****

大作业报告（论文）

**基于XX的XXXX系统**

**课程名称 网络程序设计**

**专 业 计算机（商务智能）**

**班 级 18计算机 （商务智能）**

**学 号 188888888**

**姓　 名 王阳明**

**指导教师 江左文**

**完成日期 2019 年 12 月 26**

**目　录**

[1 绪论 1](#_Toc512333357)

[1.1 选题背景 1](#_Toc512333358)

[1.2 系统目标 1](#_Toc512333359)

[1.3 运行环境要求 1](#_Toc512333360)

[1.4 论文具体结构 1](#_Toc512333361)

[2 相关技术知识 2](#_Toc512333362)

[2.1 工具知识 2](#_Toc512333363)

[2.1.1 Android Studio…… 2](#_Toc512333364)

[2.1.2 Visual Studio Code 2](#_Toc512333365)

[2.2 技术内容 2](#_Toc512333366)

[2.2.1 Android 2](#_Toc512333367)

[2.2.2 Node.js 3](#_Toc512333368)

[2.2.3 MySQL 3](#_Toc512333369)

[3 系统分析 5](#_Toc512333370)

[3.1 可行性研究 5](#_Toc512333371)

[3.1.1 经济可行性 5](#_Toc512333372)

[3.1.2 技术可行性 5](#_Toc512333373)

[3.1.3 操作可行性 5](#_Toc512333374)

[3.1.4 其他方面 6](#_Toc512333375)

[3.2 需求分析 6](#_Toc512333376)

[3.2.1 功能需求分析 6](#_Toc512333377)

[3.2.2 性能需求分析 7](#_Toc512333378)

[4 概要设计 8](#_Toc512333379)

[4.1 流程图 8](#_Toc512333380)

[4.2 E-R图 9](#_Toc512333381)

[4.3 功能图 12](#_Toc512333382)

[4-3.1首页 13](#_Toc512333383)

[4-3.1商家信息 13](#_Toc512333384)

[4-3.2选择外卖 13](#_Toc512333385)

[4-3.2订单 13](#_Toc512333386)

[4-3.2.1购买记录 14](#_Toc512333387)

[4-3.2.2订单详情 14](#_Toc512333388)

[4-3.3个人 14](#_Toc512333389)

[4-3.3.1登陆／注销／注册 14](#_Toc512333390)

[4-3.3.2饮食搭配 14](#_Toc512333391)

[4-3.3.3食堂新闻 14](#_Toc512333392)

[4-3.3.4设置性别 14](#_Toc512333393)

[4-3.3.5设置昵称 14](#_Toc512333394)

[4-3.3.6设置地址 14](#_Toc512333395)

[5 系统设计与实现 14](#_Toc512333396)

[5.1数据库设计 15](#_Toc512333397)

[5.1.1数据表设计 15](#_Toc512333398)

[5.2 App操作 1](#_Toc512333399)6

[5.2.1 加载图 1](#_Toc512333400)7

[5.2.2 底部切换 1](#_Toc512333400)8

[5.2.3 个人中心 .......19](#_Toc512333401)

[5.2.4 登录 2](#_Toc512333402)0

[5.2.5 注册 2](#_Toc512333403)1

[5.2.6 修改地址 22](#_Toc512333404)

[5.2.7 修改性别 2](#_Toc512333405)2

[5.2.8 修改昵称 2](#_Toc512333406)3

[5.2.9主页 2](#_Toc512333407)3

[5.2.10 菜品浏览 24](#_Toc512333408)

[5.2.11 评价 2](#_Toc512333409)5

[5.2.12 购菜查看 25](#_Toc512333410)

[5.2.13 结账页面 2](#_Toc512333411)6

[5.2.14 座位选择 27](#_Toc512333412)

[5.2.15 配送详情 2](#_Toc512333413)8

[5.2.16 订单 2](#_Toc512333414)9

[5.2.17 查看订单 2](#_Toc512333415)9

[6 总结与展望 3](#_Toc512333396)1

[6.1总结 31](#_Toc512333397)

[6.2 展望 3](#_Toc512333399)2

[7 参考文献 3](#_Toc512333396)3

[8 致谢 3](#_Toc512333396)4

[9 附录 3](#_Toc512333396)5

# 绪论

* 1. 选题背景

在当今社会互联网的发展速度已经超过我们的想象，越来越多的科技正往高速发展的道路上前行。越来越多的厂商将产品由PC端转向移动端。美团、饿了么、百度外卖等移动点餐系统正改变着人们的传统生活习惯。并且传统店内点餐的方式正在由移动端点餐逐渐替代，移动外卖行业不仅加速了社会经济的发展，还提高了饮食行业的所产生的经济价值。

虽然外卖有很多种类，但仍然有很多学生觉得食堂菜多且健康。但每逢饭点时候，学生在食堂进行点餐时，看到各种新菜品都会纠结很久，不清楚自己要购买什么菜品，因此会花费较长时间，造成食堂高峰饭点期拥挤，等待时间过长。

在这样的背景下，又鉴于通过移动端点餐的人们是一个基数特别庞大的群体，食堂在每日饭点时间都异常拥挤。但大部分食堂在用餐高峰期并不足以同时处理较多外卖订单，因此若有一种方式可以解决高校食堂在饭点时候拥挤的现象同时又能拓展食堂的业务，必定会为各大高校带来庞大的经济效益，加速社会发展。

* 1. 系统目标

本系统旨在设计与实现一款高校食堂移动订餐系统。该系统包含用户Android端、服务器端，通过该系统，用户可以登录查看食堂的地址及信息，也可以进行食堂的菜品、价格的查看，同时该app的特点在于下单配送选择为3个：店内就餐、配送上门、上门自取，该目的为了让用户有多种选择，也降低了食堂的拥挤率。并且该系统还支持用户性别、用户昵称用户送货地址的更改。

* 1. 运行环境要求

用户端要求Android移动设备，且开启开发者选项-允许USB调试

服务器端运行于node.js+MySQL环境。

* 1. 论文具体结构

本论文正文总分为六章。①绪论，主要说明开发该食堂外卖系统的背景、系统的目标、系统对运行环境的要求。②相关技术知识，主要介绍Android、Node.js等相关技术情况。③系统分析，主要从可行性和需求方面对系统进行了阐述。④概要设计，阐释系统的各个功能模块，解释各个功能设计的目的与实现效果。⑤系统设计与实现，解释各个数据库表设计的字段与结构及app实现的操作步骤。⑥总结与展望，对本文做出简单的总结与思考。

# 相关技术知识

* 1. 工具知识
     1. Android Studio

Android Studio是一款开发工具，主要针对Android，具有以下优点：在响应速度、启动速度、内存占用方面表现得都非常优秀；该开发工具使用的界面简洁，让开发人员使用起来易懂；提示功能和补全功能对开发者来说都有非常重要的意义，在这方面，Android Studio表现得比同类开发软件更为智能，也能够智能保存，这让it人员开发效率提高很多；同时UI编辑器让安卓开发工具变得更加高科技，该编辑器不仅有很多自带的实时查看预览功能，给开发人员省下很多时间，而且该编辑器内置终端也是它的特色，很多it人员都习惯于command-命令提示符操作，但终端与开发界面的重复切换较为繁琐。这恰好解决了这个问题；同时，它还拥有完整的插件系统（Git、Markdown、Gradle等），通过搜索可下载需要的插件，整合了多个版本控制系统（GitHub、Git、SVN等），整合了Gradle构建工具。

* + 1. Visual Studio Code

Visual studio Code是一个自带GUI的代码编辑器，可在很多操作系统上运行：Mac OS X、Windows和Linux，能够针对于现代Web级云应用完成简单的代码编辑功能，是一个它拥有典型VS特色的界面，支持‘亮’和‘黑’两种主题。它具有现代编辑器应该具备的各种特性如可定制的热键绑定、代码片段收集、括号匹配、语法高亮。

* 1. 技术内容
     1. Android

Android系统的架构是分层的，从下到上分别是Android运行时和Linux内核、系统库、应用框架层、应用程序层，共包括4层。它的层次特别显明、架构十分清晰，协同一起工作。

Linux的内核对整个Android最核心的系统来说有着很强的重要性，在网络协议栈、内存管理、进程管理、驱动管理、安全性等方面Android核心系统服务依赖于它。Linux内核也是作为硬件与软件栈的抽象层。驱动：flash内存驱动、WiFi驱动、Audio驱动、Binder（IPC）驱动、键盘驱动、显示驱动、摄像头驱动、电源管理等等。

系统库包含了WebKit、SSL、SGL、libc、SQLite、FreeType、OpenGLEState、媒体库、图层管理这九个子系统。Android运行时包含了Dalvik虚拟机和核心库。Dalvik的主要功能是实现对线程的管理、堆栈管理、安全和异常的管理、生命周期的管理和垃圾回收，它是一种以寄存器为基础的Java虚拟机。

应用程序框架层是与程序员联系最为紧凑，对安卓开发来说，应用程序框架层就是基础，它包括了10部分，分别是通知管理器、资源管理器、活动管理器、位置管理器、包管理器、电话管理器、窗口管理器、内容提供者、视图系统、XMLPP服务。在Android平台上，核心应用程序所使用的API框架可以被开发人员完全访问到。而且，Android系统具有一个优秀的重用机制，无论哪个应用程序都可以发布其自身的一些能模块，其他的应用程序也可以使用这些已经发布了的功能模块。以这样一个重用机制为基础，对用户替换平台本身的各个应用程序组件来说就具有方便性。

对于开发高校食堂订餐系统app的我来说，在这个论文中主要从事的是Android app设计，主要的思考是Android的框架编写应用及程序的编写应用。

* + 1. Node.js

Node.js是一个运行环境，它针对于Javascript，是创建的操作平台，基于Chrome JavaScript运行，能够较简单地创造相应高速度、易扩展的应用网络。它的高效，十分适合在分布式设备上运行数据十分密集型的应用。原因是它使用的是事件驱动，非阻塞的I／O模型。Node采用的架构称为“事件循环”，因为采用了这个架构，编辑可扩展性高的服务器变得容易而且安全。Node能够编写出可扩展性高速服务器归结于利用了JavaScript是一个事件驱动语言这一特点。Node有一个十分重要的特性，即它选择了一种不但能增强性能还能降低开发复杂度的架构，而且它避开了许多并发编程会遇到的问题，展现出的性能十分良好。Node.js是一个流行的前端框架，还具有很多优点，如Restful API、单线程、非阻塞IO、V8虚拟机、事件驱动。

Node.js在用之前需进行安装，与Node.js一起安装的包及相关的工具还有npm，它是整个Node.js社区支持第三方模块最多、使用最多的包管理器。它可以解决Node.js在代码配置上的许多问题，npm允许用户从npm服务器下载别人编写的第三方包到本地使用，别人编制的命令程序也可以被用户也可以从npm服务器上下载并安装到本地使用，同样用户也可以将自己编写的包或者命令行程序上传到npn服务器上供别人使用。

* + 1. MySQL

MySQL是一个现在it人员喜欢用的关系型数据库管理系统，其优点是：速度迅速、占容量小、开源，属于Oracle公司。所谓关系型数据库就是定义了存储信息的结构，将数据库保存在不同的表中，能够通过索引的方式快速准确地定位到目标内容，从而避免了遍历整个库寻找数据，具有较高的灵活性，消耗的资源较少，速度较快。MySQL之所以能够成为最流行的数据库之一是因为它的使用非常简单，门槛低，可以说任何一个程序员都能够通过阅读使用手册和教程顺利安装和使用它；MySQL也是开源的，供广大用户免费使用，技术爱好者也可以相互探讨交流学习；具有复制功能，能够把一个数据库从实例A准确地复制到实例B。

在用该数据库开发之前，必须配置好自己的环境。不同的开发系统用到的数据库不一样，但mysql较为合适。其次，可以运用一些查看数据库的软件查看数据库的内容。我采用的是navict for mysql。具体操作的话可以参考百度上的详情。

# 3 系统分析

3.1 可行性研究

可行性研究是在系统内容提出后，对系统在经济、技术和操作等方面能否可行所进行的综合分析与评价，它决定了系统是否可行。而在进行可行性分析之前必须确认好自己要开发的系统，不然重复研究多个系统的可行性将会对开发造成阻碍。

就本毕设而言，为了确定食堂订餐系统是否符合实际需求，本设计主要四个方面进行研究：操作是否可行、经济是否可行、技术是否可行及其他方面。下面就这四个方面进行简单的介绍。

3.1.1 经济可行性

经济可行性分析，是对该系统开发的期间所投入的资金与系统进入使用后期所带来的经济效益进行比较，确定该系统是否符合经济的收益，是否可以持续使用下去。

从绪论当中可以看到，本系统的设计主要是为了科院二村服务的，它的开发投入小，只需要计算机和安卓手机就可以完成该系统的开发，因为整个APP的规模不是很大，所以不需要投入很多的人力与物力。另外，在具有一定食堂用户的基础之上，就可以向那些需要为食堂打广告的商家收取一定的费用，比如说在app真正开发出来之后，可以在APP相关页面上添加别家食堂的广告，作为该系统的盈利模式。因此，本系统在经济方面是可行的。

3.1.2 技术可行性

本课题是采用C/S模式的系统，它所使用的工具是AndroidStudio3.0和VSCode和Mysql数据库，这些都可以在网上免费下载安装后就可以使用。具体安装教程可以参考百度。C/S模式的系统功能完成的核心部分主要在服务器上，换句话说只要将服务器管理好即可管理系统的基本功能。至于为什么用C/S呢，主要是因为它将充分发挥客户端PC的处理技能，并且一些常见的事件[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)先处理再转接给服务器。其对应的优点就是[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)响应速度快。

后台则使用nodejs+mysql完成开发，考虑到每一所高校食堂的并发达不到十万级以上，因此选择该轻量级的服务器。

因此该系统在技术上也是可行的。

3.1.3 操作可行性

本尽量采用简洁美观的界面和尽可能简单的操作来提高用户使用的舒适性。用户可以很方便的查看食堂的信息，直观的看到食堂的电话及地址。并且点外卖操作不需很复杂，只需完成最简单的查看菜品、选择菜品以及下单即可。同时该app还要确定用户是否可以随时修改昵称和收货地址。

3.1.4 其他方面

在其他方面，例如二村食堂和一村食堂利益是否冲突，是否符合学校食堂的发展等方面，或者这个app的开发是否能够有效地解决食堂饭点拥挤的问题等等。经过相关资料的查询和相关人员的咨询后发现都是可行的。

经过上面几方面的可行性分析，发现本系统具有很高的可行性，因此，该系统的开发是可行的，有必要的。

3.2 需求分析

需求分析指的是存在一个系统，或是不存在这个系统，在创建它的时候，你需要确定你的目的、范围以及想实现的功能时所要做的全部的工作。而这需求分析是系统在开发前必须要做的提前工作。一个好的系统分析、用户需求编辑为系统开发打下一个好的基础，一个好的app开发必然会浪费很多东西，比如说：开发的人员、使用的物件、需要的时间以及在开发中可能产生的问题，而如何降低这些问题的存在会对系统开发带来很大的影响。因此，对本系统在功能、性能以及界面上进行需求分析。

3.2.1 功能需求分析

本次设计开发的移动订餐系统的分为前台和服务器。前台浏览页面主要是用户注册，登录系统，浏览商家信息，浏览商品信息，购买商品，以及修改用户昵称与密码，查看订单等功能；服务器主要为数据之间的传输。通过以上的简单分析，可以知道本系统主要为用户所设计。

1、用户对系统的需求

用户对app的需求主要是登录注册，浏览商家信息，浏览商品信息，浏览评论信息，下单功能，查询订单信息，管理个人信息，管理地址信息等。

根据用户实际的操作流程，可以分析得到以下的具体功能。主要包括：

（1）用户注册。输入自己的手机号码及密码。数据写入数据库的用户信息表中，如果通过，则在数据库种增加用户信息。

（2）用户登录：已注册成功的用户可以在app显示的登录的界面输入正确的手机号及密码，点击登录按钮。系统则会将输入的用户名及密码和数据库中对应的数据做对比。信息匹配的的则成功登陆，默认显示个人中心界面。错误则提示相应信息，重新返回登录页面。

（3）用户浏览商家信息，可以在主页查看食堂的相关信息.。

（4）用户查看商品的概况，比如菜价格，菜照片，菜的月售、获赞数、简介等。

（5）用户选择菜品，进行下单支付操作。

（6）用户进入个人中心，管理地址信息

（7）用户进入个人中心，管理昵称。

（8）用户进入个人中心，管理性别。

（9）用户进入订单界面，查询相关订单。

3.2.2 性能需求分析

性能需求有很多方面，一般来说可以分为几个方面：各个界面功能传递消息顺畅，各种协议之间的分析得到正确的保障，向用户展示的界面看着要友善，处理事件的效率高效

可以从以下4方面来考虑：

（1）运行可靠：该app的运行速度是否顺畅，是否会出现app相关页面显示不及时的情况，该app的运行是否会影响其他app的运行且造成阻碍。

（2）信息安全：是否会泄露用户的信息，如手机号、收货地址。

（3）操作方便：app是否让用户看着操作简便，用户是否能够简洁的看到外卖界面，以及点外卖的时候操作是否方便，复杂的下单方式则不适合。同时用户在对历史订单查询、个人信息修改的时候是否轻松方便。

（4）反馈及时：用户在注册时能否及时的将注册信息通过服务器记录到数据库中，用户在点外卖的时候菜品信息能否及时显示以及用户点了什么菜品能否及时的显示在界面上。

而这些都是进行下一步系统设计时所需要考虑的性能方面的内容。

据阐述系统开发过程所用到的关键技术点，以及关键代码的说明。第六

# 4 概要设计

**4.1 流程图**

登录成功

我的

注册失败

注册成功

开始

个人中心

登录

判断

商家信息浏览

进入外卖界面

商品信息浏览

挑选商品

下达订单

注册

用户注销

结束

订单详情浏览

个人昵称修改

地址信息修改

个人性别修改

注销登录

登录失败

订单

历史订单记录

历史订单详情

首页

图4.1 系统流程图

**4.2 E-R图**

用户

图4.2.1 用户实体属性图

介绍：首先用户手机号、密码这2个属性在用户注册的时候需要进行填写，设置这2个属性是为了每个用户都有自己的账号可以登录进行点外卖。这性别、昵称属性在后期用户个人中心修改的时候可以进行设置。用户ID会自动生成，并且与地址实体属性表、用户订单实体属性表中的用户id绑定且为同一个用户ID。

商品

图4.2.2 商品实体属性图

介绍：首先该外卖app只限一家二村食堂，所以商家id为1，但后期不免可以增入其他的食堂，所以商家id为了区分不同的食堂。并且不同的菜有不同的商品id，只要知道商品id就知道每一个的具体信息，该id自增且不重复。商品名称、商品价格、图片地址、已售份数、商品评分、商品介绍都会在用户外卖界面-商品详情可以看到，为了让用户更方便看到菜品的具体情况。另外商品种类能够让这些商品得到分类统一，设置该属性能够方便用户更方便了解自己的需求大类，从而在种类里找自己喜欢的菜。该实体属性图中的商家id与商家实体属性中的商家id一致，即该商品为商家实体属性中的商品。该id相同就不会出现点A家商店出现B家商店的菜品了。

商家

图4.2.3 商家实体属性图

介绍：该商家实体属性在外卖界面得以体现。该商家id与商品信息实体属性中的商家id一致，而在外卖界面用户能看到商家的名字和图片。同时在下方菜品上端会显示该商家的优惠活动：满（价格）减（优惠价格）。所以这些属性是需要的。

地址

图4.2.4 用户地址实体属性图

介绍：首先该用户id与用户实体属性标志的id一致，通过查看该属性图中的用户id就能知道哪个登录用户修改了收货地址。而地址id为自增，只有不同的用户的收货地址新增才会增加。同样收货的姓名、地址、电话为用户点外卖配送上门的具体信息。同样该实体属性图中的用户id也绑定了订单实体属性图中的用户id，两者相同。而该实体属性在用户修改收货地址的时候会发生数据的改变。

订单

图4.2.5 购买订单实体属性图

介绍：订单id为自增，每一笔订单都会有一个id，而这些订单对应的用户id可以是相同的也可以是不同的。订单详情里有以下信息：购买的菜品种类、菜品图片、菜品名字、菜品价格、菜品月售、菜品获赞数，这些都是用户在购买的时候所能看到的信息。付款方式的话则是说明用户是用支付宝还是微信支付的。购买总价计算了用户所购买的菜品的所有价钱。订单进度分为商家接单、订单配送、订单完成，这在订单实体属性中也是必须的。配送方式则能看出用户是店内就餐还是上门自取或是配送上门的。座位信息显示了用户在店内就餐选择的位置详情。该地址id与地址实体属性图中的地址id一致，由该地址id能够绑定到地址实体属性图，从而能绑定到用户id，进而能看到是哪个具体用户下了这个订单。

**4.3 功能图**

食堂移动订餐系统

登陆状态

首页

订单

个人

商家信息

选择外卖

购买记录

订单详情

饮食搭配

食堂新闻

设置性别

设置昵称

设置地址

菜品浏览

下单结账

支付方式

配送方式

确认下单

配送详情

图4.3 系统功能模块图

4-3.1首页

首页分为两块内容，一块为商家信息显示，一块为“去点外卖”按钮

4-3.1商家信息

在界面上显示食堂的相关信息：图片、电话、地址及公告。用户在打开该app时，即可看到商家的相关信息。这让用户很直接的可以看到该食堂的信息。

4-3.2选择外卖

用户在主页的界面下方上能够看到“去点外卖”按钮。用户点击该按钮，界面就会跳到食堂菜品详情界面。

**4-3.2.1菜品浏览**

用户在进入食堂外卖界面，可以按照菜品分类进行菜品的查看。在外卖界面上可以看到菜品的具体信息：图片、价格、月售量、获赞数等。用户也可将看中的菜加入购物车。

**4-3.2.2下单结账**

用户对意向购买的菜进行添加选择，界面跳转到支付页面。

**4-3.2.2.1支付方式**

用户在进入商品结账界面，可以看到自己加购的菜的数量价格是否正确。可以选择支付的方式：微信支付或者支付宝支付

**4-3.2.2.2配送方式**

用户可以选择商品配送的方式：上门自取、配送上门，或者选择店内就餐，选择店内就餐则会进入位置选择界面。可以进行位置的选择。

**4-3.2.2.3确认下单**

用户在核对好自己的收货地址、购买的商品、支付方式、选择好是店内就餐或是配送上门或上门自取后，点击确认支付按钮，界面会跳转到配送界面

**4-3.2.2.3.1配送详情**

用户在确认自己的购买菜后，确认下单，可以看到菜品配送的详细概括：商家是否接单、订单是否配送中、订单是否已完成等步骤。同时用户还能在该页面看到自己下单的菜品及商家设置的订单说明

4-3.2订单

用户登录app后，在界面的中间有一个订单的按钮可以点击。未登录的用户查看到的订单的界面是没有任何数据的的，登录的用户可以点击该按钮，然后相应的界面为历史订单，用户可以就其中的一份历史订单进行点击查看

4-3.2.1购买记录

在界面上显示用户以前购买的菜品信息：图片、名字、份数及价格。

4-3.2.2订单详情

用户点击订单详情按钮，可以查看以前购买的订单配送详情

4-3.3个人

4-3.3.1登陆／注销／注册

用户可以通过输入正确的账号和密码登录系统。点击退出登陆按钮能退出当前帐号。通过注册可以获得系统的用户权限，用户需按照注册流程进行注册，当输入的数据格式与系统规定的数据格式不一致时，不能成功注册用户，系统会给出信息提示。

4-3.3.2饮食搭配

显示食堂的每日就餐搭配，暂未开发该功能

4-3.3.3食堂新闻

显示食堂的最新新闻，暂未开发该功能

4-3.3.4设置性别

设置用户的性别，男／女。

4-3.3.5设置昵称

设置用户的昵称。

4-3.3.6设置地址

设置用户的收货地址。

5 系统设计与实现

5.1数据库设计

5.1.1数据表设计

本系统涉及到的数据信息表主要包括：用户信息表（ User），商品信息表（Product），商家信息表（Shop），地址信息表（Address）和订单信息表（Food\_order）及字段具体如下：

（1）用户信息表User：记录注册用户的一些相关信息

表5-1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名 | 英文名 | 类型 | 是否能为空 | 备注 |
| 用户ID | id | int | 否 | 自增，主键 |
| 性别 | sex | tinyint | 否 | 无 |
| 手机号 | phone | varchar | 否 | 无 |
| 昵称 | name | varchar | 是 | 无 |
| 登录密码 | password | varchar | 否 | 无 |

（2）商品信息表 Product：记录食堂外卖菜品的具体信息

表5-2 商品信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名 | 英文名 | 类型 | 是否能为空 | 备注 |
| 商品ID | id | int | 否 | 自增，主键 |
| 商家ID | business\_id | int | 否 | 无 |
| 商品名称 | name | varchar | 否 | 无 |
| 价格 | price | int | 否 | 无 |
| 图片地址 | imgUrl | tinytext | 是 | 无 |
| 已售 | sold | int | 是 | 无 |
| 点赞数 | rate | int | 是 | 无 |
| 介绍 | content | tinytext | 是 | 无 |
| 商品类别 | category | varchar | 否 | 无 |

（3）商家表Shop：记录食堂的相关信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名 | 英文名 | 类型 | 是否能为空 | 备注 |
| 商家ID | id | int | 否 | 自增，主键 |
| 商家名字 | name | varchar | 否 | 无 |
| 店铺图片 | imgUrl | tinytext | 是 | 无 |
| 价格 | price | int | 是 | 无 |
| 满减价格 | saleprice | int | 是 | 无 |

表5-3 商品表

（4）地址表Address：记录用户的外卖地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名 | 英文名 | 类型 | 是否能为空 | 备注 |
| 地址id | id | int | 否 | 自增，主键 |
| 收货姓名 | name | varchar) | 否 | 无 |
| 收货地址 | address | tinytext | 否 | 无 |
| 收货电话 | phone | varchar | 否 | 无 |
| 用户id | uid | int | 否 | 无 |

表5-4 用户收货地址表

（5）商品二级类型表Food\_order：记录用户所点外卖的具体详情

表5-5 用户外卖详情表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名 | 英文名 | 类型 | 是否能为空 | 备注 |
| 订单id | id | int | 否 | 自增，主键 |
| 用户id | uid | int | 否 | 无 |
| 订单详情 | Pids | varchar | 否 | 无 |
| 付款方式 | payment | int | 否 | 无 |
| 购买价格 | price | int | 否 | 无 |
| 配送方式 | orderWay | int | 否 | 无 |
| 订单进度 | progress | int | 否 | 无 |
| 地址信息 | address\_id | int | 否 | 无 |
| 座位信息 | seat | varchar | 是 | 无 |

5.2 App操作

对于Android平台上高校食堂移动订餐系统的设计，前台界面均由Android原生开发，通过网络请求得到相关数据，并将数据填充到手机界面上。需要访问到本地数据库。在客户端中需要得到网络访问权限、文件读写权限、GPS权限。

首次安装该软件，需要先在系统的应用权限设置中，将本软件的所有申请权限设置为始终允许，不同的终端界面不同，否则在本软件进行操作的时候会弹出询问框，影响软件使用。设置界面如图5.2所示

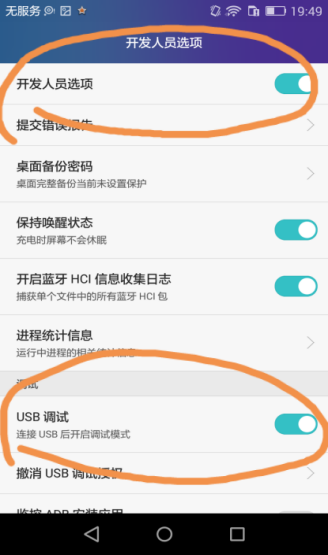


图5.2 手机权限设置图

客户端网络请求采用的是retrofit+rajava+gson的框架。首先客户端设置apiclient，将baseUrl定义好，数据解析器使用gson，数据流转换使用rajava：

private static final String BASE\_URL = "http://192.168.1.102:3000"

private ApiClient() {

retrofit = new Retrofit.Builder()

addConverterFactory(GsonConverterFactory.create()) //数据解析器

.addCallAdapterFactory(rajava2CallAdapterFactory.create()) //数据流转换

.baseUrl(BASE\_URL) //ip地址

.build();

}

然后在apiservice中定义请求接口例如：

@POST("/user/register") //注册接口

@FormUrlEncoded

Observable<Msg> register(

@Field("phone") String phone,

@Field("password") String password

);

@Post则是使用post方法请求服务器，@FormUrlEncoded则是设置requestbody的格式为表单，@Field("phone") String phone，@Field(“password") String password则是需要填写的两个参数，分别为手机与密码。当配置好参数之后，则可以在页面中进行网络请求

**5.2.1 加载图**

用户打开该外卖app见到的图，如图5.2.1。首先该窗口java文件调用onCrate函数创建一个窗口，再采用setContentView函数调取布局中的加载图的xml文件。该xml文件中就存在该张图。调用时即显示该张图。设置3s后自动跳到主页图。



图5.2.1 app初始图

**5.2.2 底部切换**

用户从加载图进入默认显示主页图5.2.2，一个Activity中包含3个Fragment，极大地方便用户进行切换tab，查看各种不同的界面。而在该页面可以看到下方有三个选择：主页、订单、个人中心。先生成该xml文件，并且该3个按钮均为RadioButton，先将3个按钮的id变为对应的再由对应的字符串，进而当点该按钮时，通过switch来判断是哪个按钮，从而显示对应的Fragment。



图5.2.2 默认显示图

②主要控件如下表5.2.2所示：

表5.2.2 主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| fl\_container | FrameLayout | 帧布局-按钮上方的总体框 |
| rg\_tab | RadioGroup | 单选组合框-3个单选按钮组合 |
| btn\_home | RadioButton | 主页单选按钮 |
| btn\_bill | RadioButton | 订单单选按钮 |
| btn\_mine | RadioButton | 个人中心单选按钮 |

**5.2.3 个人中心**

①用户点击：个人中心，显示5.2.3所示图。先从java文件中调取对应的xml文件。用sharedPreferences初始化数据判断用户是否登录，进而界面顶部和底部显示不一样的内容。mTvBtnNickName.setOnClickListener通过点击不同的按钮触发不同的事件来操作地址、性别、昵称修改。用if (isLogin)判断用户是否登录，初始化默认用户未登录。如果用户没有登录，在未登陆的情况下，对“我的地址”“设置性别”“设置昵称”进行修改，均不能进行修改。



图5.2.3 用户个人中心图

②主要控件如下表5.2.3所示：

表5.2.3 主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| img\_avatar | ImageView | 用户头像 |
| tv\_name | TextView | 顶部用户昵称显示 |
| tv\_btn\_address | TextView | 用户地址修改 |
| tv\_btn\_setting\_sex | TextView | 用户性别修改 |
| tv\_btn\_nick\_name | TextView | 用户昵称修改 |
| btn\_login\_out | Button | 登录按钮 |

**5.2.4 登录**

①用户登录图如5.2.4.首先加载该页面的xml文件，当用户输入手机号和密码的时候，Java会判断输入的手机号和密码长度是否正确。接口处理并从Userbean中调取对应的name和password判断用户登录的信息是否正确，正确则成功登陆，失败的话调用ToastUtils文件显示出失败的弹出框。



图5.2.4用户登录图

②主要控件如下表5.2.4所示：

表5.2.4 主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| edit\_mobile | EditText | 手机号输入 |
| edit\_password | EditText | 密码输入 |
| btn\_confirm | Button | 切换注册/切换登录 |
| tv\_change | TextView | 登录/注册 |

③测试数据如下表所示：

登录测试表

|  |  |  |
| --- | --- | --- |
| phone | password | 是否登录成功 |
| 13336167807 | 1234567 | 是 |
| 13336167806 | Hss1233455 | 是 |
| 1333616780 | 123456 | 否 |
| 13336167807 | 12345 | 否 |

**5.2.5 注册**

①用户可在登录界面切换为注册界面如图5.2.5，只是隐藏了ll\_step和ll\_query\_password。通过按钮监听事件mBtnComfirm.setOnClickListener判断是什么界面，账号注册要求:账号为十一位数字，注册的密码为六位以上，注册时通过phone.length()和passWord.length()判注册是否达标。并且注册页面要求输入两次相同的密码，否则注册失败。如果注册成功则通过服务器与接口将数据保存至本地数据库User表中。



图5.2.5 用户注册图

②主要控件如下表5.2.5所示：

表5.2.5 主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| ll\_step | LinearLayout | 注册步骤 |
| edit\_mobile | EditText | 手机号输入 |
| edit\_password | EditText | 密码输入 |
| ll\_query\_password | LinearLayout | 密码确认 |

③测试数据如下表所示

注册测试表

|  |  |  |  |
| --- | --- | --- | --- |
| phone | psaaword | query\_password | 是否注册成功 |
| 15336896400 | Hss987654 | Hss987654 | 是 |
| 15336896400 | 12345 | 12345 | 否 |
| 1533689640 | Hss987654 | Hss987654 | 否 |
| 15336896400 | Hss987654 | Hss9876 | 否 |

**5.2.6 修改地址**

①用户可在个人中心界面点击“我的地址”，界面图如图5.2.6.首先会调用getSharedPreferences根据登陆的用户id去调用address表中的地址id，调用AddressBean中的地址信息显示到界面上，新注册用户地址为空。用户若要修改地址，则通过api接口新增地址信息，通过服务器将新的地址信息保存到本地address数据库中。同样修改性别和修改昵称功能类似，只是修改性别和昵称中调用的是User表中的sex、name属性。下方不在对这两功能做解释。



图5.2.6 用户地址修改图

②主要控件如下表5-2-6所示：

表5-2-6主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| et\_address | TextView | 地址信息 |
| et\_name | TextView | 姓名 |
| et\_phone | TextView | 联系方式 |

**5.2.7 修改性别**

用户可在个人中心界面点击“设置性别”，界面图如图5.2.7.默认性别为空，用户可以点击要选择的性别。即可保存。



图5.2.7 用户性别修改图

**5.2.8 修改昵称**

用户可在个人中心界面点击“设置昵称”，修改昵称。选择新的昵称后，点击保存。新的性别即被保存。在个人中心的最上端也能看到新修改的昵称。新用户昵称为空，且个人中心页面顶上不显示昵称

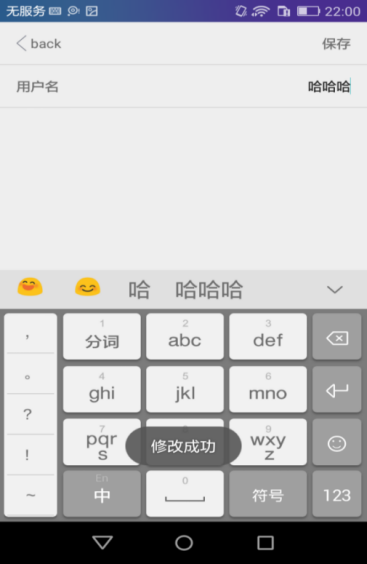


图5.2.8 用户昵称修改图

**5.2.9 主页**

①用户点击主页按钮，如图5.2.9。该块内容比较简单。首先看到食堂的图片、电话、食堂地址、上班时间及公告，即是xml文件中静态存在的数据。该界面就包含了一个按钮-去点外卖，当用户点击该按钮，通过getShopList（）函数调用到店铺的id，并且通过ShopListBean调取该店铺的具体信息。进入到外卖界面



图5.2.9 app主页图

②主要控件如下表5-2-9所示：

表5-2-9 主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| mTvBtnGo | TextView | 去点外卖按钮 |

**5.2.10 菜品浏览**

①用户进入菜品详情界面，如图5.2.10。首先该界面分为三块，第一块为食堂信息的展示，第二块为中间部分包括商品和评价，第三块为底部菜品浏览和结账。由3个avtivity组成。第一块由ShopDetailActivity构成，商家信息由前一个界面传输过来的信息所决定，并且设置了中间那块由ShopDetailFragment所决定。第三块通过OrderWindow监听如果有加购得菜则可以结账。并且点击底部查看购买的菜品。mTvSendPrice.setOnClickListener结账按钮监听事件跳转到结账界面。



图5.2.10 菜品详情浏览图

②主要控件如下表5-2-10所示：

表5-2-10 主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| container | FrameLayout | 开辟空块，放菜品浏览和评价 |
| ll\_fl\_orders | FrameLayout | 底部结账界面 |
| tv\_send\_price | TextView | 立即结账按钮 |

**5.2.11 评价**

用户点击评价界面，即可查看该食堂的商家总体评分、质量、速度及购买用户对配送的满意度。该界面调用到CommentBean中的一些属性，在CommentFragment中设置该些属性的值，进而在界面上显示。



图5.2.11 菜品评级图

**5.2.12 购菜查看**

①用户点击菜品详情左下角显示如图5.2.12，首先通过调用ProductBean，让加购得菜品显示在弹窗中，弹窗的弹出大小、透明度都在OrderWindow文件中功能设置好。如果用户要对加购得菜品进行增加或减少会调用到OrderListAdapter适配器。进行数量的增减。



图5.2.12 用户已选菜品浏览图

②主要控件如下表5-2-12所示：

表5-2-12主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| tv\_btn\_clear\_all | TextView | 清空购物车 |
| recy\_order | RecyclerView | 已加购得菜品 |
| btn\_remove | Button | 菜品数量-1 |
| btn\_add | Button | 菜品数量+1 |

**5.2.13 结账页面**

①结账界面如图5.2.13，首先该界面通过AddressBean判断登录用户是否填写地址，未填写则先去个人中心填写，填写过的自动填充。其次调用OrderCommitAdapter显示已经加购菜品的名字、份数、价格。并且在支付、配送方式mPay.setOnCheckedChangeListener单选框按钮选择。特别的点在于点击店内就餐按钮会弹出座位选择图。该功能的实现如5.2.14.



图5.2.13 订单详情确认图

②主要控件如下表5-2-13所示：

表5-2-13主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| tv\_address | TextView | 收货人地址 |
| tv\_name | TextView | 收货人名字 |
| recy\_orders | RecyclerView | 已选中的菜品 |
| tv\_btn\_commit\_order | TextView | 确认支付 |

**5.2.14 座位选择**

①用户选择“店内就餐”，5.2.14.首先该界面调用了SeatTable文件夹，该文件中有座位图的设计和布局。其次，用checked函数表明用户选择了哪些座位并且显示在下面。用户可在该页面选择相应的位置，最多选择5个。点击确认即可回到结账页面。当用户确定支付的时候，该座位字段会通过服务器自动保存到food\_order表的seat属性中。



图5.2.14 店内就餐座位选择图

②主要控件如下表5-2-14所示：

表5-2-14主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| seatView | SeatTable | 座位选择视图 |
| mTvConfirm | TextView | 座位确认 |
| tv\_seat | TextView | 选择了哪些座位 |

**5.2.15 配送详情**

①配送详情如图5.2.15。首先根据food\_order里的字段progress设置配送进度哪个显示为红。其次通过OrderBean.OrdersBean调取上个界面用户购买的菜品详情。具体界面传输均有BillAdapter与BillInfoAdapter适配器相关数据进行传输



图5.2.15 菜品配送详情图

②主要控件如下表5-2-15所示：

表5-2-15主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| recy | RecyclerView | 购买菜品详细清单 |

**5.2.16 订单**

①用户点击总界面下的订单按钮，能够查看到以往购买的记录如图5.2.16。该页面主要是通过接口调取Order的订单记录。显示在页面上。

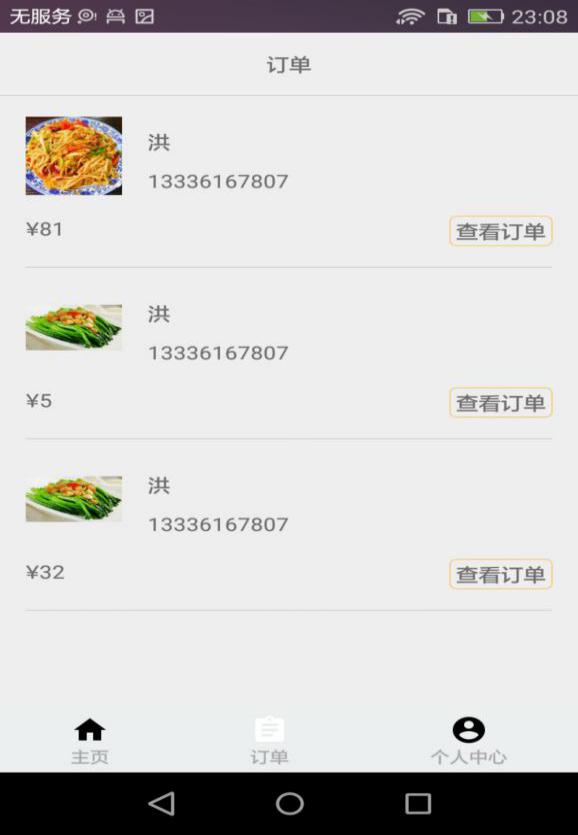


图5.2.16 历史订单图

②主要控件如下表5-2-16所示：

表5-2-16主界面控件表

|  |  |  |
| --- | --- | --- |
| 控件名称 | 空间类型 | 控件功能描述 |
| recy\_bill | RecyclerView | 购买菜品历史记录 |

**5.2.17 查看订单**

①用户点击单个购买记录详情的查看订单，页面会跳转到该订单的购买详细情况，功能类似5.2.15



图5.2.17 历史订单详情图

6 总结与展望

6.1总结

Android系统是当下十分流行的一个移动设备系统，外卖app也是当下社会十分热议的话题，越来越多的商家加入到了外卖的热潮之中，从而本课题就显得特别有意义。这个系统能够让学生吃到健康的食堂饭菜，有能够减轻食堂饭点拥挤的情况，整个系统能够更好地满足用户多元化的食堂外卖需求。

经过一个多学期的学习和实践，我终于完成了我的《基于Android的高校食堂移动订餐系统》的设计与制作。在制作过程中，我遇到了许许多多的难题，比如学习一门自己不会的安卓开发，以及第一次接触node.js和服务器这个东西。不过毕业设计的制作过程本身就是一个学习的过程，在遇到问题解决问题的过程中我对了Android系统有了一个更深入的了解，对Java程序开发也知道了更多，整个过程大大提升了我自身的开发能力。在信息日新月异的时代下，编程技术也飞快地发展，安卓技术的发展也在当今社会得以体现。而在开发这个app时，我更以自己是一个食堂点餐者，从一开始的界面到之中的点菜支付到最后的订单查看，都需要思考很多问题，虽然在开发学习中碰到了很多困难，比如：如何将Java语言运用到功能实现中，服务器和数据库及界面如何连接，一些看不懂的代码要如何运用到开发中等等，都是一些巨大的挑战，也正是这些七七八八的问题才让我对该系统的需求及如何开发进行了一定的调整，并且在某一次去食堂看到桌上放着的书来抢位子的这个充电让我想到了毕设中应该加入一个特别的元素，即加入店内就餐-位置选择功能，更加强了该软在食堂点餐使用过程中的巨大作用。

而本论文主要介绍了系统的选题背景、系统的设计目标、运行环境，说明了与Android开发相关的一些技术知识，着重介绍了该系统的数据库设计、流程功能图及相关的app操作介绍。通过一整个系统的制作，我对软件工程也有了一个更系统的理解，从问题的定义与分析到开发一个完整的系统，必须先从需求分析，一开始的需求分析如果没有分析好，会导致后面的开发不断地修改导致开发失败。其次是软件设计-一个好的数据库设计是一个系统开发是否成功的垫脚石，再到程序编码-编码的话就需要开发者不断的学习与钻研研究，如何运用自己学过的东西和百度上的新知识，将其实践到自己的系统中，最后到软件测试-测试功能是否都实现，是否成功运行成功。我学到了很多以前在课堂上从未接触到过的东西，提高了自己的研发能力和解决问题的能力，收获颇多。

6.2展望

本设计虽然初步简单的实现了食堂外卖app的最基本功能，但在一些具体体现上还存在着很多的不足之处。首先没有设计限定用户点外卖的权利，可以设定只针对于大学生使用的a食堂外卖app，或者说首页界面还不够复杂，我仅仅设计了最基本的首页界面，日后的开发中可以增加商家图片轮播，点外卖按钮应设计的更为高大。另外在食堂点餐界面也要更时尚化。订单界面要加强显示的信息量。同时个人中心要增加多个功能，比如修改密码等。如以后要使用，必须先纠正那些错误，并将某些功能进行完善。此外，用户的感受比较重要，比如说对界面的美观满意度上也存在着很多的不足之处，很多页面没有很好地进行设计和规划，在安全性上也有一些问题，用户直接查看数据库即可查看信息，这个需要日后多加修改。而在功能上完全也能够再进行深化，可以将系统再扩大，不仅仅局限于二村食堂，还可以是周边的商家的，希望做成一个属于高教园区范围内的订餐系统。

日后希望自己在就业的道路上能够多学习多尝试不同的编码语言，开发工具。多看看别人开发的系统，从中学习他们的好。将有效地运用到自己以后的开发中，不适合的并不需要学习，因为先掌握适合自己的比较有效果。

虽然这是一个虽毕业设计，但也花了近1年的时候去开发区编写。虽然时间与那些正式的系统相比花费的很少很少，但从另外一个方面说这是对自己能力的一种检验，虽然只是一个未应用的食堂外卖点餐系统，虽然没有美团、饿了吗那些外卖平台的壮大，但也是自己的一种成果。总之没有完美的app，只有尽力完成的app。希望自己在日后的开发中能够完善自己，尽可能开发出让用户满意度高的app。

7 参考文献

1. 段海霞.基于Android平台的软件开发关键技术的应用[J/OL].电子技术与软件工程,2017,(21):46
2. 王柯,马宏斌,王一圣.基于Android平台的软件开发若干关键技术研究[J].测 绘与空间地理信息,2014,3709:14-16+24
3. 单继周,马红.基于 Android 平台的软件开发若干关键技术分析[J].信息技术与信息化,2014(10):191-192.
4. 谭大海,刘兴勇.面向对象分析与设计[J].科技信息,2014,(10):160+165.
5. 张伟丽,江春华,魏劲超.MySQL复制技术的研究及应用[J].计算机科学,2012,39(S3):168-170.
6. 王金龙,宋斌,丁锐.Node.js:一种新的Web应用构建技术[J].现代电子技术,2015,38(06):70-73.
7. 仇晶,黄岩,柴瑜晗.基于Node.js中间层Web开发的研究与实现——以微信图书借阅平台为例[J].河北工业科技,2017,34(02):118-124.
8. 王平,贾化萍.C/S和B/S结合模式下应用系统的研究与开发[J].电脑与信息技术,2006,(01):50-53.
9. 杨斯博.基于安卓平台的手机订餐系统的设计与实现[D].吉林大学,2016.
10. 秦亚萍，基于LBS的移动订餐系统的设计与实现[D].华中师范大学，2014
11. 吴俊聪.基于移动终端网上订餐系统的设计与实现[D].电子科技大学,2013.
12. Acosta,Almeida.Android development and performance analysis[J].Journal of SuperComputing,2014,70:649-659.
13. Chaniotils,Kyriakou,Tselikas.Is Node.js a viable option for building modern web applications?A performance evaluation study.Springer-Verlag New York, Inc.,2015,97(10):1023-1044

8 致谢

从选题、开题到程序开发、论文编写，时间差不多过了一个学期多。期间也遇到了很多困难，修改了很多次。现在论文完善也进入到了尾声。从一开始的不知所措，不知道该做什么毕业设计，该用如何语言开发。到后来的重新选题。一个人的力量较为有限，但是一些人的力量却非常强大，在我制作毕业设计的过程中得到了多方的帮助，很幸运我能有工作负责关爱学生的老师们，热情善良的同学们，他们给了我许多帮助，使我的毕业设计制作过程更为顺利。

首先，向我的导师江左文老师表示最真挚的感谢，一开始选择江老师作为导师，在最开始并不知道选择什么题目，江老师让我慢慢考虑，给了我足够的时间思考毕设选题。江老师为人友善，专业知识十分厚实，同时有着严谨治学的态度待人宽厚谦逊，给人一种德高望重的感觉，江老师深深地影响着我。从选题到定稿，江老师都会十分耐心地引导我，帮助我，给我提出十分中肯的意见，江老师在大四app制作、论文编写上给了我莫大的鼓励。江老师的品质，学术水平都让我十分佩服与爱戴，感谢江老师的付出。

其次，要感谢我的好友们，室友在我自学安卓遇到困难的时候尽力大帮助我，在旁边耐心教我认识安卓界面，让我知道了如何开发安卓界面，如何用Java语言编写功能。并且感谢百度，让我初学了用node.js+mysql开发服务器。好友们一起学习讨论的过程总是充满趣味，当问题出现时，我的好友们能够热情友善地帮助我思考解决，向我提供了一些宝贵的资料，正是同学们的帮助，使得我的学习过程更为轻松快乐。

最后，我要衷心地感谢曾经基于我无私帮助的老师同学和家人，是他们成就了现在的我，接下来我将再接再厉，不负众望，继续前行。

9 附录

|  |  |
| --- | --- |
| 功能模块 | 添加商品与立即结账ShopDetailActivity |
| |  |  | | --- | --- | |  | /\*\*  \*将选购的商品添加到弹窗中  \*/  @Subscribe(threadMode = ThreadMode.MAIN) | |  | public void getProduct(ProductBean.Product.ProductDeatail productDeatail) { | |  | mOrderWindow.addData(productDeatail); | |  | }  /\*\*  \*如果未登录前往登录页面，如果登录了，则前往订单详情页  \*/ | |  | mTvSendPrice.setOnClickListener(new View.OnClickListener() { | | |  | @Override | | |  | public void onClick(View v) { | | |  | if (!SPUtil.getBoolean(ShopDetailActivity.this, "login")) { | | |  | startActivity(RegisterActivity.newIntent(ShopDetailActivity.this)); | | |  | return; | | |  | } | | |  | startActivity(OrderActivity.newIntent(ShopDetailActivity.this, mOrderWindow.getData())); | | |  | } | | |  | }); | | | |

|  |  |
| --- | --- |
| 功能模块 | 订单结账页-OrderActivity |
| 订单界面，自动同步地址到最顶端   |  | | --- | |  | |  | ApiClient.getInstance().getApiService().getAddress(uid) | |  | .subscribeOn(Schedulers.io()) | |  | .observeOn(AndroidSchedulers.mainThread()) | |  | .subscribe(new Observer<AddressBean>() { | |  | @Override | |  | public void onSubscribe(Disposable d) { | |  |  | |  | } | |  |  | |  | @Override | |  | public void onNext(AddressBean addressBean) { | |  | if (addressBean.result.size() == 0) { | |  | Toast.makeText(OrderActivity.this, "请在个人中心界面填写地址", Toast.LENGTH\_SHORT).show(); | |  | return; | |  | } | |  | mTvAddress.setText(addressBean.result.get(0).address); | |  | mTvName.setText(addressBean.result.get(0).name + " " + addressBean.result.get(0).phone); | |  | } | |  |  | |  | @Override | |  | public void onError(Throwable e) { | |  |  | |  | } | |  |  | |  | @Override | |  | public void onComplete() { | |  |  | |  | } | |  | }); | |  | } | | |