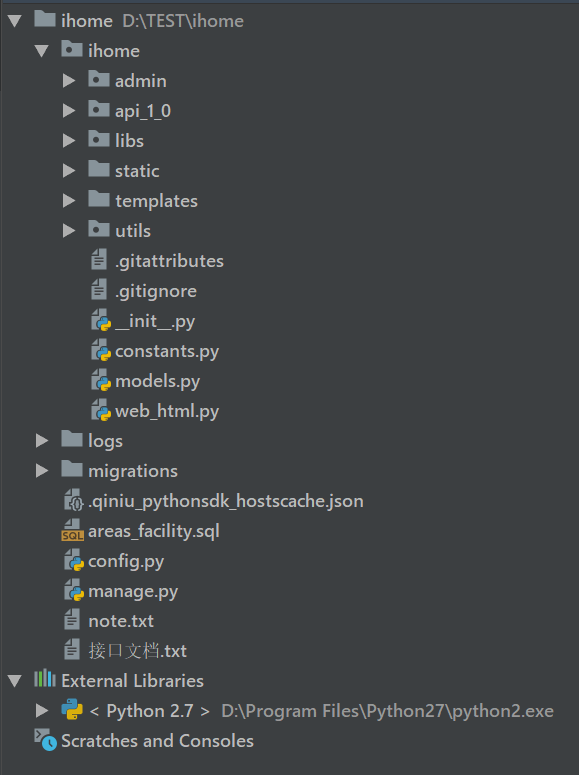


图6-1 程序目录结构图

ihome为该项目的名称，其下还有一个ihome文件夹，包括了admin（管理员模块设计）、api\_1\_0（主要功能模块）、libs（短信验证接口）、static（前端静态文件）、templates（管理员页面）、utils（图片上传接口）、constants.py（常量）、models.py（数据库）、web\_html.py（获取静态文件蓝图）config.py（连接数据库）、manage.py（主要起始文件）。具体描述见下：

1. admin
2. api\_1\_0

包含文件夹keys（支付宝公钥和软件私钥）、\_\_init\_\_.py（蓝图）、demo.py、house.py（房屋信息模块，主要功能有：从后端数据库中获取城区信息、将前端发过来的房屋基本信息数据保存到是数据库中、将房屋图片保存到数据库中、获取房东发布的房源信息条数、获取主页幻灯片展示的房屋基本信息、获取房屋详情、获取房屋的列表页面（搜索页面））、order.py（订单模块，主要功能有：保存订单、查询用户订单、接单 拒单、保存订单评论信息）、passport.py（注册登录模块，主要功能有：注册、用户登录、检查登录模块、登出）、pay.py（支付宝支付模块，主要功能有：发起支付宝支付、保存支付订单结果）、profile.py（用户信息模块，主要功能有：设置用户头像、修改用户名、获取个人信息、获取用户的实名认证信息、保存实名认证信息）、verif\_code.py（注册验证模块，主要功能有：获取图片验证码、获取短信验证码）

1. libs

主要提供一个短信验证的第三方APi，从而获取用户注册时需要用到的短信验证码服务，本项目使用的是聚合数据提供的短信验证接口。

1. static

static目录下包括了项目前端页面的所有html、css、js文件，images提供了测试用的图片样张，faviaon.ico是网站logo。

Js文件夹下包括前端展示页面的所有js文件，其目录结构如下图6-2所示：

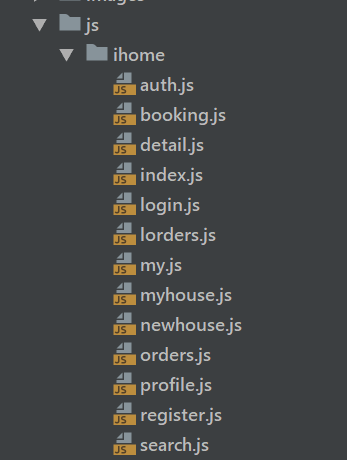


图6-2 js目录结构图

其中的js文件详情如下：

|  |  |
| --- | --- |
| auth.js | 实名认证页面 |
| booking.js | 房屋预定页面 |
| detail.js | 房屋详情页面 |
| index.js | 项目主页 |
| login.js | 用户登录页面 |
| lorders.js | 房东订单页面 |
| my.js | 用户详情页面 |
| myhouse.js | 我的房源页面 |
| newhouse.js | 发布房源页面 |
| order.js | 客户订单页面 |
| profile.js | 用户信息修改页面，可以修改头像和用户名 |
| register.js | 用户注册页面 |
| search.js | 用户搜索页面 |

表6-1 js目录文件列表

1. utils

该文件夹下文件主要用来实现调用第三方图片存储接口，将前端上传的图片通过该目录下的image\_storage.py上传到七牛云上，并将图片的文件名返回给后端。上传的图片内容如下图6-2所示：

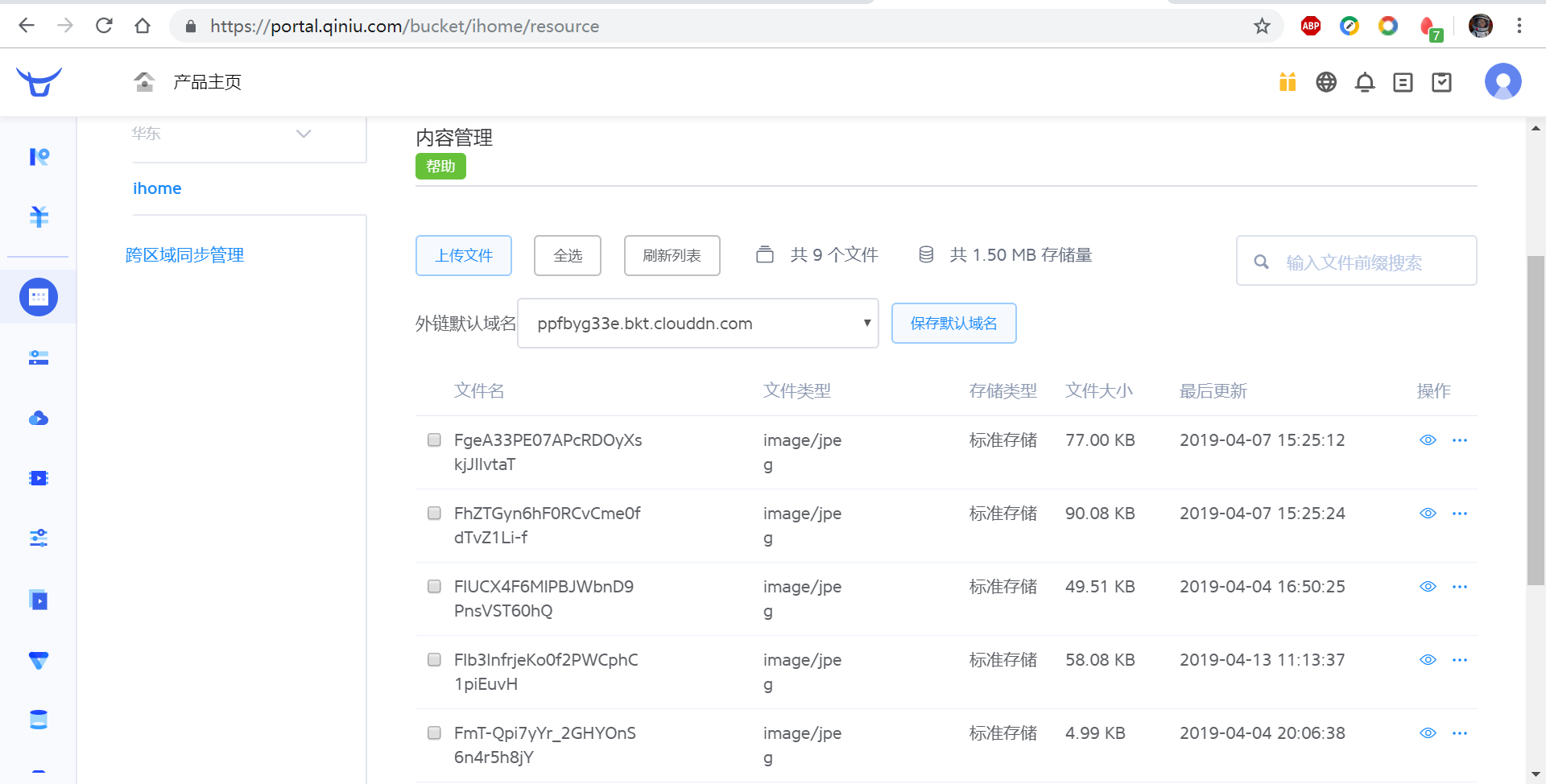


图6-2 七牛云服务图片管理

## 6.2基础信息模块实现

该部分主要是要把以上提出的功能进行实现，是用户可以在浏览器中使用本系统的各种功能。具体介绍如下。

### 6.2.1主页

用户打开浏览器，输入该系统网址，将进入该系统的主页，如用户未登录，则显示“注册”和“登录”按钮，如已登录，则显示用户昵称；还有主要房源图片幻灯片展示；最下面是“选择城区”和“入住日期”，可按需求进行搜索。如下图6-3所示

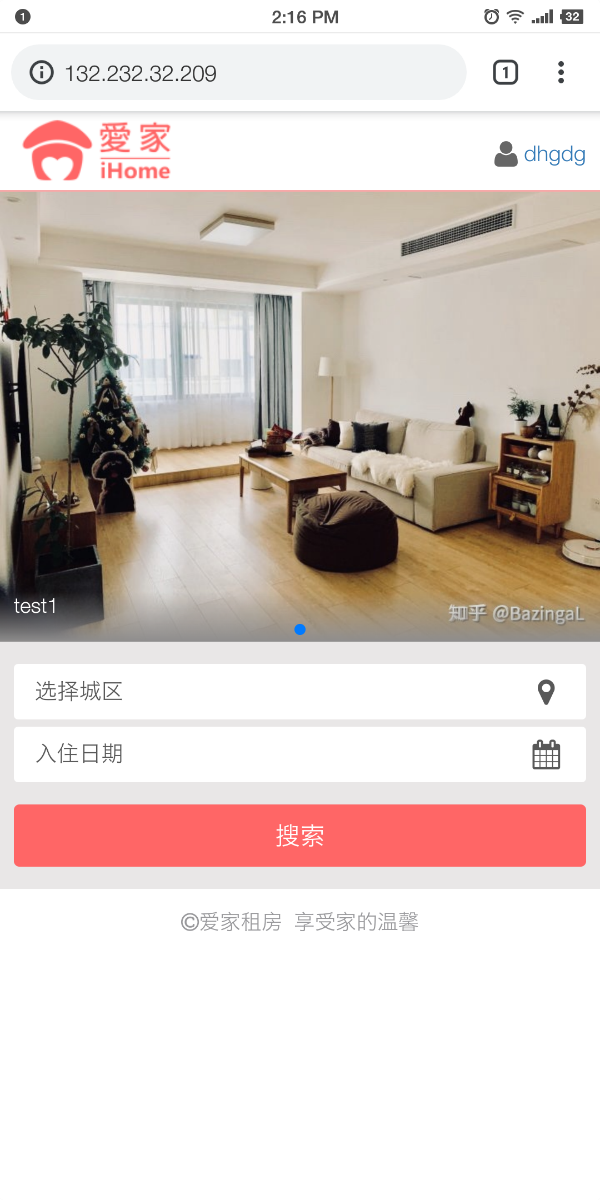
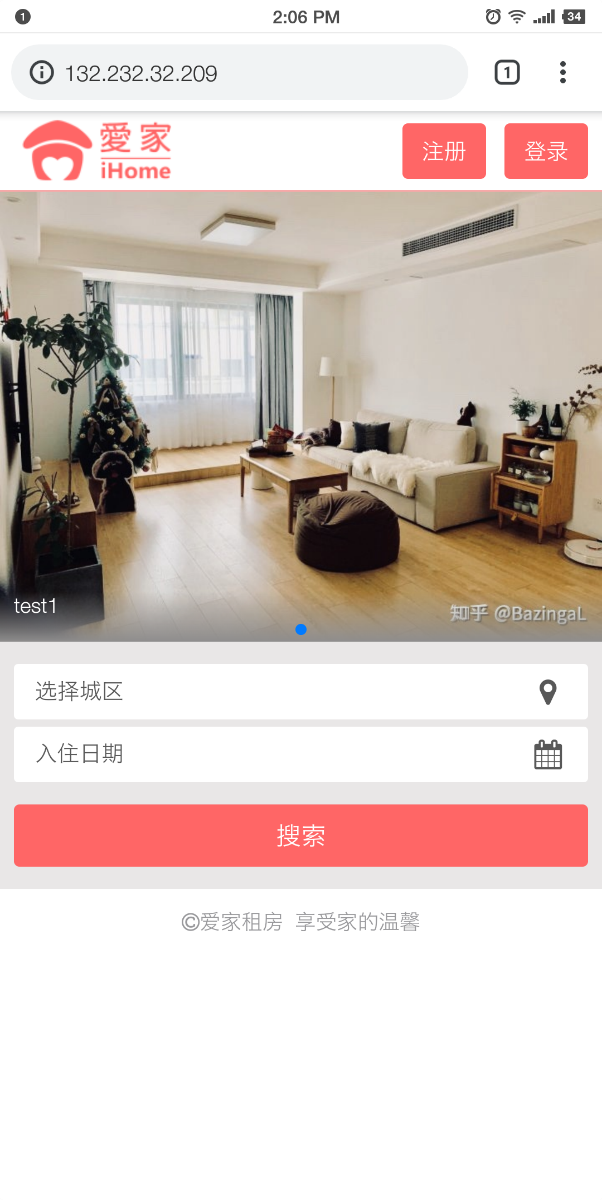


图6-3 项目主页

### 6.2.2注册页面

用户可以使用手机号码进行注册，填写手机号码和图形验证码，然后获取短息验证码，输入密码进行注册，两次密码必须输入一致，否则提示两次密码不一致;手机号码必须输入正确，而且要图形验证码输入正确，不然则提示图像验证码错误。图形验证码的有效期只有60s，并当获取短信验证码时便会失效，短信验证码的有效期也是有60s，之后便会失效。注册完成后自动跳会主页。具体情况如下图6-4~6-6所示：

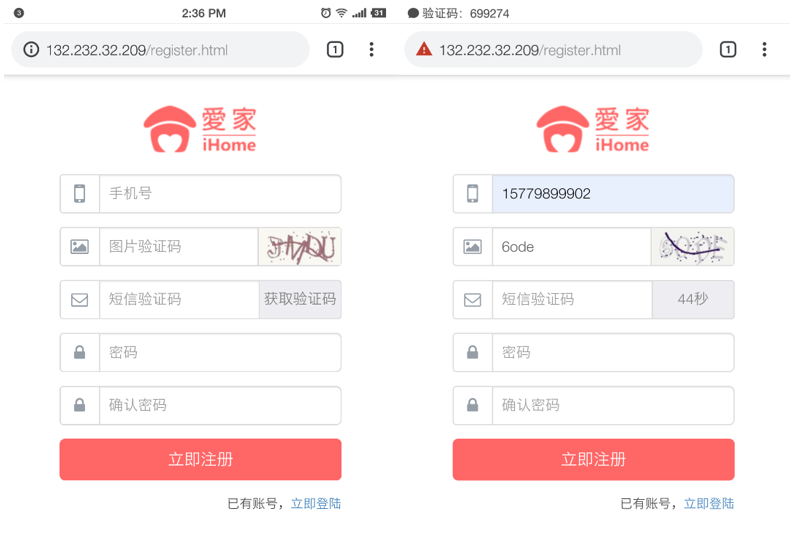


图6-4

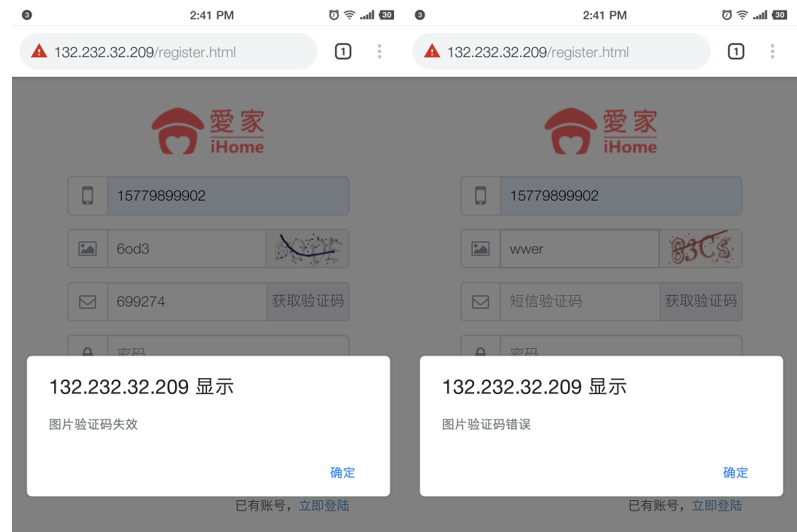


图6-5

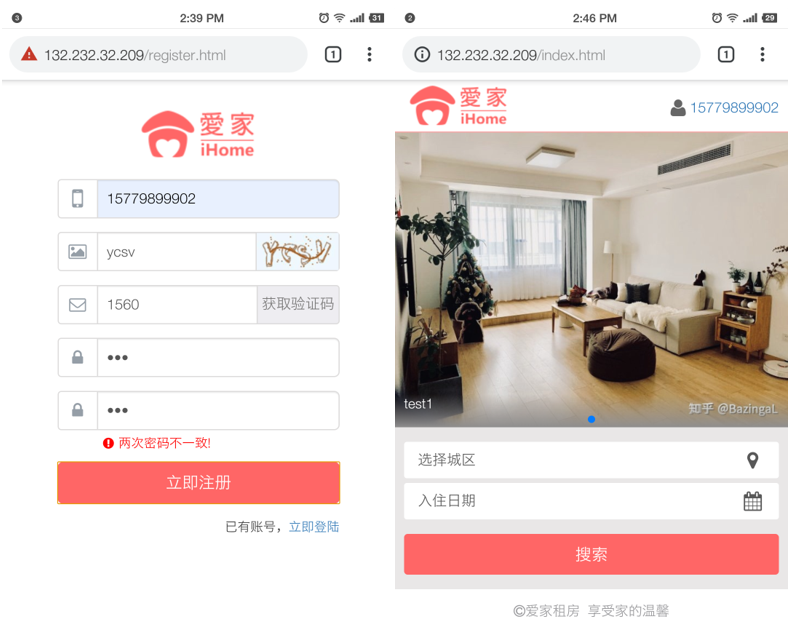


图6-6

### 6.2.3登录页面

使用手机号和密码进行登录，若账号或密码输入错误，则提示“用户名或密码错误”，登陆成功则跳转到首页。具体情况如图6-7所示：



图6-7 登录界面

### 6.2.4用户详情页面

用户登录完成之后，点击用户昵称，进入用户个人信息页面，该页面包括用户的个人信息模块、我的订单模块、实名认证模块、我的房源模块、客户订单模块和退出按键。主要界面如下图6-8所示：



图6-8 用户详情页面

1. 个人信息模块

用户点击个人信息的修改按键，可以跳转到个人信息页面，可以在此页面点击“选择文件”按钮选择本地图片，点击上传更换头像；也可以修改用户名，点击“保存”完成修改。详情如下图6-9所示：



图6-9 个人信息页面

1. 我的订单

房客选择完自己满意的房子后进行预定，之后可以在我的订单页面查看订单状态，若房东还未对订单进行处理，则订单状态为待接单，若房东已接单，则订单状态显示为待支付，用户可以点击“去支付”按钮，跳转到支付页面进行支付操作。若房东拒单，则订单状态为已拒单，告知拒单原因。具体情况如下图所示：



图6-10 待接单

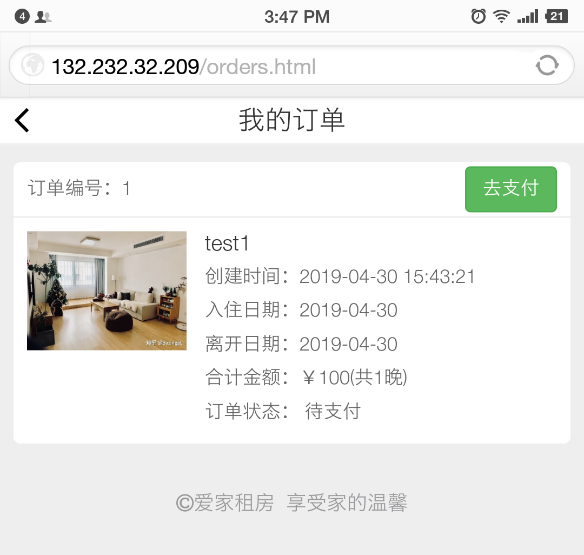


图6-11 已接单 图6-12 已拒单

房客点击“去支付”按钮后跳转到支付宝支付接口，完成支付后，点击回到我的订单，此使订单已经完成，可以发表评价。如下图



图6-13 订单完成

1. 实名认证

点击跳转到实名认证页面，填写真实姓名和身份证号进行实名认证，只有通过实名认证的用户才可以发布房源。界面如下图6-14所示：



图6-14 实名认证页面

1. 我的房源

点击跳转到我的房源页面，房东可以在此页面查看已经发布了的房源，也可以发布新的房源。点击已发布房源，可以查看房源详细信息，点击发布新房源，可以跳转到发布新房源页面，填写房屋详细信息，点击发布，跳转到上传图片页面，上传房屋图片（最多上传5张图片）。页面如下图6-15所示：

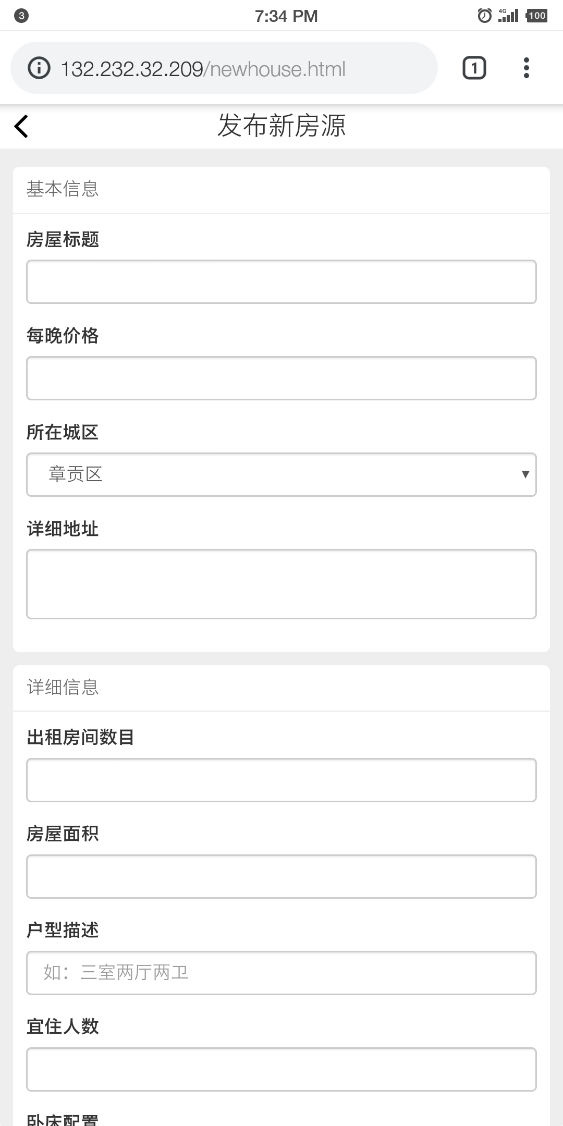


图6-15 我的房源页面

1. 客户订单

房东可以点击此模块跳转到客户订单页面，查看房客对自己房源的订购状态，有待接单、已完成、已拒单三种。拒单要填写拒单原因，对已完成订单可以查看用户评价。页面如下图6-16所示：



图6-16 客户订单页面

1. 登出

点击此模块可以退出当前账号，回到主页。

### 6.2.5用户搜索页面

用户可以点击首页的预览图片可以直接大道房屋详细信息页面；用户选择地区或入住时间或者同时选择两者或直接点击搜索按钮都可以进入房源列表页面，可以查看符合搜索条件的房源列表，点击其中一个房源可以进入房源写详细信息页面。页面如下图6-17所示



图6-17 房源搜索页面

## 6.3系统关键现技术实现

在系统开发过程中，一定不是一帆风顺的，会遇到一些问题。简单的可以利用百度或谷歌搜索很快得到解决，但一些比较复杂的问题想要解决就没那么容易。下面我们对其中几个复杂问题进行解释。

### 6.3.1图片验证控制

在用户进行登陆时，要输入图片验证码进行验证，但在测试这部分时发现，同一个图片验证码可以多次使用，按照正常的逻辑，即使用户验证失败了，也不能再次使用同一个图片验证码，应当点击图片更换验证码，重新进行填写。但是要在要在验证图片验证码和验证手机号之前进行删除，防止违法用户不停的请求图片验证码来撞库，获取数据库中已有的手机号。因此要在verify\_code.py中第76行添加以下代码，删除redis中的图片验证码，来防止用户使用同一个图片验证码多次。

try:

redis\_store.delete("image\_code\_%s" % image\_code\_id)

except Exception as e:

current\_app.logger.error(e)

### 6.3.2缓存数据同步问题

由于数据库与前端页面的速度不一致，因此为了提升访问速度，需要设置redis缓存，用户先从redis中获取数据，如果redis中有数据，则直接返回；若reids中没有，去MySQL中查询，然后将拿到的数据保存到redis中，将数据返回给前端。但若是在数据库中添加一个新的城区，此使MySQL中存在新的城区数据，redis中却没有，因此需要对MySQL和redis进行数据同步。下面有两种方式可以解决此问题：

1. 在操作MySQL时，删除redis缓存数据；
2. 给redis缓存数据设置有效期，保证过了有效期，缓存数据就会被删除。

第一种方法需要操作数据库的人足够细心，每一次操作时都要清除redis缓存，不太方便，因此我们采用第二种方式。

house.py中46行

[redis\_store.setex("area\_info",constants.AREA\_INFO\_REDIS\_CACHE\_EXPIRES, resp\_json)]

Constants.py中添加

[AREA\_INFO\_REDIS\_CACHE\_EXPIRES = 7200]

# 系统测试

## 7.1引言

系统测试一般是在项目投入生产运营之前，对整个项目与某些支持硬件、软件、数据和人员等其他系统元素及环境结合在一起进行测试，与系统的需求分析来作对比，从而发现其中那些和系统定义不匹配或矛盾的地方。这是为了发现软件种错误的必要步骤。

## 7.2测试计划

测试的最佳方法就是测试类中定义的方法，但是方法太多的话比较复杂，因此我们根据实际情况将测试方案设计为以下三个部分：

1. 单元测试部分： 在写完某个模块后进行，使用模块内部的变量进行测试，是独立的，与其他功能模块不相干；
2. 组装测试模块：将相关模块全部编写完整后进行综合测试，测试用例只涉及改组块的全部变量；
3. 确认测试模块：在所有模块编写完成后进行，是整个是系统的综合测试。

本项目测试的内容见表7-1，测试条件见表7-2.

表7-1 测试内容

|  |  |  |
| --- | --- | --- |
| 测试模块 | 功能点 | 测试类型 |
| 注册登录模块 | 显示符合性、系统测试性 | 界面测试、功能测试 |
| 房屋管理模块 | 显示符合性、系统测试性 | 界面测试、功能测试 |
| 订单管理模块 | 显示符合性、系统测试性 | 界面测试、功能测试 |

表7-2 测试条件

|  |  |  |
| --- | --- | --- |
| 名称 | 说明 | 数量 |
| 测试硬件配置 | Intel(R) Core(TM) i5 、高通660 | 2 |
| 操作系统 | Windows 10 、Android 7.1.1 | 2 |
| 浏览器 | Chrome 、Edge | 2 |

## 7.3测试项目

根据GB/T 17544-1998 《信息技术软件包质量要求和测试》国家标准，测试规则如表7-3所示：

表7-3 测试项目表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目软件 |  | | 程序版本 |  | |
| 功能模块 |  | | 编制人 |  | |
| 功能特性 |  | | | | |
| 测试目的 |  | | | | |
| 测试数据 |  | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 状态 |
|  |  |  |  |  |  |

### 7.3.1注册登录模块

该模块主要测试，用户注册时若是使用不合规范的手机号码或填写错误的图片验证码能否获取短信验证码；填写错误的短信验证码呢否注册成功；填写两次密码不一致能否注册成功。测试情况如表7-4；

表7-4 注册测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目软件 | 房屋出租租赁平台 | | 程序版本 | V1.0 | |
| 功能模块 | 注册 | | 编制人 | 程资源 | |
| 功能特性 | 验证用户注册 | | | | |
| 测试目的 | 测试用户注册的合理性 | | | | |
| 测试数据 | 手机号码、图形验证码、短信验证码、密码 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 状态 |
| 1 | 填写手机号或图片验证码 | 错误手机号或错误密码 | 无法获取短信验证码 | 提示图片信验证码错误或手机格式错误 | 正确 |
| 2 | 在步骤1正确的情况下填写短信验证码 | 填写错误的短信验证码 | 注册失败 | 提示短信验证码错误 | 正确 |
| 3 | 在步骤1、2正确的情况下填写两次密码 | 填写两次密码不一致 | 注册失败 | 提示两次密码不一致 | 正确 |
| 4 | 正确输入 | 正确格式的信息 | 注册成功 | 跳转到主页 | 正确 |

测试用户登陆不成功时会不会阻止登录并给出登录失败错误提示信息，测试情况如表7-5所示：

表7-5 登录测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目软件 | 房屋出租租赁平台 | | 程序版本 | V1.0 | |
| 功能模块 | 登录 | | 编制人 | 程资源 | |
| 功能特性 | 验证用户登录 | | | | |
| 测试目的 | 阻止错误登录 | | | | |
| 测试数据 | 手机号码、密码 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 状态 |
| 1 | 不输入数据 | 空值 | 登录失败 | 提示填写手机号 | 正确 |
| 2 | 只输入手机号 | 15711111111 | 登录失败 | 提示填写密码 | 正确 |
| 3 | 只输入密码 | 111 | 登录失败 | 提示填写手机号 | 正确 |
| 4 | 输错 | 错误的数据 | 登陆失败 | 提示用户名或密码错误 | 正确 |
| 5 | 正确输入 | 正确的账户 | 登录成功 | 进入系统 | 正确 |

### 7.3.2房屋管理模块

用户要想发布房源必须实名认证，测试用户在未实名认证的情况下能否发布房源，填写房源信息时如果没有完全填写能否发布房源，测试情况如表7-6：

表7-6 发布房源测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目软件 | 房屋出租租赁平台 | | 程序版本 | V1.0 | |
| 功能模块 | 发布房源 | | 编制人 | 程资源 | |
| 功能特性 | 房东发布房源 | | | | |
| 测试目的 | 测试发布房源的合理性 | | | | |
| 测试数据 | 实名认证信息、房屋配置信息 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 状态 |
| 1 | 不实名认证 | 空值 | 不能发布 | 提示填写实名信息 | 正确 |
| 2 | 不填或少填房源信息 | 房源配置信息 | 发布失败 | 提示填写房源配置信息 | 正确 |
| 3 | 填写完整信息 | 完整信息 | 发布成功 | 发布成功 | 正确 |

### 7.3.3订单管理模块

测试房客搜索房源时是否显示不在搜索范围内的房屋列表；预定房源时是否不经过房东接单就能支付。详细测试情况如表7-7所示：

表7-7 订单管理表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 项目软件 | 房屋出租租赁平台 | | 程序版本 | V1.0 | |
| 功能模块 | 发布房源 | | 编制人 | 程资源 | |
| 功能特性 | 房东预定房源 | | | | |
| 测试目的 | 测试预定房源的合理性 | | | | |
| 测试数据 | 搜索范围、房屋订单 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 状态 |
| 1 | 不填写范围 | 空值 | 全部订单 | 展示全部订单 | 正确 |
| 2 | 填写地区范围 | 地区 | 该地区房源 | 展示该地区全部房源 | 正确 |
| 3 | 填写入住时段 | 入住起始和结束时间 | 显示该时段没被预定的房源 | 展示该时段没被入住的房源 | 正确 |
| 4 | 填写地区和入住时段 | 地区、入住时段 | 符合条件的房源 | 展示该地区此时段还未入住的房源 | 正确 |
| 5 | 预定某个房源 | 预定房源 | 等待接单 | 订单状态为待接单 | 正确 |

## 7.4测试结论

经过多次测试，得到以下测试结论：

1. 系统的功能已满足需求分析所要求的内容，各个模块之间的联系符合逻辑设计；
2. 用户注册登录错误时被阻止跳转问题解决；
3. 前端页面与后端数据同步问题解决。

# 总结与展望

至此，该项目的设计和实现都已经全部完成了。本次项目的开发把之前学习的Python语言的知识做了实际的应用，小型租房平台可以为特定地区的租客解决租房困难的问题，提高了房东与租客之间的交易效率，同时也在本次项目开发的过程中学习了新的技术知识、巩固了之前所学，但毕竟该接触不久，可能其中有很多不足之处和值得优化的地方，希望读者可以进行批评指正。

本项目主要使用Python的flask框架实现，采用前后端分离的模式，方便后期的升级和维护；数据库使用MySQL，可以满足小型软件的部署利用，便且免费易用，数据库的建立主要使用Python语句，然后使用数据库迁移语句进行建立数据库，这样只要有代码，即使在不同的电脑上也能进行本地测试，而不用重新建立数据库；用户登录时采用图形验证+短信验证的方式，防止某些用户多次注册；前后端数据的同步之间添加redis缓存，来弥补前端和后端数据库之间速度不一致，提升访问速度。

如今房价居高不下，对于在外打拼奋斗年轻人来说，租房成为解决住房问题的首要选择，并且现在人们的生活与互联网之间的联系日益紧密，手机早已成为人们日常生活不可或缺的一部分，因此本项目可以使用户在手机上就能查看适合自己的房源，并与卖家完成交易，使租房变得十分便捷，一定会得到用户们的喜爱。由于我才疏学浅，本项目可能还有许多不足之处，还有很多功能可以实现，今后我会继续学习，来进一步的增加需要的功能，给用户更加便捷，更加舒适的体验。

# 参考文献

1. HTML & CSS设计与构建网站 [美] Jon Duckett / 刘涛、陈学敏 / 清华大学出版社 / 2013-1
2. 深入浅出HTML David Schultz、Craig Cook / 谢廷晟 / 人民邮电出版社 / 2008-5
3. Flask Web开发：基于Python的Web应用开发实战 [美] Miguel Grinberg / 安道 / 人民邮电出版社 / 2014-12
4. Flask Web开发实战 : 入门、进阶与原理解析 李辉 / 机械工业出版社 / 2018-8-1
5. Ubuntu Linux入门到精通 李蔚泽 / 机械工业出版社 / 2007-5
6. Ubuntu Hacks : Tips & Tools for Exploring, Using, and Tuning Linux Jonathan Oxer、Kyle Rankin、Bill Childers / O'Reilly Media / 2006-6-24
7. 高效前端：Web高效编程与优化实践 李银城 著 / 机械工业出版社 / 2018-3-15
8. 流畅的Python [巴西] Luciano Ramalho / 安道、吴珂 / 人民邮电出版社 / 2017-5-15
9. 从Python开始学编程 Vamei / 电子工业出版社 / 2016-11-24
10. 算法图解 [美] Aditya Bhargava / 袁国忠 / 人民邮电出版社 / 2017-3
11. Python Cookbook 中文版，第 3 版 David M. Beazley、Brian K. Jones / 陈舸 / 人民邮电出版社 / 2015-5-1
12. JSP程序开发范例宝典 明日科技 / 人民邮电出版社 / 2007-4
13. 高性能MySQL : 第3版 施瓦茨 (Baron Schwartz)、扎伊采夫 (Peter Zaitsev)、特卡琴科 (Vadim Tkachenko) / 宁海元、周振兴、彭立勋、翟卫祥,刘辉 / 电子工业出版社 / 2013-5-1
14. MySQL技术内幕 : SQL编程 姜承尧 / 机械工业出版社华章公司 / 2012-4-15
15. 精通CSS（第2版） : 高级Web标准解决方案 [英] Andy Budd、[英] Simon Collison、[英] Cameron Moll / 陈剑瓯 / 人民邮电出版社 / 2010-5

# 致谢

随着本次毕业设计的结束，大学四年的生活也将画上句号。心中充满着不舍之情，回想起这几个月做本次毕业设计过程中，遇到了许许多多的难以解决的问题和bug，使项目的开发进度一度停缓，但总是在此关键时刻，我的指导老师毛伊敏教授常常给予我很大的帮助，总是悉心教导我，鼓励我去找到解决问题的办法，在我们的共同努力之下，终于将项目开发中遇到的那些棘手问题得以解决，尽管并不轻松，却也从中学到了很多，我十分感谢老师在此次项目开发中对我技术上的帮助，教导我怎么去找到解决问题的方法。

同时，我也要对和我一同走过毕业设计这段时光的同学们。在这段日子里，我们一同学习，一同进步，正是因为你们的存在，才使得本该枯燥无味的事情变得生动有趣起来；感谢同学们在本次项目开发中对我的帮助，大家提出了许多对我十分有用的建议和想法，才使得我的这个项目能很好的完成。

在这里，我向那些曾经帮助过我的老师、同学和朋友们献上最为真挚的感谢和祝福。愿大家前途似海，来日方长。

# 附录

**主要源代码**

**项目启动文件manage.py**

from ihome import create\_app, db

from flask\_script import Manager

from flask\_migrate import Migrate, MigrateCommand

# 创建flask的应用对象

app = create\_app("develop")

app.debug = False

manager = Manager(app)

Migrate(app, db)

manager.add\_command("db", MigrateCommand)

if \_\_name\_\_ == '\_\_main\_\_':

manager.run()

**连接数据库config.py**

import redis

class Config(object):

"""配置信息"""

DEBUG = True

SECRET\_KEY = "XHSOI\*Y9dfs9cshd9"

# 数据库

SQLALCHEMY\_DATABASE\_URI = "mysql+pymysql://root:czy

@127.0.0.1:3306

/ihome"

SQLALCHEMY\_TRACK\_MODIFICAIONS = True

# redis

REDIS\_HOST = "127.0.0.1"

REDIS\_PORT = 6379

# flask\_session配置

SESSION\_TYPE = "redis"

SESSION\_REDIS = redis.StrictRedis(host=REDIS\_HOST, port=REDIS\_PORT)

SECRET\_USE\_SIGNER = True # 对cookie中的session\_id进行隐藏处理

PERMANENT\_SESSION\_LIFETIME = 86400 # session数据的有效期，单位：秒

class DevelopmentConfig(Config):

"""开发模式的配置信息"""

DEBUG = True

class ProductionConfig(Config):

"""生产环境配置信息"""

pass

config\_map = {

"develop": DevelopmentConfig,

"product": ProductionConfig

}