

## 家政服务云平台系统的设计与实现

### 摘 要

家政服务云平台是家政行业的“淘宝网”，用户可通过 APP 登录家政服务云平台,选择合适的家政服务项目进行下单。

家政服务云平台的所有家政人员均为实名认证，家政服务云平台提供的工种已覆盖全部家政服务，包括家居养护、洗护服务等主要功能，并将在未来拓展更多家庭生活服务，满足用户的所有家庭服务需求。

家政服务云平台致力于为家庭用户提供家政服务信息及最匹配的家政服务，确保获得最快捷、舒心的家政服务，让生活更简单、更轻松。本系统开发了 App，进而实现智慧社区，所以本课题具有重要的实际意义。

关键词：家政服务；App；智慧社区

## 目 录

1 需求分析.....	1
1.1 可行性分析.....	1
1.1.1 技术可行性分析.....	1
1.1.2 经济可行性分析.....	1
1.1.3 操作可行性分析.....	1
1.2 功能分析.....	1
1.2.1 用户 app 模块.....	2
1.2.2 后台管理系统模块.....	3
1.2.3 员工 app 模块.....	5
2 概要设计.....	6
2.1 数据库设计.....	6
2.1.1 数据库概要设计.....	6
2.1.2 数据库表设计.....	7
3 详细设计.....	11
3.1 订单管理.....	11
3.1.1 订单数据流.....	11
3.2 产品管理.....	12
3.2.1 产品数据流图.....	12
3.2.2 产品状态图.....	13
4 软件实现.....	14
4.1 登录实现.....	14
4.1.1 核心代码.....	14
4.1.2 运行截图.....	14
4.2 顾客预约服务.....	16
4.2.1 核心代码.....	16
4.2.2 运行截图.....	16

---

4.3 管理员派送订单.....	17
4.3.1 核心代码.....	17
4.3.2 运行截图.....	17
4.4 员工接单.....	17
4.4.1 核心代码.....	17
4.4.2 运行截图.....	18
4.5 员工完成服务.....	18
4.5.1 核心代码.....	18
4.5.2 运行截图.....	19
4.6 顾客确认完成.....	19
4.6.1 核心代码.....	19
4.6.2 运行截图.....	20

## 1 需求分析

### 1.1 可行性分析

开发任何一个系统都需要进行可行性分析，可以减少项目的开发风险，避免人力物力财力的浪费。下面从技术，经济，操作三个方面介绍。

#### 1.1.1 技术可行性分析

本系统开发工具为 vscode、mysql、idea、maven、git。前端基础开发语言为 html、css、Javascript，框架使用的是目前企业级开发用的非常频繁的 vue 全家桶，其中包含了 vueRouter、vuex、axios、element 等。后端基础开发语言为 Java，框架使用的也是目前企业级开发常用的 springBoot 系列技术，其中包含 spring、springmvc、mybatis、mybatis generator、swagger-ui、fastdfs 等。移动端采用的与前端技术基本一致，但是需要学生安装一个 android 环境用于到 app 进行打包。这些技术已经在大一至大三期间接触过，参照官方文档使用难度不大。

#### 1.1.2 经济可行性分析

本系统采用的软件与技术均为开源免费，后期在部署的时候会用到云服务器，而现在市面上提供的阿里云服务器或者腾讯云服务学生体验机只需要 10 元/月，所以完成这个系统经济上是可行的。

#### 1.1.3 操作可行性分析

本系统是 B/S 架构项目，管理员只需要打开浏览器就可以对数据进行管理和维护，对于顾客和员工，只需要下载一个 app 即可，app 的用户体验也非常好。操作上也是可行的

### 1.2 功能分析

家政服务平台分为三个端，分别为管理员端，用户端，员工端。用户端和员工端都为 app，用户使用 app 查看平台中提供的家政服务，并且选择服务进行下

单，平台接收到订单后进行派单，将订单指派给某个员工，员工选择接单或者拒单，如果选择接单，员工需要按照指定的时间和地点进行服务，服务完成后拍照回传进行备份，用户确认订单然后进行评价。订单完成后按照比例为员工分配薪酬。

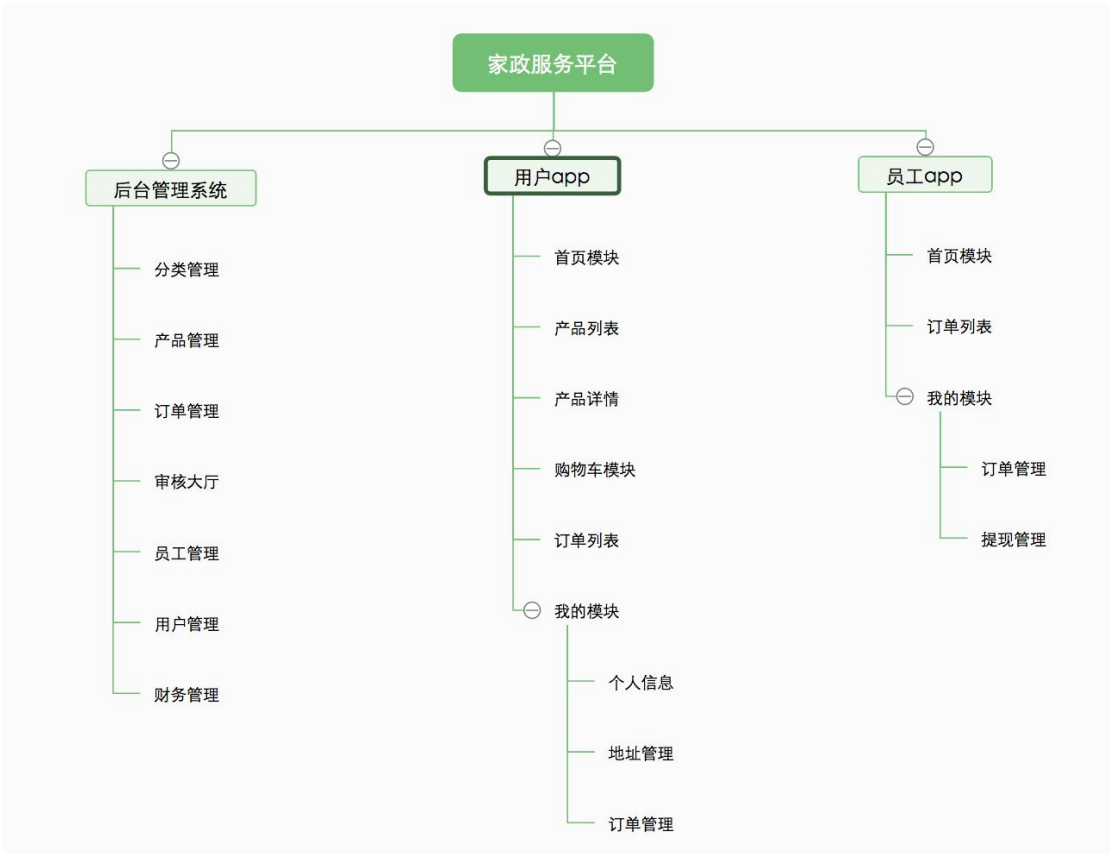


图 1-1 产品功能结构图

### 1.2.1 用户 app 模块

用户是该平台的金主，现在的生活工作压力巨大，下班后无暇家务，他们可以像滴滴约车一样在我们的家政服务平台上预约家政服务，预约完成在线支付，服务员服务完成后，用户也可以进行点评，如果不满足服务质量，也可以进行投诉，申请平台管理员进行仲裁。其最终目的是让用户满意。用户 app 中具有如下核心功能：

用户注册

用户输入手机号，通过验证码完成注册，注册成功后系统要求用户完成初始密码的设置。

## 用户登录

用户输入账号（默认为手机号）、密码完成登录

## 编辑个人信息

登录成功后用户可以编辑个人信息，包括头像的设置，服务地址的管理新增、修改、删除、设置默认地址。

## 首页模块

首页中轮播显示宣传大图，显示常用栏目，动态显示点击量较高的产品

## 产品列表

在首页中点击常用栏目和产品都可以进入到产品列表，产品列表的布局应该是左侧为栏目名称，右侧为产品列表，点击左侧栏目名称可以动态筛选产品，点击产品名称可以跳转到产品详细介绍页面。在产品列表和产品详情页面中具有添加购物车按钮，用户点击该按钮可以将产品加入到购物车中

## 购物车

用户进入到购物车中，可以查看购物车中产品列表，并且可以修改产品数量。

## 订单确认

在购物车中选择要结算的产品，点击去结算进入到订单确认界面，在订单确认界面中需要确认已下单的产品、服务地址、服务时间、备注

## 订单管理

用户可以根据订单状态过滤出未支付订单、待派订单、待接订单、待服务、待确认、已完成订单。点击订单可以进入到订单详情页面

## 评价

订单服务完成后用户可以对本次服务进行评价，评价结果影响到后期员工的派单量

## 1.2.2 后台管理系统模块

### 管理员登录

管理员输入用户名和密码进行登录，不可以进行注册，账号在项目初始化的时候进行分配。

### 分类管理

不同的产品应该归属不同的分类，通过分类更好管理产品，例如，干洗羊绒大衣，干洗羽绒服产品属于洗护分类，分类支持多级分类。管理员可以添加分类信息，修改分类信息，删除分类信息，对分类信息进行排序。

#### 产品管理

产品也就是本系统中的服务，服务应该具有名称，价格，所属栏目，介绍，主图等。管理员可以添加产品

#### 订单管理

管理员可以查看所有订单，并且可以按照订单状态筛选订单。管理员可以给未派订单绑定员工。员工可以选择是否接单，如果接单，员工需要按照约定的时间，地点进行服务。如果取消订单，平台将会记录，然后重新派单。管理员也可以查看订单的细节。

#### 员工审核

员工是该平台的服务人员，对于平台的正常运转起着至关重要的功能，所以员工的审核工作也变得至关重要。管理员需要对员工的身份证照片，头像等基本信息进行认真审核

#### 财务管理

管理员可以在该模块看到所有员工的收入记录，并且可以设置收入分配比例，例如：一个订单 100 元，平台收取 15%作为平台手续费，员工可以收入 85%。也可以设置提现基准值，例如，只有当余额超过 100 时才能提现。

#### 提现审核

当员工的余额满足提现基准值时，员工可以发起提现申请，管理员负责审核并且进行转账，转账需要上传转账凭证，暂时只支持支付宝转账和银行卡转账。

#### 用户角色

管理员在这里可以进行用户，角色，权限的管理。用户包含所有的用户，一个用户对应一到多个角色，一般一个用户对应一个角色，角色可以为管理员，员工，顾客...等，管理员可以对角色信息进行自定义。一个角色具有多个权限，一个权限也可以属于多个角色。权限分为两种，菜单和接口，菜单为后台管理的菜单，支持多级菜单（一般为二级菜单），接口与后台提供的接口保持一致，当为角色分配了接口前端开发者才可以通过 ajax 调用后台接口，否则将会出现权限

异常。最终要实现不同角色的用户登录后看到的菜单不同，当用户调用未授权的接口时也会提示权限异常。

### 1.2.3 员工 app 模块

员工是本平台的主要劳动力，这部分人主要是附近兼职的阿姨，她们可以通过在本平台上进行接单为用户提供服务以此来获取额外的收益。

员工注册登录

员工可以通过手机号进行注册，登录

实名制

员工为用户提供服务，所以员工的素质决定了平台的质量，员工只有实名制之后系统才会为其派单。实名制的时候用户需要上传本人头像，身份证正反面

员工接单

系统管理员为在线员工派单，员工根据自己时间安排选择是否接单，接单后员工可以看到用户的联系方式

服务完成

服务完成后员工点击服务完成按钮，然后需要让员工上传照片（5 张），待用户确认服务完成后本次服务的费用就会进入员工的私人账户

收入明细

用户可以看到收入明细，并且可以发起提现申请，申请后 3 个工作日到达私人账户。提现前用户需要绑定提现账户，这里支持支付宝和银行卡

订单管理

用户可以根据订单状态查看订单信息



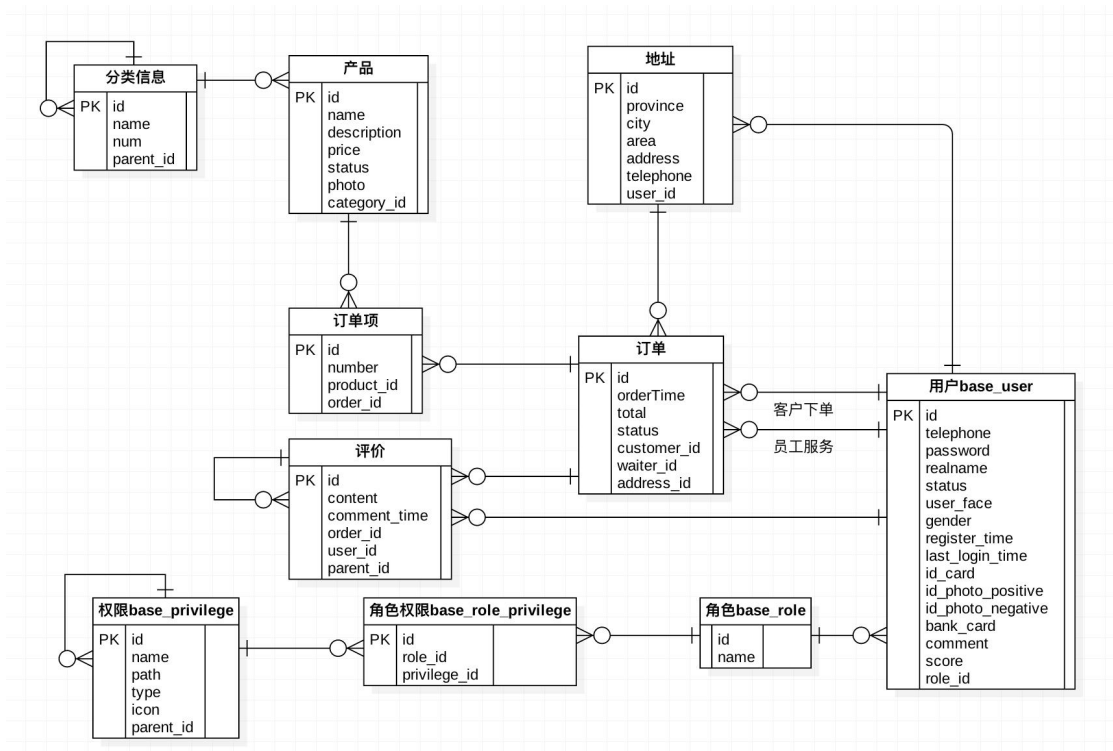
## 2 概要设计

### 2.1 数据库设计

科学技术日新月异的进步，让人类生活发生了巨大的变化，计算机技术的飞速发展，使各行各业在计算机技术应用方面得到了广泛的普及和使用。因此，家政服务平台解决了用户寻找家政服务人员难，家政服务人员寻找工作机会难的问题，通过使用家政服务平台，为用户与家政服务人员建立起一个相互信任的平台。用户可以选择服务，进行下单，付款后资金托管到平台上，更加安全一些。家政服务人员为用户提供服务，服务完成后平台将用户托管的资金再划拨到家政服务人员账户中。

#### 2.1.1 数据库概要设计

结合实际要求，对 Mysql 数据库管理系统、SQL 语言原理、Navicator、idea 进行了较深入的学习和应用，主要完成对家政服务平台的需求分析、功能模块划分、数据库模式分析，并由此设计了数据库结构和应用程序。系统运行实现了栏目管理，产品管理，订单管理等功能。E-R 图是指导开发者创建数据库的根本，如图是家政服务平台的 E-R 图



### 2.1.2 数据库表设计

结合需求分析与 E-R 图创建数据库表，本系统使用的是 mysql 数据库。

用户表中应该维护用户基本数据，这里的用户包含了系统中的系统管理员，用户，员工，我们是通过角色来区分他们。

表 2-1 用户表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
telephone	手机号	varchar		
password	密码	varchar		
realname	姓名	varchar		
status	状态	varchar		
user_face	用户头像	varchar		
gender	性别	varchar		
register_time	注册时间	bigint		
last_login_time	上次登录时间	bigint		

id_card	身份证号	varchar		
id_card_positive	身份证正面	varchar		地址
id_card_negative	身份证反面	varchar		地址
bank_card	银行卡号	varchar		
comment	注释	varchar		
score	积分	int		
role_id	角色 id	bigint	外键	

角色表维护系统中的角色信息，每一个角色为一条记录，一个用户拥有一个角色，一个角色可以具有多个权限。

表 2-2 角色表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
name	角色名称	varchar		

权限表维护系统中的权限信息，权限分为两种，一种为菜单一种为权限，菜单主要控制着页面的菜单显示，用户登录后可以查询到该用户拥有的权限，系统会根据查询到的菜单动态渲染出左侧栏目，超级管理员可以为不同的角色分配权限。

表 2-3 权限表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
name	角色名称	varchar		
path	路径	varchar		网页/路由
type	类型	varchar		菜单/权限
icon	图标	varchar		
parent_id	父权限	bigint	外键	

栏目表维护栏目信息，一个栏目下可以包含若干个子栏目，一个栏目下也可以包含多个产品信息。

表 2-4 栏目表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
name	栏目名称	varchar		
nu	序号	Int		
parent_id	父栏目	bigint	外键	

产品表维护产品信息，产品也就是本系统中的服务。

表 2-5 产品表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
name	名称	varchar		
description	介绍	varchar		
price	单价	double		
status	状态	varchar		
photo	图片	varchar		
category_id	所属栏目	bigint	外键	

订单表维护着订单的信息，一个订单包含多个订单项，订单项是订单的细节，例如某一个产品买了 3 件，单价为 12.5 元。

表 2-6 订单表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
order_time	下单时间	bigint		时间戳
total	总价	double		
status	状态	varchar		

customer_id	顾客 id	bigint	外键	
waiter_id	员工 id	bigint	外键	
address_id	服务地址	bigint	外键	

表 2-7 订单项表

列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
number	数量	int		
product_id	产品 id	bigint	外键	
order_id	订单 id	bigint	外键	

评论表维护用户对于一次服务的评价。

表 2-8 评论表

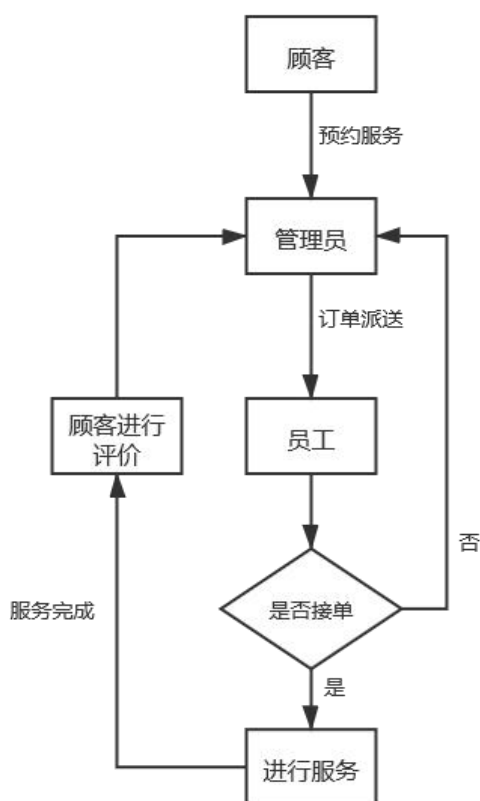
列名	列说明	类型	约束	备注
id	主键	bigint	主键，自增	
score	评分	int		
content	评论内容	varchar		
comment_time	评论时间	bigint		时间戳
order_id	订单 id	bigint	外键	
user_id	用户 id	bigint	外键	
parent_id	父评论	bigint	外键	

### 3 详细设计

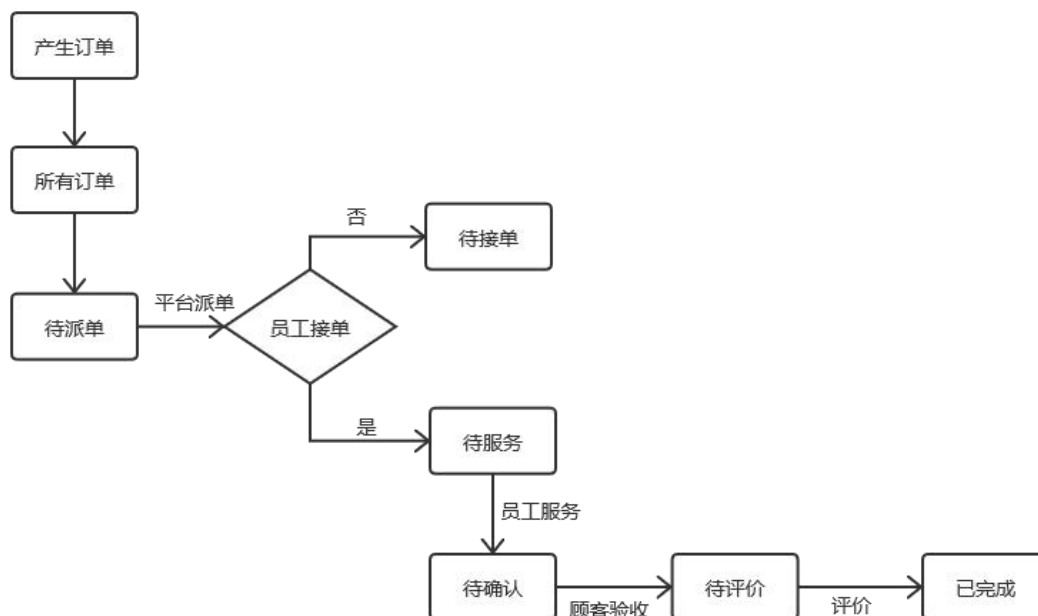
#### 3.1 订单管理

用户通过家政服务平台，选择需要的服务后，向平台提交订单，平台接收到订单后，管理员可以向该平台的员工派送此订单,此时员工可以选择是否接收平台派送的订单，如果该员工没有接受订单，管理员将派送此订单给另外的员工；如果该员工接收了订单，他将在规定时间内，到顾客给定的地址进行服务，服务完成后，员工提交订单，此时在管理员端，该订单的状态将变为“已完成”，之后，顾客付款完成并对此服务进行评价后，该订单即完成。

##### 3.1.1 订单数据流



##### 3.1.2 订单状态图

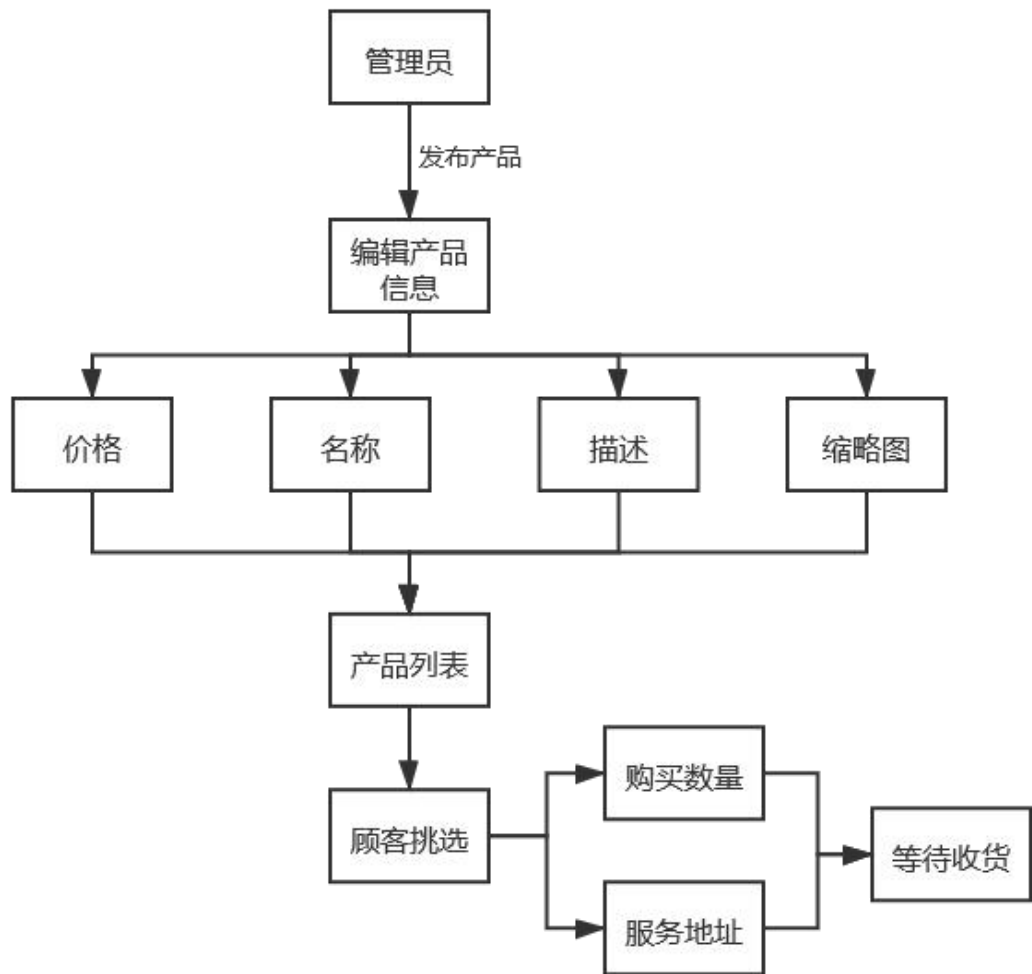


## 3.2 产品管理

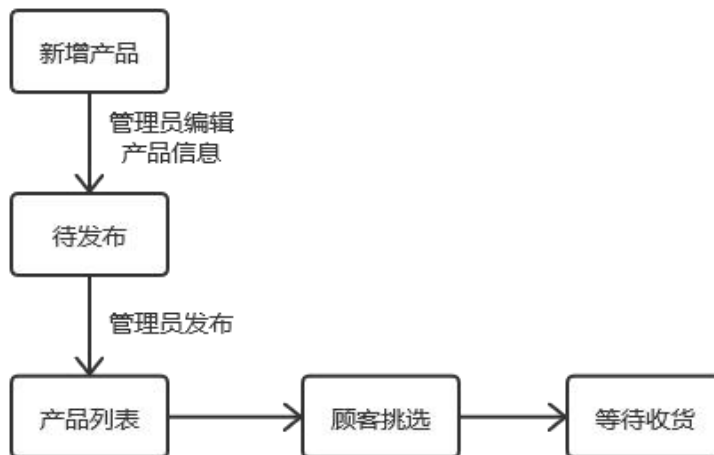
管理员可以通过平台发布各类产品的售卖信息，对于产品的名称、价格进行编辑，并且可以添加该产品的详细描述，通过详情，可以查看该产品的缩略图，管理员发布后，顾客就可以在 app 上查看到此产品，并选择是否购买，如果购买，则选择购买数量后立即下单，进而选择自己的地址以确认订单，之后等待产品送达即可。

### 3.2.1 产品数据流图

该流程图介绍了一款产品从发布到顾客购买的流程，具体流程如下：



### 3.2.2 产品状态图





## 4 软件实现

### 4.1 登录实现

登录主要对用户输入的用户名和密码进行校验，如果不是正确的账号和密码就无法登陆。如果用户名和密码正确，服务端认证成功，那么在服务端会返回 Token 给前端。前端可以在每次请求的时候带上 Token 来证明自己的合法地位。

#### 4.1.1 核心代码

```
@PostMapping(path="login")
@ApiOperation("登录")
public Message login( @RequestBody UserVM userVM) throws Exception{
    BaseUser user = baseUserService.findUser(userVM.getUsername(),userVM.getPassword(),
userVM.getType());
    if( user != null ){
        String uuid = UUID.randomUUID().toString().replaceAll("-", "");
        context.setAttribute(uuid,user);
        Map<String,String> map = new HashMap<>();
        map.put("token",uuid);
        return MessageUtil.success("success",map);
    } else {
        return MessageUtil.error("用户信息不符");
    }
}
```

#### 4.1.2 运行截图

logo

家政服务平台·顾客端

请输入用户名

请输入密码

登录

## 4.2 顾客预约服务

### 4.2.1 核心代码

```
@PostMapping("save")
```

```
@ApiOperation("保存订单信息")
```

```
public Message saveOrUpdate(@Valid @ModelAttribute OrderAndOrderLineVM order) throws
Exception{
    orderService.save(order);
    return MessageUtil.success("操作成功");
}
```

### 4.2.2 运行截图



## 4.3 管理员派送订单

### 4.3.1 核心代码

```
@GetMapping("sendOrder")
```

```
@ApiOperation("派单")
```

```
public Message sendOrder(
```

```
    @NotNull @RequestParam("waiterId") Long waiterId,
```

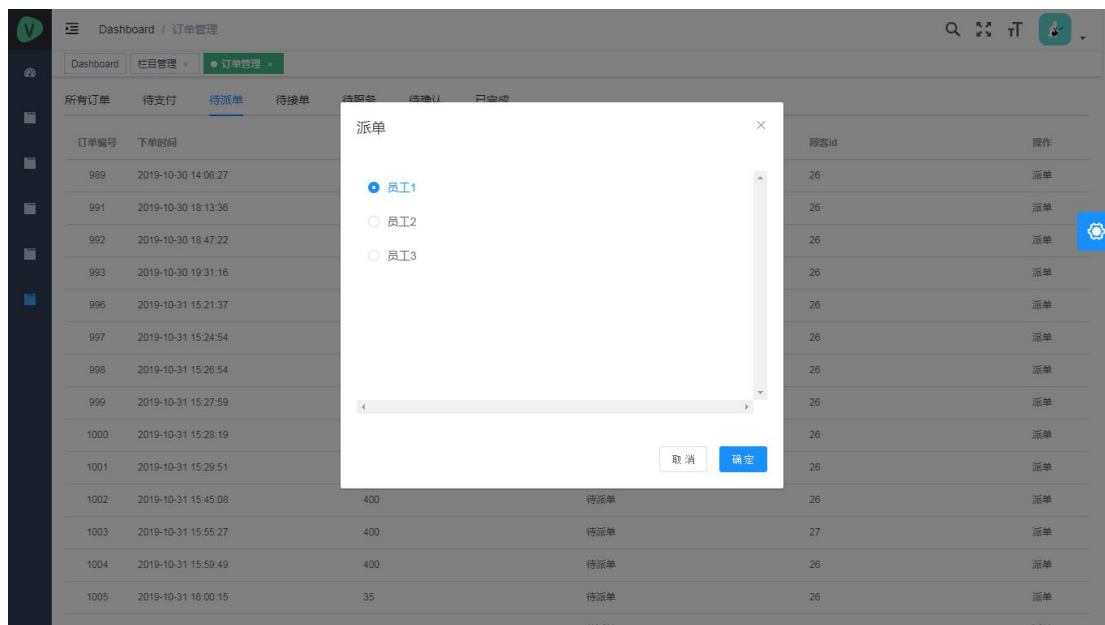
```
    @NotNull @RequestParam("orderId") Long orderId) throws Exception{
```

```
    orderService.sendOrder(waiterId,orderId);
```

```
    return MessageUtil.success("派单成功");
```

```
}
```

### 4.3.2 运行截图



## 4.4 员工接单

### 4.4.1 核心代码

```
@GetMapping("takeOrder")
```

```
@ApiOperation("接单")

public Message takeOrder(

    @NotNull @RequestParam("orderId") Long    orderId) throws Exception{

    orderService.takeOrder(orderId);

    return MessageUtil.success("接单成功");

}
```

#### 4.4.2 运行截图



#### 4.5 员工完成服务

##### 4.5.1 核心代码

```
@GetMapping("serviceCompleteOrder")
@ApiOperation("员工服务结束")
public Message serviceCompleteOrder(
    @NotNull @RequestParam("orderId") Long orderId) throws Exception{
    orderService.serviceCompleted(orderId);
    return MessageUtil.success("服务完成");
}
```

## 4.5.2 运行截图



## 4.6 顾客确认完成

### 4.6.1 核心代码

```
@GetMapping("confirmOrder")
@ApiOperation("确认订单")
public Message confirmOrder(
    @NotNull @RequestParam("orderId") Long orderId) throws Exception{
    orderService.confirmOrder(orderId);
    return MessageUtil.success("确认订单");
}
```

#### 4.6.2 运行截图

