

桂林航天工业学院

课程设计存档材料

课程名称: 软件工程课程设计

开课学期: 2019—2020 学年第一学期

专业: 软件工程

班级: 2017 软件工程 4 班

姓名: 钟祯、李文洁、何嘉毅、陈荣煜、黄钰淞

学号: 2017070030429、2017070030412

2017070030411、2017070030423

2017070030434

指导老师: 何丽华、邓建华

报告日期: 2019 年 12 月 30 日

桂林航天工业学院课程设计任务书

设计题目： 房屋租赁信息管理系统

学生姓名	钟祯、何嘉毅、李文洁、陈荣煜、黄钰淞		
课程名称	软件工程课程设计	专业班级	2017 软件工程 4 班
地 点	巡天楼 308、309	起止时间	2019 年 12 月 23 日 至 2019 年 12 月 27 日
设计内容	1. 针对一个需要开发的软件系统按照软件工程中的生命周期进行完整的项目分析，完成系统的概述，进行系统的可行性分析，系统的需求分析，系统的总体设计，系统的详细设计，系统交互界面的实现。 2. 利用 Microsoft Visio 和 ProcessOn 软件作为项目系统的设计建模工具，运用前端 Vue 框架以及 Web 框架 Express 设计系统交互界面。 3. 按照课程设计报告的标准范文完成报告的撰写，要求报告内容涵盖系统概述、可行性分析、需求分析、系统总体设计、系统详细设计、系统交互界面实现、结语等组成部分。		
设计参数	无		
设计进度	第 1 天，选定课程设计题目，做好相关设计准备，完成系统概述的撰写 第 2 天，针对项目进行可行性分析，并进行需求分析，确定最终需求。 第 3 天，针对项目的需求结果进行总体设计，同时完成详细设计。 第 4 天，针对详细设计结果进行系统实现，完成交互界面的设计与开发。 第 5 天，完善设计内容，最后进行总结，并完成课程设计报告。		
设计成果	1. 完成系统的最终需求 2. 实现系统的主要功能 3. 经过小组成员测试，功能不断完善。		
参考资料	[1] 王立峰. 软件工程理论与实践 [M]. 清华大学出版社, 2003. [2] 吕云翔, 王洋, 肖咚. 软件测试案例教程 [M]. 2011. [3] 吴洁明, 袁山龙. 软件工程应用实践教程 [M]. 2003. [4] 李彤, 王炜, 郁湧. 软件工程概论 [M]. 科学出版社, 2012.		
说明	1. 本表应在每次实施前由指导教师填写一式 2 份，审批后所在教学单位和指导教师各留 1 份。		

桂林航天工业学院本科课程设计报告

- | | |
|--|--|
| | <ul style="list-style-type: none">2. 多名学生共用一题的，在设计内容、参数、要求等方面应有所区别。3. 若填写内容较多可另纸附后。 |
|--|--|

指导教师：刘建华、何丽华
教研室主任：张新伦

2019年12月30日



本科课程设计报告

开课学期： 2019—2020 学年第一学期

课程名称： 软件工程课程设计

题 目： 房屋租赁信息管理系统

院 系： 计算机科学与工程学院

专业班级： 2017 软件工程 4 班

学 号： 2017070030429、2017070030412
2017070030411、2017070030423
2017070030434

学生姓名： 钟祯、李文洁、何嘉毅、陈荣煌、黄钰淞

指导教师： 何丽华、邓建华

完成日期： 2019 年 12 月 30 日

成绩： （五级）

桂林航天工业学院本科课程设计报告

课程设计评分标准

平时 (30%)	全勤、学习态度端正、实验认真、积极回答问题	优秀 (90-100)	
	偶有缺勤、实验认真、回答问题较积极	良好 (80-89)	
	旷课 2 次以内、偶有迟到、实验认真、回答问题较好	中 (70-79)	
	旷课 2 次以上、学习态度一般、基本能回答出问题	及格 (60-69)	
	经常旷课，实验过程不认真、问题回答不积极	不及格 (<60)	
考核 (70%)	作品功能完全实现、作品演示操作熟练、问题回答准确；设计论文结构合理、内容符合要求、论文格式符合规范要求、参考文献新	优秀 (90-100)	
	作品功能基本完全实现、作品演示操作熟练、问题回答准确；设计论文结构合理、内容符合要求、论文格式符合规范要求、参考文献新	良好 (80-89)	
	作品功能实现 70%、作品演示操作基较熟练、问题回答基本准确；设计论文结构基本合理、内容基本符合要求、论文格式基本符合规范要求、参考文献较新	中 (70-79)	
	作品功能实现 50%、作品演示操作较熟练、部分问题回答准确；设计论文结构较合理、论文格式基本符合规范要求、有参考文献	及格 (60-69)	
	作品功能不能实现、作品演示操作不熟练、问题回答效果差；设计论文结构不合理、论文格式不符合规范要求、无参考文献	不及格 (<60)	
总评			

目 录

第 1 章 绪论.....	1
1. 1 背景及意义.....	1
1. 2 国内外研究现状.....	1
第 2 章 网站后台管理系统分析.....	2
2. 1 网站后台管理系统的定义.....	2
2. 2 网站后台管理系统的应用.....	2
第 3 章 系统可行性分析.....	3
3. 1 技术可行性.....	3
3. 2 经济可行性.....	3
3. 3 操作可行性.....	4
3. 4 社会因素可行性分析.....	4
第 4 章 需求分析.....	5
4. 1 系统目标.....	5
4. 2 系统功能模块图.....	5
4. 3 系统业务流程图及分析.....	5
4. 3. 1 顶级业务流程图.....	6
4. 3. 2 二级业务流程图-房源管理.....	7
4. 3. 3 二级业务流程图-房东管理.....	7
4. 3. 4 二级业务流程图-合同管理.....	8
4. 3. 5 二级业务流程图-租金管理.....	8
4. 3. 6 二级业务流程图-出租管理.....	8
4. 4 数据流图及分析.....	9
4. 4. 1 顶层数据流图.....	9
4. 4. 2 0 层数据流图.....	10
4. 4. 3 1 层数据流图-管理员模块.....	10
4. 4. 4 1 层数据流图-房东模块.....	11
4. 5 房源管理时序图.....	11
4. 6 系统用例图.....	12
第 5 章 系统总体设计.....	13
5. 1 系统功能设计目标.....	13
5. 2 房屋租赁信息管理系统功能概述.....	14
5. 3 系统总体功能的层次划分.....	15
第 6 章 数据库设计.....	16
6. 1 数据库 E-R 图.....	16
6. 2 数据字典.....	16
6. 3 数据库字段表.....	17
第 7 章 系统详细设计.....	18
7. 1 房源出租流程图.....	18
7. 2 房东审核功能流程图.....	18
7. 3 房东信息修改流程图.....	19

7.4 添加用户信息操作流程图.....	20
7.5 房东删除操作流程图.....	21
7.6 房源租金增加操作流程图.....	22
第8章 系统实现.....	23
8.1 系统实现说明.....	23
8.2 用户登录界面.....	23
8.3 登录成功进入管理系统首页.....	23
8.4 房客申请租房记录.....	24
8.5 数据报表.....	24
8.6 用户（管理员、房东）列表界面.....	24
8.7 用户添加界面.....	25
8.8 房源添加界面.....	25
8.9 系统默认设置界面.....	26
课程设计工作总结 - 黄钰淞.....	27
1. 工作概述.....	27
2. 系统测试过程.....	27
3. 体会.....	28
课程设计工作总结 - 钟 禱.....	29
1. 工作概述.....	29
2. 系统开发过程.....	29
3. 体会.....	31
课程设计工作总结 - 陈荣煜.....	32
1. 工作概述.....	32
2. 系统分析过程.....	33
3. 体会.....	33
课程设计工作总结 - 何嘉毅.....	34
1. 工作概述.....	34
2. 数据库设计过程.....	34
3. 体会.....	35
课程设计工作总结 - 李文洁.....	35
1. 工作概述.....	35
2. 图的编写绘画过程.....	35
3. 体会.....	36
结语	36
参考文献.....	37
附录 A	38
附录 B	41
附录 C	42
附录 D	44
附录 E	45
附录 F	46

第1章 绪论

1.1 背景及意义

随着科学技术的不断发展，计算机科学技术日渐成熟，不同的芯片，算法，软件，硬件喷薄而出，计算机的性能越来越完善越来越成熟，为企业或者个人的工作开发做出了杰出的贡献。作为计算机的功能之一，使用计算机对各种信息进行管理，具有手工管理所无法比拟的优点，对于大部分人来说，通过计算机开发出来的软件，功能，不断在人类社会扩散。其中最主要的就是检索迅速，查找方便，可靠性高，存储量大，保密性好，运行周期长，成本低等。

同时随着我国的经济发展，改革的深入使得企业的生产任务日益繁重，从而对房源管理的要求也更加严格。时代在不断进步，信息在不断扩大，工作不断繁杂。我们需要好的计算机设计系统及软件来增加我们对信息数据的分类及管理，减轻企业或者政府管理部门的工作积压，加快了企业的资金周转速度，避免了人工的失误率。由此可见，房屋租赁管理系统是人们不可缺少的一个设计系统。由于对房屋资源的详细分类，适用人群收益大，因此对房屋的信息进行具体管理(价格，类型，地点，房东信息，租户信息，出/到租时间，房源号等)大大满足了需求。因此房屋租赁的信息管理必须编制一套库存管理信息系统，实现计算机化操作。

1.2 国内外研究现状

在西方发达国家，类似的管理信息系统已经成熟，尤其是基于 web 应用技术环境下的不动产管理信息系统在房产行业中的应用日趋完善，其运作模式已经从连锁店的线性管理转变为跨地区，跨领域的扁平化管理，从而获得了显著的经济和社会效益。在我国，随着国民生活水平的不断提高，房产行业的发展和居民租房需求的多元化带动了房产公司的发展，虽然存在着比较多的后台管理网站，但房产的信息化管理还不够成熟，不够完善，不能对提高效率、降低成本、共享资源等方面起到至关重要的作用。在我国，房产行业的管理仍然限于传统的方式，店面彼此都是信息孤立，难以为公众提供及时、互动的信息，阻碍了房产交易市场的健康、快速发展。未来，房屋信息管理系统将朝着信息化之路前进，从而实现房产交易中共享信息的功能，增加交易的机会，缩短交易的时间。

第 2 章 网站后台管理系统分析

2.1 网站后台管理系统的定义

网站后台管理系统主要是用于对网站前台的信息管理，如文字、图片、影音、和其他日常使用文件的发布、更新、删除等操作，同时也包括会员信息、订单信息、访客信息的统计和管理。简单来说就是对网站数据库和文件的快速操作和管理系统，以使得前台内容能够得到及时更新和调整。

对于企业网站而言，适合企业网站的网站后台管理系统应该具备的特点如下：

1. 企业网站一般作为形象名片而存在，那么企业网站必须美观精致而有气质；
2. 企业网站能够很一目了然的把企业产品，性质给表达出来；
3. 企业网站应该有产品展示功能，需要交易的还需要有在线订单的电子商务功能。

在网站后台管理系统中，有些功能比较简单，有些功能比较复杂，还有一些如同自助建站系统，当然自助建站与网站管理系统的异同还是有一些的。在这些系统中其实并不是功能复杂的就很好，也不是功能比较简单的就很差。因为每一套网站后台管理系统的出现就有他存在的价值，不同的行业，不同的需求，就有不同的市场。功能比较复杂的对于维护不方便，而且有些功能完全用不上；而功能比较简单的对于后期维护和升级方面就比较方便。其实无论什么样的网站后台管理系统只要做到在灵活性，维护性，方便操作的解决方案才能引起客户的满意，只有客户满意的网站后台管理系统才是最优秀的系统。

2.2 网站后台管理系统的应用

鉴于网站后台管理系统功能的强大且方便，其几乎用于所有的动态网站系统中，如asp、jsp构建的网站中。

根据不同的需求，网站后台管理系统有几种不同的分类方法。比如，根据应用层面的不同，可以被划分为：

1. 重视后台管理的网站后台管理系统；
2. 重视风格设计的网站后台管理系统；
3. 重视前台发布的网站后台管理系统；

在网络上比较流行的各种网站后台管理系统，风格千差万别，但都具有很好的灵活性，方便灵活变动。

网站后台管理系统开发者的出发点是为了让不熟悉网站的用户有一个直观的表示方法，也让各种网络编程语言用户可以通过简单的方式来开发个性化网站。让不同的用户在懂得少许HTML(标准通用标记语言下的一个应用)的基础上，就能够快速的构建一个风格个性化的而功能强大的专业企业网站。

第3章 系统可行性分析

3.1 技术可行性

本系统采用 JavaScript 作为开发语言，JavaScript（简称“JS”）是一种具有函数优先的轻量级，解释型或即时编译型的编程语言。虽然它是作为开发 Web 页面的脚本语言而出名的，但是它也被用到了很多非浏览器环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式和声明式（如函数式编程）风格。

本系统采用前后端分离的方式实现，前端使用 Vue 框架，后端使用基于 JS 的服务端 Node，Express 框架编写，数据库使用 MongoDB。编写的程序具有结构清晰、易于扩充等优良特性，适合于各种应用软件、系统软件的程序设计。再结合数据库技术，实现强大的查询、修改、录入、删除操作。此外，从开发人员的水平考虑，作为计算机科学与技术专业大三的学生，大学三年所积累的专业知识使我们从一定程度上具备了开发一个小型系统的能力。

3.2 经济可行性

由于本系统所要求的硬件（计算机及相关硬件）和软件环境（Visual Studio Code 及 MongoDB），市场上都容易购买到或从相关网站下载，系统成本主要集中在本管理系统的开发与维护上，对用户不造成过重的经济负担。而一旦此系统投入使用，不仅可以减少工作人员的工作强度，提高工作效率，而且方便了对房源信息的管理，延续了传统的管理模式，极大限度地方便了管理人员，而且，房屋信息管理系统是一个小型的管理系统，在开发时目标系统不是很复杂，开发的周期较短，人员经济支出不大，使得房屋管理工作变繁为简，使其更加合理化、规范化、工作效率的提高会给企业带来整体水平的提高，所节省的大量人力、财力

产生的效益将远远大于本系统的开发成本，所以从经济方面讲开发此系统是可行的，是非常值得投资的。

3.3 操作可行性

本系统采用Visual Studio Code作为该系统的开发环境，使用基于Vue的UI组件库Element-UI设计友好的界面，使其美观，操作简单。采用MongoDB工具建立数据库。因为管理人员已经有丰富的信息管理经验，通过本系统的设计人员与之的交流沟通，可对本系统的功能应用作进一步完善以满足企业的需求。

3.4 社会因素可行性分析

该系统不违法任何合同法、专利、版权、合同、法律法规方面的问题，具有法律方面可行性；该系统主要用于房屋租赁管理方面的使用，具有使用方面的可行性。本系统的功能只会给人们带来社会效益，不会侵犯人们的安全、隐私、经济等权益。因此该系统具有社会方面的可行性。

第4章 需求分析

4.1 系统目标

采用现有的软件，硬件环境以及合理可行的系统开发方案，建立一个界面友好，操作简单，结构完善，布局规划合理的房屋租赁管理系统。

4.2 系统功能模块图

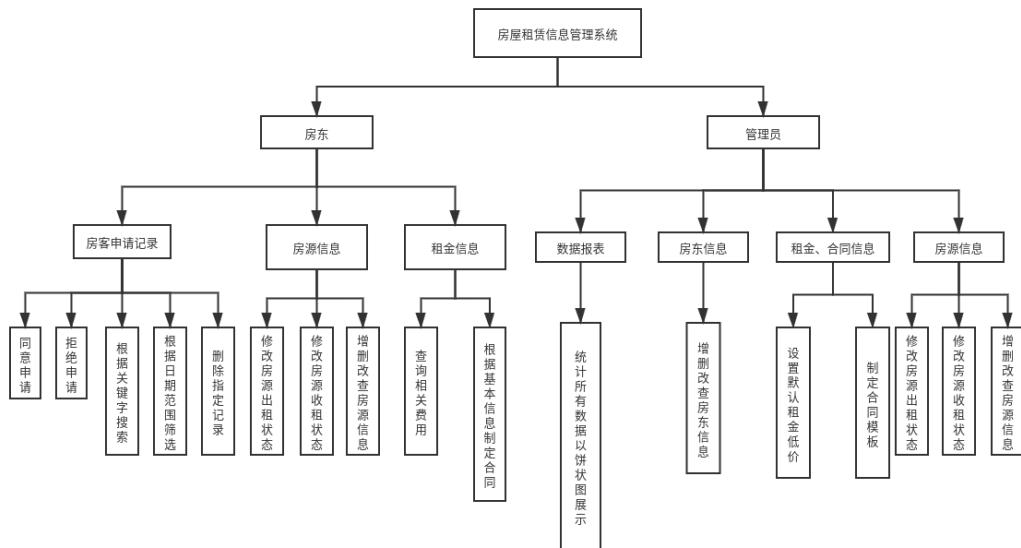


图 4.1 系统功能模块图

4.3 系统业务流程图及分析

整个系统从整体上分为以下四个大部分：

1. 房屋信息管理：实现房屋基本资料的管理；
2. 用户资料管理：实现房屋出租者和房屋租赁者基本资料（合同）的管理；
3. 租赁管理：实现用户对现有房屋情况的查询、预定；
4. 数据报表：实现对现有的数据进行图表可视化。

4.3.1 顶级业务流程图

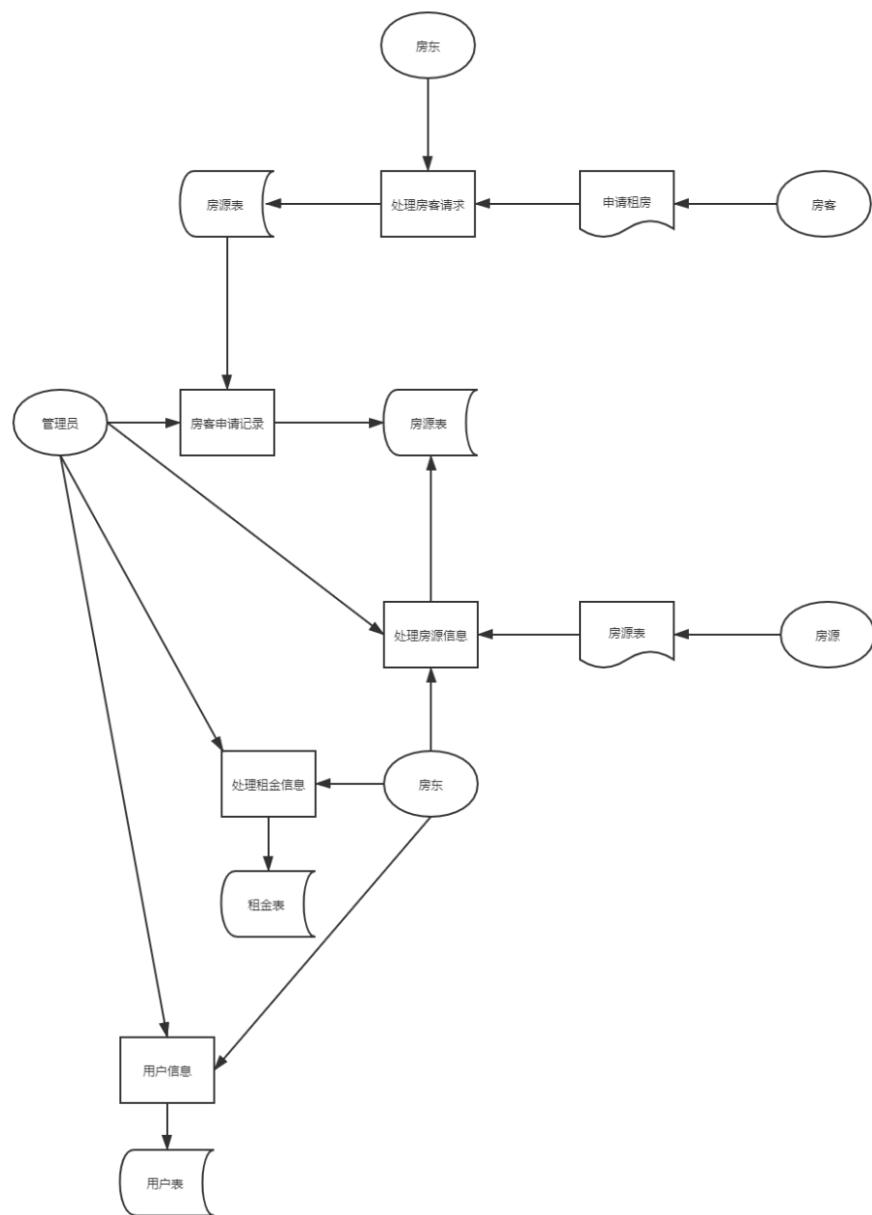


图 4.2 顶级业务流程图

4.3.2 二级业务流程图-房源管理

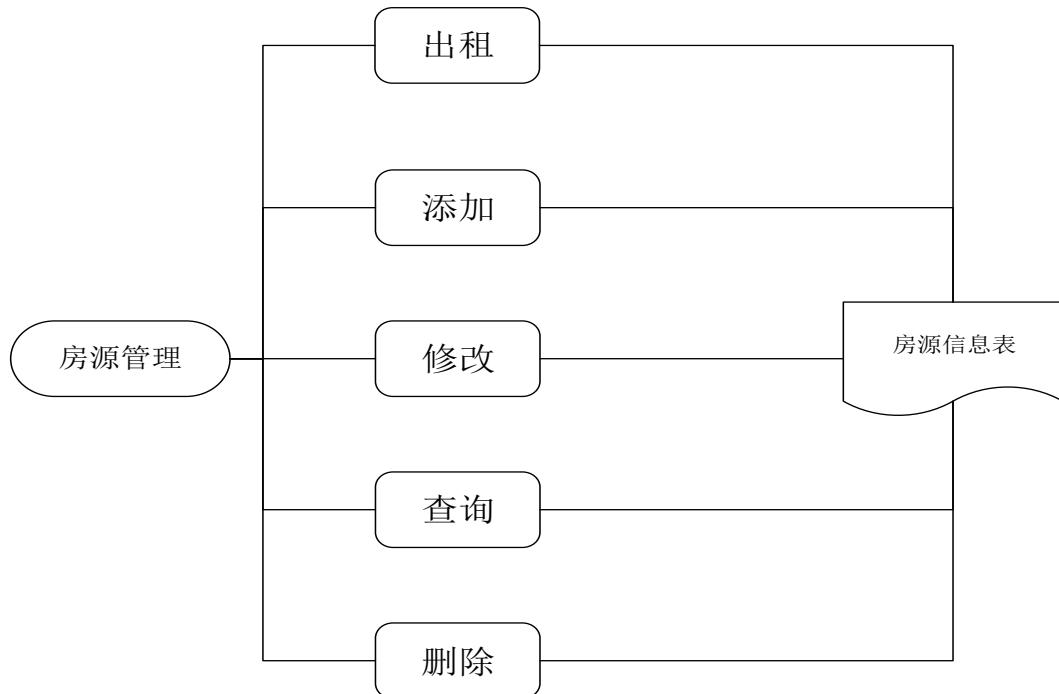


图 4.3 二级业务流程图-房源管理

4.3.3 二级业务流程图-房东管理

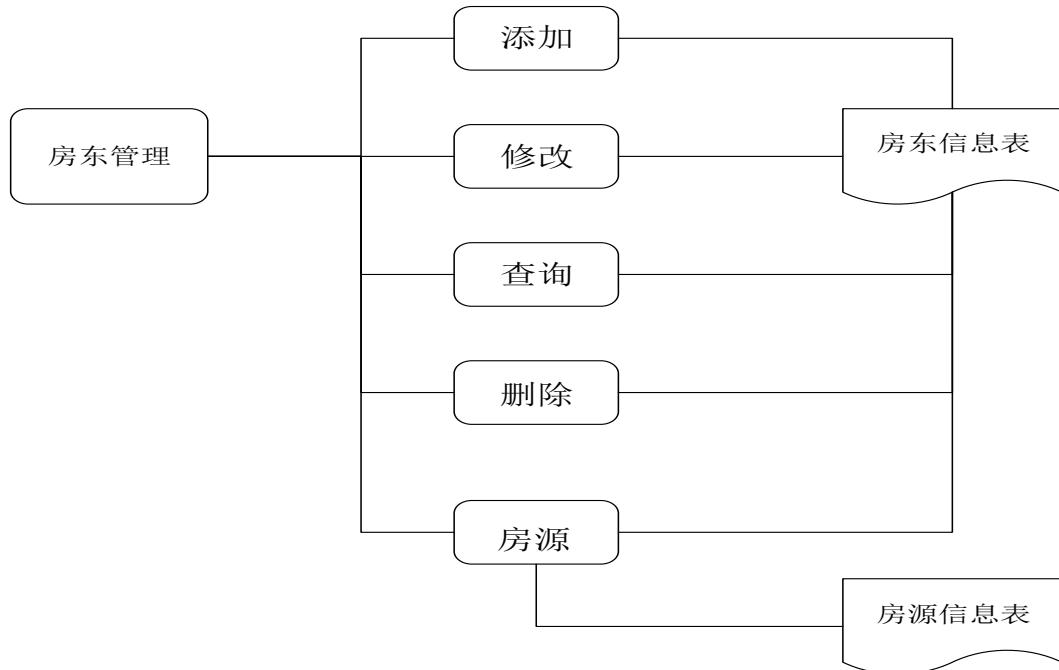


图 4.4 二级业务流程图-房东管理

4.3.4 二级业务流程图-合同管理

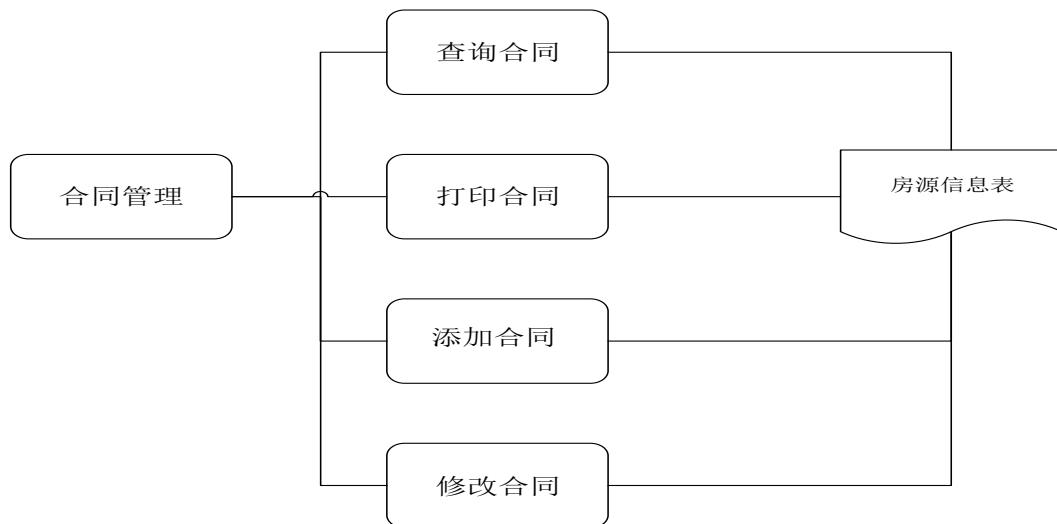


图 4.5 二级业务流程图-合同管理

4.3.5 二级业务流程图-租金管理

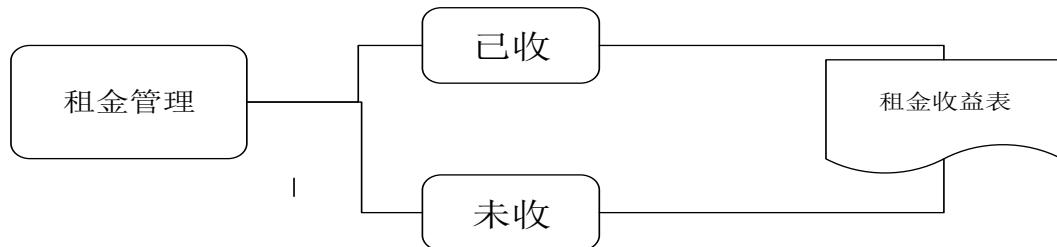


图 4.6 二级业务流程图-租金管理

4.3.6 二级业务流程图-出租管理

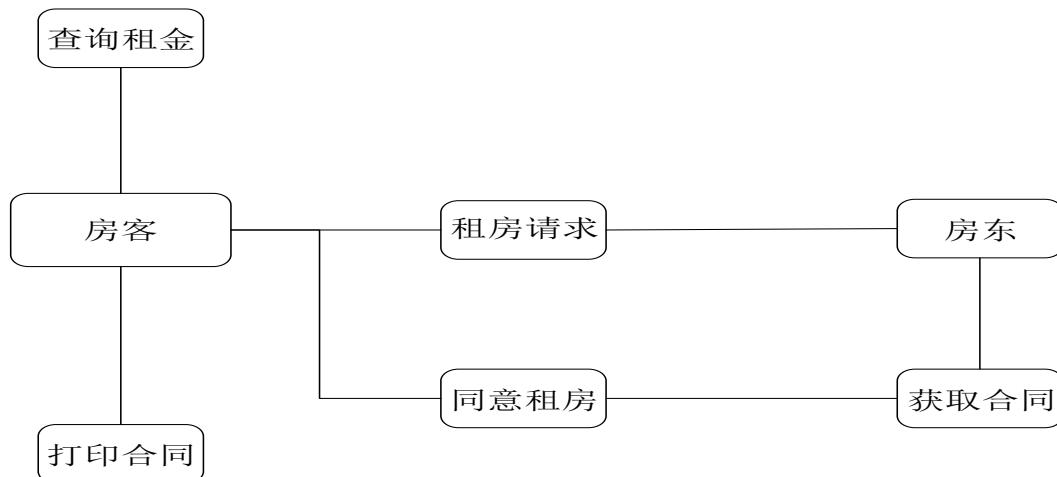


图 4.7 二级业务流程图-出租管理

4.4 数据流图及分析

经过对用于需求的考察导出了初次的数据流图, 用户可以根据数据流图考察系统是否完成了他们基本的需要。系统模块主要包括房源管理模块, 房东管理模块, 租金管理模块, 房客申请信息管理模块。

4.4.1 顶层数据流图

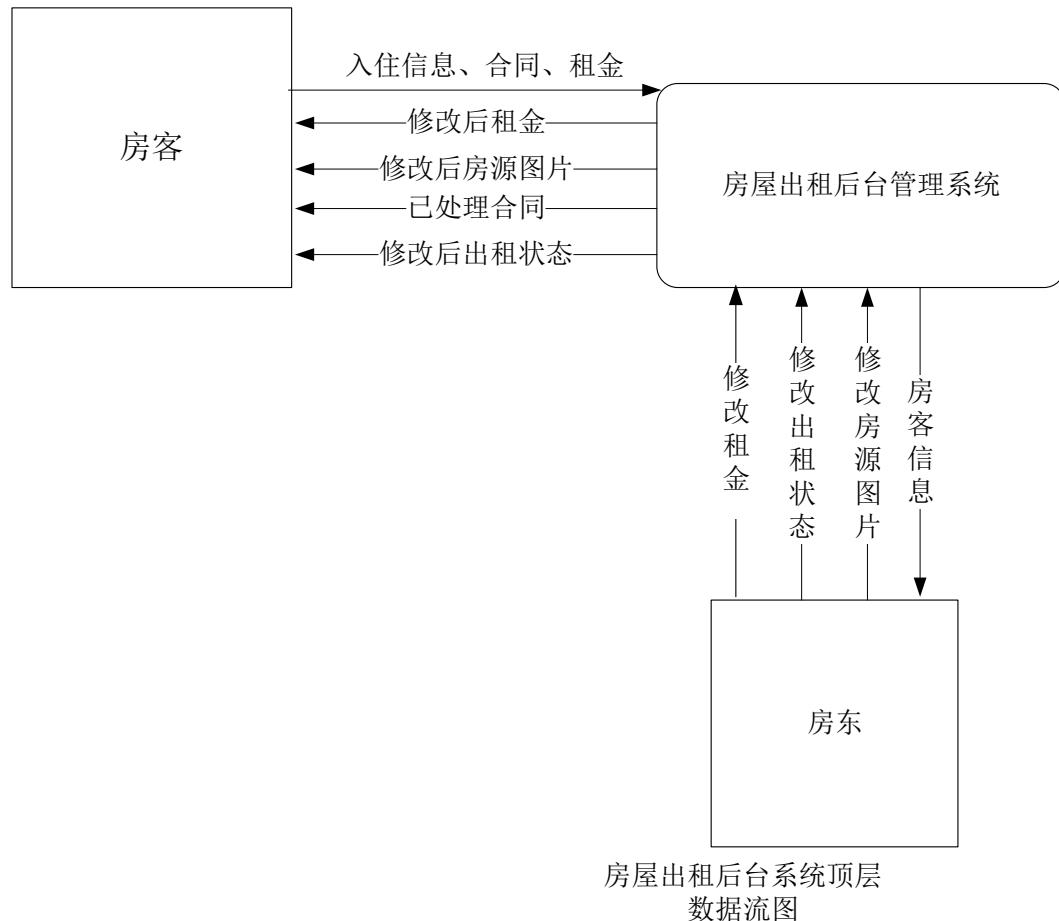


图 4.8 顶级数据流图

4.4.2 0层数据流图

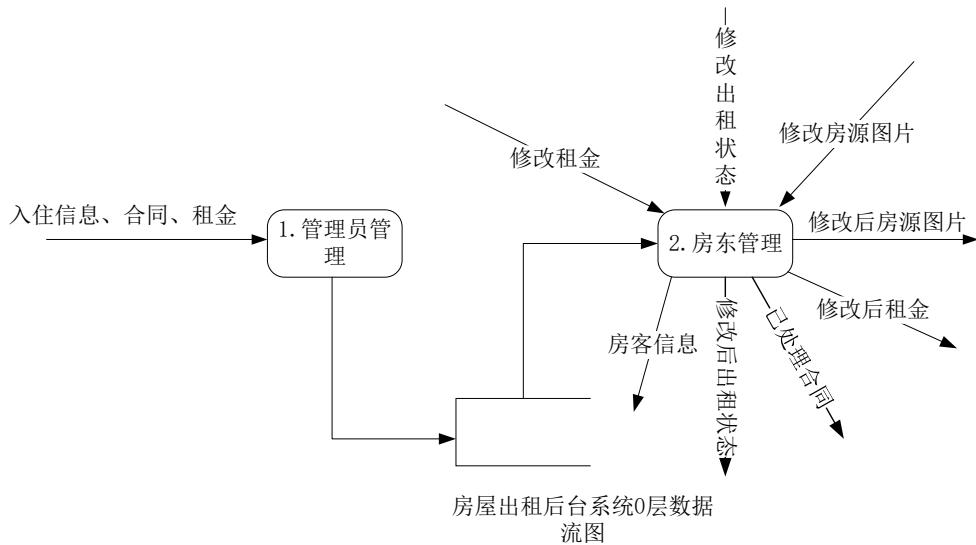


图 4.9 0 层数据流图

4.4.3 1层数据流图-管理员模块

管理员模块可以对系统内一切数据进行操作。其数据流程图如下图：

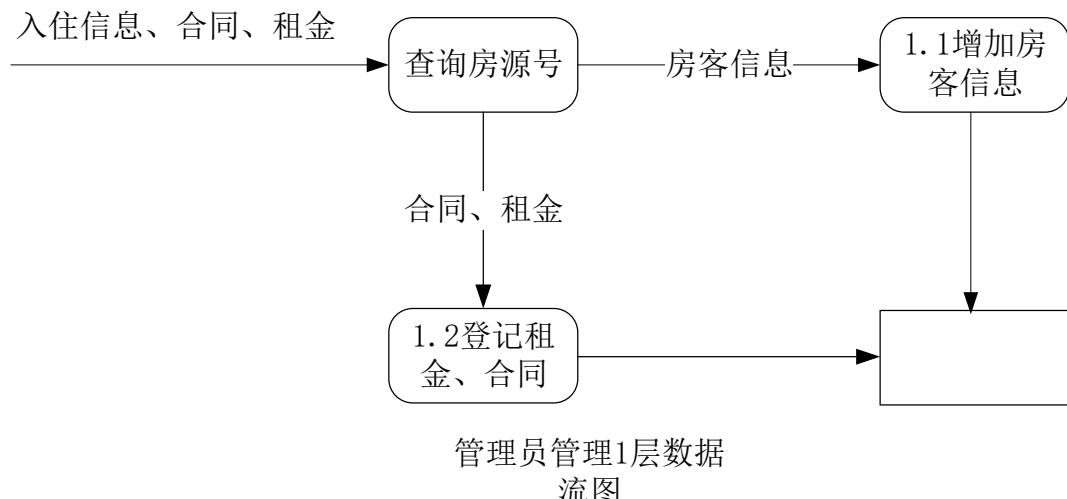


图 4.10 1 层数据流图-管理员模块

4.4.4 1 层数据流图-房东模块

房东模块可以对租金,房源,租房情况进行修改查询操作。其数据流程图如下图:

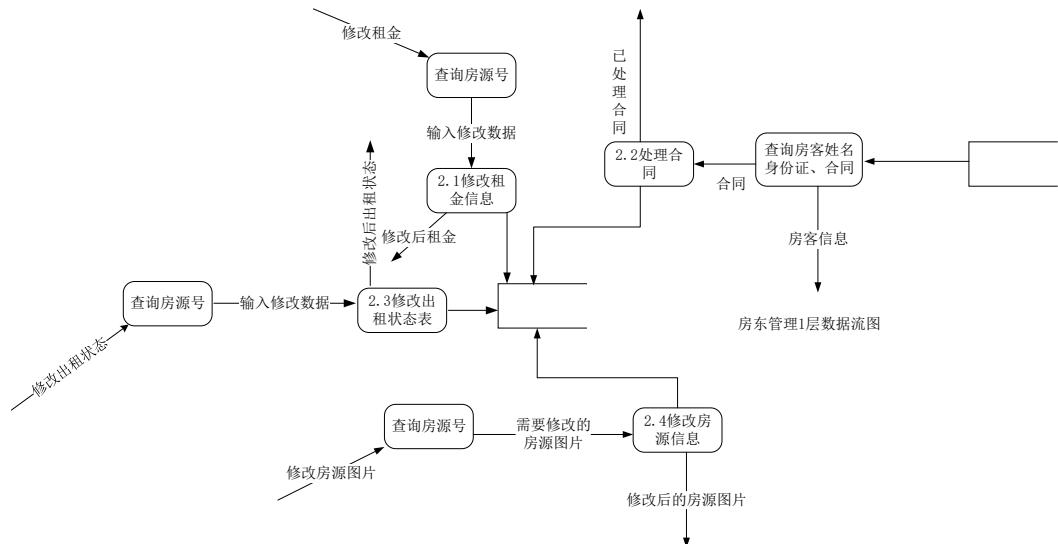


图 4.11 1 层数据流图-房东模块

4.5 房源管理时序图

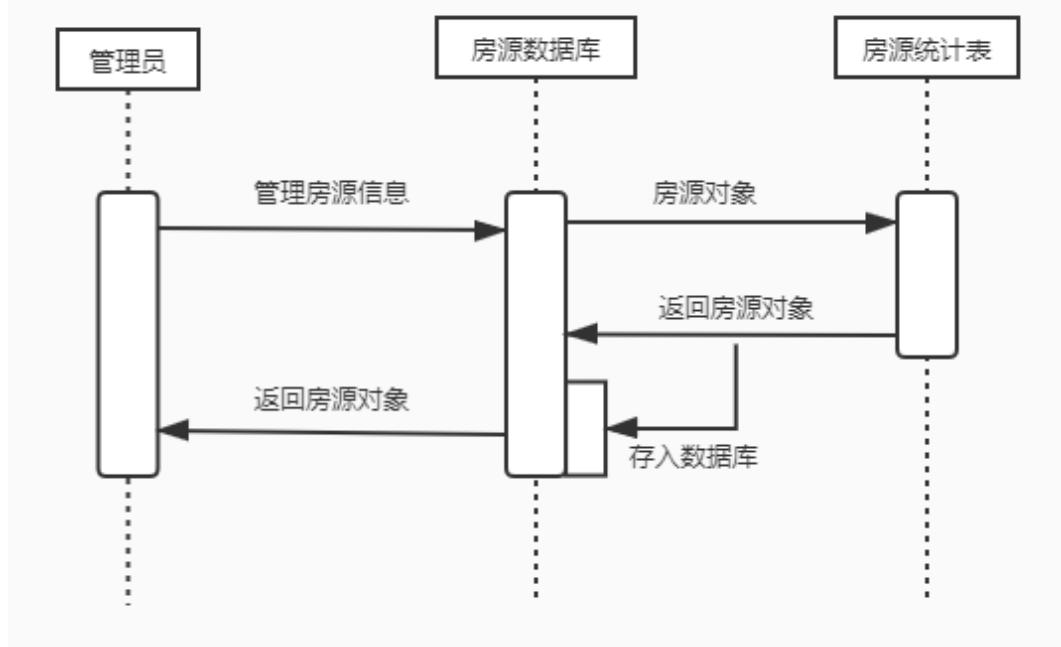


图 4.12 房源管理时序图

4.6 系统用例图

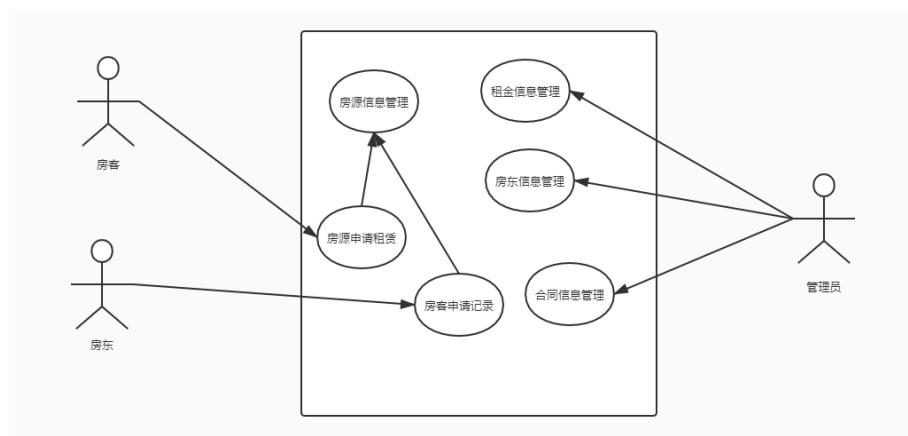


图 4.13 系统用例图

第 5 章 系统总体设计

5.1 系统功能设计目标

通过对系统的需求分析，对本房屋租赁管理系统，提出了以下设计目标：

1. 系统安全性：对于不同的用户系统应该赋予不同的权限，每一个用户都不能越权，从而使得数据的安全性得到了应有的保证。系统的要具备较高的可靠性，以及系统的可恢复性和数据的准确性。
2. 可维护性：客户会在系统使用过程中不断对系统提出新要求，扩展系统功能，这就要求系统的可升级性必须良好，以满足客户长期实用的要求，并且在用户有新要求的时候可以快速扩展功能。
3. 用户操作：本系统的设计应该更加人性化，从而使系统具有友好的人机交互能力。对于用户对信息进行的查询、删除、新增等操作能够及时做出准确的反映，同时要保证系统的响应时间应在一定范围内。
4. 运行速度：衡量系统性能好坏的一个重要参考依据就是系统的响应时间，一个好的系统的响应时间要短，数据转换时间也要短，后台的服务器响应要迅速。由于系统使用者的特殊性，系统必须高效地响应并且在安全的前提下高效的响应。
5. 界面：友好的用户界面，业务操作简单，各个功能模块应清晰可见便于用户操作。界面要美观大方，操作步骤清晰易学，操作对应的功能及实现不出现二义性。主界面应包括菜单栏以及文本域两个基本组件。菜单栏的功能分布：主要包括对系统的操作，信息的操作以及有关该软件一些相应的帮助信息。用户界面应该做到简单明了，方便用户操作。

5.2 房屋租赁信息管理系统功能概述

根据网站的需求分析和系统功能设计目标，本系统功能模块如下：

1. 用户注册：此模块用户可以注册账号，确认密码。在进行注册时，用户名不允许重复。
2. 用户登陆：登陆模块主要是对进入系统的用户身份的安全性检查，以防止非法用户进入系统，对系统用户提供一定的特殊权限。用户注册之后，就可以登陆系统登记所要出租的房屋的基本信息，或者是查找自己所需要的房屋的信息。
3. 填写个人信息：用户填写个人信息，主要是为了在房屋出租时方便联系。
4. 填写房屋信息：登陆到本系统的用户可以填写准备出租的房屋的信息，房屋信息应详细填写，使客户可以充分了解该房屋信息。
5. 用户查询：用户可以根据自己的需要查询房屋信息，比如：按照面积、租金查询房屋信息等。
6. 查看房屋状态：用户可以查询系统中所有房屋的现状，即是否已出租。
7. 修改信息：用户可以修改个人信息，包括联系电话，家庭住址等信息。同时，房屋出租者还可以修改自己准备出租但还未租出的房屋的信息。
8. 租房：租赁者在找到满意的房屋后，确定要租赁，可以现在网上确认租赁，然后再去公司缴纳租金，签订合同。
9. 用户管理：管理员可以添加和删除用户。
10. 租金管理：房屋租赁者到服务台缴纳租金时，管理员可以登陆系统确认租金并确认房屋状态。

5.3 系统总体功能的层次划分

根据详细的系统数据流图导出系统的功能模块, 即系统的功能层次的划分。

下图为系统总体的功能模块的层次划分, 系统的基本模块分为两个:

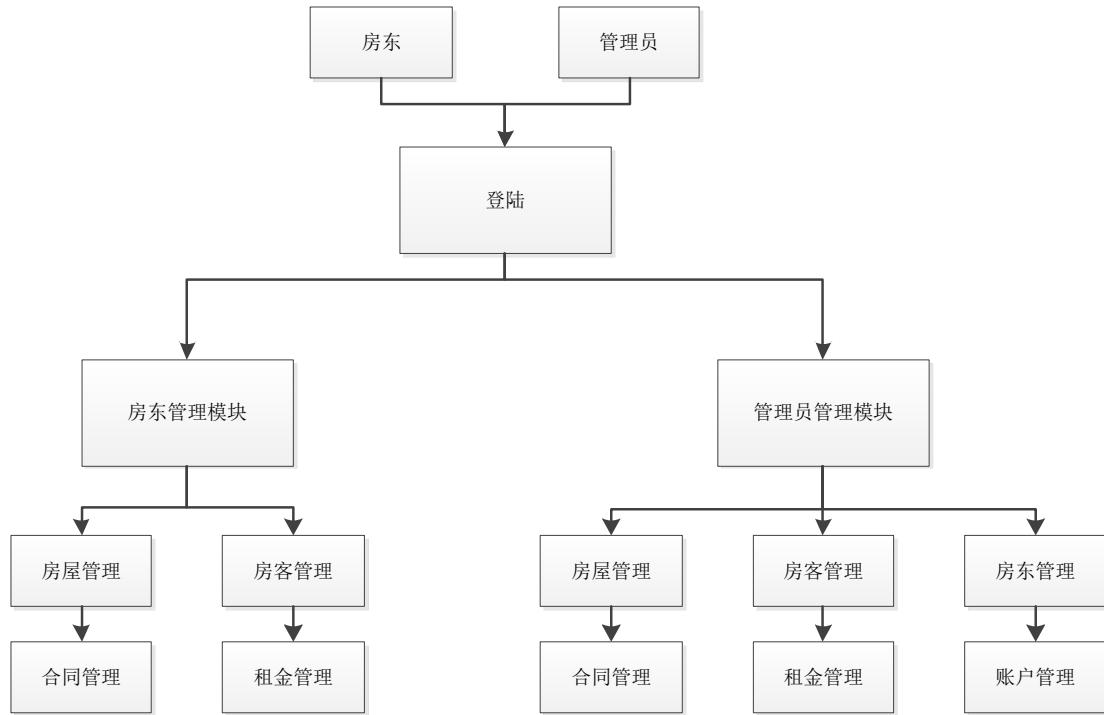


图 5.1 系统主要功能模块图

第 6 章 数据库设计

6.1 数据库 E-R 图

经过对系统的分析系统中涉及的实体有 4 个, 分别为房东, 房源, 房客, 管理员, 得出了他们的实体联系 E-R 图如下。

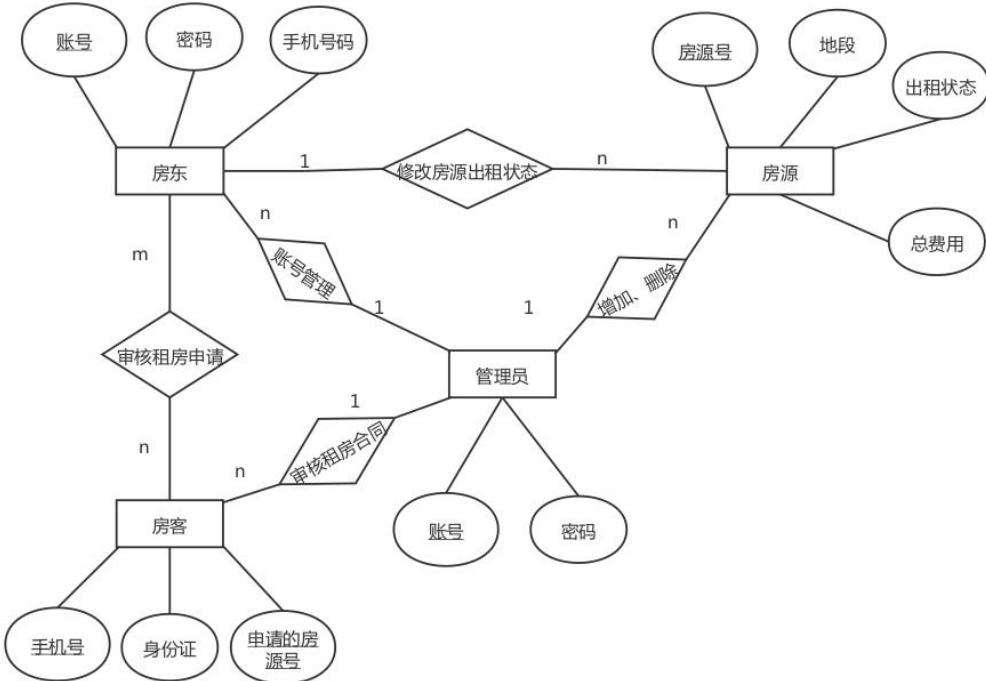


图 6.1 数据库 E-R 图

6.2 数据字典

用户信息	房源信息
描述:房东、管理员的信息 定义:账号+密码+层级+旗下房源 备注:层级为0代表管理员, 层级为1代表房东, 层级为1时, 房东旗下有房源	描述:房源的信息 定义:房源号+房东名+地段+出租状态(是否已出租)+租金状态(是否已收)+出租时间+到租时间+总费用
收益信息	房客信息
描述:收益包括的信息 定义:已收租金+未收租金	描述:房客的信息 定义:房客名(合同签约者)+房客身份证证号+电话号码+房源号+合同+审核状态(房客申请房源)

图 6.2 数据字典

6.3 数据库字段表

由于系统涉及房屋一些信息的查询,这些信息既包括房屋本身的属性也包括了众多房屋有关的信息。所以根据上边的 E-R 模型可以确定数据库实际存储的数据表单,数据表单如下。

表 1. 用户表

用户 (管理员、房东) **tb_User**

字段	含义	类型	备注
username	账号	varchar(15)	主键
password	密码	varchar(20)	
level	层级 (0 代表管理员、1 代表房东)	tinyint	
houses	当层级为 1 时, 房东旗下所管理的房源	varchar(20)	

表 2. 房源表

房源 **tb_House**

字段	含义	类型	备注
houseNum	房源号	varchar(20)	主键
landlord	房东名	varchar(20)	不可为空
address	地段	varchar(50)	不可为空
state	出租状态(是否已出租)	tinyint	0 代表否, 1 代表是
rentState	租金状态(是否已收)	tinyint	0 代表否, 1 代表是
outTime	出租时间	datetime	
inOutTime	到租时间	datetime	
totalMoney	总费用	money	

表 3. 房客表

房客 **tb_Tenant**

字段	含义	类型	备注
tenantID	房客身份证号	varchar(18)	主键
houseNum	房源号	varchar(20)	外键
tenant	房客名(合同签约者)	varchar(20)	不可为空
phone	手机号	int	不可为空
contract	合同	varchar(200)	不可为空
applyState	审核状态(房客申请房源)	tinyint	0 代表否, 1 代表是

表 4. 收益表

收益 **tb_Earnings**

字段	含义	类型	备注
received	已收租金	money	
unReceived	未收租金	money	

第 7 章 系统详细设计

以下程序流程图是针对总体设计中的系统功能层次图中的模块进行的详细设计。

7.1 房源出租流程图

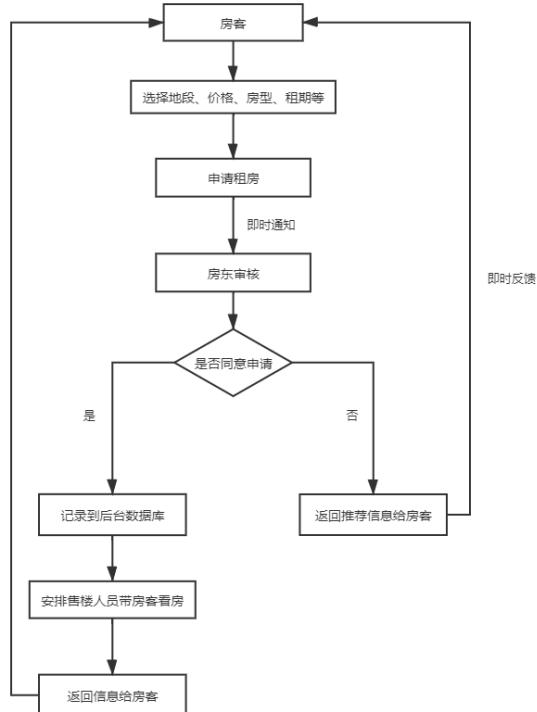


图 7.1 房源出租流程图

7.2 房东审核功能流程图

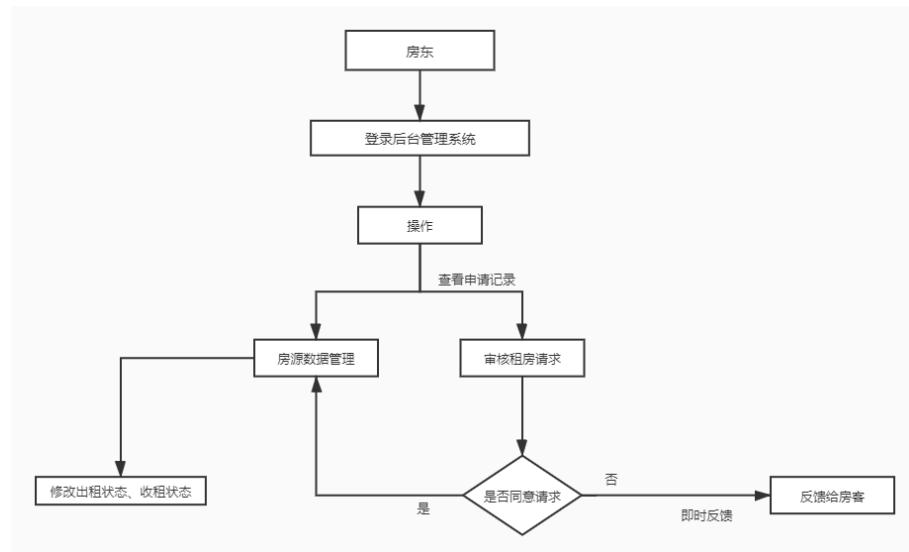


图 7.2 房东审核功能流程图

7.3 房东信息修改流程图

该程序流程图为房东信息修改操作的具体过程, 程序开始后首先连接数据库, 连接成功后输入房东信息。系统需要判断输入的房东信息的合法性, 不合法则显示错误提示, 并返回至输入房东信息处; 如果合法, 则再输入要修改的信息并存储房东信息, 程序结束。

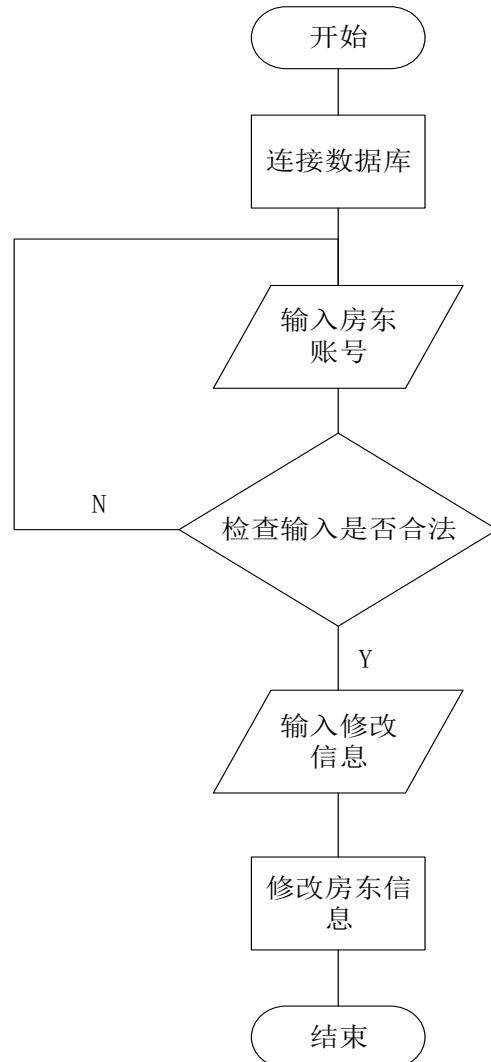


图 7.3 房东信息修改流程图

7.4 添加用户信息操作流程图

该程序流程图为房东信息添加操作的具体过程, 程序开始后首先连接数据库, 连接成功后输入房东信息。系统需要判断输入的房东信息的合法性, 不合法则显示错误提示, 并返回至输入用户信息处; 如果合法, 则将添加的用户信息保存到用户信息表中, 程序结束。

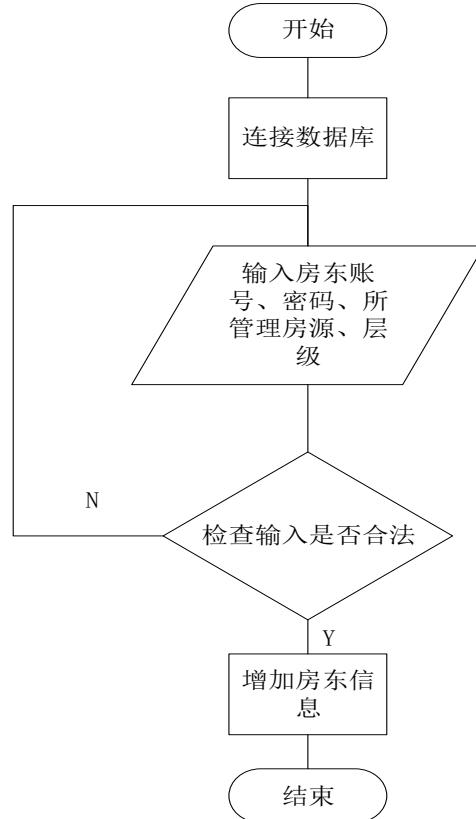


图 7.4 添加用户信息操作流程图

7.5 房东删除操作流程图

该程序流程图为房东信息删除操作的具体过程, 程序开始后首先连接数据库, 连接成功后输入房东信息。系统需要判断输入的房东信息的合法性, 不合法则显示错误提示, 并返回至输入用户信息处; 如果合法, 则将删除的用户信息从用户信息表中删除, 程序结束。

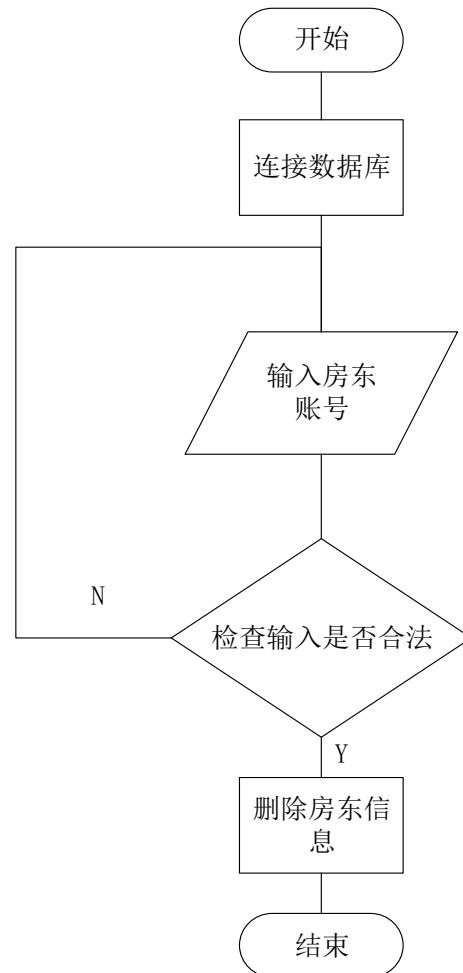


图 7.5 房东删除操作流程图

7.6 房源租金增加操作流程图

该程序流程图为房源租金添加操作的具体过程, 程序开始后首先连接数据库, 连接成功后输入房源信息。系统需要判断输入的房源信息的合法性, 不合法则显示错误提示, 并返回至输入房源信息处; 如果合法, 则输入租金数据, 并检查其合法性, 不合法则重新输入, 合法则对其进行保存。

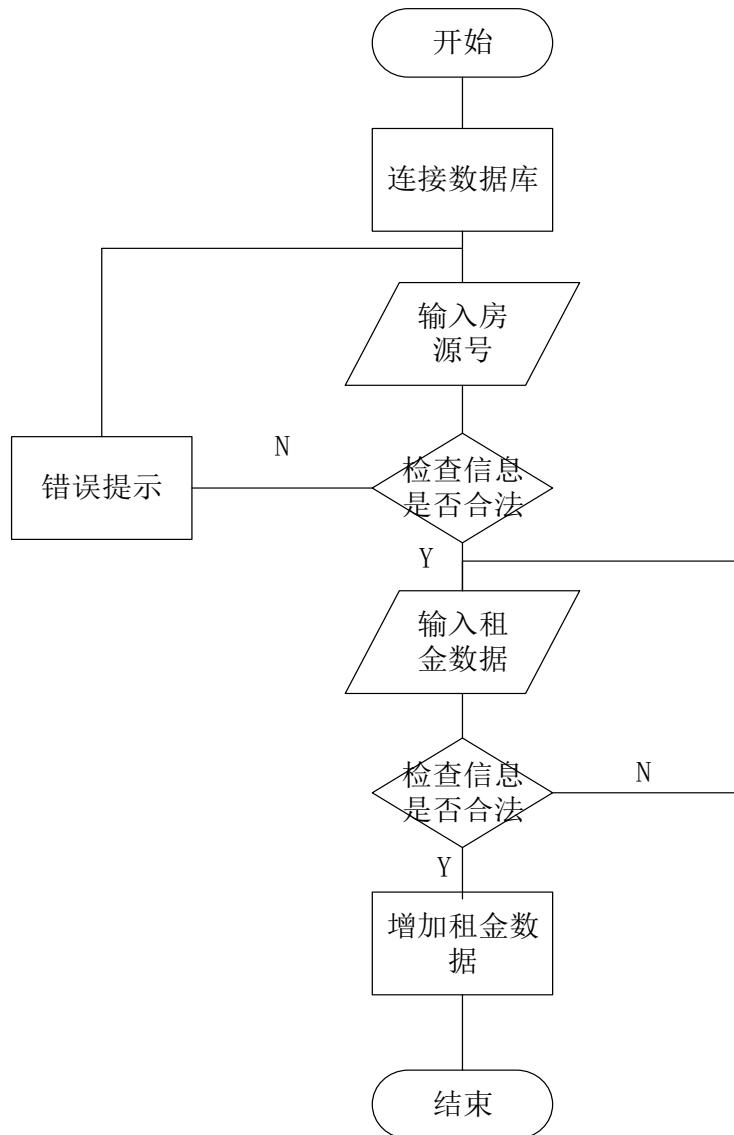


图 7.6 房源租金增加操作流程图

第8章 系统实现

8.1 系统实现说明

根据前面的需求分析和系统设计内容进行程序设计。本系统是运用前端 Vue 框架结合数据库 MongoDB 开发，运用 Web 框架 Express 来做服务端接口设计。

8.2 用户登录界面

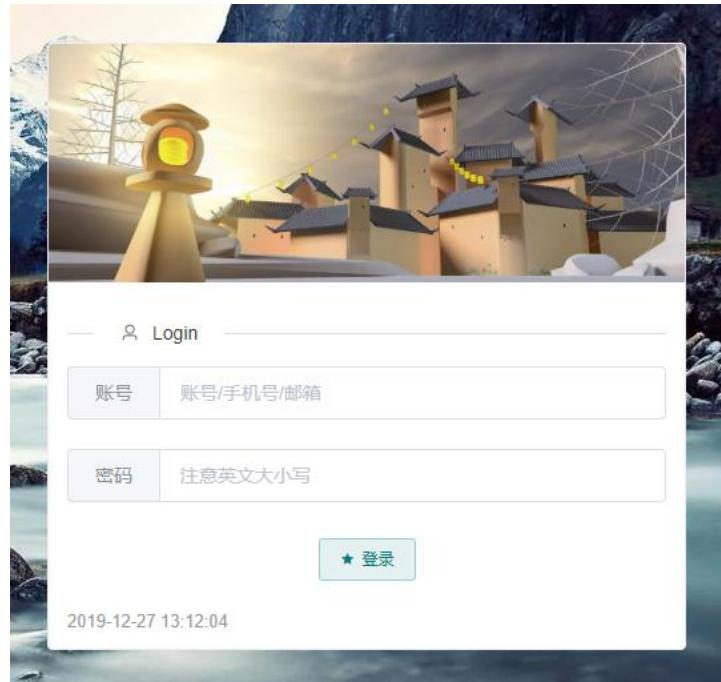


图 8.1 用户登录界面

8.3 登录成功进入管理系统首页

欢迎 – admin											管理员： admin
房号	房东	地段	出租状态	租金状态	房客	出租时间	到期时间	合同	基本租金 (RMB)	操作	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	
天下一号	admin	天和苑	未出租	未收	小明	2019-12-27 13:12:25	2019-12-27 13:12:25	已下发	99999	<button>已出租</button> <button>已收租</button> <button>待审核</button>	

图 8.2 系统首页

8.4 房客申请租房记录



The screenshot shows a table titled "房屋租凭信息管理系统" (House Rent Management System) with the following columns: 房客 (Tenant), 房号 (Room Number), 房东 (Landlord), 地址 (Address), 收租状态 (Rent Collection Status), 出租时间 (Lease Start Time), 到租时间 (Lease End Time), 合同 (Contract), 基本租金 (RMB) (Basic Rent (RMB)), 审核状态 (Review Status), and 操作 (Operations). There are 12 rows of data, each representing a tenant application. The "操作" column contains three buttons: "编辑" (Edit), "查看" (View), and "删除" (Delete).

图 8.3 房客请求记录

8.5 数据报表

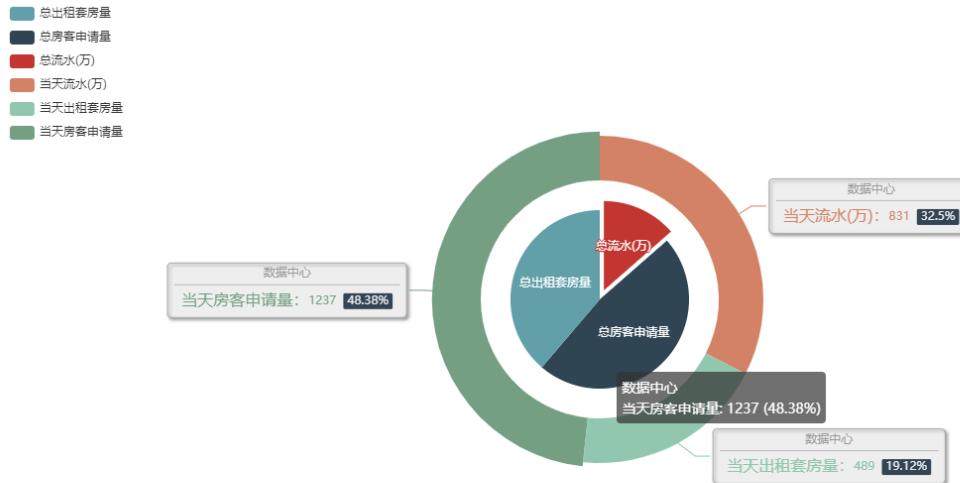
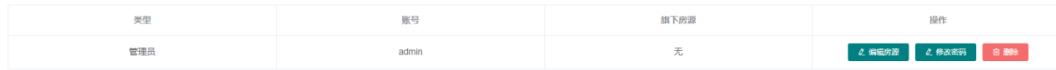


图 8.4 数据可视化

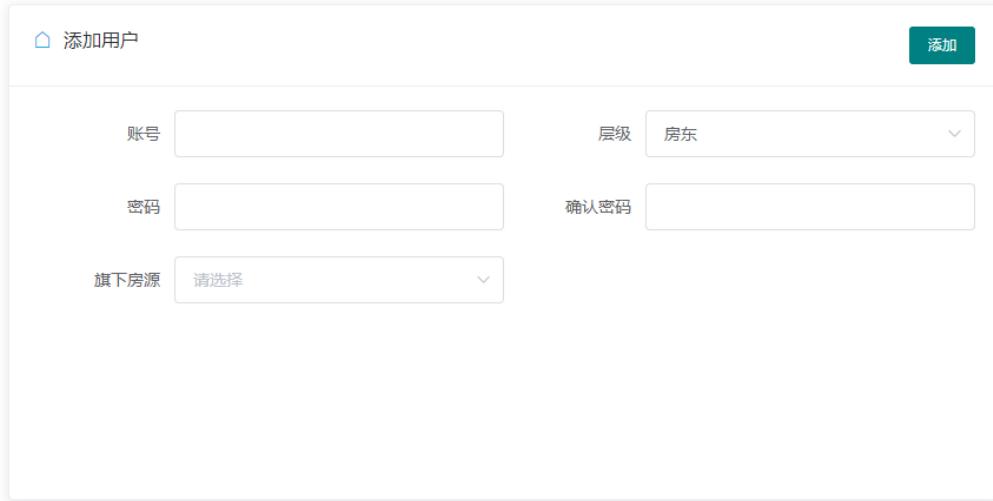
8.6 用户（管理员、房东）列表界面



The screenshot shows a table with the following columns: 类型 (Type), 账号 (Account), 旗下房源 (Subordinate Properties), and 操作 (Operations). There is one row of data for the administrator "admin". The "操作" column contains three buttons: "编辑房源" (Edit Properties), "修改密码" (Change Password), and "删除" (Delete).

图 8.5 系统用户界面

8.7 用户添加界面



The screenshot shows a user addition form titled "添加用户". It includes fields for "账号" (Account), "层级" (Level) set to "房东" (Landlord), "密码" (Password), "确认密码" (Confirm Password), and "旗下房源" (Properties under management) with a dropdown menu showing "请选择" (Please select). A green "添加" (Add) button is located in the top right corner.

图 8.6 用户添加界面

8.8 房源添加界面



The screenshot shows a property addition form titled "添加房源". It includes fields for "房号" (Room Number), "房东" (Landlord) with a dropdown menu showing "请选择" (Please select), "地段" (Location), "基本租金" (Basic Rent), "出租状态" (Lease Status) set to "未出租" (Not rented), and "收租状态" (Collection Status) set to "未收" (Not received). A green "添加" (Add) button is located in the top right corner.

图 8.7 房源添加界面

8.9 系统默认设置界面



图 8.8 系统默认设置界面

课程设计工作总结 - 黄钰淞

完成人: 黄钰淞

1. 工作概述

这次的软件工程课程设计我在组内主要负责系统测试方面。

2. 系统测试过程

首先是登陆测试，不能做到同时只能一个用户登陆的权限管理。但在密码错误后会有密码错误提示内容。登陆方式分为了用户和管理员两种登陆模式，管理员与用户操作界面不同之处在于管理房源信息等问题。进入主界面就是我们功能分类。第一次系统测试，我对比了做的比较大的同类型网站之间差别，我找出了我们缺少的需要增加的一些主要功能告示内容，例如地段，周边设施这类居家所关心的问题，以及我们作为一个中介人如何与房东租客取得联系以及核实身份问题，我也及时与我们开发人员联系。

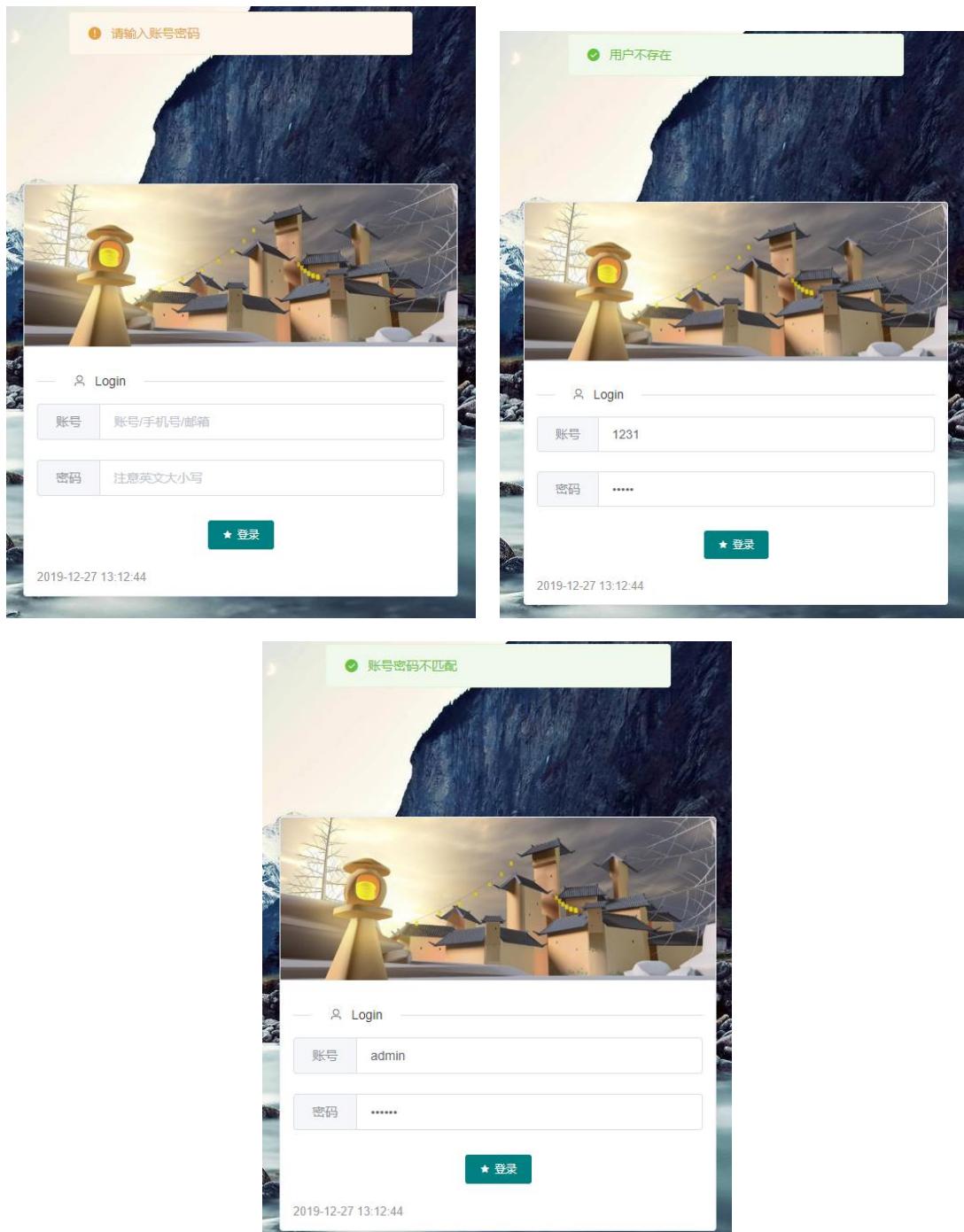
第二次的测试是有关租客方面内容，涉及到了这个系统安全问题。我们现在的技术和所学知识无法做到对用户信息保密，这是我们通过以后学习可以改进的地方。同样，对照其他网站我发现了我们缺少提供房源照片信息功能，因为时间关系我们只能在此一提。我们在第二次测试中也完善了一个房东管理多个房源信息的功能，房东能够更轻松处理租房退房的事情，也由于时间问题，在最后一次测试才发现缺少房东联系方式，这可能导致了租客不能及时联系房东租房。

以下是一些测试图例

修改出租状态与修改成功提示:



登录错误提示：



3. 体会

我三次自己测试发现团队开发这个网站的不足，只有向好的网站学习，并且虚心接受意见我们才能把功能和内容做的更完善。而通过此次分工也让我更深刻明白合作力量。

课程设计工作总结 - 钟 祯

完成人: 钟祯

1. 工作概述

在本次课程设计中，我主要负责系统实现部分（编码工作），以及部分详细设计，需求分析。

2. 系统开发过程

网站后台管理系统主要是用于对网站前台的信息管理，如文字、图片、影音、和其他日常使用文件的发布、更新、删除等操作，同时也包括房源信息、申请信息、房客信息的统计和管理。

在权限管理设计过程中，设计一套权限管理规则，我们需要先了解权限分配的几个关键要素。后台系统的用户是公司的房东和管理员，跟他们相关联的要素就有了房源、房客；而权限代表的则是一个个功能，每一个功能的使用都是由不同的菜单和按钮触发完成，所以菜单和按钮就可以作为权限的基本要素。

例如同一个部门下相同职位的两个用户，也可能拥有不同的权限，比如运营甲负责头像图片审核，运营乙负责举报信息审核，这样的话也便不能将部门和职位作为连接权限的基础单位。一开始本想将用户与权限直接关联，每创建一个用户账号时就给它选择相对的权限，但这种操作过于繁琐，相同权限的用户就需要重复添加权限。因此我们又创建了一个角色的概念，我们只需要将创建好的角色与权限关联，然后再给创建好的用户账号赋予指定的角色，对于相同权限的账号给它们赋予同一个角色，需要修改权限时我们也只需要修改相对的角色就好。由于该项目中没有太多的业务代码，所以还是相对比较简单的，不过从开发中还是积累了一些经验，在其它项目中可以更加熟练地开发。

关于后台系统的.设计我也还在不断的学习当中，主要是思维模式的一种转变。也尝试了很多没有尝试过的东西，可以跳脱我以往只关注视觉设计的角色思维，也算是一种新技能的加持。最后感谢在课程设计中和我配合的同学以及指导我们的老师。

以下是接口源代码：

```
module.exports = app => {
  const express = require('express')

  // 使用 express 内置路由中间件
  const router = express.Router()

  // 引入数据库模型
  const User = require('../models/User')
  const House = require('../models/House')

  // 登录
```

```

router.post('/login', async (req, res) => {
  const { username, password } = req.body
  const user = await User.findOne({ username })
  let code =
  if (!user) {
    code = '用户不存在'
    res.send(code)
    return
  }
  if (password != user.password) {
    code = '账号密码不匹配'
    res.send(code)
    return
  }
  res.send({ username: user.username, level: user.level })
})

// 注册
router.post('/register', async (req, res) => {
  res.send(await User.create(req.body))
})

// 查询房源信息
router.get('/houseInfo', async (req, res) => {
  res.send(await House.find())
})

// 添加房源
router.post('/addHouse', async (req, res) => {
  res.send(await House.create(req.body))
})

// 删除房源
router.delete('/deleteHouse/:id', async (req, res) => {
  res.send(await House.findByIdAndDelete(req.params.id))
})

// 修改出租状态
router.post('/editState', async (req, res) => {
  res.send(await House.findByIdAndUpdate(req.body._id, req.body))
})

// 修改收租状态
router.post('/editRentState', async (req, res) => {

```

```
res.send(await House.findByIdAndUpdate(req.body._id, req.body))
})

// 查询用户
router.get('/userList', async (req, res) => {
  res.send(await User.find())
})

// 添加用户
router.post('/addUser', async (req, res) => {
  res.send(await User.create(req.body))
})

// 删除用户
router.delete('/deleteUser/:id', async (req, res) => {
  res.send(await User.findByIdAndUpdateDelete(req.params.id))
})

// 使用 router 中间件
app.use('/api/client', router)
}
```

3. 体会

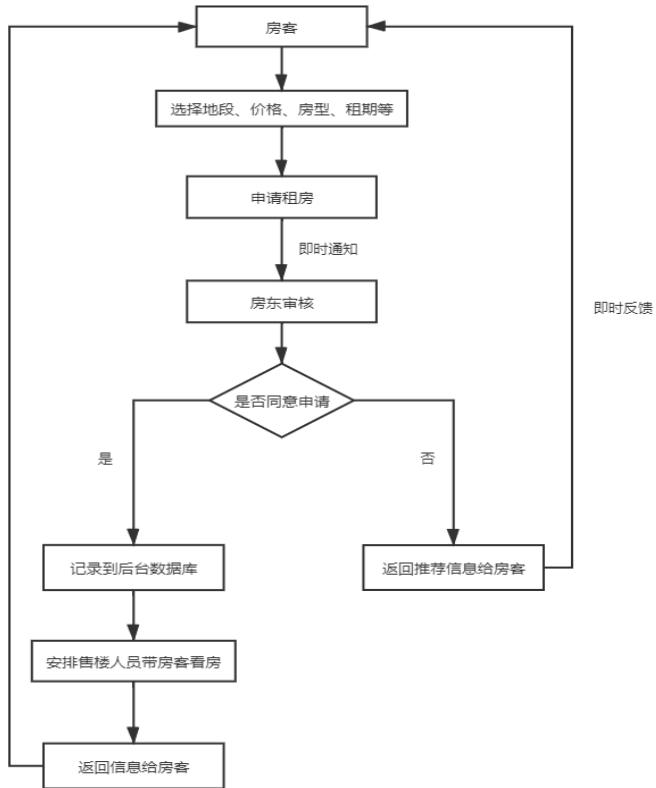
在系统开发与设计过程中，通过对出现的错误问题的解决过程中，我发现有些问题出现的原因其实是需求分析没有分析全面，对应不上导致的，没有能够完全考虑实际情况。因此，在今后学习过程中，我们一定要细心细心再细心，争取设计和开发出一个功能更加完善、全面的信息系统。

课程设计工作总结 - 陈荣煜

完成人: 陈荣煜

1. 工作概述

在这次的课程设计中，我主要负责报告编写，参与系统需求分析、可行性分析和数据库设计工作，完成了对房屋租赁后台管理系统的初步开发设想。在完成需求分析的过程中我对系统的业务进行了分析，画出了系统的业务流程图。我觉得一个能够满足用户使用的系统必须要不断地进行需求分析和可行性分析的讨论，在讨论中发现问题、提出问题、解决问题。我们这次课程设计采用的是面向对象的设计方法，面向房东，面向管理员，面向房客，逐步进行分析讨论。我主要负责了房东、房客模块分析，房东拥有房源的增删改查，对房客信息的查询，租金的修改等，房客租房流程包括：房客在前台系统中浏览房源信息，看中房源后申请租房，后台管理系统就将请求通知给房东，房东登录后台管理系统后处理请求，如何房东同意请求，后台系统将安排人员带领房客去看房。如果房东不同意请求或者过期未处理，后台管理系统则将信息反馈给房客，并向其推荐类似的房源。



2. 系统分析过程

在系统分析的过程中我觉得比较重要的是分析的方法与过程，在这次的课程设计中，我们小组采用了面向对象分析的方法。面向对象分析有着比传统分析方法更多的优点：

①面向对象方法基于构造问题领域的对象模型，以对象为中心构造软件系统。当功能需求发生变化时，往往仅需要一些局部性的修改。这个比较符合我们这次的系统开发，前期讨论完后，在后期的开发中发现了不少的问题，但是更改起来还是比较容易的，只修改了一些局部的东西。

②面向对象的软件稳定性比较好，比较容易修改、易于测试和调试等，面向对象设计开发有着良好的人机交互界面，用户容易理解和操作。

③面向对象方法的基本原则是按照人类习惯的思维方法建立问题域的模型，开发出尽可能直观、自然地表现求解方法的软件系统。面向对象的软件系统中使用的对象，是对客观世界中实体的抽象。

3. 体会

经过了这次为期四天的软件工程课程设计，我更好得巩固了这个学期从软件工程这门课上学到的知识。从项目的可行性研究到需求分析、结构化设计，然后再到 UML 建模，都运用到了我们这个学期的学习内容，这次的课程设计是组队完成的，通过与队员间的交流讨论，我们不断的对系统的开发进行及时的更正，在发现错误，修改错误中共同进步，在最开始的时候由于我们对系统需求分析得不够正确，出现了好多的问题，比如对房客的租房需求的分析，房客如何进行租

房，后台系统又是如何对租房请求进行处理的等等的问题上没有处理的很好，在最初的文档描述上也没有很好的直观解决这个问题，最后在请老师进行提点时，发现了这个错误，然后我们进行了讨论，对房客租房，房东管理，和管理员的操作流程、解决流程都做出了更好的描述，也画出了能一目了然的图表。

在最后的一天，房屋租赁信息后台管理系统基本完成。虽说不算尽善尽美，但基本功能与要求也尽可能的全部实现在内。此次课程设计我收获颇多，学习到了很多很多东西，了解了一个系统开发的整个过程，如果涉及系统功能，如何画数据流程图等，同时也看到了自己的不足点，在与队友的合作间也学到了很多东西，很高兴这次能和他们一起完成软件工程的课程设计。

课程设计工作总结 - 何嘉毅

完成人： 何嘉毅

1. 工作概述

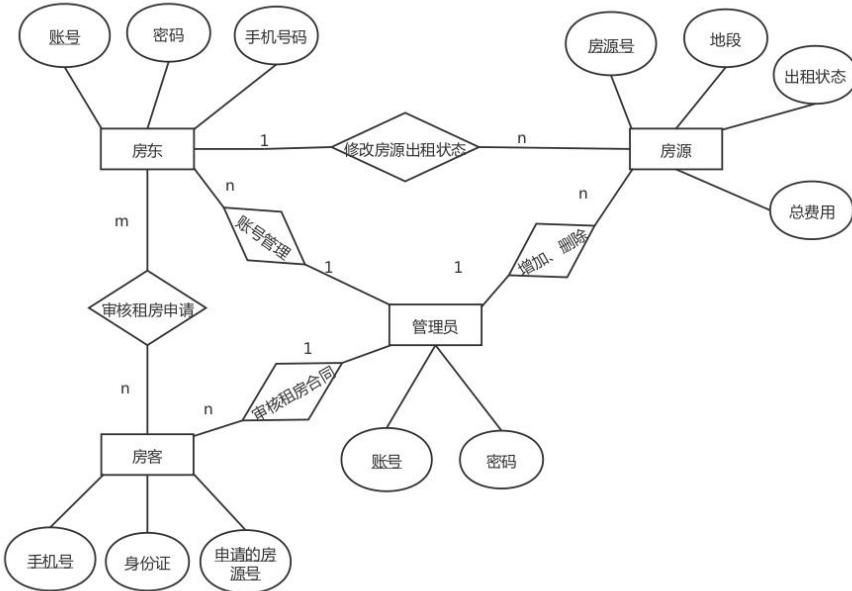
这次的软件工程课程设计我在组内主要数据库设计，负责画图，报告方面。

2. 数据库设计过程

进行数据库设计分析时，要了解我们的项目是什么，流程是什么，实体是什么。了解之后又要对使用者与需求者进行需求分析，画出 E-R 图，比如用户，我们要确定用户的账号为主键的用户表以及其他信息的分析，用户密码，用户信息，用户发布，房源的各种信息，房屋位置，价格，板式，是否出租，房主信息，地段，确定房号为主键做房源表，一些外键根据需求逐个添加，再根据需求分析对受益表进行设计，一些主键外键的链接。保证系统的顺利运行。

第二，在做完数据库分析后要熟悉 SQL 语句操作，好的设计对之后的升级数据库或者修改数据库可以产生更好的效果，接着就是 E-R 图的设计，数据库是连接各个项目的重点，

好的 E-R 图设计会让思维更条理化，清晰化，可视化。更好的，更方便的分析我们的需求，或者项目的需求，项目的更改，总流程图依靠着 E-R 图的样式。可以说 E-R 图在项目初起到了关键的作用。



第三，将报告更清晰化改动，对后来负责的同学起到了作用。

3. 体会

我们做完后还是觉得有些不足，不仅是数据库的部分，还有更多，我们做的只是表面的东西，更深层的东西没有规划好，如果以后做项目这样做的话，客户的需求没有得到充分的满足，意味着我们的项目是失败的，在今后的学习中，我们应该更加完善自己，争取学习更多方面的知识，对于突如其来的需求能更好的满足。

课程设计工作总结 - 李文洁

完成人： 李文洁

1. 工作概述

在本次课程设计中，我主要负责系统的流程图和数据流图的编写和绘画。

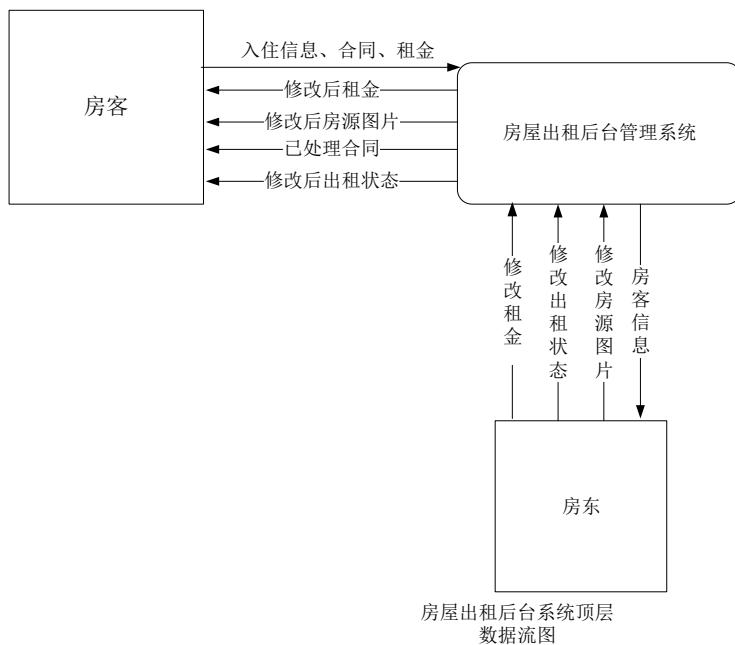
2. 图的编写绘画过程

要绘制功能流程图首先要确认系统如何实现功能，需要调用什么业务。明确业务后才能进行对功能流程图的绘制。而绘制过程中要理清楚各个业务之间的联系和调度关系；

而数据流图则是要清楚数据流的走向，数据是如何从外部实体流入系统，然后在系统中调用什么模块将信息处理后反馈信息给外部实体。数据流图要进行 3 层的绘制：顶层数据流图是外部时和整个系统之间数据流动的关系；0 层图是数据流和各大功能模块之间的关系；

1 层图则是数据流与各个业务之间是如何交互的。理清楚以上的关系才能够绘制出关系明确的数据流图。确定好图是如何绘制后就可以使用 vision 进行绘制。

在经过询问老师后，得知之前图不够明确，无法让阅读者看明白各个业务之间的关系。在老师的指导下又重新设计图的关系逻辑，经过小组讨论后确定图的最终绘制



3. 体会

在图的绘制过程中，我发现图表表达不明确出现的原因其实是需求分析没有分析全面，对应不上导致的，没有能够完全考虑实际情况；还有一部分是数据库的设计导致无法明确表示各属性之间的关联。因此，在今后我们需要对需求分析进行再三讨论，结合实际使用情况后再开始着手对图进行绘制，数据库设计方面也应该考虑到图的表达性进行属性设计，而不是发现问题后再修改这种被动的方式进行绘图。

结语

通过为期一周的课程设计，该房屋租赁信息管理系统报告基本完成。虽说不算尽善尽美，但基本功能与要求也尽可能的全部罗列在内。

在这次课程设计中，我们重新复习了课本中关于软件工程设计的基本步骤与要求，尽可能的严格按照书中所说的进行。关于研究背景与可行性分析的叙述，增强了我们对本次房屋租赁信息管理系统必要性及可行性的了解；需求分析方面，使我们对本次设计的系统要求有了总体概况，基本流程图及数据字典等的描述使我们对软件设计的功能描述更加熟练；总体设计与详细设计是最消耗时间与精力的，需要把方方面面尽可能的考虑完全，让每一个模块都尽可能的完善。虽然工作

量有点大,但每完成一部分,就能收获一份喜悦。总的来说,在这次房屋租赁信息管理系统的设计中遇到了不少问题,最后经过坚持不懈的努力,终于得到解决,而且感觉到了平时所学的知识有了实用的价值,是理论与应用的真正结合。

这次课程设计不仅学到了不少知识,而且锻炼了自己的能力。当然系统功能没有十分完善和具体,在以后的学习过程当中可以进一步加以完善。最后,对给过我们帮助的所有指导老师和同学再次表示衷心的感谢!

参考文献

- [1] DavidFlanagan, 弗兰纳根,&淘宝前端团队. (2012). Javascript 权威指南.
- [2] 王立峰. 软件工程理论与实践[M]. 清华大学出版社, 2003.
- [3] Robert C. Martin. 代码整洁之道[M]. 2010.
- [4] 吕云翔, 王洋, 肖咚. 软件测试案例教程[M]. 2011.
- [5] 吴洁明, 袁山龙. 软件工程应用实践教程[M]. 2003.
- [6] 李彤, 王炜, 郁湧. 软件工程概论[M]. 科学出版社, 2012.

附录 A

后台 API 源代码如下：

```
module.exports = app => {
    const express = require('express')

    // 使用 express 内置路由中间件
    const router = express.Router()

    // 引入数据库模型
```

```
const User = require('../models/User')
const House = require('../models/House')

// 登录
router.post('/login', async (req, res) => {
    const { username, password } = req.body
    const user = await User.findOne({ username })
    let code = ''
    if (!user) {
        code = '用户不存在'
        res.send(code)
        return
    }
    if (password != user.password) {
        code = '账号密码不匹配'
        res.send(code)
        return
    }
    res.send({ username: user.username, level: user.level })
})

// 注册
router.post('/register', async (req, res) => {
    res.send(await User.create(req.body))
})

// 查询房源信息
router.get('/houseInfo', async (req, res) => {
    res.send(await House.find())
})
```

```
// 添加房源
router.post('/addHouse', async (req, res) => {
    res.send(await House.create(req.body))
})

// 删除房源
router.delete('/deleteHouse/:id', async (req, res) => {
    res.send(await House.findByIdAndDelete(req.params.id))
})

// 修改出租状态
router.post('/editState', async (req, res) => {
    res.send(await House.findByIdAndUpdate(req.body._id, req.body))
})

// 修改收租状态
router.post('/editRentState', async (req, res) => {
    res.send(await House.findByIdAndUpdate(req.body._id, req.body))
})

// 查询用户
router.get('/userList', async (req, res) => {
    res.send(await User.find())
})

// 添加用户
router.post('/addUser', async (req, res) => {
    res.send(await User.create(req.body))
})
```

```
// 删除用户
router.delete('/deleteUser/:id', async (req, res) => {
  res.send(await User.findByIdAndDelete(req.params.id))
})

// 使用 router 中间件
app.use('/api/client', router)
}
```

附录 B

服务端入口程序源代码如下：

```
// 引入 express
const express = require('express')

// 创建实例
const app = express()

// 使用 cors 解决跨域
app.use(require('cors')())

// 使用 json 格式传输数据
app.use(express.json())

// 挂载路由
require('./routes')(app)

// 挂载数据库
require('./plugins/db')(app)
```

```
// 启动服务，监听 2017 端口
app.listen(2017, () => {
    console.log('http://localhost:2017')
})
```

附录 C

前端路由配置源代码如下：

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import Login from '../views/Login.vue'

Vue.use(VueRouter)

const routes = [
    {
        path: '/',
        name: 'login',
        component: Login,
        meta: {
            isLoggedIn: true
        }
    },
    {
        path: '/home',
        name: 'home',
        component: () => import('../views/Home.vue'),
        children: [
            {

```

```
path: '/house',
      name: 'house',
      component: () => import('../views/House/List.vue')
    },
    {
      path: '/addHouse',
      name: 'addHouse',
      component: () => import('../views/House/Create.vue')
    },
    {
      path: '/addUser',
      name: 'addUser',
      component: () => import('../views/User/Create.vue')
    },
    {
      path: '/landlord',
      name: 'landlord',
      component: () => import('../views/User/List.vue')
    },
    {
      path: '/tenant',
      name: 'tenant',
      component: () => import('../views/Tenant/List.vue')
    },
    {
      path: '/data',
      name: 'data',
      component: () => import('../views/Chart/Data.vue')
    },
    {
```

```
path: '/setting',
      name: 'setting',
      component: () => import('../views/Set/SetCenter.vue')
    }
  ]
}

]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

router.beforeEach((to, from, next) => {
  if (!to.meta.isLogin && !localStorage.getItem('username')) {
    return next('/')
  }
  next()
})

export default router
```

附录 D

前端状态管理源代码如下：

```
import Vue from 'vue'
import Vuex from 'vuex'
```

```
Vue.use(Vuex)
```

```
export default new Vuex.Store({
  state: {
    username: '' || localStorage.getItem('username'),
    userlevel: '' || localStorage.getItem('userlevel')
  },
  mutations: {
    saveUserName: (state, userData) => {
      state.username = userData.username
      state.userlevel = userData.level
      localStorage.setItem('username', userData.username)
      localStorage.setItem('userlevel', userData.level)
    }
  },
  actions: {},
  modules: {}
})
```

附录 E

前端入口程序源代码如下：

```
import Vue from 'vue'
import App from './App.vue'
import router from './router'
import store from './store'
import dayjs from 'dayjs'
import './plugins/element.js'
import http from './http'
```

```
Vue.config.productionTip = false

// 将 axios 实例 (http) 挂载到 Vue 的原型上
Vue.prototype.$http = http

// 定义全局过滤器，格式化时间戳为标准格式
Vue.filter('date', value => {
  return dayjs(value).format('YYYY-MM-DD HH:MM:ss')
})

// 创建 Vue 实例
new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app')
```

附录 F

源代码仓库（GitHub）地址：

<https://github.com/zhenzhenChange/HousingRental-Software.git>