# 产品概要设计文档

目录

[1. 引言 2](#_Toc72185143)

[1.1. 编写目的 2](#_Toc72185146)

[1.2. 准则与限制 2](#_Toc72185146)

[1.3. 范围 3](#_Toc72185146)

[1.4. 参考资料 3](#_Toc72185146)

[1.5. 设计规范 3](#_Toc72185146)

[2. 软件系统实现方案 4](#_Toc72185144)

[2.1. 组网结构 4](#_Toc72185146)

[2.2. 逻辑结构 5](#_Toc72185147)

[2.3. 层级结构 5](#_Toc72185146)

[2.4. 子系统划分 5](#_Toc72185146)

[2.5. 数据流图 5](#_Toc72185146)

[3. 接口 4](#_Toc72185144)

[3.1. 用户接口 4](#_Toc72185146)

[3.2. 外部接口 5](#_Toc72185147)

[3.3. 内部接口 5](#_Toc72185146)

[3.3.1 内部模块间关系 5](#_Toc72185146)

[3.3.2 接口数据描述 5](#_Toc72185146)

[4. 开发技术 4](#_Toc72185144)

[5.1. 开发框架 4](#_Toc72185146)

[5.2. 开发语言 5](#_Toc72185147)

[5.3. 开发技术评估 5](#_Toc72185147)

[5. 系统安全设计· 4](#_Toc72185144)

[6. 系统出错处理设计· 4](#_Toc72185144)

[6.1. 出错信息 4](#_Toc72185146)

[6.2. 补救措施 4](#_Toc72185146)

[6.3. 系统维护设计 4](#_Toc72185146)

# 1.产品原型设计规范

## 1.1 编写目的

   本说明的目的是对“美丽租”小程序进行概要设计的功能说明，以便用户及项目开发人员了解产品大致的设计与实现。以下叙述将结合文字描述、数据流图、变换图等来描述“美丽租”小程序的体系结构设计、接口设计，数据设计等方面。本说明的预期读者有客户、项目经理、开发人员等以及跟该项目相关的其他竞争人员。

## 1.2 准则与限制

为了评价该设计阶段的设计表示的“优劣程度”，必须遵循以下几个准则:  
  
1.软件设计应当表现出层次结构，它应巧妙地利用各个软件部件之间的控制关系。  
2.设计应当是模块化的，即该软件应当从逻辑上被划分成多个部件，分别实现各种特定功能和子功能。  
3.设计最终应当给出具体的模块(例如子程序或过程)，这些模块就具有独立的功能特性。  
4.应当应用在软件需求分析期间得到的信息，后采取循环反复的方法来获得设计。

## 1.3 范围

本概要设计由公司产品经理团队负责制定并授权更改，产品原型设计活动遵循此过程。本规范影响的范围包括：  
产品经理  
项目经理  
UI设计师  
DBA数据库管理员

BA业务顾问

## 1.4 参考资料

①软件工程导论 ---------张海藩 编著 清华大学出版社

②软件架构设计实践教程 ---------张家浩 编著 清华大学出版社

## 1.5 设计规范

设计规范详细阐述本规范的维度、要求和示例等内容，产品原型设计工作者应共同努力使用规范、维护规范和更新规范。设计规范更新是一个不断调整和丰富的动态过程，通过持续更新以符合公司产品原型设计的实际情况，保证企业级、可运营性、安全性、可维护性、扩展性、鲁棒性、专业性、规范性与实用性的统一。  
①小程序的所有页面右上角位置，都固定放置了小程序的菜单，在设计界面时需预留出该区域空间。  
②微信提供了深浅两种配色样式，以便更好的融合到各种风格的页面中，需注意保持小程序菜单清晰的辨识度。  
③如果要在小程序菜单附近放置交互元素，要考虑是否有交互冲突，应尽量避免误触的可能。  
微信有提供小程序的底部标签样式，建议标签数量在2-4个适宜。  
④全局加载是小程序名称左侧的加载图标。模态的加载样式将覆盖整个页面的，由于无法明确告知具体加载的位置或内容将可能引起用户的焦虑感。  
⑤小程序设计应遵循友好、高效、一致的设计原则，每个页面都应有明确的重点，清晰明确地告知用户身在何处、又可以往何处去，确保用户在页面中游刃有余地穿梭而不迷路。

# 2.软件系统实现方案

## 2.1 组网结构

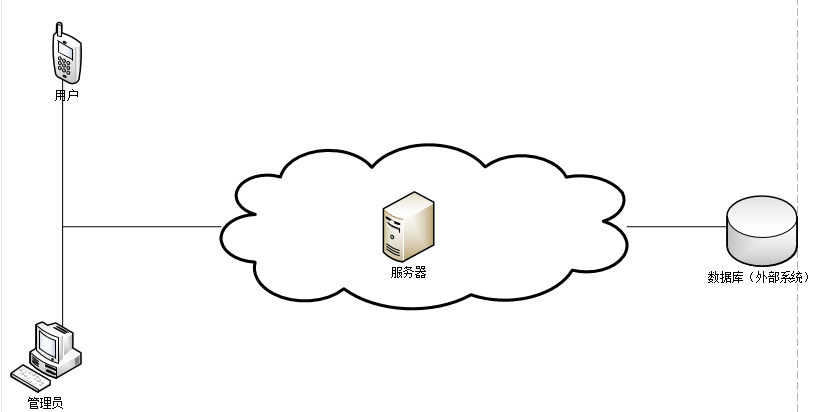


图2.1.1 组网结构图

## 2.2 逻辑结构

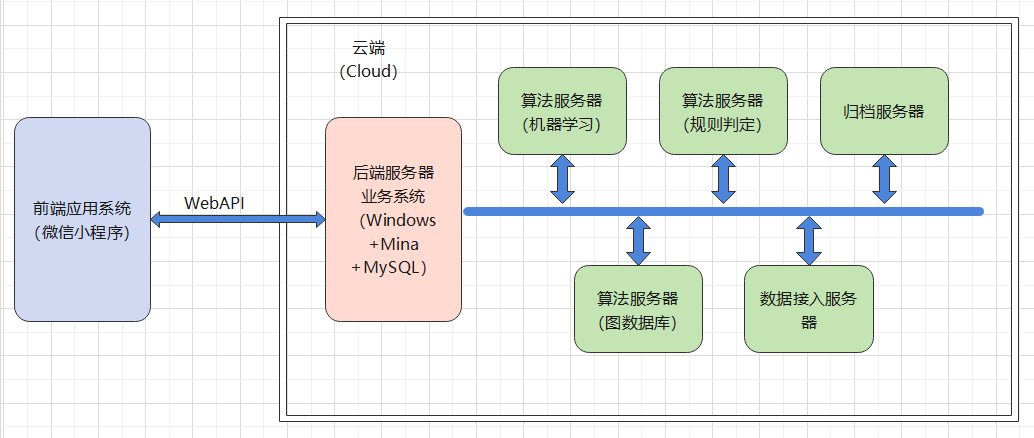


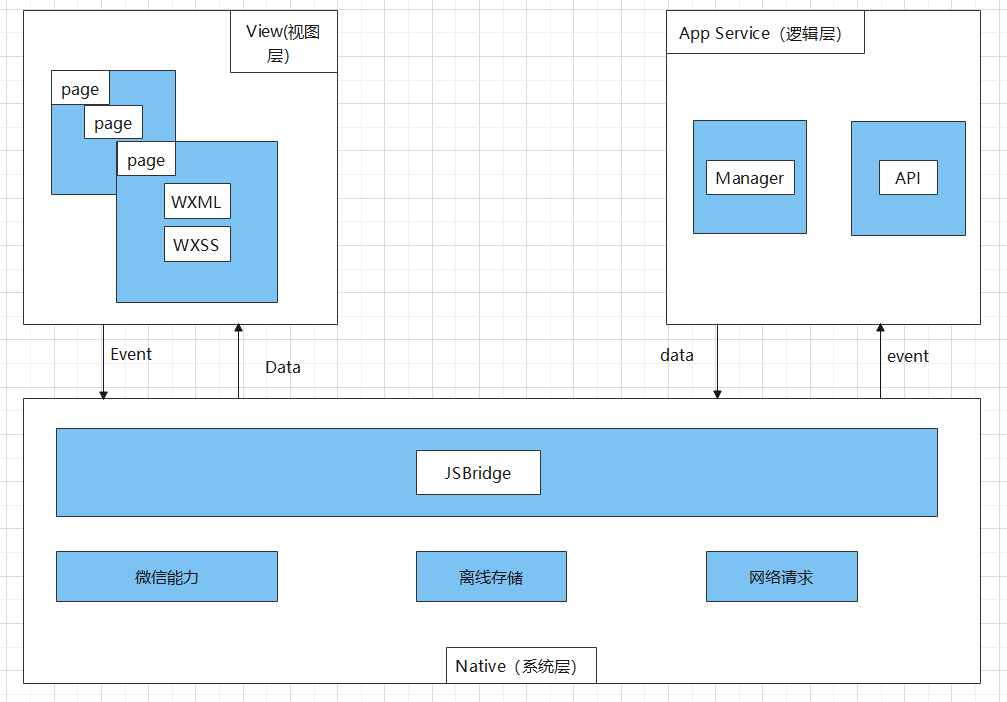
图2.2.1 逻辑结构图

图2.2.2 逻辑结构图之内部结构

小程序分为两个主要部分独立运行：view 模块和 service 模块。在开发者工具中，它们独立运行于不同的 webivew tag 中。  
  
view 模块负责前端界面显示，它由 wxml 和 wxss 转换后代码以及微信提供相关辅助模块组成。 一个 view 模块对应一个 页面, 小程序支持同时多个 view 存在。  
service 模块负责后台逻辑，它由 js 代码以及微信提供的相关辅助模块组成。 一个应用只有一个 service 进程，它同样也是一个页面。它在程序生命周期内后台运行，service 模块通过与 view 模块实现不同但接口格式一样的微信JSBridge 对象跟后台通信。  
视图层和逻辑层分离，通过数据驱动，事件交互，不直接操作DOM；视图层负责渲染页面结构，逻辑层负责逻辑处理、数据请求、接口调用等。  
  
视图层与逻辑层通过数据data和事件event进行通信，逻辑层提供数据给视图层，视图层通过绑定/捕获事件发起交互让逻辑层处理视图使用WebView渲染，JS由JSCore(IOS)/ X5(Android) /nmjs(DevTool)渲染解析

## 2.3 层级结构

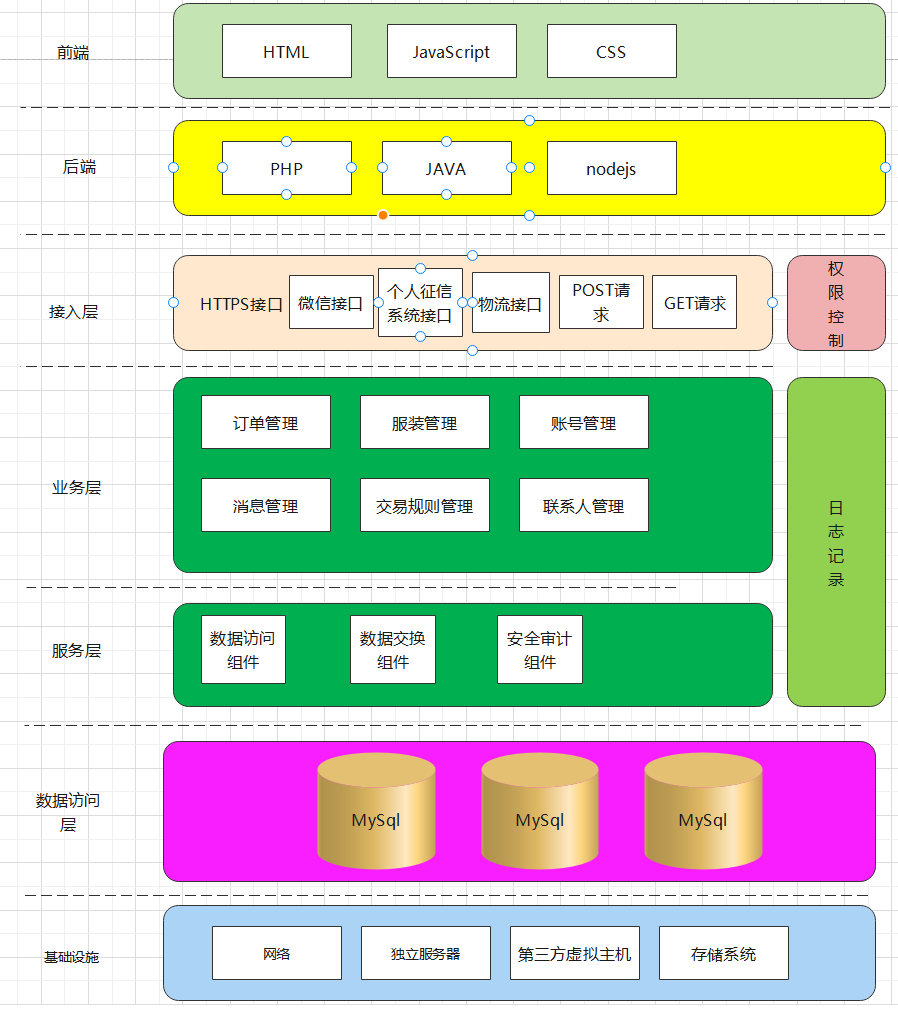


图2.3.1 层次图

## 2.4 子系统划分及功能

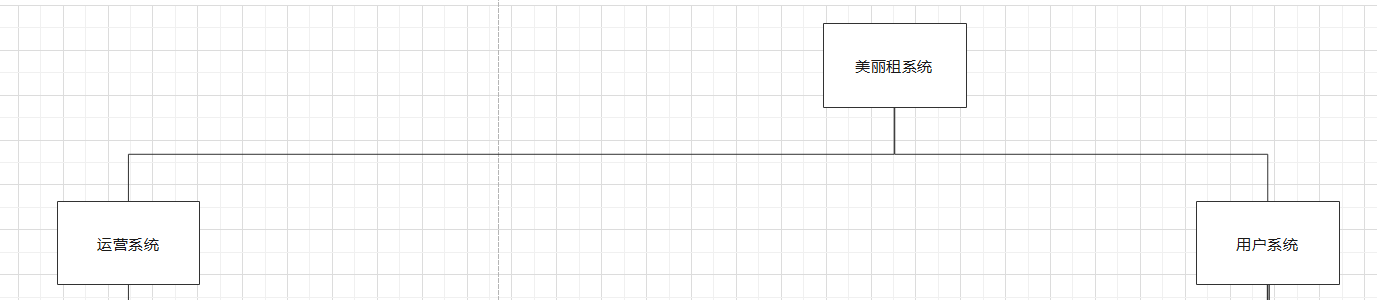


图2.4.1 子系统划分

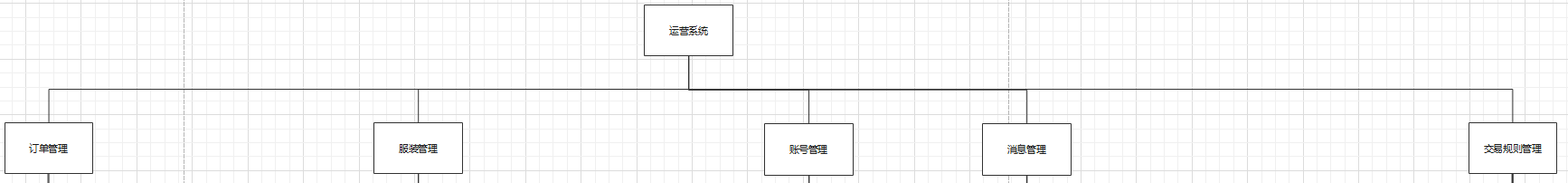


图2.4.2 运营子系统

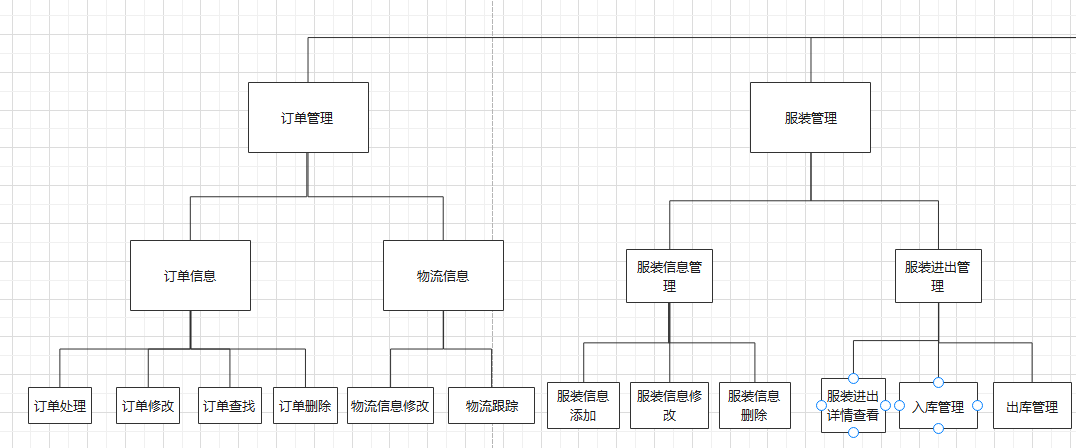


图2.4.3 用户管理子系统及其功能1

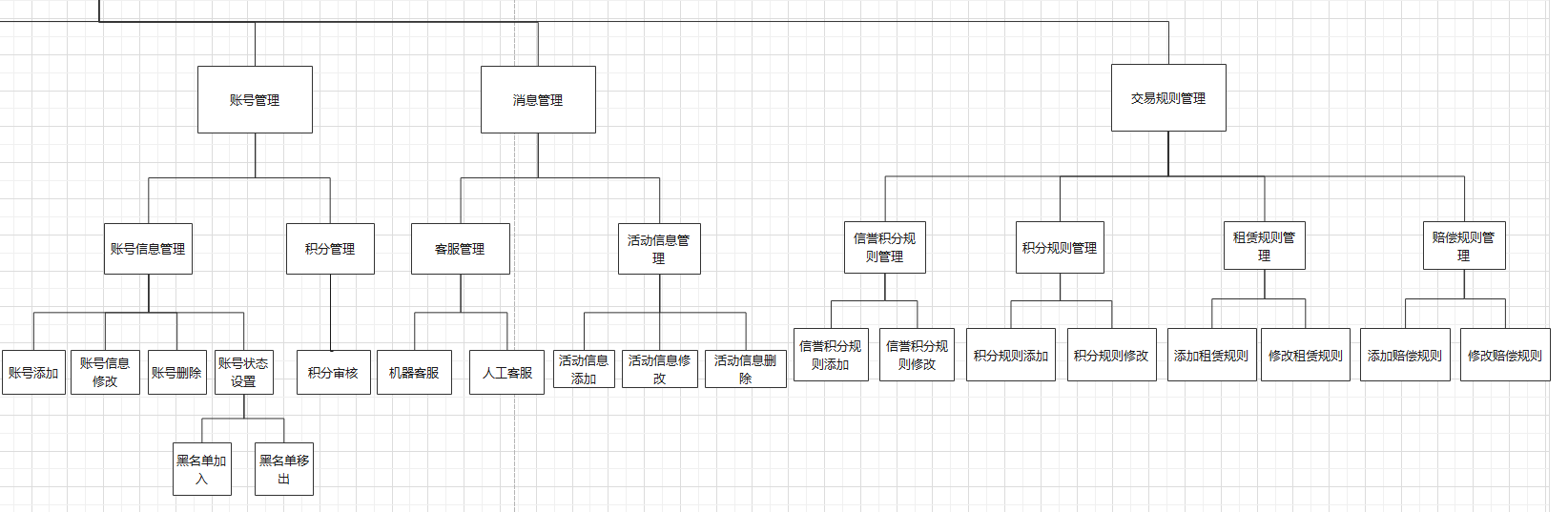


图2.4.4 用户管理子系统及其功能2

## 2.5 数据流图

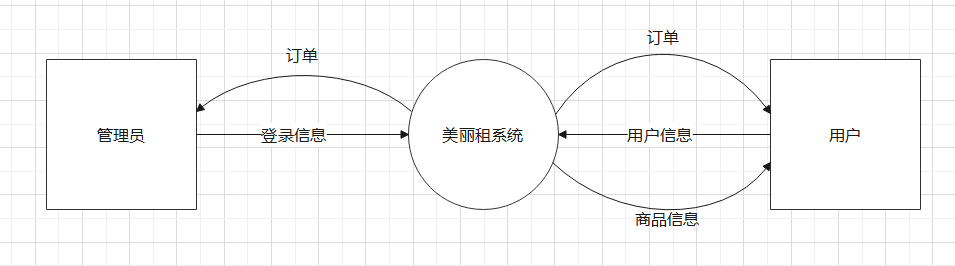


图2.5.1 数据流图第0层

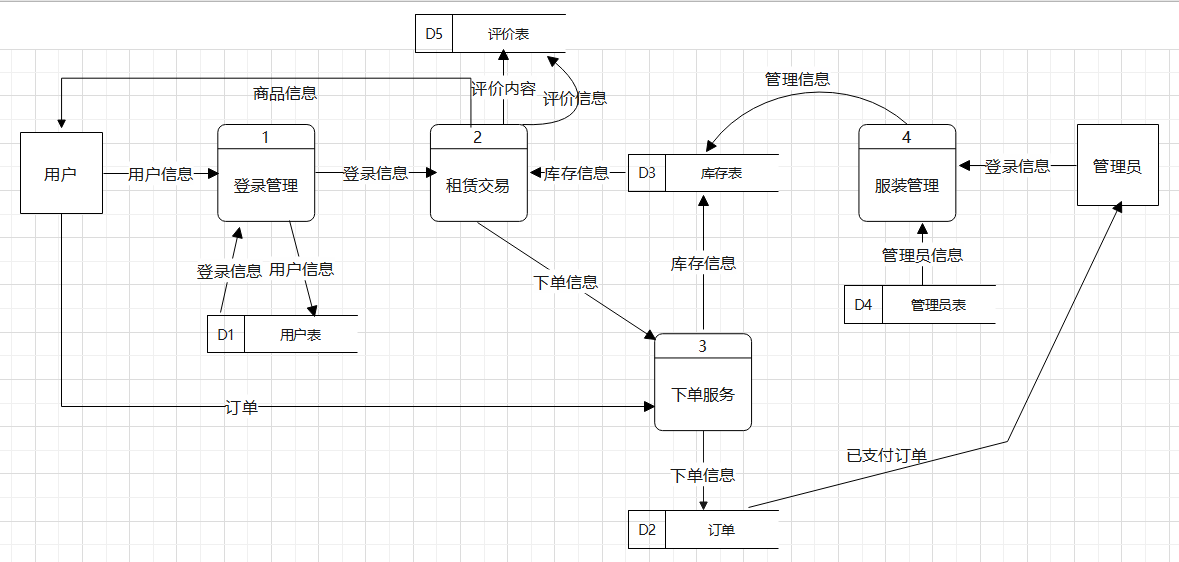


图2.5.2 数据流图第1层

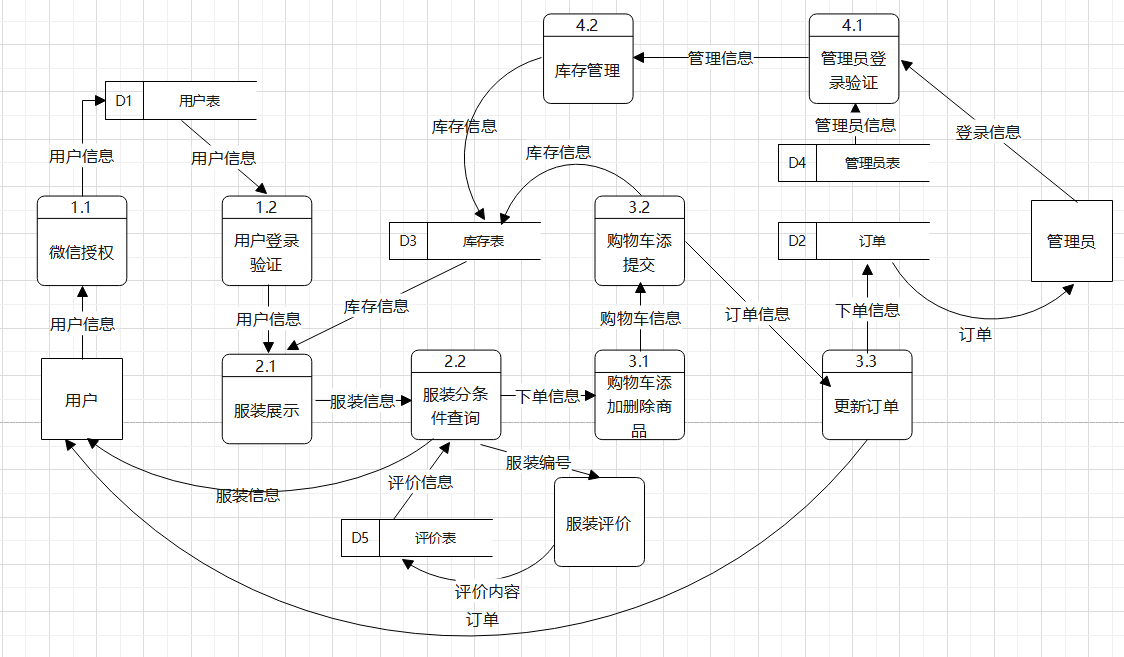


图2.5.3 数据流图第2层

# 3.接口

## 2.1 用户接口

**用户登录界面**：当用户访问小程序时，进入用户登录界面，即微信授权登录页面。

**主界面**：用户进入小程序后，跳转到该界面。该界面放置相应的广告模块，衣服分类专区选择，最近上新商品展示。并且提供搜索衣服的输入栏。

**礼服专区页面：**在界面首页的分类专区点击“礼服”按钮即可进入礼服专区页面；

礼服专区包括有焦点图、礼服励志语、专区服装列表。

**详情信息页面**：在服装专区页面、上新页面或者搜索出的符合关键字的服装页面，点击服装图片或者信息即可进入服装详情页面；

用户点击列表商品图片，跳转到该商品的详细信息页面，该页面显示商品信息图片、名称、详细描述、租金等信息以及是否加入愿望衣橱操作。

**搜索结果显示界面**：通过读取主界面的搜索关键字在商品信息库中匹配相关商品，并将相关商品显示到该界面。

**个人中心界面**：用户在主界面点击个人中心跳转按钮可以跳转到该界面。该界面显示用户的昵称、会员等级、头像、积分数、愿望衣橱、足迹、信誉值等基本信息，并且提供给风格方案（风格、尺码、场合、偏好）、常用功能（足迹、我的地址、签到、设置）、关于我们（联系客服、常见问题、关于我们、评价）

**“风格”页面**:用户设置自己喜欢的风格系统根据用户的设置推荐相关衣物 。  
  
**“尺码”页面:**用户可以设置自己的尺码，该页面需要有“确定”按钮方便用户操作 。  
  
**“场合”页面:**我们会根据衣服适合的场合，给用户推荐 。  
  
**“偏好”页面:**该页面是用户自己设置自己喜欢的衣物，风格以及适用场合 ，该页面需要有确定按钮，方便用户操作 。  
   
**“足记”页面:**该页面显示用户的浏览记录 ，用户可随时删除自己的浏览记录 。  
  
**“我的地址”页面:**该页面是用户设置自己的收货地址页面 ，该页面是提供输入框以及确定或者取消按钮，方便用户操作 。

**签到页面：**用户登录成功后可点击“签到”按钮可跳转到签到页面，该页面显示用户签到日期以及签到次数，用户点击对应日期可完成签到，该页面需要有“确定”按钮方便用户操作。  
  
**设置界面：**用户点击“设置”按钮可跳转到设置界面，该界面显示用户账号管理，包括用户名、地址设置、用户头像设置、账户与安全、支付设置等功能，并提供给用户“返回”按钮跳转贵个人中心界面  
  
**“关于我们”页面:**是关于该校程序的一些详细介绍以及注意事项 ，该页面可以跳转到首页。  
  
**“联系客服”页面:**用户可点击衣服详情页底端联系客服跳转到该页面 ，并进行衣服的详细询问，该页面中会有人工客服 回答用户的问题，该页面需提供输入框以及“发送”按钮 。  
  
**“评价”页面:**该页面是对用户租借过的衣物进行评价，用户可选择评价或不评价，也可以随时删除自己的评价 。该页面需提供输入框以及确定按钮。  
  
**“常见问题”页面:**该页面是用户询问机器客服的页面，该页面中需提供输入框 ，用户群问的问题，会有机器客服给出答案 。

**客服页面**：用户点击“消息中心”按钮可跳转到该页面，“消息中心”页面可以查看商家的广告推送自己好友分享的链接，也可以在消息中心页面找到官方客服“小猪”。该页面通过输入框输入信息，点击“发送”按钮进行发送，客服对用户问题进行相应回答。

**下单页面**：用户在详情信息页面点击“租赁”按钮跳转到下单页面，该页面用户通过点击按钮进行尺码、数量选择，选择完毕点击“确定”按钮进行下单操作。  
  
**联系人页面：**点击消息中心页面右上方的人像按钮即可进入联系人页面心页面；

联系人页面包括有联系人消息条（自上而下介绍）。

**信誉积分页面：**在个人中心页面点击“信誉值”按钮即可进入信誉积分页面；

信誉积分页面包括有用户目前的信誉值、邀请好友得积分（用于提升账号等级）、信誉积分明细；

## 2.2 外部接口

硬件接口：  
  
  系统运行服务器要求运行内存大于1G，存储空间大于2G；  
  
软件接口：  
  
  微信开发者工具、MySQL8.0，安卓手机端。

## 2.3 内部接口

### 2.3.1 内部模块间关系

2.3.1.1系统总模块：

Main（）:void

2.3.1.2系统主控页面：

（1）首页：HomePage（）:void

搜索框页面：Search（index：string…）:list<question>

广告页面：Advertising():void

礼服专区页面：PrefectureFormal():void

汉服专区页面：PrefectureHanfu():void

职业装专区页面：PrefectureCareer():void

cos专区页面：PrefectureCos():void

今日上新页面：NewArrival():void

（2）个人中心页面：

个人信息设置页面：InfoUpdata(info:string…):void

积分页面：Integral():void

愿望衣橱页面：Collection():void

信誉值页面之查看页面：Credit():void

风格页面：Style():void

尺码页面：Size():void

场合页面：Occasion():void

偏好页面：Preference():void

足迹页面：Records():void

我的地址页面之新建地址：LocationUpdata(location:string…):void

我的地址页面之删除地址：LocationDelete(add:string…):list<add>

我的地址页面之修改地址：LocationChange(add:string…):list<add>

签到页面：Signin():void

关于我们：AboutUs():void

联系客服：ContanctService():void

常见问题：Seequestion(answered:list):list<answer>

（3）消息中心：

好友联系页面：Friends():void

交易物流消息提醒页面：Logistics():void

机器人小猪页面：ServiceRobot():void

机器人评价页面：ServiceRobotEvaluate():void

（4）订单页面（管理员）：

订单查看页面:OrderSee(order:string…):list<order>

订单修改页面：OrderChange(order:string…):list<order>

订单删除页面：OrderDelete(order:string…):list<order>

### 2.3.1 接口数据描述

管理员（Administrator）:账号、密码、姓名。



图2.3.1.1 管理员数据描述

订单信息（order）：订单编号、用户微信号、支付状态、下单日期、地址、发货时间、收货时间。



图2.3.1.2 订单数据描述

商品信息（close）：衣服编号、衣服名字、衣服类型、新旧程度、衣服信息、租金、押金、衣服图片。

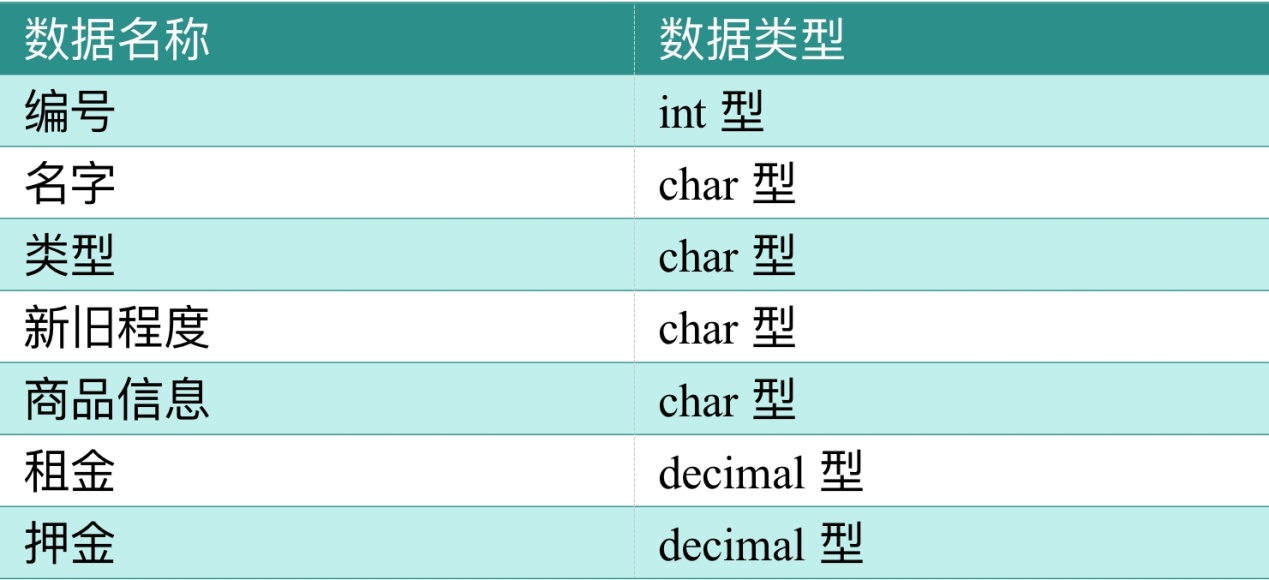


图2.3.1.3 商品数据描述

用户（user）:包括用户微信号、昵称、性别、地址、联系方式。



图2.3.1.4 用户数据描述

库存信息（inventory）：衣服编号、衣服名字、small码库存量、middle库存量、large库存量。



图2.3.1.5 库存数据描述

# 4.开发技术

## 4.1 开发框架

小程序提供的开发框架为MINA框架，它类似于淘宝Weex、Vue框架。MINA框架通过封装微信客户端提供的文件系统、网络通信、任务管理、数据安全等基础功能，对上层提供一整套JavaScript API,让开发者能够非常方便地使用微信客户端提供的各种基础功能与能力，快速构建一个应用。

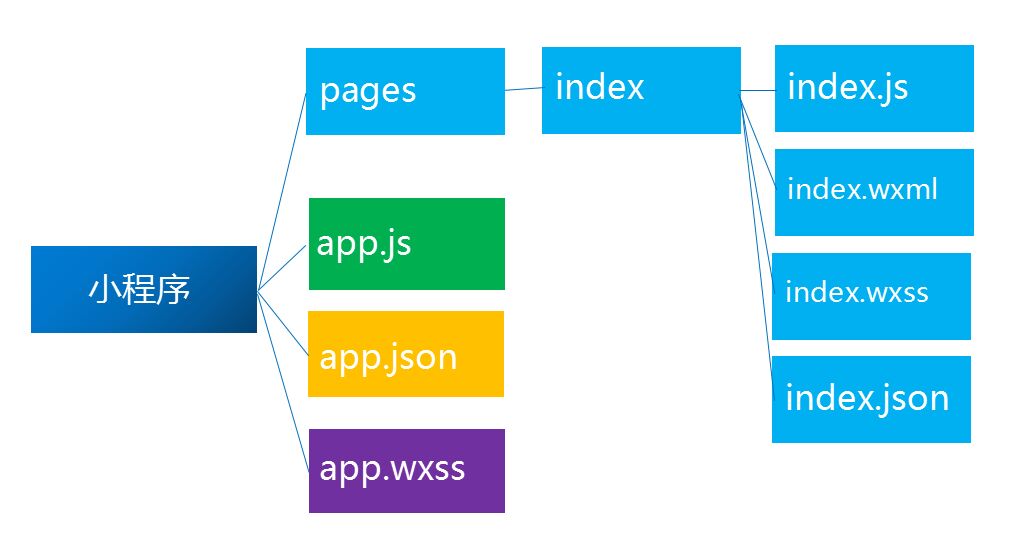


图4.1.1 小程序框架结构

一个小程序主体部分由三个文件组成，必须放在项目的根目录，如下：



图4.1.2 小程序主体

pages目录下为小程序页面，每个页面须建立独立的文件夹，例如上面生成的小程序有两个页面，分别为pages/index、pages/logs，一个页面由四个文件组成，分别是：



图4.1.3 页面的四个文件



图4.1.4 app.json提供的配置项

## 4.2 开发技术介绍

开发语言：WXML、WXSS、JavaScript、Json、SQL

数据对接：微信原生API、中国个人征信接口、物流接口，为微信端各项动态数据操作提供的功能接口

开发环境：微信开发者工具

数据库：MySQL，存放各种数据，需要根据业务需求创建各种数据表。  
后台管理系统：管理系统是为PC端的平台提供的WEB管理工具，功能对应与微信小程序的各项功能，如订单管理等等。

## 4.3 开发技术评估

1.小程序仍然使用WebView渲染，并非原生渲染  
2.需要独立开发，不能在非微信环境运行。  
3.开发者不可以扩展新组件。  
4.服务端接口返回的头无法执行，比如：Set-Cookie。  
5.依赖浏览器环境的js库不能使用，因为是JSCore执行的，没有window、document对象。  
6.WXSS转化成js 而不是css，为了兼容rpx。  
7.WXSS不支持级联选择器。  
8.小程序无法打开页面，无法拉起APP。

9.征信接口会崩溃。

# 5.开发技术

场景1：管理员或普通人员删除数据操作  
例如：以下（修复待验证安全问题）删除数据请求，通过参数codes的值，来删除数据。如果没有验证登陆的用户是否有权限删除codes，那么通过遍历codes值，可以删光所有codes数据。这种问题在一般系统应该比较常见。后续修改成本都很高，有些可能修复不了，涉及框架、业务设计都需要修改。应在设计时就考虑：验证用户合法性、权限  
防止用户误操作，如删除有提示  
防止恶意操作，如删除3条数据以上，需要输入验证码才能删除成功（需要跟踪业务需求来设计）

系统维护设计 ：  
由于系统较小没有外加维护模块，仅靠数据库的一些基本维护措施即可。为便于维护，应该设计了三种日志：系统运行日志、操作日志、出错日志。三种日志根据不同的重要程度采取存放在文件和数据库的方式，系统管理员可以很轻松的监控系统运行情况，数据表的建立和删除有数据系统管理员予以维护。

如下是最常见的由设计不当产生的安全漏洞分类：  
1、输入验证漏洞：嵌入到查询字符串、表单字段、cookie 和 HTTP 头中的恶意字符串的攻击。这些攻击包括命令执行、跨站点脚本(XSS)、SQL 注入和缓冲区溢出攻击。  
2、身份验证漏洞：标识欺骗、密码破解、特权提升和未经授权的访问。  
3、授权漏洞：非法用户访问保密数据或受限数据、篡改数据以及执行未经授权的操作。  
4、敏感数据保护漏洞：泄露保密信息以及篡改数据。  
5、日志记录漏洞：不能发现入侵迹象、不能验证用户操作，以及在诊断问题时出现困难。  
对于以上的漏洞，可用的防范措施有：  
1、针对输入验证漏洞，在后台代码中必须验证输入信息安全后，才能向服务层提交由用户输入产生的操作。  
2、针对身份验证漏洞，程序设计中，用户身份信息必须由服务器内部的会话系统提供，避免通过表单提交和页面参数的形式获取用户身份。  
3、针对授权漏洞，在访问保密数据或受限数据时，一定要根据用户身份和相应的权限配置来判断操作是否允许。  
4、针对敏感数据漏洞，在储存敏感数据时，一定要采用合适的加密算法来对数据进行加密。  
5、针对日志记录漏洞，程序设计中，对改变系统状态的操作，一定要记录下尽可能详细的操作信息，以便操作记录可溯源。  
  
 就程序部署及操作系统安全性而言，可用以下的防范措施：  
1、无论部署于何种操作系统，需要保证操作系统在部署前，安装了全部的安全升级补丁，关闭了所有不需要的系统服务，只对外开放必须的端口  
2、定期查看所部署服务器系统安全通告，及时安装安全补丁。  
3、定期检查系统日志，对可疑操作进行分析汇报。  
4、应用服务器程序在服务器中文件系统中的目录结构位置应该尽量清晰。目录命名需要尽可能的有意义。  
5、应用服务器程序不能以具有系统管理员权限的操作系统用户运行。最好能建立专门的操作系统用户来运行应用服务器。  
  
 就数据库安全性而言，可用以下的防范措施：  
1、数据库监听地址要有限制，只对需要访问的网络地址进行监听。  
2、定数据库备份制度。定期备份库中的数据。  
3、数据库操作授权限制，对表一级及其以上级别的数据库操作授权不应对应用服务器开放。  
  
 就网络安全性而言,可用以下的防范措施：  
1、选用企业级防火墙。  
2、根据具体网络环境，制定尽可能周密的防火墙规则。  
3、需要在外网中传输的数据，应选用合适的加密算法进行加密。  
  
 就物理安全性而言，可用以下的防范措施：  
1、服务器应部署于专业的数据机房，做好机房管理工作。  
2、对于支持热插拔的各种接口，需要在部署前在系统BIOS中关闭。服务器在运行过程中，应该做好各种防护措施。

# 6.系统出错处理设计

6. 1出错信息

|  |  |  |
| --- | --- | --- |
| 出错或故障 | 系统输出信息 | 处理方法 |
| 服务器有问题 | 系统提示无法正确连接服务 | 暂停服务 |
| 系统故障 | 突然死机或黑屏 | 使用恢复子系统自动恢复 |

6. 2补救措施  
  
a.定期进行数据备份，每天进行增量转储，每月一次进行海量转储，可使用静态转储也可使用动态转储;  
  
b.若发生服务器崩溃的致命性错误，需要进行手工操作，并记录下所需要记录的数据，当系统恢复正常工作时，再把在此期间记录的数据重新放入系统中;  
  
c.使用具有检查点的恢复技术，检查点记录的内容包括:建立检查点时刻所有正在执行的事务清单;这些事务最近一一个日志记录的地址。利用系统故障恢复的方法(即REDO和UNDO),同时建立检查点，保存数据库状态，具体步骤为:1)将当前日志缓冲中的所有日志记录写入磁盘的日志文件上; 2)在日志文件中写入一个检查点记录; 3) 将当前数据缓冲的所有数据记录写入磁盘的数据库中; 4)把检查点记录在日志文件中  
  
的地址写入一个重新开始文件。系统出现故障时恢复子系统将根据事务的不同状态采取不同的恢复策略。  
  
6.3系统维护设计  
  
专用模块为:系统备份模块和系统恢复模块  
  
系统备份模块即为系统自动拷贝模块，在每天都进行一-次动态增 量转储;