

东 莞 理 工 学 院

本 科 毕 业 设 计

毕业设计题目： 基于 iOS 的“i 莞工”客户端

学生姓名： 林添福

学 号： 201641402126

院 系： 计算机科学与技术学院

专业班级： 2016 级计算机科学与技术 1 班

指导教师姓名及职称： 潘晓衡 高级工程师

起止时间： 2019 年 12 月—— 2020 年 5 月

摘 要

近年来,随着我国移动互联网的不断进步与快速发展,高校办学规模的发展也日益壮大,但是校园信息传达的及时性和信息交流的互动性方面却存在着一定的局限性,校园内部的资讯和生活服务等资源也仍未能得到充分利用。为了解决上述现状以及满足全校师生对移动应用服务日渐高涨的需求,结合东莞理工学院目前尚未推出此类移动应用 iOS 客户端的实际情况,特此设计开发了基于 iOS 平台的“i 莞工”客户端。本客户端的设计模式是基于 MVC 架构,用 Objective-c 语言进行编写与测试。客户端主要包括莞工支付、视频会议、校园资讯、办事大厅、事物日程、消息通知、联络通讯和个人数据整合等八大模块。所有功能模块已通过测试,实际效果与预期基本一致,并成功上架苹果应用商店,在全校推广使用,受到广大师生的一致好评。

关键字 iOS、Objective-c、MVC、移动应用、支付

Abstract

In recent years, with the continuous progress and rapid development of China's mobile Internet, the development of university scale is growing, but there are certain limitations in the timeliness of campus information transmission and the interaction of information exchange, and the resources such as information and life services within the campus are still not fully utilized. In order to solve the above-mentioned situation and meet the growing demand of teachers and students for mobile application services, combined with the actual situation that Dongguan Institute of technology has not yet launched such mobile application clients, we hereby design and develop the "i-guangong" client based on iOS platform. The design mode of this client is based on MVC architecture, written and tested with Objective-C language. The client mainly includes eight modules, such as Dongguan payment, video conference, campus information, service hall, business schedule, message notification, contact communication and personal data integration. All the functional modules have passed the test, and the actual effect is basically the same as expected, and successfully put on the Apple App store, which is promoted and used in the whole school, and is highly praised by teachers and students.

Keywords iOS, Objective-C, MVC, mobile application, payment

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 研究现状	1
1.3 研究意义	2
1.4 研究内容	2
1.5 章节安排	2
第 2 章 相关技术和开发环境	4
2.1 相关技术简介	4
2.1.1 iOS 系统介绍	4
2.1.2 iOS 开发语言介绍	4
2.1.3 iOS 开发框架库介绍	5
2.1.4 iOS 设计模式介绍	5
2.1.5 iOS 数据库介绍	6
2.2 开发环境	6
2.2.1 iOS 运行环境介绍	6
2.2.2 iOS 开发工具介绍	7
第 3 章 iOS 客户端分析	9
3.1 可行性研究	9
3.1.1 经济可行性	9
3.1.2 技术可行性	9
3.1.3 运行可行性	9
3.2 客户端需求分析	10
3.2.1 功能需求分析	10
3.2.2 用例分析	11
3.2.3 业务流程分析	12
3.2.5 数据需求分析	13
第 4 章 概要设计	14
4.1 体系结构设计	14

4.2 功能模块概要设计.....	14
4.3 数据库概要设计	15
第 5 章 详细设计	16
5.1 功能模块详细设计.....	16
5.1.1 用户登录模块设计	16
5.1.2 校园卡付款功能模块	17
5.1.3 校园卡充值功能模块	18
5.1.4 校园卡转账功能模块	18
5.1.5 校园卡提现功能模块	19
5.1.6 校园卡管理功能模块	20
5.1.7 交易记录查询功能模块	21
5.1.8 个人信息编辑功能模块	22
5.1.9 消息通知功能模块	22
5.2 数据库详细设计	23
5.2.1 数据库表定义.....	23
第 6 章 客户端实现	24
6.1 客户端工程结构	24
6.1.1 配置客户端应用信息	24
6.1.2 配置客户端框架	24
6.1.3 配置客户端权限	25
6.2 用户登录模块	26
6.2.1 中央账号认证登录	26
6.2.2 微信授权登录.....	26
6.2.3 访客登录.....	27
6.3 校园卡付款模块	28
6.4 校园卡充值模块	29
6.5 校园卡转账模块	30
6.6 校园卡提现模块	31
6.7 校园卡管理模块	32

6.8 个人信息整理模块.....	33
6.9 消息通知模块	34
第 7 章 客户端测试	36
7.1 测试环境.....	36
7.2 测试方法	36
7.3 测试用例设计	36
7.3.1 用户登录测试用例	36
7.3.2 莞工支付测试用例	37
7.3.3 消息通知测试用例	37
7.4 结果和分析	38
总结与展望	39
参考文献	40
致谢	41

第 1 章 引言

1.1 研究背景

随着东莞理工学院办学规模的日益壮大，在建设高水平理工科大学的前进道路上不断发展。在信息传达方面，我们更多时候想要了解校园资讯动态就得通过登录学校官网进行查阅，学校部门发送通知文件时需要通过校园公众号进行推送或者通过院级部门的逐层转达；在业务办理方面，我们需要登录学校的办事大厅进行相应的业务操作，或者需要到学校的有关部门进行实地申请；在生活服务方面，我们进行校园消费，出入图书馆，校园充值等生活服务时，都得随身携带一张校园卡，这也难免出现校园卡时常丢失的意外。我们一般很少随身携带电脑或者熟记相应的业务办理网址，因此整个校园内部的资讯和生活服务等资源仍未能得到充分的利用，存在着一定的局限性；

而“i 莞工”客户端就能够很好的解决上述局限性，这是一款集成校园资讯、莞工支付、办事大厅、视频会议、事务日程、消息通知、联络通讯和个人数据整合等多功能于一体的手机客户端，我们通过 iOS 手机移动端，就能实时了解校园动态，随手在线办理校园业务，以及通过手机付款码进行校园卡消费，出入图书馆等等，极大的为全校师生生活服务提供了便利性。

“i 莞工”客户端通过在苹果应用商店下载安装到 iPhone 手机上即可运行。师生可通过中央认证账号和密码或者微信授权，在同意《i 莞工服务协议》之后，登录成功之后即可正常使用功能，为生活服务提供了极大的便利性，在师生们的使用过程中受到了一致好评，同时应用也在不断的增加实用的新功能，版本迭代迅速，打造成具有极致用户体验的东莞理工官方移动客户端。

1.2 研究现状

目前，随着我国移动互联网的高速普及以及信息技术的快速发展，高校内的大学生绝大多数都有属于自己的移动智能手机，智能手机的发展也推动着校园移动平台的建设与完善，并且各大高校开发自己校内使用的移动应用已经早已成为势在必行的一个大趋势，其中有原生移动应用和以 HTML5 技术混合开发应用两种，但是综合考虑，目前原生移动应用无论是在性能还是稳定性方面，原生移动应用的开发仍然占据优势^[1]。

根据调查显示，在国内的大多数高校来说，校园移动应用仍然处于起步阶段，但是随着互联网的发展推进，其具有非常大的发展潜力。在调查近300所高校后，我了解到有约29%的高校目前还未有此类校园移动应用开发发展的规划；有约32%的高校调查结果显示正在规划中，但是未实现；有约25%的高校调查结果显

示正在规划中并逐步实现；只要14%的高校已经开发好专属校园移动客户端并且投入使用。

我们可以知道国内很多高校都在努力完善移动应用客户端的研发工作，东莞理工学院也还未曾推出此类服务型客户端的使用，从东莞理工学院实际出发，积极响应东莞理工学院向高水平理工科大学建设的步伐迈进，在和导师的详细沟通之后，我决定设计和开发出一个面向全校师生的多功能于一体的移动客户端，希望能够满足全校师生对新型服务移动平台的需求。

1.3 研究意义

本次课题的研究意义是积极响应东莞理工学院向高水平理工科大学建设的步伐迈进。为全校师生提供一个便捷操作的移动应用平台。我结合大学4年学习的专业知识，具备了良好的计算机基础，在移动应用开发的 iOS 平台上积累了一定的项目经验之后，通过理论知识与实际项目相结合，开发了基于 iOS 的“i 莞工”客户端。本应用的开发，真正意义上的解决了校内资讯传达，校园生活服务资源的局限性，实现了东莞理工学院的信息化，智能化，完善了创新体系的卓越发展。

1.4 研究内容

本课题主要的研究内容是运用移动应用 iOS 开发平台，采用了 Objective-c 语言进行编写与测试，整个客户端的架构是 MVC (Model,View,Controller)，对客户端的代码进行分层，降低了项目的耦合性，实现了高内聚，低耦合，在后期新版本的迭代与新功能的增加，以及后期维护都有很大的优势^[1]。本客户端有莞工支付、视频会议、校园资讯、办事大厅、事物日程、消息通知、联络通讯和个人数据整合等八大模块，我研究的是用户登录模块、莞工支付模块、个人信息整合模块等三个大模块。

1.5 章节安排

本文章节安排如下：

第 1 章 绪论：主要介绍了“i 莞工”客户端的研究背景、通过调查数据阐明了研究现状，研究意义，并列出了研究内容与章节安排。

第 2 章 技术介绍：主要介绍 iOS 客户端开发的相关技术和开发环境。

第 3 章 客户端分析：主要进行可行性研究和客户端需求分析。

第 4 章 概要设计：先提出体系结构设计，在对功能模块和 UI 视图进行概要设计。

第 5 章 详细设计：对功能模块以及 UI 视图进行详细设计。

第 6 章 客户端实现：在客户端设计的基础上，描述客户端主要功能的实现过程。

第 7 章 客户端测试：介绍客户端的测试环境和部分测试用例，然后分析测试结果。

第 2 章 相关技术和开发环境

2.1 相关技术简介

2.1.1 iOS 系统介绍

iOS (iPhone OS), 是由苹果公司开发的移动操作系统, 这个系统最初是运行在苹果手机 iPhone 上面的。随着技术的不断发展, 现在也已经成功运行在 iPad, iPod touch 等设备上面。iOS 是一个生态环境十分优秀的移动系统平台, 有自己的专属应用商店—App Store, 上面有很多优秀的 iOS 应用。开发者需要先注册苹果开发者账号, 审核通过成为一名正式开发式者后, 才可以在应用商店上面上架自己开发的应用, 但是苹果公司出于安全考虑, 审核机制极为严格^[2]。用户通过登录自己的 Apple ID, 就能在应用商店直接下载应用到设备。iOS 系统的优点数不胜数, 首先 iOS 与苹果设备的整合度极高, 无论是在 iPhone 还是 iPad 上, 运行流畅性远超 Android; 其次是 iOS 系统的界面优化十分简约, 避免了复杂的界面跳转机制与 UI 布局, 给人一种直截了当的清晰感; iOS 系统的安全稳定性是其最大的优点之一, 我们每个人的数据都不希望遭到泄漏, 而 iOS 有着强大的防护能力来保证用户的信息安全^[2]。虽然 iOS 系统有些封闭, 但是生态里面却有高达 50 万个应用供用户选择下载使用。随着 iOS 系统每年的不断升级, 系统应用也一直在创新发展。iOS 是基于 UNIX 的, 系统架构主要分为四层, 分别是: Cocoa Touch (构建了 iOS 应用的基本系统服务), Media (提供了音视频, 图形的处理操作), Core Service, Core OS^[2]。总之, iOS 系统目前是世界上较为完善的移动系统平台之一。

2.1.2 iOS 开发语言介绍

iOS 开发语言主要包括 Objective-C 和 Swift, Objective-C 是基于 C 语言和 C++ 的进阶的面向对象的编程语言, 苹果公司最初推出的编程语言就是 Objective-C。Swift 是苹果公司在 2014 年 WWDC (苹果开发者大会) 推出的一门全新的编程语言, 它采用了安全的编程模式和现代语法使得编程更加简单^[3], 同时其语法简单, 代码简洁的可阅读性也更强, 但是其每年的语法变化过大, 业界还是偏向于使用 Objective-C。“i 莞工”的编写, 我选择的是 Objective-C 语言, 主要是因为 Objective-C 的运行效率特别高, 同时也是面向对象的语言, 方便后期的测试与维护, Objective-C 的底层是 C 语言和 C++, 有着强大的运行时机制, 我们可以解决很多底层方面的问题, 提升开发效率。

2.1.3 iOS 开发框架库介绍

一、Foundation 框架，是 iOS 其他框架中的基础，其包含了很多数据类型，例如结构体 Struct，枚举 Enum，类 Class 等等^[3]，我们可以通过简单的引入语句 `#import<Foundation/Foundation.h>` 导入头文件，常用的类都以 NS 作为前缀，NSString, NSArray, NSDictionary, NSData, NSNumber 等，在开发过程中使用的基本对象都是基于 Foundation 框架的。

二、UIKit 框架，提供了一系列的类来建立和管理 iOS 应用程序的用户 UI 界面的接口，事件传递，绘图模型，视图控制，控制触摸屏幕等^[4]，通过 `#import<UIKit/UIKit.h>` 导入头文件，引入头文件之后，就可以在任何地方声明 UIKit 里面的类，例如常用的 UIButton, UILabel, UIView, UIImageView。

三、AFNetworking 网络请求框架，是目前 iOS 开发最受欢迎的第三方网络通信库，对苹果官方的 NSURLConnection 和 NSURLSession 进行了封装，使用起来十分高效，整个框架库使用了工厂类的设计模式，通过 Block 接口进行回调，我们主要用来进行网络数据请求，例如用户登录和注册与后台数据交互，便捷的发送 Get 和 Post 请求，`#import<AFNetworking/AFNetworking.h>` 导入头文件就能直接使用，内部封装处理了多线程和内存管理，极大的提升了网络请求方面的编程效率^[5]。

四、SDWebimage 图片加载框架，因为其具有图片缓存机制，所以要进行网络请求加载图片的时候，SDWebimage 是我们的首选^[5]，因为它可以异步的加载图片，提升了程序的性能，通过巧妙的缓存机制，每次我们要进行网络请求加载图片的时候，会首先去 Cache 缓存里面查找，在逐步去硬盘文件查找，如果都没查找到的时候，才会出发网络请求^[6]。这种机制在性能优化方面做出了很大的贡献，我们可以通过 `#import<SDWebimage/SDWebimage.h>` 导入头文件使用。

五、SDAutolayout 是我们常用的 UI 布局库，内部通过链式编程的封装，实现 UI 的自动布局，通过设定边距离 Offset 和宽高，运用于视图 View 布局操作。通过 `#import<SDAutolayout/SDAutolayout.h>` 即可使用在相应的布局上。

六、YYmodel 是一种将数据转模型的常用框架库，当从后台获得 Json 数据时，不用对每个字段进行解析，直接转换为我们定义好的 Model 数据模型，是一件效率极高的操作，我们可以通过 `#import<YYmodel/YYmodel.h>` 引入使用。

2.1.4 iOS 设计模式介绍

MVC 设计模式，即模型 Model，视图 View，控制器 Controller，使用的时候是将 Model 和 View 进行分离，两者之间不能直接通信，要通过控制器才能进行数据通信。模型主要是封装了应用程序的数据模型，并定义该数据的属性以及操

作。视图主要是定义显示的 UI 界面，是用户可以看得到的对象，视图对象也可以响应用户的一些事件，其主要目的就是显示模型对象的数据。控制器类似于模型和视图之间的传输媒介，模型的数据更改可以通过视图显示，视图更改了数据也要保存到模型当中，此时都是通过控制器来进行数据操作的^[7]。MVC 因其具有低耦合的特性，将视图层和业务层分离，具备了良好的可维护性，受到广大开发人员的青睐。

单例设计模式，在某个对象在程序过程中，我们只需要一个对象，但是需要在不同的地方调用并获取属性资源，就会采用单例的设计模式，顾名思义，就是为了保证在程序的创建过程中只是创建一次，例如我们在数据库操作打开数据库进行相应的增删查改，或者调用接口进行网络请求的时候，都是通过 GCD 的语句 `Dispatch_Once` 来保证执行，方便的控制实例个数，节约系统资源并提升程序性能。

工厂设计模式，指的是专门定义一个类来负责创建其他类的实例，被创建的实例常常具有共同的父类，实际上就是一个工厂类，根据传入不同的参数，动态的决定出哪个类的产品实例。我们通常在工厂类负责创建的实例对象比较少的时候使用。

观察者设计模式，iOS 中常用的观察者模式是 KVO，指通过建立观察者与被观察者的关系，每当被观察者的属性发生变化的时候，观察者就能检测监控并且做出相应的操作，就是一个对象在状态变化的时候会通知另外一个对象，这是一种降低耦合度的设计模式。

2.1.5 iOS 数据库介绍

Sqlite 数据库诞生于 2000 年，是一个开源的嵌入型的关系型数据库，类似于我们常用的 Mysql，其特点主要是占用的内存资源非常小，通常只有几百 KB，能够支持多个系统，而且处理效率极高，我们在 iOS 开发中，经常用来存储轻量型的用户数据。Sqlite 常用的关键字有：Creat、Update、Delete、Select 等等，也包括数据定义语句，数据操作语句，数据查询语句。在使用 Sqlite 之前，要先导入 Libsqlite3.tbd 库，然后导入头文件^[7]。现在数据库操作也有常用的第三方库 FMDB，简单直白的创建数据库表的存储路径，并创建数据库，随后进行增删查改操作，而不用自己编写底层的 C 语言语句。

2.2 开发环境

2.2.1 iOS 运行环境介绍

基于 iOS 的“i 莞工”客户端的具体开发环境如下表所示：

表 2-1 客户端开发环境表

硬件配置	CPU: 2.2GHz 四核 Intel Core i7, 内存: 16G
操作系统	macOS Catalina 10.15.1
开发工具	Xcode,Git,Charles,Cocoapods
数据库	Sqlite3

2.2.2 iOS 开发工具介绍

一、Xcode，是非常高效的集成开发工具，使用的是 GCC 编译，iOS 开发者可以通过 Xcode 进行各种任务的开发，包括 Mac OS 和 iOS 的应用程序开发，具有统一的用户界面设计，编码，测试，调试。同时也内置了 Git，我们可以直接在 Xcode 里面进行 Git 版本的控制与维护。在调试的时候也能通过 DDLB 命令在控制台打印调试信息^[8]。在 UI 布局界面，Xcode 提供了 Interface Builder 界面设计器添加并设置约束，在设计的时候就可以更具可视化视图看到效果，在控制器跳转方面也便于查看。测试的时候我们可以利用 Xcode 的模拟器进行测试。

Xcode 界面图如图 2-1 所示：

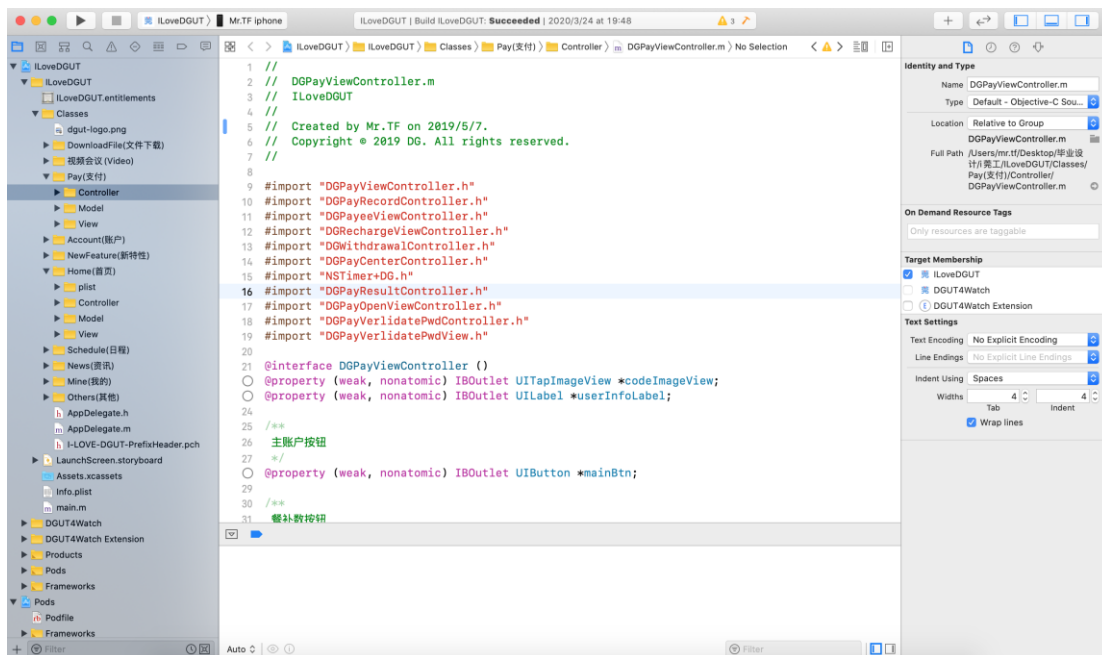


图 2-1 Xcode 界面图

二、Cocoapods，是 iOS 的项目的库的管理工具，通过 Cocoapods 我们可以优雅的扩展项目，便捷的导入第三方开源库。在使用的时候，我们要先在项目中安装 Cocoapods. 通过 Pod init 初始化项目，然后通过 Open Podfile 文件添加要引入的第三方库，最后通过 Pod install 下载安装。

Cocoapods 命令行注册如图 2-2 所示：

```
total 32
drwxr-xr-x  5 mr.tf  staff   160  9 28  2019 DGUT4Watch
drwxr-xr-x 13 mr.tf  staff  416 10  7  2019 DGUT4Watch Extension
drwxr-xr-x@ 10 mr.tf  staff  320  9 20  2019 ILoveDGUT
drwxr-xr-x@  6 mr.tf  staff  192  9 29  2019 ILoveDGUT.xcodeproj
drwxr-xr-x@  5 mr.tf  staff  160  5 30  2019 ILoveDGUT.xcworkspace
-rw-r--r--@  1 mr.tf  staff 1073  5  6  2019 LICENSE
-rw-r--r--@  1 mr.tf  staff  829  9 27  2019 Podfile
-rw-r--r--  1 mr.tf  staff 3779  9 27  2019 Podfile.lock
drwxr-xr-x@ 31 mr.tf  staff  992  3 21 14:59 Pods
-rw-r--r--@  1 mr.tf  staff   13  5  6  2019 README.md
192:i莞工 mr.tf$ pod init
```

图 2-2 Cocoapods 命令行注册图

三、Charles，是一款代理服务器，通过成为浏览器，电脑，模拟器，手机真机的代理，来截取请求达到抓包的目的，Charles 客户端上可以清楚明了的显示每一条请求信息，我们通过分析携带的参数和报文内容来测试我们的客户端请求数据^[8]。

Charles 界面图如图 2-3 所示

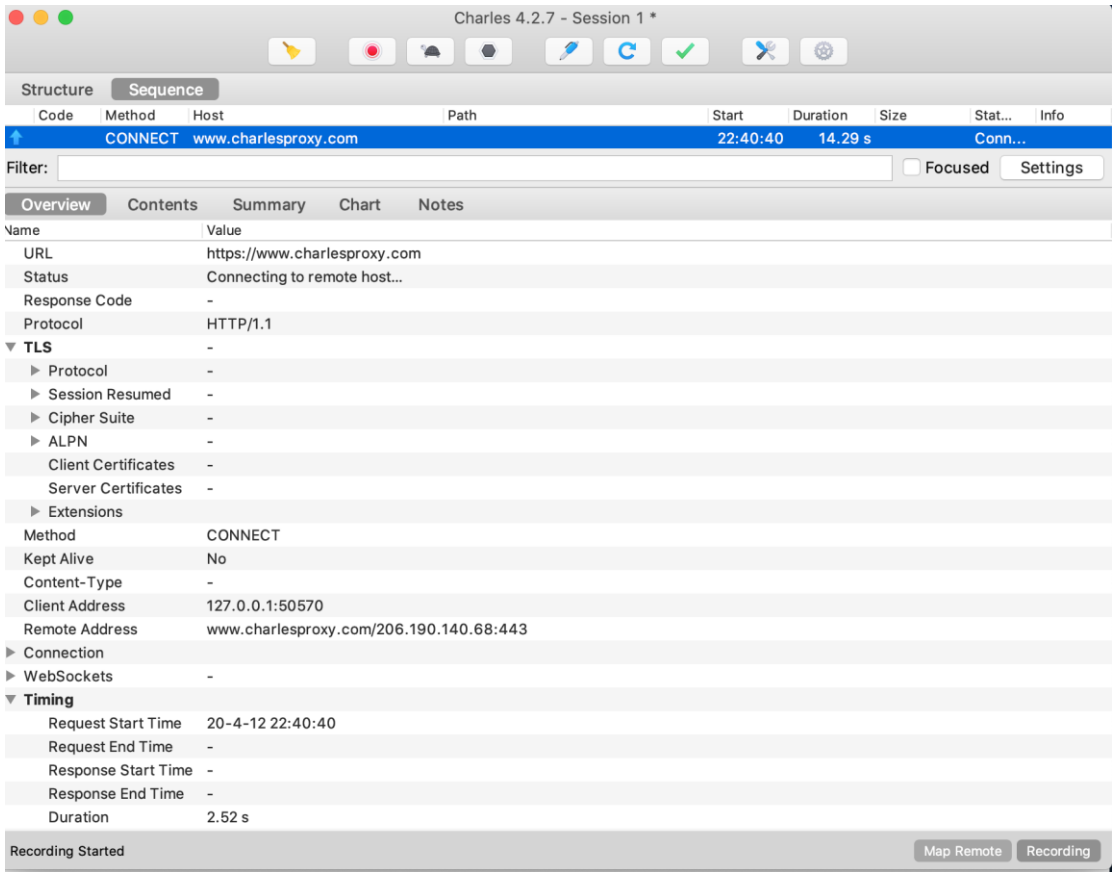


图 2-3 Charles 界面图

第 3 章 iOS 客户端需求分析

3.1 可行性研究

3.1.1 经济可行性

“i 莞工”客户端，用户对象是面向东莞理工学院的全体教职工和学生，在苹果应用商店 App Store 免费下载到 iPhone 移动端，是校园内部的免费应用，用户通过学校中央认证账户密码登录使用。莞工支付模块涉及到的校园卡充值以及金额提现至微信账户等功能涉及到金钱交易也不收取任何手续费，不但节约了时间成本，而且提升了充值帐户的时间效率。同时学校配备了配合莞工支付的扫码枪和扫描摄像头等硬件设配，已经部署在各个食堂，生活超市等消费点，用户可以直接用手机打开客户端出示付款码进行付款服务，整个操作流程清楚方便，在经济上的可行的。

3.1.2 技术可行性

“i 莞工”客户端使用的编程语言是 Objective-C, 采用的设计模式是 MVC，为了降低耦合性，方便后期功能的迭代与新功能的添加。Objective-C 语言的有着强大的运行时机制和高效的运行效率。利用其中的 Foundation 框架和 UIKit 框架可以将基本的数据模型以及简单的 UI 界面布局实现，并且 Xcode 里面配置了详细的 Objective-C 接口文档，可以迅速定位查阅，在 iOS 开发的学习路线上，网上也有很多开源项目与博客可以进行研究。遇到问题可以在 Stackflow 上面进行交流查阅，同时也有指导老师的辛勤指导，每次遇到问题都能够得到很好的解决，iOS 封装了很多高效的第三方库，使用起来较为轻便，最主要的是库都很好的解决了多线程与内存管理问题，极大的提升了我们的编程效率。在编写之后，Xcode 可以通过模拟器来测试我们的开发效果，内置的 Instrment 工具可以检测应用的运行状况，方便查找问题。因此在技术上是可行的。

3.1.3 运行可行性

“i 莞工”客户端只是需要通过苹果应用商店免费下载 iPhone 手机上运行，同时因为要涉及到数据交互，要打开蜂窝网络或者 Wifi 联网操作，同时也配置了人脸识别与指纹识别操作，功能界面优化性强，具有较为机制的用户体验，可适配多种 iPhone 机型，因此在运行上是可行的。

3.2 客户端需求分析

3.2.1 功能需求分析

本客户端有莞工支付、视频会议、校园资讯、办事大厅、事物日程、消息通知、联络通讯和个人数据整合等八大模块，我研究的是用户登录模块、莞工支付模块、个人信息整合模块等三个大模块。

第一，用户登录模块：本系统的主要使用对象为东莞理工学院的全校师生。为方便所有师生的便捷操作，可以通过经学校中央认证的账户密码进行登录，学生用户使用学号/密码，教师用户使用教职工号/密码，同意《i 莞工服务协议》之后即可点击登录。其他登录渠道包括微信登录，需要用户自己微信授权；访客登录，需要填写访客的手机号码，短信验证码验证通过后即可以访客的身份登录。

第二，校园卡付款模块：通过和学校后台数据交互之后，付款码根据支付类型（主账户，补助金，餐补数）以二维码的形式呈现，实时检测付款码的消费状况以及呈现二维码可在校园支付设备消费，二维码以每 60 秒自动刷新，点击二维码实现立即刷新。

第三，校园卡充值模块：通过微信，支付宝支付方式对电子校园卡进行充值操作，实时检测账户余额，充值金额可自行选择，10 元到 500 元不等，但是充值金额不可超过上限六千元，充值成功后回调充值信息。充值操作是通过唤醒本地应用支付宝或微信，若本地未安装，通过相应的网页版账户登录支付。

第四，校园卡转账模块：转账方式包括主账户和餐补数，选择之后生成相应的收款二维码，此二维码仅用于向我转账当面支付，对方扫码之后确认转账金额便转账成功，转账收款码点击立即刷新^[9]。

第五，校园卡提现模块：通过获得当前账户的总余额，输入需提现的金额后，申请提现，符合规定后即可提现到自己的微信，提现用户必须为微信实名用户，提现金额最少为 1 元，单笔限额 20000 元，通过显示当前余额，可以点击全部提现，提现成功后可通过消息推送以消息通知的形式告知。

第六，交易记录查询模块：以列表的形式呈现校内码交易和校外码交易流水信息，主要包括消费地点，消费类型与交易时间，交易类型，交易金额，也能查询通过网上充值中心莞工 APP 微信或者支付宝充值显示记录，下拉刷新加载更多交易信息，有线上，扫码，其他形式的消费方式。

第七，校园卡管理模块：管理校园卡的付款、充值、余额显示、向我转账、提现、交易记录。校园卡挂失，暂停使用的安全操作，需要通过输入支付密码进行验证。账户安全主要是重新设置支付密码，设置免密额度^[10]。

第八，个人信息管理模块：个人资料编辑，包括姓名、性别、班级、邮箱、微信号、QQ、短号、手机号、地址等等，这些个人资料信息与通讯录信息同步。

第九，消息通知模块：校园通知公告推送，校园卡管理信息，消费信息，充值信息，提现信息等推送，会议消息，系统消息。信息可以一键阅读和通过关键字搜索查询。

客户端功能需求分析图如图 3-1 所示：

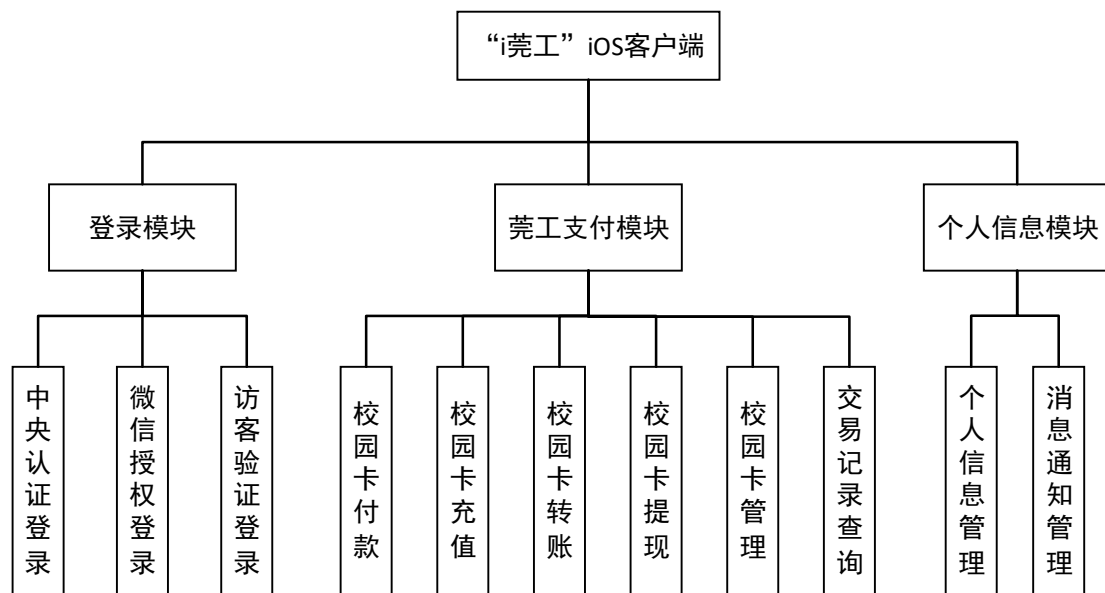


图 3-1 客户端功能需求分析图

3.2.2 用例分析

用例图是指由参与者、用例，以及它们之间的关系构成的用于描述系统功能的视图，用例图中常见的关系有包含关系和扩展关系。包含关系（include）是指几个子用例可以提取出他们的公共用例；扩展关系（extend）是指在原来用例的基础上增加新用例。本系统的使用对象为学生和教师，都可作为用户进行用例分析并画出用例图。

登录用例包含 3 个子用例，分别是：中央认证登录用例，微信授权登录用例，访客验证登录用例。莞工支付用例包含 6 个子用例，分别是：付款用例、充值用例、转账用例、提现用例、卡管理用例、交易记录查询用例。个人信息用例包含 2 个字用例，分别是：信息编辑用例、消息通知用例。

用户用例图如图 3-2 所示：

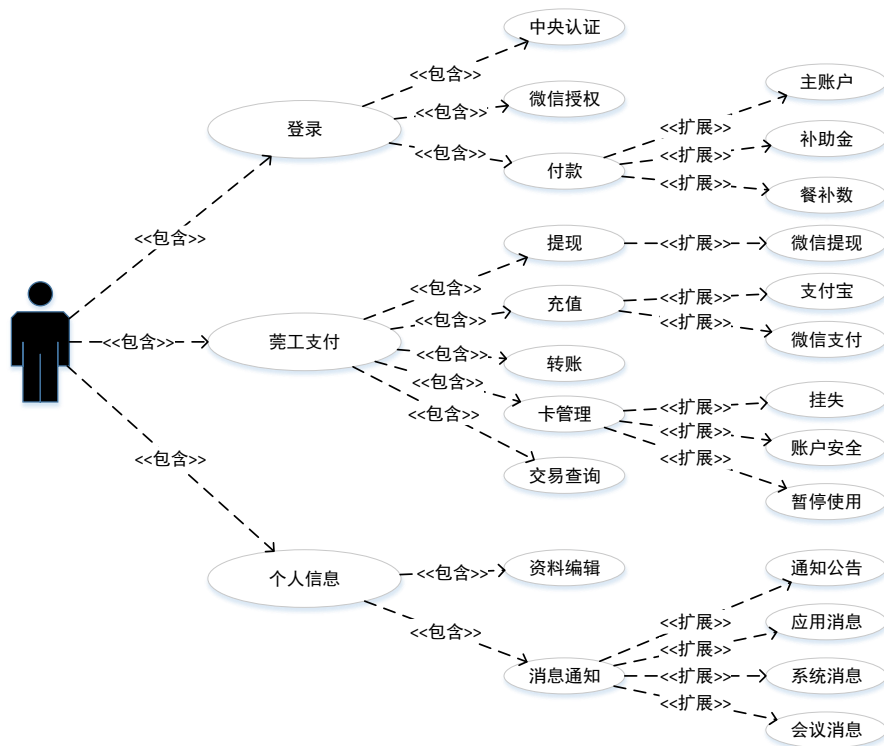


图 3-2 用户用例图

3.2.3 业务流程分析

用户通过中央认证账号密码、微信、访客其中一种方式登入成功进入客户端主页后，可以点击莞工支付标签，进入莞工支付页面，默认是校园卡付款码界面，付款码有主账户、补助金、餐补数供付款形式进行付款，点击切换即可。点击右上角的交易记录查询，进入交易记录查询模块，里面分校外码和校内码可供查询。点击充值进入充值模块，有金额选择，微信支付、支付宝支付两种支付方式。转账模块会以二维码的形式呈现，对方通过扫描即可，有主账户和餐补数两种类型可供选择，进入提前模块后可以根据自己的余额情况进行微信提现操作。校园卡管理模块有挂失，账户安全，暂停使用^[10]。点击我的标签，进入个人信息整理模块，里面有消息编辑和通知信息，通知信息里面可查看通知公告、应用消息、系统消息、会议消息。

用户的流程操作活动图如 3-3 所示：

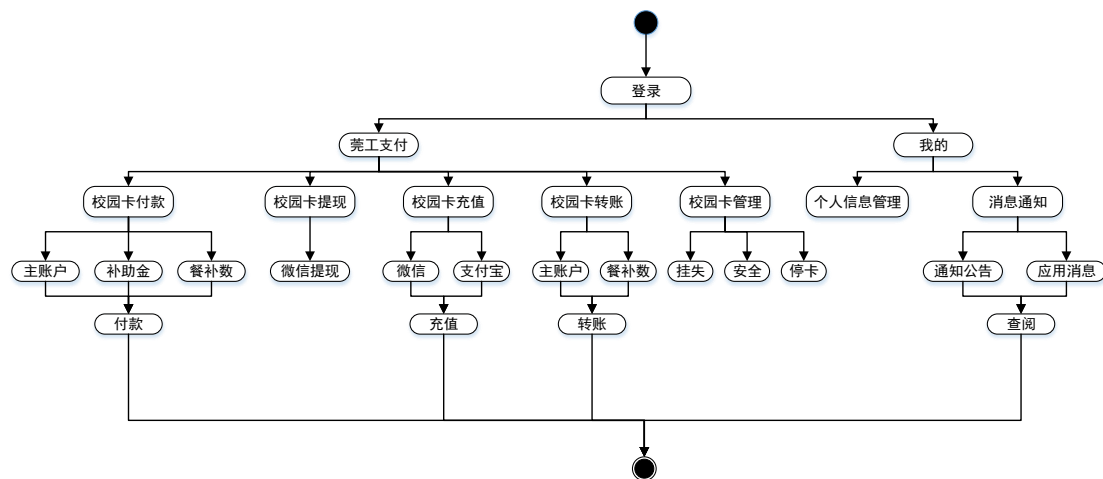


图 3-3 用户流程操作活动图

3.2.5 数据需求分析

1、实体集分析

同一类型实体的集合为实体集。本客户端的基本的实体主要是用户（DGUser）。

2、实体数据项分析

用户实体：用户 ID，令牌，姓名，学号/工号，密码，用户状态。

第 4 章 概要设计

4.1 体系结构设计

“i 莞工”移动客户端使用的开发语言是 Objective-C 和 C,开发工具是 Xcode,开发设备是 MacBook Pro,整个移动应用客户端是基于 iOS 的 CocoaTouch 层,架构模式是 MVC,模型-视图-控制器。

本客户端采用 B/S 模式,这种模式统一了客户端只是通过发送 HTTP 请求以及及时响应服务端返回的 Json 数据并解析使用,数据库可以作为整个应用的数据仓库^[11]。

客户端的系统架构图如图 4-1 所示:

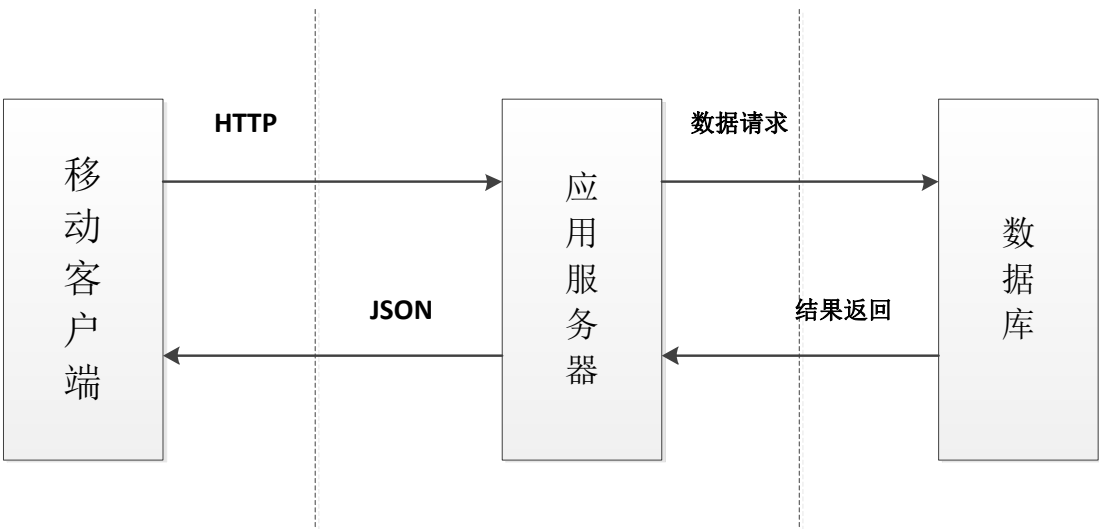


图 4-1 系统架构图

4.2 功能模块概要设计

用户登录成功之后,可以使用的功能模块有:莞工支付、视频会议、校园资讯、办事大厅、事物日程、消息通知、联络通讯和个人数据整合等八个大模块,其中莞工支付模块的主要功能:校内码支付、校园卡充值、校园卡提现、校园卡管理的挂失、账户安全、暂停使用。消息通知模块的校内消息通知和应用消息通知。

客户端的总体模块图如图 4-2 所示:

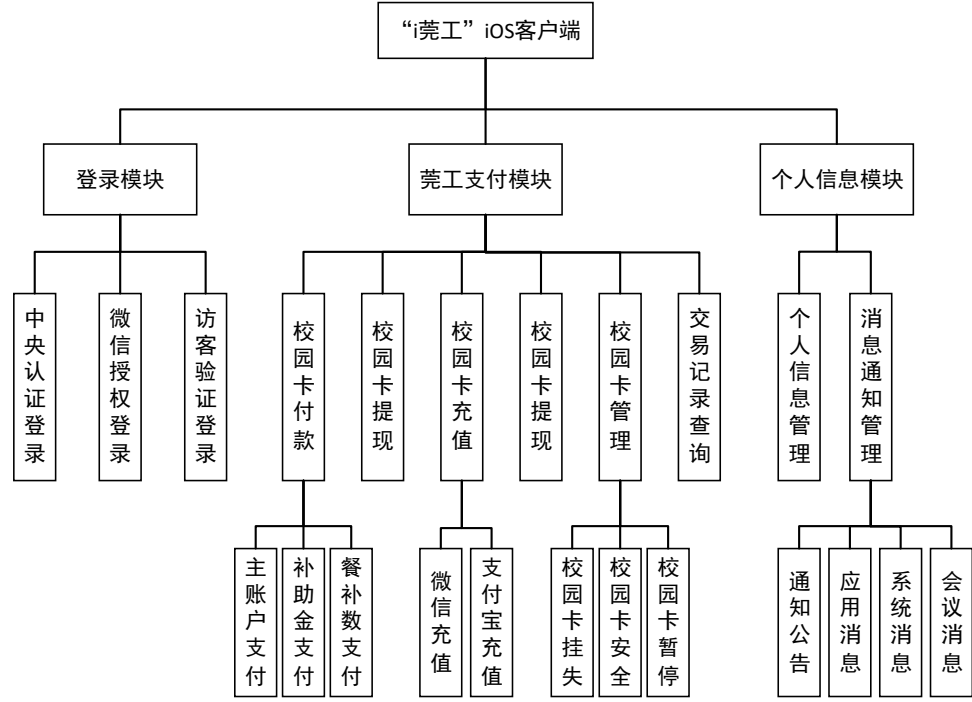


图 4-2 客户端的总体模块图

4.3 数据库概要设计

用户实体：用户 ID，令牌，姓名，学号/工号，密码，用户状态。

客户端的用户实体属性图如图 4-3 所示：

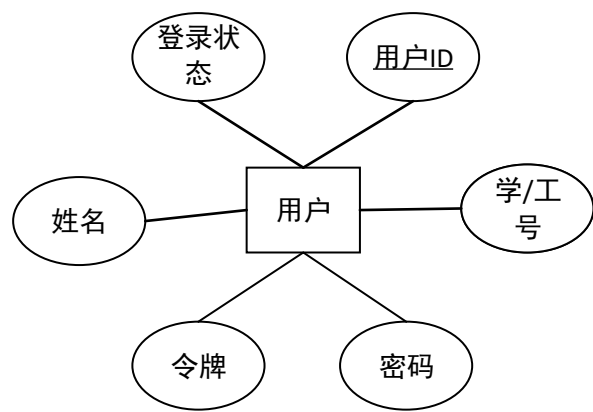


图 4-3 客户端的用户实体属性图

第 5 章 详细设计

5.1 功能模块详细设计

本章主要介绍了用户登录模块，校园卡付款模块，校园卡充值模块，校园卡提现模块，校园卡转账模块，校园卡管理模块，校园卡交易记录查询模块，消息通知模块的设计与实现。

5.1.1 用户登录模块设计

用户登录功能流程：首先，启动应用，通过读取本地数据库 Sqlite 查看用户是否为空，若用户为空直接进入登录界面，若用户不为空，则查看用户的登录状态，如果用户的状态为已登录，则可以通过人脸识别或者指纹识别进行验证进入客户端首页，若检测到用户状态为登出或者默认（未登录），直接跳转至用户登录界面，用户登录界面有 3 种登录方式，分别为中央账户密码登录，微信授权登录，访客登录。全部输入使用虚拟键盘进行输入，并判断字符串是否符合要求，本地验证数据格式无误之后将数据传至后台服务器响应，客户端通过服务器 Json 返回的数据信息进行判断能否登录成功。

用户登录功能流程图如图 5-1 所示：

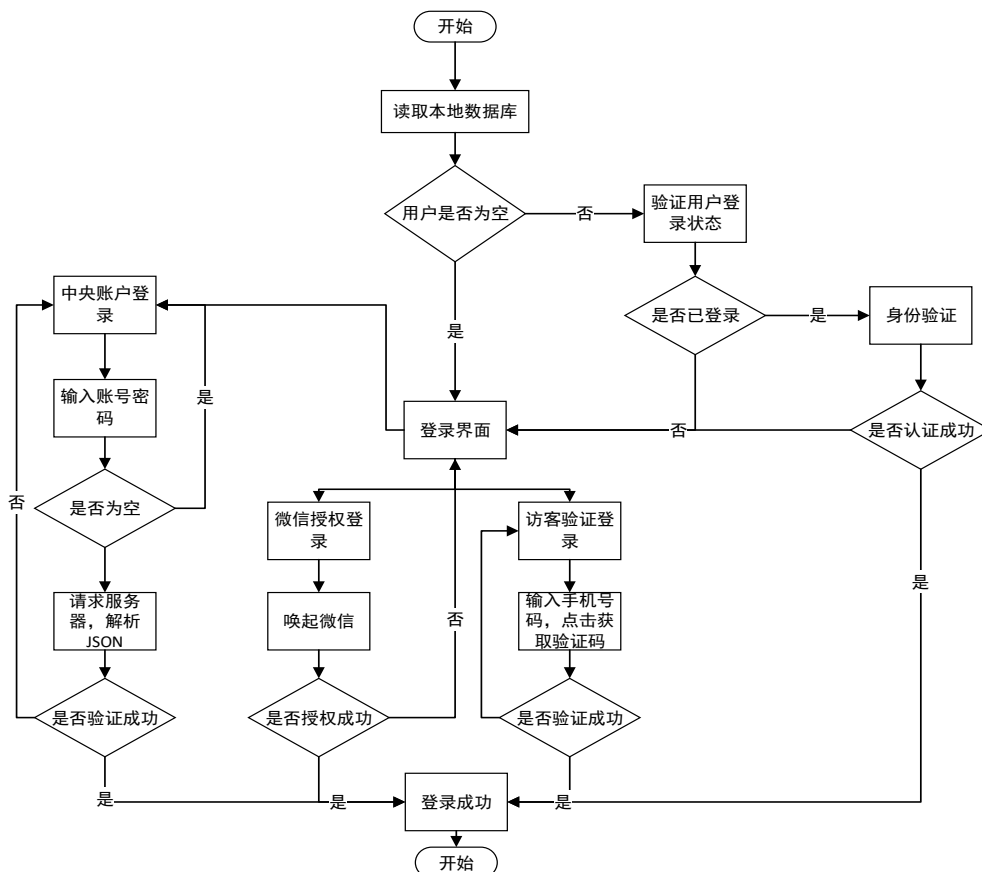


图 5-1 用户登录功能流程图

5.1.2 校园卡付款功能模块

校园卡付款功能流程：首先点击莞工支付标签，进入莞工支付页面，先判断用户是否已经开通莞工支付功能，若无，先获取开户的 OpenID，通过获取 OpenID 后进行开户请求并保存用户信息，点击选择支付方式，有主账户，补助金，和餐补数，根据消费方式选择对应的支付方式，与后台数据交互后，生成对应的付款码，付款码生成后以二维码的形式呈现，通过定义一个定时器，每隔 2 秒用运行循环 Runloop 机制^[12]，实时检测当前二维码的消费情况，若 60 秒以内都没消费，此二维码就会过期。点击二维码即立即刷新付款码，消费成功后，会出现消费金额，支付方式等信息，点击完成回到付款界面，此时会以推送消息的形式出现付款扣除金额信息，在消息通知模块的应用消息可以查询。

校园卡付款功能流程图如图 5-2 所示：

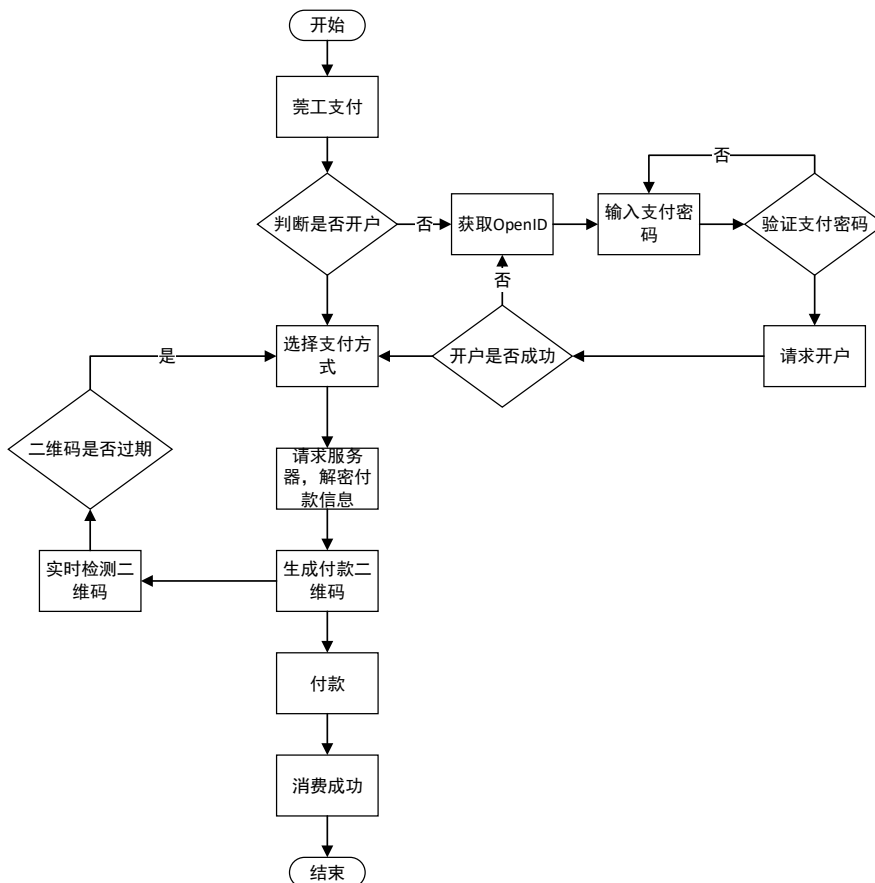


图 5-2 校园卡付款功能流程图

5.1.3 校园卡充值功能模块

校园卡充值功能流程：首先进入莞工支付页面，选择充值的金额，有 10 元-500 元可供用户选择，但是充值金额不得超过六千元，选择好充值金额后，选择充值方式，有微信支付和支付宝支付两种，首先客户端先向服务器请求充值订单，客户端解析服务器端返回的 json 信息后进行解密，提取到充值信息，确定无误后，调用第三方微信支付，支付宝支付的 SDK，打开本地的相应应用进行充值，充值成功后，在 SDK 的回调中发送通知告知充值成功，此时再请求服务器获得充值成功后的余额显示^[12]。

校园卡充值功能流程图如图 5-3 所示：

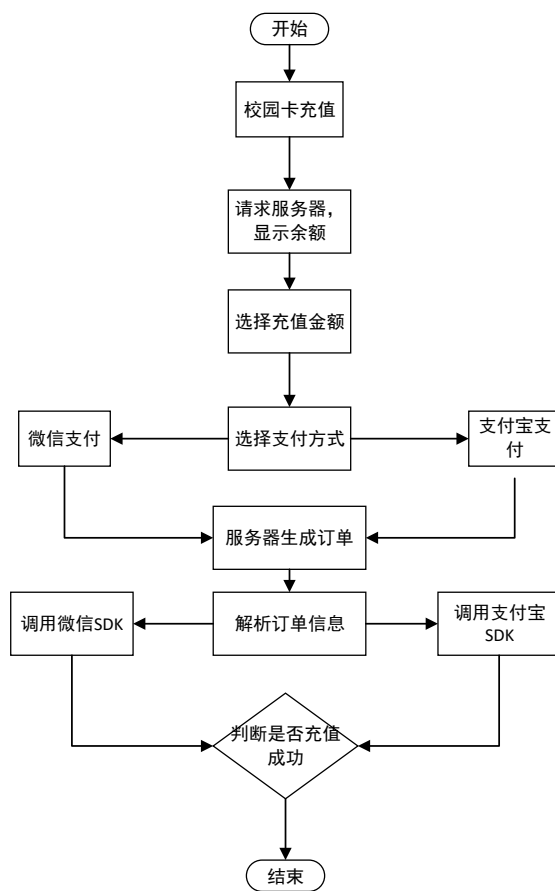


图 5-3 校园卡充值功能流程图

5.1.4 校园卡转账功能模块

校园卡转账功能：首先进入莞工支付页面，点击转账功能，收款码会以二维码的形式呈现，收款方可以选择收款码是主账户，还是餐补数，点击可以随时切换，付款方打开客户端的扫一扫功能，通过扫描对方收款码就能进入付款页面，

输入对应的金额，输入支付密码验证成功后，点击确认转账，就可以转账成功，此时收款方会受到一条推送消息，显示账户到账金额信息，付款方也会收到一条推送信息显示扣除账户余额相应的转账金额^[13]。

校园卡转账功能流程图如图 5-4 所示：

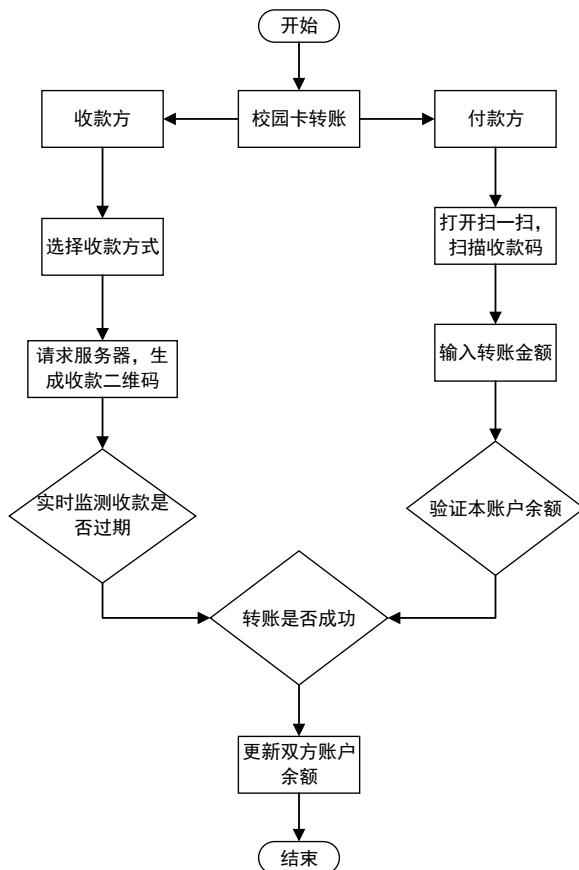


图 5-4 校园卡转账功能流程图

5.1.5 校园卡管理功能模块

校园卡管理功能：主要包括挂失功能，选择挂失功能，首先要输入支付密码以验证身份，初始密码为电子校园卡卡号后 6 位，通过虚拟键盘输入密码进行身份验证，验证成功后挂失，点击取消，可以取消挂失。账户安全的功能是重置支付密码，也要输入原支付密码以验证身份，验证成功后输入两次新的支付密码，验证成功后即修改成功^[12]；设置免密金额可以设置为支付不需要密码，50，100，200 元/笔。暂停使用通过对话框出现温馨提示，确定是否要暂停使用莞工支付。

校园卡管理功能流程图如图 5-6 所示：

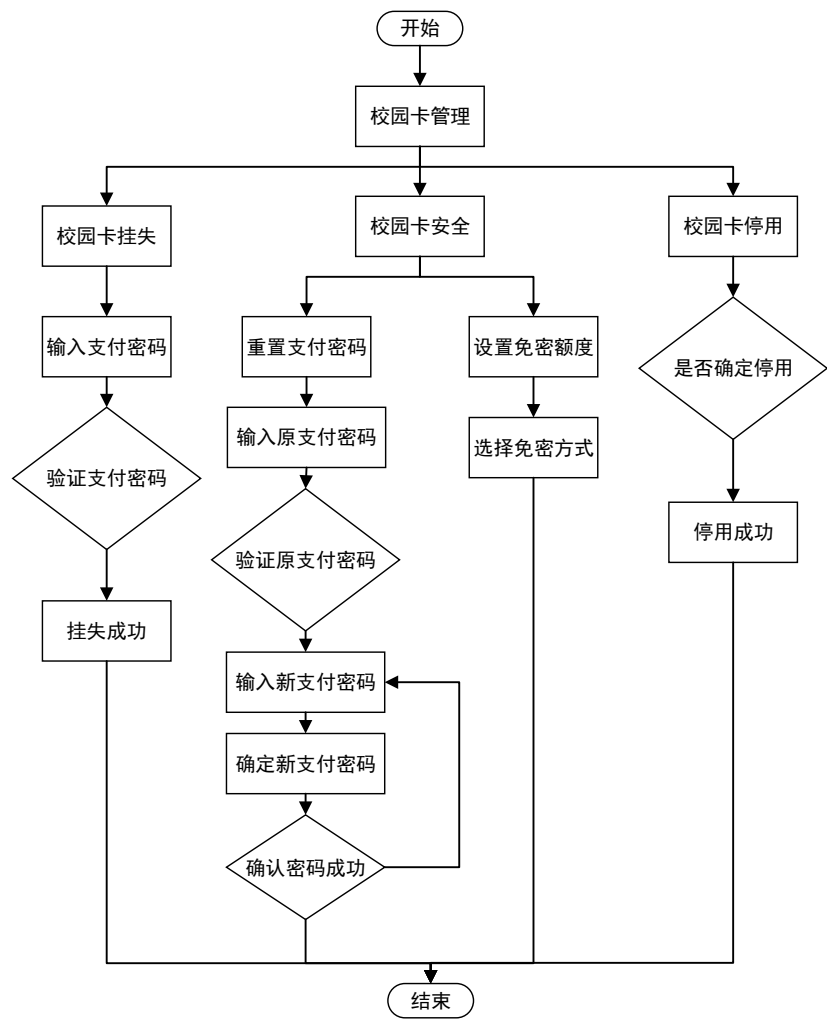


图 5-5 校园卡管理功能流程图

5.1.6 校园卡提现功能模块

校园卡提现功能：首先根据查询本账户的当前总余额，选择提现金额的方式有两种，分别是全部提现，和输入金额提现的方式，点击全部提现按钮，提现金额框上会显示当前账户的全部余额，输入金额提现方式的输入金额必须少于等于当前可提现余额，否则会出现余额不足的提示信息，提现用户必须为微信实名用户，提现金额最小为 1 元，且单日单笔限额 20000 元，通过点击申请提现按钮，就能提现至微信绑定用户，提现金额的是立马到账，同时以推送消息的形式出现提现信息，扣除相应的账户余额，打开微信，也可以看到微信交易的入账提示。

校园卡提现功能流程如图 5-5 所示：

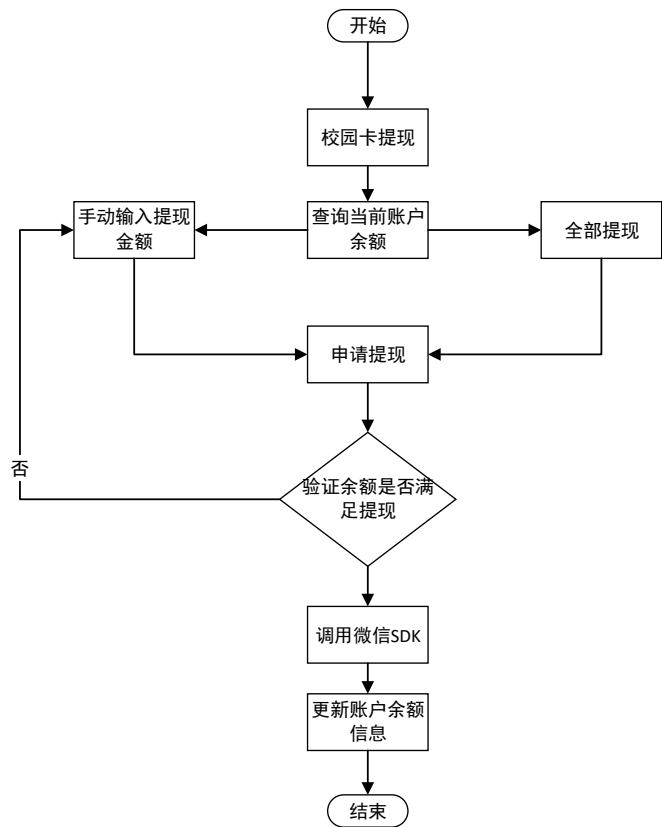


图 5-6 校园卡提现功能流程图

5. 1. 7 交易记录查询功能模块

交易记录查询功能：主要分为校内码交易查询和校外码交易查询，查询信息通过列表的方式进行显示，上拉加载更多交易记录信息，主要包括：消费地点，支付方式，交易时间，消费方式，消费金额。

交易记录查询功能流程图如图 5-7 所示：

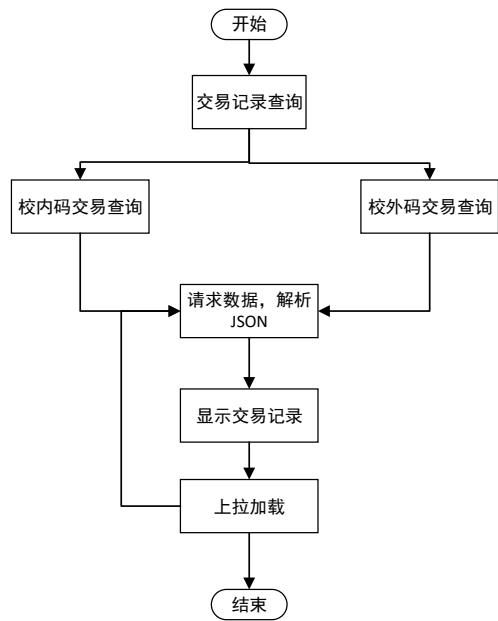


图 5-7 交易记录查询功能流程图

5.1.8 个人信息编辑功能模块

个人信息编辑包括编辑：姓名，性别，班级，邮箱，微信号，QQ, 短号，长号，地址等等，这些信息除了头像外，其余信息与通讯类信息同步。

个人信息编辑功能流程如图 5-8 所示：

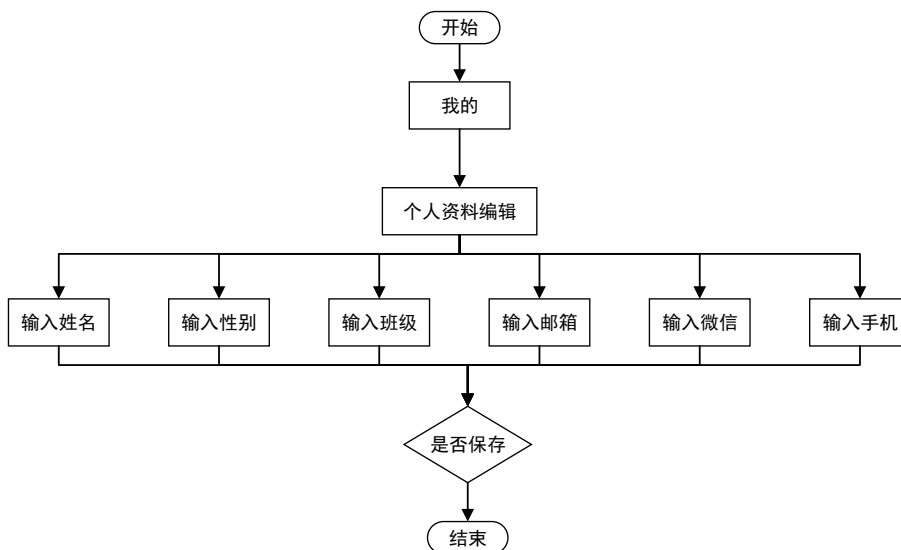


图 5-8 个人信息编辑功能流程图

5.1.9 消息通知功能模块

消息通知模块功能：通知公告，应用消息，系统消息，会议消息。通过点击一键已读按钮全部标记为已读，搜索栏内的查询根据关键字进行搜索^[13]。

消息通知功能流程如图 5-9 所示：

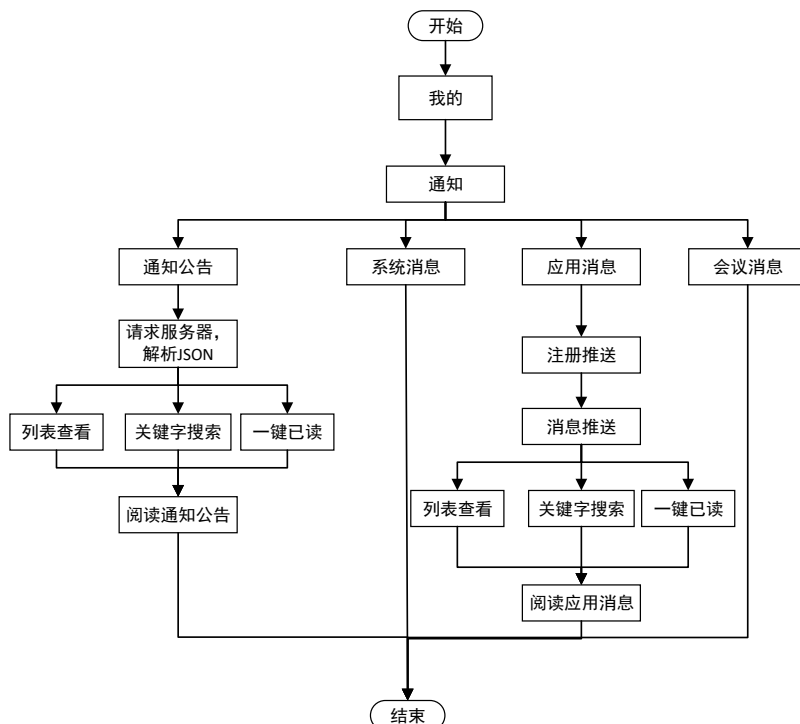


图 5-9 消息通知功能流程图

5.2 数据库详细设计

5.2.1 数据库表定义

(1) 用户信息表字段：用户的 id，用户的令牌 token，姓名 username，学号/工号 usernumber，密码 password，用户状态 userstatus。

(2) 数据来源：用户请求后台数据成功返回 Json，对 Json 数据解析并解密。出于数据安全考虑，存储在本地数据库的信息是经过 AES 加密算法进行加密之后来存储的，因此在插入的时候要进行加密，读取的时候要进行解密。因为本客户端的所有数据都是通过 HTTP 请求与学校后台服务器直接交互，本地数据库只需要存储用户登录信息，故只有一张用户 User 表，存储用户登录信息，其余数据库表都是存储在学校服务器中，不在本客户端的设计范围内。

User 表如表 5-1 所示：

表 5-1 user 表

序号	字段名	数据类型	长度	允许空	主键/外键/索引	描述
1	id	bigint	20	NOT NULL	PK	用户 ID，自动递增
2	Username	varchar	255			用户名
3	Usernumber	varchar	255			学号/工号
4	Userstatus	bigint	20			登录状态
5	Usertoken	varchar	255			令牌
6	password	varchar	255			密码

第 6 章 客户端实现

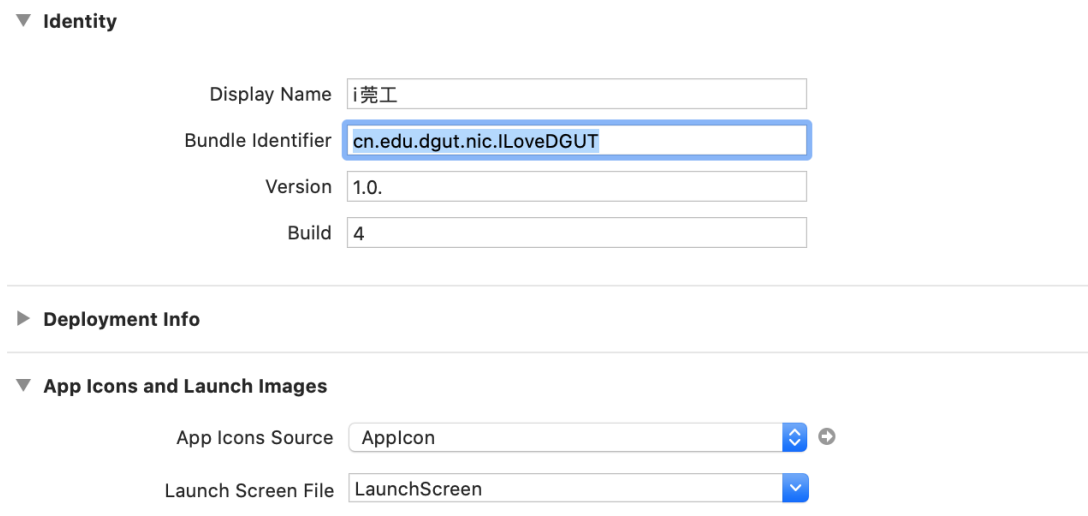
第三章主要对本客户端进行需求分析，回答了客户端的需求是做什么，具体要实现哪些功能。第四章在第三章的基础对客户端进行了概要设计，对客户端进行了功能模块的初步划分。第五章在第四章的基础上，进一步对各功能模块进行了详细的设计。第六章在以上各章的基础上，给出了客户端各功能模块的主要实现方法和运行截图。

6.1 客户端工程结构

6.1.1 配置客户端应用信息

首先配置的是 iOS 客户端的身份信息 identity，Display Name 应用名称：i 莞工，Bundle Identifier 应用标识：cn.edu.dgut.nic.ILoveDGUT，应用图标已经启动页加载。

应用配置图如图 6-1 所示：



▼ Identity

Display Name i 莞工

Bundle Identifier cn.edu.dgut.nic.ILoveDGUT

Version 1.0

Build 4

► Deployment Info

▼ App Icons and Launch Images

App Icons Source AppIcon

Launch Screen File LaunchScreen

图 6-1 应用配置图

6.1.2 配置客户端框架

iOS 应用开发常见的第三方框架库可以给我们的开发带来极大的便利，特别是其对多线程和内存管理的封装处理，替我们解决了很多的常见问题。我们管理第三方库使用的是 Cocoapods，将框架的名称通过 Pod，我们就可以获取至程序中，成为动态库。

Cocoapods 配置第三库如图 6-2 所示：

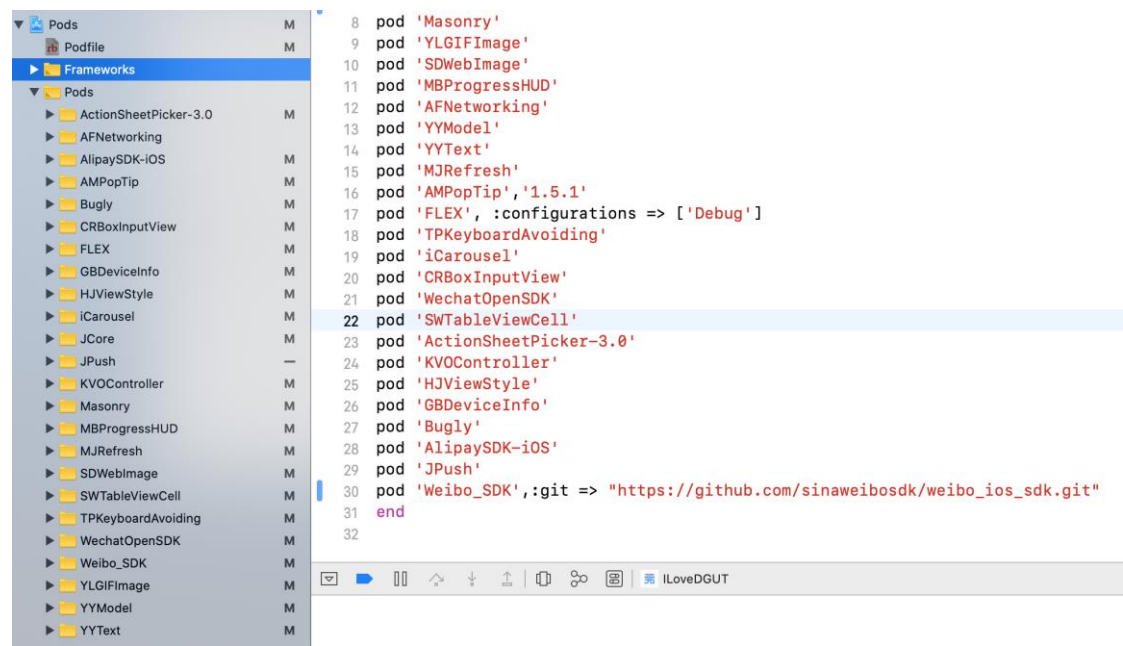


图 6-2Cocoapods 配置第三方库图

6. 1. 3 配置客户端权限

iOS 应用开发中用一个全线配置文件 info.plist，该文件包含了整个 iOS 应用的权限信息，包括数据请求安全，日历访问权限，麦克风访问权限，相册照片访问权限，消息通知权限，地理位置获取权限等等，我们都需要直接配资在 info.plist 资源文件中，该文件的底层是 XML 文件，便于代码话编译修改^[13]。

资源文件权限配置图如图 6-3 所示

Key	Type	Value
Information Property List	Dictionary (27 items)	
Localization native development re...	String	\$(DEVELOPMENT_LANGUAGE)
Bundle display name	String	i 莞工
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle version string (short)	String	\$(MARKETING_VERSION)
URL types	Array (5 items)	
Bundle version	String	\$(CURRENT_PROJECT_VERSION)
LSApplicationQueriesSchemes	Array (28 items)	
Application requires iPhone enviro...	Boolean	YES
App Transport Security Settings	Dictionary (1 item)	
Privacy - Calendars Usage Descri...	String	我们需要您的允许才能访问日历建立日程
Privacy - Camera Usage Description	String	为了让其他人可以看到您的视频，请点击“好”以允许访问
Privacy - Face ID Usage Description	String	我们需要您的允许才能进行脸部识别解锁应用
Privacy - Location Always Usage D...	String	为了让客服可以看到您的位置，请点击“好”以允许访问
Privacy - Location When In Use Us...	String	为了让客服可以看到您的位置，请点击“好”以允许访问
Privacy - Microphone Usage Desc...	String	为了让其他人可以听到您的声音，请点击“好”以允许访问
Privacy - Photo Library Additions...	String	我们需要您的同意才能保存图片到相册
Privacy - Photo Library Usage Des...	String	我们需要您的同意才能读取您的照片上传头像
Required background modes	Array (3 items)	
Launch screen interface file base...	String	LaunchScreen
Required device capabilities	Array (1 item)	
Supported interface orientations	Array (1 item)	
Supported interface orientati...	Array (4 items)	
User Interface Style	String	Light

图 6-3 资源文件权限配置图

6.2 用户登录模块

6.2.1 中央账号认证登录

虚拟键盘输入校园中央认证账户密码之后，首先进行格式判断，格式无误后，点击登录按钮，将账户密码信息加密后，请求服务器信息，解析回传的 Json 数据，判断用户是否登录成功。

部分代码：

```
#pragma mark 用户名密码登录

-(void)request_Login_withUsername:(NSString *)username password:(NSString *)pwd andBlock:(void (^)(id response, NSError * error))block{
```

中央账号登录图如图 6-4 所示：



图 6-4 中央账号登录图

6.2.2 微信授权登录

若本地安装了微信客户端，其他登录渠道会出现微信登录图标，点击微信登录，随机唤醒微信客户端。当用户已将账户通过微信绑定时，会出现微信登录“i

莞工”客户端的授权信息，点击同意，即可登录成功，登录成功一次后，后面通过微信登录都可以直接一键微信登录。

部分代码：

```
#pragma mark 微信登录

-(void)request_Login_withWechatCode:(NSString *)code andBlock:(void (^)(id response, NSError * error))block{}

#pragma mark 微信回调

-(void)onResp:(BaseResp *)resp{

    if([resp isKindOfClass:[SendAuthResp class]]){//判断是否为授权登录类

        SendAuthResp *req = (SendAuthResp *)resp;

        if([req.state isEqualToString:WXAuthorizationstate]){//微信授权成功
```

6.2.3 访客登录

访客登录，指的是校外人员，可以通过手机验证进行注册登录，属于临时用户。

部分代码：

```
#pragma mark 获取手机验证码

-(void)request_PhoneCode_withPhone:(NSString *)phone type:(int)type andBlock:(void (^)(id response, NSError * error))block{}

#pragma mark 检查手机验证码是否正确

-(void)request_checkPhoneCode_withPhone:(NSString *)phone type:(int)type code:(NSString *)code andBlock:(void (^)(bool isPass, NSError * error))block{}

#pragma mark 绑定手机

-(void)request_BindPhone_withUsername:(NSString *)username password:(NSString *)password phone:(NSString *)phone code:(NSString *)code andBlock:(void (^)(id response, NSError * error))block{}
```

访客登陆图如图 6-5 所示：



图 6-5 访客登录图

6.3 校园卡付款模块

用户登录成功后，点击莞工支付标签，进入莞工支付，根据主账户，补助金，餐补数进行选择支付类型，付款码以二维码的方式出现。

部分代码：

```
#pragma mark 判断是否已经开过户

-(void)pay_isAccountOpen_withBlock:(void (^)(DGPayJudgeRegisterResult * _Nullable, NSError * _Nullable))block{}

#pragma mark 拉取二维码接口

-(void)pay_genQrCode_withType:(DGPayType)accTrType
ecardId:(NSString *)ecardId andBlock:(void (^)(DGPayGenQrCodeResult * _Nullable, NSError * _Nullable))block{}

#pragma mark 实时监测二维码状态

[[NSRunLoop mainRunLoop] addTimer:_timer
forMode:NSRunLoopCommonModes];
```

校园卡付款图如图 6-6 所示：

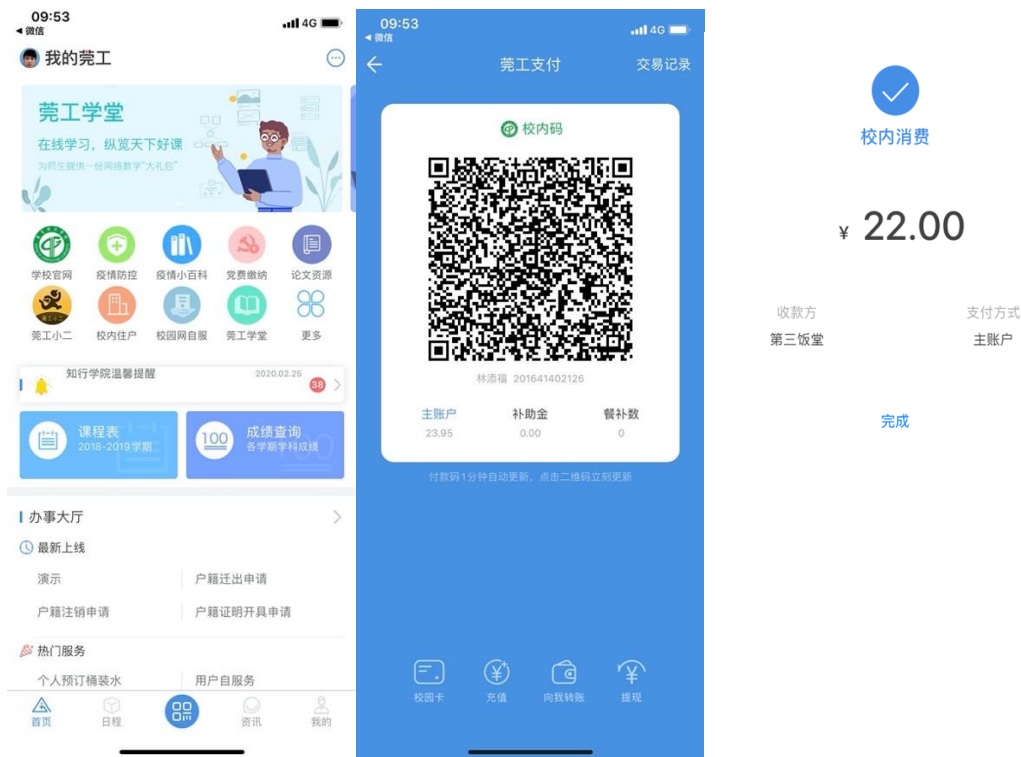


图 6-6 校园卡付款图

6.4 校园卡充值模块

点击充值功能，进入校园卡充值界面，选择充值金额，选择充值方式，点击确认充值，唤醒对应的第三方支付客户端，充值成功后推送通知。

部分代码：

```
#pragma mark 微信支付充值订单生成

-(void)pay_recharge_withTotallAmt:(NSString *)totallAmt openid:(NSString *)openid andBlock:(void (^)(DGPAYRechargeResult * _Nullable res, NSError * _Nullable error))block{}

#pragma mark 支付宝充值订单生成

-(void)pay_recharge_AliPayOrder_withTotallAmt:(NSString *)totallAmt ecardId:(NSString *)ecardId mem:(NSString *)mem andBlock:(void (^)(DGPAYAliOrderResult * _Nullable res, NSError * _Nullable error))block{}
```

校园卡充值图如图 6-7 所示：

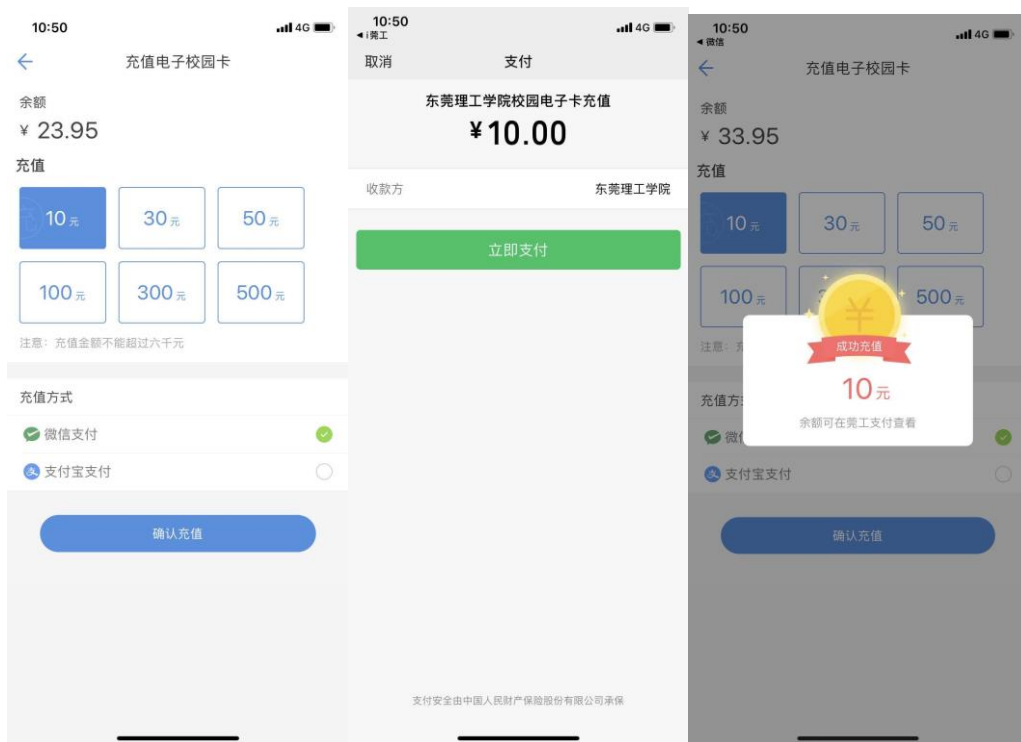


图 6-7 校园卡充值图

6.5 校园卡转账模块

收款方点击主账户或者餐补数，生成对应的收款二维码，付款方打开扫一扫功能，扫描收款方上的收款码，识别成功后进入金额确认界面，点击确认转账按钮，成功后推送通知。

部分代码：

```
#pragma mark 向我转账-收款方调用

-(void)pay_payToMeGenCode_withEcardId:(NSString *)ecardId andBlock:(void (^)(DGPayToMeResult * _Nullable, NSError * _Nullable))block{}

#pragma mark 向我转账-付款方调用

-(void)pay_transferToMe_withAccTrType:(DGPayType)type accCode:(NSString *)accCode password:(NSString *)password totallAmt:(NSString *)totallAmt ecardId:(NSString *)ecardId andBlock:(void (^)(DGPaySubmitPayInfoResult * _Nullable, NSError * _Nullable))block{}
```

校园卡转账图如图 6-8 所示：

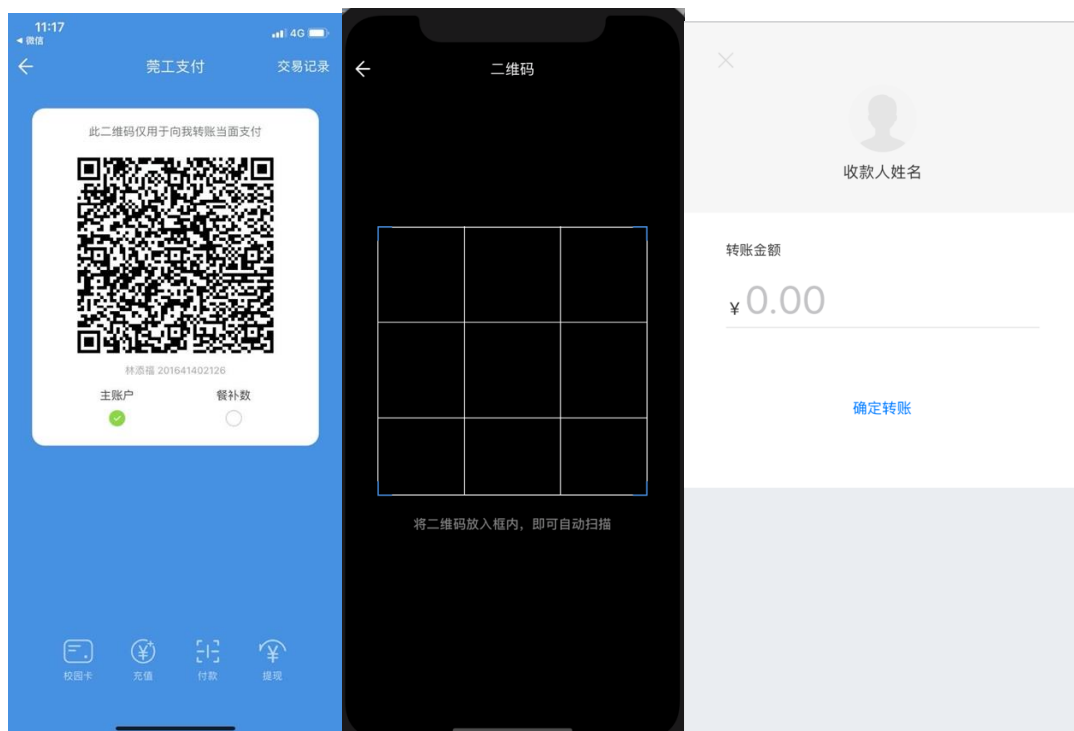


图 6-8 校园卡转账图

6.6 校园卡提现模块

首先查询当前账户余额并显示，点击全部提现，或者手动输入提现金额，请求服务器判断金额正确后，即可提现至用户微信。

部分代码：

```
#pragma mark 主账户提现

-(void)pay_refund_withRefundAmt:(NSString *)refundAmt openid:(NSString *)openid andBlock:(void (^)(DGPAYBaseResult * _Nullable res, NSError * _Nullable error))block {}

#pragma mark 获取用户绑定的微信信息

-(void)request_UserWechatInfo_withBlock:(void (^)(DGWechatInfo *info, NSError * error))block {}

#pragma mark - 提现服务协议

-(void)request_AgreementWithType:(DGAgreementType)type andBlock:(void (^)(NSString * _Nullable res, NSError * _Nullable error))block {}
```

校园卡提现图如图 6-9 所示：

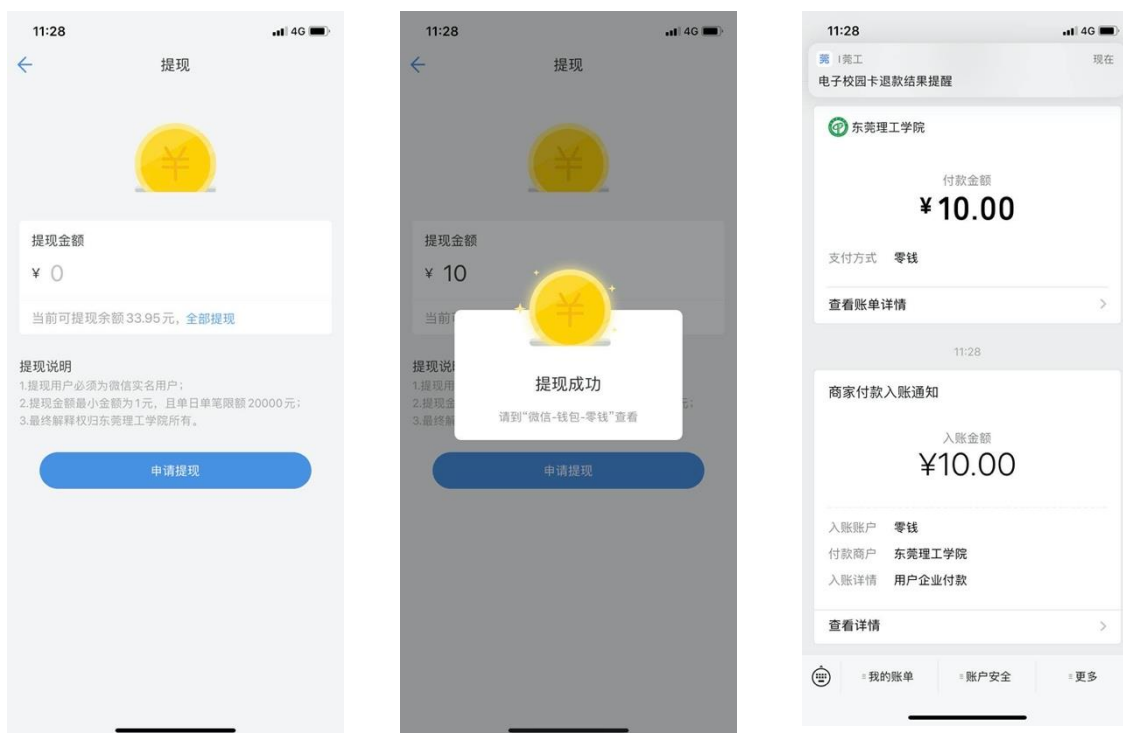


图 6-9 校园卡提现图

6.7 校园卡管理模块

校园卡进行挂失，账户安全设置，暂停使用。

部分代码：

```
#pragma mark 查询账户详情

-(void)pay_userDetail_withEcardId:(NSString *)ecardId andBlock:(void (^)(DGPAYUserDetailResult * _Nullable, NSError * _Nullable))block{

#pragma mark 暂停莞工支付使用

-(void)pay_paushSwitch_withEcardId:(NSString *)ecardId
isSwitch:(BOOL)isSwitch andBlock:(void (^)(DGPAYBaseResult * _Nullable,
NSError * _Nullable))block{}

#pragma mark 重置支付密码，获取rsa加密的的公钥

-(void)pay_pswReset_RSA_withEcardId

#pragma mark 挂失/解挂

-(void)pay_lossEntityCard_withEcardId:(NSString *)ecardId
```

校园卡管理图如图 6-10 所示：

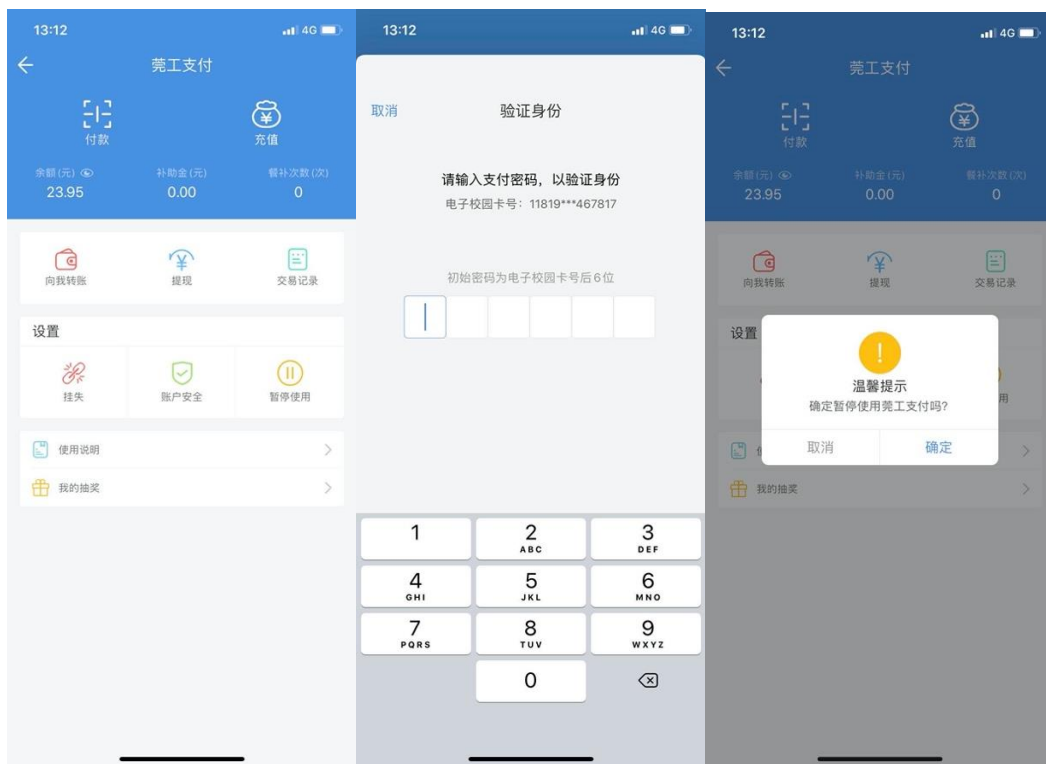


图 6-10 校园卡管理图

6.8 个人信息整理模块

主要编辑用户的姓名，性别，班级，邮箱，微信号等等。

部分代码：

```
DGUser *user = [DGSession sharedSession].user;

self.usernameLabel.text = [NSString stringWithFormat:@"%@", user.name];

[self.userImageView dg_setImageURL:user.contact_avatar
placeholder:nil];

self.userNumLabel.text = [NSString
stringWithFormat:@"%学号: %@", user.username];

self.userTypeLabel.text = @"学生";

self.userDepartmentLabel.text = [NSString
stringWithFormat:@"%班级: %@", user.class_title];

self.bottomView.backgroundColor = [UIColor
colorWithHexString:@"0x54A96A "];

self.bottomWorkBg.image = [UIImage imageNamed:@"work_bg1"];
```

个人信息编辑图如图 6-11 所示：



图 6-11 个人信息编辑图

6.9 消息通知模块

显示的是校园的通知公告，应用消息，系统消息，会议消息首先注册极光推送，注册 PUSH 服务。

部分代码：

```
#pragma mark - 注册极光推送,注册Push服务,注册后才能收到推送

- (void)registerPush{

    JPUSHRegisterEntity *entity = [[JPUSHRegisterEntity alloc] init];

    entity.types =
    JPAuthorizationOptionAlert|JPAuthorizationOptionBadge|JPAuthorizationOptionSound;

    [JPUSHService registerForRemoteNotificationConfig:entity
    delegate:self];

}

[JPUSHService setupWithOptions:launchOptions
appKey:@"a95ff6f761bfcd945ae46e39" channel:@"App Store"];
```

消息通知图如图 6-12 所示



图 6-12 消息通知图

第 7 章 客户端测试

客户端测试是软件开发中必不可少的一个重要环节。通过测试，我们可以知道程序运行中出现的 Bug 并且对号入座进行修改完善，提升程序的运行性能，打造出具有极致用户体验的 iOS 客户端^[13]。

7.1 测试环境

操作系统：MAC OS；

数据库：Sqlite；

应用软件：Xcode，Instrument。

硬件设备：MacBook Pro，iPhone XR。

软件版本号：1.0.1

7.2 测试方法

本客户端的测试方法首先使用了 Xcode 自带的测试工具和 iOS 里面的单元测试模块对所有功能进行测试，对每一个功能项的预期结果与实际结果结合对比，更好的完善应用。

7.3 测试用例设计

7.3.1 用户登录测试用例

“i 莞工”客户端的使用用户是东莞理工学院的全体师生教职工以为校外人员也可通过注册使用，但是无论是通过中央认证账户，微信登录，访客登录，都是需要先登录才能使用本客户端的所有应用功能。

用户登录测试用例如表 7-1 所示：

用户登录测试用例 7-1

编号	用例名	预期结果	实际结果
T1001	第一次启动应用	出现指引图，点击进入应用，出现登录界面	滑动指引图，进入登录界面，出现账号密码框，头像空白
T1002	使用正确的中央账号密码登录	登录成功，进入首页	登录成功，进入客户端首页
T1003	使用正确的中央账号和错误的密码	提示用户名或者密码有误，请重新输入	提示用户名或者密码有误
T1004	使用错误的中央账号和正确的密码	提示用户名或者密码有误，请重新输入	提示用户名或者密码有误，请重新输入
T1005	使用空的中央账号	提示用户名不能为空	提示用户名不能为空
T1006	使用空的密码	提示密码不能为空	提示密码不能为空
T1007	使用微信授权登录	微信弹出授权	授权成功，进入应用首页
T1008	使用访客登录，手机号为空	提示手机号不能为空	提示手机号不能为空

T1009	使用访客登录，验证码为空	提示验证码不能为空	提示验证码不能为空
T1010	短信验证码验证	验证码过期	提示验证码过期

7.3.2 莞工支付测试用例

莞工支付是“i 莞工”客户端的核心模块，里面有校园卡付款模块，校园卡充值模块，校园卡转账模块，校园卡提现模块，校园卡管理模块。

莞工支付测试用例如表 7-2 所示：

莞工支付测试用例表 7-2

编号	用例名	预期结果	实际结果
T2001	进入莞工支付	出现付款二维码	出现付款二维码
T2002	刷新二维码	点击二维码立即刷新，付款码 1 分钟自动更新	点击二维码立即刷新，付款码 1 分钟自动更新
T2003	更换支付方式	点击主账户，补助金，餐补数随机切换	切换支付方式，二维码立即更新，生成对应的付款码
T2004	检测付款码状态	实时检测付款码消费状况	实时检测，付款码消费后出现消费成功界面
T2005	校园卡充值，检测本地无第三方应用	出现网页版充值界面	出现网页版充值界面，登录后进行支付宝或者微信支付
T2006	校园卡转账	点击二主账户和餐补数随机切换	切换支付方式，二维码立即更新，生成对应的收款码
T2007	校园卡提现，输入提现金额大于余额	弹出账户余额不足	弹出账户余额不足
T2008	校园卡提现，输入提现金额不大于余额	提现至微信零钱	提现至微信零钱，并出现消息推送通知
T2009	校园卡管理挂失，支付密码正确	提示挂失成功	提示挂失成功
T2010	校园卡管理挂失，支付密码错误	提示支付密码错误，重新输入	提示支付密码错误，重新输入
T2011	校园卡管理账户安全，重置支付密码	弹出输入原支付密码	提示请输入原支付密码以验证身份
T2012	校园卡管理暂停使用	弹出温馨提示	弹出温馨提示是否确定暂停使用

7.3.3 消息通知测试用例

消息通知有校内公告通知和应用消息通知，系统消息，会议消息。

消息通知测试用例如表 7-3 所示：

消息通知测试用例表 7-3

编号	用例名	预期结果	实际结果
T3001	通知公告	学校发送通知公告，推送	以推送的形式进行消息通

		出现	知，点击进入公告。
T3002	应用消息通知	充值，消费等业务操作，推送出现	付款成功，充值成功，都以推送的形式出现。
T3003	消息一键已读	点击一键已读，全部消息标记状态为以读	点击一键已读，全部消息标记状态为以读

7.4 结果和分析

在对本客户端的所有功能进行测试后发现，测试效果与预期效果基本吻合，所有功能达到预期目的。在 UI 布局方面：UI 界面布局安排合理，用户可以便捷操作。在性能测试方面：在测试过程中，无论是下拉加载刷新数据，还是下滑加载列表，都未曾出现 UI 卡顿现象，这得益于网络编程多线程的处理与内存分配的管理。在数据安全方面：数据处理安全性高，采用了多层加密算法。在可扩展性方面：客户端采用的设计模式是 MVC，功能模块分层，无论在扩展新功能，还是完善客户端功能，都能较为容易的进行相应的操作。

总结与展望

“i 莞工”客户端，从东莞理工学院实际需求出发，实现了校园资讯的快速传达和校内生活服务资源的高效利用，客户端的实际投入使用效果与预期基本一致，但是在开发过程中，仍然存在着很多需要继续完善的地方，例如界面布局，性能优化，内存管理等方面。在莞工支付模块中，可以采用多层加密算法，实现数据传输安全性的稳定提升。与此同时，越来越多的新功能会增加到客户端上去，争取打造出具有极致用户体验的东莞理工官方 APP。

本论文从研究背景出发，探寻了“i 莞工”客户端的研究意义，以及后续的功能模块阐述，客户端的详细设计实现和测试。论文详细的说明了客户端的具体实现方法和实现效果。本客户端的创新之处就是莞工支付，从最初的想法到一步步的具体实现，都实时为东莞理工学院师生电子卡消费服务量身定制。

总之，在指导老师的辛勤指导和自己的辛勤努力下，通过半年的时间完成了本客户端的部分模块的设计开发与实现，且已通过测试上传应用商店，用户可随时下载使用，能基本满足师生的实际需求，具有真实的实际意义。本论文的编写，符合标准的论文规范，清晰的展示了客户端功能模块的具体详情，但是也会有不知之处，希望各位专家学者能够提供指导意见，我会听取并加以改正，继续完善本客户端与论文。

参考文献

- [1] 彭英杰. 浅析 iOS 应用开发——以 QA 项目为例[J]. 软件工程, 2018,21(09):8-10.
- [2] 黄浏展. iOS 开发中多线程技术的研究和实践[J]. 软件工程, 2018,21(11):38-41.
- [3] 喻晓, 陆澄澹, 黄秋霞, 刘健. 基于 iOS 的项目工时管理客户端的设计与实现[J]. 计算机时代, 2018(10):22-25.
- [4] 刘筱琪. 基于 iOS 平台的应用安全性研究[D]. 南昌大学, 2018.
- [5] 汤奇. 基于 iOS 平台的图片分享应用软件优化方案的设计与实现[D]. 东南大学, 2018.
- [6] 王娟. iOS 平台应用程序攻防技术的研究与应用[D]. 南昌航空大学, 2018.
- [7] 罗洁. 基于 iOS 的流式架构库的研究实现与应用验证[D]. 北京邮电大学, 2018.
- [8] 贺宇轩, 孟魁, 刘功申, 徐林. iOS 系统数据安全分析与加固[J]. 通信技术, 2014,47(06):668-673.
- [9] 严小金. 移动支付系统中客户端与支付平台安全性研究[D]. 武汉理工大学, 2014.
- [10] 陈佳霖. iOS 平台应用程序安全性研究[D]. 上海交通大学, 2014.
- [11] 张永. 基于 iOS 系统的手机 App 前端设计[J]. 数字通信世界, 2018(12):60-120.
- [12] 毛丰. 基于 Unreal Engine 中 ARKIT 插件的 iOS 平台 AR 图像识别应用程序开发[J]. 新媒体研究, 2019,5(03):29-30.
- [13] 罗嘉炜. 基于 iOS 平台的移动学习系统设计与实现[D]. 广西师范大学, 2019.

致谢

本次毕业设计，从最初的题目选择，到客户端的具体实现与测试，再到论文的编写，共历经了半年的时间。期间，我经历了很多的挫折与困难，特别是在实际开发中，遇到了很多难以解决的问题，例如有些问题涉及到 iOS 开发的底层原理才能解决，由于本人在学习 iOS 开发的时候，比较少接触到底层原理相关，因此走了很多弯路。在这一过程中，我成长了很多，也收获了很多，通过此 i 莞工客户端的开发，让我真正意义上的打开了在 iOS 移动应用方向深入学习的大门。

当然，我的指导老师潘晓衡高级工程师给予了我很大的帮助，从客户端的设计架构，到具体功能的实现他都给了我详细的指导意见；开发阶段每次我遇到难题时，向他请教之后都能得到良好的解决方案，在此，我由衷的向潘晓衡老师表示感谢。

最后，我由衷的向东莞理工学院表示感谢，感谢母校对我四年来的辛勤培养与谆谆教诲。作为一名莞工人，我倍感自豪。我会继续努力，不断的提升自己，争取做一名杰出的校友。