

## 摘要

为了更好的、安全的使银行全面了解企业的各类金融相关信息，需要构建相应金融数据分析软件，可将不同来源发布的各类金融信息整合后存放于数据库中，并根据相关数据和指标自动分析。其中包括用户管理子系统、融报管理子系统、商机管理子系统等。基于上述目的，本课题将实现基于 iOS 的金融数据分析软件中用户管理子系统的相关功能。

金融数据分析软件用户管理子系统可分为五个模块，分别为用户权限、好友管理、情报管理、用户评论和积分功能。系统设计采用 MVC 三层架构，客户端与服务器之间通过 HTTP 协议进行通信，服务器端根据相应接口响应客户端请求，对数据库中数据进行写入、修改、查找、删除等相关操作。

本系统功能现已基本实现，每部分模块也进行了局部的分析及整体的测试，基本满足了系统需求。

**关键词：**金融；用户管理；iOS；HTTP

## Abstract

In order to better and safely make banks fully understand all kinds of financial related information of enterprises, it is necessary to build corresponding financial data analysis software, which can integrate all kinds of financial information released from different sources and store them in the database, and automatically analyze them according to relevant data and indicators. The financial data analysis software including user management subsystem, financial report management subsystem and business opportunity management subsystem. Based on the above purpose, this topic will realize the related functions of the user management subsystem in the financial data analysis software based on the iOS.

The user management subsystem implemented in this topic can be divided into five modules, which are user rights, friend management, information management, user comments and integral functions. The system adopts MVC architecture. The client communicates with the server through HTTP protocol. According to the corresponding interface ,server responds to the client's request and writes, modifies, searches and deletes the data in the database.

The functions of this system can be basically achieved and every part of it has been partly analyzed and completely tested, which basically meets the system requirements.

**Keywords:** Finance; User Management; iOS; HTTP

## 目录

摘要.....	I
Abstract.....	II
目录.....	III
第一章 绪论.....	1
第一节 相关技术的发展现状.....	1
一、移动设备操作系统.....	1
二、相关软件.....	2
三、HTTP 通信协议.....	2
第二节 课题内容与开发环境.....	3
一、关于课题.....	3
二、开发环境简介.....	3
三、本文的内容和组织结构.....	4
第二章 基于 iOS 的金融数据分析软件用户管理子系统的需求分析.....	6
第一节 任务概述.....	6
第二节 开发模型介绍.....	6
第三节 需求分析.....	7
一、用户用例图.....	7
二、功能描述.....	7
第三章 基于 iOS 的金融数据分析软件用户管理子系统的整体结构.....	9
第一节 系统整体结构分析.....	9
第二节 各模块处理流程分析.....	11
一、处理流程示意图.....	11
二、处理流程说明表.....	11
第四章 基于 iOS 的金融数据分析软件用户管理子系统的概要设计.....	14
第一节 数据表结构设计.....	14
一、用户表.....	14
二、好友管理表.....	14
三、好友请求表.....	15
四、消息收发表.....	15
五、情报表.....	15
六、评论表.....	16
第二节 客户端设计.....	16
一、界面设计.....	16
二、界面交互.....	19
第三节 接口设计.....	19
一、用户相关.....	19
二、好友相关.....	22
三、情报相关.....	25
第四节 功能模块的传输设计.....	27
一、用户权限.....	27

二、好友管理.....	28
三、其他功能.....	28
<b>第五章 基于 iOS 的金融数据分析软件用户管理子系统的详细设计.....</b>	<b>30</b>
第一节 UI 界面的实现.....	30
一、用户登陆.....	30
二、用户注册.....	31
三、完善信息.....	32
四、个人中心.....	33
五、标签控制器.....	33
第二节 数据库的实现.....	36
一、用户表.....	36
二、好友管理表.....	36
三、好友请求表.....	36
四、消息收发表.....	37
五、情报表.....	37
六、评论表.....	37
第三节 前后端通信的实现.....	37
第四节 功能流程图.....	38
一、注册.....	39
二、登陆.....	39
三、好友.....	40
四、情报.....	40
第五节 技术难点分析.....	40
一、iOS 中的网络连接.....	40
二、全局变量的声明和使用.....	43
三、定时器的使用.....	44
四、动态列表.....	45
五、键盘隐藏.....	46
六、控件相关技术.....	47
<b>第六章 基于 iOS 的金融数据分析软件用户管理子系统的部署及测试.....</b>	<b>49</b>
第一节 系统部署.....	49
第二节 系统测试.....	49
一、测试概要.....	49
二、测试结果.....	50
<b>总结.....</b>	<b>54</b>
<b>参考文献.....</b>	<b>55</b>
<b>致谢.....</b>	<b>56</b>

## 第一章 绪论

为了更好的、安全的使银行全面了解企业的各类金融相关信息，需要构建相应金融数据分析软件，可将不同来源发布的各类金融信息整合后存放于数据库中，并根据相关数据和指标自动分析，其中包括用户管理子系统、融报管理子系统、商机管理子系统等。基于上述目的，本课题将实现基于 iOS 的金融数据分析软件中用户管理子系统的相关功能。

本章主要对金融数据分析软件用户管理子系统中的相关模块涉及到的软件发展现状进行总结分析，并对课题的相关内容和开发环境作出介绍。本系统主要依赖 iOS 平台，选择采用 Objective-C 语言编写基于 iOS 客户端的代码，客户端通过调用相关接口得到数据表中相关数据，从而完成客户端与服务器端的 HTTP 通信，设计实现用户权限、好友管理、情报管理、用户评论及积分管理等功能，最终完成高效的基于 iOS 的金融数据分析软件用户管理子系统。

### 第一节 相关技术的发展现状

本课题主要实现基于 iOS 移动设备系统平台的用户管理子系统，下面针对本课题相关技术发展状况作简单介绍。

#### 一、移动设备操作系统

当前移动设备的市场上主要有 Android、iOS 两大主流手机操作系统，本节将针对 Android 和 iOS 操作系统进行简要介绍，并对其进行分析和比较。

##### （一）Android 操作系统

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，由 Google 公司和开放手机联盟领导开发。Android 平台的最大优势是开发性，平台提供给第三方开发商宽泛、自由的开发环境，也由此诞生众多丰富、新颖、实用性好的 Android 应用。数据显示，2017 年中国智能手机市场 Android OS 份额高达 87%，占据绝对主流地位。

##### （二）iOS 操作系统

iOS 是由苹果公司开发的移动操作系统，最初是设计给 iPhone 使用的，后来陆续用到 iPod touch、iPad 等产品上。iOS 的系统架构分为四个层次：核心操作系统层（Core OS layer）、核心服务层（Core Services layer）、媒体层（Media layer）和可触摸层（Cocoa Touch layer）。其中核心操作系统层主要包含文件系统、网络基础、安全特性和一些设备驱动以及系统的 API。核心服务层主要用于提供核心服务，如字符串处理函数、网络管理、偏好设置等。媒体层的框架和服务依赖于核心服务层，并向可触摸层提供画图和多媒体服务，如声音、图片和视频等。可触摸层可直接调用如触摸事件、照相机管理等，该层包含 UIKit 和 Foundation 框架<sup>[1]</sup>。

总之，从流畅性来说，Android 系统采用的虚拟机运行机制需要消耗更多的系统资源，更易出现卡顿的现象，iOS 系统则更具有优势<sup>[2]</sup>；从安全角度来说，Android 是个开

源系统，系统内容对所有人开放，而 iOS 系统同时控制软件和硬件，使得其具有更高的安全性和封闭性。由此，本课题将选用 iOS 平台对金融数据分析软件用户管理子系统进行开发。

## 二、相关软件

本课题用户管理子系统中的好友管理模块，涉及到查找好友，添加好友，好友聊天等基本功能。其中的情报管理模块，涉及到浏览情报，修改情报，查看评论及进行评论的功能。腾讯旗下的 QQ、微信是国内最火的通信工具，下面将以 QQ 和微信朋友圈为例进行相关技术现状的分析和介绍。

QQ 因为其合理的设计、良好的应用、强大的功能、稳定高效的系统运行，赢得了广大用户的肯定。如今的腾讯 QQ 需要用户进行实名注册才可使用，用户可以通过 QQ 号码查找添加好友，也可在允许 QQ 访问通讯录的前提下，找到通讯录中的朋友，或者根据好友条件泛搜好友，在搜索列表中选择添加新的好友。发送请求需要经过对方的同意才能添加成功，添加好友成功的用户双方将会出现在彼此的联系人列表中成为对方的 QQ 好友。好友双方进行聊天时，交谈中的一方只有按下发送键后才会将消息显示在双方的屏幕上。在限定的时间之内，还提供“撤销”功能，即将已发送的消息在双方屏幕和历史消息记录中删除。消息的形式也是多种多样，用户可以使用文字、图片、语音、小视频等多种方式。QQ 不仅使人与人之间的交流更加快捷方便，更让交流的方式变得丰富多彩。目前 QQ 已经覆盖 Microsoft Windows、OS X、Android、iOS 等多种主流平台，支持在线聊天、视频通话、点对点断点续传文件、共享文件、自定义版面、QQ 邮箱等多种功能，并且可与多种通讯终端相连接。

相比于 QQ，微信同样作为一款热门聊天工具，其界面更加简洁，隐私性更高。除了聊天功能，微信推出的朋友圈功能也深受用户的喜爱。用户可以在朋友圈以文字、图片或小视频的方式分享自己的生活动态，并选择所发布动态的可见性，其授权好友便可在自己的朋友圈中浏览并评论好友的动态。

通过对市场上相关软件的总结分析，本课题以 iOS QQ 为主要模型，参照微信朋友圈的相关功能实现，采用 HTTP 协议，进行原型设计，完成基于 iOS 的金融数据分析软件用户管理子系统中的用户权限、好友管理、情报管理、用户评论及积分管理等功能。

## 三、HTTP 通信协议

HTTP (Text Transfer Protocol) 超文本传输协议基于 TCP/IP 通信协议来传递数据，是基于客户端/服务端 (C/S) 的架构模型，其通过一个可靠的链接来交换信息，是一个无状态、无连接、媒体独立的请求/响应协议<sup>[3]</sup>。

一个 HTTP “客户端”是一个应用程序，通过连接到服务器达到向服务发送一个或多个 HTTP 请求的目的。HTTP 协议具有 OPTIONS、HEAD、GET、POST、PUT、DELETE、TRACE、CONNECT 共 8 种请求类型，本课题主要用到 GET 和 POST 方法。一个 HTTP “服务端”同样也是一个应用程序，其接收客户端的请求并向客户端发送 HTTP 响应数



据。服务器收到 HTTP 请求之后，会有多种方法响应这个请求，如单进程 I/O 模型、多进程 I/O 模型、复用 I/O 模型和复用多线程 I/O 模型。

由于 HTTP 协议简单，使得其服务器的程序规模小，因而通信速度很快。协议使用 Post 请求方式进行通信，使得信息传递具有安全性。服务器端采用数据库方便对信息进行永久存储，同时具有组织性、条理性和可共享性。另外，客户端与服务器间采用 HTTP 协议进行通信，对于客户端平台无强制限定，可实现跨平台通信。

## 第二节 课题内容与开发环境

### 一、关于课题

本课题主要依赖 iOS 平台，在熟练掌握 Xcode 与 Objective-C 语言开发技术的基础上，通过分析了解金融数据分析软件中用户管理子系统的具体需求，构建在线数据库，利用 Http 协议实现前后台数据通信，设计实现用户权限、好友管理、情报管理、用户评论及积分管理等功能，最终完成高效的基于 iOS 的金融数据分析软件用户管理子系统。

金融数据分析软件用户管理子系统涉及到三个部分的实现，分别为数据库表的设计和建立、服务器端的接口设计及数据库操作的实现，以及客户端 UI 界面设计及前后端通信的实现。由于本课题为金融数据分析软件项目组中的一部分，主要实现功能为客户端 UI 界面设计及前后端通信的实现，服务器和数据库相关功能不作为本课题的重点内容，为了便于对课题的理解，本文将简要介绍数据库和服务器相关内容。

本课题主要设计思路为前端主要采用 Objective-C 技术进行客户端主要功能的实现，系统设计采用 MVC 设计模式。任务划分为两部分，第一部分设计并编程实现用户操作界面；第二部分利用 Http 协议，通过接口规范设计实现前后台数据交互，并保证数据的安全存取。

本课题主要研究内容包括：首先，了解 iOS app 开发框架，掌握 Objective-C 编程基础，了解其内部的消息机制和代理机制等，掌握 iOS 平台下控件编程和 storyboard 使用方法；其次，了解并分析相关软件的用户管理子系统基本框架及流程；再次，了解系统数据库中各类实体表的详细设计和后台数据交互的接口规范；最后，实现手机客户端与服务器端的 HTTP 通信，完善软件各个功能及手机客户端的界面设计。

### 二、开发环境简介

本课题依赖 iOS 平台，主要使用 Xcode 工具和 Objective-C 语言进行系统开发。下面将针对开发工具和开发语言进行简单介绍。

#### （一）关于 Xcode

本课题开发环境为 Xcode10.1，Xcode 是由苹果公司专门向开发人员提供的运行在操作系统 Mac OS X 上的集成开发工具，是开发 Mac OS 和 iOS 应用程序的最快捷的方式。Xcode 具有统一的用户界面设计，编码、测试、调试都在一个简单的窗口内完成<sup>[4]</sup>。

为了使涉及到 UI 界面的编程更加直观方便，Xcode 在可以进行纯代码编程的基础上引入了 Interface Builder 新功能——Storyboard，其基本思想是引入一个容器用于管理多

个 xib 文件并处理场景与场景之间的跳转，Storyboard 是实现 UI 界面编程的高效的重要的方式。

Xcode 界面布局如图 1 所示，其从左到右界面分别为：User interface file，用于查看代码文件的结构；Outline view，用于查看 UI 界面及控制器的结构；Canvas，用于实现控件的相关操作；Editor 为代码编辑界面；Attributes inspector 为相关实体的属性查看器。下部分为控制台，实现监视变量和查看输出的功能。

## （二）关于 Objective-C

Objective-C 是在 C 语言的基础上增加面向对象编程之后形成的一种面向对象语言。OC 作为一种面向对象语言，具有面向对象的三大特性：封装、继承和多态。除此之外，Objective-C 通过提供定义类、方法、属性的语法和其他可以提高类的动态扩展能力的结构等，扩展了标准的 ANSIC 语言<sup>[5]</sup>。

Objective-C 类库比 C++ 容易操作，但其不支持多重继承，也不包括命名空间机制，取而代之的是程序设计师必须为其类别名称加上前缀，虽然时常引起冲突，但作为开发 iOS 应用的主要语言，由于 iOS 操作系统的大众化，Objective-C 语言也变得十分热门。

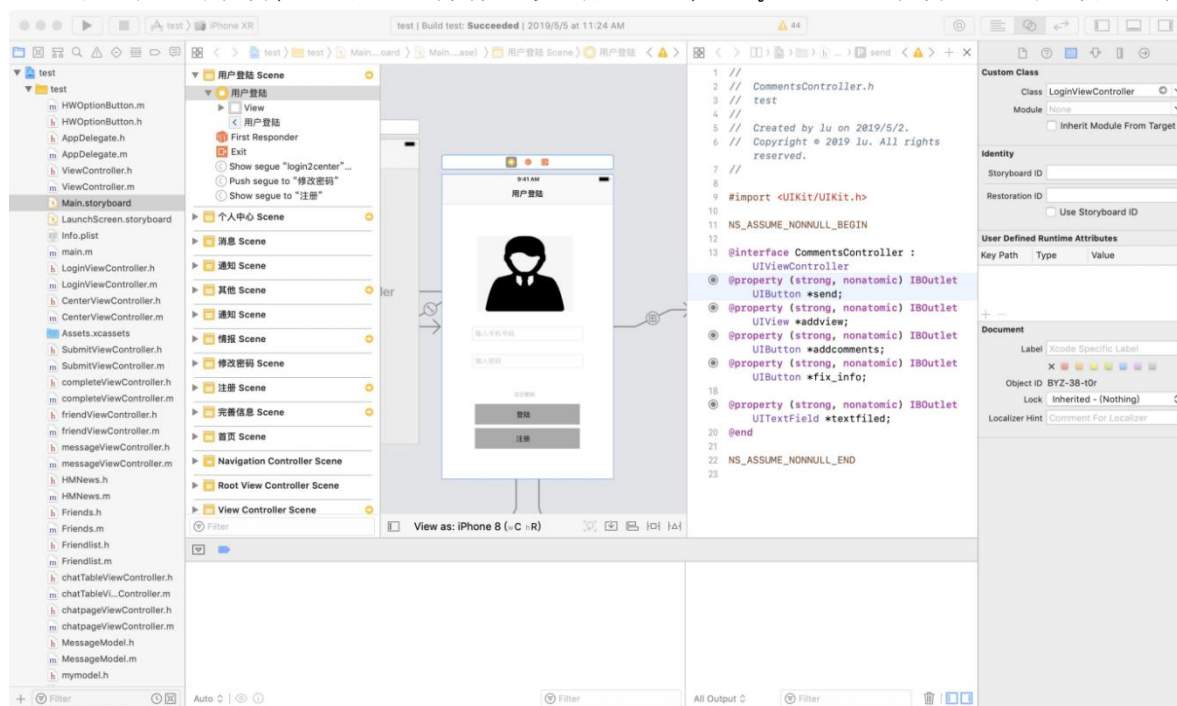


图 1 Xcode 开发环境

## 三、本文的内容和组织结构

本文主要设计和实现了基于 iOS 的金融数据分析软件用户管理子系统，并对设计过程中涉及的各个关键技术进行了深入的分析和讨论，主要包括 iOS 平台下的 Objective-C 编程、控件编程和使用方法等，并重点讨论了客户端界面设计、各界面间交互及客户端与服务器之间的网络连接。

本文共分为六章，各章节的组织结构和内容如下：



第一章，首先分析了该课题的研究背景，介绍了当前主流移动手机操作系统相关软件的发展现状，HTTP 通信协议的发展现状以及本文的主要内容，并对系统的开发环境进行了简要的介绍。

第二章，主要对基于 iOS 的金融数据分析软件用户管理子系统的任务进行了分析，并结合用户用例图讨论了系统的功能需求，并采用软件工程的管理思想，对开发模型进行了选择及简要介绍。

第三章，本章详细介绍了金融数据分析软件用户管理子系统的总体框架以及各子功能内部的处理流程。

第四章，本章主要对系统的数据结构、客户端和传输模块进行概要的设计与分析。其中涉及了系统相关数据表的介绍、客户端用户界面的设计与交互以及各功能模块的详细传输流程设计。

第五章，本章根据系统概要设计中用户界面设计及交互的相关内容对用户管理子系统客户端用户界面的功能实现做出了详细介绍，并以表格的形式具体分析了客户端与数据库通信的实现，提出了课题中的涉及的技术难点及其解决方案。

第六章，首先对系统进行部署，然后根据测试的概要说明从用户权限、好友管理、情报功能和用户评论等几个方面进行了测试，并对测试结果进行了预测和展示。

最后，总结了本文的主要研究内容和研究成果，分析了系统存在的不足，并对金融数据分析软件用户管理子系统今后的发展方向进行了展望。

## 第二章 基于 iOS 的金融数据分析软件用户管理子系统的需求分析

本章首先对基于 iOS 的金融数据分析软件用户管理子系统的任务目标进行了简要分析。然后，简要介绍本项目所选用的开发模型，并结合用户用例图，对该软件用户管理子系统的需求进行详细分析。

### 第一节 任务概述

本课题致力于实现金融数据分析软件中用户管理子系统的相关功能, 包括用户权限、好友管理、情报管理、用户评论及积分管理等。

其中, 用户权限主要包括用户注册, 用户登陆, 编辑完善个人信息的功能。好友管理主要包括按照条件搜索好友, 查看好友信息, 添加好友, 并与好友进行通信的功能。情报管理是指用户可以浏览数据库中包含的所以企业情报, 并可以添加和修改情报的内容。用户评论功能是指用户可以查看企业情报的相关评论并进行评论。积分管理功能是指用户有权对提供的情报进行打分功能。

### 第二节 开发模型介绍

采用软件工程的管理思想, 在项目开发前选用一个合适的软件生命周期模型, 是项目成功的关键因素。软件开发过程以系统需求作为输入, 以要交付的产品作为输出, 涉及活动、约束和资源使用的一系列工具和技术。瀑布模型、快速原型化模型、增量模型、螺旋模型等都是典型的软件开发过程模型。由于本课题金融数据分析软件用户管理子系统的功能需求定义十分清晰, 所以选择采用瀑布模型<sup>[6]</sup>。瀑布模型如图 2 所示。

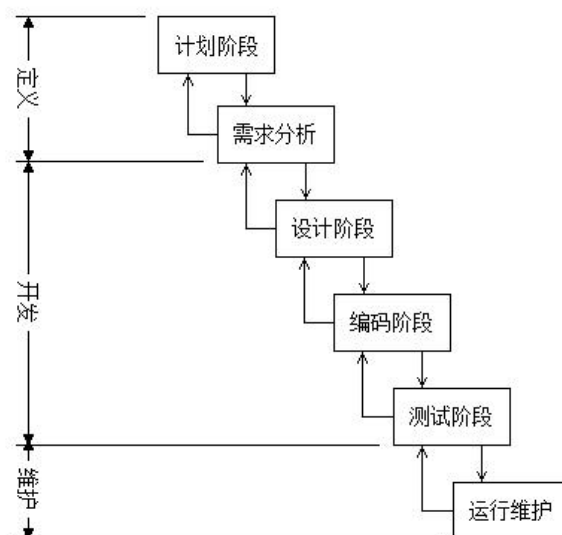


图 2 瀑布模型

瀑布模型是应用较为广泛的一个软件开发模型, 其开发过程是通过一系列阶段顺序展开的, 从系统需求分析开始直到产品发布, 每个阶段都会产生循环反馈。因此, 如果

有信息未被覆盖或者发现了问题，那么最好返回上一个阶段并进行适当的修改，开发进程从一个阶段流动到下一个阶段。瀑布开发模型的开发阶段主要分为需求分析、系统设计、实现和测试几个阶段。

在本系统开发过程中，在需求阶段通过系统功能用例图进行建模；设计阶段中首先形成系统总体架构，其次确定系统技术路线，最后完成各模块详细设计；系统实现部分中根据系统模块设计进行具体编码实现，最终对系统功能进行测试工作。

### 第三节 需求分析

需求分析是软件计划阶段的重要活动，也是软件生存周期中的一个重要环节。根据所选用瀑布模型的软件开发过程，首先进入开发的计划阶段，即对系统进行需求分析。下面将结合用户用例图，确定该金融数据分析软件用户管理子系统需要实现的具体功能。

#### 一、用户用例图

本系统中用户用例图如图 3 所示。

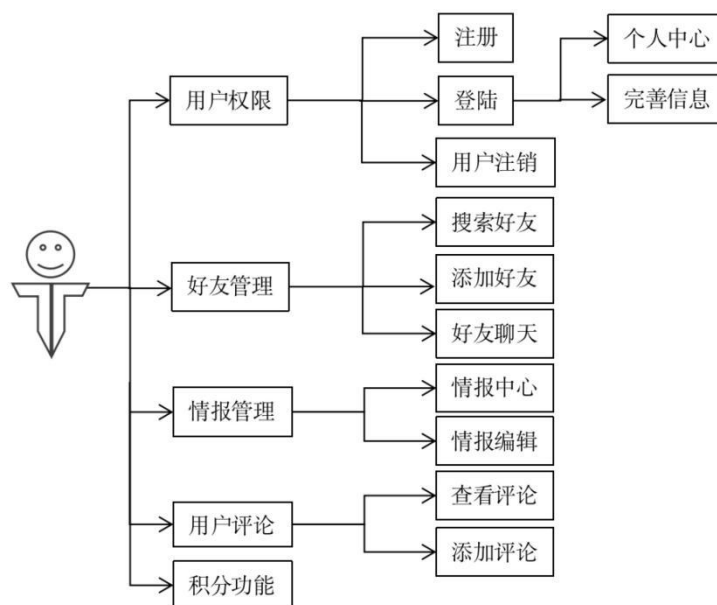


图 3 用户用例图

#### 二、功能描述

(1) 用户注册：如果用户没有可用的账号 ID，可以使用手机号码进行注册，需要经过图形验证码、短信验证码双重验证，客户端提交信息后服务器返回相关信息。若注册失败，则需根据返回的不同错误提示“该用户已注册”或“手机验证码或图形验证码错误”；

(2) 用户登陆：用户通过提交用户名和密码进行登陆，若登陆失败，需提示“手机号或密码错误”；

(3) 个人中心：用户登陆成功首先进入个人中心页面，显示用户的相关个人信息；

(4) 完善信息: 用户可以在注册成功时自动跳转到个人信息编辑页面, 进行个人信息的完善。也可在登陆之后的个人中心页面点击编辑, 跳转到个人信息编辑页面。完善信息页面需要提交相关个人信息 (姓名、所属金融机构、部门、职位、工作职责、办公电话、邮箱), 其中部门和职位不可自定义, 用户需要在提供选项中进行选择。点击提交按钮, 客户端向服务器提交相关信息;

(5) 搜索好友: 用户可以通过部门关键字查找和查看相关用户信息并进行添加好友的操作;

(6) 添加好友: 用户查看好友申请列表时会监听申请信息, 并显示到该界面。用户可以选择添加或忽略该申请消息;

(7) 好友聊天: 用户登陆成功后会监听好友发送过来的信息, 并显示到聊天界面上, 同时也可向好友发送即时消息;

(8) 情报中心: 用户进入情报中心界面, 会显示数据库中的所有企业情报详细信息;

(10) 情报编辑: 用户进入情报编辑界面, 需要填写相关情报信息 (发行人、发行日期、发行场所、注册金额、发行金额、分级、票面利率、期限、担保方式、情报等级、积分), 其中发行场所为选择项。点击提交按钮, 客户端向服务器提交相关信息;

(11) 查看评论: 用户点击某条情报可查看其他用户对该条情报的评论;

(12) 添加评论: 用户点击某条情报可对其进行评论;

(13) 用户注销: 用户在客户端进行注销, 用户退出该系统后, 可更换账号再次登陆。

## 第三章 基于 iOS 的金融数据分析软件用户管理子系统的整体结构

本章详细介绍了金融数据分析软件用户管理子系统的总体框架以及各子功能内部的处理流程。系统采用 C/S 体系结构，采用 HTTP 协议作为通信协议，通过调用数据库的相关接口实现客户端与服务器端之间的数据交换，最终完成用户管理子系统中相关模块的基本功能。

### 第一节 系统整体结构分析

基于 iOS 的金融数据分析软件，采用客户端 / 服务器端架构的体系结构，主要包括客户端、服务器、数据库三大部分。本课题主要完成用户管理子系统中相关模块的功能，完成用户管理子系统中涉及到的各个界面的设计和跳转，实现客户端与服务器间的高效的安全的通信。系统整体结构示意图如图 4 所示。

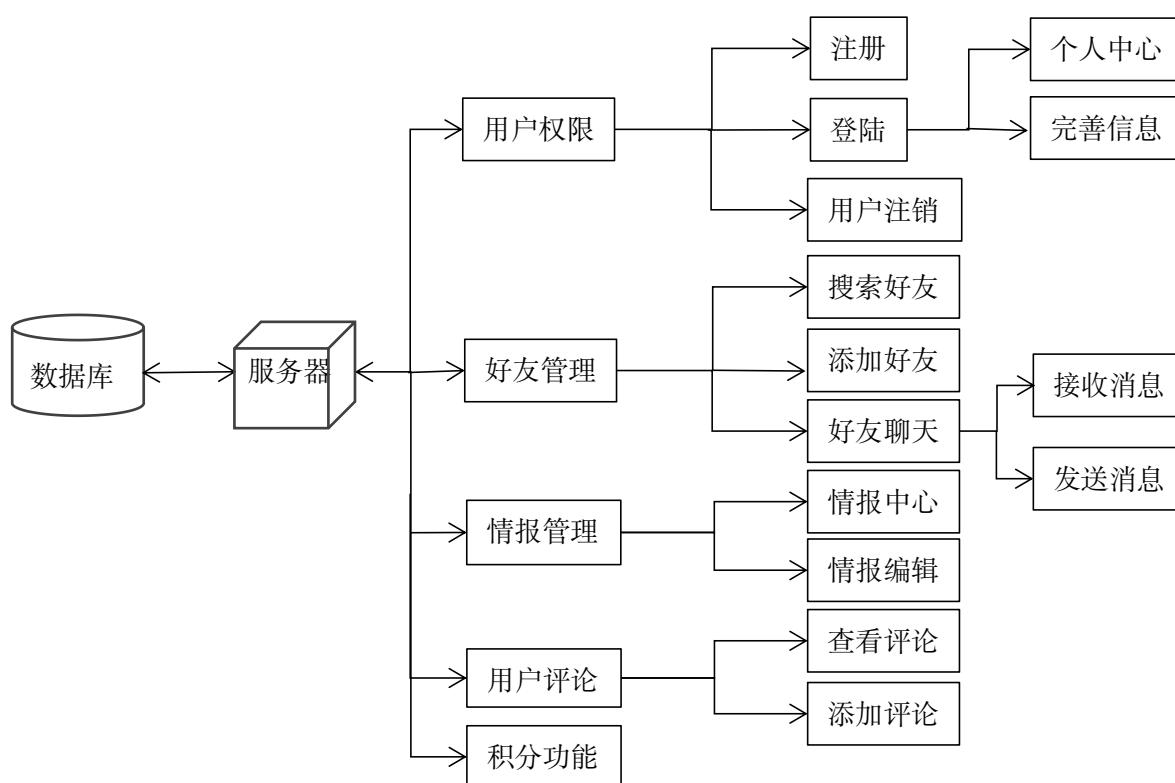


图 4 系统整体结构

客户端：以 iOS 手机为平台，用户需要进行注册、登陆，并可编辑修改个人信息；用户可按部门关键字查找并添加好友，好友间可以进行聊天；用户有权浏览系统中的企业情报，并对情报进行编辑修改和评论。客户端进行相关操作时，需要通过访问不同接口向服务器发起相关请求并等待响应。



服务器端：本项目租用阿里云空间构建相应服务器端，基于 linux 系统安装了面向 pythonweb 管理的 Flask 轻量级框架，后台数据库采用关系型数据库 MySQL，使用 python 语言编写了面向前端的数据库访问接口。本项目仅为团队合作开发的横向科研项目其中的一个子课题，接口部分的代码编写由团队其他成员负责。

数据库：用户管理子系统共涉及 6 张表，分别为用户表，存储用户 ID、用户密码及个人相关信息；好友管理表，记录好友关系；好友请求表，记录未完成建立的好友请求；消息收发表，存储好友之间发送消息的时间及内容；情报表，记录由用户添加的所有企业的详细情报信息；评论表，记录用户对某条情报的评论内容及时间。

对表数据的操作由客户端通过访问相关接口向服务器发送请求，服务器访问数据库，对数据进行读取、写入、删除等一系列操作。

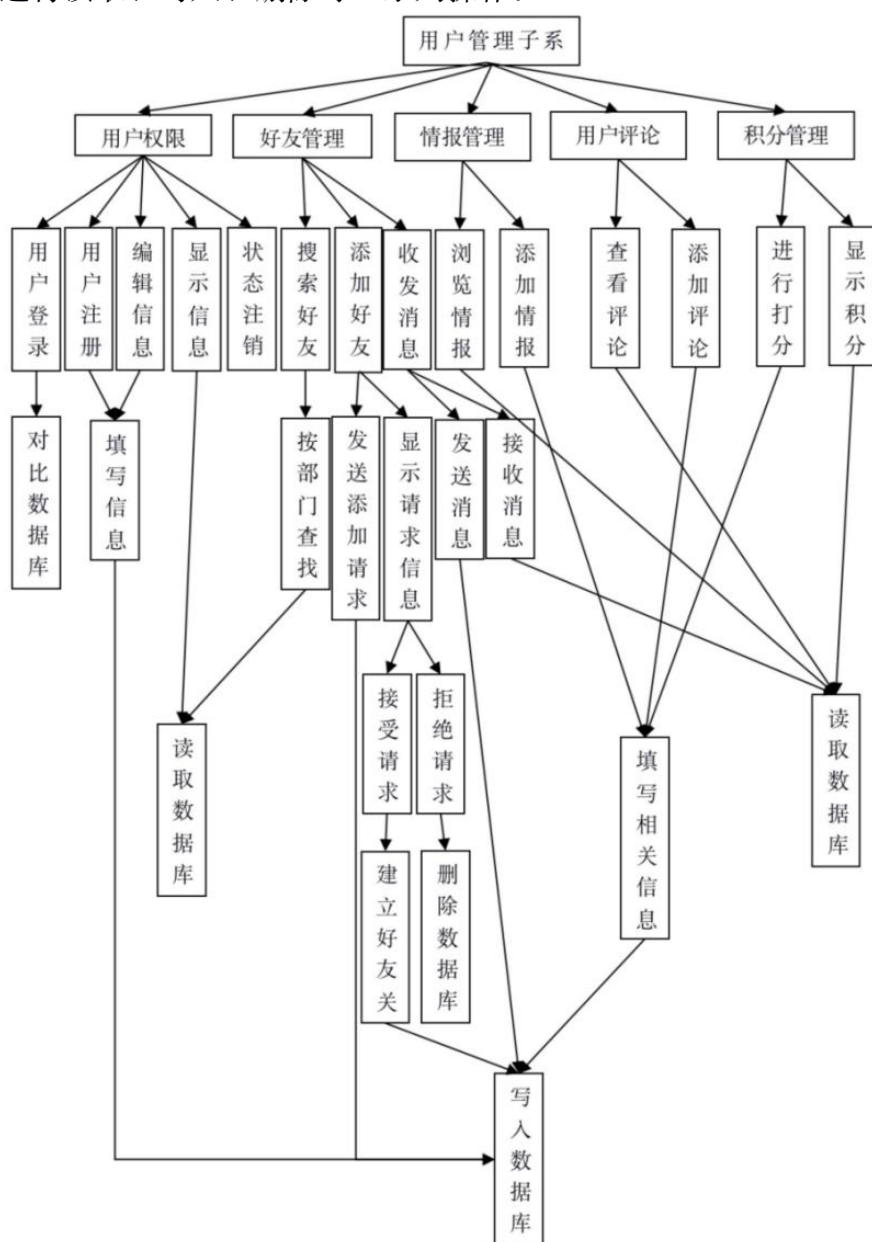


图 5 处理流程示意图

## 第二节 各模块处理流程分析

基于 iOS 的金融数据分析软件包括用户管理子系统、融报管理子系统和商机管理子系统三个主要模块，而本课题所设计的用户管理子系统结构主要可以分为五个模块，分别为用户权限、好友管理、情报管理、用户评论及积分管理。

### 一、处理流程示意图

根据上述系统整体结构，本节对各模块的处理流程进行了分析。金融数据分析软件用户管理子系统共分为五个模块，每个模块又细分出不同的功能，不同功能的实现对应了不同的用户操作和数据库操作。

以系统中好友管理模块的添加好友功能为例，添加好友涉及到好友双方的操作。首先用户 1 要发送添加信息，并将该申请信息写入数据库。用户 2 登陆系统后显示用户 1 发送的申请消息，用户 2 可选择接受或拒绝该请求，根据用户的选择服务器对数据库完成写入或删除操作，即建立好友关系或删除相关信息。

其他功能模块的具体流程示意图，如图 5 所示。

### 二、处理流程说明表

为了对上述各模块处理流程进行更加详细的说明，在此将系统按模块级别划分并分别从用户操作、数据库操作、功能和调用关系这四个方面对系统流程处理过程列出详细说明，如表 1。

表 1 处理流程说明表

一级模块	二级模块	三级模块	四级模块	用户操作	数据库操作	功能	调用关系
基于 IOS 的金融数据分析软件用户管理子系统	用户权限	用户注册	填写信息	填写手机号、图形验证码、短信验证码并设置密码，点击提交。	读取对比写入	用户填写注册相关信息，在数据库进行对比，合法用户可进行注册并写入数据库，否则返回错误。	由用户管理子系统的统初始界面调用
			获取图形验证码				
			获取短信验证码				
		用户登录		输入手机号及密码，点击登陆。	对比	用户填写登陆信息，在数据库中对对比用户名、密码。合法用户登陆成功，否则返回错误。	由用户管理子系统的统初始界面调用

# 北京工业大学毕业设计（论文）

续表 1 处理流程说明表

		编辑信息		用户填写相关信息，并点击提交。	写入	编辑个人信息并写入数据库。	注册成功自动进入或编辑个人信息时调用	
		显示信息		登陆成功自动进入个人中心界面。	读取	读取数据库，显示个人资料。	登陆成功或进入个人中心时自动调用	
		状态注销		点击注销按钮，退出系统。			用户选择注销时自动调用	
	好友管理	搜索好友		填写部门，点击搜索。	读取	根据用户所填部门，在数据库中进行查找，返回并显示在搜索界面。	搜索好友时调用	
		添加好友	发送添加好友请求		选中好友并点击添加按钮。	写入	根据用户选中好友，将相关信息写入数据库。	搜索好友并添加或接收好友请求时调用
			显示请求信息	接受	将对方信息显示在好友请求列表中，用户可选择接受或拒绝该请求。	读取写入	读取数据库相关表，将请求信息显示在用户界面上，若用户同意添加则建立好友关系写入数据库，否则删除数据库中请求信息	进入好友申请信息界面时自动调用
				拒绝		读取删除		
		收发消息	发送消息		填写发送内容，点击发送按钮。	读取写入	将发送内容写入数据库并读取数据库显示聊天内容。	进入聊天界面并发送消息时调用
			接收消息		进入聊天界面。	读取	读取数据库将聊天内容显示在聊天界面上。	进入聊天界面时调用
	情报管理	浏览情报		进入情报中心界面。	读取	读取所有企业情报显示在用户界面。	进入情报中心自动调用	

## 北京工业大学毕业设计（论文）

续表 1 处理流程说明表

		编辑情报	添加情报	点击添加或修改情报按钮进入情报编辑界面，用户填写相关信息，并点击提交。	写入	编辑情报信息并写入数据库。	添加情报或修改情报时调用。
			修改情报				
	用户评论	查看评论		选中情报。	读取	读取该条情报的相关评论并显示。	选中情报时自动调用。
		添加评论		点击添加评论按钮。	写入	用户添加评论并写入数据库。	进入评论页面并添加评论时调用。
	积分管理	由于积分针对不同情报，所以添加积分在编辑情报时同时添加，积分显示在浏览情报时同时显示。积分管理的相关功能在情报管理模块实现。					

## 第四章 基于 iOS 的金融数据分析软件用户管理子系统的概要设计

本章主要从系统数据结构、客户端交互、传输模块三个方面进行概要设计。

由于用户管理子系统需要存储用户信息、用户好友关系、聊天消息、情报信息及用户评论等内容，所以首先需要设计搭建相关的数据库表来实现数据的结构化、永久性存储。数据库相关任务并非本课题所涉及的内容，为了便于对本课题的理解，在此对数据库中与本课题相关的数据表作简要分析。

对数据库的操作要根据客户端用户的操作才能确定。客户端可获取外部用户的输入信息，并按照用户的指令进行相应的页面跳转和数据库操作。最后，通过客户端-服务器-数据库三者之间的交互完成系统的功能。

### 第一节 数据表结构设计

用户管理子系统共涉及 6 张表，分别为用户表、好友管理表、好友请求表、消息收发表、情报表和评论表。下面将对系统数据库中各类实体表的详细设计做出简要说明。

#### 一、用户表

用于存储用户 ID、注册手机、用户邮箱、姓名、密码、办公电话、金融机构、工作职责等用户个人信息，其中主要涉及到用户所属的金融机构、部门等相关信息。用户管理子系统中登陆、注册、查询好友、显示个人信息等功能都将对该表做相关操作。用户表的详细设计如表 2 所示。

表 2 用户表 (front\_users)

编号	属性名	属性描述	主键	外键	允许空值	数据类型	精度范围
1	user_id	用户唯一标识	是	否	否	VARCHAR	64
2	mobile	手机	否	否	是	VARCHAR	16
3	email	邮箱	否	否	是	VARCHAR	64
4	real_name	真实姓名	否	否	是	VARCHAR	64
5	institution	金融机构	否	否	是	VARCHAR	64
6	department	部门	否	否	是	VARCHAR	32
7	position	职位	否	否	是	VARCHAR	32
8	responsibility	工作职责	否	否	是	VARCHAR	64
9	office_phone	办公电话	否	否	是	VARCHAR	16
10	pwd_salt	密码	否	否	否	BINARY	64

#### 二、好友管理表

用于存储用户与用户之间的好友关系，包括好友双方的用户编号。好友关系相对于某一用户实体而言属于一对多的关系，所以建表时，每对用户间的好友关系采用两条记录进行双向存储，并且两个用户之间只允许建立一次好友关系。用户管理子系统中的添



加好友功能将对该表做添加记录的操作。好友管理表的详细设计如表 3 所示。

表 3 好友管理表 (Friendtable)

编号	属性名	属性描述	主键	外键	允许空值	数据类型	精度范围
1	ID	编号	是	否	否	INTEGER	10
2	UserID1	用户 1 编号	否	否	否	INTEGER	10
3	UserID2	用户 2 编号	否	否	否	INTEGER	10

### 三、好友请求表

好友请求表用于记录所有用户发送的好友申请信息。在两个用户正式成为好友之前,需要进行“请求添加”、“通过/拒绝添加请求”、“添加信息反馈”三个步骤,在此过程中,两用户的好友关系不能正式确立,需要建一个表来记录请求过程,当双方正式成为好友之后,好友关系才被记录进入好友管理表。好友请求表的详细设计如表 4 所示。

表 4 好友请求表 (FriendRequesttable)

编号	属性名	属性描述	主键	外键	允许空值	数据类型	精度范围
1	RequestID	请求编号	是	否	否	INTEGER	10
2	SendID	发送者编号	否	否	否	INTEGER	10
3	RecID	接收者编号	否	否	否	INTEGER	10
4	Sign	标记	否	否	否	VARCHAR	10

### 四、消息收发表

该表用于记录用户间聊天的信息,包括发送者和接收者的 ID、发送时间、发送内容、和是否成功发送的标记。若发送消息时,对方在线,成功发送后,发送标记置为 1,表示发送成功;若当时对方不在线,标记置为 0,表示还未发送,待对方上线后发送,再把标记置为 1。消息收发表的设计如表 5 所示。

表 5 消息收发表 (Messagetable)

编号	属性名	属性描述	主键	外键	允许空值	数据类型	精度范围
1	MsgID	消息编号	是	否	否	INTEGER	10
2	SendID	发送者编号	否	否	否	INTEGER	10
3	RecID	接收者编号	否	否	否	INTEGER	10
4	Time	发送时间	否	否	否	VARCHAR	50
5	Message	消息内容	否	否	是	VARCHAR	1000
6	Succeed	成功标记	否	否	否	VARCHAR	10

### 五、情报表

用于记录所有情报的详细信息,包括情报 uuid、提供者 id、提供情报时间、情报等级、发行人、发行场所、发行日期、注册金额等情报信息,以及用户对该条情报的相关

评论和情报积分信息。用户管理子系统中浏览情报、添加情报、修改情报等功能都将对该表做相关操作。情报表的详细设计如表 6 所示。

表 6 情报表 (intelligence)

编号	属性名	属性描述	主键	外键	允许空值	数据类型	精度范围
1	uuid	uuid	是	是	否	VARCHAR	64
2	u_id	提供者 id	否	否	是	VARCHAR	64
3	p_date	提供时间	否	否	是	DATE	10
4	t_level	情报等级	否	否	是	VARCHAR	50
5	i_issuer	发行人	否	否	是	VARCHAR	100
6	i_place	发行场所	否	否	是	VARCHAR	20
7	i_date	发行日期	否	否	是	VARCHAR	50
8	regis_amount	注册金额	否	否	是	DOUBLE	
9	i_amount	发行金额	否	否	是	DOUBLE	
10	grading	分级	否	否	是	VARCHAR	20
11	face_rate	票面利率	否	否	是	DOUBLE	
12	term	期限	否	否	是	VARCHAR	10
13	guar_way	担保方式	否	否	是	VARCHAR	100
14	comment_ids	评论者 id	否	是	否	VARCHAR	255
15	comment	评论	否	否	是	VARCHAR	1000
16	comment_time	评论时间	否	否	是	DATETIME	
17	integral	积分	否	否	是	VARCHAR	64

## 六、评论表

用于记录某条情报的所有评论，包括情报 id、评论者 id、评论时间及评论内容。其中以情报 id 为主键。评论表的详细涉及如表 7 所示。

表 7 评论表 (comments)

编号	属性名	属性描述	主键	外键	允许空值	数据类型	精度范围
1	uuid	情报的 id	是	否	否	VARCHAR	64
2	comment_ids	评论者 id	否	是	是	VARCHAR	64
3	comments	评论	否	否	是	VARCHAR	1000
4	comment_time	评论时间	否	否	否	DATETIME	

## 第二节 客户端设计

### 一、界面设计

用户在客户端，通过 UI 界面进行人机交互。根据界面的相关提示及要求，用户完成填写信息、浏览信息并选择下一步操作的任务。UI 界面作为与用户交互的最直接工具，其设计直接关系到用户的使用体验，界面设计的直观性和易用性更是软件是否受用户喜

爱的重要因素。

本课题金融数据分析软件用户管理子系统的界面设计遵循的原则主要有简易性、一致性和人性化。简易性是指界面简洁直观，便于用户的使用并且能够减少用户发生错误选择的可能性。一致性是指系统界面结构清晰一致，设计整体风格统一，使产品具有独特的品牌形象。人性化即最大限度地考虑用户的体验，系统在高效率的前提下保证用户的满意度。

下面将对金融数据分析软件用户管理子系统的全部界面设计作详细说明。

### （一）注册界面

该界面用于用户填写相关信息进行账号注册。需要填写的信息为：手机号码、图片验证码、短信验证码、密码及确认密码。信息填写完毕后点击“提交”按钮，在图片验证码、短信验证码正确并且密码一致的情况下，可以成功注册。否则提示错误信息。

### （二）登陆界面

该界面用于用户登陆金融数据分析软件。需要填写的信息为：手机号和密码。信息填写完毕后点击“登陆”按钮，在手机号和密码匹配的情况下，可以成功登陆。否则提示错误信息。

### （三）个人中心

该界面用于显示用户的个人资料。显示信息为：用户姓名、办公电话、所属金融机构和职位。界面包含“编辑”、“更多”、“注销”三个按钮。用户点击“注销”按钮弹出提示框提示用户“是否进行注销”，选择“注销”则退出软件，选择“取消”则返回个人中心界面。点击“编辑”和“更多”按钮将分别跳转到其他界面。

### （四）完善信息

该界面用于用户添加或修改个人信息。需要填写的信息为：姓名、金融机构、部门、职位、工作职责、办公电话和邮件。其中部门和职位不可自定义，部门提供选择项为：金融市场部、资产管理部、投资银行部、公司业务部、国际业务部和战略客户部。职位提供选择项为：总经理、副总经理、处长、副处长、行长、副行长、客户经理、经理、高级经理、副董事、董事、执行董事和董事总经理。用户点击“提交”按钮，则可完成个人信息的编辑。

### （五）好友界面

用于按部门关键字对好友进行搜索。用户输入部门信息，点击搜索按钮，界面显示全部符合要求的用户信息。选中某一用户，弹出提示框提示“是否加为好友”，设计提供“取消”和“确定”两个按钮供用户进行选择。

### （六）通知界面

用于显示好友申请信息。设计以列表的形式将全部申请添加好友的用户信息显示在该界面上。用户选中某条申请信息，弹出提示框提示“加为好友”，设计提供“同意”和“忽略”两个按钮进行选择。

### （七）消息界面

用于以列表的形式显示用户的全部好友，用户选中某一好友，进入与该好友的聊天界面。

### （八）聊天界面

聊天界面用于好友双方进行聊天，包括消息的收、发和显示。该界面以列表的形式显示历史消息，包括每一条消息的发送者、消息内容和消息发送的时间。界面下方为文本输入框和“发送”按钮，用户选中文本输入框时，键盘自动弹出，滚动聊天界面时，键盘自动收回。

### （九）情报界面

该界面用于以列表的形式显示数据库中所有企业情报信息。若选中某条情报，进入到该情报的相关评论界面。若用户点击界面中的“添加”按钮，跳转到情报编辑界面。

### （十）评论界面

评论界面用于显示本条情报的所有用户评论，包括评论人姓名、评论内容和评论时间。界面包含“添加评论”和“修改情报”两个按钮。用户点击“添加评论”，界面下方弹出隐藏文本框及“发送”按钮，用于用户进行评论编辑。用户点击“修改情报”，进入情报编辑界面。

### （十一）情报编辑

情报编辑界面用于用户填写情报相关信息进行情报的添加或修改。需要填写的信息为：发行人、发行日期、发行场所、注册金额、发行金额、分级、票面利率、期限、担保方式、情报等级、积分。其中发行场所不可自定义，提供四个选择项分别为：银行间交易商协会、上海证券交易所、深圳证券交易所和北京金融资产交易所。信息填写完毕后点击“提交”按钮即可成功添加或修改情报。

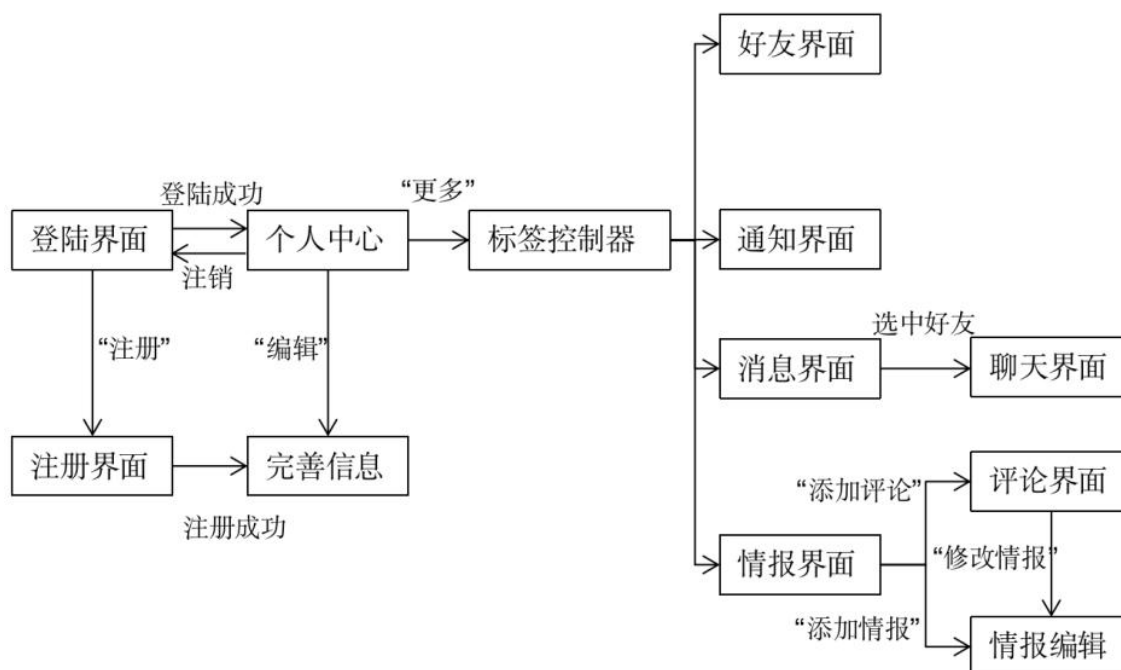


图 6 界面交互关系图

## 二、界面交互

用户管理子系统中各应用界面之间的交互关系如图 6 所示。其中矩形文本框表示界面，箭头表示界面间的跳转关系，箭头上的文字说明为跳转条件，其中带引号的说明为点击界面中的相关按钮发生跳转。

### 第三节 接口设计

基于数据表和客户端的界面设计，本节将以表格的形式详细介绍客户端访问数据库的接口设计，包括接口名称、访问地址、请求方式、提供字段以及返回值等相关信息，其中接口需要访问的地址格式均为 `http://47.92.50.218:8881/api/*`。

根据数据表结构按功能设计接口，共 16 个，分别为注册、登陆、图形验证码、短信验证码、完善个人信息、显示个人信息、搜索用户、发送好友请求、好友请求列表、接受好友请求、好友列表、发送消息、接收消息、获取情报、创建/修改情报和评论情报，接口的详细介绍见表 8-表 23。

#### 一、用户相关

##### （一）注册

注册接口用于向服务器端发送用户在登陆界面输入的相关信息，包括手机号、密码和手机验证码。服务器接到请求后其返回值有三种，分别为“注册成功”、“该用户已注册”和“手机号或验证码错误”。

表 8 注册接口

接口名称	注册	说明	提供用户信息
地址	/api/register		
请求方式	Post		
提供字段	参数	说明	示例
	mobile	手机号	13299999999
	pwd	密码	123456
	message_code	手机验证码	234567
返回值	error		
	info	successful	

##### （二）登陆

登陆接口用于向服务器发送用户在登陆界面输入的相关信息，包括手机号和密码。服务器接到请求后对比数据库，检查手机号和密码是否匹配并作出响应，其返回值有三种，分别为“登陆成功”、“手机号密码错误”和“该用户未注册”。

表 9 登陆接口

接口名称	登陆	说明	提供账号密码
地址	/api/login		




续表 9 登陆接口

请求方式	Post		
提供字段	参数	说明	示例
	mobile	手机号	13299999999
	pwd	密码	123456
返回值	error		
	token	2d3ead6ceeda11e8b5bc5800e345bbac	
	info	successful	

### (三) 图形验证码

用于登陆页面获取图形验证码。

表 10 图形验证码接口

接口名称	图形验证码		说明	图形验证码
地址	/api1/img_code			
请求方式	Get			
提供 字段	参数	说明	示例	
	无			
返回值	图形验证码图片			
				

### (四) 短信验证码

用于登陆页面获取短信验证码，服务器端的响应返回值有三种，分别为“手机号或图形验证码错误”、“该用户已注册”和“发送成功”。若成功该手机号应收到相应短信。

表 11 短信验证码接口

接口名称	短信验证码		说明	获取短信验证码
地址	/api1/message_code			
请求方式	Post			
提供字段	参数	说明	示例	
	mobile	手机号	13299999999	
	img_code	图形验证码	8972	
返回值	error			
	Info		successful	

### (五) 完善个人信息

用于向服务器端发送用户在完善信息界面输入的相关信息，包括真实姓名、金融机构、部门、职位、工作职责、办公电话、邮箱。服务器接到请求后其返回值有三种，分别为“登陆超时”、“权限不足”和“添加成功”。成功时将信息写入相关数据表。

## 北京工业大学毕业设计（论文）

表 12 完善个人信息接口

接口名称	完善个人信息	说明	提供用户信息
地址	/api1/personal_information		
请求方式	Post		
提供字段	参数	说明	示例
	real_name	真实姓名	张三
	nick_name	昵称	王五
	institution	金融机构	
	department	部门	
	position	职位	经理
	responsibility	工作职责	
	office_phone	办公电话	123456778
	email	邮箱	1234567213@163.com
	avatar	头像	
	card	名片	
返回值	error		
	info	Successful	

### (六) 显示个人信息

用于获取用户个人信息并在个人中心界面显示，服务器端接到请求后读取数据库获取用户信息并以字典数据类型返回，其中包括真实姓名、昵称、金融机构、部门、职位、工作职责、办公电话、邮箱和名片。

表 13 显示个人信息接口

接口名称	显示个人信息	说明	用户个人信息
地址	/api1/show_user_info		
请求方式	Get		
返回值	参数	说明	
	user_info	个人信息列表	
	real_name	真实姓名	
	nick_name	昵称	
	institution	金融机构	
	department	部门	
	position	职位	
	responsibility	工作职责	
	office_phone	办公电话	
	email	邮箱	

## 二、好友相关

### （一）搜索用户

用于搜索好友界面，用户输入部门关键字，服务器端收到请求后读取数据库并获得符合搜索条件的用户信息，最后返回给客户端。

表 14 搜索用户接口

接口名称	搜索用户	说明	根据部门搜索
地址	/api1/search_user		
请求方式	Get		
	参数	说明	示例
提供字段	department	部门关键字	金融市场部
返回值	error	错误信息	
	select_list	结果列表	
	0	第一个用户信息	
	0	用户 id	
	1	用户电话	
	2	邮箱	
	3	真实姓名	
	4	昵称	
	5	金融机构	
	6	部门	
	7	职位	
	8	工作职责	
	9	办公电话	
	1	第二个用户信息	内容同第一个用户
	info	解释说明	

### （二）发送好友请求

用于用户发送添加好友请求，服务器收到请求后将相关信息写入数据库并等待对方用户响应。

表 15 发送好友请求接口

接口名称	发送好友请求	说明	用户发送请求
地址	/api1/send_friend_request		
请求方式	Get		
	参数	说明	示例
提供字段	rec_id	接收用户 id	即上面搜索后得到的用户 id

续表 15 发送好友请求接口

返回值	error	错误信息	
	info	解释说明	发送好友请求成功

### （三）好友请求列表

用于通知界面显示好友请求信息，服务器自动获取数据库中与该用户相关的好友申请并返回给客户端。

表 16 好友请求列表

接口名称	好友请求列表	说明	用户查看好友请求
地址	/api1/friends_request_list		
请求方式	Get		
	参数	说明	示例
提供字段	无（自动从 session 中获取当前用户 id）		
返回值	error	错误信息	
	request_list	请求列表	
	0	请求人姓名	
	1	请求人 id	
	info	解释说明	

### （四）接受好友请求

用于用户对好友申请作出响应。

表 17 接受好友请求

接口名称	接受好友请求	说明	用户接受好友请求
地址	/api1/receive_friend_request		
请求方式	Get		
	参数	说明	示例
提供字段	send_id	发送用户 id	
返回值	error	错误信息	
	info	解释说明	好友添加成功

### （五）好友列表

用于好友列表界面显示用户的所有好友信息。

表 18 好友列表接口

接口名称	好友列表	说明	完成添加的好友列表
地址	/api1/friends_list		
请求方式	Get		
	参数	说明	示例
提供字段	无（自动从 session 中获取当前用户 id）		

续表 18 好友列表接口

返回值	error	错误信息	
	request_list	请求列表	
	0	好友姓名	
	1	好友 id	
	info	解释说明	

**(六) 发送消息**

用于聊天界面，用户给好友发送消息并显示在界面中。

表 19 发送消息接口

接口名称	发送消息	说明	给好友发送消息
地址	/api1/send_friends_message		
请求方式	Get		
	参数	说明	示例
提供字段	rec_id	接收人 id	
	content	发送内容	
返回值	error	错误信息	
	info	解释说明	消息发送成功

**(七) 接收消息**

用于聊天界面，用户接收好友发来的消息并显示在界面中。

表 20 接收消息接口

接口名称	接收消息	说明	接收好友发来的消息
地址	/api1/receive_friends_message		
请求方式	Get		
	参数	说明	示例
提供字段	send_id	发送人 id	
返回值	error	错误信息	
	friend_message_list	好友消息列表	
	0	好友姓名	
	1	发送时间	
	2	消息内容	
	me_message_list	本人消息列表	
	0	本人姓名	
	1	发送时间	
	2	消息内容	
	info	解释说明	



### 三、情报相关

#### （一）获取情报

用于情报中心界面显示全部企业情报信息，服务器自动获取数据库中所有情报信息并返回给客户端。

表 21 获取情报接口

接口名称	获取情报	说明	获取当前用户情报
地址	/api1/get_intelligence		
请求方式	Get		
	参数	说明	示例
提供字段	自动从 session 中获取		
返回值	intelligence_info	情报列表	
	0	第一条情报字典	
	uuid	记录唯一标识	
	user_id	当前用户 id	
	p_date	情报发行时间	
	t_level	情报等级	
	i_issuer	发行人	
	i_place	发行场所	
	i_date	发行日期	
	regis_amount	注册金额	
	i_amount	发行金额	
	grading	分级	
	face_rate	票面利率	
	term	期限	
	guar_way	担保方式	
	integral	积分	
	comments	评论列表	
	0	第一条评论字典	
	comment	评论内容	
	comment_id	评论人 id	
	comment_time	评论时间	
	1	第二条情报字典	内容同上

#### （二）创建/修改情报

用于客户端发送用户在情报编辑界面中输入的相关情报信息，其中情报唯一标识 uuid 仅在修改情报时发送。服务器接到添加请求则将新情报写入数据库，若是修改请求

则修改数据库中原有情报信息。

表 22 创建/修改情报接口

接口名称	创建/修改情报	说明	用户提供情报
地址	/api1/insert_intelligence		
请求方式	Post		
	参数	说明	示例
提供字段	uuid	记录唯一标识	不传 uuid 代表新建情报，会自动生成，传 uuid 代表修改
	t_level	情报等级	
	i_issuer	发行人	
	i_place	发行场所	
	i_date	发行日期	
	regis_amount	注册金额	
	i_amount	发行金额	
	grading	分级	
	face_rate	票面利率	
	term	期限	
	guar_way	担保方式	
	integral	积分	
返回值	error	错误信息	
	info	解释说明	创建/修改成功

### （三）评论情报

用于客户端发送用户在评论界面输入的相关评论信息，服务器接收到请求后将评论根据情报 uuid 写入相关数据表。

表 23 评论情报接口

接口名称	评论情报	说明	其他用户评论情报
地址	/api1/comment_intelligence		
请求方式	Post		
	参数	说明	示例
提供字段	uuid	评论的情报 id	
	content	评论内容	
返回值	error	错误信息	
	info	解释说明	评论成功

## 第四节 功能模块的传输设计

用户管理子系统包含五个模块，分别为用户权限、好友管理、情报管理、用户评论和积分功能。其中积分的相关功能在情报管理模块进行合并实现，将不做单独分析和设计。由于个别模块中某些功能的实现具有相似之处，下面将对该系统中的部分模块的传输设计进行重点分析。

### 一、用户权限

#### （一）用户注册

在注册过程中，用户输入手机号码、图片验证码、短信验证码、密码及确认密码，这些信息通过 HTTP 协议发送到服务器端。服务器端把接收到的信息与数据库进行比对，若图片验证码和手机验证码正确，且无重复的用户名则将信息写入数据库，成功后返回唯一表示该用户身份的 token\_id，若不成功则在客户端提示注册失败，其传输流程如图 7 所示。

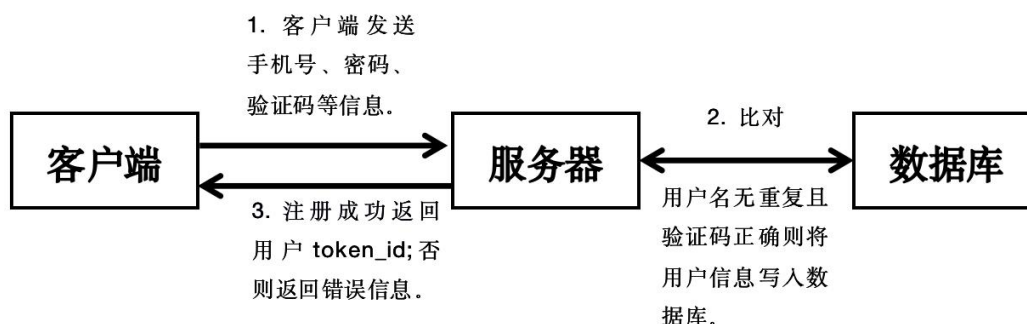


图 7 用户注册

#### （二）用户登陆

在登陆过程中，用户输入手机号和密码，客户端将相关信息通过 HTTP 协议发送到服务器端。服务器端需要根据手机号和密码，到数据库中进行查找和对比。查找操作是指在数据库中查找该用户信息，若不存在则用户为未注册状态；若存在，则将客户端发送来的密码与数据库中存储的密码进行对比，若密码一致则返回登陆成功，否则返回错误信息。其传输流程如图 8 所示。

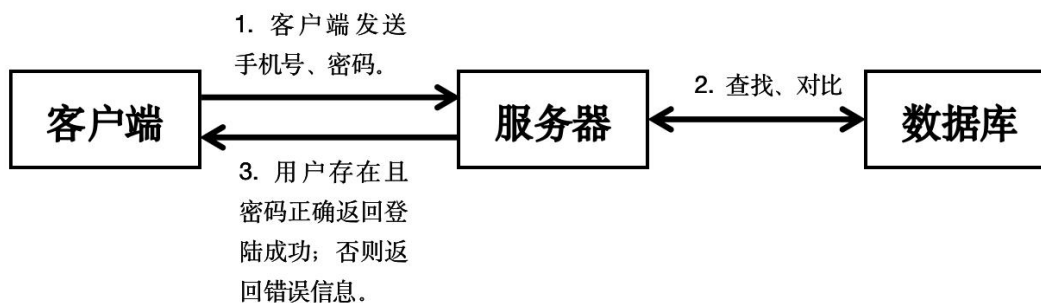


图 8 用户登陆

#### （三）编辑个人信息

编辑个人信息的过程就是客户端将用户编辑的相关信息发送给服务器，服务器根据用户的 id 将信息插入到数据库的相关数据表中。其传输流程如图 9 所示。

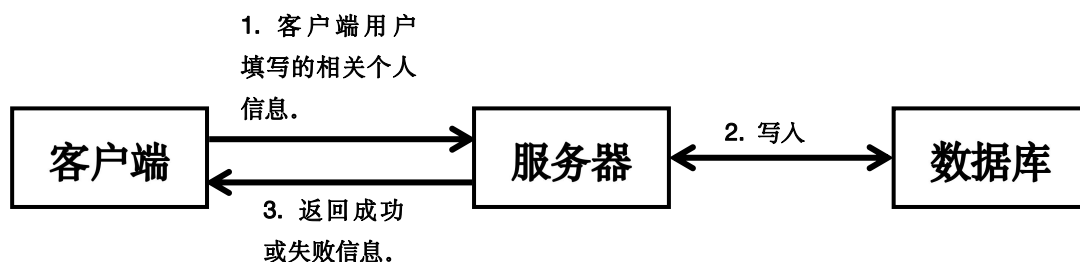


图 9 编辑个人信息

## 二、好友管理

### （一）搜索好友

搜索好友的过程即用户输入金融部门关键字进行搜索，客户端将关键字发送给服务器，服务器到数据库中进行查找，将符合搜索条件的用户信息返回给客户端。其传输流程如图 10 所示。

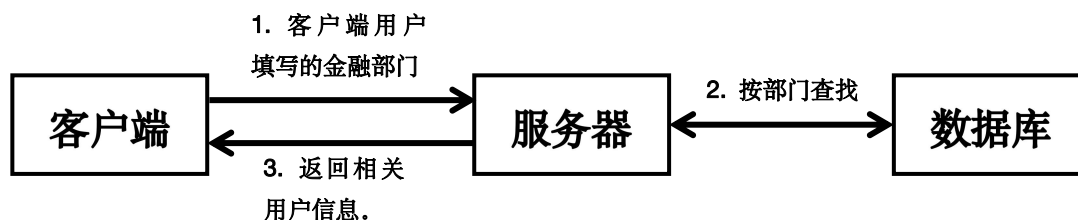


图 10 搜索好友

### （二）添加好友

添加好友涉及到两个客户端的交互，其过程相对复杂。申请消息由客户端 1 发起，用户 1 将要添加的用户 2 的 id 发送至服务器端，服务器经过数据库对比确定请求双方还未建立好友关系，并将该条申请消息写入数据库，等待对方的响应。当用户 2 登陆系统时，服务器自动读取对比是否有与该用户相关的申请信息并发送至客户端 2 提示用户对申请信息做出响应。用户 2 可以选择同意或拒绝申请，服务器根据用户 2 的响应对数据库进行插入或删除操作，并将结果返回给客户端 1。其传输流程如图 11 所示。

### （三）消息收发

过程与添加好友类似，客户端 1 发送聊天信息，通过服务器写入数据库。与此同时，客户端 2 若在登陆状态，服务器检测是否有接收者为客户端 2 的 id 的消息。若有，则响应客户端 2 的请求，返回发送给该用户的全部聊天消息。

## 三、其他功能

除上述功能之外，本课题研究的基于 iOS 的金融数据分析软件用户管理子系统还包

括情报管理、用户评论、积分等功能，其中涉及到的客户端-服务器-数据库间通信过程与上述所提到功能的实现过程基本类似，在此不再赘述。

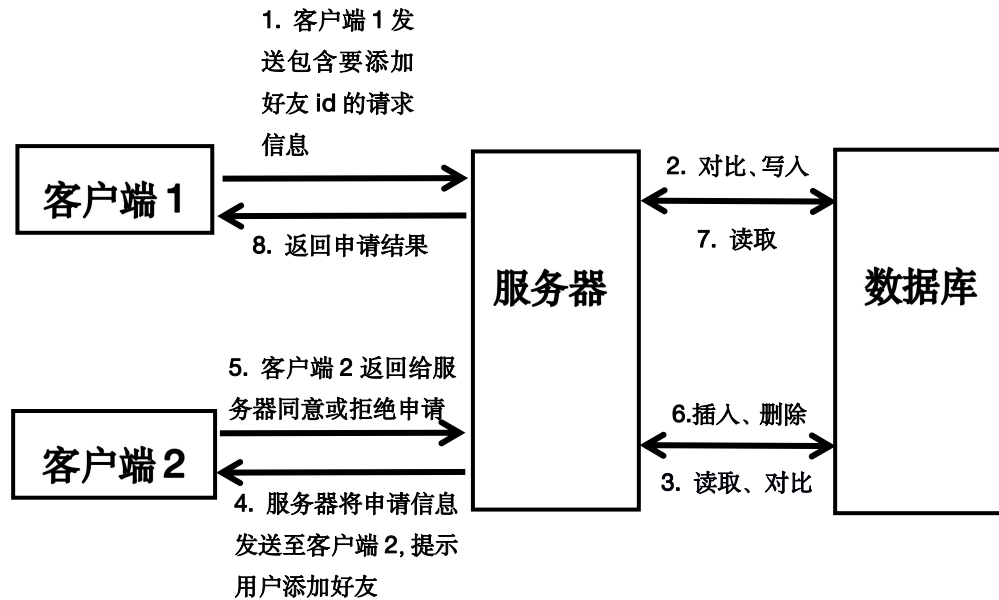


图 11 添加好友

## 第五章 基于 iOS 的金融数据分析软件用户管理子系统的详细设计

金融数据分析软件用户管理子系统涉及到三个部分的实现，分别为数据库表的设计和建立、服务器端的接口设计和数据库操作的实现，以及客户端 UI 界面设计及前后端通信的实现，下面将对本课题涉及到的客户端 UI 界面设计及前后端通信的实现做详细说明。

### 第一节 UI 界面的实现

本课题基于 iOS 的金融数据分析软件用户管理子系统的客户端共涉及到 15 个界面，分别用于实现用户注册、登陆、编辑个人信息、好友通信、浏览情报、评论情报等功能。在 Storyboard<sup>[7]</sup>中，各个界面之间的交互关系如图 12 所示。

当用户打开金融数据分析软件时，将进入用户管理子系统的登陆界面，在登陆界面插入一个导航控制器（Navigation Controller），将该导航控制器设置为初始界面并自动跳转到登陆界面。

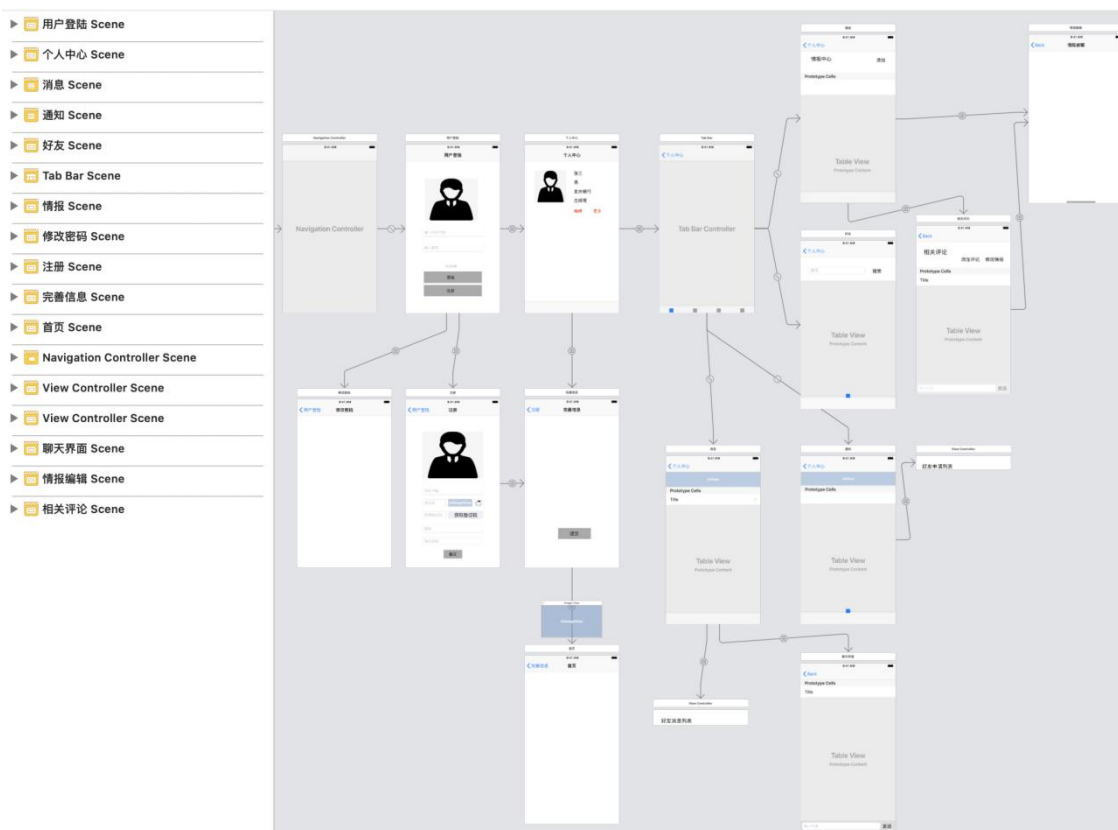


图 12 界面交互示意图

#### 一、用户登陆

用户登陆界面绑定 LoginViewController 类，使用 View Controller，并向其中添加相关控件，如图 13 所示。



表 24 对界面中的主要控件进行了说明。

表 24 登陆控件说明

控制器	控件	控件名称	功能	属性
View Controller	Image View	login	显示图片 美化界面	
	Text Filed	Password	用户输入手机号	Enabled = true; Placeholder = 输入手机号码;
		Username	用户输入密码	Enabled = true; Placeholder = 输入密码;
	Button	登陆	实现登陆	完成手机号和密码输入后, Enabled = true; 跳转->个人中心界面;
		注册	跳转到注册界面	Enabled = true; 跳转->注册界面;

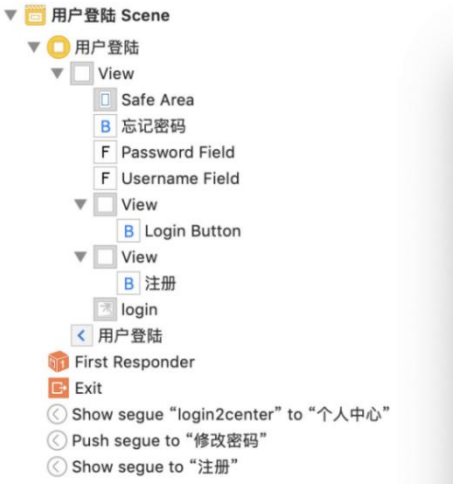


图 13 用户登陆控件

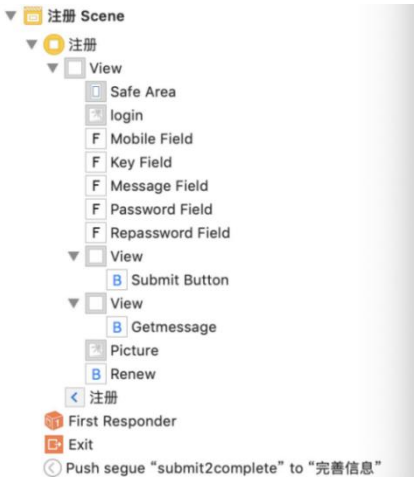


图 14 用户注册控件

二、用户注册

用户注册界面绑定 SubmitViewController 类，使用 View Controller，并向其中添加相关控件，如图 14 所示。表 25 对界面中的主要控件进行了说明。

表 25 用户注册控件说明表

控制器	控件	控件名称	功能	属性
View Controller	Image View	login	显示图片	
		Picture	显示图片验证码	
	Text Filed	Mobile	用户输入手机号	Enabled = true; Placeholder = 手机号码;
		Key	用户输入图片验证 码	Enabled = true; Placeholder = 验证码;

续表 25 用户注册控件说明表

		Message	用户输入手机验证码	Enabled = true; Placeholder = 短信验证码;
		Password	用户输入密码	Enabled = true; Placeholder = 密码;
		Repassword	再次输入密码	Enabled = true; Placeholder = 确认密码;
	Button	Submit	提交注册信息	完成全部 Text Filed 的输入后, Enabled = true; 跳转->完善信息界面;
		Renew	重新获取图片验证码	Enabled = true; Action->更新图片验证码;
		Getmessage	获取短信验证码	Enabled = true; Action->再次发送短信验证码;

三、完善信息

完善信息界面绑定 completeViewController 类，使用 View Controller，并向其中添加相关控件，如图 15 所示。使用代码实现向界面中添加 7 行 2 列的表格，并向表格第一列中依次添加 UILabel，Label 内容分别为姓名、金融机构、部门、职位、工作职责、办公电话和邮件。其中部门和职位对应的表格中分别添加一个 Button，用来显示提供的选择项。其余对应表格用 storyboard 拖拽的方法添加 Text Filed 控件，用于输入对应信息，并设置其约束使其与表格大小相适应，设置 Text Filed 的 Placeholder 属性为空，Enabled 属性均为 true。最后，向界面中添加 Subinfo Button 控件，用于信息提交，当且仅当 Text Filed 全部有输入值时设置 Subinfo 按钮的 Enabled 属性为 true。



图 15 完善信息控件

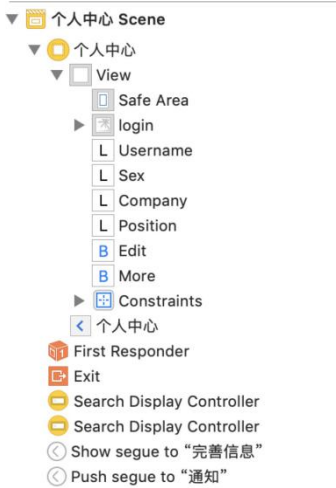


图 16 个人中心控件

## 四、个人中心

个人中心界面绑定 CenterViewController 类，使用 View Controller，并向其中添加相关控件，如图 16 所示。表 26 对界面中的主要控件进行了说明。

表 26 个人中心控件说明表

控制器	控件	控件名称	功能	属性
View Controller	Image View	login	显示图片	
	Label	Username	显示用户姓名	Enabled = true;
		Sex	显示性别	Enabled = true;
		Company	显示金融部门	Enabled = true;
		Position	显示职位	Enabled = true;
	Button	Edit	完善个人信息	Enabled = true; 跳转->完善信息界面;
		More	查看更多功能	Enabled = true; 跳转->Tab Bar Controller 界面;

## 五、标签控制器

好友、情报和评论等相关界面属于同一标签控制器。控制器包含四个导航项，分别为“好友”、“通知”、“消息”和“情报”。下面将对各导航内包含的界面进行详细介绍。

### （一）好友

搜索好友界面绑定 friendViewController 类，使用 View Controller，并向其中添加 Table View 和 Table View Cell 控件以便以列表的形式显示用户信息，内容包括用户姓名、所属金融机构、职位、工作职责、电话和邮箱。将 Table View Cell 控件的 Identifier 属性设置为“friends”，allowsSelection 属性设置为 YES，当选中某一 Cell 时弹出“加为好友”提示，用户可以选择“确定”或“取消”。最后将该控制器设置为 Table View 的数据源和代理。

界面中添加一个 Text View 控件用来输入搜索条件——金融部门名称，将其 Enabled 属性设置为 true，Placeholder 属性设置为“搜索”。之后用户点击 Dosearch Button，得到服务器返回结果并将符合条件的用户信息显示在界面中。

### （二）通知

好友申请界面绑定 messageViewController 类，使用 View Controller，并向其中添加 Table View 和 Table View Cell 控件以便以列表的形式显示好友申请信息，其中 Table View Cell 控件的 Identifier 属性设置为“friendrequest”，Style 属性设置为 Custom，allowsSelection 属性设置为 YES，当选中某一 Cell 时弹出“同意加为好友”提示，用户可以选择“同意”或“忽略”。最后将该控制器设置为 Table View 的数据源和代理。界面中仅添加一个 UIView 控件用来显示 Label——好友申请列表。

### （三）消息

#### （1）消息界面

消息申请界面绑定 chatTableViewController 类，使用 View Controller，并向其中添加 Table View 和 Table View Cell 控件以便以列表的形式显示用户的所有好友，其中 Table View Cell 控件的 Identifier 属性设置为“chat”，Style 属性设置为 Basic，allowsSelection 属性设置为 YES，当选中某一 Cell 时进入聊天界面。最后将该控制器设置为 Table View 的数据源和代理。界面中仅添加一个 UIView 控件用来显示 Label——好友消息列表。

#### （2）聊天界面

聊天界面绑定 chatpageViewController 类，使用 View Controller，并向其中添加 Table View 和 Table View Cell 控件以便显示聊天记录，其内容包括消息发送者姓名、发送时间和消息内容。将 Table View Cell 控件的 Identifier 属性设置为“message”，Style 属性设置为 Basic，allowsSelection 属性设置为 NO，即 Cell 不可选中。最后将该控制器设置为 Table View 的数据源和代理。

将控件 Send Button 和 Input Text Filed 放入同一 View 中，并将 View 添加到界面中以便后续统一管理。将 Input Text Filed 的 Enabled 属性设置为 true，Placeholder 属性设置为“输入内容”。Send Button 的 Enabled 属性设置为 False，当且仅当 Input Text Filed 添加了输入内容时，Enabled 属性更改为 true。其中键盘处于隐藏状态，当且仅当选中 Input Text Filed 时，弹出键盘。并在滚动界面时，自动收回。

控制器中控件的详细结构如图 17 所示。

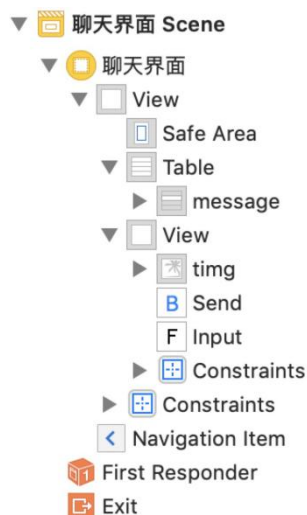


图 17 聊天界面控件

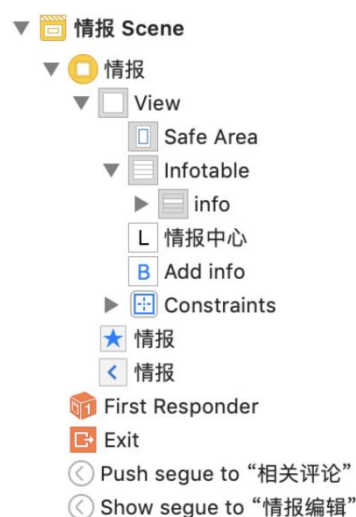


图 18 情报中心控件

### （四）情报

#### （1）情报中心

情报中心界面绑定 informationController 类，使用 View Controller，并向其中添加 Table View 和 Table View Cell 控件以便以列表的形式显示数据库中所有企业的情报信息，内容包括情报等级、发行人、发行场所、发行日期、注册金额、发行金额、分级、票面利率、期限、担保方式和积分等相关信息。将 Table View Cell 控件的 Identifier 属性设置为“info”，

Style 属性设置为 Custom, allowsSelection 属性设置为 YES, 当选中某一 Cell 时进入相关评论界面。最后将该控制器设置为 Table View 的数据源和代理。

界面中添加一个 Label 控件用来显示标题——情报中心, 以及一个 Button 控件, 其 Title 属性为“添加”, 若用户点击该按钮则跳转到情报编辑页面。

控制器中控件的详细结构如图 18 所示。

### (2) 情报编辑

情报编辑界面绑定 completeinfoController 类, 使用 View Controller, 并向其中添加相关控件, 如图 19 所示。

使用代码实现向界面中添加 11 行 2 列的表格, 并向表格第一列中依次添加 UILabel, Label 内容分别为发行人、发行日期、发行场所、注册金额、发行金额、分级、票面利率、期限、担保方式、情报等级和积分。其中发行场所对应的表格中添加一个 Button, 用来显示提供的选择项。其余对应表格用 storyboard 拖拽的方法添加 Text Filed 控件, 用于输入对应信息, 并设置其约束使其与表格大小相适应, 设置 Text Filed 的 Placeholder 属性为空, Enabled 属性均为 true。最后, 向界面中添加 Info submit Button 控件, 用于信息提交, 当且仅当 Text Filed 全部有输入值时设置 Subinfo 按钮的 Enabled 属性为 true。

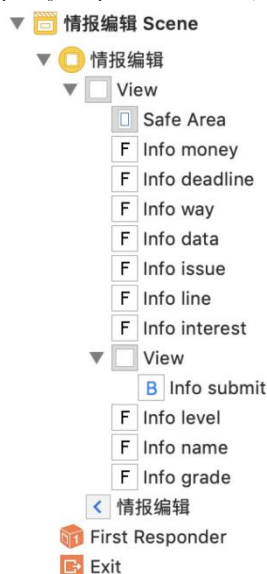


图 19 情报编辑界面控件



图 20 相关评论界面控件

### (3) 相关评论

相关评论界面绑定 CommentsController 类, 使用 View Controller, 并向其中添加 Table View 和 Table View Cell 控件以显示某条情报的相关评论, 包括评论人姓名、评论内容和评论时间等相关信息。将 Table View Cell 控件的 Identifier 属性设置为“comment”, Style 属性设置为 Basic, allowsSelection 属性设置为 NO, 即 Cell 不可选中。最后将该控制器设置为 Table View 的数据源和代理。

界面中添加一个 Label 控件用来显示标题——相关评论, 以及两个 Button 控件, 分别为“修改情报”和“添加评论”按钮。用户点击“修改情报”进入情报编辑页面, 点击“添加评论”弹出文本输入框, 用户可以添加评论。



为评论功能添加文本输入框和发送按钮，如图 20 中 Addview 所示，其构造方法类似聊天界面中的 View，在此不作赘述。唯一不同即将 Addview 的 hidden 属性设置为 YES，监听“添加评论”按钮，当用户点击该按钮时，将其 hidden 属性更改为 NO，Addview 弹出。

## 第二节 数据库的实现

用户管理子系统共涉及 6 张数据表，分别为用户表、好友管理表、好友请求表、消息收发表、情报表和评论表，对应以下图 21 到图 26 所示。图中显示为表中各字段对应的详细信息。

### 一、用户表

用户表存储系统中的所有用户信息。表中 user\_id 为用户身份的唯一标识，其他字段分别为电话、邮件、昵称、真实姓名、密码等信息。

user_id	mobile	email	nick_name	real_name	pwd_salt	pwd_hash	create_time	modify_time	user_state
0549c299bfac311ea13810790804		(Null)	(Null)	(Null)	39d75deb48095812	-hqR電#玻□J?g筱窃厝	2019-01-06 18:06:10	(Null)	NORMAL
05529bdeb6ab3ae813366903899		zhaoqi@citicleasing.	none	赵奇	5d27586b45f874e7e櫲-l▲傷?□()  □嚙▲??1	2019-03-07 08:19:14	(Null)		NORMAL
055c129a194311e9f13520895270		(Null)	(Null)	(Null)	(Null)	(Null)	2019-01-16 11:58:45	(Null)	NORMAL
14e3282e33c233bd18501088862		yhh@ccb.com.sg	none	杨旭炫	76ad3a4fcd2a9cf81€駢k?蓋置▲becMM駢	2019-04-18 20:29:10	(Null)		NORMAL
20cd8223ea973762f18811135251		110@123.com	none	ceshi2	6f456356d15188f08 東?臨蟻賠 (長€ DA	2019-04-08 09:31:04	(Null)		NORMAL
2514bb56083c3c7fb13681118720		(Null)	(Null)	(Null)	44eda71854ec0cb6t5□?旬V痼博□? )▲嚙	2019-01-13 19:42:25	(Null)		NORMAL
29b12c42c5ac3560a13991723133		476025715@qq.com	none	llw	205f2e99339c3fc87?W□鰱?駛\bs/?T?fi體	2019-01-19 14:35:21	(Null)		NORMAL
2a7a3264900b33cd15732153013		(Null)	(Null)	(Null)	f59bcdd86b3f65325 ?!域旁?h□竺:??ysN疋	2019-01-15 13:10:28	(Null)		NORMAL
2e9c6b6a19ec11e9t1860040462		(Null)	(Null)	(Null)	(Null)	(Null)	2019-01-17 08:09:40	(Null)	NORMAL
3381ead1daf33941b18614086048		zhuwenjun@bjut.edu	none	朱文军	20bca49a2914136a3A約D動\潑x駢埃^梁□	2019-01-02 19:00:14	(Null)		NORMAL

图 21 用户表

### 二、好友管理表

好友管理表存储系统中的好友关系，其中好友关系是双向的，为了防止好友双方二次添加，需要在表中写入两次。

id	userid1	userid2
5	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcbe36d2e93df
6	f07990cb428b3402954fcbe36d2e93df	d523200ae65e3aa096f08879b5265c7e
7	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcbe36d2e93df
8	f07990cb428b3402954fcbe36d2e93df	d523200ae65e3aa096f08879b5265c7e

图 22 好友管理表

### 三、好友请求表

好友请求表用来存储好友请求信息，字段分别为请求者 id 和发送者 id，其中 sign 用来标记本条好友关系是否完成建立。

requestid	sendid	recid	sign
1	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcbe36d2e93df	1
4	f07990cb428b3402954fcbe36d2e93df	d523200ae65e3aa096f08879b5265c7e	1
5	4421592b5ee93254a605386f2135ecb2	05529bdeb6ab3ae8bba065abd3651168	0
6	4421592b5ee93254a605386f2135ecb2	14e3282e33c233bd9d43865aec1afef3	0

图 23 好友请求表



## 四、消息收发表

消息收发表用来存储好友消息。表中字段分别代表发送者 id、接受者 id、发送时间、发送内容和成功标记。

msgid	sendid	recid	time	message	succeed
1	f07990cb428b3402954fcb36d2e93df	d523200ae65e3aa096f08879b5265c7e	2019-04-13 13:05:44	你好同学	1
2	f07990cb428b3402954fcb36d2e93df	323	2019-04-13 13:06:04	你好同学	0
3	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcb36d2e93df	2019-04-26 13:40:17	你好!	1
4	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcb36d2e93df	2019-05-01 14:00:46	今天星期几?	1
5	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcb36d2e93df	2019-05-01 14:03:09	你今天过得怎么样? 还	1
6	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcb36d2e93df	2019-05-01 14:09:49	行吧	1
7	d523200ae65e3aa096f08879b5265c7e	f07990cb428b3402954fcb36d2e93df	2019-05-01 14:21:21	不可以	1

图 24 消息收发表

## 五、情报表

情报表用来存储系统中全部的情报相关信息。

u_id	p_date	t_level	fin_borrower	asset_comp	asset_plan	asset_plan_type_id	inv_agency	filing_id	filing
a1f37dea17af11e99c2019-01-16 11:44:53	(Null)		中石油(北京)科技开发	爱建证券	信托资管	3	北京传家堡资产管理有	6,5,4,3	证券基金协
ab87f19217af11e9b2019-01-14 12:00:02	(Null)		上海上实霞飞日化销售	国投创新投资管理有限	保险资管	2	国投创新投资管理有限	1	中国信托登
a1f37dea17af11e99c2019-01-14 15:03:13	公开数据且可考证		湖北能源集团股份有限	北京大观投资管理有限	信托资管	3	和谐爱奇投资管理(北	1	中国信托登
2c5f05d017c411e9b2019-01-14 15:04:48	公开数据且可考证		阿里巴巴(中国)网络技	山东省高新技术创业投	保险资管	2	山东省高新技术创业投	1	中国信托登
2c5f05d017c411e9b2019-01-14 15:44:24	公开数据不可考证		北京市人民政府	北京大观投资管理有限	保险资管	2	北京东富汇通投资管理	1	中国信托登
2c5f05d017c411e9b2019-01-14 16:02:37	公开数据不可考证		上海建工(集团)总公司	光大金控(上海)投资管	基金资管	4	华安未来资产管理(上	1	中国信托登

图 25 情报表

## 六、评论表

评论表用来存储情报的评论信息。表中字段分别为情报唯一标识 uuid、评论 id、评论内容和评论时间。

uuid	comment_ids	comments	comment_time	id
08873b4c6cc311e999d400163e02e9cd	4421592b5ee93254a605386f2135ecb2	非常棒!	2019-05-02 22:20:37	1
20dad5c6cbf11e98d2200163e02e9cd	4421592b5ee93254a605386f2135ecb2	very Good!	2019-05-02 22:42:45	2
3b9929046cc411e999d400163e02e9cd	4421592b5ee93254a605386f2135ecb2	ksfjks	2019-05-05 11:21:47	3
53172e626ce911e9a62d00163e02e9cd	4421592b5ee93254a605386f2135ecb2	hkzjsa	2019-05-05 11:20:02	4
698169646ee311e9826900163e02e9cd	4421592b5ee93254a605386f2135ecb2	askha	2019-05-05 11:11:23	5
6e07f3526cc411e9995a00163e02e9cd	4421592b5ee93254a605386f2135ecb2	harsh	2019-05-05 11:11:38	6
c24771346b5111e9821c5800e345bbac	29b12c42c5ac3560a07738711a24456c	这情报靠谱!	2019-05-02 13:17:44	7
e849d8b86cc111e990bd00163e02e9cd	c1b1b51ea2343bd59313edb8310cb005	123456	2019-05-05 11:04:12	8
0166a31a6fe911e9b04100163e02e9cd	4421592b5ee93254a605386f2135ecb2	就是打开	2019-05-10 10:13:23	9
087fef066cf311e9a75000163e02e9cd	4421592b5ee93254a605386f2135ecb2	卡速度很快	2019-05-10 10:36:30	10
08873b4c6cc311e999d400163e02e9cd	4421592b5ee93254a605386f2135ecb2	Very good!	2019-05-16 22:26:29	11
08873b4c6cc311e999d400163e02e9cd	2ab586f6cfbc3748a2bb569a66b85908	这条情报非常有用!	2019-05-16 22:30:00	12
08873b4c6cc311e999d400163e02e9cd	c1b1b51ea2343bd59313edb8310cb005	这条情报根本没用!	2019-05-16 22:44:43	13

图 26 评论表

## 第三节 前后端通信的实现

上面仅针对用户界面和数据库的详细设计分别进行描述，下面将以表格的形式对其前后端通信的实现过程做详细分析，如表 27 所示。

表 27 通信接口说明表

界面	接口地址 (公共部分: http://47.92.50.218:8881/api1)	功能
用户登陆	/login	用户合法性检查
用户注册	/img_code	获取图形验证码
	/message_code	获取短信验证码
	/register	完成用户注册
完善信息	/personal_information	完善用户个人信息
个人中心	/show_user_info	获取用户个人信息
搜索好友	/search_users	搜索用户
好友列表	/friends_list	获取用户全部好友信息
好友申请	/send_friend_request	发送添加好友申请
	/friends_request_list	获取全部好友申请
	/receive_friend_request	建立好友关系
聊天界面	/receive_friends_message	接收消息
	/send_friends_message	发送消息
情报中心	/get_intelligence	获取企业情报
情报编辑	/insert_intelligence	向数据库中添加情报
相关评论	/comment_intelligence	添加评论

#### 第四节 功能流程图

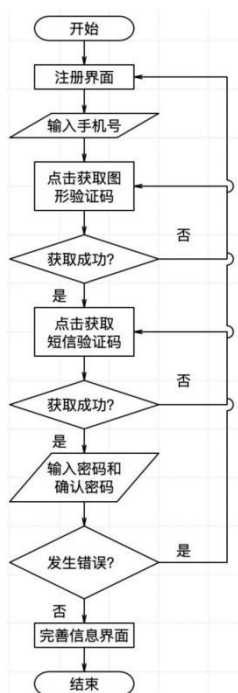


图 27 注册

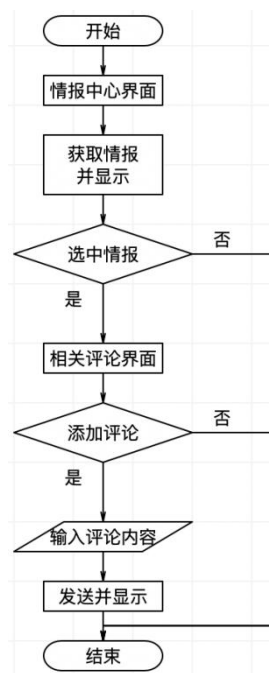


图 28 情报

## 一、注册

注册功能的实现分为三个部分：输入相关信息、数据库操作、页面跳转。输入信息中除手机号和密码外还包括图形验证码和短信验证码，获取验证码的过程若发生错误则需要重新获取。其中短信验证码的正确获取依赖于手机号和图形验证码输入正确。数据库对比包括验证码正确性检查和重复手机号检查，若输入信息全部无误，则发生页面跳转。详细流程如图 27 所示。

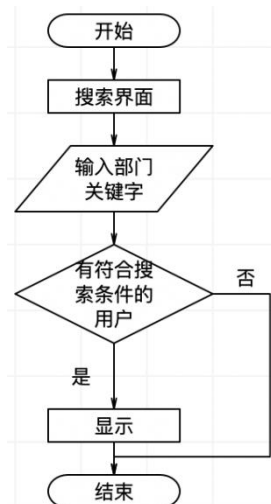


图 29 用户搜索

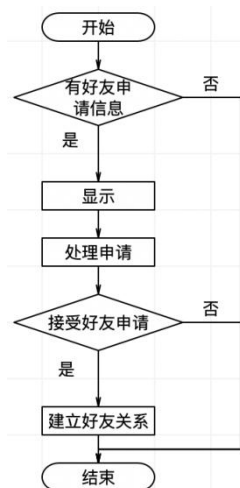


图 30 申请处理

## 二、登陆

登陆功能的实现分为三个部分：输入相关信息、数据库操作、页面跳转。输入信息包括手机号和密码。数据库对比用来检查手机号和密码是否匹配，若匹配成功，则发生页面跳转。详细流程如图 32 所示。

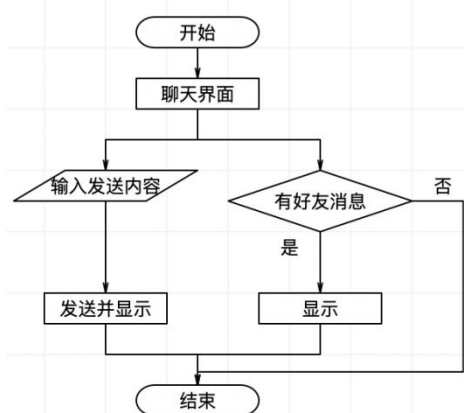


图 31 聊天

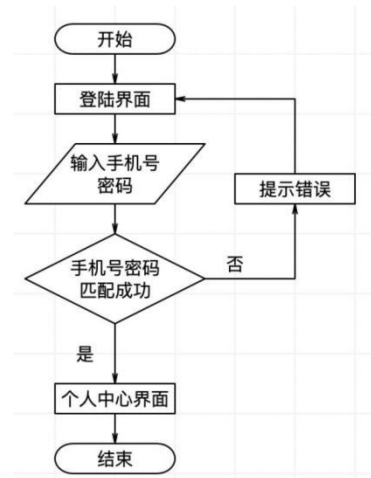


图 32 登陆

### 三、好友

好友管理功能的实现分为三个模块：搜索用户、添加好友、好友聊天。各模块详细流程如图 29-图 31 所示。

### 四、情报

情报管理功能包含了评论和积分模块的实现。详细流程如图 28 所示。

## 第五节 技术难点分析

在系统实现时，需要攻克一些技术难点，下面将对相关难点进行分析说明。在实现客户端用户管理子系统时，将涉及到网络连接、全局变量、定时器的使用、动态列表、键盘隐藏等相关技术难点，下面将进行详细分析。

### 一、iOS 中的网络连接

本系统的主要功能即是用户与服务器端的相关通信，所以，建立 iOS 客户端与服务端稳定的网络连接是开发软件过程中最基础的部分。iOS 中发送 HTTP 请求的方法有多种，本文涉及并介绍三种主要方法。分别为苹果原生的 `NSURLConnection`、`NSURLSession` 以及第三方库 `AFNetworking`。其中开源库涉及到在 Xcode 中使用 Pods 管理第三方库的相关技术。下面将分别对其进行详细的介绍和分析。

#### （一）`NSURLConnection`

`NSURLConnection` 是一个允许用户启动或停止 URL 请求的对象。通过提供 URL 请求对象，`NSURLConnection` 允许加载 URL 的内容。其接口是稀疏的，仅提供用来启动或取消 URL 请求的异步负载的控制。用户可以对 URL 请求对象本身执行大部分的配置工作。`NSURLConnection` 类提供了方便的类方法，可以使用回调块异步或同步加载 URL 请求<sup>[8]</sup>。

本课题中涉及到的 Post 方法均使用了 `NSURLConnection` 发送网络请求，其具体实现方法如下。

（1）创建一个 `NSURL` 对象，设置请求路径，如图 33 所示；

```
//获得网址名称存入url
NSURL *url=[[NSURL alloc]initWithString:@"http://47.92.50.218:8881/api1/register"];
```

图 33 设置请求路径

（2）设置传递参数并转换为 `NSData`，如图 34 所示；

```
NSMutableData *postBody=[NSMutableData data];

[postBody appendData:[NSString stringWithFormat:@"mobile=%@&pwd=%@&message_code=%@", self.mobileField.text,
self.passwordField.text,self.messageField.text] dataUsingEncoding:NSUTF8StringEncoding]];
```

图 34 设置参数及格式转换

（3）创建一个 `NSMutableURLRequest` 对象，设置请求头和请求体，如图 35 所示；



```
NSMutableURLRequest *request=[[NSMutableURLRequest alloc]initWithURL:url
    cachePolicy:NSURLRequestReloadIgnoringCacheData timeoutInterval:20.0f];

[request setHTTPMethod:@"POST"];

[request setValue:@"application/x-www-form-urlencoded" forHTTPHeaderField:@"Content-Type"];

[request setHTTPBody:postBody];

NSError *error = nil;

NSURLSessionResponse* urlResponse = nil;
```

图 35 创建请求对象

(3) 使用 `NSURLConnection` 发送异步请求，如图 36 所示。

```
NSData *responseData = [NSURLConnection sendSynchronousRequest:request returningResponse:&urlResponse error:&error];

NSDictionary *result = [NSJSONSerialization JSONObjectWithData:responseData options:NSJSONReadingMutableContainers
    error:nil];
```

图 36 发送请求

## (二) NSURLSession

`NSURLSession` 在 2013 年随着 iOS7 的发布一起面世，苹果对它的定位是作为 `NSURLConnection` 的替代者，然后逐步将 `NSURLConnection` 退出历史舞台。现在使用最广泛的第三方网络框架也都使用了 `NSURLSession`。其核心就是对网络任务进行封装，实现多线程。比如将一个网络请求交给 `NSURLSession`，最后 `NSURLSession` 将访问结果通过 block 回调返回，期间自动实现多线程，而且可以通过代理实现监听<sup>[8]</sup>。

本课题中涉及到的 Get 方法均使用了 `NSURLSession` 发送网络请求，其具体实现方法如下。

(1) 创建请求路径，如图 37 所示；

```
NSString *str = @"http://47.92.50.218:8881/api/show_user_info";
NSString *strurl = [str stringByAddingPercentEncodingWithAllowedCharacters:[NSCharacterSet URLQueryAllowedCharacterSet]];
NSURL *url = [NSURL URLWithString:strurl];
```

图 37 创建请求路径

(2) 创建请求对象，包含请求头和请求方法，如图 38 所示；

```
//创建可变的请求对象
NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url];
//发送get请求
request.HTTPMethod = @"get";
AppDelegate *myDelegate = (AppDelegate*)[UIApplication sharedApplication].delegate;
//get请求头
[request setValue:[NSString stringWithFormat:@"%s", myDelegate.token_id] forHTTPHeaderField:@"token-id"];
```

图 38 创建请求对象

(3) 创建会话对象，根据会话对象创建请求任务并执行 Task，如图 39 所示；

```
//获得会话对象
NSURLSession *session = [NSURLSession sharedSession];
//发送请求
NSURLSessionDataTask *dataTask = [session dataTaskWithRequest:request completionHandler:^(NSData * _Nullable data,
    NSURLResponse * _Nullable response, NSError * _Nullable error) {

    //解析数据
    NSDictionary *dict = [NSJSONSerialization JSONObjectWithData:data options:NSJSONReadingMutableContainers error:nil];
}];
//7.执行任务
[dataTask resume];
```

图 39 发送请求

### （三）第三方库

#### 1. CocoaPods 相关技术

本课题中注册功能的实现需要通过网络传输获取图形验证码，其中使用到了第三方开源库——AFNetworking。CocoaPods 是 iOS 最常用的类库管理工具，通过 CocoaPods 只需要相关命令行命令就可以解决类库导入、更新、下载等问题。

CocoaPods 的使用十分方便和简单，只需在命令行进入到对应项目目录下，利用 vim 命令创建 Podfile 文件，并在 Podfile 文件中写入相关下载信息并保存，最后使用 pod install 命令便可将第三方库导入到项目。

成功安装 CocoaPods，并导入 AFNetworking 和 ZJScrollView 第三方库的项目目录结构如图 40 所示。

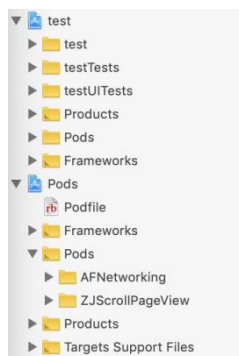


图 40 项目目录结构

#### 2. AFNetworking

AFNetworking 是 iOS 开发中最常用的第三方开源库之一，构建于熟悉的 Foundation 技术之上。其主要用于进行网络请求，是封装的 NSURLSession 的网络请求。AFNetworking 由 NSURLSession, Security, Reachability, Serialization, UIKit 五个模块组成，包括 NSURLSessionDataTask, NSURLSessionDownloadTask, NSURLSessionUploadTask 三种任务类型。AFNetworking 的请求格式有三种，分别为 AFHTTPRequestSerializer、AFJSONRequestSerializer 和 AFPropertyListRequestSerializer；返回类型有七种，分别为 AFHTTPResponseSerializer、AFJSONResponseSerializer、AFXMLParserResponseSerializer、AFXMLDocumentResponseSerializer、AFPropertyListResponseSerializer、AFImageResponseSerializer 和 AFCompoundResponseSerializer。

使用 AFNetworking 第三方开源库，分别将任务类型设置为 NSURLSessionDataTask，请求格式设置为 AFHTTPRequestSerializer，网络请求的返回格式设置为



AFImageResponseSerializer，可以简单快捷的实现用户管理子系统中用户权限模块的注册功能涉及到的图片验证码的获取。

使用 AFNetworking 发送请求的具体实现方法如下。

(1) 设置请求路径，如图 41 所示；

```
NSString *urlString = @"http://47.92.50.218:8881/api1/img_code";
```

图 41 设置请求路径

(2) 创建 AFHTTPSessionManger，并对其进行初始化，设置请求格式和返回格式如图 42 所示；

```
//1.创建管理者
AFHTTPSessionManager *manager = [AFHTTPSessionManager manager];
//2.设置请求格式
manager.requestSerializer = [AFHTTPRequestSerializer serializer];
//3.设置接受的响应数据类型
manager.responseSerializer = [AFImageResponseSerializer serializer];
```

图 42 设置相关请求方式

(3) 创建网络请求，如图 43 所示；

```
//做get请求
[manager GET:urlString parameters:nil success:^(NSURLSessionDataTask * _Nonnull task, id
_Nullable responseObject) {
    //解析数据

} failure:^(NSURLSessionDataTask * _Nullable task, NSError * _Nonnull error) {
}];
```

图 43 创建网络请求

## 二、全局变量的声明和使用

由于有关用户管理子系统的大多数接口在访问的时候需要发送表明用户身份的 token\_id 参数，为了方便使用需要将 token\_id 定为全局变量，因此需要掌握全局变量声明和使用的规则。此外，当在某一方法中向服务器端发送请求并获得返回值后，需在别的方法中调用该返回值，也需要在程序中使用到全局变量。下面，将对系统中全局变量的声明和使用作详细说明。

全局变量的声明添加在 AppDelegate.h 中，方法如图 44 所示。

```
#import <UIKit/UIKit.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate>
@property (nonatomic, strong) NSString *token_id;
@end
```

图 44 全局变量的声明

在相关程序包含 AppDelegate.h 头文件的情况下，全局变量可以通过单例模式在系统

的任意界面获取并完成对其赋值或调用，方法如图 45、46 所示。

//对全局变量赋值

```
AppDelegate *myDelegate = (AppDelegate*)[UIApplication sharedApplication].delegate;
myDelegate.token_id = result[@"token_id"];
```

图 45 全局变量的赋值

```
AppDelegate *myDelegate = (AppDelegate*)[UIApplication sharedApplication].delegate;
//get请求头
[request setValue:[NSString stringWithFormat:@"%d", myDelegate.token_id] forHTTPHeaderField:@"token-id"];
```

图 46 全局变量的使用

### 三、定时器的使用

用户管理子系统中用户权限模块的注册功能涉及到短信验证码的获取，其中第一次点击“获取验证码”之后，Button 样式应改变为倒计时按钮。该功能的实现需要用到 NSTimer 定时器<sup>[9]</sup>。

具体的实现思路如下，代码实现如图 47 所示：

- (1) 创建原始按钮，样式为“获取验证码”，并为其添加点击方法；
- (2) 用 NSTimer 定时器，每秒执行一次；
- (3) 若倒计时未结束，改变按钮的样式为“重新发送（剩余秒数）”并设置按钮不可点击；
- (4) 若倒计时结束，则关闭定时器，将按钮样式设置为“重新发送”并设置按钮可以点击。

```
__block NSInteger time = 59; //倒计时时间
NSInteger len = self.mobileField.text.length;
dispatch_queue_t queue = dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
dispatch_source_t _timer = dispatch_source_create(DISPATCH_SOURCE_TYPE_TIMER, 0, 0, queue);

dispatch_source_set_timer(_timer, dispatch_walltime(NULL, 0), 1.0*NSEC_PER_SEC, 0); //每秒执行
dispatch_source_set_event_handler(_timer, ^{

    if(time <= 0 || !b || len < 11){ //倒计时结束，关闭
        dispatch_source_cancel(_timer);
        dispatch_async(dispatch_get_main_queue(), ^{
            //设置按钮的样式
            [self.getMessage setTitle:@"重新发送" forState:UIControlStateNormal];
            [self.getMessage setTitleColor:[UIColor darkGrayColor] forState:UIControlStateNormal];
            self.getMessage.userInteractionEnabled = YES;
        });
    }else{
        int seconds = time % 60;
        dispatch_async(dispatch_get_main_queue(), ^{
            //设置按钮显示读秒效果
            [self.getMessage setTitle:[NSString stringWithFormat:@"重新发送(%.2ds)", seconds]
                forState:UIControlStateNormal];
            [self.getMessage setTitleColor:[UIColor darkGrayColor] forState:UIControlStateNormal];
            self.getMessage.userInteractionEnabled = NO;
        });
        time--;
    }
});
dispatch_resume(_timer);
```

图 47 定时器的代码实现

## 四、动态列表

在用户管理子系统中，多个界面的控制器使用到 tableView，如情报中心、好友列表、聊天界面、评论中心等。为了便于列表的动态实现，本文将用到 Cell 的数据源和代理方法<sup>[10]</sup>。

下面将以好友列表界面为例介绍相关方法的实现。用户登陆后，进入到好友列表界面，该界面通过从数据库中获取到当前用户的好友名单，然后将其显示在动态的 tableView 中，实现方法如下：

- (1) 为 Cell 数据源创建模型 Friendlist，如图 48 所示。其中方法具体实现如图 49 所示；

```
#import <Foundation/Foundation.h>
NS_ASSUME_NONNULL_BEGIN

@interface Friendlist : NSObject
@property(nonatomic, copy) NSString *title;
@property(nonatomic, copy) NSString *user_id;

+ (instancetype)newsWitharray:(NSArray *)dic;

@end
NS_ASSUME_NONNULL_END
```

图 48 模型

```
#import "Friendlist.h"
@implementation Friendlist
+ (instancetype)newsWitharray:(NSArray *)dic{
    Friendlist *news = [self new];
    news.title = dic[0];
    news.user_id = dic[1];
    return news;
}

- (void)setValue:(id)value forKey:(NSString *)key{
}

@end
```

图 49 模型方法的实现

- (2) 设置代理方法，导入模型头文件，并声明数组变量 friliat，如图 50 所示；

```
#import "chatTableViewCell.h"
#import "AppDelegate.h"
#import "Friendlist.h"

@interface chatTableViewCell() <UITableViewDataSource, UITableViewDelegate>

@property(nonatomic, strong) NSArray *friliat;

@end
```

图 50 设置代理方法

- (3) 将好友名单通过模型实例 news 存入 friliat 数组中，如图 51 所示；

```

NSArray *array = dict[@"request_list"];
NSMutableArray *mArray = [NSMutableArray arrayWithCapacity:100];
[array enumerateObjectsUsingBlock:^(id _Nonnull obj, NSUInteger idx, BOOL * _Nonnull stop) {
    Friendlist *news = [Friendlist newsWitharray:obj];
    [mArray addObject:news];
}];
self.friliat = mArray.copy;

```

图 51 存储模型数据

(4) 然后在 tableView 的 cellForRowAtIndexPath 方法中, 设置 cell 中的 text 显示为 friliat 数组中的每一行内容, 如图 52 所示;

```

-(UITableView *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath{
    static NSString *reuseId = @"chat";
    UITableViewCell * cell = [tableView dequeueReusableCellWithIdentifier:reuseId];
    if(cell == nil){
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:reuseId];
    }
    Friendlist *news = self.friliat[indexPath.row];
    cell.textLabel.text = news.title;
    return cell;
}

```

图 52 cellForRowAtIndexPath 方法

(5) 在 tableView 的 numberOfRowsInSection 方法中返回 friliat 数组的元素个数, 以加载列表行数, 如图 53 所示;

```

-(NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section{
    return self.friliat.count;
}

```

图

53 numberOfRowsInSection 方法

(6) 为了使数据库数据发生变化时, 列表随之改变。需在数组的重置方法中进行列表数据的重载, 如图 54 所示。

```

-(void)setFriliat:(NSArray *)friliat{
    _friliat = friliat;
    dispatch_async(dispatch_get_main_queue(), ^{
        [self.tableView reloadData];
    });
}

```

图 54 列表重载

## 五、键盘隐藏

用户管理子系统中好友管理模块中的消息收发功能中涉及到聊天界面的实现, 当输入框处于屏幕下方时, 若不进行键盘隐藏等相关操作处理, 编辑 UITextField 的时候, 弹出键盘就会挡住一部分界面。通过让界面随键盘一起移动的思想方法, 就可以解决键盘挡住界面的问题。键盘的位置由横坐标 x 与纵坐标 y 来确定, 由于 x 值不会发生改变, 只需关注其纵坐标即可。具体实现方法为每当需要改变键盘位置时, 首先获取当前键盘位置的纵坐标, 与屏幕高度做差计算出下一时刻键盘的 y 值, 从而确定键盘位置。详细



代码实现如图 55 所示。

```

-(void)KeyboardDidChangeFrame:(NSNotification *)noti
{
    //NSLog(@"-----%@", noti.userInfo);
    //键盘改变时 改变window颜色
    self.view.window.backgroundColor = self.table.backgroundColor;

    CGRect frame = [noti.userInfo[UIKeyboardFrameEndUserInfoKey] CGRectValue];

    //键盘实时的y
    CGFloat keyY = frame.origin.y;

    CGFloat keyDuration = [noti.userInfo[UIKeyboardAnimationDurationUserInfoKey] floatValue];

    CGFloat screenH = [[UIScreen mainScreen]bounds].size.height;

    [UIView animateWithDuration:keyDuration animations:^(
        self.view.transform = CGAffineTransformMakeTranslation(0, keyY - screenH);
    )];
}

```

图

55 键盘隐藏

当用户滑动聊天界面查看相关聊天记录时，键盘应当自动隐藏。相关代码在 tableView 的 scrollViewWillBeginDragging 方法中实现，如图 56 所示。

```

//滚动时 收回键盘
-(void)scrollViewWillBeginDragging:(UIScrollView *)scrollView
{
    [self.view endEditing:YES];
}

```

图 56 收回键盘

## 六、控件相关技术

Storyboard 作为 UI 界面开发的重要工具使用起来十分的方便简洁，但 storyboard 对控件的相关设置会过于死板不易修改，为了使用户体验更加的友好，用代码对控件的一些细节处理必不可少。下面将简单介绍本课题所涉及到的相关技术。

### （一）UIButton 的 enabled 属性

实现用户的登陆功能时，用户必须填写用户名和密码，监听文本框只有当必填信息全部填写的状态下登陆按钮才可使用，否则登陆按钮为不可点击状态。实现方法如图 57。

```

//文本框内容发生改变的时候调用
-(void)textChange{
    self.loginButton.enabled = self.usernameField.text.length > 0 && self.passwordField.text.length > 0;
}

```

图 57 UIButton 的 enabled 属性

### （二）UITableView 的 allowsSelection 属性

实现用户聊天界面时使用了 tableView 相关技术，但此时 Cell 应当为不可选中的状态。在函数 viewDidLoad 中相关代码实现如图 58 所示。

```
//cell不可选中  
self.table.allowsSelection = NO;
```

图 58 UITableView 的 allowsSelection 属性

### (三) UIView 的 hidden 属性

图 59 为相关评论界面的控制器，界面显示的是某条情报的所有评论，界面下方的文本框和“发送”按钮应当在用户点击“添加评论”按钮之后弹出。该技术的具体实现方法为将文本框和按钮添加到同一 view 中，对“添加评论”按钮进行监听，在其方法中对 view 的 hidden 属性做相关操作，如图 60 所示。

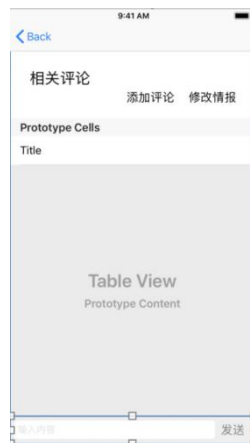


图 59 评论界面

```
-(void)addinfo{  
    self.addview.hidden = NO;  
}
```

图 60 UIView 的 hidden 属性



## 第六章 基于 iOS 的金融数据分析软件用户管理子系统的部署及测试

本章主要介绍基于 iOS 的用户管理子系统在运行测试前的系统部署以及相关功能的测试要求概述与测试结果展示。

### 第一节 系统部署

由于将系统放于真机上调试须注册苹果开发者帐号并缴纳一定费用，所以本课题所实现的系统客户端的测试通过在 PC 机上的 iOS 模拟器来完成。使用 Xcode 打开项目文件，点击左上角的“Run”键，即可以在模拟器中打开所编写的系统程序并进行测试。

### 第二节 系统测试

#### 一、测试概要

表 28 测试功能概要

测试功能	标准输入	期待输出
用户注册	手机号、图形验证码、短信验证码、密码、确认密码	成功：进入完善信息界面 失败：提示错误信息
用户登陆	手机号、密码 点击“登陆”	成功：进入个人中心界面 失败：提示错误信息
用户注销	点击“确认注销”	返回登陆界面
完善信息	姓名、金融机构、部门、职位、工作职责、办公电话和邮件，点击提交	显示修改后的个人信息
个人中心	进入个人中心界面	显示用户个人信息
搜索好友	输入部门关键字，点击搜索	符合条件的用户信息
添加好友	选中某一好友并确认添加	请求发送成功或失败
申请列表	进入通知界面	显示申请信息
处理申请	选中某一申请信息，选择同意/忽略	添加成功/失败
好友列表	进入好友界面	显示好友列表
发送消息	输入聊天内容，点击发送	发送内容显示在双方界面
接收消息	进入消息界面	显示好友发送的消息
情报中心	进入情报界面	显示情报信息
情报编辑	发行人、发行日期、发行场所、注册金额、发行金额、分级、票面利率、期限、担保方式、情报等级和积分，点击提交	显示本条情报
查看评论	选中某一情报	进入评论界面并显示评论
添加评论	输入评论内容，点击发送	显示本条评论

本系统的测试以功能模块为单位依次进行，首先对各功能模块的标准输入和期待输出做详细说明，以便后续测试的展开。测试功能概要如表 28 所示。

## 二、测试结果

### 1. 用户注册

测试要求：在注册界面输入手机号、验证码、短信验证码、密码和确认密码，其中密码以“\*”显示。注册成功后，页面跳转至完善信息界面。如手机号已注册、验证码错误、短信验证码错误或两次输入密码不一致，则弹出相应错误提示框。

测试结果：该功能经测试可完全实现。图 64 为注册见面，图 61 为手机号已被注册时结果，图 62 为图形验证码错误时结果，图 63 为密码输入不一致时结果。



图 61 该用户已注册



图 62 手机号或图形验证码错误



图 63 密码输入不一致



图 64 注册界面



图 65 登陆界面



图 66 手机号密码不匹

## 2. 用户登陆

测试要求：在登陆界面输入手机号和密码，其中密码以“\*”显示。登陆成功后，界面跳转至个人中心。如手机号和密码不匹配，则弹出错误提示框。

测试结果：该功能经测试可完全实现。图 65 为登陆界面，图 66 为手机号密码不匹配时结果。

## 3. 用户注销

测试要求：点击注销按钮，弹出“确认注销”提示框。点击确认，返回登陆界面；点击取消，回到个人中心。

测试结果：该功能经测试可完全实现。其中图 67 为弹出注销提示框结果。



图 67 注销界面



图 68 完善信息



图 69 个人中心

## 4. 完善信息

测试要求：在完善信息界面输入相关信息，点击提交按钮。用户可完成信息完善，并在个人中心显示个人资料。

测试结果：该功能经测试可完全实现。图 68 为完善信息界面，图 67 为显示信息。

## 5. 个人中心

测试要求：进入个人中心界面，显示个人信息。

测试结果：该功能经测试可完全实现，具体结果见图 69。

## 6. 搜索好友

测试要求：输入“资产管理部”，显示符合搜索条件的用户信息。

测试结果：该功能经测试可完全实现，具体结果见图 70。

## 7. 添加好友

测试要求：选中某一好友，弹出“加为好友”提示，用户可以选择“取消”或“确定”。若用户选择确定，则向服务器端发送申请信息，对方用户上线后可接收到该信息。

测试结果：该功能经测试可完全实现，具体结果见图 71。

## 8. 处理申请消息

测试要求：选中某一申请消息，弹出“同意加为好友”提示，用户可以选择“同意”或“忽略”。若用户选择同意则建立好友关系，并可在各自的联系人列表显示对方信息。

测试结果：该功能经测试可完全实现，具体结果见图 72 并可在好友列表中看到已添加好友如图 73。



图 70 搜索好友

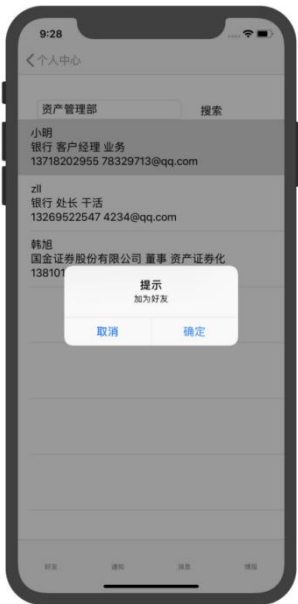


图 71 添加好友



图 72 处理申请消息

9. 聊天功能

测试要求：选中某一好友，进入聊天界面，可以和好友进行聊天并显示聊天内容。

测试结果：该功能经测试可完全实现，具体结果见图 74。

10. 情报中心

测试要求：进入情报页面可以浏览情报信息。

测试结果：该功能经测试可完全实现，具体结果见图 76。



图 73 消息列表



图 74 聊天界面



图 75 情报编辑

11. 相关评论

测试要求：进入某条情报的评论页面可以浏览情报的全部评论信息，包括评论人姓名、评论内容、评论时间。

测试结果：该功能经测试可完全实现，具体结果见图 78。

12. 情报编辑

测试要求：进入情报编辑页面，输入相关信息，点击提交。

测试结果：该功能经测试可完全实现。情报编辑界面见图 75，并可在情报中心看到该情报信息见图 76。

13. 添加评论

测试要求：点击“添加评论”，输入评论内容，点击发送，评论显示在界面上。

测试结果：该功能经测试可完全实现，具体结果见图 77 和 78。



图 76 情报中心



图 77 添加评论



图 78 显示评论

## 总结

随着互联网的高速发展，互联网日益成为了获取信息的重要渠道。为了更好地、安全的使银行全面了解企业的各类金融相关信息，需要构建相应金融数据分析软件，可将不同来源发布的各类金融信息整合后存放于数据库中，并根据相关数据和指标自动分析，其中包括用户管理子系统、融报管理子系统、商机管理子系统等。

本课题金融数据分析软件用户管理子系统在开发过程中以软件工程理论为指导，采用瀑布软件开发模型。UI 界面设计使用 Storyboard，使用约束限制确定控件位置并对页面中涉及的按钮表格添加自适应，使该系统可以适应不同 iOS 版本型号。代码进行良好的封装，以便对系统进行后期扩展和二次开发。

本文对基于 iOS 的金融数据分析软件用户管理子系统进行了相关背景分析和技术介绍、需求分析、整体结构分析、概要设计以及详细设计与实现。用户可在 iOS 客户端进行注册并登陆本系统，在系统中完善个人信息、添加其他用户为好友并实现好友聊天、浏览金融企业相关情报并对感兴趣的情报进行添加或评论。本系统所有从客户端输入的信息资料，均可通过服务器端写入数据库中，实现永久化存储。系统的全部功能现已基本实现，并分别针对每个模块及功能进行了详细的测试和分析，测试结果可基本满足用户需求。

由于开发时间和个人能力有限，本课题所研究的基于 iOS 的金融数据分析软件用户管理子系统仍存在许多需要改进和提高的地方。在界面设计方面，客户端界面中的按钮、标签等控件的样式过于简洁，界面设计风格和友好程度也有待提高；在功能实现方面，系统基本完成课题要求但仍具有一定限制，如没有实现用户头像的传输、用户评论无法删除等。

用户管理子系统作为金融数据分析软件中与用户关系最为密切的功能模块，其友好性直接关系到用户对整个软件的满意程度。希望今后能修改和完善系统中的不足，使用户管理子系统、融报管理子系统、商机管理子系统相得益彰，共同构成让用户满意的金融数据分析软件。



## 参考文献

1. (美) Jonathan Levin 著,郑思遥,房佩慈 译. 深入解析 Mac OS X & iOS 操作系统. 北京: 清华大学出版社,2014.3
2. Mark H. Goadrich, Michael P. Rogers. Smart smartphone development:iOS versus android. ACM New York,NY,USA,2011
3. (日)户根勤 著,周自恒 译.网络是怎样连接的. 北京: 人民邮电出版社,2017.1
4. (美)Fritz Ander 著,王东明 译. Xcode 实战: Apple 平台开发实用技术、技巧及最佳流程. 北京: 机械工业出版社,2016.1
5. (美) Stephen G.Kochan 著,林冀,范俊,朱亦欣 译. Objective-C 程序设计(第6版). 北京: 电子工业出版社,2014.7
6. Roger S Pressman,Bruce R.Maxim Dr. Software Engineering: A Practitioner's approach. McGraw-Hill Education,2014.1
7. 史昕,汤海波,闫珍. iOS 开发: 从零基础到精通. 北京: 清华大学出版社,2018.2
8. Zdziarski J A. Iphone open application development. O'Reilly, 2008
9. Alex Rozanski. How do I use NSTimer.  
<http://stackoverflow.com/questions/1449035/how-do-i-use-nstimer>. 2012.9
10. Dave Wooldridge,Michael Schneider. The Business of iPhone App Development,Apress 2010

## 致谢

本次本科毕业设计，从前期的准备到系统的研发再到最后论文的撰写，历时4个多月，这期间凝聚了很多人的帮助与关心。值此论文完成之际，谨向本科毕业设计工作进行期间所有给过我无私帮助的老师、同学和家人表示衷心的感谢。

首先要感谢我的毕设导师朱文军老师，在毕设期间每当我遇到疑难求助于朱老师时，无论是线上还是线下，是工作时间还是休息时间，朱老师都会第一时间回复并解答。在系统开发阶段，对于一些难以攻克的技术难点，朱老师的悉心指导和宝贵意见，不但给予我无限的信心也使我深受启发。在论文撰写阶段，朱老师耐心地对我的论文进行批改、审阅并提出修改意见，让我的论文质量有了很大的提高。此外，朱老师创新执着的科研作风和精益求精的治学态度，为我树立了求学以及做人的楷模。在此，谨向朱老师表示最诚挚的敬意！

其次，我要感谢在金融数据分析软件课题组中的所有人员，他们不仅在软件开发的技术上给予我宝贵的意见与帮助，更在生活中给了我很多的支持与鼓励。从他们身上，我学到了很多宝贵的东西。

最后，我想感谢我的家人，感谢他们在生活方面予以的关心与支持，让我可以无后顾之忧的潜心学习，是他们的爱使我的每一步都走得更加坚定并充满希望与力量。