# 摘 要

在当今中国社会经济发展中，农业、农村和农民问题一直备受瞩目。我国农村仍然面临信息不对称和商品供应链短缺等问题，制约了农业发展和农民收入的提升。结合当前三农政策背景，为农村居民提供综合服务，促进农村经济发展。通过微信小程序平台和Vue3框架构建，以提升农产品购物、社区交流和信息分享的便捷性。

本系统通过“三农E家”微信小程序，为农村居民提供了三农政策和农业科技的新闻、互助社区和农资商城于一体的服务平台，旨在解决这些问题。同时，系统还通过后台管理系统“三农E家”进行用户、商品和订单的全面管理，以确保系统高效运行。在实现方面，前台采用微信小程序和Vant-Weapp组件库，后台管理系统基于Vue3和TypeScript，采用Element-Plus组件库，均使用Node.js提供后端接口服务。系统强调社区交流模块，为农村居民提供更广泛的社交和信息分享机会，增进彼此了解。

本系统的设计与实现，不仅符合当下社会对农业发展的需求，也为农村居民提供更多社交和信息互动机会，具有显著的社会意义。

**关键词：**三农政策、微信小程序、Vue3、Node.js、农产品购物、社区交流

# ABSTRACT

In the current socio-economic development of China, issues related to agriculture, rural areas, and farmers have always been under close scrutiny. Rural areas in China still face challenges such as information asymmetry and shortages in the supply chain, which hinder the development of agriculture and the improvement of farmers' income. Aligning with the current background of the "Three Rural" policy, this system provides comprehensive services for rural residents to promote rural economic development. Built on the WeChat Mini Program platform and Vue3 framework, the system aims to enhance the convenience of agricultural product shopping, community interaction, and information sharing.

The "Three Rural E-house" WeChat Mini Program, part of this system, offers a platform that combines news, mutual assistance communities, and an agricultural supplies mall to address these issues and deliver the benefits of the "Three Rural" policy and agricultural technology to rural residents. Simultaneously, the system ensures comprehensive management of users, products, and orders through the "Three Rural E-house" backend management system, ensuring the efficient operation of the system. In terms of implementation, the front end adopts the WeChat Mini Program and the Vant-Weapp component library, while the backend management system is based on Vue3 and TypeScript, utilizing the Element-Plus component library and Node.js to provide backend interface services. The system emphasizes the community interaction module, providing broader opportunities for socializing and information sharing for rural residents, fostering mutual understanding.

The design and implementation of this system not only meet the current societal demands for agricultural development but also provide more social and interactive opportunities for rural residents, carrying significant social significance.

**KEY WORDS:** Three Rural Policy, WeChat Mini Program, Vue3, Node.js, agricultural product shopping, community interaction.

目 录

[摘 要 I](#_Toc5626)

[ABSTRACT II](#_Toc16840)

[第1章 前言 1](#_Toc19772)

[第2章 系统需求分析 2](#_Toc30779)

[2.1. 角色分析 2](#_Toc32757)

[2.2. 功能分析 2](#_Toc21267)

[2.3. 可行性分析 4](#_Toc10950)

[2.3.1. 应用可行性 4](#_Toc30066)

[2.3.2. 技术可行性 4](#_Toc21893)

[2.3.3. 经济可行性 4](#_Toc32440)

[第3章 开发工具及关键技术 6](#_Toc18892)

[3.1. 开发工具 6](#_Toc27580)

[3.1.1. H Builder X 6](#_Toc14039)

[3.1.2. Visual Studio Code 6](#_Toc21362)

[3.1.3. 微信小程序 6](#_Toc4561)

[3.2. 关键技术 6](#_Toc12372)

[3.2.1. Uni-app 6](#_Toc17754)

[3.2.2. Django 7](#_Toc31820)

[3.2.3. ORM框架 7](#_Toc24149)

[第4章 系统设计 9](#_Toc10444)

[4.1. 系统架构及功能设计 9](#_Toc10275)

[4.2. 系统流程图 10](#_Toc15513)

[4.2.1. 系统登录流程图 10](#_Toc31668)

[4.2.2. 系统注册流程图 11](#_Toc27937)

[4.2.3. 来访登记流程图 12](#_Toc24591)

[4.2.4. 处理预约记录流程图 13](#_Toc30547)

[4.3. 系统数据库设计 14](#_Toc20555)

[4.3.1. E-R图设计 14](#_Toc23627)

[4.3.2. 数据库逻辑结构设计 15](#_Toc23045)

[第5章 系统实现 21](#_Toc8412)

[5.1. 用户端功能设计 21](#_Toc24339)

[5.1.1. 注册功能实现 21](#_Toc30331)

[5.1.2. 登录功能实现 23](#_Toc15818)

[5.1.3. 修改密码功能实现 25](#_Toc29747)

[5.1.4. 首页功能实现 27](#_Toc10812)

[5.1.5. 导航栏功能实现 28](#_Toc1510)

[5.1.6. 数据功能实现 33](#_Toc23050)

[5.1.7. 图表功能实现 34](#_Toc10342)

[5.1.8. 个人中心功能实现 35](#_Toc7272)

[5.1.9. 个人资料功能实现 36](#_Toc9428)

[5.1.10. 访客记录功能实现 37](#_Toc6443)

[5.1.11. 绑定老人功能实现 37](#_Toc8839)

[5.2. 后台管理系统功能设计 38](#_Toc26044)

[5.2.1. 管理系统页面 38](#_Toc1775)

[5.2.2. 账号管理模块 39](#_Toc2260)

[5.2.3. 员工管理模块 39](#_Toc8411)

[5.2.4. 老人管理模块 40](#_Toc21779)

[5.2.5. 房间管理模块 41](#_Toc820)

[5.2.6. 监护人管理模块 41](#_Toc6495)

[5.2.7. 每日数据管理模块 42](#_Toc12901)

[5.2.8. 程序管理模块 42](#_Toc23853)

[第6章 系统测试 45](#_Toc8695)

[6.1. 系统配置 45](#_Toc20514)

[6.2. 功能测试 45](#_Toc11318)

[6.2.1. 小程序端注册功能测试 45](#_Toc4016)

[6.2.2. 小程序端登录功能测试 46](#_Toc15318)

[6.2.3. 小程序端数据展示功能测试 47](#_Toc14997)

[6.2.4. 小程序端提交预约来访功能测试 47](#_Toc24566)

[6.2.5. 后台系统老人模块搜索功能测试 48](#_Toc18796)

[6.2.6. 后台系统来访管理是否预约功能测试 48](#_Toc1260)

[6.3. 测试总结 49](#_Toc7346)

[第7章 总结与展望 50](#_Toc8384)

[7.1. 总结 50](#_Toc29667)

[7.2. 展望 50](#_Toc22599)

[参考文献 51](#_Toc24226)

# 前言

在当今中国社会，农业、农村和农民问题一直备受瞩目。随着国家三农政策的深入实施和农业科技的不断发展，我们深刻认识到农村信息不对称和商品供应链短缺等问题制约了农业的发展和农民收入的提升。为了促进农业经济的繁荣，提高农村居民的生活水平，我们设计并实现了三农E家系统。本系统以中国农村的实际情况为背景，以微信小程序和Vue3为技术支撑，致力于为用户提供便捷的购物体验、实时的社区交流和全面的信息分享。通过系统的运作，我们期望为农村居民搭建一个全方位服务平台，实现农业与信息技术的深度融合，助力农村的经济社会发展。

该系统以微信小程序作为前端应用，通过vant-weapp框架和node.js后端技术，实现了新闻浏览、互助社区交流、农资商城购物、购物车管理、用户中心等一系列核心功能。同时，后台管理系统采用vue3、ts、element-plus和node.js技术栈，为管理员提供了用户管理、菜单管理、权限管理、角色管理、商品属性管理、订单管理等全面的后台支持。本文主要介绍了系统需求分析，系统开发工具及关键技术，系统的设计、实现过程及测试结果。

# 系统需求分析

## 角色分析

农民（用户）：登录注册小程序，利用微信小程序进行农产品购物、社区交流，获取三农政策新闻和农业科技信息等操作。

管理员：系统的管理者，拥有三农E家后台系统的所有权限，进行用户管理、商品管理、订单管理，维护系统的正常运行，也负责上架商品和订单发货。

商品管理员：系统商品的管理者，拥有三农E家后台系统的商品管理和订单管理，进行商品管理、订单管理，可以进行平台属性管理，spu管理，sku管理，负责上架商品和订单发货。

## 功能分析

用户的小程序端主要功能是农民进行对信息的了解和交流以及对农资产品的购买。农民第一次进入小程序要进行注册，注册后自动登录，退出登录后要重新登录。登录小程序后，可以查看首页的三农政策和农业科技的相关新闻，可以在互助社区发表问题寻求解决，也可以点赞评论帮助别人。可以在农资商城便捷的购买所需的农资产品，如假如购物车的商品，可在购物车里查看。可以在我的，对个人资料进行填写，例如地址管理，手机号，头像，昵称等等。而没有注册登录的人只能在首页和互助社区且不能使用点赞评论功能。用户功能用例如图 2-1所示。

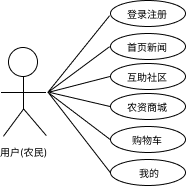


图 2-1 用户功能用例图

管理员端运行于后台管理系统，包含用户管理，角色管理，权限管理，菜单管理，平台属性管理，销售属性管理，spu管理，sku管理，订单管理，管理员功能用例图如图 2-2所示。

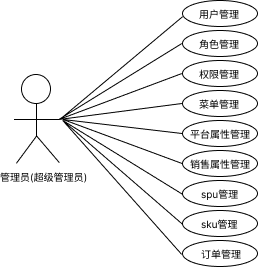


图 2-2 管理员功能用例图

商品管理员权限低于管理员，能访问除账号管理外的其他模块，商品管理员可以进行台属性管理，销售属性管理，spu管理，sku管理，订单管理，对应的信息将展示于用户所看到的小程序端，便于用户对商品的购买与发货更新。商品管理员功能用例图如图 2-3所示。

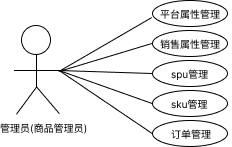


图 2-3 商品管理员功能用例图

## 

## 可行性分析

为确保三农E家系统的设计和实施能够成功，我们进行了全面深入的可行性分析，涵盖了应用可行性、技术可行性和经济可行性三个关键方面。这一综合性的分析不仅有助于发现系统可能面临的挑战，还能够明确其优势和潜在问题，为系统的进一步发展提供科学依据。

### 应用可行性

在应用可行性方面，三农E家系统旨在满足农村居民的购物需求、社区交流和信息分享。考虑到当前社会对农业发展和农民问题的高度关注，系统在应用方面呈现出了极大的市场需求。采用微信小程序平台，系统不仅提供了丰富多彩的功能，还通过互联网的便捷性为农村居民提供了全新的服务体验。在中国农村庞大人口基数的支撑下，系统在应用层面展现出了极大的潜力和可行性，为促进农业经济的发展创造了有利条件

### 技术可行性

技术可行性主要关注系统的架构和开发工具的选择。三农E家系统选用微信小程序+Vant-Weapp前端，Vue3+Element-Plus后台，Node.js提供后端接口服务。这一技术组合保证了系统的灵活性和高效性。微信小程序和Vue3等技术具有广泛的开发者基础和强大的社区支持，Vant-Weapp和Element-Plus等组件库提供了丰富的开发资源。采用Node.js作为后端服务，保障系统的稳定性。综合而言，系统在技术可行性方面拥有现代、可靠的技术基础，为顺利实施提供了坚实支持。

### 经济可行性

经济可行性主要考察项目的成本与效益关系。系统的制作过程分为前期调研、中期开发和后期测试。前期调研成本较低，中期开发和后期测试使用的工具，如Vant-Weapp、Element-Plus等，均为免费开源，降低了制作过程中的经济负担。系统的经济成本相对较低，与当前市场的投资回报比相匹配，为系统的可持续发展提供了经济基础。

# 开发工具及关键技术

## 开发工具

### 微信开发者工具

微信开发者工具是小程序开发的官方集成开发环境，具有强大的调试和实时预览功能。该工具提供了便捷的开发和调试环境，加速了小程序前端的开发过程。通过微信开发者工具，我们能够高效地构建、调试和发布微信小程序，确保用户能够顺畅地使用系统。

### Visual Studio Code

Visual Studio Code是一款轻量级、强大且免费的集成开发环境。作为一种通用编辑器，Visual Studio Code支持多种编程语言，拥有丰富的插件生态系统。在系统的开发中，Visual Studio Code为我们提供了高效的代码编辑、调试和版本控制的功能，提升了前后端开发的整体效率。

## 关键技术

### 微信小程序

微信小程序是一种轻量级的应用程序，无需下载安装即可使用。通过微信小程序，我们实现了系统的前端部分，提供了用户友好的界面和良好的响应速度。微信小程序的广泛普及也确保了系统的覆盖面和可用性。

### Vue3

Vue3是一款流行的JavaScript框架，具有响应性、灵活性和高效性。在系统的后台管理部分，我们使用Vue3框架，通过组件化的方式构建了清晰结构的用户界面，提高了系统的可维护性和开发效率。。

### Node.js + SQL

Node.js作为后端技术，结合SQL数据库，为系统提供了高性能、非阻塞I/O和强大的数据存储与管理能力。Node.js处理后端逻辑，而SQL数据库用于结构化的数据存储和高效的查询操作，共同构建了系统稳定、高效的后端基础。这种组合适用于对性能和数据操作有较高要求的应用场景，为用户提供了可靠的服务。

# 系统设计

## 系统架构及功能设计

本三农E家系统充分运用前后端分离的理念，借助微信小程序实现前端，旨在提供出色的用户体验和高效的开发效率。后端则构建于Node.js之上，采用Express框架，为管理人员提供了强大而灵活的后台管理系统。

系统核心体现为两大端面：管理员端和用户端。管理员端专为系统管理者量身定制，主要涵盖用户、商品和订单管理等关键业务。用户端则以微信小程序为媒介，提供直观而简便的操作体验，使用户能轻松浏览、查询以及使用系统中的各项功能和服务。每个模块的功能各不相同，系统架构图和功能图如图 4-1和图 4-2（下页）所示。

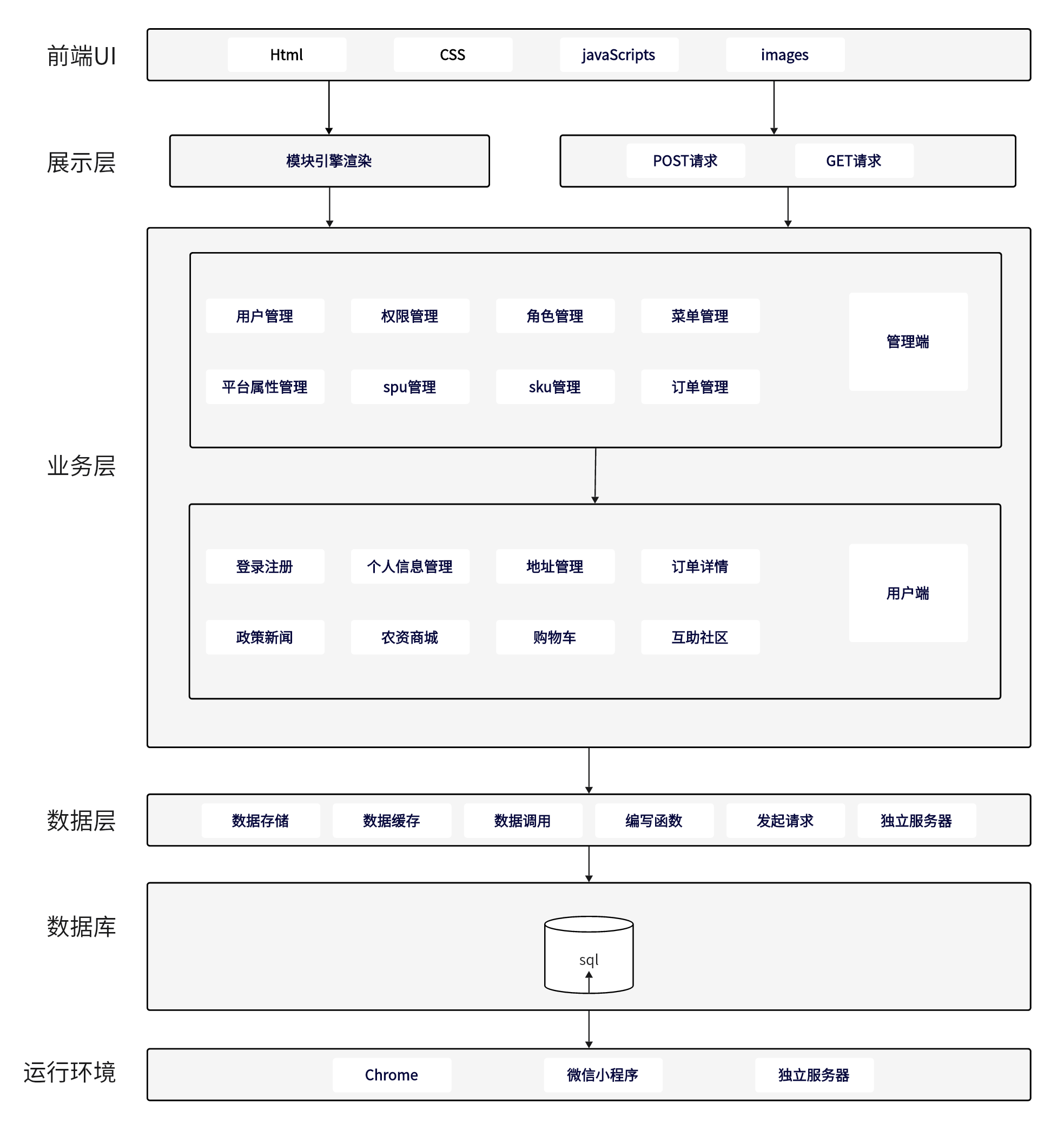


图 4-1 系统结构图

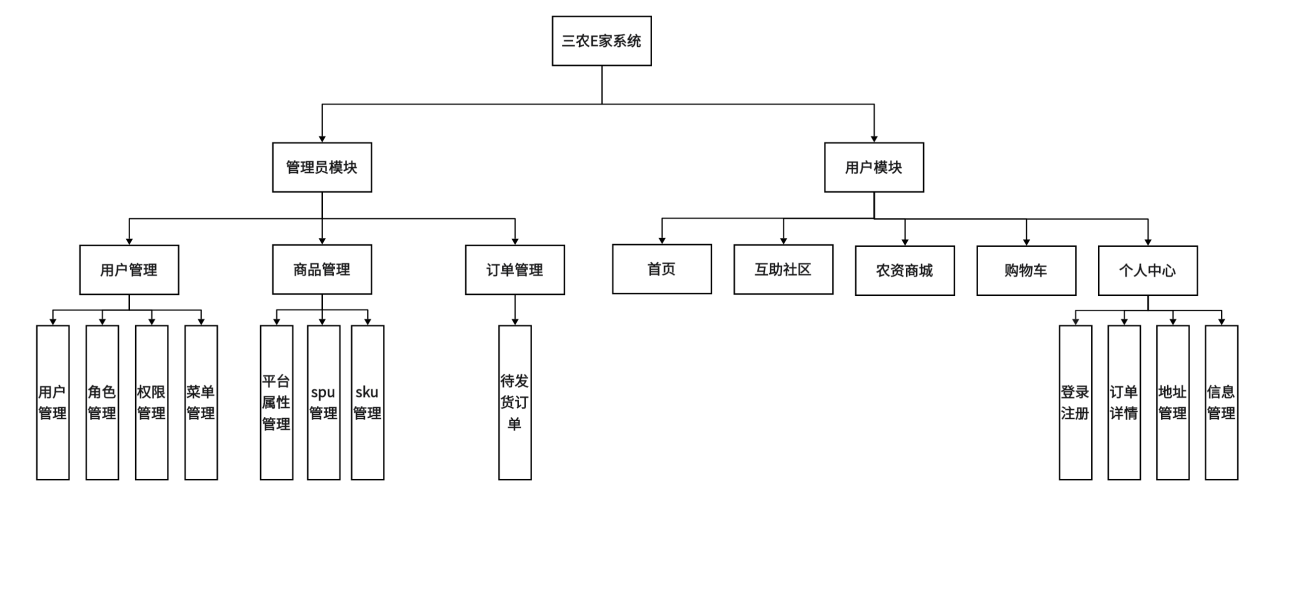


图 4-2 系统功能图

## 系统流程图

### 系统登录流程图

进入小程序后，游客可访问页面有首页以及首页中的各个跳转页面，互助社区的展示页，农资商城，购物车，订单详情，信息更改和互助社区的详情页，只有用户登录后才能访问，用户输入账号密码后会先判断是否为空，再判断用户名密码是否正确，具体登录流程图如图 4-3所示（下页）。

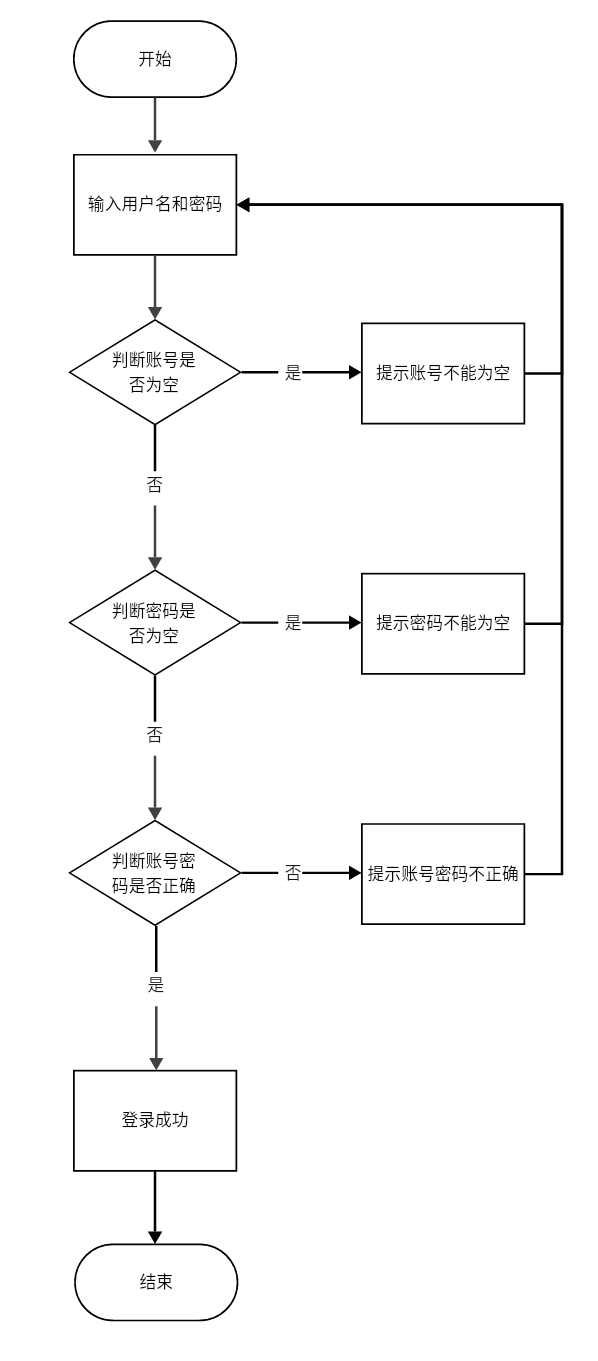


图 4-3登录流程图

### 系统注册流程图

用户要登录的话要先注册账号，通注册的账号密码才能进行登录，详细的注册过程如图 4-4所示（下页）。

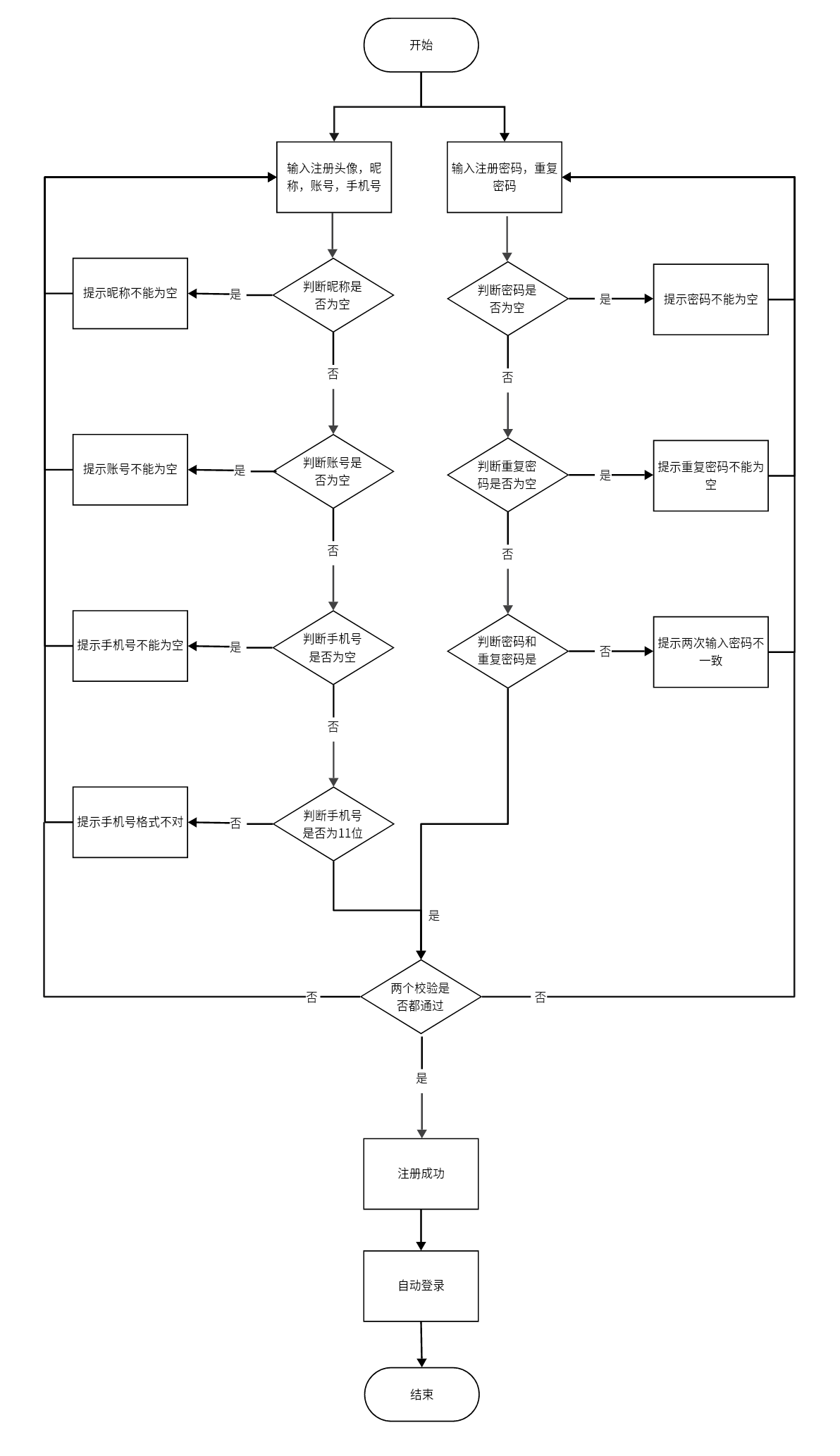


图 4-4 注册流程图

### 用户购买流程图

三农E家系统购买有两种方式：一种是直接购买，一种是购物车购买，购买后判断是否付款，付款后再到商家发货，用户收货后可以进行对商品的评论。具体流程图如图 4-5所示。

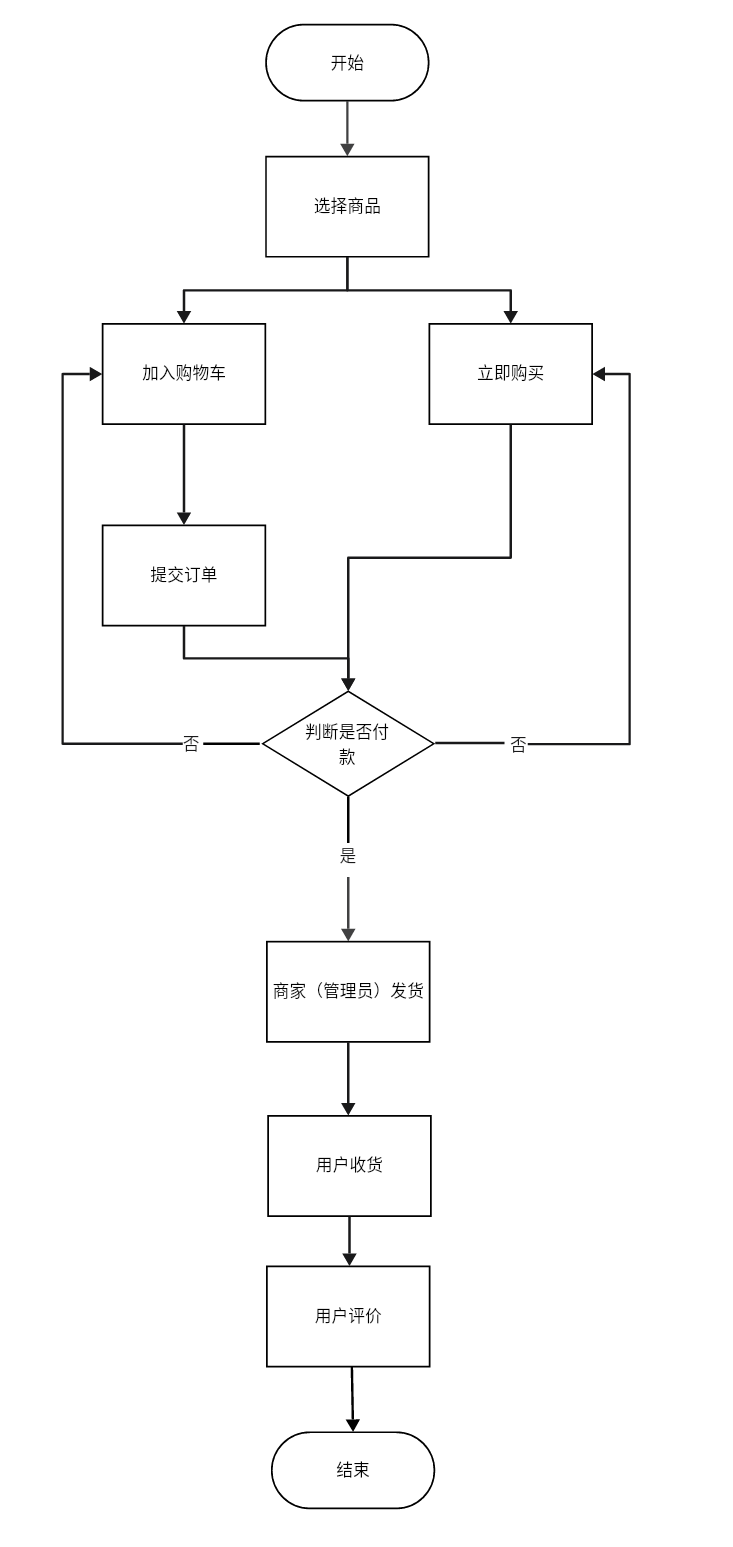


图 4-5 用户购买流程图

## 系统数据库设计

### E-R图设计

结合上述系统功能结构和功能流程图，对该智慧养老系统各角色功能及关系进行剖析，最终主要业务关系E-R图设计如图 4-7所示。

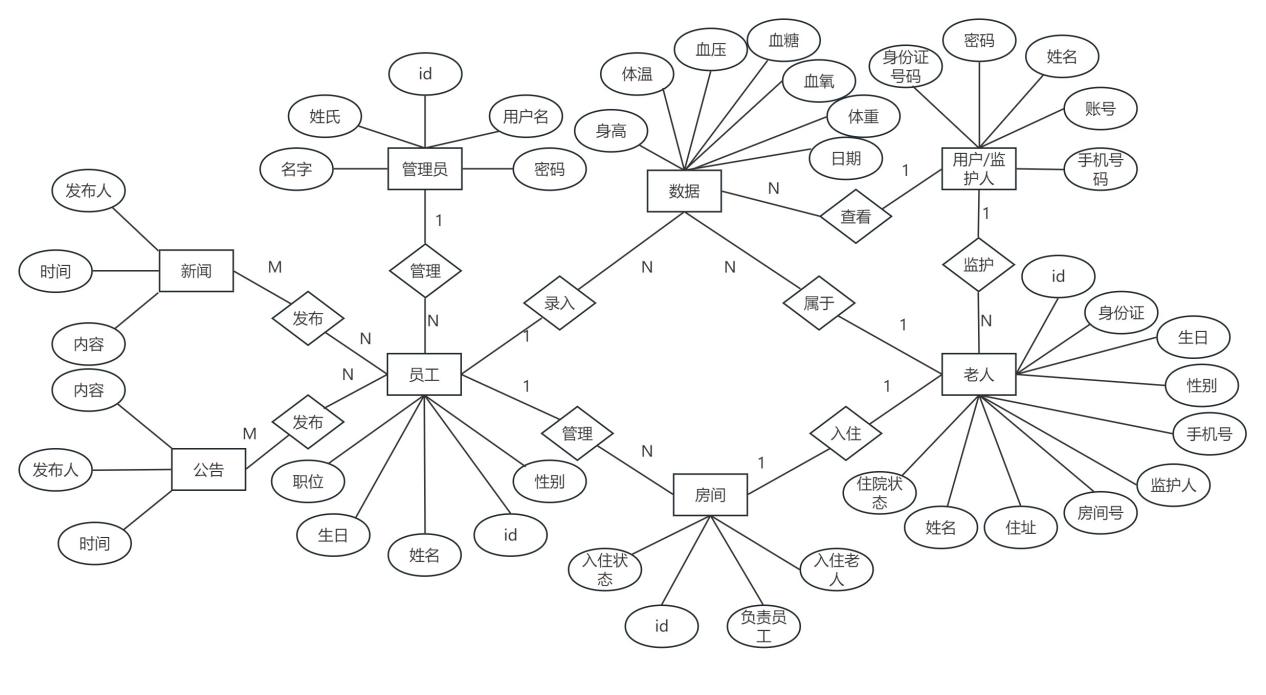


图 4-7 业务关系E-R图

### 数据库逻辑结构设计

根据上述所设计的E-R图以及系统的功能分析，设计出Django模型类，ORM框架所生成数据表如下所示，系统一共涉及到了9个数据表，具体情况如表 4-1所示。

表 4-1 数据表

|  |  |  |
| --- | --- | --- |
| 序号 | 集合名称 | 说明 |
| 1 | auth\_user | 管理员表 |
| 2 | employee\_emp | 员工表 |
| 3 | aged\_aged | 老人表 |
| 4 | guardian\_guardian | 监护人表 |
| 5 | employee\_room | 房间表 |
| 6 | data\_data | 每日数据表 |
| 7 | all\_announcement | 公告表 |
| 8 | visit\_visit | 访客表 |
| 9 | management\_book | 预约表 |
| 10 | news\_news | 新闻表 |

以下为各表的具体情况：

管理员表（auth\_user），如下方表 4-2所示。

表 4-2 auth\_user管理员表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 管理员id |
| username | varchar | 150 | NOT NULL | 否 | UNIQUE | 用户名 |
| password | varchar | 128 | NOT NULL | 否 |  | 密码 |
| first\_name | varchar | 100 | NOT NULL | 否 |  | 名 |
| last\_name | Int |  | NOT NULL | 否 |  | 姓 |
| email | varchar | 254 | NOT NULL | 否 |  | 邮箱 |
| is\_superuser | boolean |  | NOT NULL | 否 |  | 是否为管理员 |
| is\_staff | boolean |  | NOT NULL |  |  | 是否为员工 |
| is\_active | boolean |  | NOT NULL |  |  | 用户是否激活 |
| date\_joined | date time |  | NOT NULL |  |  | 创建时间 |
| last\_login | varchar | 150 | NULL |  |  | 上次登录时间 |

员工表（employee\_emp），如下方表 4-3所示。

表 4-3 employee\_emp表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| user\_ptr\_id | integer |  | NOT NULL | 是 | UNIQUE | 员工id |
| sex | varchar | 2 | NOT NULL | 否 |  | 性别 |
| birthday | date |  | NULL | 否 |  | 生日 |
| telephone | varchar | 20 | NOT NULL | 否 |  | 手机号码 |
| address | varchar | 50 | NOT NULL | 否 |  | 住址 |
| position\_id | varchar | 16 | NULL | 否 |  | 职位id |

老人表（aged\_aged），如下方表 4-4所示。

表 4-4 aged\_aged表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 老人id |
| name | varchar | 50 | NOT NULL | 否 |  | 姓名 |
| sex | varchar | 2 | NOT NULL | 否 |  | 性别 |
| idcard | varchar | 23 | NOT NULL | 否 | UNIQUE | 身份证号码 |
| birthday | date |  | NOT NULL | 否 |  | 生日 |
| telephone | varchar | 20 | NOT NULL | 否 | UNIQUE | 手机号码 |
| address | varchar | 50 | NOT NULL | 否 |  | 地址 |
| aged\_status | varchar | 3 | NOT NULL | 否 |  | 住院状态，（0：住院中，1：外出中） |
| emp\_id | integer |  | NULL | 否 | UNIQUE | 负责员工 |
| guardian\_id | bigint |  | NULL | 否 |  | 监护人 |
| room\_id | varchar | 8 | NULL | 否 | UNIQUE | 房间号 |

监护人表（guardian\_guardian），如下方表 4-5所示。

表 4-5 guardian\_guardian表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 监护人id |
| account | varchar | 50 | NOT NULL | 否 | UNIQUE | 账号 |
| gar\_id | varchar | 23 | NOT NULL | 否 | UNIQUE | 身份证号码 |
| phone | varchar | 11 | NOT NULL | 否 | UNIQUE | 手机号码 |
| password | varchar | 30 | NOT NULL | 否 |  | 密码 |
| name | varchar | 50 | NOT NULL | 否 | UNIQUE | 姓名 |
| sex | varchar | 2 | NOT NULL | 否 |  | 性别 |

房间表（employee\_room），如下方表 4-6所示。

表 4-6 employee\_room表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | varchar | 8 | NOT NULL | 是 | UNIQUE | 房间id |
| status | varchar | 10 | NOT NULL | 否 |  | 入住状态（0：使用，1：未使用） |
| emp\_id | integer |  | NULL | 否 |  | 负责员工id |

每日数据表（data\_data），如下方表 4-7所示。

表 4-7 data\_data表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer | id | NOT NULL | 是 | UNIQUE | 数据id |
| temp | real | temp | NULL | 否 |  | 体温度数 |
| blood\_pressure | real | blood\_pressure | NULL | 否 |  | 血压值 |
| blood\_sugar | real | blood\_sugar | NULL | 否 |  | 血糖值 |
| blood\_oxygen | real | blood\_oxygen | NULL | 否 |  | 血氧值 |
| weight | integer | weight | NULL | 否 |  | 体重 |
| date | date | date | NULL | 否 |  | 日期 |
| time | time | time | NULL | 否 |  | 时间 |
| aged\_id | bigint | aged\_id | NOT NULL | 否 |  | 老人id |

公告表（all\_announcement），如下方表 4-8所示。

表 4-8 all\_announcement表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 公告id |
| date\_joined | datetime | NOT NULL | NOT NULL | 否 |  | 创建时间 |
| author | varchar | 50 | NOT NULL | 否 |  | 作者 |
| content | text |  | NOT NULL | 否 |  | 内容 |

新闻表（news\_news），如下方表 4-9。

表 4-9 news\_news表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 新闻id |
| date\_joined | datetime | NOT NULL | NOT NULL | 否 |  | 创建时间 |
| author | varchar | 50 | NOT NULL | 否 |  | 作者 |
| content | text |  | NOT NULL | 否 |  | 内容 |

访客表（visit\_visit），如下方表 4-10所示。

表 4-10 visit\_visit表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 来访数据id |
| visit\_name | varchar | 50 | NOT NULL | 否 |  | 访客姓名 |
| visit\_phone | varchar | 11 | NOT NULL | 否 |  | 访客电话 |
| visit\_reason | varchar | 1024 | NOT NULL | 否 |  | 来访原因 |
| visit\_subscribe | varchar | 2 | NOT NULL | 否 |  | 是否预约（0：否，1：是） |
| visit\_date | date |  | NOT NULL | 否 |  | 来访日期 |
| visit\_time | time |  | NOT NULL | 否 |  | 来访时间 |
| visit\_room\_id | varchar | 8 | NOT NULL | 否 |  | 探望0房间id |

预约表（management\_book），如下方表 4-11所示。

表 4-11 management\_book表结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 允许空值 | 主键 | 唯一 | 字段说明 |
| id | integer |  | NOT NULL | 是 | UNIQUE | 预约数据id |
| name | varchar | 512 | NOT NULL | 否 |  | 预约人姓名 |
| gender | varchar | 2 | NOT NULL | 否 |  | 预约人性别 |
| age | integer |  | NOT NULL | 否 |  | 预约人年龄 |
| phone | varchar | 256 | NOT NULL | 否 |  | 预约人电话 |
| content | text |  | NOT NULL | 否 |  | 预约内容 |
| date | date |  | NOT NULL | 否 |  | 预计来访日期 |
| time | time |  | NOT NULL | 否 |  | 预计来访时间 |
| status | varchar | 256 | NOT NULL | 否 |  | 完成状态（0：未完成，1：已完成 |

# 系统实现

## 用户端功能设计

### 注册功能实现

首先用户需要在后台系统于监护人模块中登记个信息，身份证号码、手机号码为必填选项，其余为可选填项。用户进入小程序后，选择注册账号，注册页面如图 5-1所示，用户进行身份认证应先输入身份证号码和手机号码[[[1]](#endnote-0)]。如果检索结果为此二者所对应的数据已注册过，则弹出提示：该账号已注册；若未注册过，但二者匹配不成功则弹出提示：用户身份验证失败，匹配成功即身份验证通过，跳转下一步设置账号密码。由于账号具有唯一性，因此不能与数据库中已有的账号一致。以上所述功能点击按钮均有校验不为空。注册成功页面如图 5-2所示。



图 5-1 注册页面



图 5-2 注册成功页面

身份验证过程中，前端发送手机号码和身份证号码到达后端，后端首先判断该用户是否注册过，再进一步判断填入手机号码和身份证号码是否为空，再利用filter对验证信息进行过滤获取数据user，如果该数据存在于数据库，则表明身份验证成功，可继续往下执行注册功能；如果该数据不存在，则说明验证失败，用户需要重新输入或联系工作人员。注册验证后端逻辑代码如图 5-3所示。



图 5-3 注册验证身份代码

设置账号密码过程中，后端获取前端发送的身份证号码、手机号码等数据，先用filter过滤出数据表中是否已经存储过数据account，是的话则返回errmsg提示“该账号已存在”，若数据不存在则再往下判断输入的password和repassword是否一致，一致之后则使用update更新账号密码到后台对应用户的信息中。后端逻辑代码如图 5-4所示。



图 5-4 设置账号密码页面

### 登录功能实现

本系统的小程序端可以使用账号或者手机号码登录。账号于验证成功之后自己设置，手机号码在后台系统录入，二者均可作为用户登录养老小程序的登录凭证，用户能够注册登录的前提是有在养老院的后台系统中录入过个人信息以及绑定监护老人，否则无法进行这些操作。登录页面如图 5-5所示。



图 5-5 登录页面

前端发送账号密码到后端，手机号码和账号必须是监护人数据库中已有的数据，校验不为空后，后端获取到前端传过来的值：手机号/账号与密码，再到数据库中与现有的监护人数据进行检索，filter后有数据匹配成功则登录成功，否则将提示用户名或密码输入错误[[[2]](#endnote-1)]。后端逻辑代码如图 5-6所示。



图 5-6 登录代码

### 修改密码功能实现

点击忘记密码按钮，小程序会跳转到一个特定页面，该页面提供修改密码的功能。用户可以在该页面按照提示输入来修改密码。页面效果如图 5-7所示。



图 5-7 修改密码页面

用户需要输入账号或者手机号码、身份证号码、两次新密码的内容，校验不为空之后，后端将首先filter手机号码或者账号与身份证号码是否匹配，若filter后没有值，说明不匹配，则提示身份验证失败，filter匹配则验证成功，往下识别两次密码是否输入一致，一致则调用update对密码进行修改，不一致则提示两次密码输入不一致。修改密码后端逻辑代码如图 5-8所示。



图 5-8 修改密码代码

### 首页功能实现

首页展示了养老院的轮播图，管理员发布的公告栏，以及院内信息导航栏，如图 5-9所示。



图 5-9 首页

轮播图使用了swiper组件，设置autoplay自动播放属性，内部swiper-item循环list1展示图片，具体代码如图 5-10所示。



图 5-10 轮播图代码

公告栏利用样式写出一个卡通的公告栏效果，栏中的信息调后端的announcement接口，将最新一条数据取出，赋给对象notice，并将notice的值渲染到页面中。若管理员未曾发布过公告，即notice为空，则默认展示v-else中的内容。时间要求只展示到天，因此使用了slice()方法，只取前十个字符，具体代码如图 5-11所示。



图 5-11 公告栏代码

球形导航栏展示了院内相关其他信息，使用uni-grid实现六宫格摆放。column的属性设置为3，将表格分为三列，内部循环baseList数组，将数组内各个元素的title，navurl，iconurl渲染到每一个uni-grid-item，并且写一个openinfo方法实现点击跳转，具体导航栏代码如图 5-12所示。

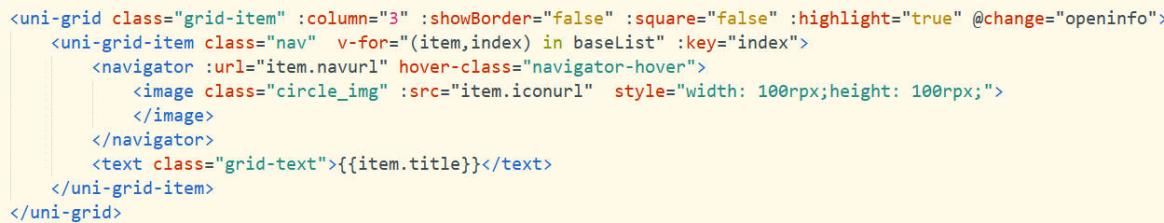


图 5-12 导航栏代码

### 导航栏功能实现

入住条件中存放了老人入住前所需具备的条件以及入住前的准备，遍历conlist和prelist，将内容渲染到每个li中，具体代码如图 5-13所示。



图 5-13 入住条件代码

在线预约中，不管是用户还是游客均可提交表格，表格中每一个选项均设置了不为空校验，提交成功之后后台系统将接收到此预约数据，预约页面如图 5-14所示。



图 5-14 在线预约页面

为了降低代码的冗余度，提高代码的复用性，因此对调用接口的方法进行了简化。首先对uni.request进行了封装，导出为request，如图 5-15所示，再将接口统一写在一个api.js文档中，导入request，编写接口再导出，如图 5-16所示。



图 5-15 request代码



图 5-16 api.js代码

这样封装接口的做法能够方便开发者对接口做修改及调用，调用提交预约表格接口如图 5-17所示。

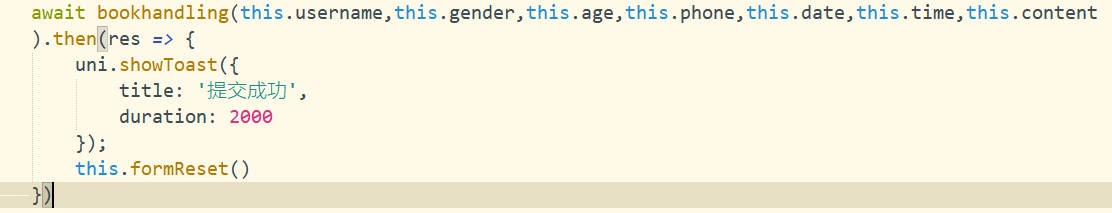


图 5-17 提交预约代码

最新动态中存放的是管理员发布的新闻，如图 5-18所示。进入页面后，mounted函数自动调getnews接口获取到newslist，再将newslist渲染到页面上，每个item点击可触发goDetail事件跳转到新闻详情页，具体代码如图 5-19所示。



图 5-18 最新动态页面



图 5-19 最新动态代码图

新闻页的新闻数据由跳转前点击事件所携带的item传入，onLoad接收为event，再使用decodeURIComponent()对接收到的event进行解码，赋给detail后渲染到页面上。新闻详情页面如图 5-20所示，具体代码如图 5-21，图 5-22所示。



图 5-20 新闻详情图

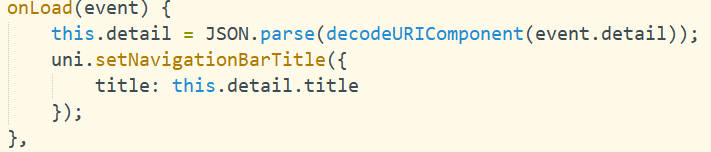


图 5-21 接收数据代码



图 5-22 新闻详情页代码

联系我们页面存放了养老院的电话、邮箱、地址以及地图，方便用户和游客联系养老院，页面展示如图 5-23所示。



图 5-23联系我们页面

### 数据功能实现

数据页面展示了监护人绑定老人后，老人每日的健康数据，若用户不登录则无法进入此页面，如图 5-24所示，登录后展示的页面如图 5-25所示。

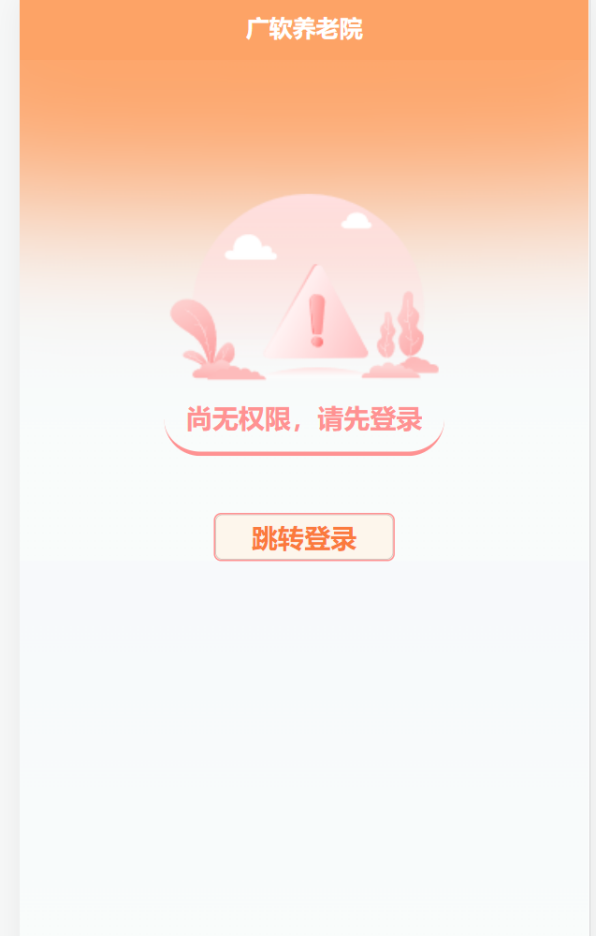


图 5-24 用户未登录



图 5-25 已登录数据

监护人可选择查看的老人数据以及日期，如图 5-26所示。



图 5-26选择器

健康数据中，如果值低于正常值，则该数值展示为蓝色，高于正常值则展示为红色，此功能是用vue2的计算属性computed写的，设置三个范围返回不同颜色，应用于style中，计算属性的代码如图 5-27所示。BMI的值则由过滤器filters写，传入身高体重的值，过滤器自动将其转为BMI的值，代码如图 5-28所示。

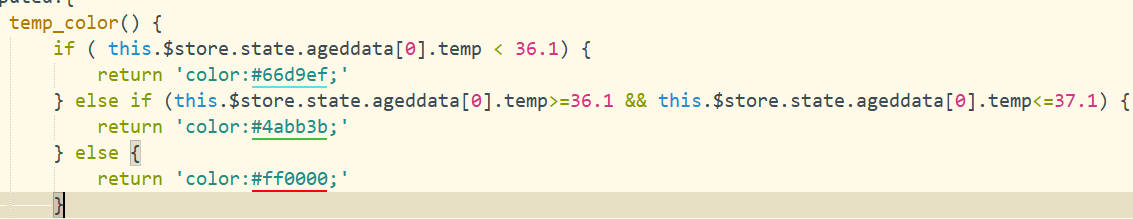


图 5-27 计算属性代码

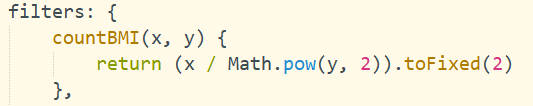


图 5-28 过滤器代码

### 图表功能实现

图表页面展示了监护人绑定的每位老人的近七条健康指标数据的可视化趋势图，使用了秋云组件（qiun-data-charts），如图 5-29所示，若老人的健康数据尚未录入过，则展示图如图 5-30所示（下页）。



图 5-29 健康指标趋势图



图 5-30 健康指标无数据

### 个人中心功能实现

个人中心展示了监护人的基本信息，用户可以跳转至绑定老人、访客记录、个人资料页面，并且可以执行退出登录操作，页面展示如图 5-31所示。



图 5-31 个人中心

### 个人资料功能实现

个人资料页面存储了监护人的信息，如图 5-32所示，点击监护人头像可以执行修改头像操作，首先调用uni.chooseImage方法，将监护人的id和图片的filePath传给后端的UploadAvatar接口，执行上传头像操作，成功后再次调用获取监护人的接口，将监护人的信息更新到本地缓存中，代码如图 5-33所示。

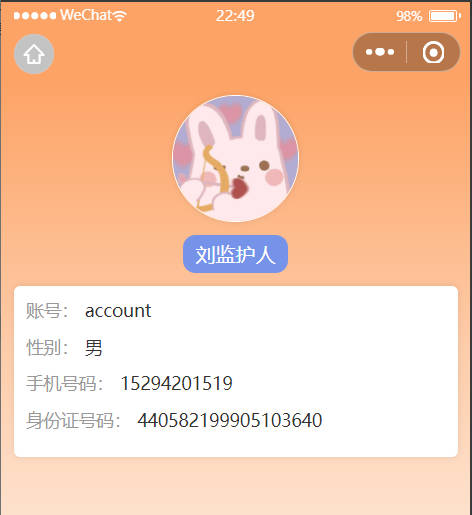


图 5-32 个人资料

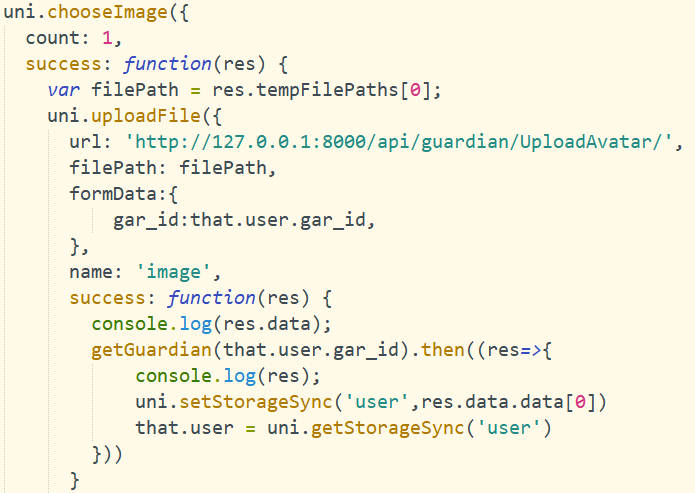


图 5-33 上传头像代码

### 访客记录功能实现

访客记录页面存储了监护人绑定的所有老人的访客信息，根据监护人id以及老人所住的房间id，筛出访客记录表中所对应的访客记录，页面如图 5-34所示。



图 5-34 访客记录

### 绑定老人功能实现

绑定老人页面展示了监护人绑定的所有老人的信息，如图 5-35所示。监护人可以直观地看到老人的姓名、房间、住院状态等情况。



图 5-35 绑定老人

## 后台管理系统功能设计

### 管理系统页面

本管理系统采用了Django框架从而快速搭建起一个较为完善的管理系统。登录系统需要拥有超级管理员的身份，创建超级管理员成功后即可登录，登录成功，管理员可操作功能模块位于左侧，这些模块涵盖了系统的所有主要功能，管理员可以根据需要灵活选择相应的功能模块进行操作，以实现对系统的全面管理和控制。系统登录页面如图 5-36所示（下页），功能模块如图 5-37所示（下页）。

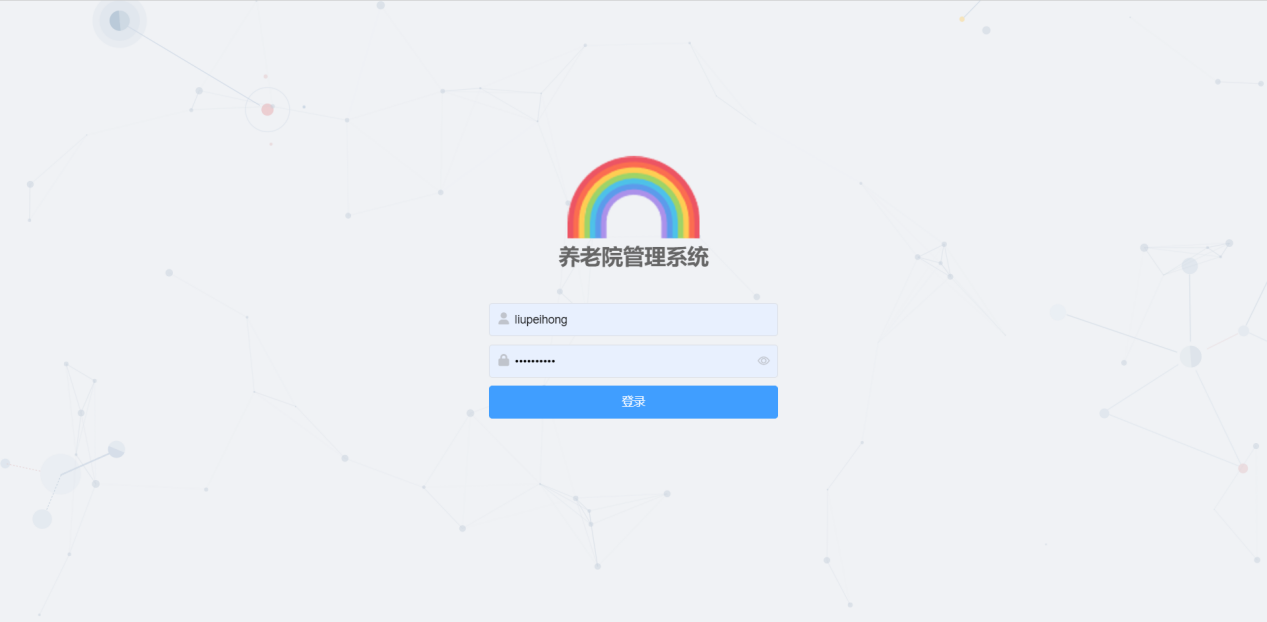


图 5-36 系统登陆页面

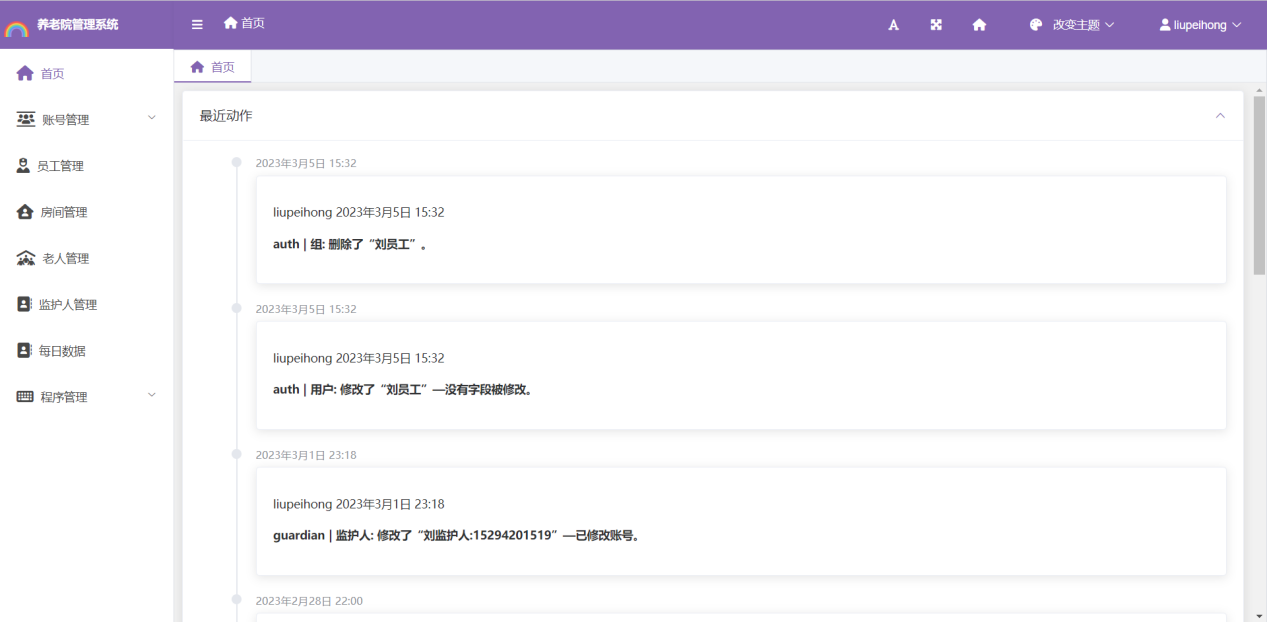


图 5-37 功能模块页面

### 账号管理模块

账号管理模块存放了管理员的个人信息。页面展示如图 5-38所示（下页）。

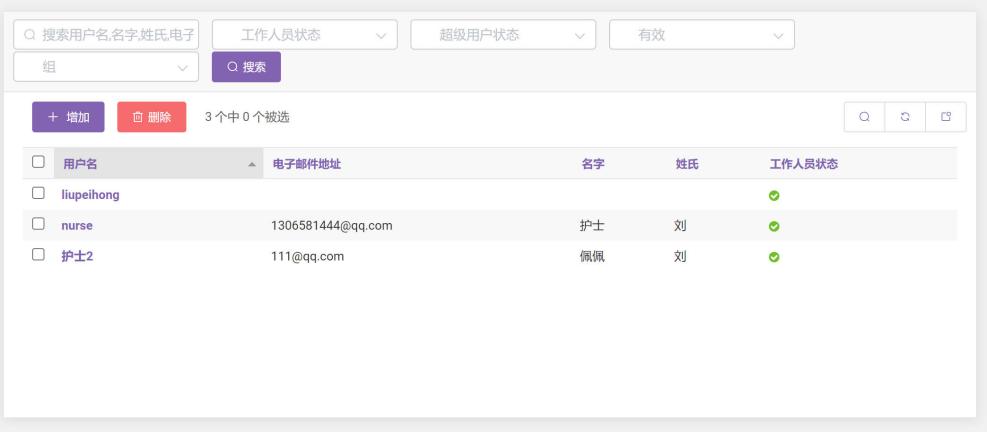


图 5-38 账号管理页面

### 员工管理模块

员工管理模块存放了所有员工的信息，顶部可在对应输入框中输入想要检索的信息，进而对员工列表进行筛选。页面展示如图 5-39所示。

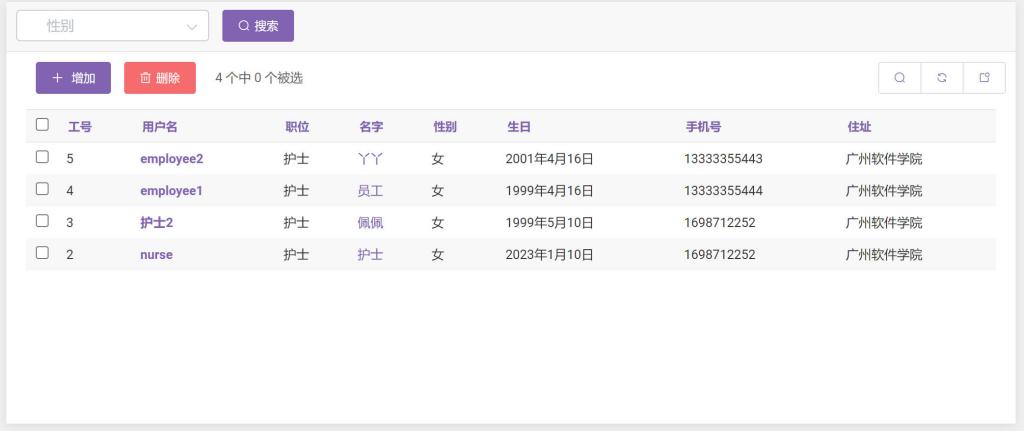


图 5-39 员工管理页面

### 老人管理模块

老人管理模块存放所有老人的个人信息以及入住房间、监护人，老人管理页面如图 5-40所示（下页）。添加老人的同时可以选择入住的房间号以及添加监护人，添加老人页面如图 5-41所示（下页）。



图 5-40 老人管理页面

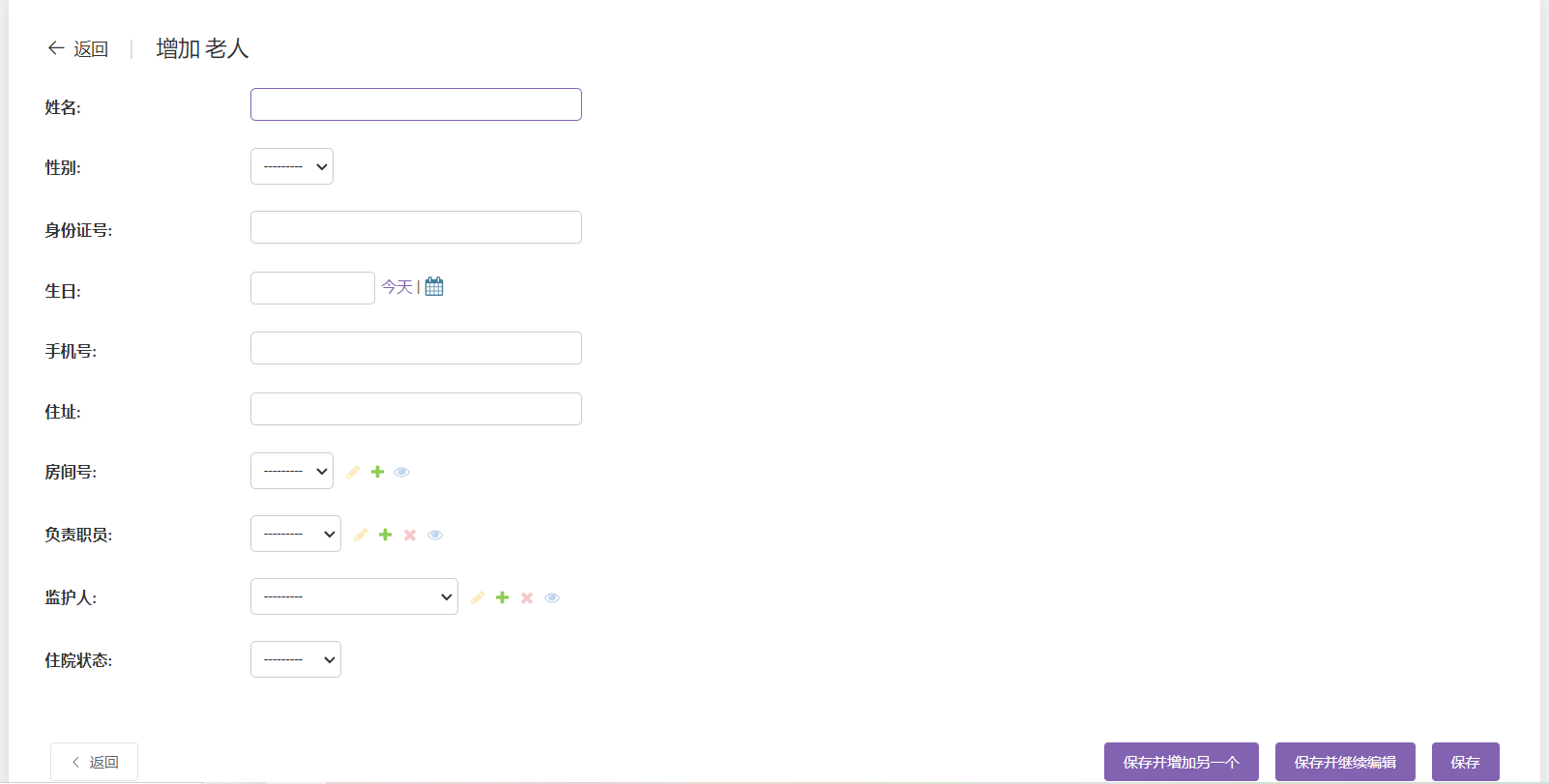


图 5-41 老人新增页面

一旦老人入住的房间确定，后端会filter筛选出变更前后的房间，并用update对两者进行更新，在房间管理模块中相对应房间的使用状态将自动更新为“已使用”，后端逻辑代码如图 5-42所示。



图 5-42 更改房间状态代码

### 房间管理模块

房间管理模块存放了院内所有房间，信息包括房间号、对应负责员工、入住老人，房间管理页面如图 5-43所示。



图 5-43 房间管理页面

### 监护人管理模块

在监护人管理模块中，管理员可以查看和管理所有老人的监护人信息，帮助管理员更好地维护老人的日常生活和健康情况。页面如图 5-44所示。

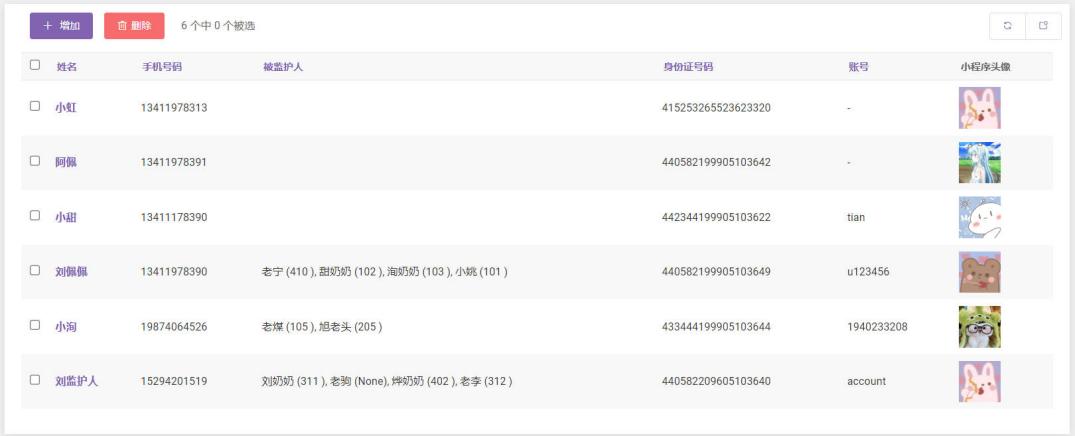


图 5-44 监护人管理页面

### 每日数据管理模块

每日数据管理模块存放了每位每日老人的最新健康数据，每日数据管理页面如图 5-45所示（下页），员工每日会对老人的数据进行更新，头部可根据老人名字、更新日期进行检索。



图 5-45 每日数据管理页面

### 程序管理模块

程序管理模块中管理员可以对在线预约记录、来访记录、公告、动态进行操作，通过该模块能够更加方便地管理和维护相关信息，以确保系统的正常运行和老人生活的便利性。目录如图 5-46所示。



图 5-46 程序管理模块目录

在线预约管理中存放了用户小程序端提交的预约内容，在线预约管理页面如图 5-47所示。管理员可查看并对预约人是否完成预约内容进行修改。

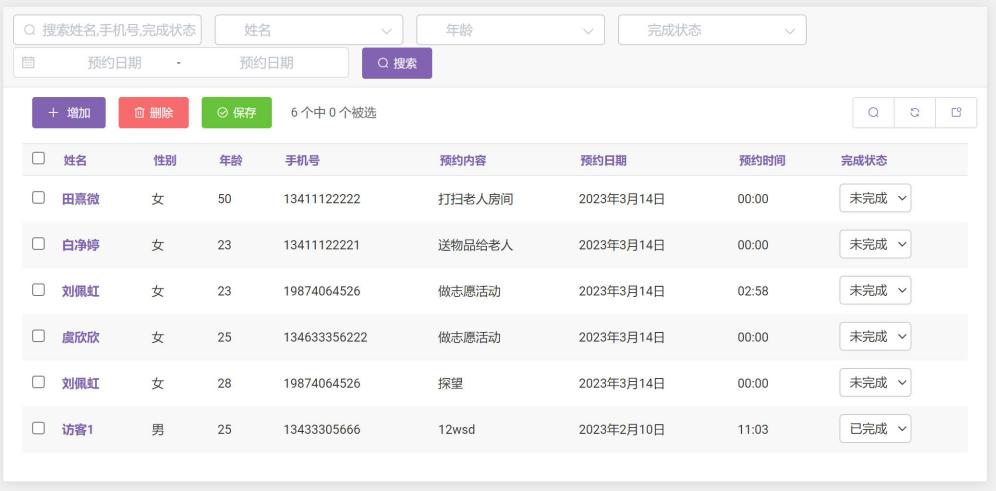


图 5-47 在线预约管理页面

来访管理模块针对到院人员的来访进行了统计，包含了线上预约后来访以及未线上预约来访，来访管理页面如图 5-48。



图 5-48 来访管理页面

若线上预约后来访，则填入来访信息后，后端会使用filter筛选是否有name，phone，date均符合条件的预约记录，如果该存在则将其的status更新为已完成，在线预约管理模块中所对应的预约内容的完成状态将自动更新为“已完成”，后端逻辑代码如图 5-49所示。

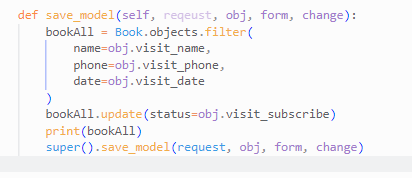


图 5-49 状态自动更新代码

公告管理模块由管理员添加内容，展示于用户小程序端的公告栏，公告管理模块页面如图 5-50所示。



图 5-50 公告管理模块

动态管理模块由管理员添加新闻动态，展示于用户小程序端的最新动态以及新闻详情页，动态管理模块页面如图 5-51所示（下页）。



图 5-51 新闻管理模块

# 系统测试

在本章节中，我们对系统的各个接口以及功能进行了测试，旨在确保系统的稳定性和可靠性，为后续系统的正常运行提供有力保障。

## 系统配置

本系统所进行测试的环境具体配置如表 6-1所示。

表 6-1 测试配置表

|  |  |
| --- | --- |
| 配置名称 | 详细配置 |
| 处理器（CPU） | AMD Ryzen 7 4800H |
| 显卡（GPU） | NVIDIA GeForce RTX 2060 |
| 内存 | 16GB |
| 硬盘空间 | 1TB |
| 操作系统 | win10X64 |
| 测试软件 | 微信开发者工具、Uni-app、Postman |
| 数据库 | DB Browser (SQLite)、Navicat Premium 15 |
| 网络环境 | 有线网络、无线网络 |
| 物理硬件 | PC 机、服务器 |
| 浏览器 | Google Chrome、FireFox、Microsoft Edge |
| 开发工具 | VS Code、Uni-app |

## 功能测试

### 小程序端注册功能测试

用户小程序端注册功能具体测试用例如表 6-2所示。

表 6-2 用户小程序端注册测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 1 | 用例名称 | 用户小程序注册 | |
| 用例描述 | 用户通过输入身份证号码和手机号码验证身份通过后输入账号和密码进行注册操作 | | | |
| 用例入口 | 小程序端登录页面注册组件点击注册按钮 | | | |
| 测试场景ID | 测试场景 | 测试步骤 | 预期成果 | 实际结果 |
| Test01 | 小程序端注册页面 | 不输入身份证号码或手机号码 | 提示“身份证号码不为空”或“手机号码不为空” | 符合预期 |
| Test02 | 小程序端注册页面 | 输入错误身份证号码或手机号码 | 提示“手机号码或身份证号码验证失败” | 符合预期 |
| Test03 | 小程序端注册页面 | 输入已存在用户的身份证号码和手机号码 | 提示“该用户已注册” | 符合预期 |
| Test04 | 小程序端注册页面 | 输入正确身份证号码和手机号码 | 提示“身份验证成功”，进入下一步设置账号密码 | 符合预期 |
| Test05 | 小程序端注册页面 | 不输入账号或密码 | 提示“账号不为空”或“密码不为空” | 符合预期 |
| Test06 | 小程序端注册页面 | 输入的两次密码不一致 | 提示“两次密码输入不一致” | 符合预期 |
| Test07 | 小程序端注册页面 | 输入已存在的账号 | 提示“账号已存在” | 符合预期 |
| Test08 | 小程序端注册页面 | 输入符合要求的账号密码 | 提示“注册成功，请返回登录” | 符合预期 |

### 小程序端登录功能测试

用户小程序端登录功能具体测试用例如表 6-3所示。

表 6-3 用户小程序端登录测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 2 | 用例名称 | 用户小程序登录 | |
| 用例描述 | 用户通过手机号或账号进行登录操作 | | | |
| 用例入口 | 小程序端个人中心点击登录按钮 | | | |
| 测试场景ID | 测试场景 | 测试步骤 | 预期成果 | 实际结果 |
| Test01 | 小程序登录页面 | 不输入账号或密码 | 提示“账号不为空”或“密码不为空” | 符合预期 |
| Test02 | 小程序登录页面 | 输入错误账号或密码 | 提示“账号或密码输入错误，请重新输入” | 符合预期 |
| Test03 | 小程序登录页面 | 输入正确账号和密码 | 提示“登录成功”，跳转到首页 | 符合预期 |

### 小程序端数据展示功能测试

用户小程序端数据展示页面只有登录成功后携带token返回值的用户才能查看，具体测试用例如表 6-4所示。

表 6-4 数据展示测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 3 | 用例名称 | 小程序数据页面展示 | |
| 用例描述 | 用户查看数据页面是否具有权限 | | | |
| 用例入口 | 小程序端点击tabbar跳转数据页面 | | | |
| 测试场景ID | 测试场景 | 测试步骤 | 预期成果 | 实际结果 |
| Test01 | 小程序数据页面 | 不登录点击数据页面 | 提示“尚无权限，请先登录” | 符合预期 |
| Test02 | 小程序数据页面 | 登录后点击数据页面 | 展示监护老人数据 | 符合预期 |

### 小程序端提交预约来访功能测试

用户小程序端具有预约来访功能，游客或已用账号用户都可提交，提交后将上传到后台管理系统的在线预约模块中，具体测试用例如表 6-5所示。

表 6-5 提交预约来访测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 4 | 用例名称 | 小程序预约来访提交 | |
| 用例描述 | 用户提交预约来访上传至后台 | | | |
| 用例入口 | 小程序端点击首页“预约来访”中的提交按钮 | | | |
| 测试场景ID | 测试场景 | 测试步骤 | 预期成果 | 实际结果 |
| Test01 | 小程序预约来访页面 | 填写信息点击提交 | 提示“提交成功”，后台系统在线预约模块中出现该预约内容 | 符合预期 |

### 后台系统老人模块搜索功能测试

对老人模块的检索功能进行测试，验证该功能的稳定性和准确性，具体测试用例如表 6-6所示。

表 6-6 老人模块搜索测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 5 | 用例名称 | 后台系统老人模块搜索 | |
| 用例描述 | 进入老人模块顶部搜索框输入关键字搜索 | | | |
| 用例入口 | 老人模块顶部搜索按钮 | | | |
| 测试场景ID | 测试场景 | 测试步骤 | 预期成果 | 实际结果 |
| Test01 | 后台系统老人模块 | 在姓名输入框中输入“刘奶奶”，点击搜索 | 展示1条结果，结果为刘奶奶的个人信息 | 符合预期 |
| Test02 | 后台系统老人模块 | 在姓名输入框中输入“刘爷爷”，点击搜索 | 展示0条结果，当前信息不存在 | 符合预期 |
| Test03 | 后台系统老人模块 | 在手机号输入框输入“19874064526”，点击搜索 | 展示1条结果，结果为手机号码为“19874064526”的老人的个人信息 | 符合预期 |

### 后台系统来访管理是否预约功能测试

在来访管理中如若填写的数据是已预约，则对应在线预约模块中的数据中的完成状态将动态更改，具体测试用例如表 6-7所示。

表 6-7 来访模块预约测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 6 | 用例名称 | 后台系统来访模块预约 | |
| 用例描述 | 进入来访模块输入已预约的数据 | | | |
| 用例入口 | 点击在线预约对应的数据完成状态 | | | |
| 测试场景ID | 测试场景 | 测试步骤 | 预期成果 | 实际结果 |
| Test01 | 后台系统来访模块、在线预约模块 | 来访模块中新增一条预约状态为“是”的数据 | 在线预约模块中匹配的预约数据的完成状态自动更新为“已完成” | 符合预期 |
| Test02 | 后台系统来访模块、在线预约模块 | 来访模块中新增一条预约状态为“否”的数据 | 在线预约模块无发生变化 | 符合预期 |

## 测试总结

本次测试主要针对小程序端用户权限功能以及后台系统一些相关联数据动态变化的测试，目的在于确保本智慧养老系统能够合理且流畅地运行，能基本达到预期。通过一次次的测试与修改，对功能的各个方面进行完善，努力带给用户本系统最好的体验。

# 总结与展望

## 总结

本智慧养老系统如若在小型企业中落地实施，将改变传统的管理体系，企业的员工可以更大提高管理效率，用户也将挣脱时间以及空间上的限制。由于个人的时间和能力有限，此系统若要投入到现实生活中落地还有许多不足，需要改进和完善的部分还有许多，本文总结如下：

目前我国人口老龄化问题严重，养老院的前景广阔，结合互联网更是符合信息化时代的主题。

通过查阅近年来养老院资料，制定了本系统管理端和用户端的规划并完成设计。

通过线上管理院内系统，解决了大量数据的使用及保存问题。

用户小程序端界面简洁，操作简单，目标明确。

本系统扩展设计仍可添加员工薪资、绩效考核、医疗项目等模块，深入完善系统的结构，使得系统的实用性更高更强。

系统的安全性不算高，如若落地还需进一步完善，提高安全性，保护数据及隐私。

## 展望

本系统的大致功能虽然已按照规划设计出来了，但是落地使用到现实生活中的话在时间、精力、经济上有更高需求。面对此系统的不足与缺陷，本人会在以后进行更加深入的调查与研究，学习更多技术，期待早日能够做出一个可以发挥作用到实际应用生活中的系统。

参考文献

1. [] 易远弘. 高校信息化背景下的碎片化服务研究与实践 [D]. 宁波大学, 2018. [↑](#endnote-ref-0)
2. [] 赵岸伟. 基于深度学习的商品推荐系统的设计与实现[D]. 华中科技大学, 2019.

   致 谢

   写到“致谢”环节，意味着我的大学生涯即将结束。一路走来，无数朋友给予了我莫大的帮助，纵有万般不舍，我也要踏出这一步迈向一段崭新的旅途。

   首先感谢我的毕业设计指导老师——赵莲芬老师，悉心指导着我的毕业设计，在我迷茫之际给我灵感和建议。其次非常感谢元数科技公司愿意给我提供实习机会，在这五个月的实习期里学到了很多，也结识了很多志同道合的朋友，也非常感谢在实习单位的同事们，每当我遇到困难的时候都向我伸出援手，悉心为我解开谜题。最重要的是非常感谢我的父母和家人，为我提供一个良好的学习环境，以及物质保障和精神支持。感激言不尽，在日后的工作和生活中，我将怀揣着一颗感恩的心勇往直前。

   行文于此，落笔未终。祝我们每个人既有前程可奔赴，亦有岁月可回首。 [↑](#endnote-ref-1)