****

**本科生毕业论文（设计）册**

学 院： 软件学院

专 业： 软件工程

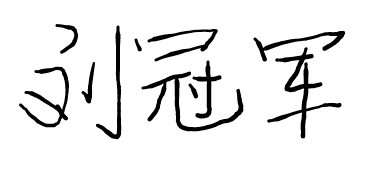
年 级： 2017 级

学生姓名： 王炎

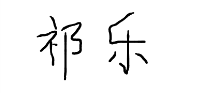
指导教师： 刘冠军

**河北师范大学本科生毕业论文（设计）任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **姓 名** | **王炎** | **学院** | **软件学院** | **专业** | **软件工程** | **年级（班）** | **201704** |
| **毕业论文（设计）题目** | | **笔记记录平台前端的设计与实现** | | | | | |
| **指导教师** | **刘冠军** | **职称** | **工程师** | **教研室** | H**5教研室** | **研究方向** | H5**方向** |
| **论文（设计）基本要求：**  基本功能：   1. 小区住户信息管理模块：对住户的基本信息，家庭住址等进行信息管理 2. 房屋信息管理模块：房屋的构造，面积，出售信息等信息管理 3. 收费模块：对小区住户进行小区物业费，水电费和暖费等进行管理 4. 人事模块：物业人员，小区保洁和保安等人员进行人事管理 5. 故障模块：小区内部的公物或者住户的房屋故障报修管理 6. 投诉模块：对投诉信息进行及时的接受和反馈 7. 环境管理模块：小区绿化方面的管理 | | | | | | | |
| **论文（设计）研究目标：**  本论文（设计）研究目标是为了小区物业更好地管理信息，开发的一个便捷、高效、使用价值高的管理系统。 | | | | | | | |
| **主要参考文献：**  [1]王鹏.智能小区物业管理系统设计与实先[D].江西财经大学,2019年.  [2]徐连霞，傅伟.小区物业管理系统的设计与实现[J].现代营销（信息版），2019年.  [3]赵强，范丰雷，孙晓霞.智能化住宅小区的物业管理[J].山东行政学院山东省经济管理干部学院学报，2004年5期.  [4]郭保卫，苑丰.基于Web的小区物业管理系统[J].信息与电脑（理论版），2018年. | | | | | | | |

****

指 导 教师： 2019年11月20日

****

教研室主任： 2019年11月20日

**河北师范大学本科生毕业论文（设计）开题报告书**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓 名** | | **王炎** | **学院** | **软件学院** | **专业** | **软件工程** | **年级（班）** | **201704** |
| **论文题目** | | **笔记记录平台前端的设计与实现** | | | | | **指导教师** | **刘冠军** |
| **课题论证** | 随着人们生活水平的不断提高，很多高楼拔地而起，小区单元房也成了人们的首要选择，小区也逐渐变得多种多样，小区的信息管理随之也变得越来越重要，对小区物业的各方面的管理要求也越来越严格。在这种情况下，便捷的小区物业管理系统成为了现代小区管理信息不错的选择，利用计算机物业系统管理小区住户信息，房屋信息，人员管理等，不仅可以对信息进行完整的管理，而且也可以大大的提高物业人员的工作效率，从而小区住户舒适度和满意度也会大大地提高。 | | | | | | | |
| **方案设计** | 1. 需求分析，对现在的小区进行大致的分析，罗列功能模块，分为住户信息、房屋管理、收费管理等模块 2. 页面设计，在墨刀等原型设计工具中进行页面设计工作 3. 相关技术，学习相关应用技术，使用React和Ant Design对系统进行前端编写，利用axios技术获取数据 | | | | | | | |
| **进度计划** | 2019.11.20-2019.11.23 论文选题，收集材料  2019.12.03-2019.12.5 确定实用技术，搭建主要的框架  2020.02.02-2019.02.30 前端页面进行编写，完成大致页面的编写  2020.03.23-2019.04.05 接口工作，实现数据库的增删改查  2020.04.10-2020.04.20 进行测试工作，对各模块进行测试 | | | | | | | |
| **指**  **导**  **教**  **师**  **意见** | 选题合理、难易适中、有一定的实际价值，同意开题  **指导教师：**  **2019 年12 月4 日** | | | | | | | |
| **教研室意见** | 选题恰当可行，同意开题  **D:\重要资料勿删\桌面\image.png**  **教研室主任：**  **2019年12 月4 日** | | | | | | | |

**河北师范大学本科生毕业论文（设计）文献综述**

|  |
| --- |
| 1. 国内该研究现状   物业管理最先出现在欧美一些国家，伴随着工业革命的产生，在经历了这么多年的不断地更新和改善，国外的物业服务很成熟了，也有了专门的物业公司，国外Reflexf公司提出运用互联网技术对信息进行智能化管理，清晰便捷的实现了对小区内部的信息管理，这就会提高服务态度和服务。现在国外已经普遍应用了专业的物业管理技术。改革开放以来，我国的经济迅速发展，城市中有大量的工作机会，从而就会有大批的工作者涌入城市，久而久之住房也就成了问题，一栋栋的单元房也逐渐耸立起来，所以优秀的物业服务也成了人们选择购房的一个选择条件之一。我国物业管理正处于发展缓慢阶段，好多小区还没有实现信息化的管理，为了解决一些复杂的数据和一些不同程度的服务，小区开始结合计算机技术来使用管理系统，适应了我国住房体制的发展形势。   1. 发展趋势   随着计算机技术的不断发展，还有一些计算机技术，可以将一些复杂的问题简单化，而且利用模块化的思想，将不同的但是一个受理人的情况放在一个系统中，这样就实现了系统化的管理，而且小区管理越来越趋向整体化，正规化，在互联网的作用下，利用有效管理系统成为一种新的发展趋势，这样就能进一步提升小区管理的质量，基于平台的系统管理也会不断地优化，在今后一段时间内，我国的信息管理也会逐渐地成为一种发展趋势。   1. 存在的问题   功能过于单一，现如今为了方便小区管理的某一功能，开发的单一式的功能软件有很多，但是将住户信息管理、房屋管理和收费管理等一体化的系统发展还是不太齐全。  小区发展速度快，但是一些设备不能跟上住户的需要，无法实现物质生活满足人们对美好生活的需求。物业管理水平有待提高，有些小区自身是没有物业的，还需要聘请一些外面专业的物业公司，这就使得小区很难管理，对小区的信息不了解等问题  物业公司和业主之间的矛盾。因为一些物业服务水平不高，使得一些在外的业主拒绝交费，从而使得一些物业公司不能继续经营。   1. 文献概括   徐连霞在《小区物业管理系统的设计与实现》中总结到：物业管理系统的应用使得小区物业管理人员可以及时对小区的资源进行管理，节省人力、物力和财力，提高了小区物业管理人员的工作效率，提升了小区物业服务水平和小区档次。这就表明了小区物业管理系统对于信息的管理有着非常重要的作用。  王鹏在《智能小区物业管理系统设计与实现》中写：物业管理行业作为微利行业，利用信息系统进行物业管理能够全面降低物业管理成本，满足人们对物业管理所提出的新要求，在有限的资源和成本范围内提升物业管理水平和物业管理质量，进而提升企业的市场竞争力。因此，结合物业管理的需要开发出智能化物业管理信息系统成为了物业管理的重中之重。   1. 参考文献   [1]李长海.小区物业管理系统的设计与实现[D].北京工业大学,2016年.  [2]王鹏.智能小区物业管理系统设计与实先[D].江西财经大学,2019年.  [3]徐连霞，傅伟.小区物业管理系统的设计与实现[J].现代营销（信息版），2019年. |

**河北师范大学本科生毕业论文（设计）翻译文章**

|  |
| --- |
| 如今，使用需要在各个页面之间导航的web应用程序模型似乎已经过时了……真的过时了,现代应用程序倾向于遵循所谓的单页应用程序（SPA）模式。在这个环境中，你不需要导航到不同的页面，甚至不用重新加载页面。相反，你的应用程序的不同视图会加载并不加载到同一页面本身。当用户与我们的应用程序交互时，我们用与用户正在尝试做的事情匹配的数据和HTML替换了虚线红色区域的内容。最终的结果是更加流畅的体验。你甚至可以使用大量的视觉技术来让你的新内容很好地过渡，就像你在移动设备或桌面看到很有意思的应用程序一样。当导航到不同的页面时，这类事情是不可能的。如果你以前从未听说过单页应用程序，这一切听起来可能有点疯狂，但你很有可能在野外遇到过其中的一些。如果你曾经使用过Gmail、Facebook、Instagram或Twitter等流行的网络应用程序，那么你使用的是一个单页应用程序。在所有这些应用程序中，内容都是动态显示的，不需要刷新或导航到另一个页面。  现在，我让这些单页应用看起来非常复杂。事实并非完全如此。由于JavaScript和各种第三方框架和库都有了很大的改进，构建单页应用变得前所未有的简单。但这并不意味着没有改进的余地。在构建单页应用程序时，您将遇到三个主要问题：   1. 在单页应用程序中，您的大部分时间将花在保持数据与UI同步上。例如，如果用户加载新内容，我们是否显式地清除搜索字段?导航元素上的活动选项卡是否仍然可见?我们在页面上保留了哪些元素，又销毁了哪些元素?这些都是单页应用程序特有的问题。当在旧模型的页面之间导航时，我们只是假设UI中的所有东西都会被破坏，然后重新构建。 2. 操作DOM非常慢。在浏览器中，手动查询元素、添加子元素、删除子树和执行其他DOM操作是最慢的操作。不幸的是，在单页应用程序中，操作DOM是响应用户操作和显示新内容的主要方式。 3. 使用HTML模板可能是一件痛苦的事情。在单页面应用程序中，导航只不过是处理HTML片段来表示希望显示的内容。这些HTML片段通常被称为模板，使用JavaScript操作它们并用数据填充它们会很快变得非常复杂。更糟糕的是，根据您使用的框架，模板的外观和与数据交互的方式可能有很大的不同。   Facebook(和Instagram)决定适可而止。考虑到他们在单页应用程序方面的丰富经验，他们发布了一个名为React的库，不仅解决了这些缺点，还改变了我们对构建单页应用程序的看法。  自动UI状态管理  对于单页应用程序，跟踪用户界面和维护状态非常困难，而且非常耗时。使用react，您只需要担心一件事:UI的最终状态。UI的初始状态并不重要。您的用户可能已经采取了哪些步骤来更改UI并不重要。所有重要的是你的UI在哪里结束。用户界面的最终或结束状态是React中重要的部分。React会处理所有其他事情，它会找出需要发生什么来确保UI的正确表示，所以所有的状态管理工作都不再是你关心的问题。  快速的DOM操作。  因为DOM修改非常慢，您永远不会直接使用React修改DOM，而是修改内存中的虚拟DOM。操作这个虚拟DOM非常快，React会在适当的时候更新真实的DOM，通过比较虚拟DOM和真实DOM之间的变化，找出哪些变化真正重要，在一个称为协调的过程中，进行最少的DOM更改以保持所有内容都是最新的。  创建真正可组合UI的API  React不将应用程序中的可视元素视为一个整体块，而是鼓励您将可视元素分解为越来越小的组件。就像编程中的其他东西一样，让东西模块化、紧凑和独立是一个好主意。我们应该如何看待用户界面。React的许多核心API使创建更小的可视组件变得更容易，这些组件可以与其他可视组件结合，从而生成更大、更复杂的可视组件，就像俄罗斯的Matryoshka玩偶。这是React的主要方式之一，它简化（和更改）了我们为web应用程序构建视觉效果的想法。  完全用JavaScript定义的视觉效果  虽然这听起来疯狂得离谱，请听我说。除了使用非常奇怪的语法之外，HTML模板传统上还遇到了另一个主要问题。除了简单地显示数据之外，您可以在其中执行的各种操作都是有限的。如果您想根据特定条件选择显示哪一块UI，例如，您必须在应用程序的其他地方编写JavaScript，或者使用一些奇怪的特定于框架的模板命令使其工作。React做的很好。通过完全用JavaScript定义用户界面，您可以使用JavaScript提供的所有丰富功能在模板中执行各种操作，只受JavaScript支持的内容的限制，而不是模板框架强加的任何限制。  只是MVC架构中的V  React主要在视图层工作，它的所有担忧和关注点都围绕着您的视觉元素，并使它们保持最新。这意味着您可以自由地对MVC体系结构的M和C部分使用任何您想要的东西。这种灵活性使您能够挑选和选择您熟悉的技术，这使得React不仅对您创建的新web应用程序有用，而且对您想要增强的现有应用程序也有用，而无需删除和重构大量代码。  出自：Learning React  LearningReact  Today, going with a web app model that requires navigating between individual pages seems dated...like, really dated. Instead, modern apps tend to adhere to what is known as a Single-page app (SPA) model. This is a world where you never navigate to different pages or ever even reload a page. Instead, the different view of your app are load and unloaded into the same page itself..  As users interact with our app, we replace the contents of the dotted red region with the data and HTML that matches what the user is trying to do. The end result is a much more fluid experience. You can even use a lot of visual techniques to have your new content transition in nicely just like you might see in cool apps on your mobile device or desktop. This sort of stuff is simply not possible when navigating to different pages. All of this may sound a bit crazy if you’ve never heard of single-page apps before, but there is a very good chance you’ve run into some of them in the wild. If you’ve ever used popular web apps like Gmail, Facebook, Instagram, or Twitter, you were using a single-page app. In all those apps, the content gets dynamically displayed without requiring you to refresh or navigate to a different page.  Now, I am making these single-page apps seem really complicated. That’s not entirely the case. Thanks to a lot of great improvements in both JavaScript and a variety of third party frameworks and libraries, building single-page apps has never been easier. That doesn’t mean there is no room for improvement, though. When building single-page apps, there major issues that you’ll encounter:   1. In a single-page application, the bulk of your time will be spent keeping your data in sync with your UI. For example, if a user loads new content, do we explicitly clear out the search field? Do we keep the active tab on a navigation element still visible? Which elements do we keep on the page, and which do we destroy? These are all problems unique to single-page apps. When navigating between pages in the old model, we just assumed everything in our UI would be destroyed and just built back up again. This was never a problem. 2. Manipulating the DOM is really slow. Manually querying elements, adding children, removing subtrees, and performing other DOM operations are some of the slowest things you can do in your browser. Unfortunately, in a single-page app, you’ll be doing a lot of this. Manipulating the DOM is the primary way you are able to respond to user actions and display new content. 3. Working with HTML templates can be a pain. Navigation in a single-page app is nothing more than you dealing with fragments of HTML to represent whatever it is you wish to display. These fragments of HTML are often known as templates, and using JavaScript to manipulate them and fill them out with data gets really complicated really quickly.   Facebook (and Instagram) decided that enough is enough. Given their abundance of experience with single-page apps, they released a library called React to not only address these shortcomings, but to also change how we think about building single-page apps.  Automatic UI State Management  With single-page apps, keeping track of your UI and maintaining state is hard and very time-consuming. With React, you need to worry only about one thing: the final state your UI is in. It doesn’t matter what state your UI started out in. It doesn’t matter what series of steps your users may have taken to change the UI. All that matters is where your UI ended up. The final or end state of your UI is what matters in React. React takes care of everything else. It figures out what needs to happen to ensure your UI is represented properly, so all of that state management stuff is no longer your concern.  Lightning-fast DOM Manipulation  Because DOM modifications are really slow, you never modify the DOM directly using React. Instead, you modify an in-memory virtual DOM instead. Manipulating this virtual DOM is extremely fast, and React takes care of updating the real DOM when the time is right. It does so by comparing the changes between your virtual DOM and the real DOM, figuring out which changes actually matter, and making the least amount of DOM changes needed to keep everything up-to-date in a process called reconciliation.  APIs to Create Truly Composable UIs  Instead of treating the visual elements in your app as one monolithic chunk, React encourages you to break your visual elements into smaller and smaller components. Just like everything else in programming, it is a good idea to have things be modular, compact, and self-contained. React extends that well-established idea to how we should think about user interfaces as well. Many of React’s core APIs make it easier to create smaller visual components that can later be combined with other visual components to make larger and more complex visual components—kind of like Russian Matryoshka dolls. This is one of the major ways React simplifies (and changes) how we think about building the visuals for our web apps.  Visuals Defined Entirely in JavaScript  While this sounds ridiculously crazy and outrageous, hear me out. Besides using a really weird syntax, HTML templates traditionally suffered from another major problem. The variety of things you can do inside them other than simply displaying data is limited. If you wanted to choose which piece of UI to display based on a particular condition, for example, you had to write JavaScript somewhere else in your app or use some weird framework-specific templating command to make it work.  Just the V in an MVC Architecture  We are almost done here! React is not a full-fledged framework that has an opinion on how everything in your app should behave. Instead, React works primarily in the View layer where all of its worries and concerns revolve around your visual elements and |

****

**本科生毕业论文（设计）**

题目：小区物业管理系统前端的设计与实现

学生姓名： 魏佳

指导教师： 刘冠军

学 院： 软件学院

专 业： 软件工程

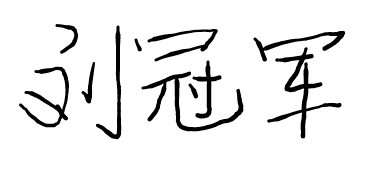
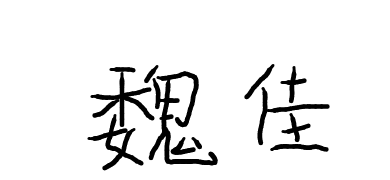
年 级： 2016级

完成日期： 2020 年 5 月 13 日

学位论文原创性声明

本人所提交的学位论文 《小区物业管理系统前端的设计与实现》 ，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

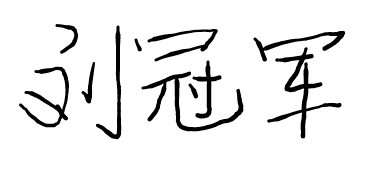
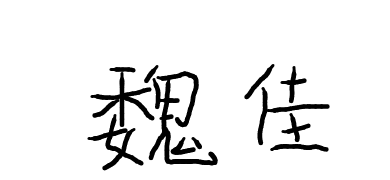
****

论文作者（签名）： 指导教师确认（签名）：

2020年 5 月 6 日 2020 年 5 月 6 日

学位论文版权使用授权书

本学位论文作者完全了解河北师范大学有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河北师范大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

****

论文作者（签名）： 指导教师（签名）：

2020年 5 月6 日 2020 年 5 月 6 日

摘 要

伴随着经济的不断发展，人们的生活水平的提高，对于舒适生活的要求也越来越高，小区物业管理的服务水平也成为了人们的条件之一。利用计算机技术，对小区信息进行整合和系统化，提高小区物业的管理水平，使得社区居民有更强的舒适度和更高的满意度。本系统是服务于物业管理人员，通过登录进入信息管理页面，对小区住户、房屋信息、收费、人事信息、故障报修等进行统一、系统化的管理，对于投诉有及时地回复和解决问题。同时也可以搜索一些信息，便于快速查找。以上模块包括了小区物业管理的各个功能，为物业人员提供了一个便捷，高效的工作环境。业主对物业管理的满意度也会提高。本系统采用React前端框架，利用AntDesign提供的组件库中的样式，实现前端的框架的构建和代码编写，并采用fetch技术实现接口功能，从而实现对数据库的添加、删除、修改、查询。

**关键词：** 物业管理 信息系统 React

目 录

[第1章 绪论 1](#_Toc37873248)

[1.1 研究背景和意义 1](#_Toc37873249)

[1.2 研究现状 1](#_Toc37873250)

[1.3 论文研究的内容和框架 2](#_Toc37873251)

[第2章 系统分析 3](#_Toc37873252)

[2.1 功能性分析 3](#_Toc37873253)

[2.1.1 系统管理员功能 3](#_Toc37873254)

[2.1.2 小区住户信息 4](#_Toc37873255)

[2.1.3 房屋信息 4](#_Toc37873256)

[2.1.4 收费信息 4](#_Toc37873257)

[2.1.5 人事信息 4](#_Toc37873258)

[2.1.6 故障投诉 4](#_Toc37873259)

[2.2 非功能性分析 4](#_Toc37873260)

[2.3 可行性分析 5](#_Toc37873261)

[2.3.1 操作可行性 5](#_Toc37873262)

[2.3.2 技术可行性 5](#_Toc37873263)

[2.3.3 经济可行性 5](#_Toc37873264)

[第3章 系统设计 6](#_Toc37873265)

[3.1 设计目标 6](#_Toc37873266)

[3.2 系统UI设计 6](#_Toc37873267)

[3.2.1 登录页面 6](#_Toc37873268)

[3.2.2 小区管理界面 7](#_Toc37873269)

[3.2.3 添加页面 8](#_Toc37873270)

[3.2.4 其他页面 8](#_Toc37873271)

[3.3 设计原则 8](#_Toc37873272)

[第4章 系统实现 9](#_Toc37873273)

[4.1 登录页面实现 9](#_Toc37873274)

[4.2 页面布局实现 9](#_Toc37873275)

[4.2.1 导航栏 10](#_Toc37873276)

[4.2.2 主要内容 11](#_Toc37873277)

[4.3 小区住户管理 11](#_Toc37873278)

[4.4 房屋信息管理 13](#_Toc37873279)

[4.5 收费管理 14](#_Toc37873280)

[4.6 故障和投诉管理 15](#_Toc37873281)

[第5章 测试和总结 16](#_Toc37873282)

[5.1 系统测试 16](#_Toc37873283)

[5.1.1 登录功能测试 16](#_Toc37873284)

[5.1.2 添加修改功能测试 16](#_Toc37873285)

[5.1.3 删除功能测试 16](#_Toc37873286)

[5.2 总结 16](#_Toc37873287)

[参考文献 17](#_Toc37873288)

[致谢 18](#_Toc37873289)

小区物业管理系统前端的设计与实现

1. 绪论

## 1.1 研究背景和意义

基于国内的经济迅速发展，住宅小区已经成为了普遍的存在，我国现在小区物业管理的水平还是远不如国外一些有经验的物业公司，由于一些现实问题，物业公司管理就会受到各方面的影响，出现管理效率低下，一些繁琐的事件，例如收取物业费，会出现因为描述不清楚，业主很难接受和自愿去缴费，会觉得费时又费力，这也就会使得物业公司的经济不景气，而且久而久之公司就会很难再运营下去，继而又会引发一些其他的矛盾。

类似于这种情况，归结为就是物业公司的管理水平低下，效率不高。利用信息管理系统不仅可以使信息变得一目了然，而且会简化一些复杂的问题。将小区内部的信息进行统一的整理，结合线上线下的交互，上面所说的问题就会显而易见，减少不必要的矛盾。据统计利用计算机技术可以大大地提高人们的工作效率，使用物业管理系统，这样小区管理也就成了非常方便高效的事情，让物业服务水平得到有效地提高。国内因为经济发展根据地区不平衡的原因，有些一线城市的物业管理发展迅速，当然随着经济的发展，全国各地的物业管理发展也会迅速提升。

## 1.2 研究现状

因为国内的小区模式是遇到问题解决问题，这就会出现大量的功能单一的管理软件，而且有的软件是对外收费的，这就会使得本来就是低微盈利的物业公司雪上加霜。而一些小区开发时间比较久远，用的系统就会像如图1-1所示的页面，整个页面比较复杂并且不清晰，信息比较杂乱，这反倒起到了反向的作用，当然现在也有一些专门为小区设定管理系统的公司，但是出于高昂的费用没有多少老式小区会使用。这就会出现一些发展不平衡的现象，一些经济较为发达的一线城市例如北京、上海、香港等地区物业管理水平是相对于其他城市发展较快的，而近几年二线城市的发展也是紧随其后，但是经济再落后的城市就会出现上面提到的问题，所以物业管理在我国还是处于一个发展阶段，物业管理系统也没有得到充分地利用。随着我国家居越来越智能化，物联网发展也越来越迅速，愈来愈多的年轻一代在装修房子的时候，同时都会考虑智能家居，远程控制家居，夏天提前将空调打开，智能扫地机器人等，有了这些智能家居就会让人们的生活幸福感极具增加，生活更加简单化，这就进一步要求物业管理方面的提升，人性化和智能化。不论是从业主的角度还是从物业人员的角度出发，物业管理发展趋势都是越简约化。



图1-1 老式小区物业管理界面

## 1.3 论文研究的内容和框架

第一章主要从不同的角度出发，讲述研究物业管理系统的背景和意义，对国内物业管理现状进行阐述。

第二章对系统分别进行功能性、非功能性和可行性分析。功能性主要是描述系统不同模块的功能，可行性分析从三个方面来说明此系统的可行性。

第三章则进行系统设计，首先是提出系统的设计目标，然后对系统的UI设计进行展示。讲述系统UI的设计风格，页面布局以及颜色搭配等。为了实现设计目标应该遵循的设计原则阐述。

第四章是在前面理论的基础上，对系统进行实践，每个模块的功能实现展示。对每个模块的实现代码进行分析。

第五章是对系统进行最终测试，然后对整个系统实现进行最后的总结。

1. 系统分析

## 2.1 功能性分析

小区系统可以分为小区住户信息、房屋信息、收费、人事、投诉和故障模块。通过对系统进行模块划分，对小区信息进行整合，清晰有效。整个用例如图2-1所示：

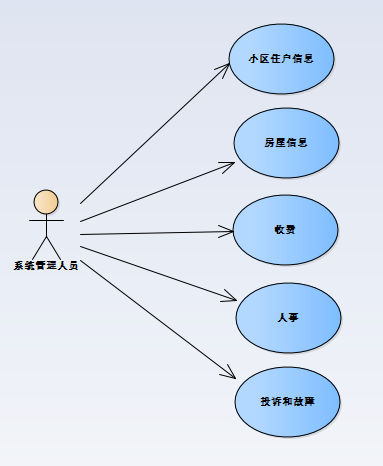


图2-1 系统用例图

### 2.1.1系统管理员功能

系统的操作者主要是小区物业管理员，通过登录系统，验证身份，成功之后可以进入关于本小区的信息管理界面，查看小区信息，并且可以对小区的业主，房屋等信息进行动态地修改，还可以按照自己的需求查看想要的信息，增加有需要或删除一些不需要的信息。整个系统是管理员所设定，所以系统的大部分功能系统管理人员是都可以参与并且执行的。

### 2.1.2小区住户信息

这个模块是管理小区业主的一些基本信息，这就也需要小区业主的一些详细信息，需要业主的配合。住户信息适当地详细，可以方便物业管理人员查看以及更新信息，给维护系统带来了极大的方便，从而使得物业服务能够更加的精确。

### 2.1.3房屋信息

房屋信息模块是对小区的房屋信息进行整合，区分一些已出售和未出售的房屋，存储房屋的基本信息，例如面积、构造等，这样有利于物业方面对房屋信息进行及时修改和维护，方便出售房屋，也方便物业公司管理。

### 2.1.4收费信息

现在人们干什么事情，一部手机就可以搞定，在网上缴费已经成为了现实，也不必物业挨家挨户地要，系统将需要缴费的单子发给业主，业主在缴费之后，后台可以看到已缴费的状态，未缴费的也可以看到，一个后台系统就可以解决，但是需要一些物业人员的人力，从而少了一些时间上的成本。

### 2.1.5人事信息

一个小区的人员不仅仅是物业和业主，也有很多其他部门的，比如保洁和保安，再这么复杂的人事关系中，人事信息模块就可以很快地整理一些人员信息，这也需要小区工作者的一些基本信息，同时确保信息的准确性。在人事管理方面也缩短了时间。

### 2.1.6故障投诉

投诉和故障是每个小区都需要具备的服务小区人们的功能，业主或者其他人员投诉或者报修，之后系统管理人员会从后台接收到投诉信息，这样以便及时处理这些故障。从而提高服务质量。

## 2.2 非功能性分析

系统管理员的工作就是对小区物业进行管理，需要对小区的各种信息以及人事非常的了解，经过专业的培训之后可以很快地熟练掌握系统的使用。系统管理人员需要输入用户名和密码，才能拥有权限进入系统，如果是外界人员或者外网则不能对系统进行访问，这样以来就保证了系统信息的安全。系统需要兼容chrome浏览器，在整个浏览器中能够实现页面的完整显示。

## 2.3 可行性分析

### 2.3.1操作可行性

如今城市化越来越突出，即便不是城市，有些乡村也是可以利用这样的管理系统来存储信息和管理信息。因为这样可以高效地解决问题。而且现在每个小区都会有物业部门和居委会，一些小区的收费问题，人员管理问题都是可以录入到信息系统，这样子小区就会保存近几年的数据，方便查看，减少收费不清楚或者人员工作不到位等问题，而且操作比较简单，了解电脑的基本信息就可以操作，所以操作也是可行的。

### 2.3.2技术可行性

我们要考虑实现此系统需要的各种技术，哪些技术可以满足提出来的要求，比如实现前端页面需要具备HTML5、CSS等关于样式技术，还要具备JavaScript关于实现一些动态功能的技术，结合React前端框架，利用React中Ant Design组件库，里面有很多的样式模板，尤其是表格，登录等组件，这样就会大大减少技术的难度，再利用其他交互工具，这样项目的实现就是可行的。

### 2.3.3经济可行性

经济可行性是我们在开发之前首先要考虑的。这个系统的开发成本是不是特别大，会不会带来很大的经济负担，而且以后能不能带来利益。信息管理系统首先对于一些开发企业都是很有必要的，其次是对信息管理的作用很大。随着互联网的不断发展，有些我们可以在电脑上办公，有事不方便了还可以在手机上办公，投入的成本也是不大的，所以经济方面也是可行的。

1. 系统设计

## 3.1 设计目标

本次设计的主要目的是服务于物业公司，物业公司是服务于小区业主，对小区内部的各种信息管理都需要有一个规范和系统性的整合。系统的UI设计也一定要给用户有交互感，将系统划分成不同的信息模块，使信息既有序的分类又高效的整合在一起，前端连接数据库，通过接口可以及时获取各个模块的信息，及时反馈业主的投诉问题，接收业主的缴费情况等信息，并且能快速对这些问题做出一些有效的措施。

物业管理系统和ERP系统极为相似，利用计算机技术管理信息系统，利用B/S模式和windows界面实现对信息管理功能，从而本系统的管理效率一定要提升，修改数据或者读取数据要统一并且精确。小区物业信息使用模块化思想，整个系统要更加的灵活。

## 3.2 系统UI设计

整个系统采用了白，黑，灰为主要基调，蓝绿色为字体的主要颜色，主要颜色是#f0f0f0、#607D8B、#001529、#f0f2f5。页面Header部分是Logo、小区名和用户信息等，左侧是导航栏，导航栏中的菜单分为一级、二级以及三级，通过样式的变化让用户有不错的交互感，了解当前所在的位置。页面右侧是主要内容部分，浅灰色为背景，用表格等形式展示一些基本信息，突出页面的主要内容。

### 3.2.1登录页面

登录页面的整个背景为灰白色，中间区域是管理员登录的入口，主要是由两个输入框组成，同样采用了主题颜色中的蓝色，会与整个系统内部颜色相契合，通过用户名和密码则可以进入到整个系统的首页。如果不成功则会提示用户密码不正确或者用户名不正确。如图3-1所示。



图3-1系统登录页面图

### 3.2.2小区管理界面

系统管理员登录成功之后，会进入此页面，也是整个系统的首页，左边就是导航部分，“小区住户信息”的字样会随之变成蓝色，业主信息就会以表格的形式呈现在右边区域的下面，内容区域的上面是展示所在的当前位置，下面则是搜索框，如图3-2所示：

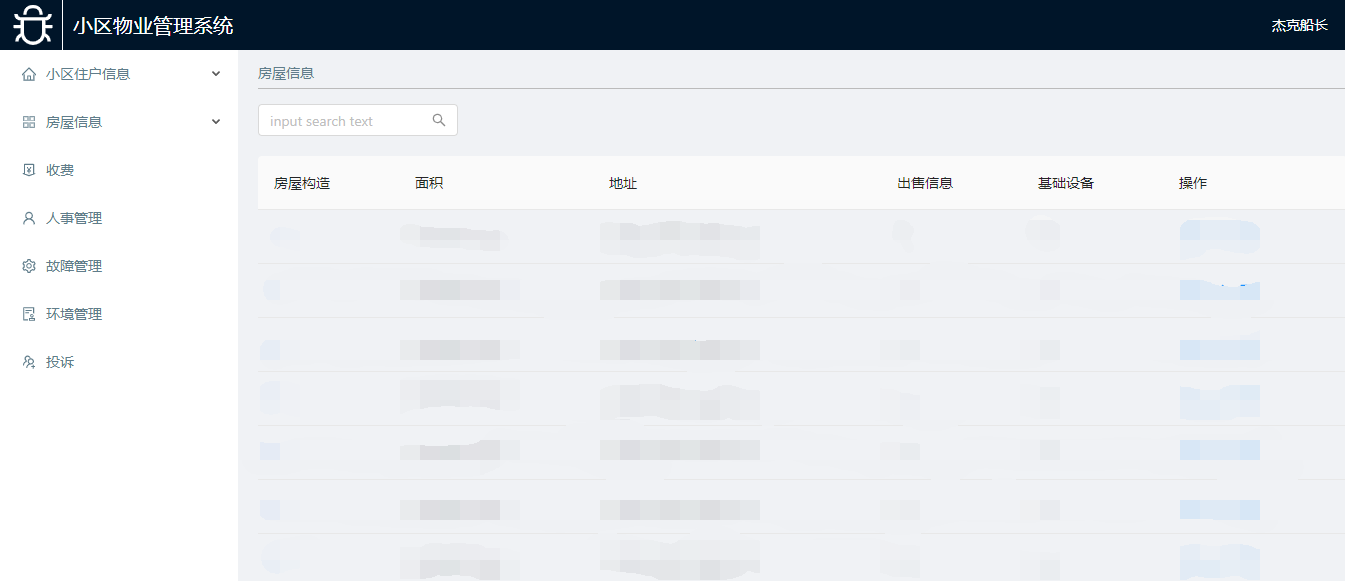


图3-2 小区住户管理页面图

### 3.2.3添加页面

页面也是由这主色调，管理员点击编辑信息时就会跳转出如图3-3所示页面，信息由表单的形式呈现，根据每个模块的不同信息，表单的内容也是会改变的，在最下面会有两个按钮“确定”和“取消”，点击“确定”会显示信息修改成功或失败，点击“取消”就会清空表单。



图3-3 添加页面图

### 3.2.4其他页面

其他页面的UI设计和上面的风格主题是一样的，在主要信息展示的时候就会像图3-2所示的模板，如果添加或者编辑信息时和图3-3所示的页面基本相同，一些基本的信息都会以表格或者表单的形式呈现，按照不同的模块或者功能会有不同的样式，所展示的信息也会有所变化。

## 3.3 设计原则

首先本系统是基于Web页面开发的，所以从视觉上一定要给人一种舒服干净的效果，其次是既然是交互型的系统，一定要考虑到系统本身和用户的互动性，对于信息管理系统，系统的数据管理一定是统一的，信息存储方式以及页面展示部分也应该遵循统一的原则，这样以来整个系统的质量就会被提高。最基本就是要遵循系统设计基本理念，有一定的安全性，系统在只能是公司内部人员可以访问，保证信息数据的保密性，系统设计要按照规范的编码协议保证系统的稳定性和可维护性。也就是一定要具有稳定性、安全性、可维护性和规范性等一些系统最基本的设计原则。能够最大的限度实现所需要实现的目标。

1. 系统实现

## 4.1 登录页面实现

系统管理人员需要通过填写账户和密码进入系统主页，系统需要检验账户名和密码是否准确，当系统管理人员输入不正确的用户名或者密码的时候，页面会提示用户输入的用户名或者密码不正确，需要改正，直至输入正确的用户名以及密码才能通过验证，进入系统的主页。这部分的实现步骤是先获取输入框中的数据，然后利用fetch将数据传到数据库中，与数据库中的用户名和密码进行匹配，匹配正确或者错误都会从后台发出信号，只不过正确信号和错误信号是不一样的，前端收到这个信号之后，再根据信号的不同前端页面就会传递出不同的提示。与后端的数据交互使用的是JSON格式。

## 4.2 页面布局实现

整个页面布局采用的是顶部，左边导航栏，右边内容页的分布格局，在App.js中渲染，实现布局代码如下：

<Layout>

{/\* Header \*/}

<Header className="header" style={{height:50,paddingLeft:5,paddingRight:5}}>

</Header>

{/\* Middle \*/}

<Layout>

{/\* 导航栏 \*/}

<Sider>

</Sider>

{/\* 右边组件路由 \*/}

<Content style={{ marginLeft: 60, minHeight: 280}}>

</Content>

</Layout>

</Layout>

布局效果大概如图4-1所示：

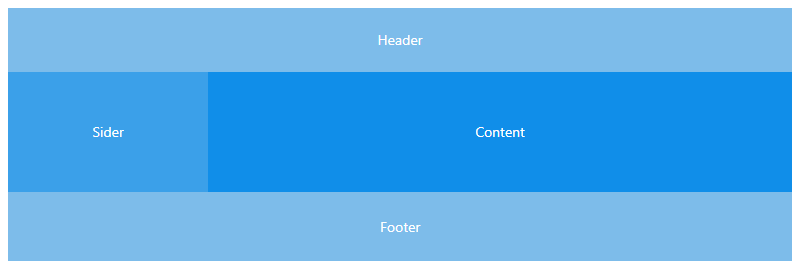


图4-1 页面布局图

### 4.2.1导航栏

整个导航栏部分在整套系统中是不改变的，所以在系统的页面中，左边导航栏是在App.js中实现，利用Sider标签和Menu标签进行导航框架的编写，Ant Design组件中Menu标签的子标签SubMenu和Menu.Item标签可以对导航栏的样式进行排序，左边导航栏用的是下拉功能即点击一级菜单会弹出来二级菜单，点击二级菜单下拉三级菜单。如果没有下拉三角形的标志就代表此菜单只有一级菜单。实现效果如图4-2所示：



图4-2 导航栏截图

### 4.2.2主要内容

页面跳转实现是用React中的路由Route和Link，在App.js中对整个页面分成不同的部分，页面顶部和导航栏部分不变，只变换内容部分，这就需要路由和组件的支持，将主要内容部分也按照模块的不同分成了不同的组件，如图表4-3所示：

表4-3 模块分类组件表

|  |  |
| --- | --- |
| **模块名** | **组件名** |
| 小区住户信息 | Residents.js |
| 房屋信息 | Houses.js |
| 收费模块 | Charge.js |
| 人事管理模块 | Personnel.js |
| 故障管理模块 | Fault.js |
| 投诉管理模块 | Complaint.js |

在跳转页面的入口处加入Link标签，代码如下：

<Link to="/residents">一单元</Link>

之后，在Content部分填入需要渲染的组件名称：

<Content style={{ marginLeft: 60, minHeight: 280}}>

<Route path="/residents" component={Resident}/>

</Content>

在整个框架中，渲染组件需要在App.js中实现。

## 4.3 小区住户管理

小区住户管理模块中，是把整个模块封装到一个组件当中，渲染时只需在需要用到的地方引入组件即可。系统管理者可以在这个页面看到小区住户的姓名、电话号码、地址和合同链接等信息，如图4-4所示：

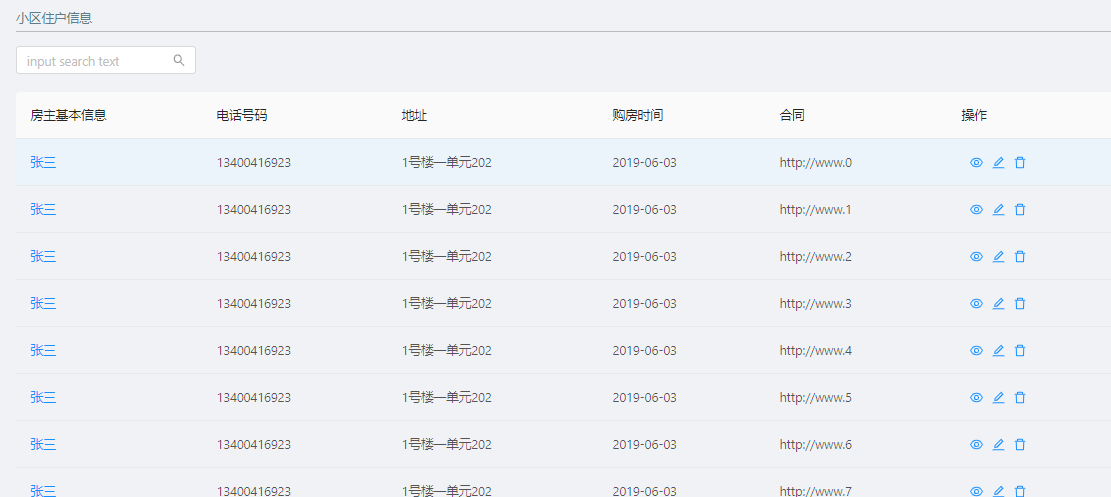


图4-4 小区住户管理页面图

页面的表格部分使用Table标签实现，表头是由数组变量来固定的，将表头的数组变量通过Table标签的属性传给Table。信息列表中的数据需要从数据库中获取，但是数据库中的数据并不是一条两条的，有的是几百条甚至上千条都有，所以不可能一个一个的获取数据，对数据展示的实现，利用了for循环，将从数据库拿到的数据push到已建数组中，“dataSourse”属性接收此数据。

系统管理员也可以对信息进行增、删、改、查，当点击操作中的“眼睛”图标时可以进入小区住户的基本信息详情页面，如图4-5所示：



图4-5 详细信息图

对小区住户信息的添加和修改，添加界面如图4-6所示：

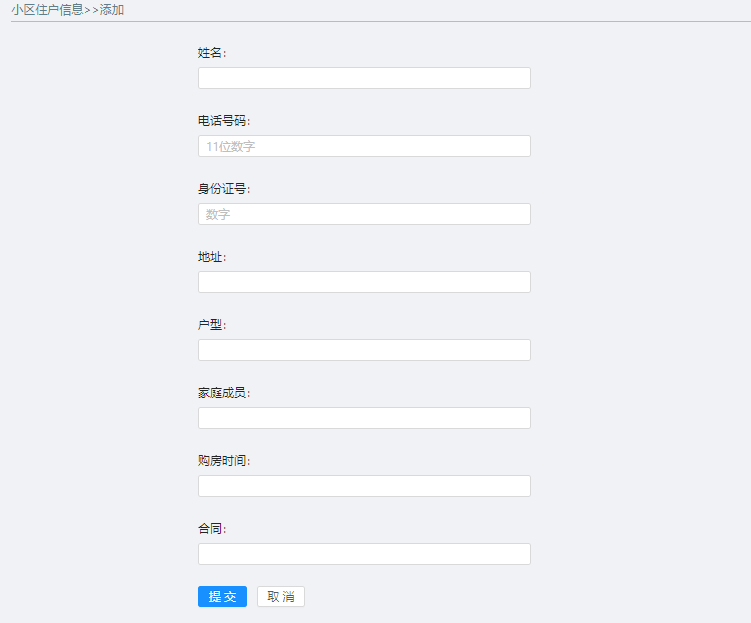


图4-6 添加小区住户界面图

添加和修改功能首先是需要从前端获取到表单数据，然后将获取的数据通过接口传到数据库中，实现将数据库的修改才是真正意义上实现对信息的修改，如果只改变页面当中的数据，数据库的信息会和前端数据不一致，造成不必要的问题。

## 4.4 房屋信息管理

房屋信息管理是对小区中的房屋构造，面积，地址等信息进行存储，及时更新房屋信息，以便能够获取最新的信息，管理系统提供了增添、查看、删除功能，页面上方的搜索框是用来提供系统管理员搜索功能，根据关键字可以从数据库中找到相应的数据并渲染到页面中，如果系统管理员修改或添加某一项数据，数据库中的数据也会跟着改变。具体页面效果如图4-7和图4-8所示：



图4-7 房屋信息管理界面图



图4-8 添加房屋信息界面图

## 4.5 收费管理

系统管理员首先需要对收费项目进行一一的录入，如图5-5-1所示，添加的信息包含物业费、电费、暖气费、时间等，同时添加之后就可以从页面中看到，系统管理员也可以对每一项数据进行修改、查询和删除操作。没有缴费之前的状态是未缴费，当小区业主缴费时后，缴费状态就会变成已支付。如图4-9所示。

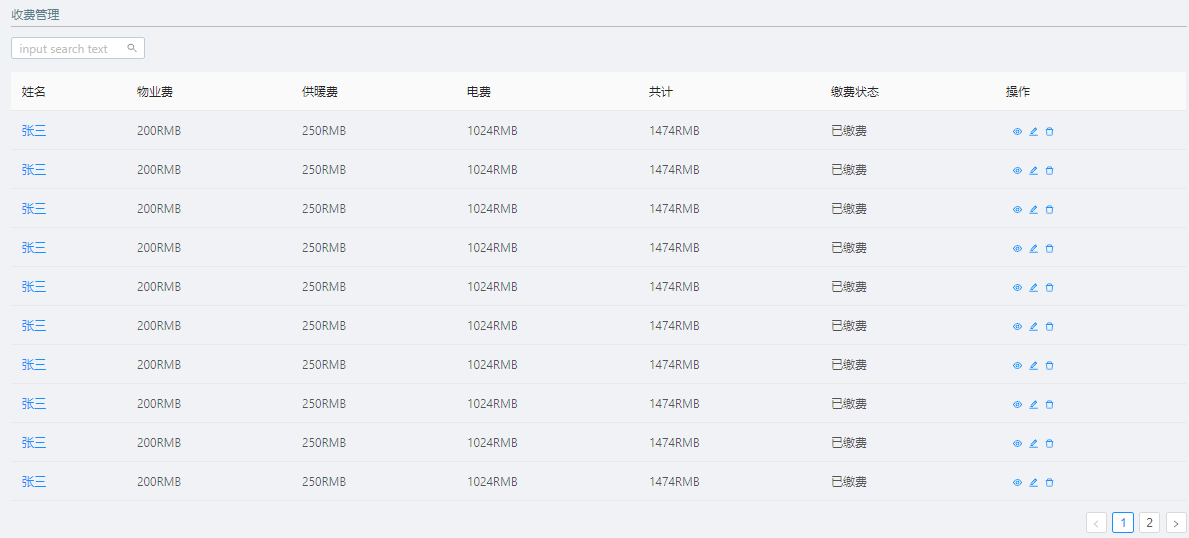


图4-9 收费页面图

## 4.6 故障和投诉管理

故障和投诉管理是提升小区物业服务水平的方法之一，这样可以快速接收到业主的问题，并及时反馈给业主处理方法和处理结果，问题应该具体到个人，收到故障问题也应该及时找人去上门维修，这样就会有效地提高问题解决的速度和精准度。具体页面如图4-10所示：



图4-10 故障投诉页面图

1. 测试和总结

## 5.1 系统测试

### 5.1.1登录功能测试

用户输入正确的用户名和密码，可以成功进入系统首页。

用户输入错误的用户名会弹出“用户名不正确”的提示框。

用户输入错误的密码会弹出“密码不正确”的提示框。

### 5.1.2添加修改功能测试

用户点击添加信息可以进入添加信息界面，对信息进行添加，添加完成而且符合正确的格式之后，点击“添加”就可以将添加的数据保存到数据库，并且页面就会渲染出新的数据信息，点击“取消”则信息不增加。

用户点击修改进入修改信息页面，将需要改变的信息修正，且需要符合正确的输入格式，点击“完成”数据库中的信息就会改变，页面的信息也会随着改变，点击“取消”信息不会修改。

### 5.1.3删除功能测试

在信息管理模块中如果对整条信息进行删除，点击垃圾桶的小图标，那么这条信息就会从界面当中消失，并且数据库有关这条信息也会被删除。这个操作是不可逆的，当然也可以进行添加。

## 5.2 总结

在这次小区物业管理系统中我是参与前端的设计与实现。从开始对整个项目进行需求分析，再对其进行模块的划分，设计UI，整理和分析，复习相关的技术知识，搭建整个前端框架，再到逐步实现每个模块的功能。重新将前端需要的知识串联起来，包括CSS、HTML、JavaScript等基础知识，还要需要有良好的页面结构，先分析布局，划分框架，再规划结构，编写代码。之后利用XML、JSON获取数据，将这些数据库中的数据实时渲染到页面中。一个合格的前端开发者代码的质量也是重要考虑的问题，这次项目的代码还是需要不断的完善。

由于一些客观原因的限制，这次毕业设计项目的功能点还是有些少，每个模块的功能不是很完善，对于物业管理系统的创新点也比较少，而且对于缴费和故障报修模块的线上和线下的交互没有完成，对于手机端的上报和实行线上缴费功能没有进一步的实现，但是在之后，我会继续完善此系统的功能，对系统还没有完成的功能继续实现。

参考文献

[1]郭保卫，苑丰.基于Web的小区物业管理系统[J].信息与电脑（理论版），2018年.

[2]徐连霞，傅伟.小区物业管理系统的设计与实现[J].现代营销（信息版），2019年.

[3]赵强，范丰雷，孙晓霞.智能化住宅小区的物业管理[J].山东行政学院山东省经济管理干部学院学报，2004年5期.

[4]周婧.社区物业管理系统的设计与实现[D].东北大学，2015年.

[5]张进.物业管理智慧化发展的标准化和系信息化建设[J].河南科技，2019年.

[6]周晓雨.智慧物业管理系统的发展研究[J].现代物业（中旬版），2018年.

[7]王鹏.智能小区物业管理系统设计与实现[D].江西财经大学,2019年.

致 谢

路漫漫其修远兮，吾将上下而求索，四年的大学生活转眼就要结束了，求学岱下，筑梦山农，始于2016初秋，终于2020盛夏。四年光阴如烟火，满眼繁华，点滴生活，历历在目。在这所学校留下的是青春和沉甸甸的收获。我虽平淡无奇，但仍心怀感激。

首先，诚挚的感谢刘冠军老师对我的毕业论文和毕业设计的指导，他在忙碌的教学工作中挤出时间审查和修改我的论文。从选题到修改，再到最后定稿，老师就如朋友一般给予了我最大的帮助。其次我要感谢从入学到现在的每一位专业课老师，你们的严谨细致、一丝不苟的作风一向是我工作、学习中的榜样，也因为你们我的专业得以丰富扎实。感谢学院给了我这四年学习的机会，让我不断地充实自己，也让我学会不断地提高自己才会看到不一样的风景，用知识填满自己。我将怀着感激之情，所学所想继续前行。

2020年是特殊的一年，这个夏天也是特别的毕业季，冬季的黑暗尽管漫长，但永远挡不住黎明的曙光。纵然山河有恙，不敌世间盛情。

在即将毕业之即，我真心祝愿：老师们身体健康、事业有成。愿河北师范大学年年桃李，岁岁芬芳，我们不说再见。愿各位同窗以梦为马，不负韶华。