

# 电商营销平台

# 总体设计说明书



实验课程名称 软件工程基础

专 业 班 级 计算1501&计算1502

学 号 31501324 31501098 31501322

学 生 姓 名 陈董锴 吴安之 吕莉

实验指导老师 杨枨

[电商营销平台 1](#_Toc499484205)

[总体设计说明书 1](#_Toc499484206)

[版本控制 3](#_Toc499484207)

[1引言 4](#_Toc499484208)

[1.1编写目的 4](#_Toc499484209)

[1.2背景 4](#_Toc499484210)

[1.2.1 项目名称及开发成员 4](#_Toc499484211)

[1.2.2项目的用户 4](#_Toc499484212)

[1.2.3 项目主要承担部门 4](#_Toc499484213)

[1.2.4 项目建设背景 4](#_Toc499484214)

[1.3定义 5](#_Toc499484215)

[1.4参考资料 5](#_Toc499484216)

[2总体设计 5](#_Toc499484217)

[2.1功能需求规定 5](#_Toc499484218)

[2.2性能需求规定 6](#_Toc499484219)

[2.2.1时间特性要求 6](#_Toc499484220)

[2.2.2灵活性 6](#_Toc499484221)

[2.2.3输入输出要求 6](#_Toc499484222)

[2.3基本设计概念和处理流程 6](#_Toc499484223)

[2.3.1数据流图 6](#_Toc499484224)

[2.3.2系统流程图 13](#_Toc499484225)

[2.4结构 13](#_Toc499484226)

[2.4.1状态转换图 13](#_Toc499484227)

[2.4.2HIPO图 14](#_Toc499484228)

[2.5功能需求与程序的关系 14](#_Toc499484229)

[2.6人工处理过程 15](#_Toc499484230)

[2.7尚未解决的问题 15](#_Toc499484231)

[3接口设计 15](#_Toc499484232)

[3.1用户接口 15](#_Toc499484233)

[3.2外部接口 15](#_Toc499484234)

[3.3内部接口 15](#_Toc499484235)

[4运行设计 16](#_Toc499484236)

[4.1运行模块组合 16](#_Toc499484237)

[4.2运行控制 21](#_Toc499484238)

[4.3运行时间 21](#_Toc499484239)

[5系统数据结构设计 21](#_Toc499484240)

[5.1数据库模型 21](#_Toc499484241)

[5.2数据字典 22](#_Toc499484242)

[6系统出错处理设计 23](#_Toc499484243)

[6.1出错信息 23](#_Toc499484244)

[6.2补救措施 23](#_Toc499484245)

[6.3系统维护设计 23](#_Toc499484246)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本控制 | | | | | |
| 版本 | 修改日期 | 修改问题 | 跟踪情况 | 修改人 | 审核人 |
| V0.1 | 2017.11.25 | 初步完成总体设计说明 | 已跟踪 | 吕莉 | 陈董锴 |

**概要设计说明书**

# 1引言

## 1.1编写目的

概要设计主要是利用比较抽象的语言对整个需求进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，人机界面，实现对系统的初步设计。我们根据需求分析得到的数据流图，将之转化为软件结构和数据结构，建立起目标系统的逻辑模型。使软件编程人员能对目标系统有一致的认识。

## 1.2背景

### 1.2.1 项目名称及开发成员

|  |  |
| --- | --- |
| 项目名称 | 个体移动端销售平台 |
| 提 出 者 | 陈董锴 |
| 开 发 者 | 陈董锴 吴安之 吕莉 |
| 实现中心 | 个人电脑 |

### 1.2.2项目的用户

原本使用朋友圈推送方式售卖商品的具体个体卖家

### 1.2.3 项目主要承担部门

浙江大学城市学院计算机15级 G03小组

### 1.2.4 项目建设背景

目标用户：

一级用户:基于微信社交平台进行销售商品的微商店主。

二级用户:微商店主的现有顾客以及潜在顾客。

背景：

现行的交易方式中，全部是通过朋友圈发送消息分享商品，消息被好友浏览到有可能购买，但是需要私聊交流，在此流程上有麻烦，如果未能浏览到消息则就此错过此商品信息。

暴露的问题：

信息交流不明确，也有可能额外流失用户。未能及时浏览到商品信息的好友则是另一部分额外流失的用户，用户间的区分也不明确。

因此希望通过开发一个基于微信社交平台的移动端的电商售卖平台解决此问题。

## 1.3定义

物理模型：主要包含数据仓库的软硬件配置，资源情况以及数据仓库模式。

逻辑模型：系统数据的逻辑结构。

系统流程图：用于描绘物理系统的工具。

数据流图：结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

数据字典:数据字典是关于数据的信息的集合，也就是对数据流图中包含的所有元素的定义的集合。

HIPO图：在层次结构图的基础上推出的一种描述系统结构和模块内部处理功能的工具

额外流失：本来可以购买商品的用户的用户因为没有发现商品或其他原因没有购买造成的流失。

卖家=用户：我们项目的对象

二级用户：卖家（用户）的客户

## 1.4参考资料

【1】《软件工程导论》 张海藩 牟永敏 清华大学出版社

【2】该报告引用ISO软件工程项目开发计划模板

【3】《人月神话》FrederickP.Brooks.Jr 清华大学出版社

【4】《PMBOK》美国项目管理学会

【5】《SWEBOK》IEEE计算机学会职业实践委员会2项目概述

# 2总体设计

## 2.1功能需求规定

## 2.2性能需求规定

### 2.2.1时间特性要求

响应时间：立即(延迟200毫秒以内)

更新处理时间：1秒以内

数据的转换和传送时间：1秒以内

### 2.2.2灵活性

服务器可升级。

购买方式可以进行多渠道的完成，可以通过网页上清算购物车也可以通过公众号完成交易。

如果微信公众号在后续工作中完成了认证，就可以把网站迁移到微网站上。

### 2.2.3输入输出要求

用户账号最多16个字节且不可重复

用户密码最多16个字节

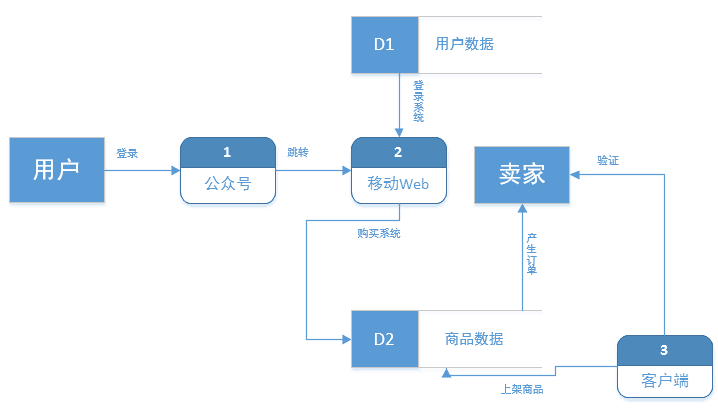
购买数量不能超过库存量

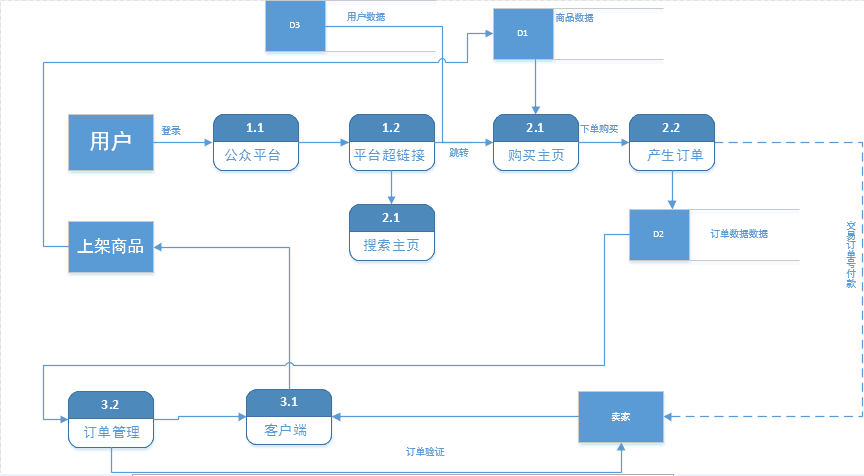
## 2.3基本设计概念和处理流程

### 2.3.1数据流图

1. 系统级数据流图

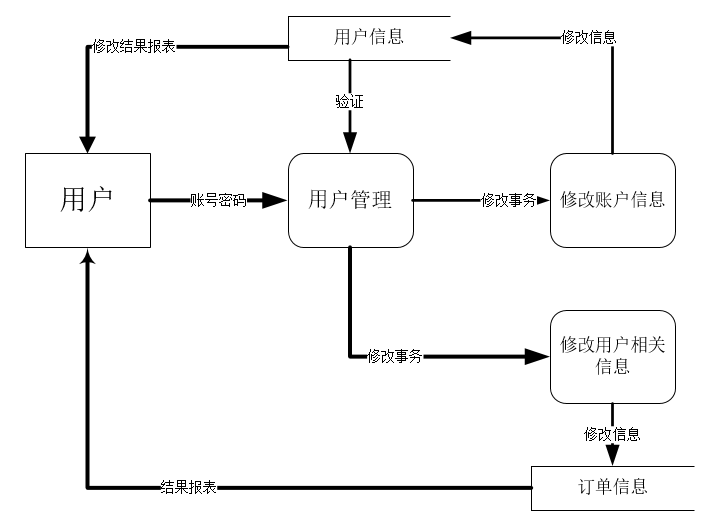




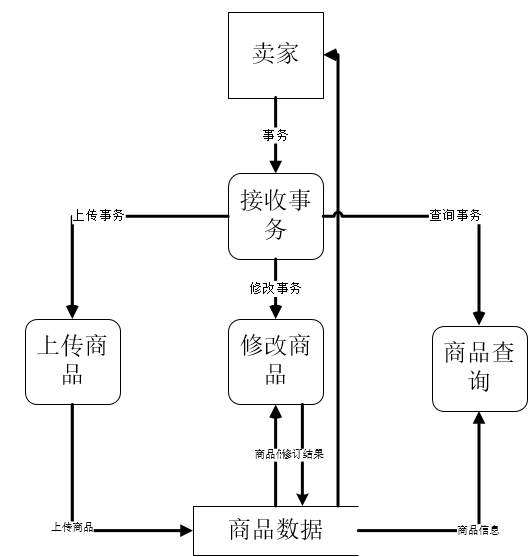


(2)各模块数据流图：

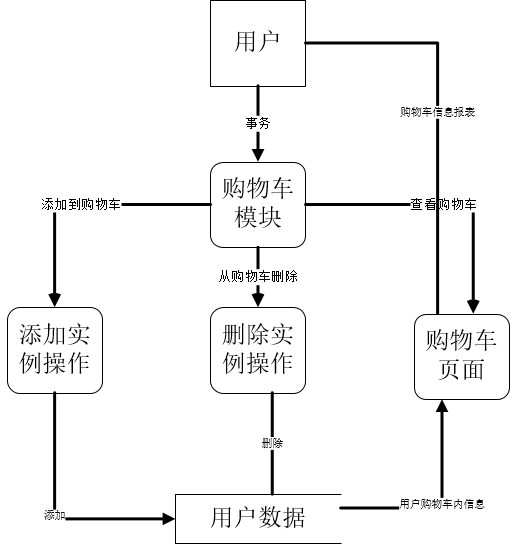
1.用户管理模块：



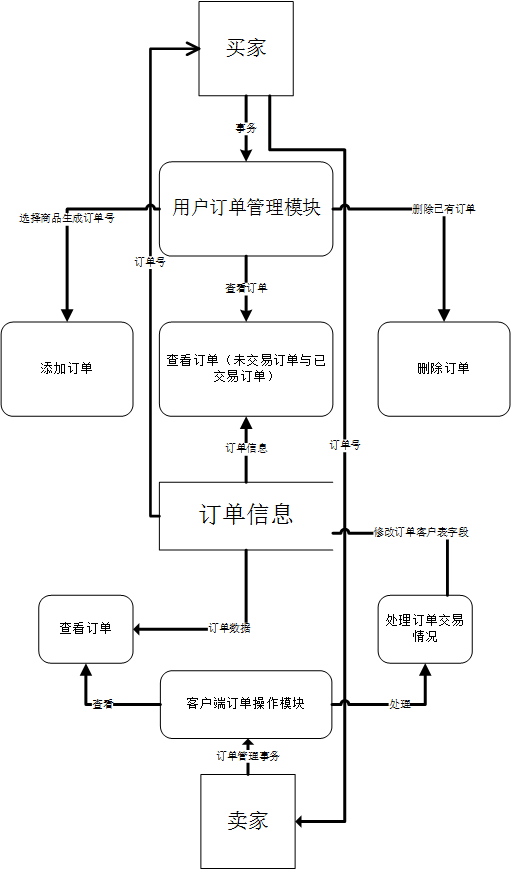
1. 商品管理模块：



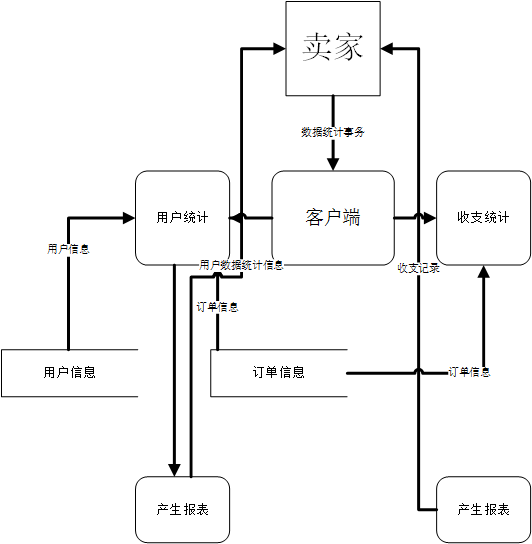
3.购物车模块：



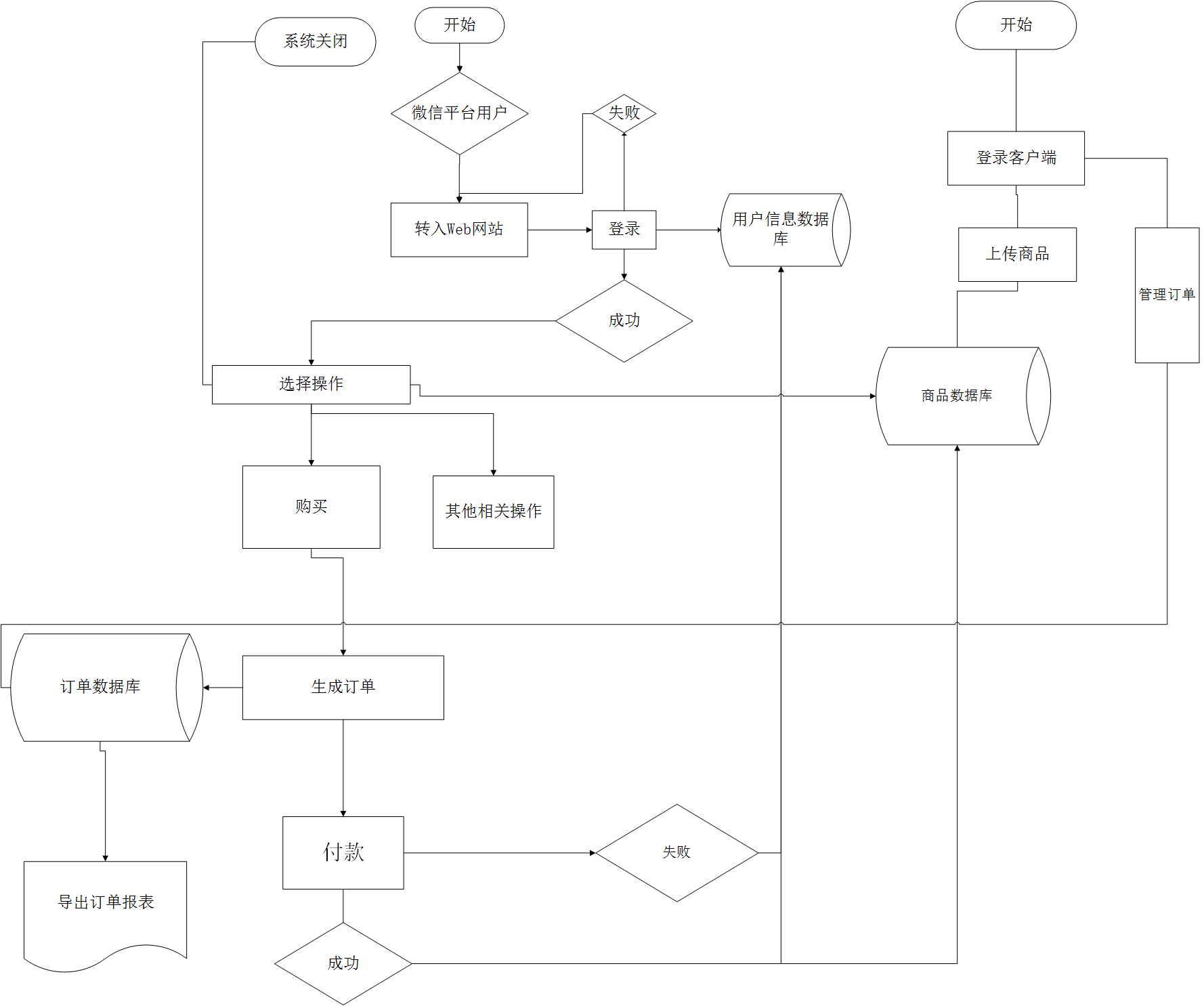
1. 订单管理模块：



1. 数据信息管理：



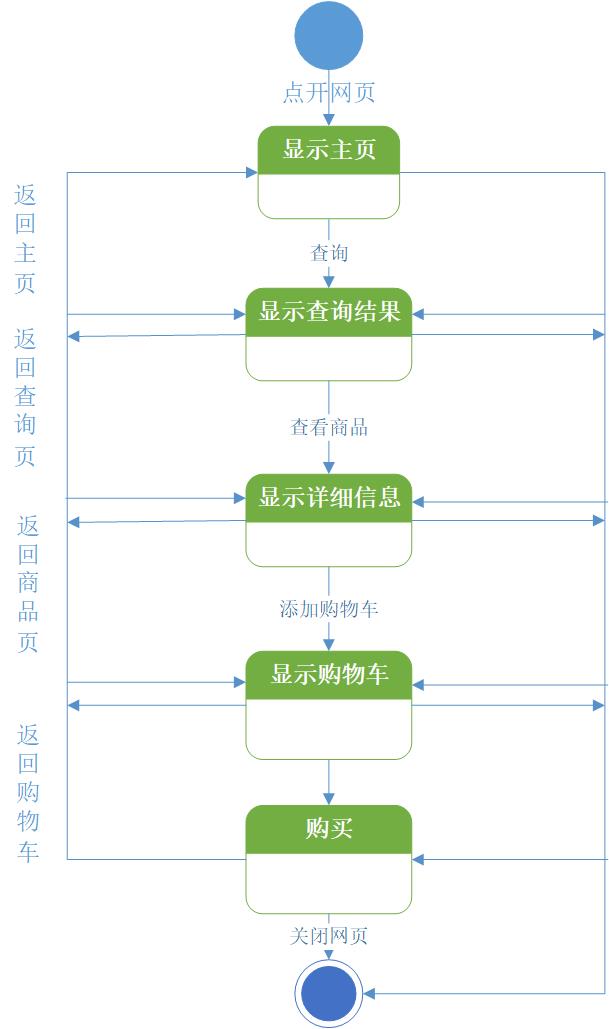
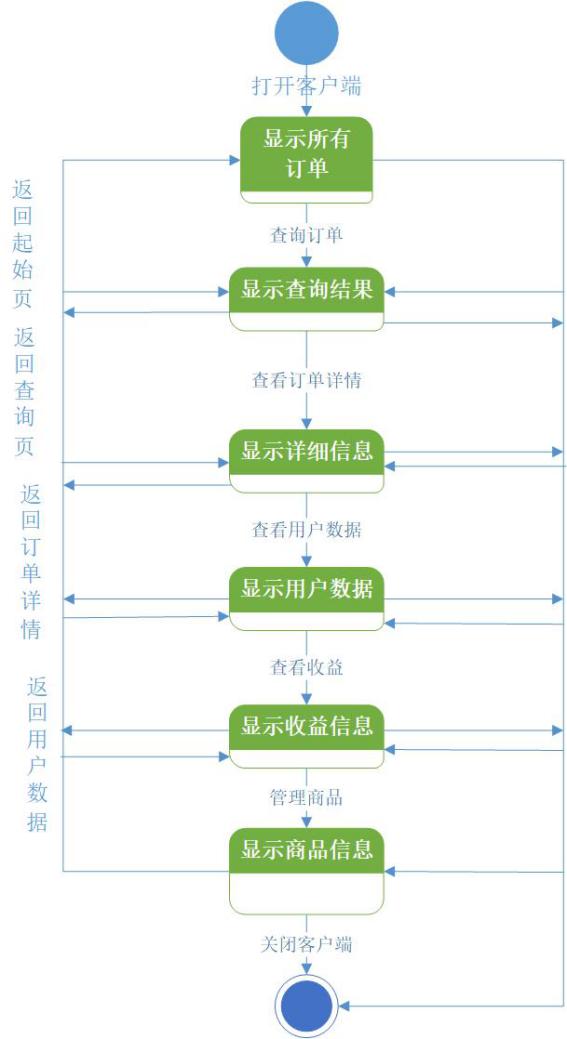
### 2.3.2系统流程图



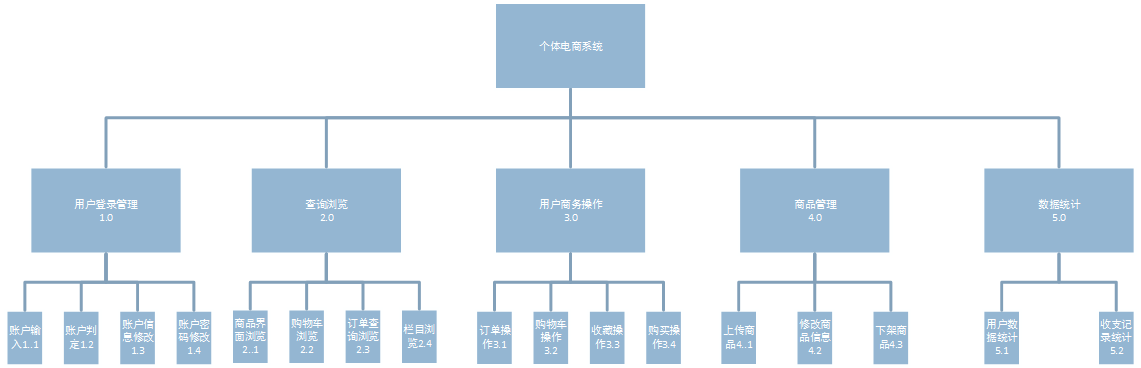
## 2.4结构

### 2.4.1状态转换图

网页： 客户端：

### 2.4.2HIPO图



## 2.5功能需求与程序的关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 用户管理 | 商品管理 | 购物车 | 订单管理 | 数据管理 |
| 注册账号 | √ |  |  |  |  |
| 登录账号 | √ |  | √ | √ |  |
| 修改密码 | √ |  |  |  |  |
| 添加收货地址 | √ |  |  | √ |  |
| 修改收货地址 | √ |  |  | √ |  |
| 删除收货地址 | √ |  |  |  |  |
| 添加商品 |  | √ |  |  | √ |
| 删除商品 |  | √ |  |  | √ |
| 修改商品 |  | √ |  |  | √ |
| 查询商品 |  | √ |  |  | √ |
| 添加购物车 |  |  | √ |  |  |
| 删除购物车 |  |  | √ |  |  |
| 查看购物车 |  |  | √ |  |  |
| 提交订单 |  |  |  | √ |  |
| 修改订单 |  |  |  | √ | √ |
| 查看订单 |  |  |  | √ | √ |
| 删除订单 |  |  |  | √ | √ |
| 收支信息查询 |  |  |  |  | √ |
| 用户数据统计 |  |  |  |  | √ |

## 2.6人工处理过程

需要店家自己手动添加商品，手动完成订单的交易即金钱的支付。

## 2.7尚未解决的问题

暂无

# 3接口设计

## 3.1用户接口

说明将向用户提供的命令和它们的语法结构，以及软件的回答信一级用户：采用C/S架构，用户界面的具体细节将在概要设计文档中描述。

二级用户：微信公众平台

## 3.2外部接口

微信公众平台

## 3.3内部接口

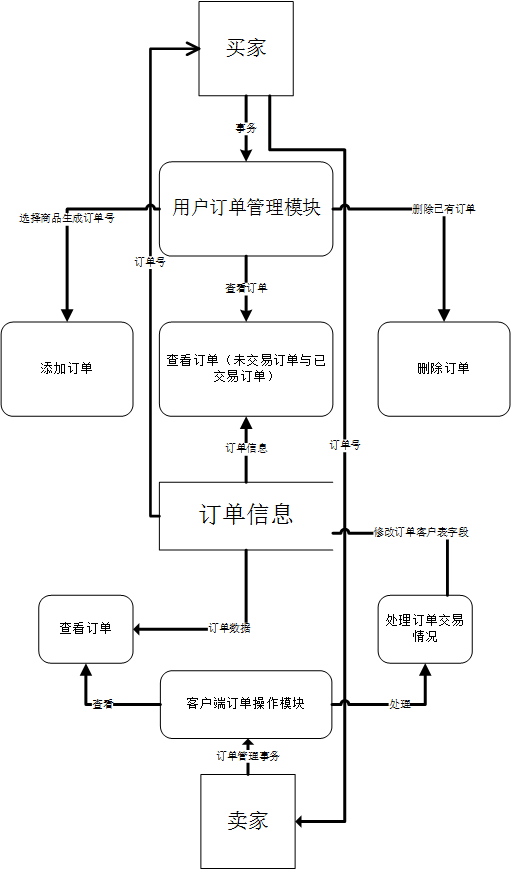
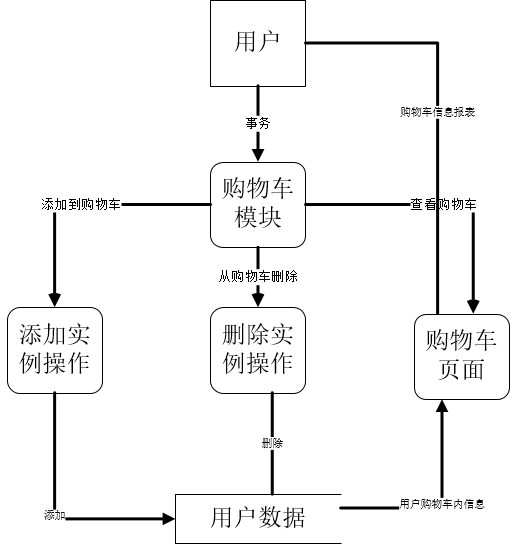
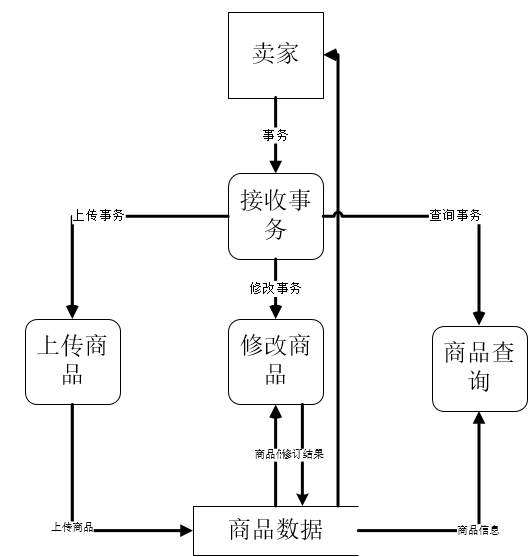
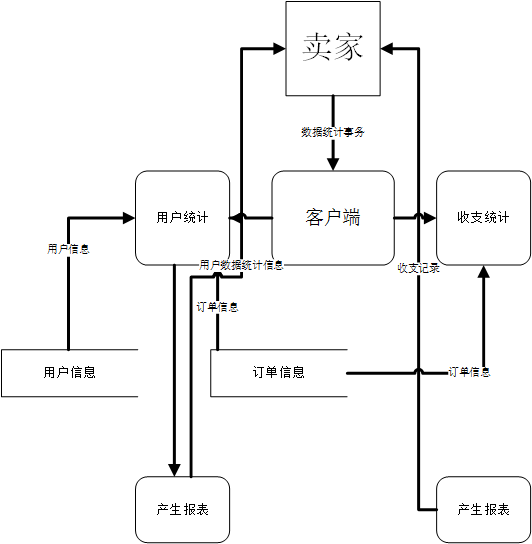
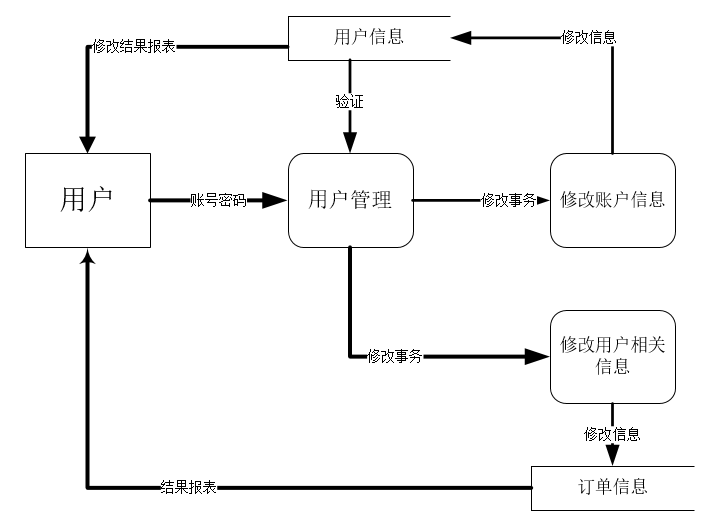
1. 微信公众平台与网页之间的接口

2. 用户访问浏览商品，查看购物车，查看订单情况这些内容与数据的接口

3. 店家在本地客户端上获取实时数据，即客户端与数据库的接口。

# 4运行设计

## 4.1运行模块组合



## 4.2运行控制

1.买家（二级用户）通过关注的微信公众平台，点击相关链接，页面跳转进入网站主页，在主页上用户可进行上述与用户相关的各类操作。买家在进行交易时，先确定订单，数据库录入订单信息，用户得到订单序列号，再将金额通过微信转账的方式转给卖家完成交易。

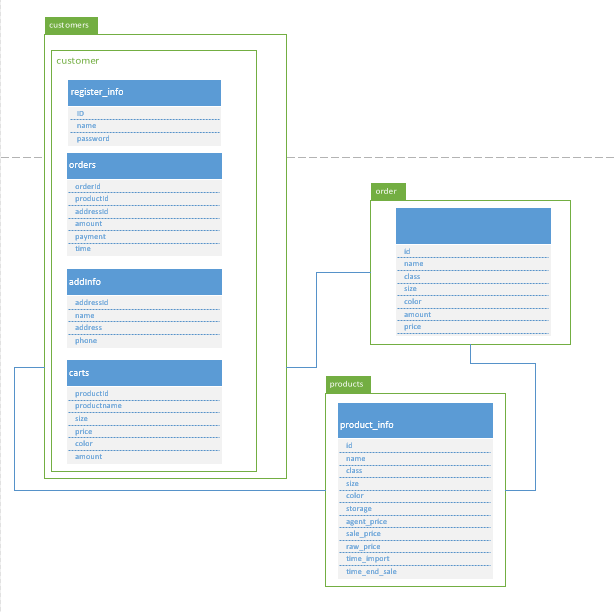
2. 卖家(一级用户)通过提供的客户端，可以随时登陆客户端查询已注册用户的交易情况，查询商品信息，并且可以完成商品的上架/下架。

## 4.3运行时间

每个模块响应时间在1秒钟内。

# 5系统数据结构设计

## 5.1数据库模型



## 5.2数据字典

-----------------------------------------------------------------

用户信息

名称：用户信息

别名：

描述：用来存储每一个用户与该系统相关行为需要的数据集合

定义：用户信息=用户编号+姓名+联系方式+收货地址

位置：注册

获取买家信息

------------------------------------------------------------------

商品信息

名称：商品信息

别名：

描述：用来存储每一个商品与该系统相关行为需要的数据集合

定义：商品信息=商品编号+名称+价格+详情

位置：上架

获取商品信息

------------------------------------------------------------------

订单信息

名称：订单信息

别名：

描述：用来存储每一次用户交易与系统相关行为需要的数据集合

定义：订单信息=订单编号+用户编号+商品编号+数量+价格+收货地址

位置：生成订单数据记录

------------------------------------------------------------------

名称：用户编号

别名：

描述：唯一地标识用户信息中一个特定用户的关键域

定义：用户编号=12｛数字｝12

位置：用户信息、订单信息

------------------------------------------------------------------

名称：商品编号

别名：

描述：唯一地标识商品信息中一个特定商品的关键域

定义：用户编号=12｛数字｝12

位置：商品信息、订单信息

------------------------------------------------------------------

名称：订单编号

别名：

描述：唯一地标识订单信息中一个特定订单的关键域

定义：订单编号=12｛数字｝12

位置：订单信息

# 6系统出错处理设计

## 6.1出错信息

|  |  |  |
| --- | --- | --- |
| 错误类型 | 原因 | 解决办法 |
| 数据库连接错误 | 数据库设置不正确或异常 | 取消本次操作，提醒用户检查数据库 |
| 服务器连接错误 | 网络连接错误等 | 提醒用户检查网络连接 |
| 输入错误 | 输入不规范 | 通过对话框，提醒用户，然后再次操作 |
| 其他操作错误 | 用户的不正当操作使程序发生错误 | 中止操作提醒用户中止的原因和操作的规范 |
| 不可预知错误 | 未知异常 | 进行数据库备份，帮助开发者完善程序 |

## 6.2补救措施

1. 输入错误：这主要是用户输入不规范造成的，我们在尽量减少用户出错的条件的情况下，主要通过对话框，提醒用户，然后再次操作。
2. 其他操作错误：对于用户的不正当操作，有可能使程序发生错误。我们主要是中止操作，并提醒用户中止的原因和操作的规范。
3. 数据库连接错误：这类错误主要是数据库设置不正确的，服务端会给客户端发送一个提示，服务器维护人员需立即处理。
4. 其他不可预知的错误：程序也会有一些我们无法预知或没考虑完全的错误，我们对此不可能作出安全的异常处理，这时我们主要要保证数据的安全，所以要经常的进行数据库备份，并能及时的和我们联系，以逐步的完善我们的程序。。

## 6.3系统维护设计

软件的维护主要包括数据库的维护和软件功能的维护。

对于数据库的维护，本软件暂不提供备份恢复数据库的方法，需要管理员手动操作实现。

对于软件功能方面的维护，由于我们采用的是模块化的设计方法，每个模块（窗口）之间相互独立性较高，这样对软件的维护带来了很大的方便，对于单独功能的修改只需修改一个窗口就行了。而对于功能的添加，只要再添加菜单项的内容即可，维护人员应定期的对软件进行维护修改。