

EZ-Shopping APP

项目设计说明书

项目实训 第五组

目录

1.引言	1
1.1 编写目的	1
1.2 背景	2
1.3 定义	3
1.4 参考资料	4
1.5 项目概述	5
2.总体设计	6
2.1 需求规定	6
2.1.1 系统功能.....	6
2.1.2 系统性能.....	6
2.1.3 输入输出要求.....	6
2.1.4 数据管理能力要求.....	6
2.2 运行环境	7
2.2.1 设备	7

2.2.2 支持软件	8
2.3 基本设计概念和处理流程	9
2.4 结构	10
2.4.1 按功能划分（横向划分）	10
2.4.2 按处理流程（客户端/服务器端）划分（纵向划分）	16
2.5 尚未解决的问题	18
3.详细说明	20
3.1 用户登录模块设计说明	20
3.1.1 用户注册模块	20
3.1.2 用户登录模块	24
3.1.3 用户注销模块	29
3.1.4 用户添加银行卡信息模块	32
3.1.5 用户修改银行卡信息模块	36
3.1.6 用户添加收货地址模块	40
3.1.7 用户修改收货地址模块	44

3.2 按钮子系统设计说明.....	48
3.2.1 绑定按钮模块.....	48
3.2.2 查看按钮模块.....	53
3.2.3 清除按钮模块.....	56
3.3 订单子系统设计说明.....	60
3.3.1 用户下订单模块.....	60
3.3.2 用户查看订单模块.....	64
3.3.3 用户确认收货模块.....	68
3.3.4 卖家发货模块.....	72
3.4 商品信息管理模块设计说明.....	75
3.4.1 添加商品信息模块.....	75
3.4.2 删除商品信息模块.....	80
3.4.3 编辑商品信息模块.....	84
4.接口设计.....	89
4.1 界面设计.....	89

4.1.1 用户登录/用户主页模块.....	89
4.1.2 用户注册模块.....	91
4.1.3 按钮模块.....	92
4.1.4 订单管理模块.....	93
4.2 外部接口	93
4.2.1 硬件接口	93
4.2.2 软件接口.....	94
4.2.3 通信接口.....	95
5.系统数据结构设计	96
5.1 概念结构设计	96
5.2 逻辑结构设计	96
5.3 物理结构设计	97
6.运行设计	100
6.1 运行模块组合	100
6.2 运行控制	100

6.3 运行时间	100
7.系统出错处理设计	101
7.1 出错信息	101
7.2 补救措施	103
7.2.1 后备技术	103
7.2.2 降效技术	103
7.2.3 恢复及再启动技术	103
7.3 系统维护设计	103

1.引言

1.1 编写目的

从本阶段开始，项目进入正式开发阶段。本设计说明书的编写目的，是以本项目的需求分析说明书为依据，从总体设计的角度，明确软件工程教学辅助系统的总体架构，流程，数据结构，数据库设计。并且，本说明书将为今后关于软件工程教学辅助系统详细设计说明书提供依据。

目的在于：

- 为编码人员提供依据
- 为修改，维护提供条件
- 明确各模块外部接口，内部接口，用户接口
- 项目负责人将按计划说明书的要求布置和控制开发工作全过程

本说明书的预期读者包括：

- 软件客户
- 项目经理
- 项目开发人员

- 软件质量分析员
- 软件维护人员

1.2 背景

- 软件系统名称

EZ-Shopping APP for Android

- 任务提出者

浙江大学计算机学院

- 开发者

浙江大学 2017-2018 学年春夏学期项目实训课程第五小组

- 用户

所有使用产品的用户

- 实现该软件的计算机网络

由若干台服务器和手机，以及 EZ-Shopping 按钮组成的网络。

- 该软件系统同其他系统或其他机构的基本的相互来往关系

项目组需要与市场商品管理系统（或卖方）对接以获取初始数据

- 相关背景介绍

网络购物模式日臻完善，人们对网络购物的操作复杂性也有了更高的要求，对于一些不需要挑选的必需品进行快捷购买也变为部分人群的需求之一。

“EZ Shopping”就是基于此需求进行研发，是一个市场机会。扫码绑定 EZS 的新颖功能弥补了 Dash Button 商品少的缺陷，通过移动端 APP 进行配置也降低了对使用人群的素质要求，扩大了适用面。EZS 可以满足不同家庭对不同生活必需品快捷购买的需求。

1.3 定义

MySQL：一个小型关系型数据库管理系统。

Apache：世界使用排名第一的 Web 服务器软件，由于其跨平台和安全性被广泛使用，是最流行的 Web 服务器端软件之一。

PHP：PHP 是一种 HTML 内嵌式的语言，是一种在服务器端执行的嵌入 HTML 文本的脚本语言，语言的风格有类似于 C 语言，被广泛的运用。

JavaScript：Javascript 是一种面向对象的动态类型的区分大小写的客户端脚本语言。

AJAX :即“Asynchronous JavaScript and XML”(异步 JavaScript 和 XML) ,是指一种创建交互式网页应用的网页开发技术。

SQL 注入 通过把 SQL 命令插入到 Web 表单递交或输入域名或页面请求的查询字符串,最终达到欺骗服务器执行恶意的 SQL 命令。

数据库连接池 :数据库连接池负责分配、管理和释放数据库连接,它允许应用程序重复使用一个现有的数据库连接,这项技术能明显提高对数据库操作的性能。

UTF-8 : UTF-8 是 UNICODE 的一种变长字符编码又称万国码。

安全证书 :安全证书如同在进行网上交易时的身份证,或者说是私人钥匙,安全证书是唯一的,与任何其他人的证书都不相同。

MD5 加密算法 : Message Digest Algorithm MD5 (中文名为消息摘要算法第五版)为计算机安全领域广泛使用的一种散列函数 ,用以提供消息的完整性保护。

Android APP : 在安卓机器上运行的应用程序

NFC :又称近距离无线通信。是一种允许电子设备之间进行非接触式点对点短距离高频无线数据传输的技术

1.4 参考资料

- 《项目实训案例 170704》 甲方给出的项目描述

- 《软件工程-实践者的研究方法》 作者 Roger S.Pressman (美)

译者郑人杰马素霞白晓颖 出版社机械工业出版社 出版时间 2007 年 1 月

1.5 项目概述

一个完整的软件工程教学辅助系统，应实现以下功能模块：

- 用户信息模块：

用户将自己的信息储存在用户管理子系统中，在登录、绑定商品到按钮等操作时会使用该模块。

- 按钮管理模块：

用于管理用户自己的个人信息和绑定的按钮与商品信息。

- 订单模块：

用户和商家都可以使用订单模块，其中用户通过按钮下订单，并通过 APP 查看订单，商家同样可以查看订单并修改订单状态信息。

- 商品模块：

商家可以商品模块管理商品子，用以修改商品的信息。用户扫描条形码时同样要用到商品模块。

2. 总体设计

2.1 需求规定

2.1.1 系统功能

用户界面，首先是登陆界面，登陆后进入主页，主页即为按钮管理页，用户可以查看按钮或者绑定按钮，左侧是订单页，用户可以查看订单，右侧是个人信息管理页，用户可以修改银行卡和收货地址等个人信息。

2.1.2 系统性能

本系统的性能将由服务器端数据库，网络数据传输延时，以及并发访问该系统的用户数量决定。客户端方面将实现 AJAX 技术，减轻服务器压力，增加带宽利用率，并为用户提供良好的界面体验。

2.1.3 输入输出要求

系统通过 APP 展现给用户一个友好的界面，用户可以通过提交表单或者点击超链接向服务器提供数据与命令。服务器后台处理后将结果显示到用户的 APP 上。

2.1.4 数据管理能力要求

安全：服务器将予以数据库最高等级的保护，以防止黑客从后台下载数据库，防止通过 SQL 注入的方式从数据库中获取信息或者破坏数据库。

性能：对于频繁访问数据库的操作，后台需要建立持久的数据库连接，以避免重复连接数据库耗费资源。

2.2 运行环境

条件有限，本项目不能提供强大可靠的服务器，只能用性能较好的 PC 机代替。

2.2.1 设备

手机：

系统：安卓 4.0 以上

内存： $\geq 1.0\text{GB}$

服务器设备：

硬盘： $\geq 100\text{GB}$

硬盘转速： $\geq 5400\text{rpm}$

并发处理能力： > 100 次每秒

通讯设备：

无线网：具有良好数据传输能力

无限路由器：具有良好的网络稳定性

客户端的设备要求如下：

能够上网的安卓手机，推荐使用最新的安卓版本。

2.2.2 支持软件

服务器端：

操作系统：Windows 7

Php:php5.0

数据库：MySQL

Web 服务器：Apache

MySQL 管理软件：PHPMyAdmin 或其他软件

开发工具：subline3 , dreamweaver3.0

客户端：

手机：Android 4.0

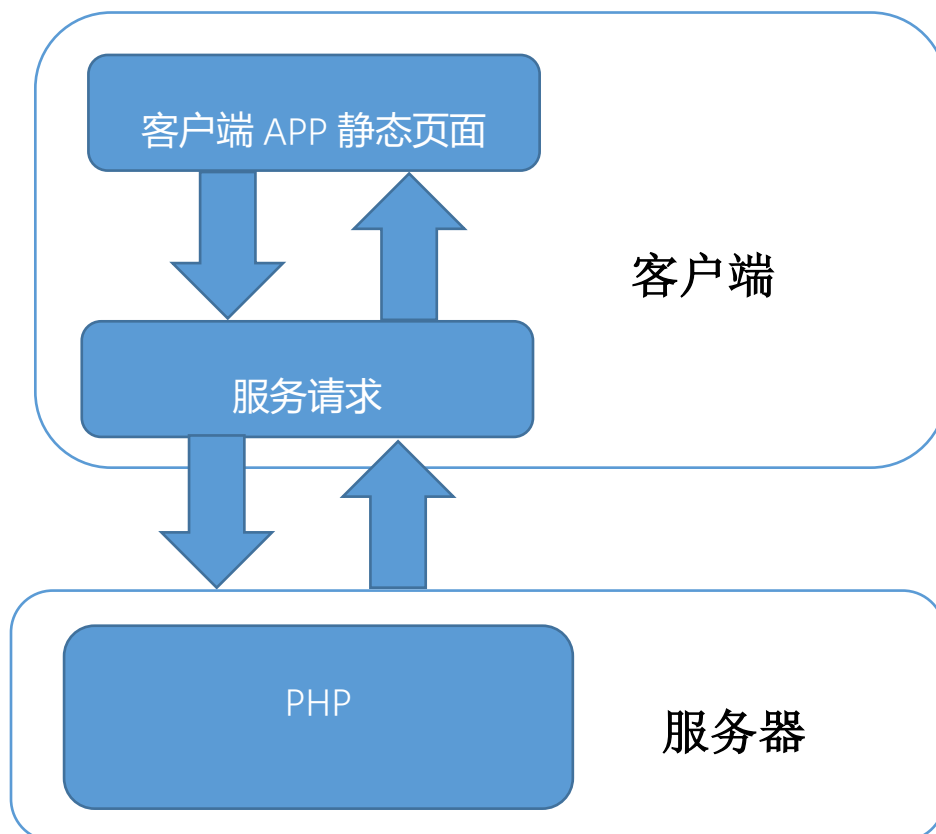
2.3 基本设计概念和处理流程

管理员模块分为客户端和服务端。

服务器：以apache为web服务器，服务器端使用PHP脚本语言编写。数据库使用mysql。

客户端：以安卓4.0版本为最低版本要求，有一个APP客户端。

处理流程图如下：



客户端静态页面：通过Android Studio，设计一个简洁，美观的界面。

服务请求 :客户端界面中 ,各种文本框与按键确认的操作均能触发脚本函数 ,脚本通过创建POST对象 ,与服务器进行异步交互。并且 ,对象接受服务器反馈信息后 ,能对APP客户端页面实现更新信息。

PHP访问/修改数据库 :此模块从客户端得到数据与命令 ,并且对其进行安全检测 ,而后按要求访问/修改数据库 ,并且返回操作结果或是查询结果。

2.4 结构

2.4.1 按功能划分 (横向划分)

教学辅助系统 HIPO 图 :

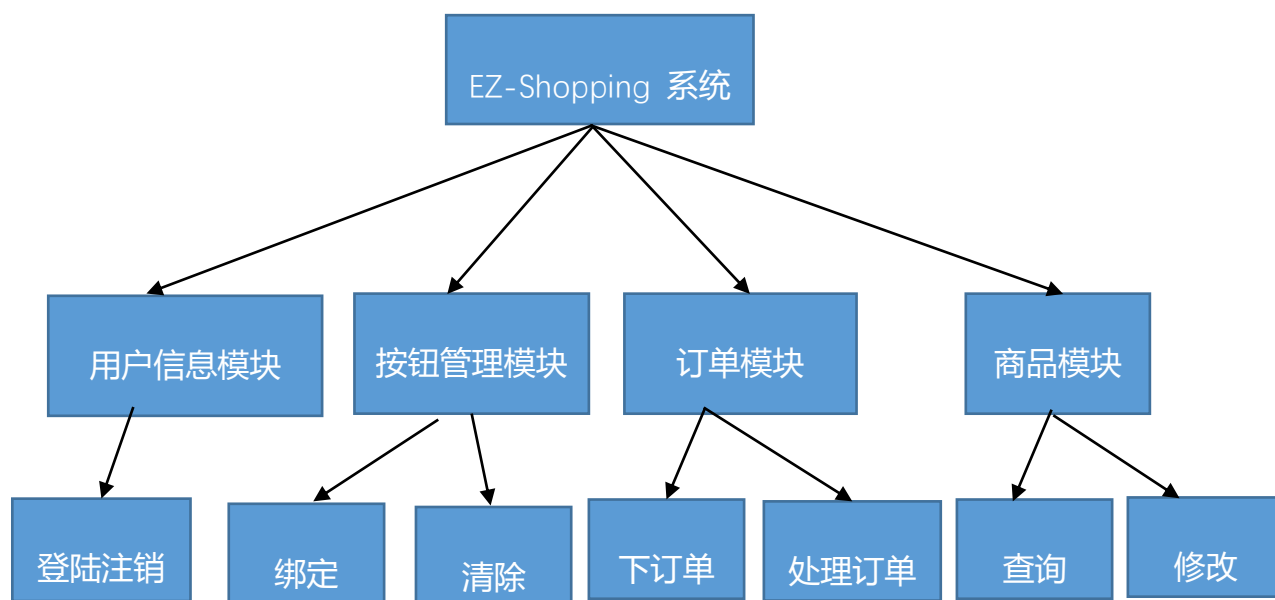


图2.4.1.1 教学辅助系统层次图

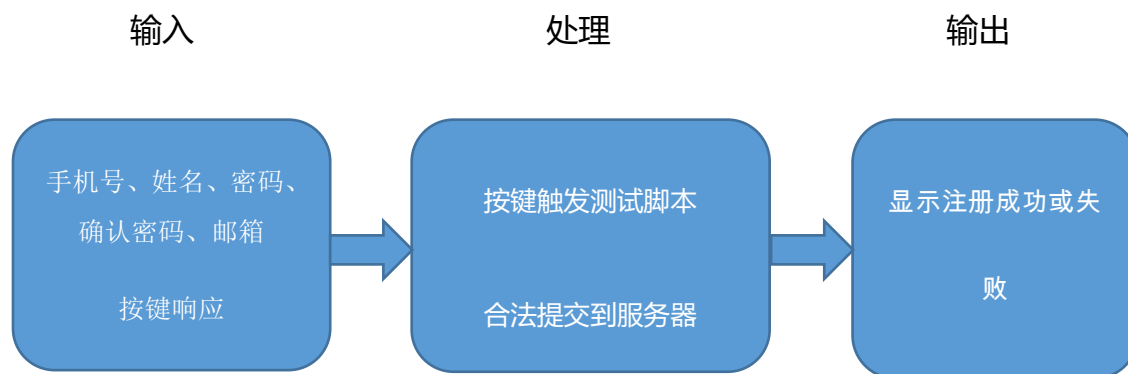


图2.4.1.2用户注册模块IPO图

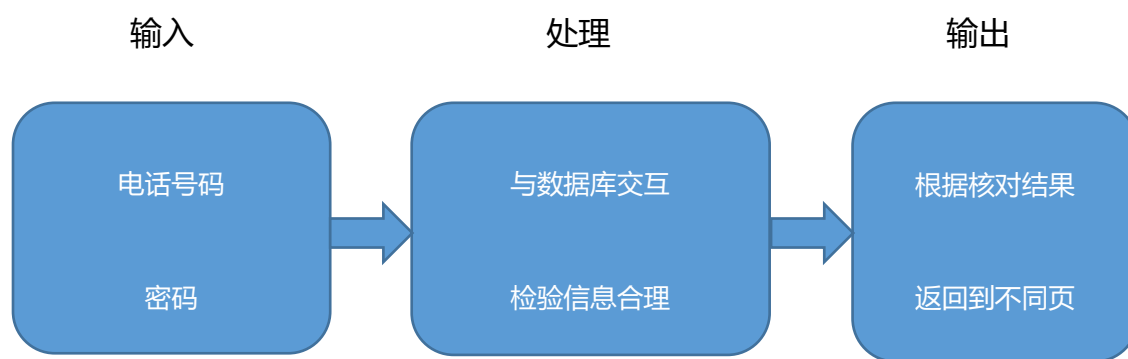


图2.4.1.3 用户登录模块IPO图

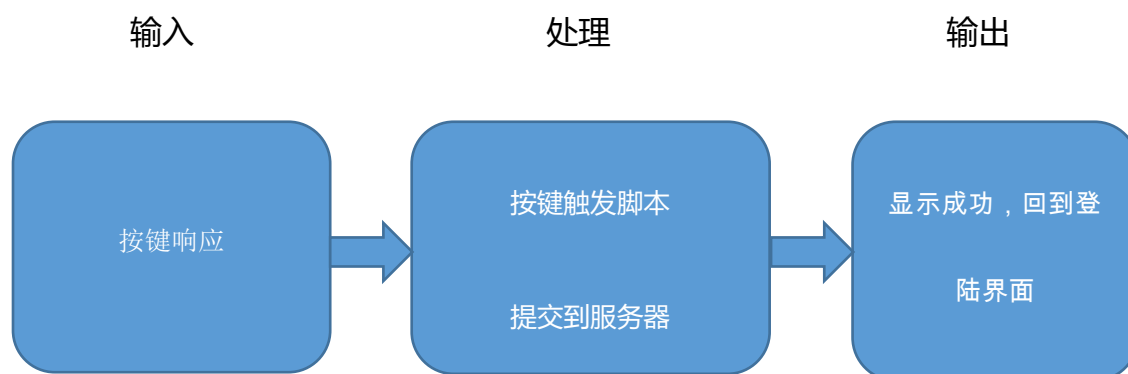


图2.4.1.4用户注销模块IPO图

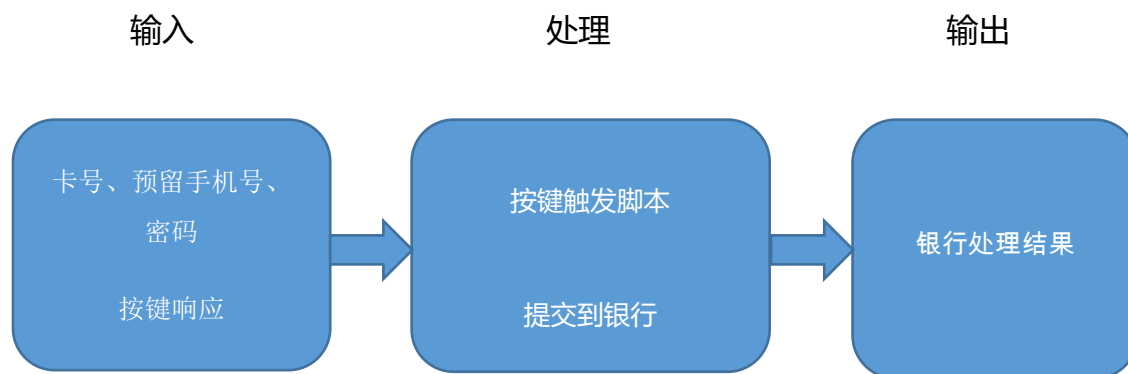


图2.4.1.5用户添加银行卡信息模块IPO图

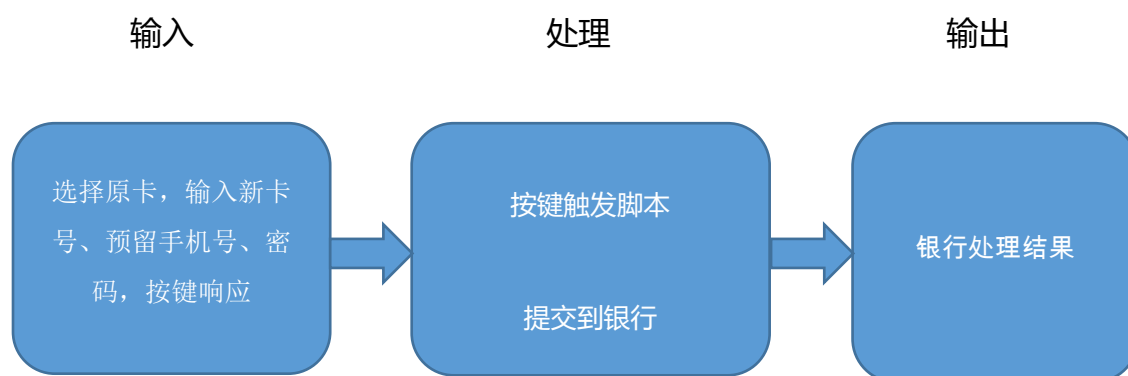


图2.4.1.6 用户修改银行卡信息模块IPO图

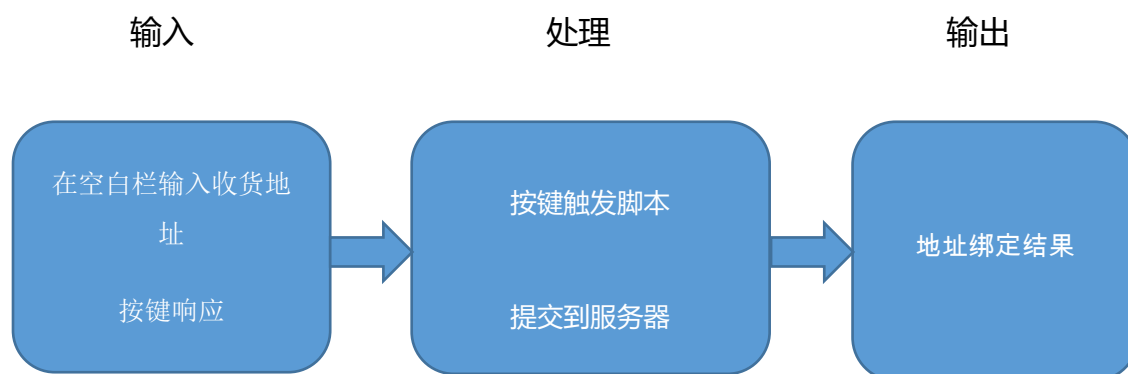


图2.4.1.7用户添加收货地址模块IPO图

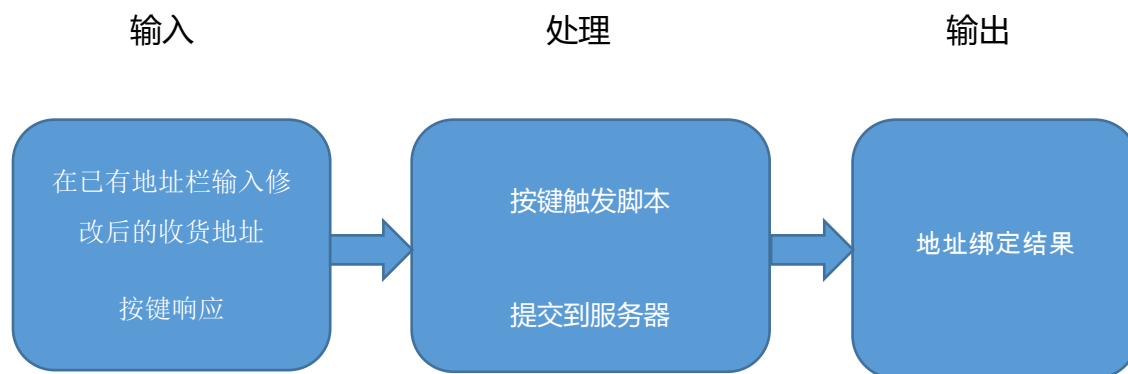


图2.4.1.8用户修改收货地址模块IPO图

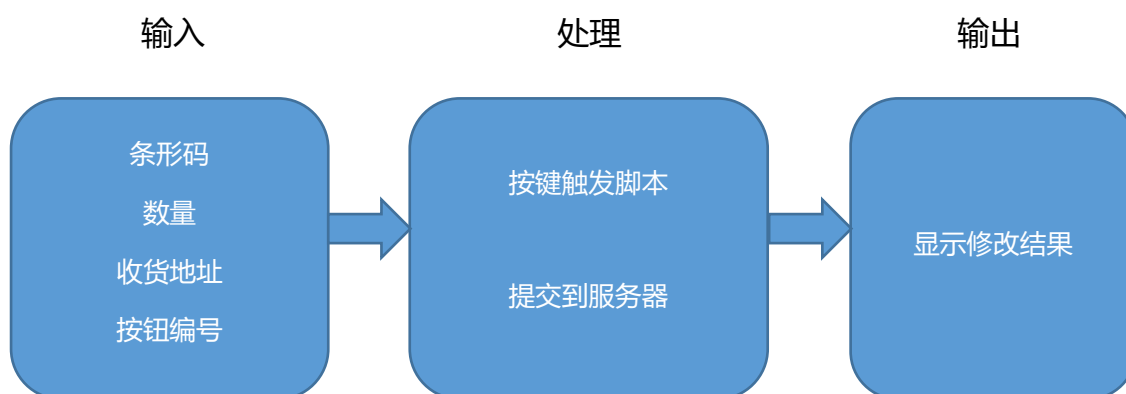
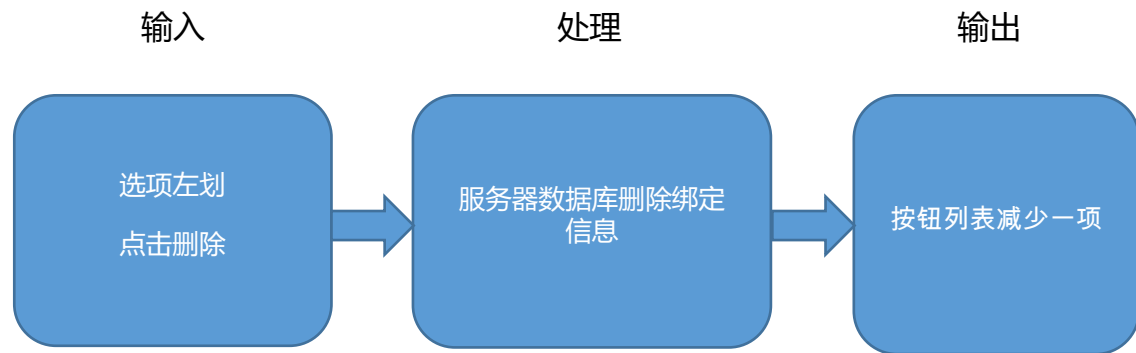
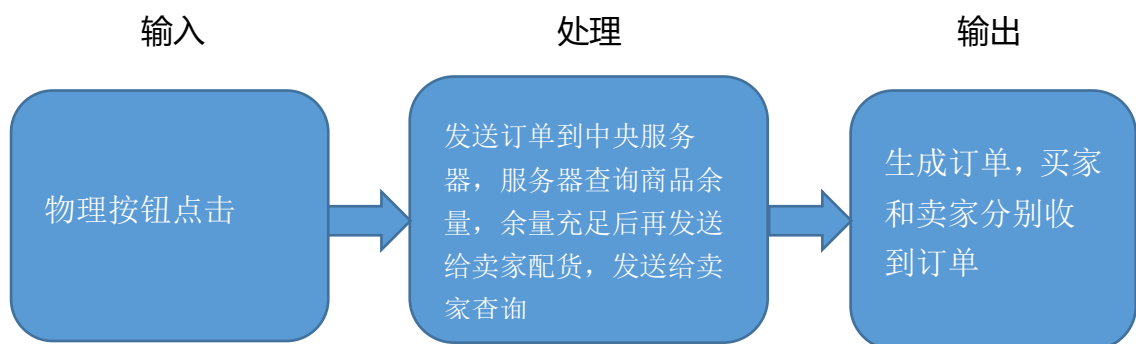
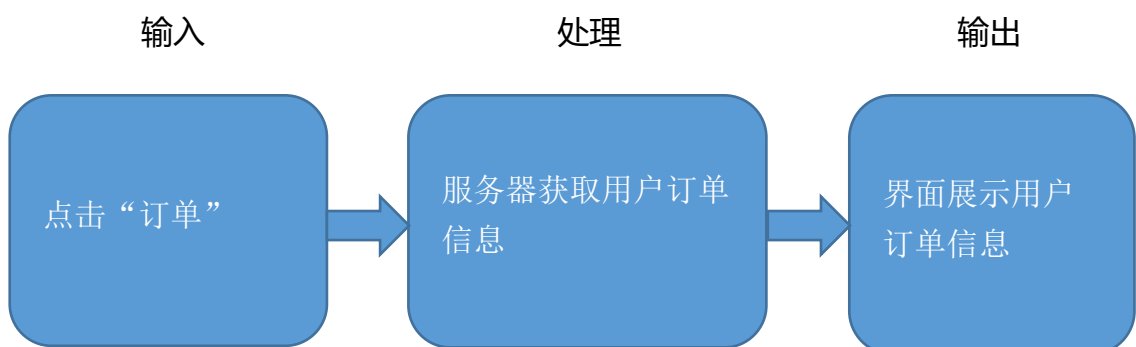


图2.4.1.9 绑定按钮模块IPO图



图2.4.1.10 查看按钮模块IPO图

**图2.4.1.11 清除按钮模块IPO图****图2.4.1.12 用户下订单模块IPO图****图2.4.1.13 用户查看订单模块IPO图**

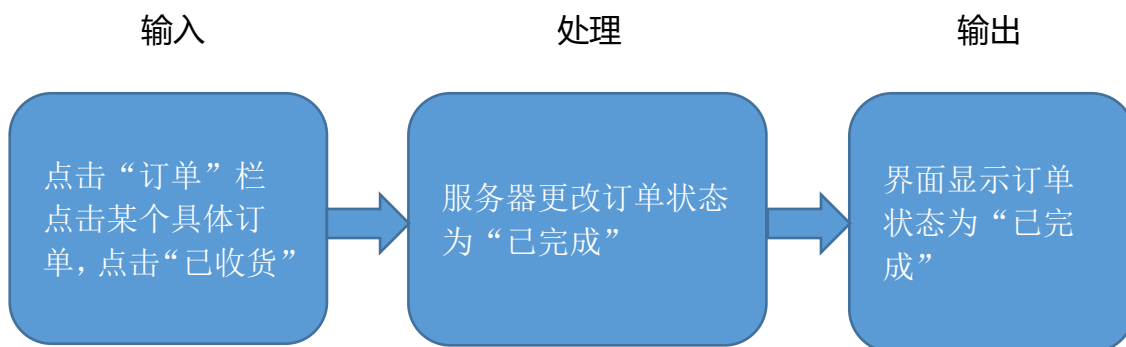


图2.4.1.14 用户确认收货模块IPO图

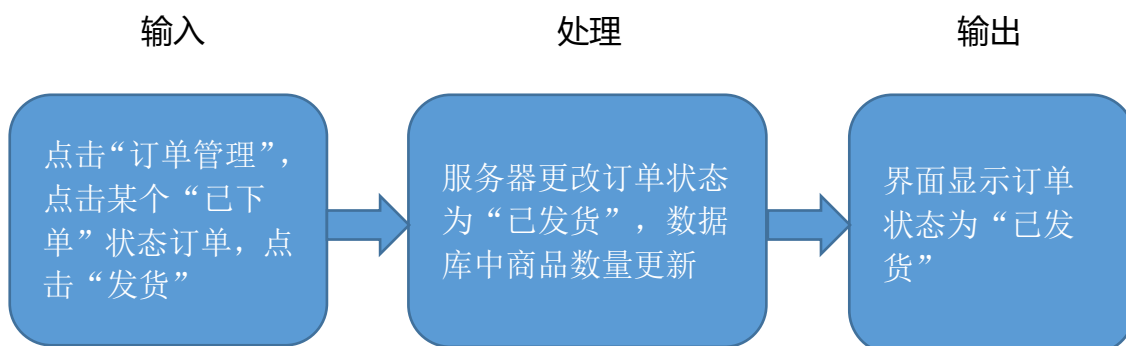


图2.4.1.15 卖家发货模块IPO图

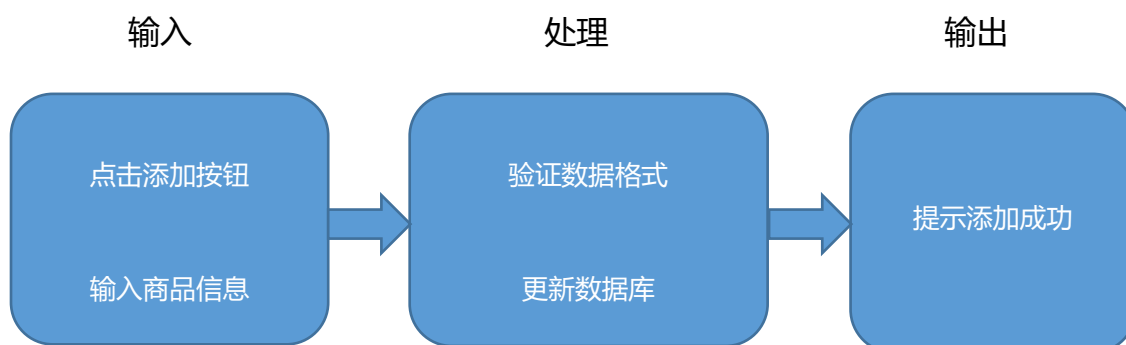


图2.4.1.16 添加商品信息模块IPO图



图2.4.1.17 删除商品信息模块IPO图

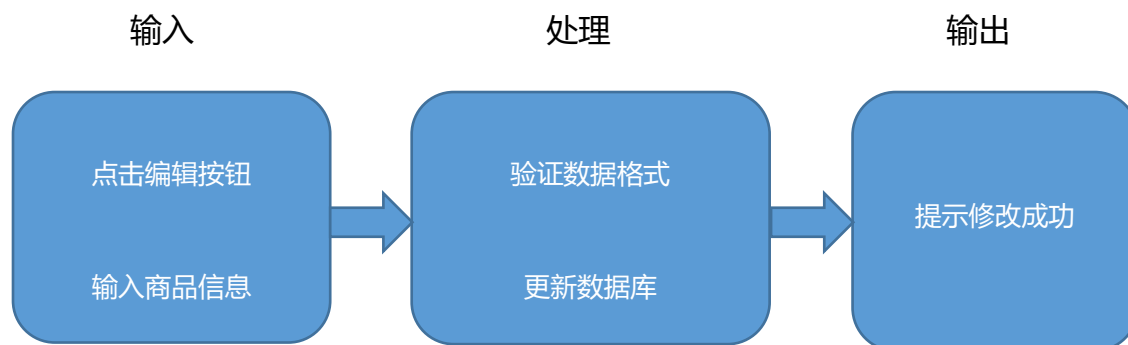


图2.4.1.18修改商品信息模块IPO图

2.4.2 按处理流程（客户端/服务器端）划分（纵向划分）

2.4.2.1 客户端

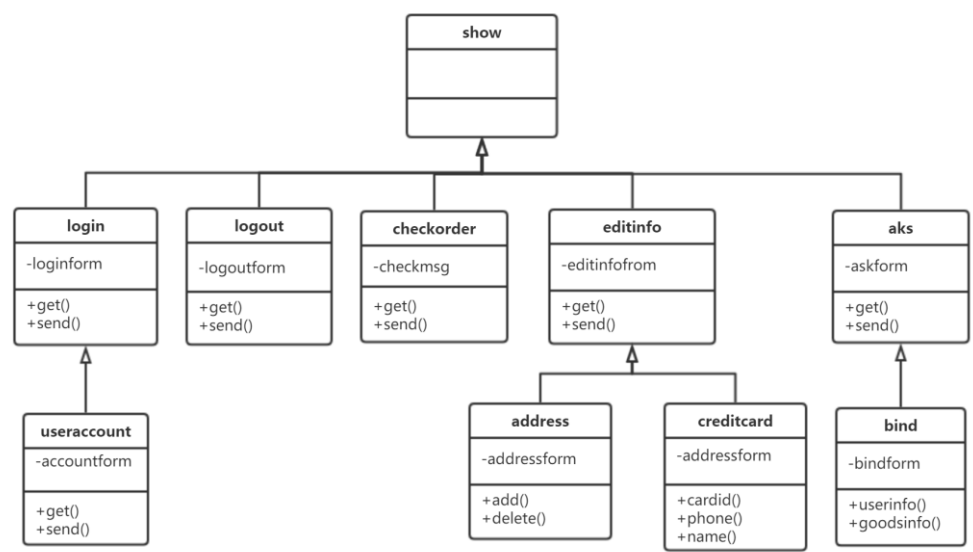


图2.4.2.1.1 客户端功能抽象类图

2.4.2.2 服务器端

服务器主要由 PHP 实现

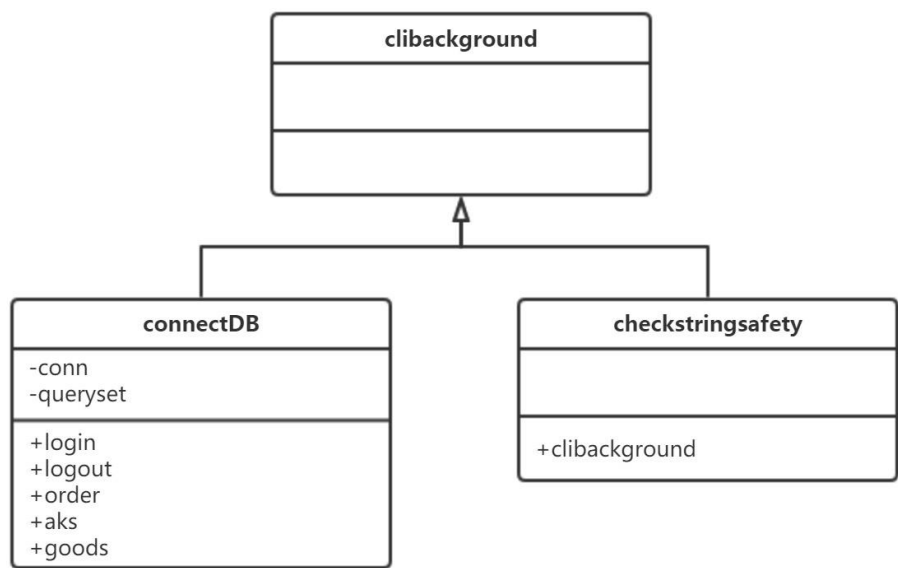


图2.4.2.2.1 服务器端功能抽象类图

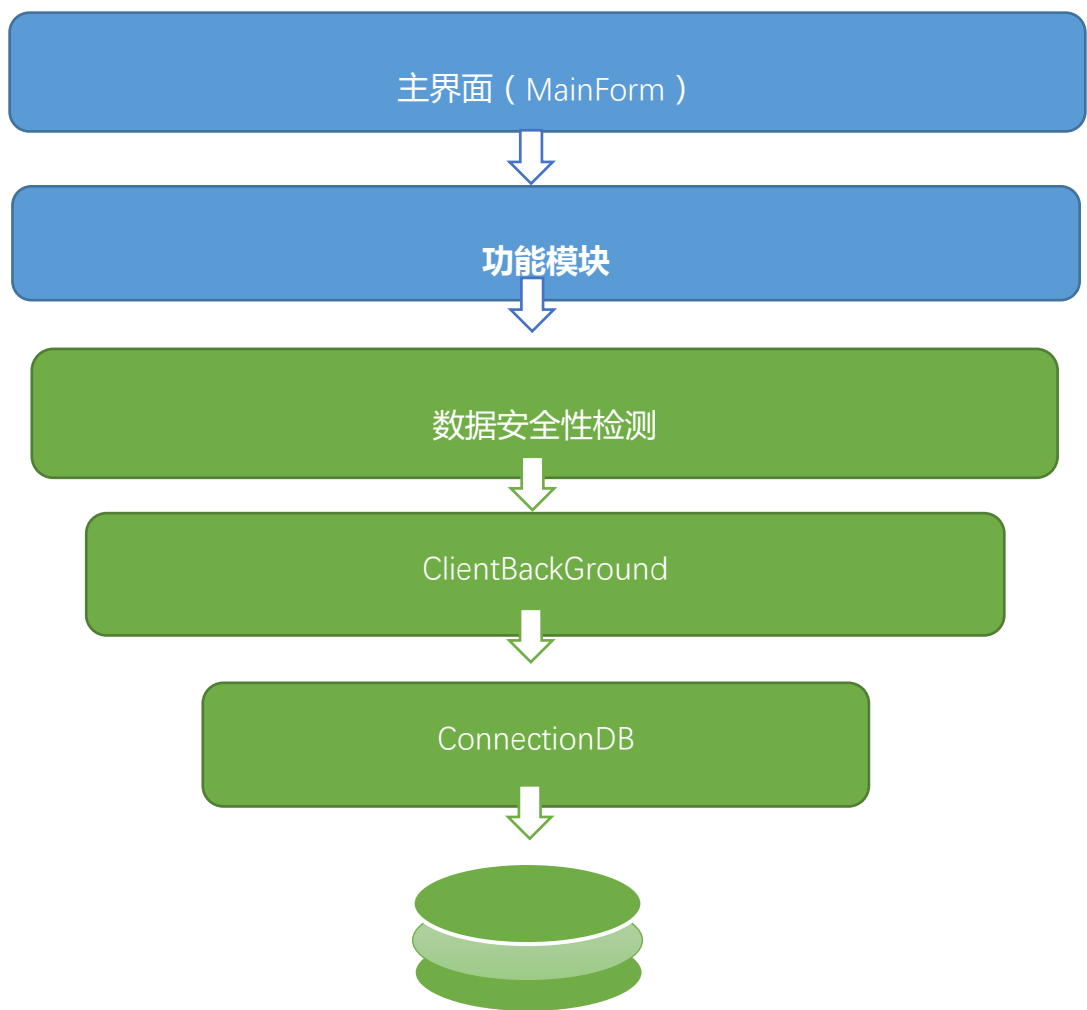


图 2.4.2.2.2 各模块隶属调用关系

2.5 尚未解决的问题

编号	涉及用例/需求	描述	优先级
TBD-01	按钮报废	达到一定使用次数或由于意外情况，按钮不再提供服务，考虑将按钮报废，服务器	较低

		内清空所有关于该按钮的信息	
TBD-02	卖家接收订单	本 APP 限定只要有货，服务器自动为卖家接单，考虑是否提供卖家拒绝订单的功能	较低
TBD-03	卖家更新商品信息	卖家更改商品价格时，通知绑定相关商品的买家	较高
TBD-04	用户确认收货	用户对收到的商品不满意，可退货	一般
TBD-05	用户下订单	用户不小心按到按钮（误操作），在 APP 内取消订单	较高

3.详细说明

3.1 用户登录模块设计说明

3.1.1 用户注册模块

3.1.1.1 模块描述

此模块是用户初次使用 app 时注册用模块 ,用户需提交手机号 姓名 密码 , 邮箱等必须信息。

3.1.1.2 功能

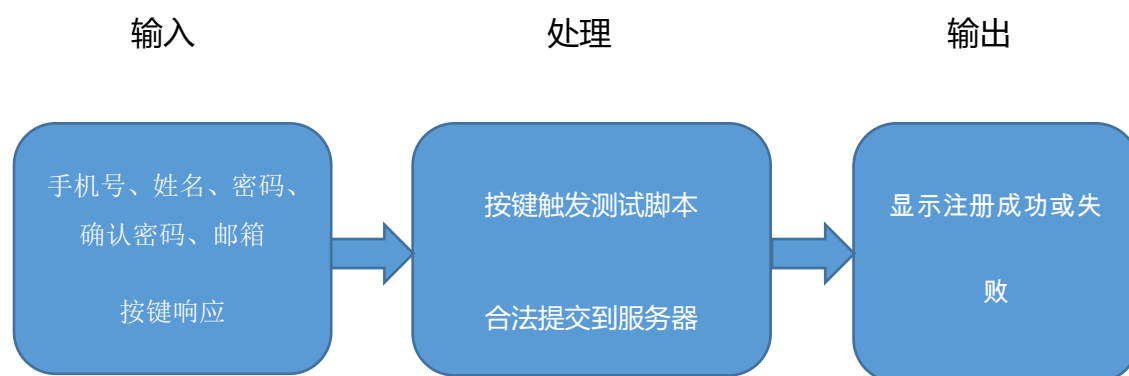


图3.1.1.2.1用户注册模块IPO图

3.1.1.3 性能

用户通过手机号、姓名、密码、邮箱注册新的用户。

3.1.1.4 输入项

表3.1.1.4.1 用户注册模块输入项列表

名称	标识	类型和格式	输入方式
手机号	Phone	String	文本框输入
姓名	Name	String	文本框输入
密码	Password	String	文本框输入
验证密码	ChkPwd	String	文本框输入
提交	Submit	Button	用户点击

3.1.1.5 输出项

表3.1.1.5.1 用户注册模块输出项列表

名称	标识	类型和格式	输出方式
结果	Res	Enum{注册成功 , 注册失败}	msgbox

3.1.1.6 设计方法

客户端把注册表单提交给服务器，服务器执行如下代码：

```
<?php
    public function register($info){
        if(check($info))
            return false
        else
            InsertUserIntoDatabase($info)
    }
?>
```

3.1.1.7 流程逻辑

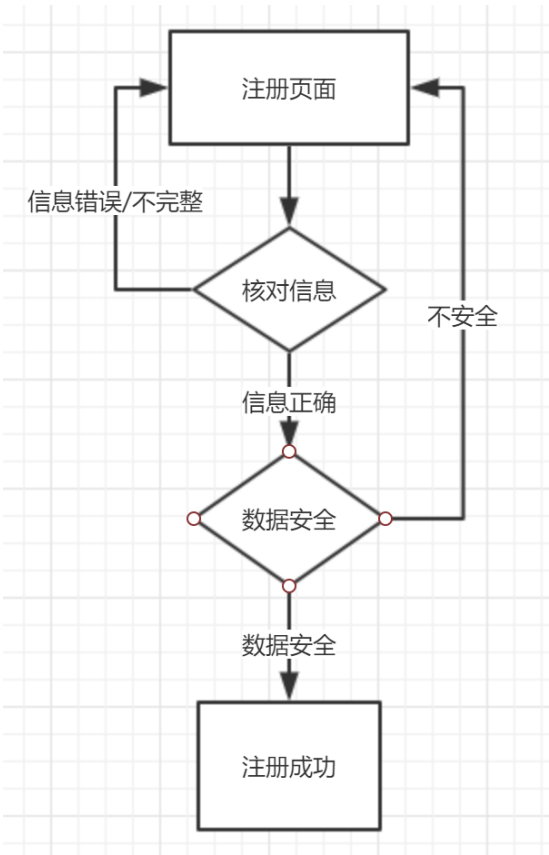


表3.1.1.7.1 用户注册模块流程图

3.1.1.8 接口



图3.1.1.8.1 用户注册模块用户接口

3.1.1.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.1.9.1 用户注册模块测试列表

测试用例	测试数据	预期结果
UC-BY-01-001	输入不合法的手机号(123)	提示手机号不合法

UC-BY-01-002	输入已有手机号(13033600725)	提示手机号不合法
UC-BY-01-003	输入不合法的邮箱(1111)	提示邮箱不合法
UC-BY-01-004	输入空的姓名	提示用户名不合法
UC-BY-01-005	输入超过 50 位的密码(abcd...)	提示密码不合法
UC-BY-01-006	输入的密码与确认密码不一致(abc,123)	提示密码不一致
UC-BY-01-007	输入完全合法的信息 (13033600725,maoyiming,123,123,123@qq.com)	提示注册成功 ,返回 登陆界面

3.1.2 用户登录模块

3.1.2.1 模块描述

用户在访问 APP 主页时将首先访问登录页面，登录所需的账号密码用户应在注册时自行保管。

3.1.2.2 功能

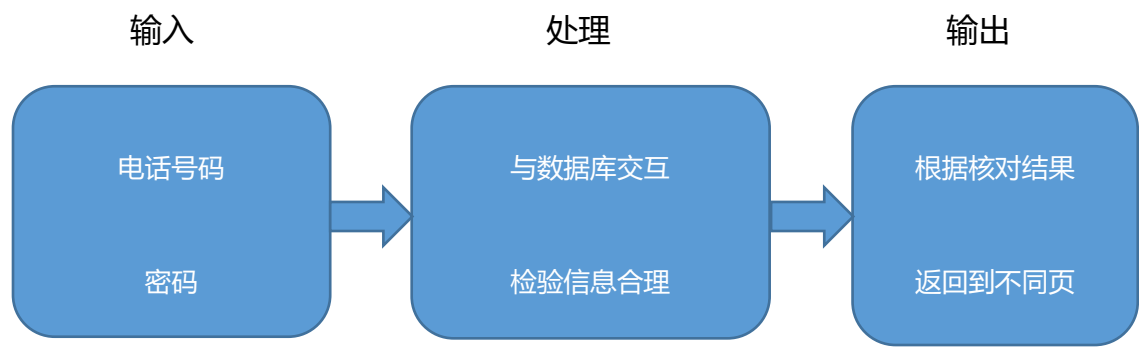


图3.1.2.2 用户登录模块IPO图

3.1.2.3 性能

输入账号密码后点击登录，便能检查后登录。

3.1.2.4 输入项

表3.1.2.4.1 登录模块输入项列表

名称	标识	类型和格式	输入方式
用户名	User_Id	String	键盘输入
密码	Password	String	键盘输入
登录	Login	Button	点击按钮

3.1.2.5 输出项

表3.1.2.5.1 用户登录模块输出项列表

名称	标识	类型和格式	输出方式
登录结果	Loginresult	Enum{登陆成功， 登陆失败}	脚本输出

3.1.2.6 设计方法

客户端把已经简单验证的登录表单提交给服务器，服务器执行如下代码：

```
<?php
    public function login($info){
        if(checklogin($info))
            return ture
        else
            return false
    }
?>
```

3.1.2.7 流程逻辑

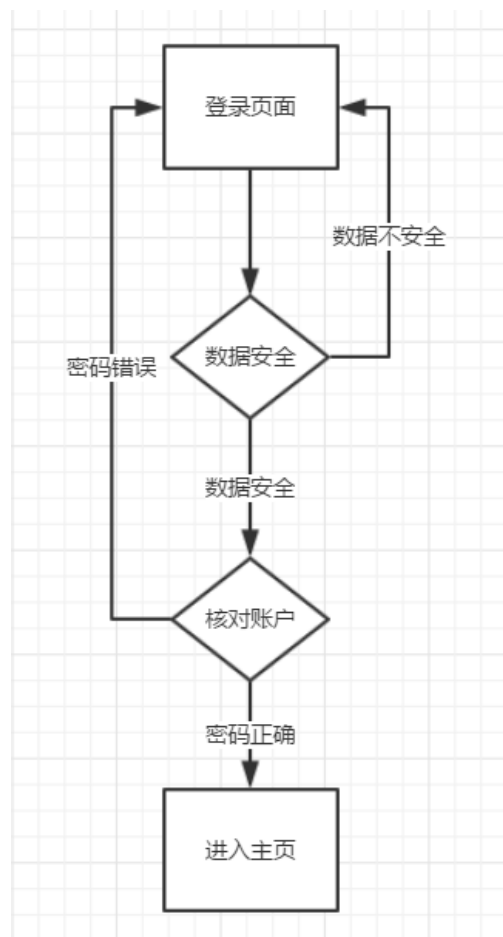


图3.1.2.7 用户登录模块流程图

3.1.2.8 接口



图表3.1.2.8 用户登录模块用户接口

3.1.2.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.2.9.1 用户登录模块测试列表

测试用例	测试数据	预期结果
UC-BY-02-001	用户名密码为空(NULL,NULL)	提示用户名密码不能为空

UC-BY-02-002	仅输入用户名，密码为空 (admin,NULL)	提示请输入密码，登录失败
UC-BY-02-003	用户名为空，仅输入密码 (NULL,admin)	提示用户名不能为空，登录失败
UC-BY-02-004	输入正确用户名错误密码 (admin,a)	提示用户名或者密码错误，登录失败
UC-BY-02-005	输入错误用户名正确密码 (a,admin)	提示用户名或者密码错误，登录失败
UC-BY-02-006	输入正确用户名和密码 (admin,admin)	登录成功，进入页面
UC-BY-02-007	输入较长的用户名和密码，超过 50 位 (abcde...,abcde...)	提示字符串长度超出限制
UC-BY-02-008	输入包含特殊字符的用户名 (例如各种标点符号)	提示用户名不能含有特殊字符
UC-BY-02-009	输入大小写不一致的用户名和密码 (Admin,Admin)	提示用户名或者密码错误，登录失败

3.1.3 用户注销模块

3.1.3.1 模块描述

此模块是用户注销退出 APP 时使用的模块。

3.1.3.2 功能

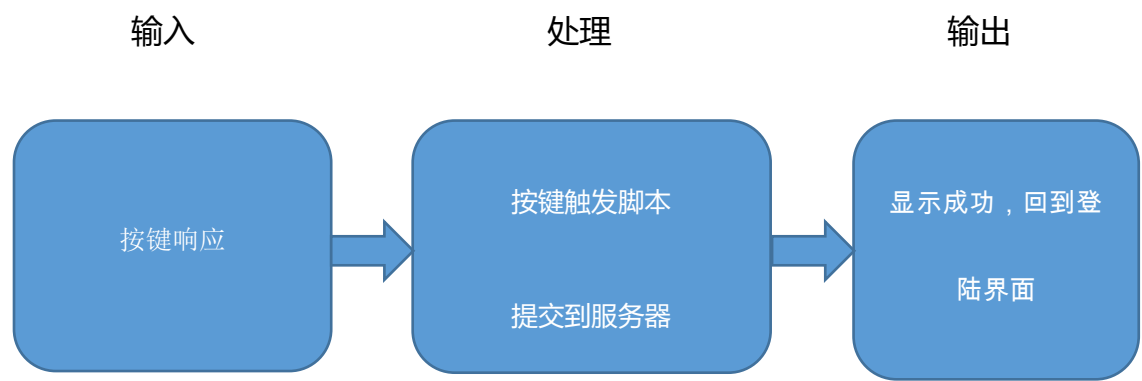


图3.1.3.2.1用户注销模块IPO图

3.1.3.3 性能

用户通过点击按钮注销当前登陆，返回登陆界面。

3.1.3.4 输入项

表3.1.3.4.1 用户注销模块输入项列表

名称	标识	类型和格式	输入方式
提交	Submit	Button	用户点击

3.1.3.5 输出项

表3.1.3.5.1 用户注销模块输出项列表

名称	标识	类型和格式	输出方式
结果	Res	Enum{成功，失败}	msgbox

3.1.3.6 设计方法

客户端把注册表单提交给服务器，服务器执行如下代码：

```
<?php
    public function logout($info){
        delete_session();
    }
?>
```

3.1.3.7 流程逻辑

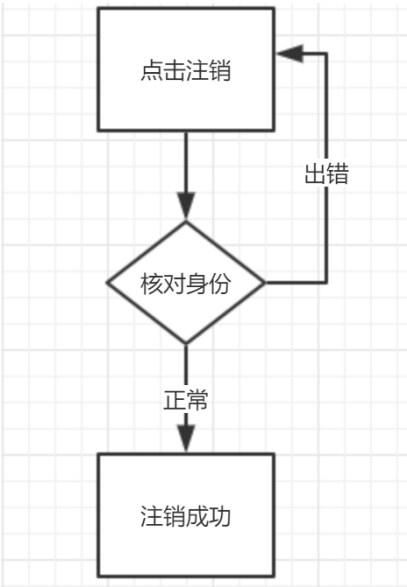


表3.1.3.7.1 用户注销模块流程图

3.1.3.8 接口



图3.1.3.8.1 用户注销模块用户接口

3.1.3.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.3.9.1 用户注销模块测试列表

测试用例	测试数据	预期结果
UC-BY-03-001	登陆一个用户，按操作注销	提示注销成功，返回登陆界面

3.1.4 用户添加银行卡信息模块

3.1.4.1 模块描述

此模块是用户添加个人账户银行卡信息使用的模块。

3.1.4.2 功能

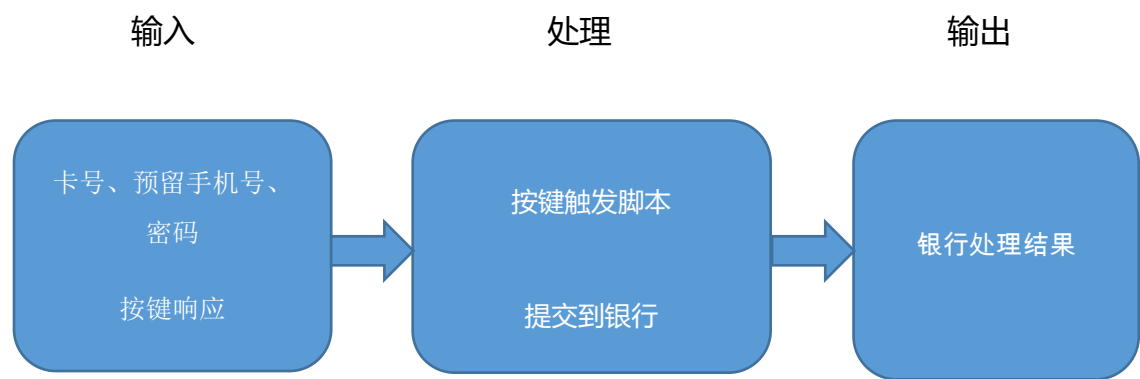


图3.1.4.2.1用户添加银行卡信息模块IPO图

3.1.4.3 性能

用户通过输入卡号、预留手机号、密码并点击按钮提交哦，返回银行处理结果。

3.1.4.4 输入项

表3.1.4.4.1 用户添加银行卡信息模块输入项列表

名称	标识	类型和格式	输入方式
卡号	Card_id	String	键盘输入

预留手机号	Phone	String	键盘输入
密码	Password	String	键盘输入
提交	Submit	Button	用户点击

3.1.4.5 输出项

表3.1.4.5.1 用户添加银行卡信息模块输出项列表

名称	标识	类型和格式	输出方式
结果	Res	Enum{成功, 失败}	msgbox

3.1.4.6 设计方法

客户端把完整的银行卡信息表单提交给服务器，服务器执行如下代码：

```
<?php
    public function BindCard($info){
        if(!check($info))
            return false
        else
            BindCardToUser($info)
    }
?>
```

3.1.4.7 流程逻辑

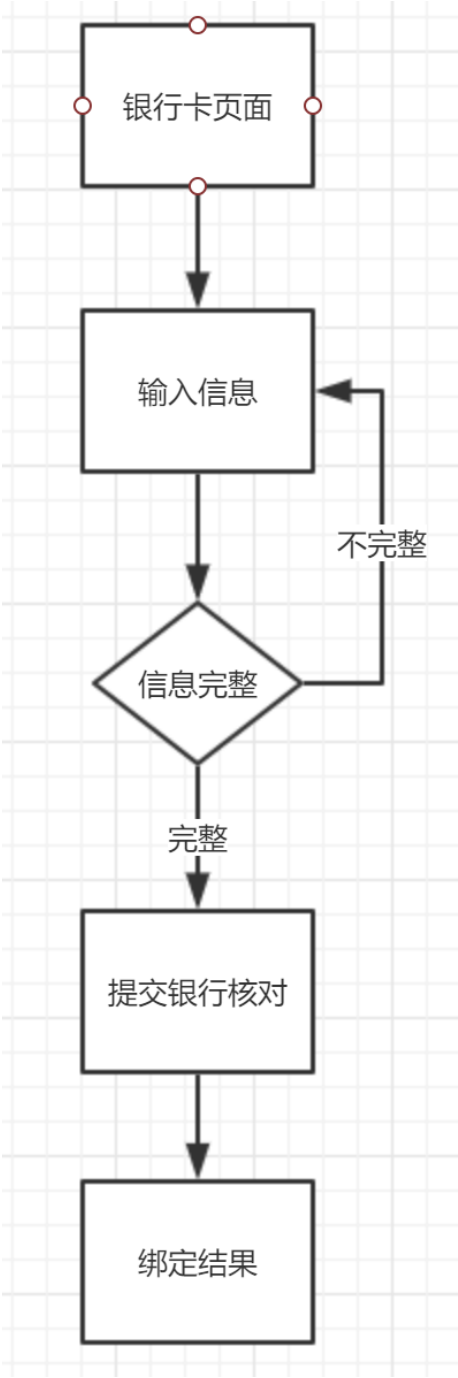


表3.1.4.7.1 用户添加银行卡信息模块流程图

3.1.4.8 接口



图3.1.4.8.1 用户添加银行卡信息模块用户接口

3.1.4.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.4.9.1 用户添加银行卡模块测试列表

测试用例	测试数据	预期结果
UC-BY-04-001	卡号为空(NULL)	提示卡号不能为空

UC-BY-04-002	手机号为空(NULL)	提示手机号不能为空
UC-BY-04-003	密码为空(NULL)	提示密码不能为空
UC-BY-04-004	信息完整，银行反馈失败	提示绑定失败
UC-BY-04-005	信息完整，银行反馈成功	提示绑定成功

3.1.5 用户修改银行卡信息模块

3.1.5.1 模块描述

此模块是用户修改个人账户银行卡信息使用的模块。

3.1.5.2 功能

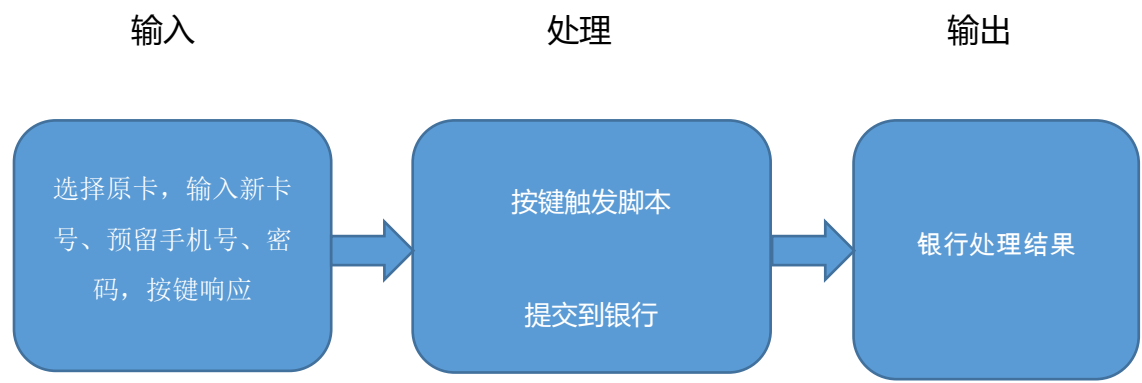


图3.1.5.2.1用户修改银行卡信息模块IPO图

3.1.5.3 性能

用户通过选择原卡，输入新卡号、预留手机号、密码并点击按钮提交，返回银行处理结果。

3.1.5.4 输入项

表3.1.5.4.1 用户修改银行卡信息模块输入项列表

名称	标识	类型和格式	输入方式
原卡号	OCard_id	String	用户点击
卡号	NCard_id	String	键盘输入
预留手机号	Phone	String	键盘输入
密码	Password	String	键盘输入
提交	Submit	Button	用户点击

3.1.5.5 输出项

表3.1.5.5.1 用户修改银行卡信息模块输出项列表

名称	标识	类型和格式	输出方式
结果	Res	Enum{成功，失败}	msgbox

3.1.5.6 设计方法

客户端把注册表单提交给服务器，服务器执行如下代码：

```
<?php
    public function ChangeCard($info){
        if(CheckCard($info))
            return false
        else
            UpdateUserCard($info)
    }
?>
```

3.1.5.7 流程逻辑

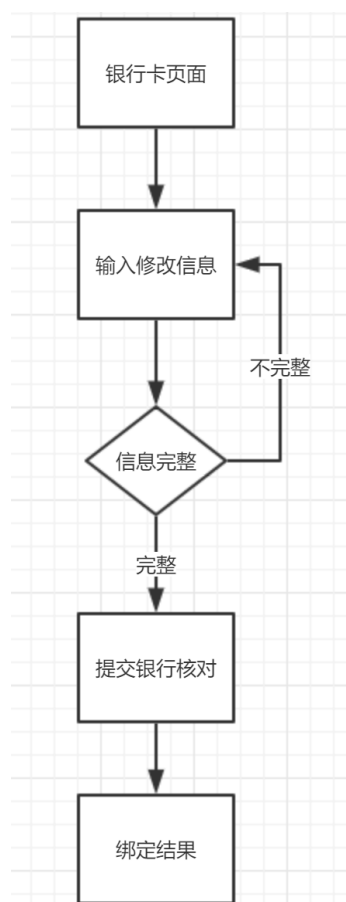


表3.1.5.7.1 用户修改银行卡信息模块流程图

3.1.5.8 接口



图3.1.5.8.1 用户修改银行卡信息模块用户接口

3.1.5.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.5.9.1 用户修改银行卡模块测试列表

测试用例	测试数据	预期结果
------	------	------

UC-BY-05-001	卡号为空(NULL)	提示卡号不能为空
UC-BY-05-002	手机号为空(NULL)	提示手机号不能为空
UC-BY-05-003	密码为空(NULL)	提示密码不能为空
UC-BY-05-004	信息完整，银行反馈失败	提示绑定失败
UC-BY-05-005	信息完整，银行反馈成功	提示绑定成功

3.1.6 用户添加收货地址模块

3.1.6.1 模块描述

此模块是用户添加收货地址信息使用的模块。

3.1.6.2 功能

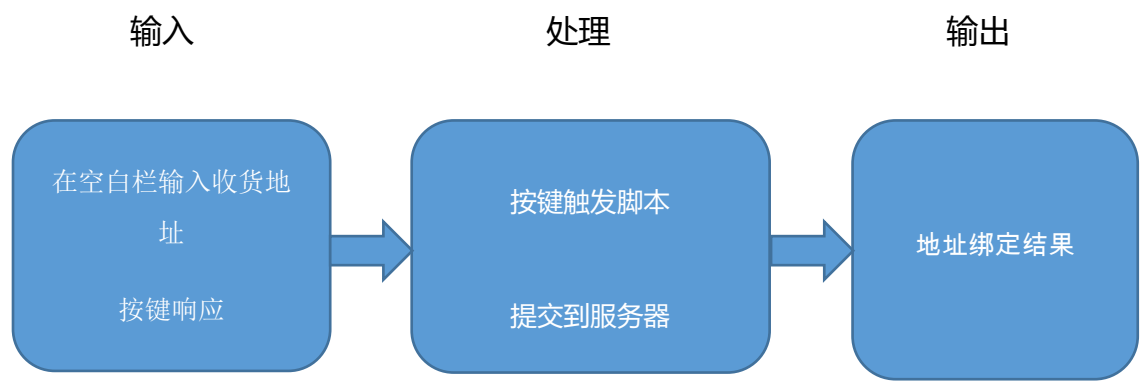


图3.1.6.2.1用户添加收货地址模块IPO图

3.1.6.3 性能

用户通过在空白栏输入新的地址并点击按钮提交，返回添加结果。

3.1.6.4 输入项

表3.1.6.4.1 用户添加收货地址模块输入项列表

名称	标识	类型和格式	输入方式
地址	Address	String[]	键盘输入
提交	Submit	Button	用户点击

3.1.6.5 输出项

表3.1.6.5.1 用户添加收货地址模块输出项列表

名称	标识	类型和格式	输出方式
结果	Res	Enum{成功，失败}	msgbox

3.1.6.6 设计方法

收货地址代表地点的正确性不予以验证，验证只验证输入符号，点击确认客户端会向服务器发送地址表单，服务器会执行如下代码：

```
<?php  
  
    public function Address($addressss){  
        InsertAddressToUser($address)  
    }  
?>
```

3.1.6.7 流程逻辑

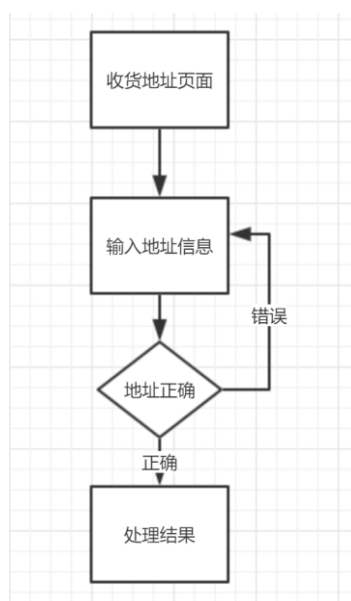


表3.1.6.7.1 用户添加收货地址模块流程图

3.1.6.8 接口

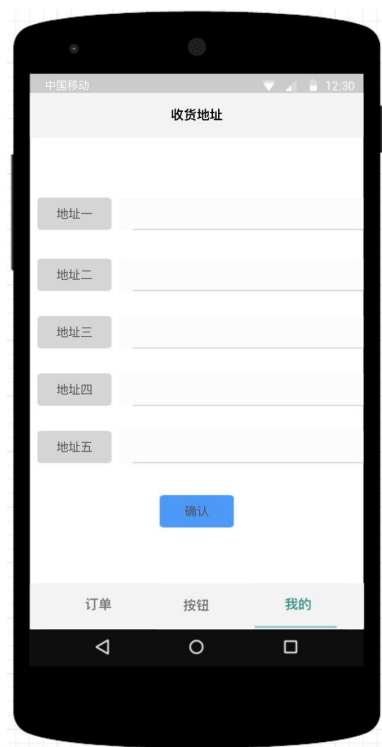


图3.1.6.8.1 用户添加收货地址模块用户接口

3.1.6.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.6.9.1 用户添加收货地址模块测试列表

测试用例	测试数据	预期结果
UC-BY-06-001	地址为空(NULL)	提示地址为空
UC-BY-06-002	地址合理(浙江大学)	提示绑定成功

3.1.7 用户修改收货地址模块

3.1.7.1 模块描述

此模块是用户修改收货地址信息使用的模块。

3.1.7.2 功能

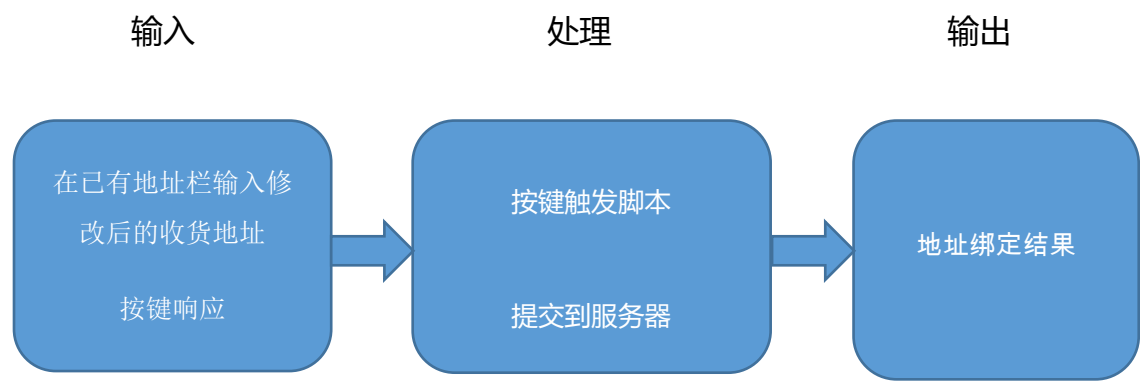


图3.1.7.2.1用户修改收货地址模块IPO图

3.1.7.3 性能

用户通过在已有地址的栏输入新的地址并点击按钮提交，返回修改结果。

3.1.7.4 输入项

表3.1.7.4.1 用户修改收货地址模块输入项列表

名称	标识	类型和格式	输入方式
旧地址	OAddress	String[]	用户点击

地址	NAddress	String[]	键盘输入
提交	Submit	Button	用户点击

3.1.7.5 输出项

表3.1.7.5.1 用户修改收货地址模块输出项列表

名称	标识	类型和格式	输出方式
结果	Res	Enum{成功，失败}	msgbox

3.1.7.6 设计方法

客户端把注册表单提交给服务器，服务器执行如下代码：

```
<?php
    public function ChangeAddress($info){
        UpdateAddress ($info)
    }
?>
```

3.1.7.7 流程逻辑

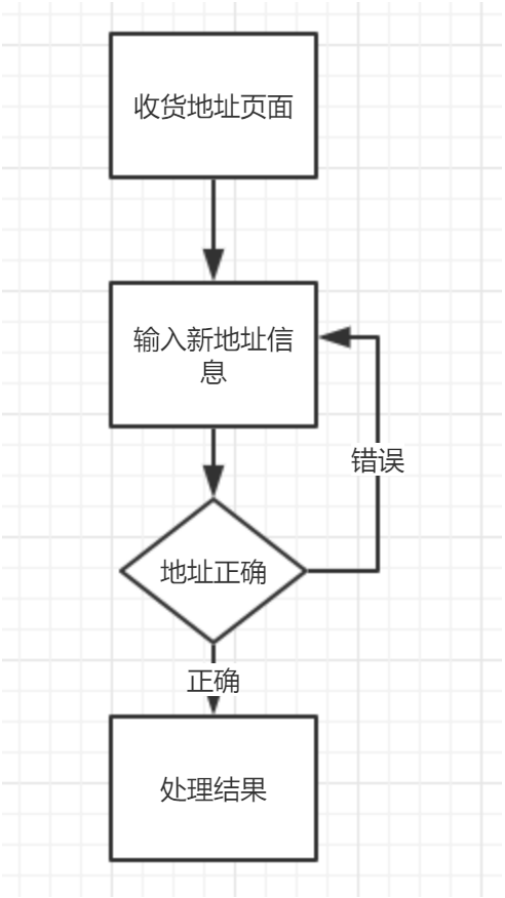


表3.1.7.7.1 用户修改收货地址模块流程图

3.1.7.8 接口

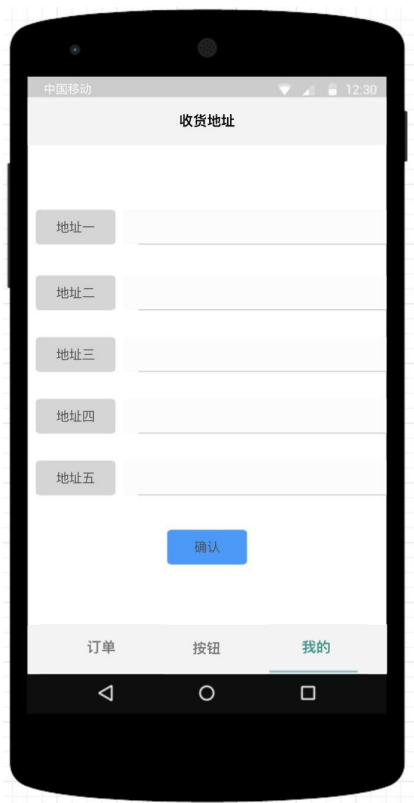


图3.1.7.8.1 用户修改收货地址模块用户接口

3.1.7.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.1.7.9.1 用户修改收货地址模块测试列表

测试用例	测试数据	预期结果
UC-BY-07-001	地址为空(浙江大学->NULL)	提示地址为空
UC-BY-07-002	地址合理(浙江大学->清华大学)	提示绑定成功

3.2 按钮子系统设计说明

3.2.1 绑定按钮模块

3.2.1.1 模块描述

用户需登录 APP ,在绑定银行卡和填写了收货地址之后可以进行绑定按钮操作。先进行商品条形码扫描,获取商品信息,选择商品数量和收货地址,将 EZS 贴在手机后面进行绑定操作。将按钮和用户、商品进行绑定,绑定后可以进行一键购物操作。

3.2.1.2 功能

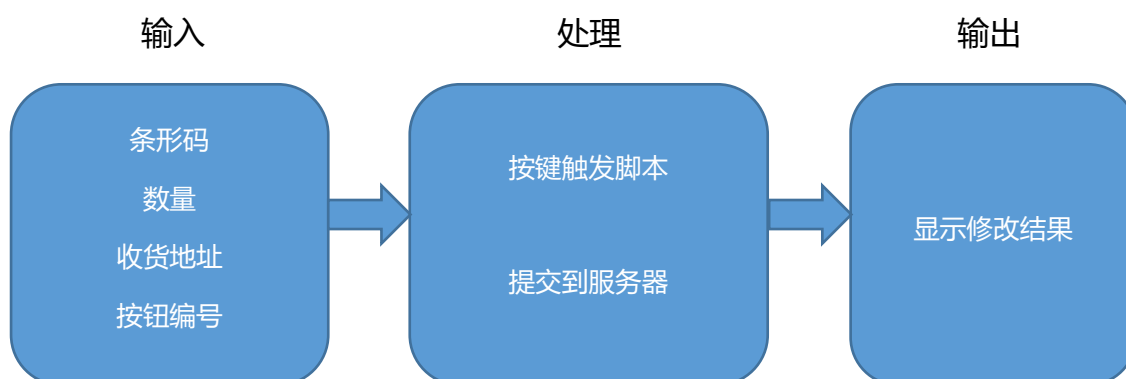


图3.2.1.2.1 绑定按钮模块IPO图

3.2.1.3 性能

客户端进行条形码扫描确定商品,并选择单次数量和收货地址,通过 NFC 读取按钮编号,点击绑定,信息通过表单传递到后台,触发脚本函数 BindButton(),将数据传入服务器。

3.2.1.4 输入项

表3.2.1.4.1 用户修改密码模块输入项列表

名称	标识	类型和格式	输入方式
商品编号	G_ID	string	条形码扫描
单词数量	G_Number	string	下拉框选择
收获地址	U_Address	string	下拉框选择
按钮编号	B_ID	string	NFC 读取
绑定	bind_button	button	手指触摸

3.2.1.5 输出项

表3.2.1.5.1 用户修改密码模块输出项列表

名称	标识	类型和格式	输出方式
修改结果	Button_feedback	Enum{修改成功， 商品不存在，EVS 已绑定商品}	Msgbox

3.2.1.6 设计方法

这个模块需要两个客户端和服务器的交互,第一次在扫描条形码获取信息后需要和服务器的商品信息进行匹配,返回商品信息;第二次在点击绑定按钮之后,将绑定信息储存到服务器数据库。服务器会执行以下代码:

```
<?php
    public function GetGoodsInfo($goodsid){
        $goodsinfo=GetGoodsInfoFromDatabase($goodsid);
        return $goodsinfo;
    }

    public function Bind($info){
        $result=UpdateButtonToDatabase($info);
        return $result;
    }

?>
```


3.2.1.7 流程逻辑

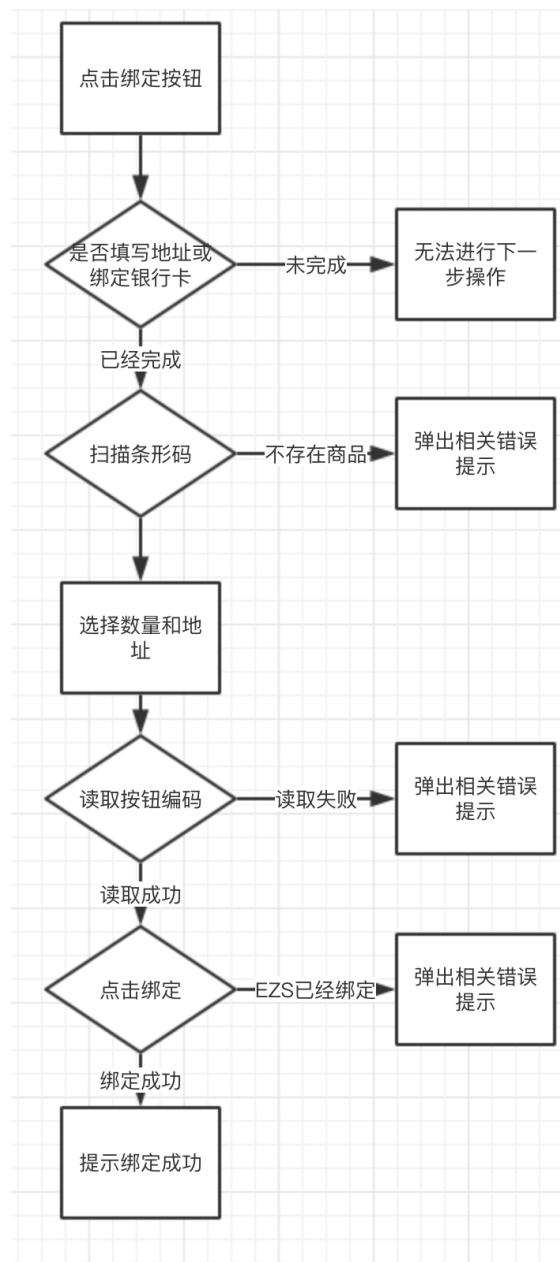


图3.2.1.7.1 绑定按钮模块流程图

3.2.1.8 接口

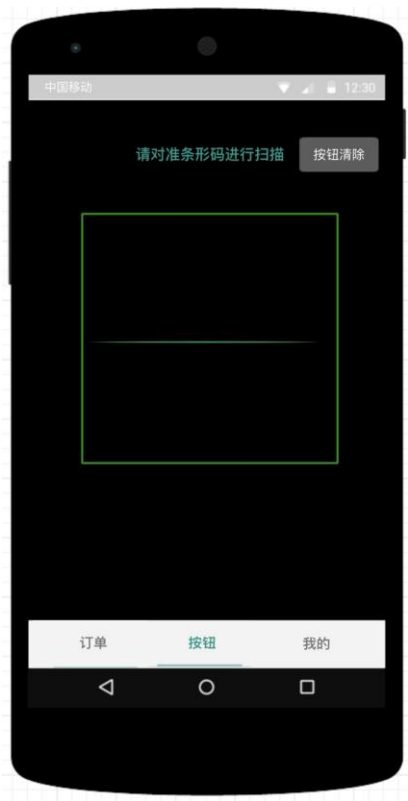


图3.2.1.8.1 绑定按钮模块用户接口

3.2.1.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.2.1.9.1 按钮绑定模块输出项列表

测试用例	测试数据	预期结果
UC-BY-08-001	错误的条形码	Msgbox(无法识别商品)
UC-BY-08-002	条形码代表的商	Msgbox(无法识别商品)

	品不存在	
UC-BY-08-003	EZS 出现故障	Msgbox(无法读取按钮编码)
UC-BY-08-004	未填写地址或未 绑定银行卡	无法进行绑定按钮操作（无法进入界面）
UC-BY-08-005	正确的条形码， EZS 正常读取	Msgbox(绑定成功)

3.2.2 查看按钮模块

3.2.2.1 模块描述

用户登录后，点击 EZS 按键，可以查看已经绑定到本账户的按钮。

3.2.2.2 功能



图3.2.2.2.1 查看按钮模块IPO图

3.2.2.3 性能

用户点击按钮按键后，即可查看所有和此账户绑定的按钮信息。

3.2.2.4 输入项

表3.2.2.4.1 查看按钮模块输入项列表

名称	标识	类型和格式	输入方式
EZS 按钮	Button_button	Button	鼠标点击

3.2.2.5 输出项

表3.2.2.5.1 查看按钮模块输出项列表

名称	标识	类型和格式	输出方式
按钮列表	Buttonlist	String[]	页面显示

3.2.2.6 设计方法

当用户点击屏幕上的按钮按键后，进入显示按钮列表页面，页面显示所有已经绑定的按钮信息。在完成这个操作时，客户端会提交给服务器一个查询请求，服务器会执行如下代码：

```
<?
public function GetButtonInfo($user){
    $buttonlist=GetButtonInfoByUser($user);
    return $buttonlist;
```

```
}  
?>
```

3.2.2.7 流程逻辑

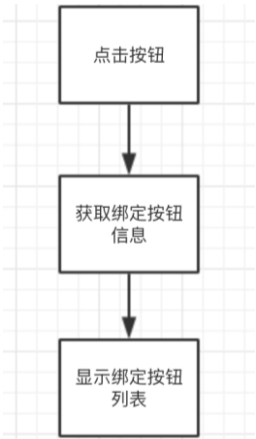


图3.2.2.7.1 查看按钮模块流程图

3.2.2.8 接口



图3.2.2.8.1 查看按钮模块用户接口

3.2.2.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.2.2.9.1 查看模块输出项列表

测试用例	测试数据	预期结果
UC-BY-13-001	点击按钮按键	显示绑定按钮列表

3.2.3 清除按钮模块

3.2.3.1 模块描述

用户登录并进入按钮列表，将列表中的按钮选项左移并点击删除即可完成清空按钮操作。

3.2.3.2 功能

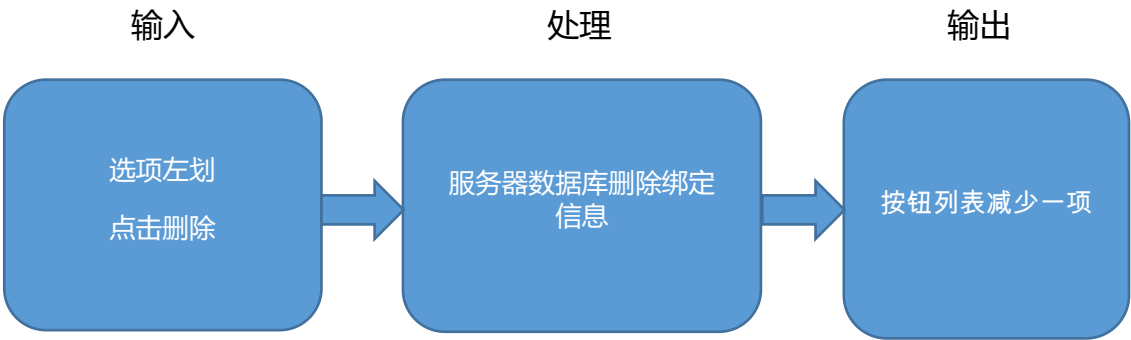


图3.2.3.2.1 清除按钮模块IPO图

3.2.3.3 性能

按钮选项左划，点击删除完成清除操作。

3.2.3.4 输入项

表3.2.3.4.1 清除按钮模块输入项列表

名称	标识	类型和格式	输入方式
删除按钮	Delete_button	Button	手指触摸

3.2.3.5 输出项

表3.2.3.5.1 清除按钮模块输出项列表

名称	标识	类型和格式	输出方式
按钮列表	Buttonlist	String[]	页面显示
操作结果	Feekback	Bool	Msgbox

3.2.3.6 设计方法

用户左划并点击删除后发送给服务器一个请求，服务器返回操作结果和按钮列表，服务器会执行如下代码：

<?

```
public function Bind($info){  
    $result=UpdateButtonToDatabase($info);  
    return $result;  
}
```

?>

3.2.3.7 流程逻辑

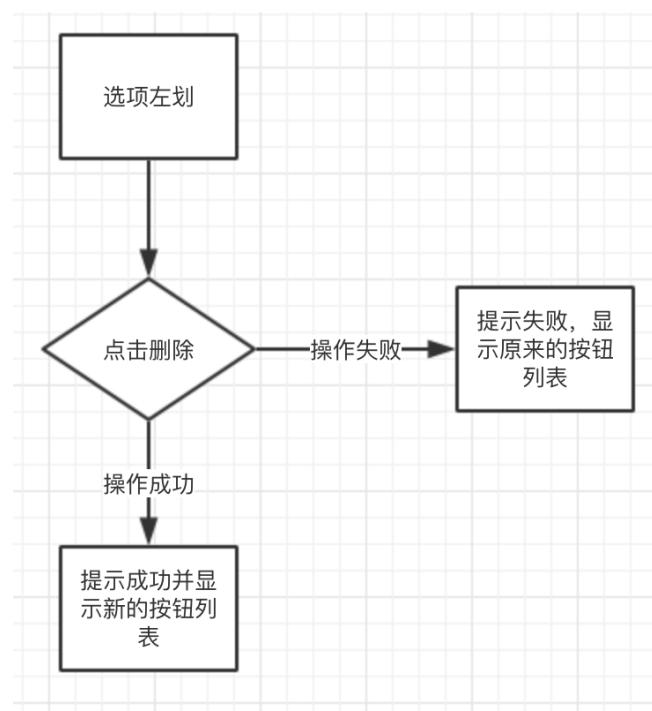


图3.2.3.7.1 清除按钮模块流程图

3.2.3.8 接口



图3.2.3.8.1 清除按钮模块用户接口

3.2.3.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.2.3.9.1 清除模块输出项列表

测试用例	测试数据	预期结果
UC-BY-09-001	用户未绑定按钮	无法进行清除操作
UC-BY-09-002	已经绑定按钮	完成清除操作

3.3 订单子系统设计说明

3.3.1 用户下订单模块

3.3.1.1 模块描述

用户点击物理按钮，即发送订单信息，同时手机 APP 端接收到订单信息。

若商品余量不足或绑定银行卡余额不足，则订单发送失败，手机 APP 端接收到失败原因。若下单成功，则订单同时发送给买家和卖家，用户自动付款。

3.3.1.2 功能

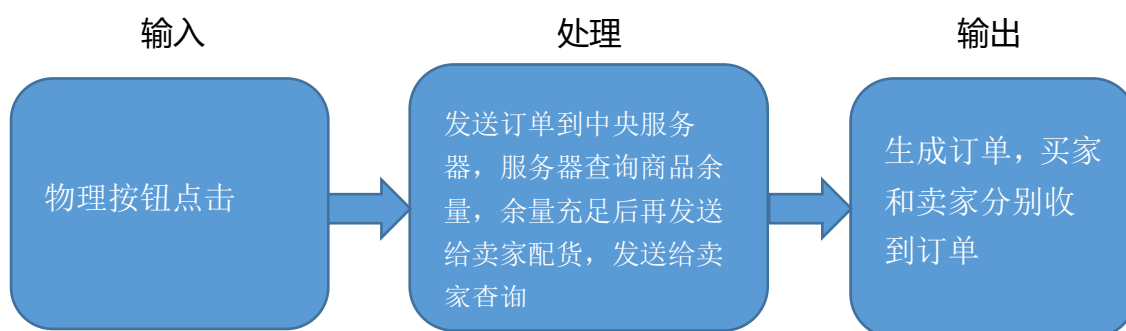


图3.3.1.2.1 用户下订单模块IPO图

3.3.1.3 性能

用户点击物理按钮后，服务器查询检测相关数据，符合则订单下单成功，随后将订单同时发给买家卖家。

3.3.1.4 输入项

表3.3.1.4.1 用户下订单模块输入项列表

名称	标识	类型和格式	输入方式
物理按钮	aksid	实体按键	点击按钮

3.3.1.5 输出项

表3.3.1.5.1 用户下订单模块输出项列表

名称	标识	类型和格式	输出方式
订单号	orderid	String	文本信息
商品号	goodsid	String	文本信息
订单状态	ordercondition	String	文本信息
订单日期	orderdate	date	文本信息
收货地址	address	String	文本信息

3.3.1.6 设计方法

当用户点击物理按钮，调用函数 CheckOrder()检测订单是否符合要求，若符合则生成订单同时发送给买家和卖家。

< ?

```
ConnectionDB Db; //连接数据库
Db.open();
if($getPressedInfo){ //按击物理按钮
    $order=GetOrder(); //生成订单信息
    if(CheckOrder($order)){
        BuyerPayMoney(); //买家付款
        PutOrdeToSeller(); //推送订单给卖家
        PutOrderToBuyer(); //推送订单给买家
    }
    else {
        DeleteOrder(); //销毁订单
        PutFailInfoToBuyer(); //推送失败原因给用户
    }
}
Db.close(); //关闭数据库连接
```

? >

3.3.1.7 流程逻辑

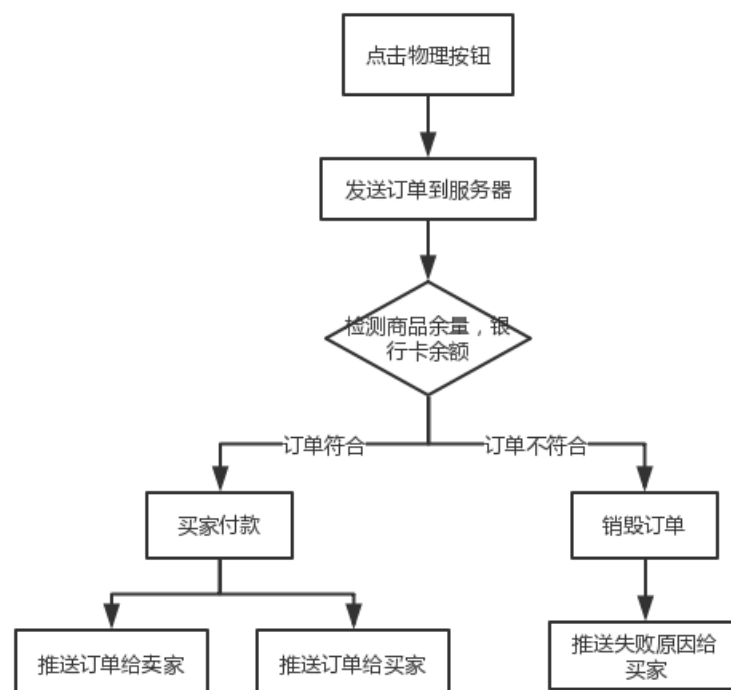


图3.3.1.7.1 用户下订单模块流程图

3.3.1.8 接口



图3.3.1.8.1 用户下订单模块用户接口（左买家，右卖家）

3.3.1.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.3.1.9.1 用户下订单模块输出项列表

测试用例	输入数据	预期结果
UC-BY-10-001	点击已匹配商品的按钮，且商品 余量充足，银行卡余额充足	下单成功，买家卖家 APP 端接均收到订单信息
UC-BY-10-002	点击未匹配商品的按钮	下单失败"

UC-BY-10-003	点击已匹配商品的按钮，但商品 余量不足	下单失败，买家 APP 端接 收到订单失败原因提示
UC-BY-10-004	点击已匹配商品的按钮，但银行 卡余额不足	下单失败，买家 APP 端接 收到订单失败原因提示

3.3.2 用户查看订单模块

3.3.2.1 模块描述

用户点击进入 APP，点击订单，查看订单信息。

3.3.2.2 功能

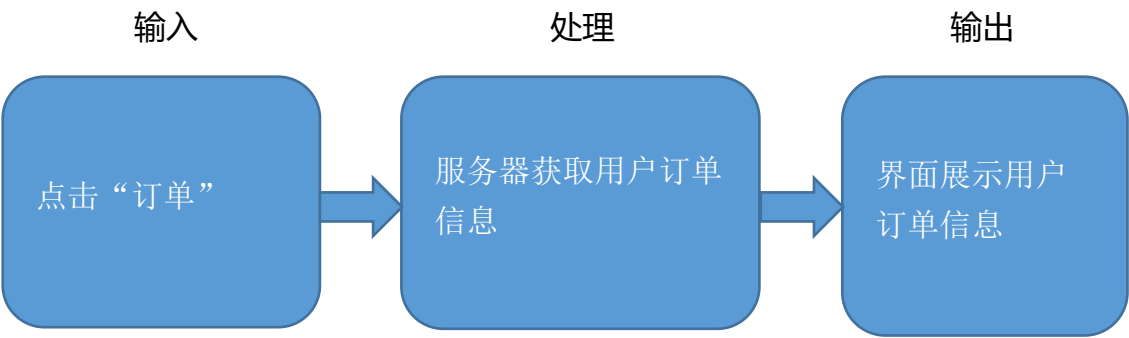


图3.3.2.2.1 用户查看订单模块IPO图

3.3.2.3 性能

用户点击进入 APP，登录后点击订单，界面展示用户的订单信息。

3.3.2.4 输入项

表3.3.2.4.1 用户查看订单模块输入项列表

名称	标识	类型和格式	输入方式
订单	order	button	单击按钮

3.3.2.5 输出项

表3.3.2.5.1 用户查看订单模块输出项列表

名称	标识	类型和格式	输出方式
订单列表	orderid	String	文本信息
商品号	goodsid	String	文本信息
订单状态	ordercondition	String	文本信息
订单日期	orderdate	date	文本信息

3.3.2.6 设计方法

当用户登录后点击“订单”，APP 向服务器发送订单信息请求，随后界面显示服务器返回的订单信息。

```
< ?  
    If(session("login")) //检测用户已登录  
    {  
        ConnectionDB Db; //连接数据库  
        Db.open();  
        SelectOrderByPhone();
```

```
        Db.close(); //关闭数据库连接
        ShowOrderList();
    }
    echo "尚未登录"
? >
```

3.3.2.7 流程逻辑

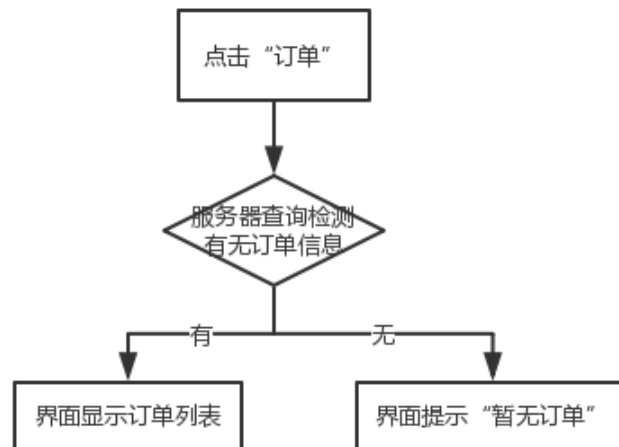


图3.3.2.7.1 用户查看订单模块流程图

3.3.2.8 接口



图3.3.2.8.1 用户查看订单模块用户接口

3.3.2.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.3.2.9.1 用户查看订单模块测试列表

测试用例	输入数据	预期结果
UC-BY-11-001	用户已有下单记录，查看订单	显示详细订单信息
UC-BY-11-002	用户暂无下单记录，查看订单信息	订单列表为空

3.3.3 用户确认收货模块

3.3.3.1 模块描述

用户点击进入 APP，点击订单，选择某个具体的订单确认收货。

3.3.3.2 功能

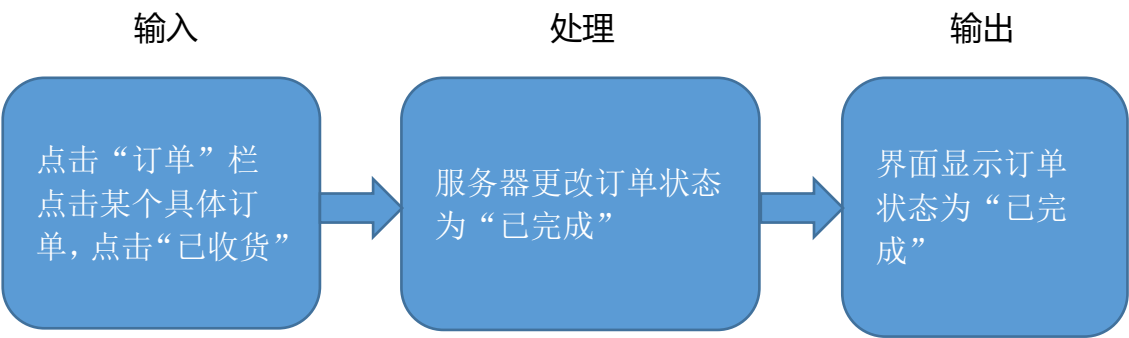


图3.3.3.2.1 用户确认收货模块IPO图

3.3.3.3 性能

用户点击进入 APP，登录后点击订单，选择某个具体的订单，点击“已收货”，界面显示订单状态为“已完成”。

3.3.3.4 输入项

表3.3.3.4.1 用户确认收货模块输入项列表

名称	标识	类型和格式	输入方式
订单状态	ordercondition	button	单击按钮

3.3.3.5 输出项

表3.3.3.5.1 用户确认收货模块输出项列表

名称	标识	类型和格式	输出方式
订单号	orderid	String	文本信息
商品号	goodsid	String	文本信息
订单状态	ordercondition	String	文本信息
订单日期	orderdate	date	文本信息

3.3.3.6 设计方法

当用户登录后点击“订单”，选择某个具体的订单，点击“已收货”。服务器更改该订单状态，并将买家和卖家的该订单状态均改为“已完成”。

```
< ?
  If(session("login")) //检测用户已登录
  {
    ConnectionDB Db; //连接数据库
    Db.open();
    UpdataOrderByOederid();
    Db.close(); //关闭数据库连接
  }
  echo "尚未登录"
? >
```

3.3.3.7 流程逻辑

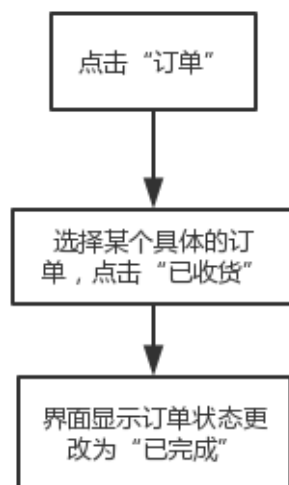


图3.3.3.7.1 用户确认收货模块流程图

3.3.3.8 接口



图3.3.3.8.1 用户确认收货模块用户接口

3.3.3.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.3.3.9.1 用户确认收货模块测试列表

测试用例	输入数据	预期结果
UC-BY-12-001	用户已有下单记录，点击“已收货”	订单状态更改为“已完成”
UC-BY-12-002	用户无下单记录	无订单列表，无“已收货”按钮接口

3.3.4 卖家发货模块

3.3.4.1 模块描述

卖家登陆后，进入订单管理模块，选择某个“已下单”状态订单确认发货。

3.3.4.2 功能

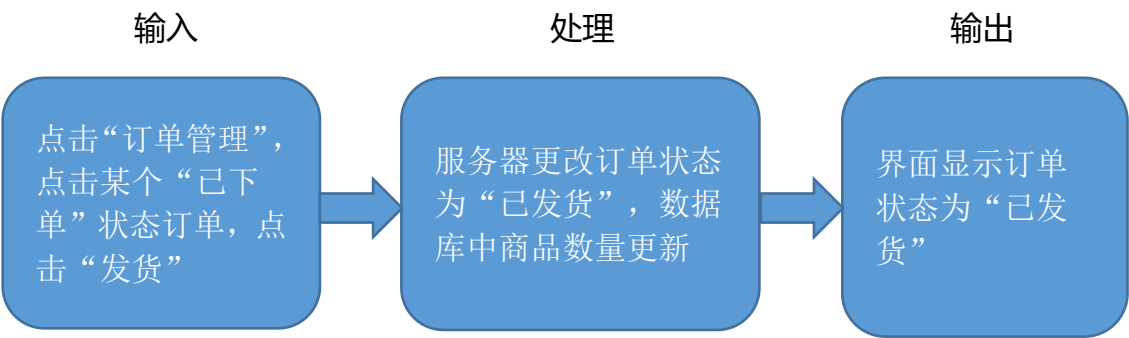


图3.3.4.2.1 卖家发货模块IPO图

3.3.4.3 性能

卖家登录后进入订单管理模块，选择某个状态为“已下单”的订单，点击“发货”，界面显示订单状态为“已发货”，数据库中该商品总量更新。

3.3.4.4 输入项

表3.3.4.4.1 卖家发货模块输入项列表

名称	标识	类型和格式	输入方式
订单状态	ordercondition	button	单击按钮

3.3.4.5 输出项

表3.3.4.5.1 卖家发货模块输出项列表

名称	标识	类型和格式	输出方式
订单号	orderid	String	文本信息
商品号	goodsid	String	文本信息
订单状态	ordercondition	String	文本信息
订单日期	orderdate	date	文本信息
商品数量	goodsnum	int	文本信息

3.3.4.6 设计方法

卖家选择某个状态为“已下单”的订单，点击“发货”。服务器更改该订单状态为“已发货”，数据库中该商品总量根据订单中的购买数量更新。

```
< ?
    If(session("seller_login")) //检测卖家已登录
    {
        ConnectionDB Db; //连接数据库
        Db.open();
        orderid=$_GET[ "orderid" ];
        UpdateOrderByOrderid();
        Db.close(); //关闭数据库连接
    }
    echo "尚未登录"
? >
```

3.3.4.7 流程逻辑

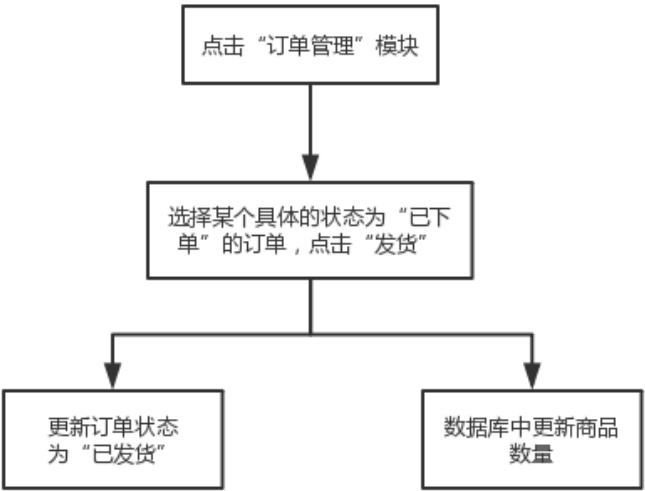


图3.3.4.7.1 卖家发货模块流程图

3.3.4.8 接口



图3.3.4.8.1 卖家发货模块用户接口

3.3.4.9 测试计划

测试要求通过远程的计算机，创建相关的数据库，并添加测试数据。

表3.3.4.9.1 卖家发货模块测试列表

测试用例	输入数据	预期结果
UC-SL-01-001	用户已有下单记录， 卖家发货	1. 订单状态更改为“已发货” 2. 数据库中商品数量更新
UC-SL-01-002	用户无下单记录	无订单列表，无“已发货”按钮接口

3.4 商品信息管理模块设计说明

3.4.1 添加商品信息模块

3.4.1.1 模块描述

本模块主要添加商品信息，卖家可以添加商品。

3.4.1.2 功能

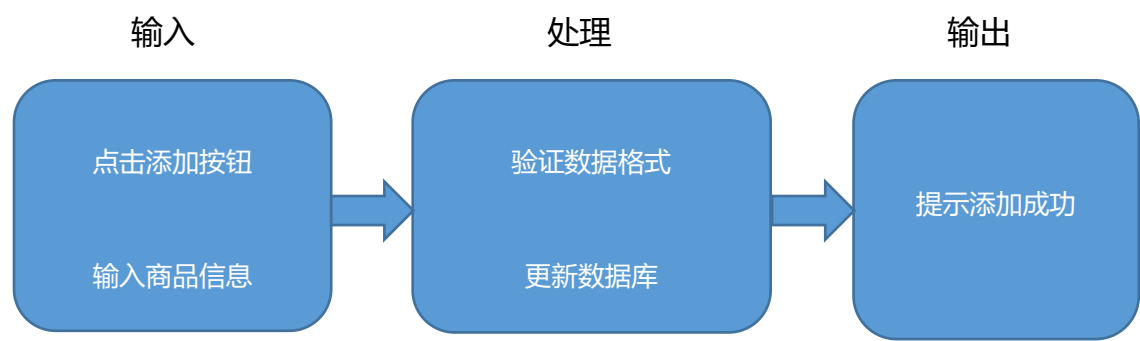


图3.4.1.2.1 添加商品信息模块IPO图

3.4.1.3 性能

本模块的性能要求不大，只对数据的正确性有要求。

3.4.1.4 输入项

表3.4.1.4.1 添加商品信息模块输入项列表

名称	标识	类型和格式	输入方式
商品名	goodsname	string	文本框输入
商品数量	goodsnum	string	文本框输入
商品价格	goodsprice	string	文本框输入
添加	add_buttonn	button	触屏点击

3.4.1.5 输出项

表3.4.1.5.1 添加商品信息模块输出项列表

名称	标识	类型和格式	输出方式
添加结果	add_feedback	Enum(添加成功、错误)	Msgbox

3.4.1.6 设计方法

当卖家客户端点击添加时，跳转到添加视图，待用户输入，输入完之后点击提交，触发脚本函数 AddGoods(), 将信息提交给服务器，服务器执行如下代码

< ?

```
ConnectionDB Db ;  
Db.open();  
Save(goodsname,goodsnum,goodsprice);//添加新商品  
Db.close();  
//关闭数据库连接  
echo "添加成功";  
Page flash();//刷新页面
```

? >

3.4.1.7 流程逻辑

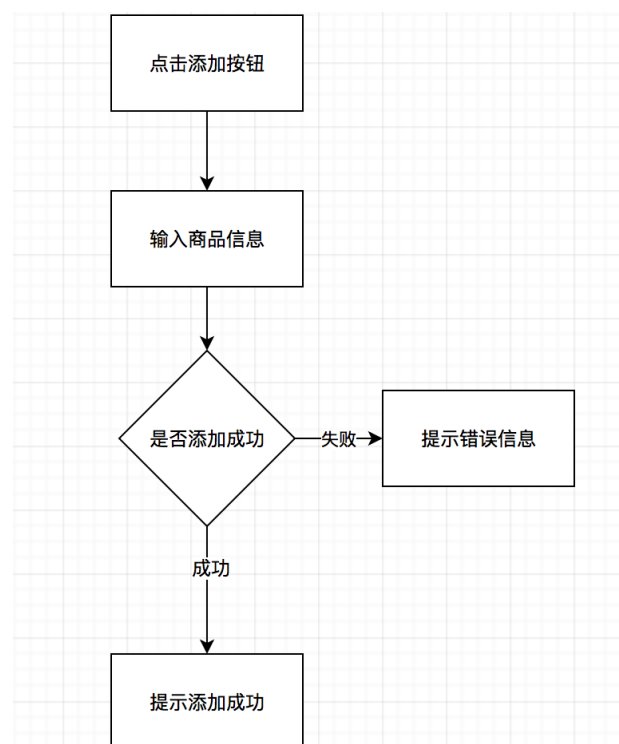


图3.4.1.7.1 添加商品信息模块流程图

3.4.1.8 接口



图3.4.1.8.1 添加商品信息模块用户接口

3.4.1.9 测试计划

测试要求通过连接远程的计算机，创建相关的数据库，并添加测试数据。

表3.4.1.9.1 添加商品信息模块测试计划表

测试用例	测试数据	预期结果
------	------	------

UCSL02001	输入符合规则的商品信息	添加成功
UCSL02002	商品名为空	提示商品名不能为空
UCSL02003	商品数量为空	提示商品数量不能为空
UCSL02004	商品价格为空	提示商品价格不能为空
UCSL02005	商品名输入特殊字符	提示商品名不允许特殊字符
UCSL02006	商品数量（商品价格）含有数字以外字符	提示商品数量（商品价格）格式不正确

3.4.2 删除商品信息模块

3.4.2.1 模块描述

本模块主要删除商品信息，卖家可以删除商品。

3.4.2.2 功能

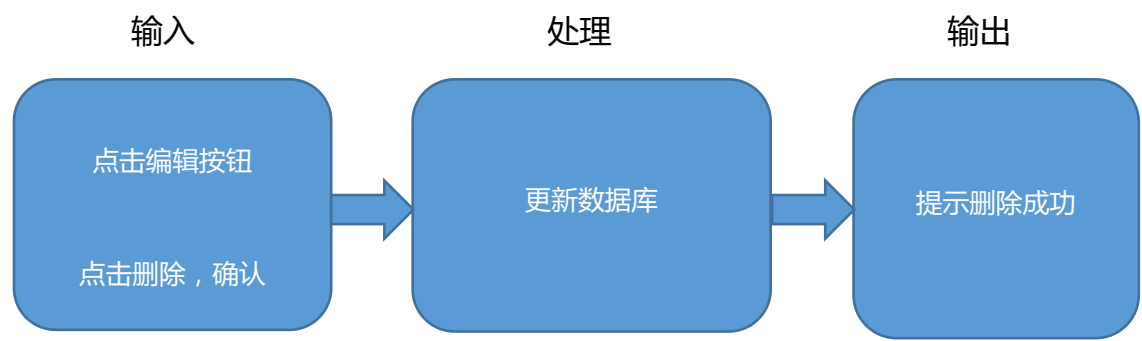


图3.4.3.2.1 删除商品信息模块IPO图

3.4.2.3 性能

本模块的性能要求不大，只对数据的正确性有要求。

3.4.2.4 输入项

表3.4.2.4.1 删除商品信息模块输入项列表

名称	标识	类型和格式	输入方式
删除	delete_button	button	触屏点击

3.4.2.5 输出项

表3.4.2.5.1 删除商品信息模块输出项列表

名称	标识	类型和格式	输出方式
删除结果	delete_feedback	Enum(删除成功、错误)	Msgbox

3.4.2.6 设计方法

当卖家客户端点击编辑时，左滑商品列表可以删除，确认，触发脚本函数 DeleteGoods()，将信息提交给服务器，服务器执行如下代码

```
< ?
    ConnectionDB Db ;
    Db.open();
    Detele(goodsid);//删除id对应的商品
    Db.close();
    //关闭数据库连接
    echo "删除成功";
    Page flash();//刷新页面
? >
```

3.4.2.7 流程逻辑

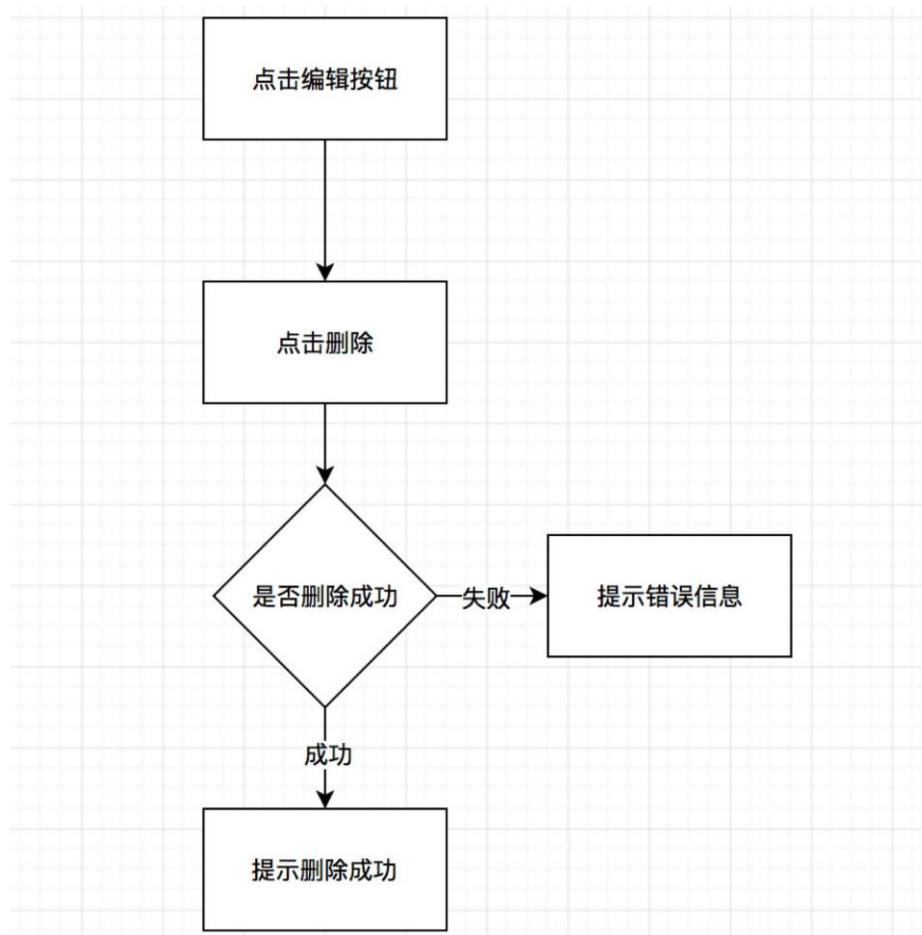


图3.4.2.7.1 删除商品信息模块流程图

3.4.2.8 接口



图3.4.2.8.1 删除商品信息模块用户接口

3.4.2.9 测试计划

测试要求通过连接远程的计算机，创建相关的数据库，并添加测试数据。

表3.4.2.9.1 删除商品信息模块测试计划表

测试用例	测试数据	预期结果
UCSL02013	点击删除点击取消	商品未删除
UCSL02014	点击删除点击确认	提示商品删除成功

3.4.3 编辑商品信息模块

3.4.3.1 模块描述

本模块主要编辑商品信息，卖家可以修改商品信息。

3.4.3.2 功能

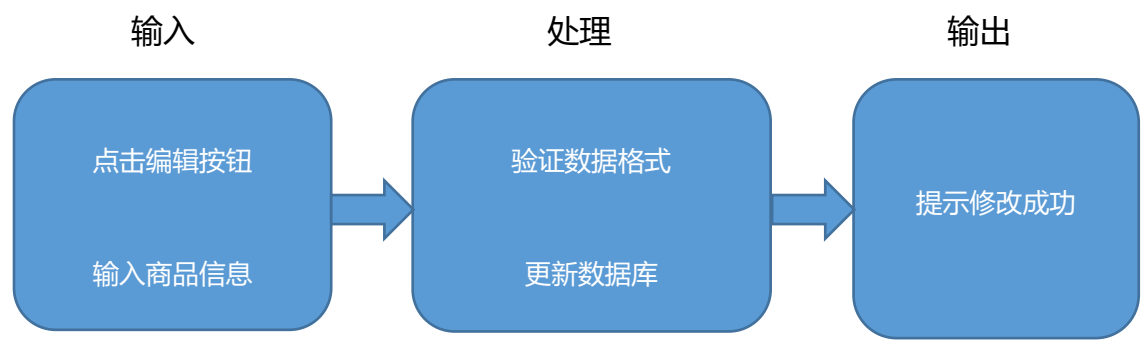


图3.4.3.2.1 修改商品信息模块IPO图

3.4.3.3 性能

本模块的性能要求不大，只对数据的正确性有要求。

3.4.3.4 输入项

表3.4.3.4.1 修改商品信息模块输入项列表

名称	标识	类型和格式	输入方式
商品名	goodsname	string	文本框输入

商品数量	goodsnum	string	文本框输入
商品价格	goodsprice	string	文本框输入
编辑	modify_button	button	触屏点击

3.4.3.5 输出项

表3.4.3.5.1 修改商品信息模块输出项列表

名称	标识	类型和格式	输出方式
修改结果	modify_feedback	Enum(修改成功、错误)	Msgbox

3.4.3.6 设计方法

当卖家客户端点击修改时，跳转到修改视图，待用户输入，输入完之后点击提交，触发脚本函数 UpdateGoods(), 将信息提交给服务器，服务器执行如下代码

```
< ?
    ConnectionDB Db ;
    Db.open();
    Update(goodsname,goodsnum,goodsprice);//更新商品信息
    Db.close();
    //关闭数据库连接
    echo "更新成功";
    Page flash();//刷新页面
? >
```

3.4.3.7 流程逻辑

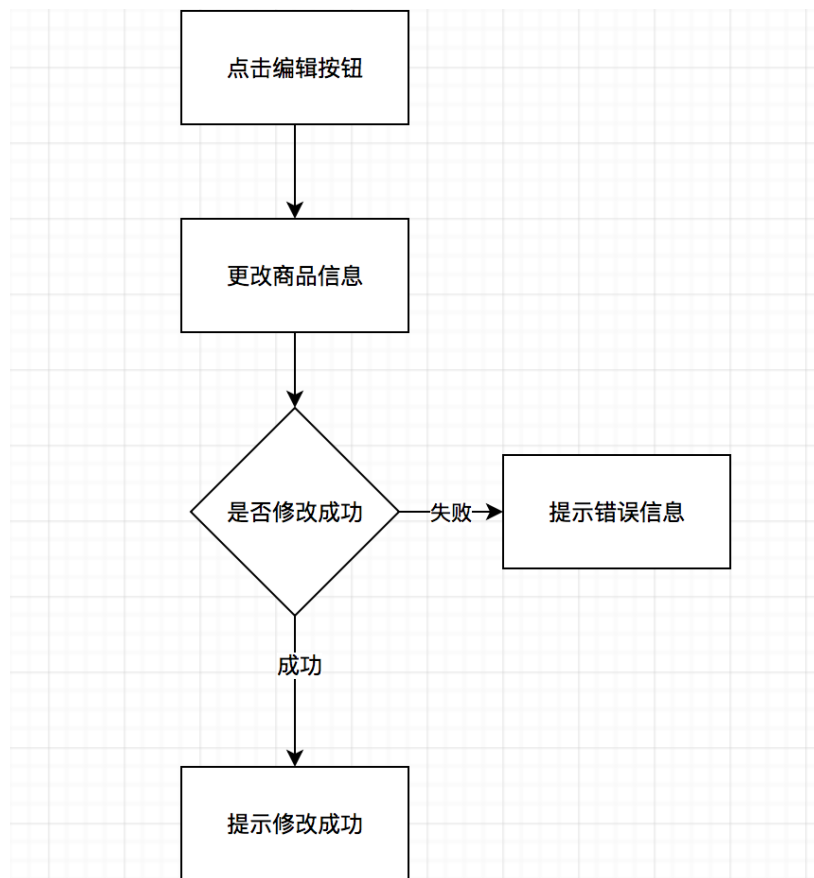


图3.4.3.7.1 修改商品信息模块流程图

3.4.3.8 接口



图3.4.3.8.1 修改商品信息模块用户接口

3.4.3.9 测试计划

测试要求通过连接远程的计算机，创建相关的数据库，并添加测试数据。

表3.4.3.9.1 修改商品信息模块测试计划表

测试用例	测试数据	预期结果
UCSL02007	输入符合规则的商品信息	修改成功

UCSL02008	商品名为空	提示商品名不能为空
UCSL02009	商品数量为空	提示商品数量不能为空
UCSL02010	商品价格为空	提示商品价格不能为空
UCSL02011	商品名输入特殊字符	提示商品名不允许特殊字符
UCSL02012	商品数量（商品价格）含有数字以外字符	提示商品数量（商品价格）格式不正确

4.接口设计

4.1 界面设计

4.1.1 用户登录/用户主页模块



图4.1.1.1 用户主页



图4.1.1.2 用户登录



图 4.1.1.3 绑定银行卡

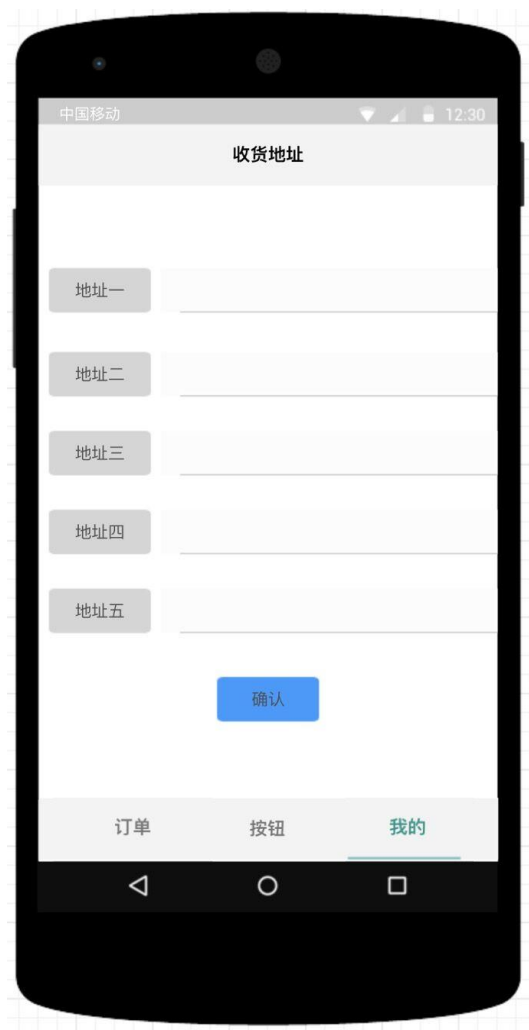


图 4.1.1.4 绑定收货地址

4.1.2 用户注册模块



图 4.1.2.1 用户注册

4.1.3 按钮模块

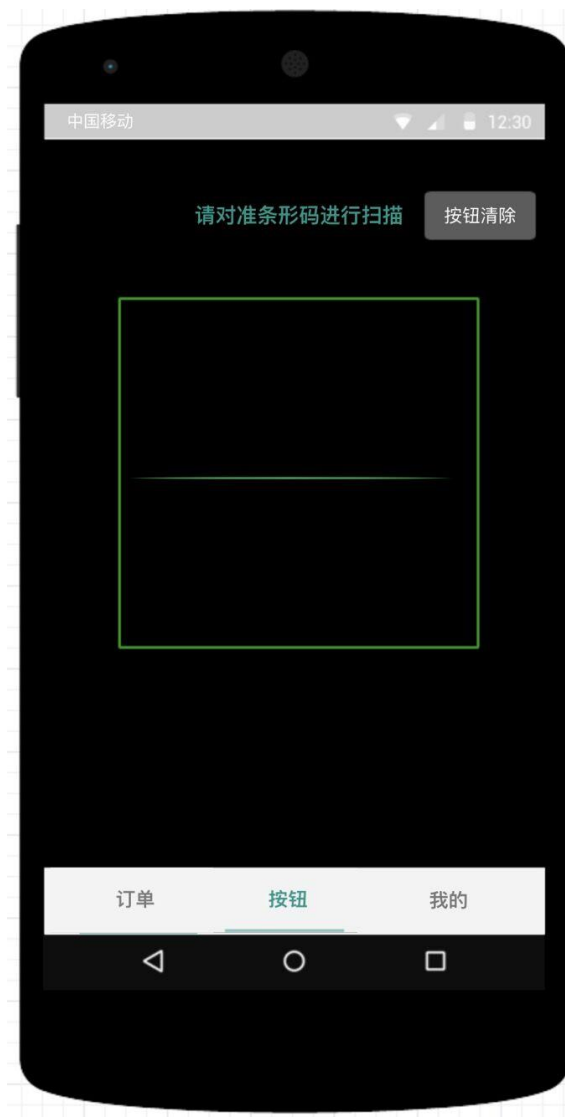


图4.1.3.1 扫描条形码

4.1.4 订单管理模块



图4.1.4.1 买家订单管理



图 4.1.4.2 卖家订单管理

4.2 外部接口

4.2.1 硬件接口

4.2.1.1 服务器

表 4.2.1.1.1 服务器端硬件接口

项目	信息
处理器	Intel i 5
内存	ECC DDR3 8
存储	物理磁盘，SCSI 接口转速 10000 转
网卡	支持浙江大学校园网(L2TP)、100M
备份	RAID5

4.2.1.2 客户端

表 4.2.1.2.1 客户端硬件接口

项目	信息
操作系统	Android 4.0 以上
内存	128M 以上

4.2.1.3 NFC

表 4.2.1.2.2 NFC 硬件接口

项目	信息
ID	按钮出场自带的 ID

4.2.2 软件接口

4.2.2.1 服务器

表 4.2.2.1.1 服务器端软件接口

项目	信息
----	----

操作系统	Windows7
服务器软件	Apache 5.5
数据库软件	MySQL 5.6

4.2.2.2 客户端

表 4.2.2.2.1 客户端软件接口

项目	信息
操作系统	Android
移动终端应用	Android-APP

4.2.2.3 银行接口

表 4.2.2.3.1 客户端软件接口

项目	信息
账号	用户 APP 登录账号的绑定银行卡
密码	银行卡密码
付款额	交易金额

4.2.3 通信接口

本系统使用的通信功能有电子邮件、Web 浏览器、网络通信标准或协议及通信安全或加密问题、10/100M 数据传输速率和同步通信机制。

表格 4.2.3.1 通信接口

项目	信息
网络环境	浙江大学校园网(L2TP)

5.系统数据结构设计

5.1 概念结构设计

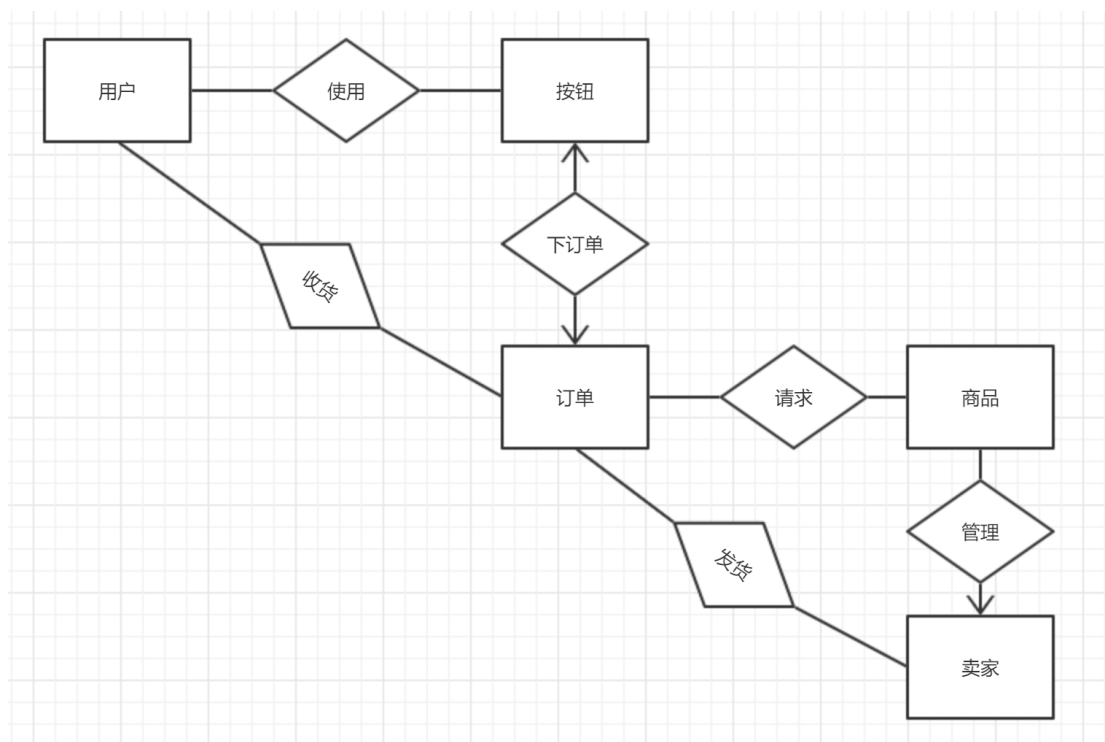


图 5.1.1 E-R图

5.2 逻辑结构设计

用户信息：userinfo(username, passwd, email, phone, creditcard)

一键购物按钮：aks(aksid, phone, goodsid, address, goodsnum)

商品信息：goods(goodsid, goodsname, goodsnum, goodsprice)

地址 : address(phone, address)

订单 : order(ordered, goodsid, phone, address, condition, date, goodsnum)

5.3 物理结构设计

1. userinfo(用户信息)

字段	类型	是否可为空	是否主键	是否 Unique	备注
username	varchar(50)	N	N	N	用户名
passwd	varchar(50)	N	N	N	密码
email	varchar(50)	N	N	N	邮箱
phone	int(11)	N	Y	Y	手机号
creditcard	int(19)	N	N	N	银行卡

2. aks(一键购物按钮)

字段	类型	是否可为空	是否主键	是否 Unique	备注
aksid	int(10)	N	Y	Y	按钮 id
phone	int(11)	N	N	Y	手机号

goodsid	int(10)	N	N	Y	商品 id
address	varchar(255)	N	N	N	收货地址
goodsnum	int(10)	N	N	N	购买数量

3. goods(商品信息)

字段	类型	是否可为空	是否主键	是否 Unique	备注
goodsid	int(10)	N	Y	Y	商品 id
goodsname	varchar(255)	N	N	N	商品名
goodsnum	int(10)	N	N	N	库存
goodsprice	decimal(5,2)	N	N	N	价格

4. address(地址)

字段	类型	是否可为空	是否主键	是否 Unique	备注
phone	int(11)	N	Y	Y	手机号
address	varchar(255)	N	N	N	收货地址

5. order(订单)

字段	类型	是否可为空	是否主键	是否 Unique	备注
----	----	-------	------	-----------	----

orderid	int(10)	N	Y	Y	订单 id
goodsid	int(10)	N	N	Y	商品 id
phone	int(11)	N	N	Y	手机号
address	varchar(255)	N	N	N	收货地址
goodsnum	int(10)	N	N	N	购买数量
condition	int	N	N	N	订单状态
date	date	N	N	N	日期

6.运行设计

6.1 运行模块组合

本系统按照不同用户类型划分模块，每个模块又按流程划分为客户端界面，客户端脚本，服务器后台程序。功能模块之间相互不会共享界面(主界面除外)，客户端脚本只共用一个创建 XMLHttpRequest 类的方法，后台程序只共享建立数据库连接的方法。

6.2 运行控制

不同类型的用户能够进入不同的页面，通过页面上的导航选择不同功能。

6.3 运行时间

由于本系统的功能简易、明确、高效，本系统运行速度快，因此唯一会影响系统的运行时间的，应该只是系统频繁建立销毁数据库连接所带来的时间开销。

7.系统出错处理设计

7.1 出错信息

图7.1.1 系统出错一览表

系统输出信息的形式	含义	处理方法
数据库连接不上	由于并发操作的用户数量很大，导致数据库访问读写效率降低。或者数据库配置不对，导致数据库连接失败。	修改数据库配置，并限制同一个用户两次访问数据库之间的时间间断限制。
Sql 语句执行错误	由于一些用户恶意在表单中插入sql 语句，企图破坏数据库，导致访问数据库出错。	对表单中的数据在服务器端进行过滤，检测。
网站信息丢失	由于一些用户恶意不通过登陆界面而直接访问网站或者后台，导致网站信息破坏或被窃取。	禁止直接访问一些需要登陆的网页。

服务器崩溃	服务器由于长时间连续运作，导致错误积累过多，负荷过大，最终崩溃。	定期对服务器进行维护
账户被盗	由于客户端被黑客植入木马或者其他方式引起用户账户被盗，密码被修改。	建议用户经常对电脑进行杀毒，不要在陌生的地点用其他人的电脑登陆客户端。
数据库中账户信息泄露	黑客通过后台下载服务器中的数据库，盗取所有用户的账户信息。	对账户中密码进行加密，如3重DES 加密算法，MD5加密算法等等。
磁盘损坏	由于需要频繁地操作数据库，导致磁盘损坏，数据丢失。	使用磁带对磁盘进行周期性地备份。
系统无法处理一些非法交易指令	股票代码不存在，购股需要金额远大于资金账户中可用金额。	在将指令存入数据库时，需要做指令合理性判断。
数据库读取乱码	在操作数据库过程中，由于写入出错或者读出出错导致出现乱码	一般出现乱码的错误为数据库编码与写入数据编码不一致缘故，此时需要统一编码

7.2 补救措施

7.2.1 后备技术

周期,频繁备份数据库,并且将其储存在更加稳定的介质上,比如磁带。

将系统所需的不同数据库部署到不同的计算机上,减小因硬件问题而导致数据全部丢失的可能性。

7.2.2 降效技术

紧急情况下,启用人工,代替系统撮合买卖指令。

7.2.3 恢复及再启动技术

系统崩溃后,通过系统运行日志记录恢复数据。

7.3 系统维护设计

连接数据库方面,需要在创建数据库连接,销毁数据库连接,执行sql 语句的模块使用try catch 语句捕获异常。于是发生此类错误便能很快得知。

网络方面需要检测是否有特定的IP 地址频繁访问系统,不能排除其攻击服务器的可能性。内部人员如(开户人员,系统维护人员)操作留下操作痕迹,使用权管理层可以定期或不定期地稽核系统。