

**本科毕业论文（设计）**

题 目 基于iOS平台小额贷款审批

APP的设计与实现

专 业 计算机科学与技术

作者姓名 刘剑

学 号 2012204288

单 位 计算机学院

指导教师 李树榜

**2016年5月**

**教务处编**

原创性声明

本人郑重声明：所提交的学位论文是本人在导师指导下，独立进行研究取得的成果。除文中已经引用的内容外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得聊城大学或其他教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均在文中以明确的方式表明。本人承担本声明的相应责任。

学位论文作者签名： 日期：

指 导 教 师 签 名: 日期：

**目 录**

前 言 1

1.1 课题提出的背景目的及意义 1

1.1.1 课题提出的背景 1

1.1.2 课题提出的目的 1

1.2 设计思路 1

1. 开发工具简介 2

1.1 Objective-C简介 2

1.2SQLite简介 2

1.3 ios系统架构 3

1.3.1 核心操作系统层 3

1.3.2 核心服务层 3

1.3.3 应用程序框架 4

2. 可行性分析 5

2.1 可行性分析 5

2.1.3 经济可行性 7

2.1.4 社会可行性 7

2.2 总体设计原则 7

3. 系统设计 8

3.1 系统功能分析 8

3.2模块的设计 8

4. 系统实现 9

4.1 首页模块 9

4.1.1 首页界面 9

4.1.2 登录、注册以及注册完善信息界面 12

4.2 创建公司模块 14

4.2.1 创建公司界面 14

4.3 二维码模块 16

4.3.1 选择公司二维码界面 16

4.3.2 动态二维码界面 18

4.4 客户模块 19

4.4.1 客户界面 19

4.4.2 授信资料及公司详情界面 20

4.5 消息模块 23

4.5.1 消息界面 23

4.5.2 授信助手界面 25

4.5.3 新增客户界面 27

4.5.3 催款助手及系统公告界面 28

4.6 设置模块 29

4.6.1 设置界面 29

参考文献 32

致 谢 33

**摘 要**

近年来随着智能手机和平板电脑的普及与越来越高的使用率,随着社会潮流化以及国家经济的增长,国民的消费意识的增强和消费观念的转变,小额贷款现有的线下运行模式逐渐满足不了用户的需求,因此越来越多的小额信贷APP应运而生,小额借款指的是：个人或者是家庭经营类的借款项目，主要是个体户、小业主。借款金额一般是在 5000到50万之内。主要就是微小借款在技术和实际应用上面的延生。主要就是服务于三农、中小企业、小额借款公司的设立，将一些民间资金集中了起来，规范了民间借贷市场，同时也有效地解决了三农、中小企业融资难的问题。移动平台的信贷有点主要：程序简单、放贷快、手续简便、还款方式灵活、营销模式灵活、质量高等特点。

本项目是一个基于iOS平台的移动信贷申请APP---“异卡经理”,这是一款对面向异卡经理在线审批的APP,可以让客户直接在线审批不同额度的信贷业务,用于消费、资金周转等用途,方便快捷。

**关键词：**小额信贷；异卡经理；iOS；APP

Abstract

In recent years, with the use of smart phones and tablet computer and popularity of more and more high rate, with the trend of the social and national economic growth, enhance national consumer awareness and consumer attitudes change, existing lines of microfinance operation mode gradually to meet the needs of users, so more and more microfinance app arises at the historic moment, small loan is: personal or family business loan projects, mainly self-employed, small owners. The loan amount is generally between 5000 to 500 thousand. Is tiny to borrow in technology and practical application of the above extension. Is the main service in the establishment of agriculture, small and medium-sized enterprises, small loan companies, will focus on the number of private funds up, regulate the private lending market, and effectively solves the agriculture, small and medium-sized enterprise financing difficult problem. Mobile platform's credit is a little Main: the procedure is simple, the loan is quick, the procedure is simple, the repayment method is flexible, the marketing pattern is flexible, and the quality is high.

This project is a based on the IOS platform mobile credit application app --- different card manager ", which is a opposite to different card manager approval online app, can let the customer directly online approval different amount of credit business, for consumption, capital turnover use, convenient and quick.

**Key words:** Microfinance ; Different card manager ; iOS ; APP

**基于iOS平台小额贷款审批APP**

**的设计与实现**

# 前 言

## 1.1 课题提出的背景目的及意义

### 1.1.1 课题提出的背景

随着信息化的到来，智能移动平台功能的日益强大，现有的小额信贷线下运行模式逐渐满足不了用户的需求,因此越来越多的小额信贷APP应运而生,小额借款指的是：个人或者是家庭经营类的借款项目，主要是个体户、小业主。借款金额一般是在 5000到50万之内。主要就是微小借款在技术和实际应用上面的延生。主要就是服务于三农、中小企业、小额借款公司的设立，将一些民间资金集中了起来，规范了民间借贷市场，同时也有效地解决了三农、中小企业融资难的问题。移动平台的信贷有点主要：程序简单、放贷快、手续简便、还款方式灵活、营销模式灵活、质量高等特点。

### 1.1.2 课题提出的目的

此次设计的目的很明显，就是给予急需资金周转的信贷客户一移动的在线申请后客户经理审批流程，节省客户的时间也节省了公司的人力成本。

## 1.2 设计思路

APP的设计流程主要是按照软件设计的一般流程：需求分析、概要设计、详细设计、测试与维护的基本流程来设计开发的。首先，从整体分析软件功能，将功能模块化；然后对每个模块进行调试。最后对各个模块进行系统集成。

# 1. 开发工具简介

## 1.1 Objective-C简介

Objective-C，通常写作ObjC和较少用的Objective C或Obj-C，是扩充C的面向对象编程语言。它主要使用于Mac OS X和GNUstep这两个使用OpenStep标准的系统，而在NeXTSTEP和OpenStep中它更是基本语言。Objective-C可以在gcc运作的系统写和编译，因为gcc含Objective-C的编译器。1980年代初布莱德·确斯(Brad Cox)在其公司Stepstone发明Objective-C。他对软件设计和编程里的真实可用度问题十分关心。

优点及缺点：Objective-C是非常“实际”的语言。它使用一个用C写成、很小的运行库，只会令应用程序的大小增加很小，和大部分OO系统使用极大的VM执行时间会取代了整个系统的运作相反。ObjC写成的程序通常不会比其原始码大很多。而其函式库(通常没附在软件发行本)亦和Smalltalk系统要使用极大的内存来开启一个窗口的情况相反。 Objective-C的最初版本并不支持垃圾回收。在当时这是争论的焦点之一，很多人考虑到Smalltalk回收时有漫长的“死亡时间”，令整个系统失去功用。Objective-C为避免此问题才不拥有这个功能。虽然某些第三方版本已加入这个功能(尤是GNUstep), Apple在其Mac OS X 10.3中仍未引入这个功能。 另一个问题是ObjC不包括命名空间机制(namespace mechanism)。取而代之的是程序设计师必须在其类别名称加上前缀，时常引致冲突。在2004年，在Cocoa编程环境中，所有Mac OS X类别和函式均有“NS”作为前缀，例如NSObject或NSButton来清楚分别它们属于Mac OS X核心;使用“NS”是由于这些类别的名称在NeXTSTEP开发时定下。 虽然Objective-C是C的母集，但它也不视C的基本型别为第一级的对象。 和C++不同，Objective-C不支援运算子多载(它不支持ad-hoc多型)。亦与C++不同，但和Java相同，Objective-C只容许对象继承一个类别(不设多重继承)。Categories和protocols不但可以提供很多多重继承的好处，而且没有很多缺点，例如额外执行时间过重和二进制不兼容。

## 1.2SQLite简介

sqlite是一款轻型的数据库，是遵守ACID的关联式[数据库管理系统](http://baike.baidu.com/view/68446.htm" \t "_blank)，它的设计目标是嵌入式数据库，而且目前已经在很多嵌入式产品中应用到，它占用资源非常低，同时能够跟很多程序语言相结合，而且比起Mysql、PostgreSQL这两款开源世界著名的数据库管理系统来讲，它的处理速度比它们都快。

在iOS平台上，集成了一个嵌入式关系型数据库——SQLite，SQLite3支持NULL、INTEGER、REAL(浮点数字)、TEXT(字符串文本)和BLOB(二进制对象)数据类型，虽然它支持的类型只有五种，但实际上SQLite3也接受varchar(n)、char(n)、decimal(p,s)等数据类型，只不过在运算或保存时会转成对应的五种数据类型。SQLite最大的特点是可以把各种类型的数据保存到任何字段中，而不用关心字段声明的数据是什么类型。

## 1.3 ios系统架构

iOS的系统架构和其操作系统一样，采用了分层的架构。iOS分为四层，分别为核心操作系统(the Core OS layer)，核心服务层(the Core Services layer)，媒体层(the Media layer)，Cocoa触摸架构层(the Cocoa Touch layer)。

### 1.3.1 核心操作系统层

位于iOS系统架构最下面的一层是核心操作层，它包括内存管理，文件管理，电源管理以及一些其他的操作系统业务，它可以直接和硬件设备交互。核心操作系统层包括以下这些组件：OS X Kernel，Mach 3.0，BSD，Sockets，Power Mgmt，File System Keychain，Certificates，Security，Bonjour等。

### 1.3.2 核心服务层

开发人员也可以完全访问核心服务层来方问iOS的一些服务，包括数据库连接，URL学习等等，它主要包括以下这些组件：Collections，Address Book，Networking，File Access，SQLite，Core Location，Net Services，Threading，Preferences，URL Utilities等。

SQLite是嵌入式的和轻量级的sql数据库，广泛用于iOS应用。iOS系统内置SQLite数据库减小了开发人员的负担，是系统的一部分。

Net Servicer主要负责网络服务，在开发过程中，承担应用程序与服务器之间连接的角色，有很强的交互性。

媒体层，通过它可以在应用程序中使用各种媒体文件，进行音频与视频的录制，图形的绘制，以及制作基础的动画效果。它包括以下这些组件：Core Audio，OpenGL，Audo Mixing，Audio Recording，Video Playback，JPG，PBG，TIFF，PDF，Quartz，Core Animation，OpenGL ES。最上面一层是可触摸层，这一层为应用开发提供了各种有用的框架，并且大部分与用户界面有关，本质上来说它负责用户在iOS设备上的触摸交互操作。它包括以下这些组件：Multi-Touch Events，Core Motion，Camera，View Hierarchy，Localization，Alerts，Web Views，Image Picker，Multi-Touch Controls。

在Cocoa Touch层中的很多技术都是基于Objective-C语言的。Objective-C语言为iOS提供了相机集合，文件管理，网络操作等支持。例如UIKit框架，它为应用程序提供了各种可视化组件，窗口(window)，视图(view)和按钮组件(UIButton)。Cocoa Touch层中的其他框架，对我们在应用程序中的开发来说也是非常有用的，如访问用户通信录功能框架，获取照片信息功能的框架，负责加速感应器和三维陀螺仪等硬件支持的框架等等。

### 1.3.3 应用程序框架

编写应用程序项目，都是从Cocoa Touch层开始的，具体来说是从UIKit Framework开始的。iOS常用的SDK框架如下：

AddressBook.framework提供一个用户界面，用于显示存储在地址薄中的联系人详情。

AudioUnit.framewor提供一个接口，让我们的应用程序可以对音频进行处理。

AudioToolbox.framework和AVFoundation.framework提供音频录制和回放的底层API，同时也负责管理音频硬件。

CFNetwork.framework访问和配置网络，像HTTP，FTP等。

CoreFoundation.framework提供抽象的常用数据类型，如Unicode、strings、XML、URL等。

CoreGraphics.framework提供2D绘制的基于C的API。

Foundation.framework提供Object-C的基础类(像NSObject)、基本数据类型和操作系统服务等。

GameKit.framework为游戏提供网络功能，点对点互联和游戏中的语音交流。

MapKit.framework为应用程序提供内嵌地图的接口。

MediaPlayer.framework提供视频播放和音频的功能。

OpenGLES.framework提供简洁而高效的绘制2D和3D图形的OpenGL API。

QuartzCore.framework提供动画特效以及通过硬件进行渲染的能力。

StoreKit.framework为应用程序提供在程序运行中消费的支持。

SystemConfiguration.framework检测当前网络是否可用和硬件设备状态的能力。

**1.4开发环境配置**

Xcode 7.2，整合大量框架，开发iOS程序必须的开发工具。

iOS SDK(iOS 软件开发包)，系统开发中导入的SDK。

# 2. 可行性分析

## 2.1 可行性分析

下面从四个方面进行分析系统的可行性，分别是技术条件可行，性硬件条件可行性，社会因素可行性，市场因素可行性。

2.1.1技术条件可行性

使用Objective-C编程技术

Objective-C是C语言的面向对象扩展，可以将其称为“带有对象的C”。相对于其他面向对象语言，Objective-C更加容易学习。

（1）动态绑定

函数和消息的最根本不同是：函数和它的参数是在编译的时候已经确定下来，但是消息和接收对象知道程序已经在运行，消息在传递时才能确定下来。确切的消息方法实现依赖于消息的接收者，不同的接收者可能有相同的方法名不同的方法实现，只有在消息接收时才能确定(动态类型)，通过源代码的类型定义并不能完全确认接收者的类型。运行时消息例程查找接收者和消息方法，找到消息方法实现的机器码，呼叫这个方法，传递给它消息实例变量指针。

（2）多态性

在Objective-c中消息的调用看上去和标准C中的函数调用相似，但因为方法属于一个对象，消息的行为与函数调用还是不同。在一些情况下，一些对象仅能对针对它涉及的方法起作用，不能被其他类的方法使用，即使这些类有同样名称的方法。这意味着两个对象对同样的消息有着不同的响应，即多态性。Objective-c中多态性和动态绑定配合使用，可以使代码提供给任意数量不同类型的对象使用，这些对象不需要再写代码的时候就已经定义好，它们可以是以后开发的对象，也可以是其他项目的对象。

（3）面向对象

面向对象程序构建在多种对象之上，基于Cocoa框架的程序可能使用NSMatrix对象、NSWindow对象、NSDirectonary对象、NSFont对象以及许多其它对象。在Objective-c中通过定义类来定义对象。面向对象让开发者更容易开发应用程序软件，以及更加规范化。

（4）继承性

类的定义是累进的，任一个新类都是在另一个类的基础上定义的，新类继承了它的方法和实例变量。继承使所有的类构成一颗树形的结构，在Objective-c中使用类一般要继承或间接继承于基类：NSObject，因为NSObject提供了大量的有用特性。比如，一些图形类，需要继承于Shape类，而Shape是Graphic类的子类，Graphic又继承于NSObject类，因此，还是间接继承于NSObject类。类的继承关系图2-2所示：

NSObject

Graphic

Image

Text

Shape

Line

Rectangle

Circle

Square

图2.2类的继承关系图

（5）使用工具方面：开发环境为xcode 7.2。

（6）操作系统方面：使用mac 10.11操作系统。

2.1.2硬件条件可行性

开发本系统所需的硬件环境只需要需要Mac电脑一台即可，运行环境是苹果的ios手持设备，比如iphone，ipad，itouch都是市面上的主流产品，有一定的市场占有率。该系统硬件条件可行具备开发价值。

### 2.1.3 经济可行性

和润汇海公司已经拥有比较成熟的线下业务,移动端项目开发是和润汇海公司的年度审批预算中，因此开发经费完全可以接受。和润汇海公司拥有自己的后台运维团队,硬件设施都已具备,因此无需再投入资金购买硬件设施以及组件维护团队。

由以上分析可知，基于iOS平台的小额信贷线上审批APP在经济上是可行的。

### 2.1.4 社会可行性

本系统的社会可行性主要从法律因素、用户使用可行性两方面进行研究。

(1)法律因素

本系统是根据和润汇海公司的业务流程实际情况开发研制的，是通过大量的调研得出的，系统的软件设计是在独立的环境下完成的，无可供抄袭的软件产品。

(2)用户使用可行性

本项目对用户的要求，除了需要具备在智能iOS手机上基本使用的知识外，并不需要特别的技术能力。无需对用户进行培训。

由以上分析可知，本系统具有社会可行性。

## 2.2 总体设计原则

本项目为了设计、实现和后期维护的方便，以及系统用户使用的便利，所以必须采取一定的设计原则。其主要设计原则有：

简单性：在实现项目的功能的同时，尽量让用户操作简单易懂，这对于一个APP来说是非常重要的。

针对性：本项目设计是针对和润汇海公司小额信贷业务线上申请的需求定向开发设计，所以具有专业突出和很强的针对性。

实用性：本项目能够用户线上申请,推荐客户,提款等，因此具有良好的实用性。

一致性：界面整体设计风格以及命名规则的一致性：整体界面布局和用图用色风格及变量、类名和其他元素的命名规则保持一致。功能一致性：完成同样的功能应该尽量使用同样的元素。 元素风格一致性：界面元素的美观风格、摆放位置在同一个界面和不同界面之间都应该是一致的。变量命名规则的一致性：变量应该用统一的规则进行命名，做到任意变量均能从上下文推断其义。

# 3. 系统设计

## 3.1 系统功能分析

本系统主要具有如下功能：

（1）创建公司。

用户打开软件通过点击添加公司，可以进入添加公司模块，业务经理可通过自己的需求创建公司。

（2）二维码。

该模块客户经理可获得已创建公司的列表，并可生成对应的二维码，供客户端扫描，绑定次业务经理。

（3）消息。

消息模块用户查看授信助手，新增用户，催款通知，系统公告等实时更新消息。

（4）客户。

该模块业务经理可查看与自己绑定的所有客户信息，也可查看所有已创建的所有公司的详细信息。

（5）设置。

设置模块，可以意见反馈，评价，重置密码，退出登录功能。

## 3.2模块的设计

本设计开发一款应用于iOS操作系统的信贷线上审批系统，主要模块功能为，当用户下载并安装本软件后，选择或点击系统桌面上的该应用图标，接着展现的是欢迎进入的界面，之后来到主界面。主页分为四个模块，以及设置模块入口；通过点击添加公司，可以进入添加公司模块，业务经理可通过自己的需求创建公司；通过点击二维码，可获得已创建公司的列表，并可生成对应的二维码，供客户端扫描，绑定次业务经理；客户模块业务经理可查看与自己绑定的所有客户信息，也可查看所有已创建的所有公司的详细信息。消息模块用户查看授信助手，新增用户，催款通知，系统公告等实时更新消息；设置模块，可以意见反馈，评价，重置密码，退出登录功能。

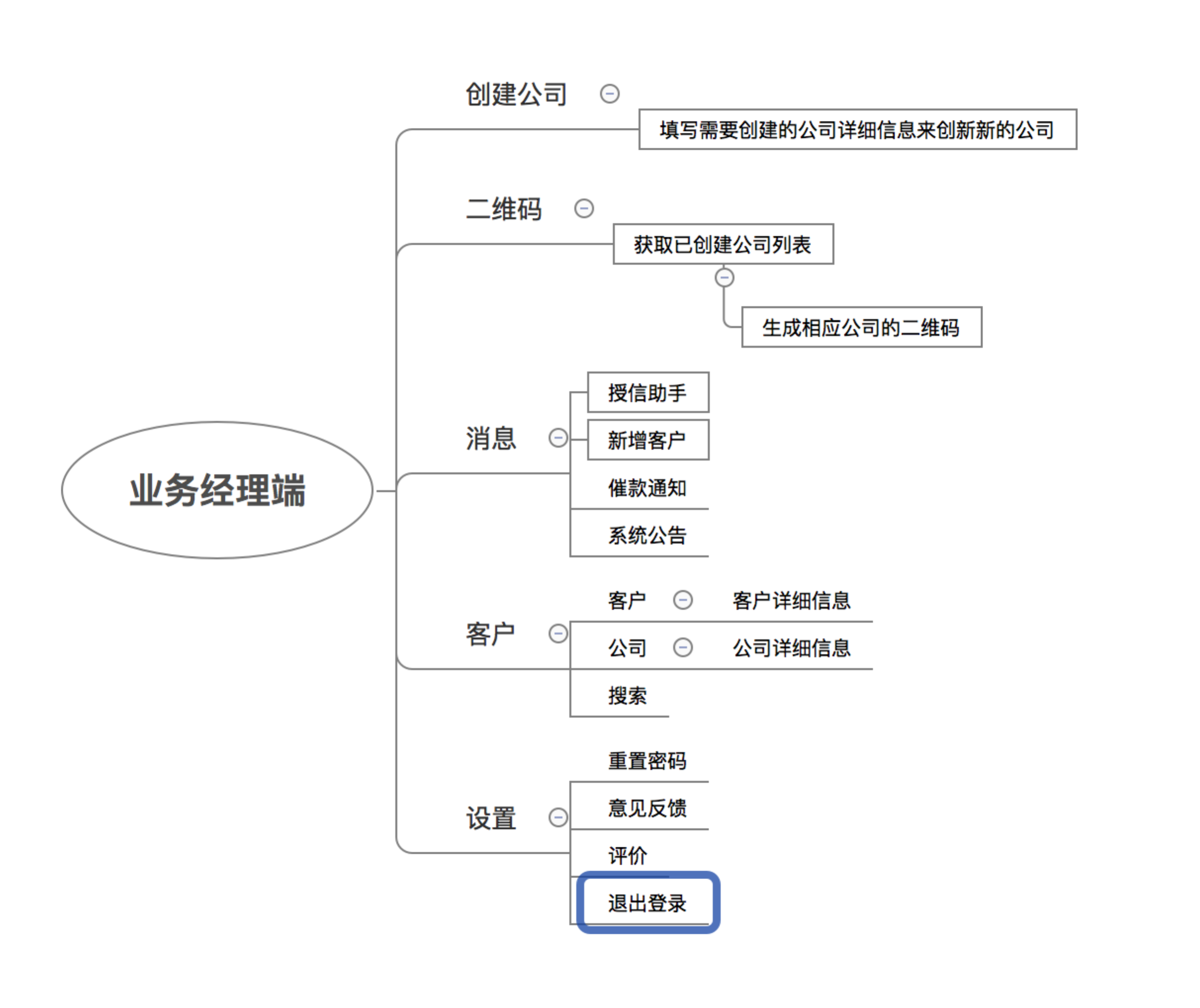


图3.1 系统模块图

# 4. 系统实现

## 4.1 首页模块

### 4.1.1 首页界面

此模块主要功能是显示本系统的主要功能模块，当用户点击了各个模块的时候，先判断当前APP用户是否已经登录，如果没有登录，则弹出次登录界面，用户通过手机号码和密码进行登录，用户可以也可点击注册按钮进行注册，如果用户忘记了自己的密码也可点击忘记密码按钮进行找回密码操作。登录成功后如果此客户经理的还未通过后台审核，则会弹出提示框提醒此客户经理耐心等待审核，只有审核公共后才能正常使用。如图4.1、4.2所示：



图4.1 登录界面

- (void)installSubView

{

UIImageView \* bgImageView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"homeTypeBg"]];

bgImageView.frame = self.view.bounds;

[self.view addSubview:bgImageView];

bgImageView.userInteractionEnabled = YES;

NSArray \* titleArr = @[@"创建公司", @"二维码", @"消息", @"客户"];

NSArray \* titleCol = @[@"#E35D39", @"#82CF97", @"#1FB0EB", @"#F6BC30"];

CGFloat typeW = kScreenWidthScale(120);

CGFloat mag = (kScreenWidth - 2 \* kScreenWidthScale(120))/3;

for (int i = 0; i < titleArr.count; i++) {

//列号

int columnNum = i%2;

//行号

int lineNum = i/2;

UIView \* type = [[UIView alloc] initWithFrame:CGRectMake(mag\* (columnNum + 1) + typeW \* columnNum, kScreenHeightScale(100) + mag\* lineNum + typeW \* lineNum, typeW, typeW)];

type.backgroundColor = [UIColor whiteColor];

type.tag = 100 + i;

type.layer.cornerRadius = 5;

[bgImageView addSubview:type];

UIImageView \* typeImage = [[UIImageView alloc] initWithImage:[UIImage imageNamed:[NSString stringWithFormat:@"homeType%d", i]]];

[type addSubview:typeImage];

typeImage.contentMode = UIViewContentModeCenter;

typeImage.frame = CGRectMake(0, kScreenWidthScale(30), typeW, kScreenWidthScale(30));

UILabel \* titleLab = [[UILabel alloc] initWithFrame:CGRectMake(0, CGRectGetMaxY(typeImage.frame) + 10, typeW, kScreenHeightScale(25))];

titleLab.text = titleArr[i];

titleLab.font = [UIFont systemFontOfSize:18];

titleLab.textColor = [UIColor colorWithHexString:titleCol[i]];

titleLab.textAlignment = NSTextAlignmentCenter;

[type addSubview:titleLab];

UITapGestureRecognizer \* typeTap = [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(clickType:)];

[type addGestureRecognizer:typeTap];

}

}

-(void)checkLoginStatus:(UIViewController \*)currentController businessBlock:(void(^)())block{

if ([[HNSingleton defaultSingleten]isLogin]) {

if ([[HNSingleton defaultSingleten].userLogin.appState isEqualToString:@"02"]) {

if (block) {

block();

}

}else{

[GCDNotificationView initDiscreetNotificationView:@"您的资料暂时还未通过审核,请联系管理员"];

}

}else{

[self presentLoginPrompt:currentController];

}

}

-(void)presentLoginPrompt:(UIViewController \*)currentController{

LoginViewController \*ctl = [[LoginViewController alloc]init];

CustomNavigationController \*nav = [[CustomNavigationController alloc]initWithRootViewController:ctl];

// nav.navigationBarHidden = YES;

[currentController presentViewController:nav animated:YES completion:^{

}];

}

### 4.1.2 登录、注册以及注册完善信息界面

如果客户经理已经存在账号，则可以直接在登录界面进行登录操作，如果还未有账号，则进入注册界面进行注册，注册后调到完善信息界面，此时客户经理需要填写姓名和身份证号码以供后台审核时候为本公司的业务经理。

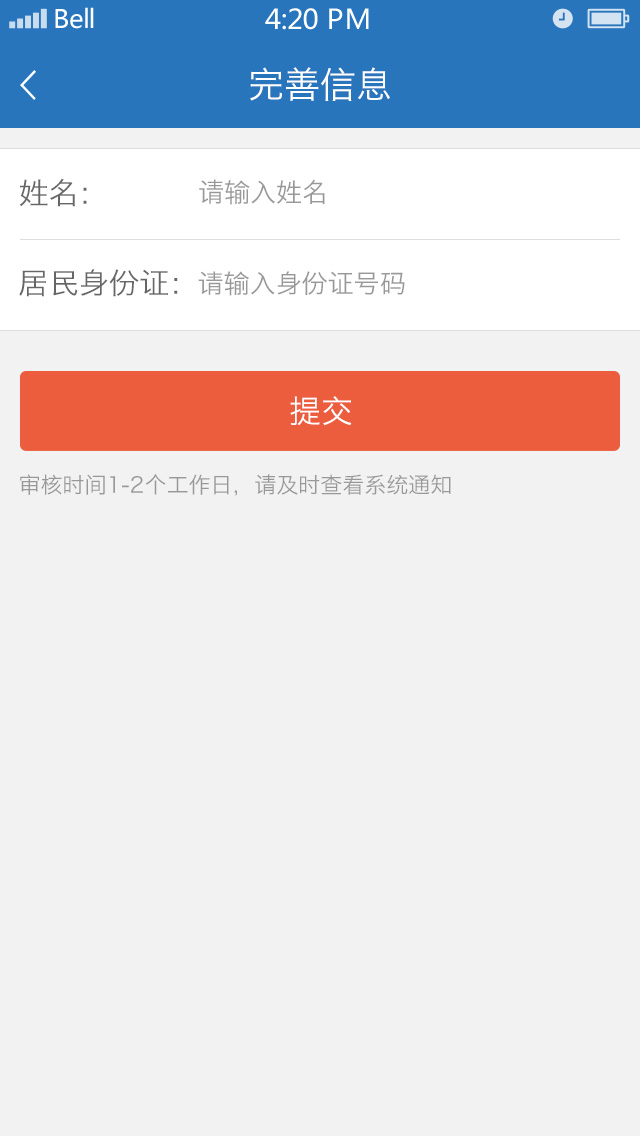
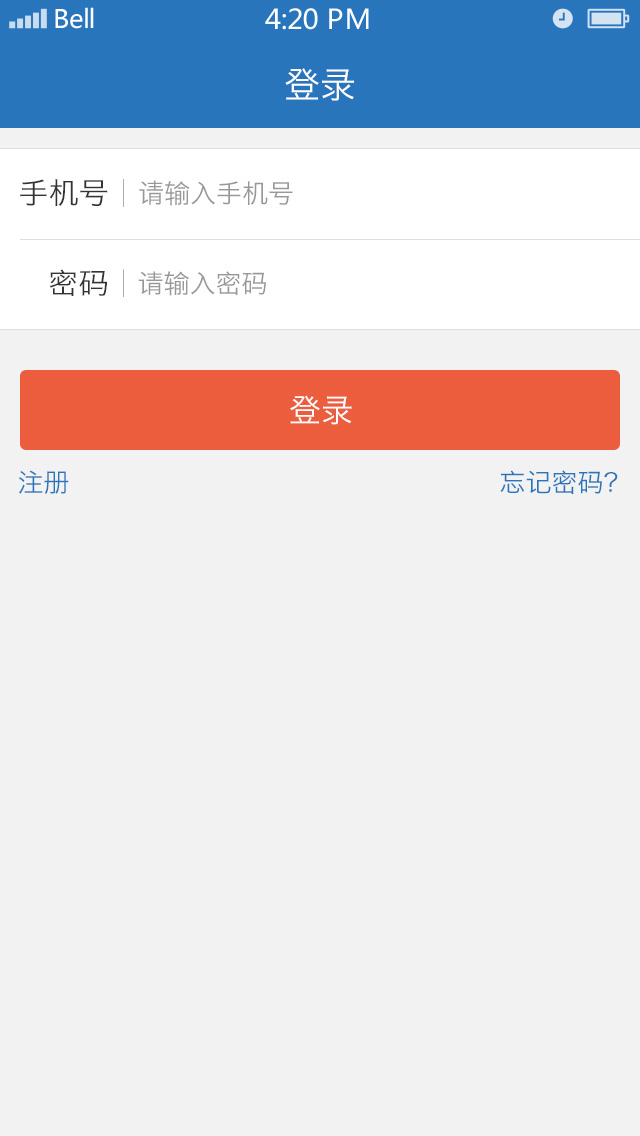


图4.2 登录、注册以及注册完善资料界面

/\*\*

\* 登录成功

\*/

- (void)loginSucces

{

NSNotificationCenter \* notifi = [NSNotificationCenter defaultCenter];

[notifi postNotificationName:kLoginSuccess object:nil];

[self dismissViewControllerAnimated:YES completion:^{

[notifi removeObserver:self];

}];

}

/\*\*

\* 点击了注册按钮

\*/

- (void)clickRegisterBtn{

RegisterViewController \* registerView = [[RegisterViewController alloc] init];

[self.navigationController pushViewController:registerView animated:YES];

}

/\*\*

\* 点击了忘记密码

\*/

- (void)clickForgetBtn

{

ResetPasswordController\* resetPassword = [[ResetPasswordController alloc] init];

[self.navigationController pushViewController:resetPassword animated:YES];

}

（2）程序流程

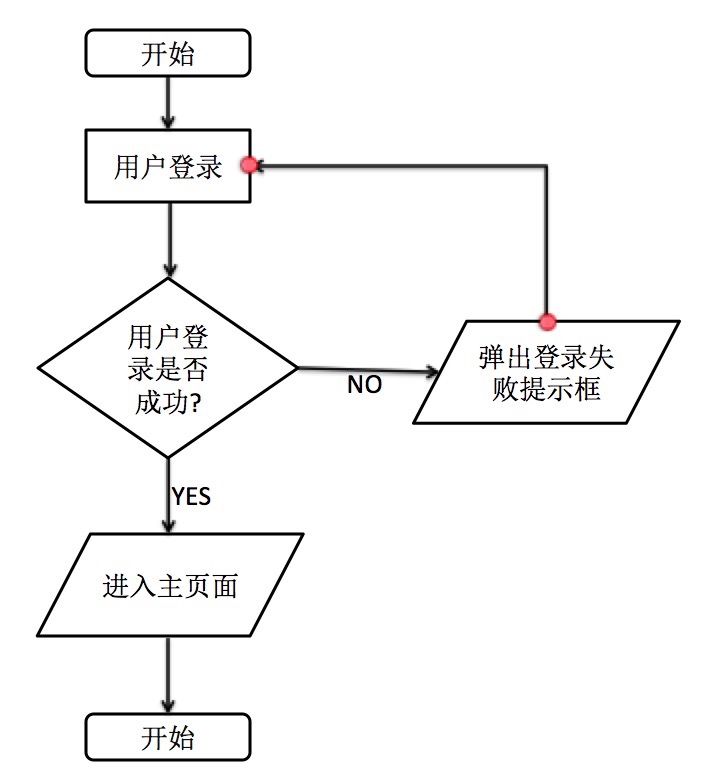


图4.3 登录模块程序流程图

## 4.2 创建公司模块

### 4.2.1 创建公司界面

此界面主要主要是为客户经理创建公司，主要是用了系统原声的输入框以及自定义的选择空间，公司类型、成立时间、发薪日都是不可输入的，只可根据提供的进行选择，工牌照片以及公司门头照则是需要调用系统的图库或者系统自带相机进行选择或者拍摄。如图4.4所示：



图4.4 创建公司界面

- (IBAction)selectCompanyType:(UIButton \*)sender

{

[self resignFirstResponse];

[self cancelLocatePicker];

if (kScreenHeight<568) {

self.superview.frame=CGRectMake(0, -43\*2, kScreenHeight,self.superview.frame.size.height);

[UIView animateWithDuration:0.3 animations:^{

self.superview.frame=CGRectMake(0, -43, kScreenHeight,self.superview.frame.size.height);

}];

}

if (!self.locatePicker) {

WDPickerView \* locatePicker = [[WDPickerView alloc] initWithStyle:WDPickerWithSelectNomalType delegate:self];

self.locatePicker = locatePicker;

self.currentSelectBtn = sender;

if (sender.tag == 1) {//公司类型

locatePicker.nomalArray = @[@"国企",@"外资(欧美)",@"民营公司",@"外资(非欧美)",@"创业公司",@"政府机关",@"事业单位"];

}else if (sender.tag == 2){//发薪日

NSMutableArray \* dayArr = [NSMutableArray array];

for (int i = 1; i < 32; i++) {

[dayArr addObject:[NSString stringWithFormat:@"%d", i]];

}

locatePicker.nomalArray = dayArr;

}

}

[self.locatePicker showInView:self.superview];

}

-(void)cancelLocatePicker

{

[self cancelDatePicker];

[self.locatePicker cancelPicker];

self.locatePicker.delegate = nil;

self.locatePicker = nil;

[self.locatePicker removeFromSuperview];

if (kScreenHeight<568) {

[UIView animateWithDuration:0.3 animations:^{

self.superview.frame=CGRectMake(0, 0, kScreenHeight,self.superview.frame.size.height);

}];

}

}

#pragma mark -- WDPickerView代理方法

- (void)pickerDidCheckCity:(WDPickerView \*)picker

{

[self cancelLocatePicker];

if (self.currentSelectBtn.tag == 1) {//公司类型

self.companyType.text = picker.locate.nomalString;

}else if (self.currentSelectBtn.tag == 2){//发薪日

self.payDay.text = picker.locate.nomalString;

}

}

## 4.3 二维码模块

### 4.3.1 选择公司二维码界面

从主页点击二维码后，进入选择公司二维码，当进入时从服务器请求当前已审核通过可供生成二维码的公司列表，并通过UITableView显示出来。如图4.5所示：



图4.5 选择公司二维码界面

- (void)layoutSelfView

{

\_tableView = [[UITableView alloc] initWithFrame:(CGRect){0,0,kScreenWidth,kContentViewHeight+49} style:UITableViewStylePlain];

\_tableView.backgroundView = nil;

\_tableView.backgroundColor = [UIColor clearColor];

// \_tableView.separatorStyle = UITableViewCellSelectionStyleNone;

\_tableView.delegate = self;

\_tableView.dataSource = self;

[self.view addSubview:\_tableView];

[self requestCommonProblemWithPageNum:@"1" withRefreshType:RefreshTypeHeader];

self.currPageNum = @"1";

// 下拉刷新

\_tableView.mj\_header= [MJRefreshNormalHeader headerWithRefreshingBlock:^{

[self requestCommonProblemWithPageNum:@"1" withRefreshType:RefreshTypeHeader];

self.currPageNum = @"1";

}];

// 上拉刷新

\_tableView.mj\_footer = [MJRefreshBackNormalFooter footerWithRefreshingBlock:^{

NSString \* newPange = [NSString stringWithFormat:@"%d", [self.currPageNum intValue] + 1];

[self requestCommonProblemWithPageNum:newPange withRefreshType:RefreshTypeFooter];

self.currPageNum = newPange;

}];

}

- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section

{

return \_tableArr.count;

}

- (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

{

static NSString \* ID = @"ChooseCompanyQrCodeController";

UITableViewCell \* cell;

if (cell == nil) {

cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:ID];

cell.textLabel.font = [UIFont systemFontOfSize:14];

cell.textLabel.textColor = [UIColor colorWithHexString:@"#333333"];

cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;

}

QrCompany \* company = \_tableArr[indexPath.row];

cell.textLabel.text = [NSString stringWithFormat:@"%@", company.companyName];

return cell;

}

- (void)tableView:(UITableView \*)tableView didSelectRowAtIndexPath:(NSIndexPath \*)indexPath

{

[tableView deselectRowAtIndexPath:indexPath animated:YES];

QrCodeViewController \* qrVc = [[QrCodeViewController alloc] init];

qrVc.company = \_tableArr[indexPath.row];

[self.navigationController pushViewController:qrVc animated:YES];

}

### 4.3.2 动态二维码界面

在选择公司二维码界面选择相应的公司后进入动态二维码界面，使用ZXing生成包含客户经理id、公司id、以及时间戳的二维码。并显示出来，客户端扫面次二维码就可以绑定对应的公司以及客户经理。如图4.6所示：



图4.6 动态二维码界面

- (void)reloadDataWithCompanyDetail:(QrCompanyDetail \*)companyDetail{

self.nameLab.text = companyDetail.custName;

self.companyAddress.text = companyDetail.companyAddress;

self.companyName.text = companyDetail.companyName;

[self.iconImageView sd\_setImageWithURL:[NSURL URLWithString:kWDUrl(companyDetail.picPath)] placeholderImage:[UIImage imageNamed:@"icon"]];

NSString \*curTime = [[NSString alloc] initWithFormat:@"%f", [[NSDate date] timeIntervalSince1970]\*1000];

NSString \* qrStr = [NSString stringWithFormat:@"%@;%@;%@", companyDetail.companyId, [HNSingleton defaultSingleten].userLogin.custID, [curTime substringToIndex:13]];

self.qrImageView.image = [UIImage qrImageWithContent:qrStr logo:[UIImage imageNamed:@"4444"] size:self.qrImageView.frame.size.width red:20 green:100 blue:100];}

## 4.4 客户模块

### 4.4.1 客户界面

当从主页点击客户按钮后进入客户模块，进入客户模块后同事请求此客户经理对应的客户客户列表以及已经创建的公司列表，然后使用UITableView显示，对cell进行重用，提高系统性能。如图4.7所示：



图4.7 客户界面

+ (CustomerTableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

{

[tableView registerNib:[UINib nibWithNibName:@"CustomerTableViewCell" bundle:nil] forCellReuseIdentifier:@"CustomerTableViewCell"];

CustomerTableViewCell \* cell = [tableView dequeueReusableCellWithIdentifier:@"CustomerTableViewCell"];

return cell;

}

- (void)setCustomer:(Customer \*)customer

{

\_customer = customer;

self.cusName.text = customer.custName;

self.cusPhoneNum.text = customer.phoneNum;

self.loanNum.text = [NSString stringWithFormat:@"发生借款%@笔", customer.borrNumb];

self.loanMoneyNum.text = [NSString stringWithFormat:@"共: %@元", customer.totalNumb];

}

### 4.4.2 授信资料及公司详情界面

在客户界面点击客户所在行就会进入授信资料界面，在公司界面选中某个公司后就会进入公司详情界面，这两个界面都是单纯的对请求回来的数据进行展示作用，根据界面中的图片数量开启一个或者多个异步子线程请求下载图片，当下载完成后再回到主线程刷新对应的UIImageView来显示下载的图片。如图4.8所示：

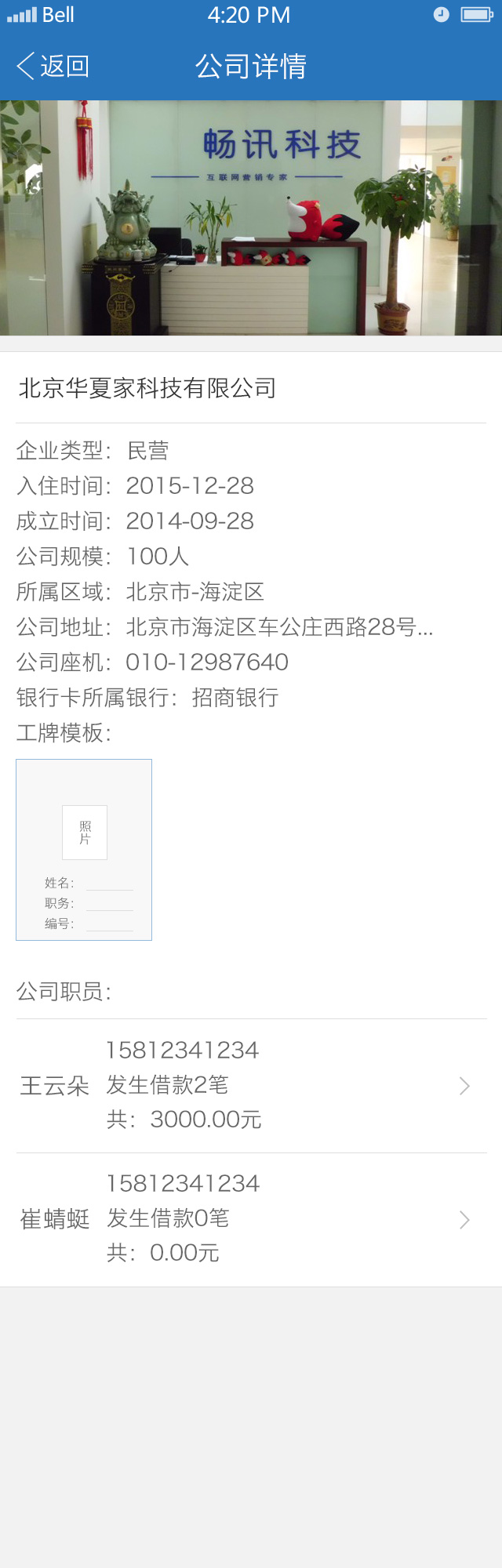


图4.8 授信资料及公司详情界面

- (void)sd\_setImageWithURL:(NSURL \*)url placeholderImage:(UIImage \*)placeholder options:(SDWebImageOptions)options progress:(SDWebImageDownloaderProgressBlock)progressBlock completed:(SDWebImageCompletionBlock)completedBlock {

[self sd\_cancelCurrentImageLoad];

objc\_setAssociatedObject(self, &imageURLKey, url, OBJC\_ASSOCIATION\_RETAIN\_NONATOMIC);

if (!(options & SDWebImageDelayPlaceholder)) {

dispatch\_main\_async\_safe(^{

self.image = placeholder;

});

}

if (url) {

\_\_weak \_\_typeof(self)wself = self;

id <SDWebImageOperation> operation = [SDWebImageManager.sharedManager downloadImageWithURL:url options:options progress:progressBlock completed:^(UIImage \*image, NSError \*error, SDImageCacheType cacheType, BOOL finished, NSURL \*imageURL) {

if (!wself) return;

dispatch\_main\_sync\_safe(^{

if (!wself) return;

if (image) {

wself.image = image;

[wself setNeedsLayout];

} else {

if ((options & SDWebImageDelayPlaceholder)) {

wself.image = placeholder;

[wself setNeedsLayout];

}

}

if (completedBlock && finished) {

completedBlock(image, error, cacheType, url);

}

});

}];

[self sd\_setImageLoadOperation:operation forKey:@"UIImageViewImageLoad"];

} else {

dispatch\_main\_async\_safe(^{

NSError \*error = [NSError errorWithDomain:SDWebImageErrorDomain code:-1 userInfo:@{NSLocalizedDescriptionKey : @"Trying to load a nil url"}];

if (completedBlock) {

completedBlock(nil, error, SDImageCacheTypeNone, url);

}

});

}

}

## 4.5 消息模块

### 4.5.1 消息界面

点击主页的消息按钮进入此界面后会先读取消息数据库中的内容，消息数据库中的内容是手机接到收到的推送通知后存储。如图4.9所示：



图4.9 消息界面

/\*\* 从数据库中读取历史推送消息 \*/

- (void)loadTuiSong {

FMDatabase \* db = [FMDatabase databaseWithPath:tuisongDataBasePath];

if ([db open]) {// 1.执行查询语句

FMResultSet \*resultSet = [db executeQuery:@"SELECT \* FROM tuisong"];

while ([resultSet next]) {// 2.遍历结果

NSString \* messType = kIsEmptyString([resultSet stringForColumn:@"MESSTYPE"]);

NSString \* messContext = kIsEmptyString([resultSet stringForColumn:@"MESSCONTEXT"]);

NSString \* messDate = kIsEmptyString([resultSet stringForColumn:@"MESSDATE"]);

NSString \* messState = kIsEmptyString([resultSet stringForColumn:@"MESSSTATE"]);

if ([messType isEqualToString:@"0001"]) {//授信小助手

NSDictionary \* dic = [NSDictionary dictionaryWithObjectsAndKeys:messType, @"MESSTYPE",messContext, @"MESSCONTEXT",

messDate, @"MESSDATE",

messState, @"MESSSTATE",

@"messageIcon0", @"ICONNAME",

self.titleArray[0], @"CELLNAME",

nil];

MessageModel \* model = [MessageModel messageModelWithDict:dic];

[self.tableViewArray replaceObjectAtIndex:0 withObject:model];

}else if([messType isEqualToString:@"0002"]){//新增客户

NSDictionary \* dic = [NSDictionary dictionaryWithObjectsAndKeys:messType, @"MESSTYPE",messContext, @"MESSCONTEXT",

messDate, @"MESSDATE",

messState, @"MESSSTATE",

@"messageIcon1", @"ICONNAME",

self.titleArray[1], @"CELLNAME",

nil];

MessageModel \* model = [MessageModel messageModelWithDict:dic];

[self.tableViewArray replaceObjectAtIndex:1 withObject:model];

}else if([messType isEqualToString:@"0003"]){//催款通知

NSDictionary \* dic = [NSDictionary dictionaryWithObjectsAndKeys:messType, @"MESSTYPE",messContext, @"MESSCONTEXT",

messDate, @"MESSDATE",

messState, @"MESSSTATE",

@"messageIcon2", @"ICONNAME",

self.titleArray[2], @"CELLNAME",

nil];

MessageModel \* model = [MessageModel messageModelWithDict:dic];

[self.tableViewArray replaceObjectAtIndex:2 withObject:model];

}else if([messType isEqualToString:@"0004"]){//系统公告

NSDictionary \* dic = [NSDictionary dictionaryWithObjectsAndKeys:messType, @"MESSTYPE",messContext, @"MESSCONTEXT",

messDate, @"MESSDATE",

messState, @"MESSSTATE",

@"messageIcon3", @"ICONNAME",

self.titleArray[3], @"CELLNAME",

nil];

MessageModel \* model = [MessageModel messageModelWithDict:dic];

[self.tableViewArray replaceObjectAtIndex:3 withObject:model];

}

}

}

self.messageView.tableViewArray = self.tableViewArray;

}

### 4.5.2 授信助手界面

已经绑定的客户提交授信申请后，次界面就会收到一条推送消息，提醒客户经理填写建议额度，当点击建议额度按钮后进入到建议额度界面，在建议额度界面客户可点击授信资料富文本查看此用户的授信资料，以此作为客户经理提交建议额度的依据。如图4.10所示：



图4.10授信助手及建议额度界面

//计算点击是否在link range内

- (BOOL)calculateTouchesRange:(CGPoint)location

{

//获取触摸点字符位置

location.y -= \_sapcingTop;

NSUInteger touchedChar = [\_layoutManager glyphIndexForPoint:location inTextContainer:\_textContainer];

for (int i = 0; i < self.linkArray.count; i++) {

NSArray \*rangeArray = [self.linkArray[i] allKeys];

NSString \*rangeString = [rangeArray firstObject];

NSRange range = NSRangeFromString(rangeString);

if (touchedChar>=range.location && touchedChar<=range.location+range.length) {

NSArray \*valueArray = [self.linkArray[i] allValues];

\_index = i;

WYLabelLinkStyle linkStyle = WYLabelLinkStyleALL;

if (i<\_webArray.count&&i>-1) {//点击的是网址

linkStyle = WYLabelLinkStyleWeb;

}else if (i<\_phoneArray.count+\_webArray.count&&i>-1) {//点击的是手机号码

linkStyle = WYLabelLinkStyleIphoneNumber;

}else if (i>-1)//点击用户自定义文字

{

linkStyle = WYLabelLinkStyleUserText;

}

NSString \*valueString = [valueArray firstObject];

if ([self.delegate respondsToSelector:@selector(labelLinkClickedWithLabel:withCilckedText: withLinkStyle:)]) {

[\_delegate labelLinkClickedWithLabel:self withCilckedText:valueString withLinkStyle:linkStyle];

}

return YES;

}

}

return NO;

}

### 4.5.3 新增客户界面

当客户提款成功成功后就会在新增客户中显示出来，点击客户就会查看客户详情。如图4.12所示：



图4.12 新增客户界面

- (instancetype)initWithDict:(NSDictionary \*)dict

{

if (self = [super init]) {

NSDictionary \* mainDict = dict[@"main\_data"][@"data"][0];

self.custId = kIsEmptyString(mainDict[@"CUSTID"]);

self.cusName = kIsEmptyString(mainDict[@"CUSTNAME"]);

self.phoneNum = kIsEmptyString(mainDict[@"PHONENO"]);

self.custSex = kIsEmptyString(mainDict[@"CUSTSEX"]);

self.certId = kIsEmptyString(mainDict[@"CERTID"]);

self.withdrLimit = kIsEmptyString(mainDict[@"WITHDRLIMIT"]);

self.autidId = kIsEmptyString(mainDict[@"AUDITID"]);

NSArray \* histInfoArr = dict[@"histInfo"][@"data"];

NSMutableArray \* obj = [NSMutableArray array];

for (NSDictionary \* dic in histInfoArr) {

AddCustomerWithdrawDetail \* withDe = [AddCustomerWithdrawDetail addedCustomerWithdrawDetailWithDict:dic];

[obj addObject:withDe];

}

self.withdrawArr = obj;

}

return self;

}

### 4.5.3 催款助手及系统公告界面

当已绑定客户有逾期的时候就会收到催款提醒，点击催款可以直接拨打客户的手机，提醒客户还款；系统公告是显示和润汇海公司的对客户经理的规章制度或者业务变更信息。如图4.13所示：

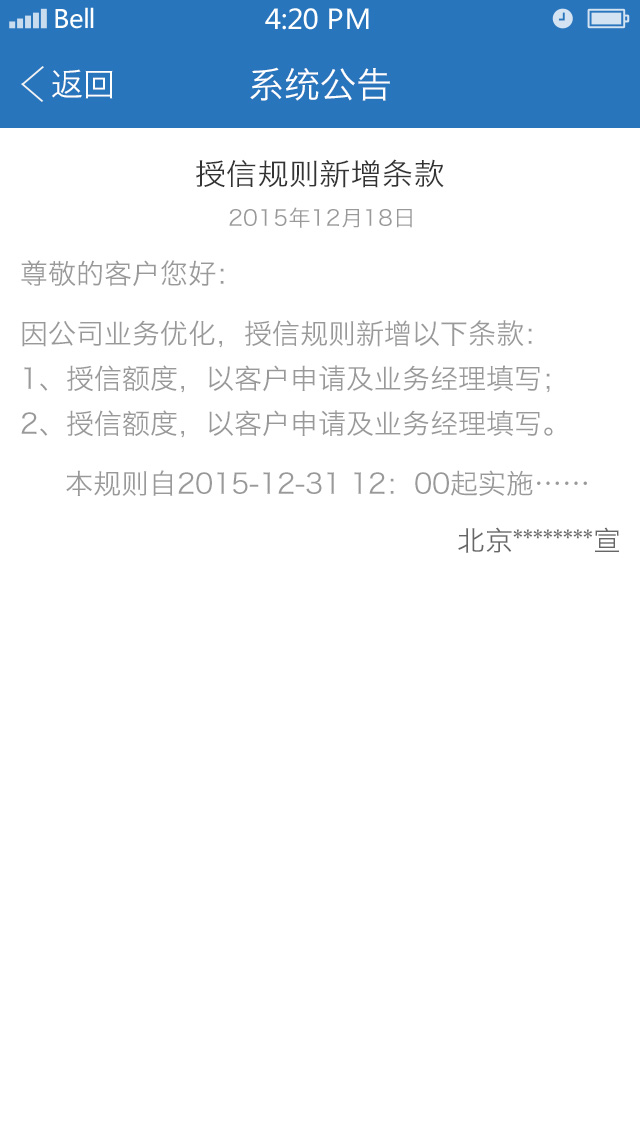


图4.13 催款助手及系统公告界面

- (void)tableView:(UITableView \*)tableView didSelectRowAtIndexPath:(NSIndexPath \*)indexPath

{

[tableView deselectRowAtIndexPath:indexPath animated:YES];

self.customer = \_tableArr[indexPath.row];

UIAlertView \* alert = [[UIAlertView alloc] initWithTitle:nil message:self.customer.phoneNum delegate:self cancelButtonTitle:@"取消" otherButtonTitles:@"呼叫", nil];

[alert show];

}

- (void)alertView:(UIAlertView \*)alertView clickedButtonAtIndex:(NSInteger)buttonIndex

{

if (buttonIndex == 1) {

[[UIApplication sharedApplication] openURL:[NSURL URLWithString:[NSString stringWithFormat:@"tel://%@", self.customer.phoneNum]]];

}

}

- (void)viewDidLoad {

[super viewDidLoad];

self.navigationItem.title = @"系统公告";

UIWebView \* webView = [[UIWebView alloc] initWithFrame:self.view.bounds];

[self.view addSubview:webView];

webView.opaque = NO; //不设置这个值 页面背景始终是白色

webView.backgroundColor = [UIColor clearColor];

webView.delegate = self;

webView.scalesPageToFit = YES;

[webView loadHTMLString:self.htmlStr baseURL:nil];

}

- (void)webViewDidFinishLoad:(UIWebView \*)webView

{

//字体大小

[webView stringByEvaluatingJavaScriptFromString:@"document.getElementsByTagName('body')[0].style.webkitTextSizeAdjust= '350%'"];

}

## 4.6 设置模块

### 4.6.1 设置界面

设置模块主要有意见反馈、评价、重置密码、退出登录等功能，评价是直接跳转到AppStore对此APP进行评价。如图4.14所示：



图4.14 设置界面

- (void)setupGroup0

{

SettingItem \*feedback = [SettingArrowItem itemWithIcon:@"feedback" title:@"意见反馈" destVcClass:[FeedBackController class]];

SettingItem \*evaluation = [SettingArrowItem itemWithIcon:@"evaluation" title:@"评价" destVcClass:nil];

evaluation.option = ^{

[[UIApplication sharedApplication]openURL:[NSURL URLWithString:@"itms-apps://itunes.apple.com/WebObjects/MZStore.woa/wa/viewContentsUserReviews?id=958489515&pageNumber=0&sortOrdering=2&type=Purple+Software&mt=8"]];

};

// SettingItem \*resetGesture = [SettingArrowItem itemWithIcon:@"resetGesture" title:@"修改手势密码" destVcClass:[UIViewController class]];

SettingItem \*reset = [SettingArrowItem itemWithIcon:@"reset" title:@"重置密码" destVcClass:[ResetPasswordController class]];

SettingGroup \*group = [[SettingGroup alloc] init];

group.items = @[feedback, evaluation, reset];

[self.data addObject:group];

}

结束语

本次毕业设计到此已经顺利结束，通过这次的毕业设计,我学到了很多。在4年时间内我学习了很多的课程及编程语言，虽然通过学习对这些课程有所了解和掌握，但是始终处于理论知识的掌握阶段，并没有真正应用于实践设计开发，所以对知识的掌握还不够全面和熟练。而且并不能真正独立的开发出一个完整的应用于实际的系统。通过这次的毕业设计，我已经可以完成这一任务，对自身的编程能力也有了很大的提高。本课题由于采用的是纯软件开发系统，采用的系统又是很封闭的IOS系统而且对界面要求和内存管理比较严格，故我有如下总结：

（1）UI界面要符合iphone一贯的风格，动画要流畅，对于怎么设计可以随意发挥自己想象。

（2）内存管理要注意，自己创建的对象要自己管理好自己的内存，不要造成内存过度释放或者内存泄漏而造成程序崩溃。

（3）注意设计的算法，一本小说很长需要分割字符串，要确定怎么样分割，然后要计算满屏的字数，和下一页的，这样需要谨慎考虑和设计。

（4）注意oc语言的风格和程序的美观性。

（5）要熟练掌握Xib设计控件和代码写控件。

（6）要有严谨的逻辑思维能力和不断尝试的创新能力。

总体说来，在这次长达一个学期的毕业设计中，我真正学到了很多有用的东西，是一些在课堂内所学不到的知识，收获很大。

# 参考文献

1. 王志刚，《深入浅出objective－c 2.0》,电子工业出版社，2011。
2. Dave Mark，Jack Nutting，Jeff LaMarche（著），漆振，杨越，孙文磊（译）《iphone4与ipad开发基础教程》，人民邮电出版社，2011。
3. Erica Sadun（著），张彩霞，高颖，易磊（译），《iphone开发秘籍（第2版）》,人民邮电出版社，2011。
4. 史嘉权，《数据库与系统理论》，清华大学出版社，2005。
5. 苹果开发者官方网站<https://developer.apple.com>，和Coco China 苹果开发中文站<http://www.cocoachina.com/>。
6. （美）丹皮内等著，《深入浅出iphone开发》，人民邮电出版社，2011。
7. 张英烽，刘超，《iphone应用程序开发指南》，山东科学技术出版社，2011。
8. (美)戴利，《iphone游戏开发实践指南》，机械工业出版社，2011.
9. Jiva DeVoe，《好学的Objective－c》，人民邮电出版社，2011。
10. (美)Carlo Chung（著），刘威（译），《Objective-C编程之道：iOS设计模式解析》，人民邮电出版社，2011。
11. (美)詹姆斯 著，彭舰 译，《Cocoa入门－使用Objective-C》，人民邮电出版社，2011
12. （美）克莱尔 著，李强 译，《Objective-C 2.0 Mac和iOS开发实践指南》，中国电力大学出版社，2011。
13. 杨正洪 著，《Objective-C程序设计》，清华大学出版社，2011。
14. （美）雷 著，袁国忠，郝记生 译，《iphone开发入门经典（第2版）》，人民邮电出版社，2011。
15. （美）坎贝尔 著，刘红伟 译，《iphone SDK3 开发快速上手》，机械工业出版社，2011。
16. 张英蜂，刘超，《iphone应用程序开发指南（基础篇）》，山东科学技术出版社，2011。
17. （美）诺伊贝格 著，《IOS4 编程》，山东大学出版社，2011。

# 致 谢

经过几个月的努力，作者终于顺利的完成了基于IOS系统的小额信贷审批APP的开发设计。在完成本项目开发设计的整个过程中，作者得到了计算机学院多位老师的帮助，特别是李树榜老师更是提出了许多宝贵的意见和建议，并指出了许多不足之处，使系统功能更加完善合理，在此表示特别感谢。

另外，特别感谢11级的同学，他们给予我特别大的帮助，帮我解决了不少的难点问题，使系统能及时开发完成。