编号：160409-20141450



****

毕 业 设 计

**题 目**：手机银行客户端系统的设计与实现

**学 院：**计算机科学与工程学院

**专 业：**软件工程

**年 级：**160409

**姓 名：**刘育超

**指导教师：**谷钰 讲师

**完成日期：**

# 摘 要

计算机和互联网技术的发展，传统的支付和理财方式也迎来了其新一轮的变革，“虚拟银行”、“线上交易”等词越来越多的出现在人们生活与交易中。本系统—手机银行客户端系统的开发与实现，其目的在于通过使用APP开发技术，最新的Java开发框架，数据库等新兴技术来实现将支付转账等可以不在线下进行，解决传统交易、支付受到场地、时间和安全等的限制。

该篇论文详细的描述了如何采用SpringBoot框架技术开发一个可以转账以及进行理财的手机银行客户端系统；实现登录，注册，支付，转账，申购，赎回、收益计算，后台管理等特色功能。本系统使用SpringBoot进行开发，使用Angular、Ionic前端框架实现系统的前后端分离，优化开发流程，提升了用户体验。

该篇论文从需求阶段开始，充分考虑到了虚拟银行系统的实际需求，详细描述了从分析，设计到实现的每一个步骤。首先，系统前后台相辅相成，设计了系统的前台“银行动账，产品筛选，申购赎回等交易操作的界面展示”；其次，设计了“虚拟银行后台管理系统”，用于管理系统中的理财产品的上线以及调整，对管理员进行管理；该系统按照需求分析和详细开发文档编码完成后，进行系统功能测试（黑盒测试），并使用Postman测试工具对前后端独立单元、接口进行测试，测试的前台页面展示和后台数据库的操作结果均达到预期的效果。该系统的设计和开发符合软件的开发流程和相关法规，并且能够满足用户对于一个线上交易的需要，具有一定的现实意义。

**关键词**： 手机银行客户端系统 SpringBoot Postman MySQL8 IntellijIDEA

# 目 录

[摘 要 I](#_Toc4218)

[目 录 I](#_Toc14158)

[Abstract III](#_Toc19916)

[第1章 绪 论 1](#_Toc6160)

[1.1课题的来源及意义 1](#_Toc17407)

[1.2国内外发展状况 1](#_Toc17455)

[1.3系统可行性研究 2](#_Toc2836)

[1.3.1经济可行性 2](#_Toc9652)

[1.3.2技术可行性 2](#_Toc30425)

[1.3.3社会可行性(法律可行性) 3](#_Toc5009)

[1.4本文的组织结构图 3](#_Toc29539)

[1.5本章小结 3](#_Toc16043)

[第2章 系统分析 2](#_Toc13961)

[2.1系统业务需求分析 2](#_Toc13309)

[2.2系统性能需求分析 3](#_Toc30190)

[2.3 系统安全性需求分析 3](#_Toc1251)

[2.4系统运行环境需求分析 3](#_Toc3739)

[2.3.1 硬件环境最低需求 3](#_Toc28382)

[2.3.2 服务器软件环境需求 4](#_Toc12863)

[2.5系统数据流图 4](#_Toc26188)

[2.5.1 系统顶层数据流图。 4](#_Toc3977)

[2.5.2 一层数据流图 4](#_Toc17371)

[2.6系统的用例图及用例描述 6](#_Toc22072)

[2.6.1用例图 6](#_Toc4819)

[2.6.2 用例描述 7](#_Toc15580)

[第3章 系统设计 10](#_Toc11876)

[3.1系统功能分析 10](#_Toc3732)

[3.1.1 系统功能概述 10](#_Toc9876)

[3.1.2系统功能模块设计 3](#_Toc27019)

[针对上述功能，客户端系统划分为2个子系统，并包括10个主要模块。子系统包括前台用户子系统和后台管理子系统。其中前台用户子系统含有注册登录模块、开户绑卡模块、理财模块、充值提现模块和查看交易记录模块等；后台管理子系统包括登录模块、管理员上架模块、产品审核模块、管理员管理模块、产品调整模块等。 3](#_Toc21809)

[3.2系统功能模块结构图 3](#_Toc13387)

[3.3系统主要功能模块流程图 3](#_Toc12746)

[3.4系统类模型 10](#_Toc9750)

[3.5数据库设计 10](#_Toc2892)

[3.5.1数据库实体关系图（E-R图）设计 10](#_Toc15372)

[3.5.2数据库总体关系设计图 13](#_Toc21652)

[3.5.3数据库表设计 14](#_Toc26322)

[3.6动态建模 19](#_Toc18443)

[3.6.1 用户管理设计时序图 19](#_Toc6709)

[第4章 系统实现 22](#_Toc18729)

[4.1 本系统开发过程中技术特色说明 22](#_Toc30506)

[4.2欢迎引导页的设计与实现 23](#_Toc31488)

[4.3前台子系统的设计与实现 25](#_Toc19104)

[4.3.1 用户注册模块的设计与实现 25](#_Toc31813)

[4.3.2 用户登录模块的设计与实现 26](#_Toc12279)

[4.3.3 申购赎回模块的设计与实现 28](#_Toc31852)

[4.3.4 理财及筛选产品模块的设计与实现 31](#_Toc23581)

[4.3.5 充值提现模块的设计与实现 35](#_Toc15055)

[4.4 后管系统的设计与实现 44](#_Toc32410)

[4.4.1 后管系统登录效果如图所示： 44](#_Toc18434)

[4.4.2 后管系统通过上架产品效果如图所示： 45](#_Toc20511)

[4.4.3 后管系统上架产品效果如图所示： 46](#_Toc25274)

[4.4.4 后管系统产品审核结果效果如图所示： 46](#_Toc25161)

[4.4.5 后管系统产品列表效果如图所示： 47](#_Toc21742)

[4.5 安全功能的设计与实现 48](#_Toc8885)

[4.5.1加解密的设计与实现 48](#_Toc6953)

[4.5.2验签的设计与实现 50](#_Toc21414)

[4.6 本章小结 51](#_Toc12054)

[第5章 系统测试 52](#_Toc3772)

[5.1 测试的目的 52](#_Toc22486)

[5.2 测试的方法 52](#_Toc19094)

[5.2.1 黑盒测试 52](#_Toc13728)

[5.2.2 单元测试 52](#_Toc25172)

[5.3 系统测试 53](#_Toc11188)

[5.3.1黑盒测试方案和效果 53](#_Toc22096)

[5.3.3单元测试方案和效果 61](#_Toc8884)

[5.4 本章小结 61](#_Toc30608)

[结 论 62](#_Toc11987)

[参考文献 63](#_Toc31668)

[致 谢 65](#_Toc16515)

# Abstract

With the development of computer and Internet technology, traditional payment and financial management methods have also ushered in a new round of changes. Words such as "virtual bank" and "online transaction" are increasingly appearing in people's lives and transactions. The development and realization of this system aims to realize that payment and transfer can no longer be carried out offline by using network technology and other emerging technologies, so as to solve the limitation of traditional transaction and payment by place, time and security.

This paper describes in detail how to use SpringBoot framework technology to develop a banking system that can transfer money and conduct financial management, to achieve login, registration, payment, transfer, subscription, redemption, income calculation and other features. The system is developed with SpringBoot, and Angular and Ionic front-end frameworks are used to realize the separation of front and rear ends of the system, so as to optimize the development process and user experience.

This paper starts from the demand stage, fully considers the actual demand of the virtual banking system, and describes each step from analysis to implementation in detail. First, the front and back of the system complement each other, and design the front of the system "bank transaction operation interface display"; Secondly, the "virtual bank management system" is designed to manage the data involved in each module of the system and manage the system. After the system was developed, a simple black box test was conducted and Postman test tool was used to test the front and rear independent units and interfaces, and the test results all achieved the desired results. The design and development of the system conforms to the software development process and related regulations, and can meet the needs of users for an online transaction, has certain practical significance.

**Key Words:** Virtual banking system SpringBoot Postman MySQL8 IntellijIDEA

**第1章** **绪 论**

**1.1课题的来源及意义**

手机作为当下的随身必备携带的物品,已经和我们的生活密不可分了。当下金融产业在互联网的推动下快速发展,货币的实体化模式正在慢慢的被虚拟化货币流通方式取代。手机作为当下现实世界与虚拟网络世界的一个重要的桥梁枢纽,手机的流通化的便利是我们可以随时随地的了解目前身边的社会各种动态,对于当下银行的各种利率信息变化也有着极其精准的了解。手机用户们可以利用手机联网的便利条件以及优势,对于自己银行账户有个细致入微的查阅以及各种银行办理业务的操作。金融银行实体经济已向轻型网点转化,目前的金融产业的经济管理内部高层人员已经把重点放在了研发手机设备上的金融银行软件上。当然任何行业都存在这竞争与压力,但当前“大经济世界”下竞争力尤其的激烈,所以,如何运用当下自己拥有的资源以及技术人才开发出符合大众要求且让大家的生活变得更加便利的手机金融银行软件,这个问题已经成为了当下决策者经理们应该解决的问题重中之重,也是应有的题中之义。

我们这篇文章中所集中论述的基于Android操作系统的手机银行软件APP的设计方面的工作,主要将会分为大概四个大方面来进行叙述展开,分别为软件工程规格要求的需求分析阶段工作、总体设计阶段工作、详细设计阶段工作、实现以及测试阶段工作等。这个手机银行软件运行的环境系统分为服务器端系统以及手机上运行的Android环境客户端系统。这个手机银行的系统服务器端是基于J2EE平台设计研发出来的,这个服务器的研发的框架是用的springboot框架来搭建编写的,这是当下一种比较流行的开源框架所衍生出的一个框架模型,非常的贴合我们当下追求的时尚的手机功能。接着,这个时下比较流行的手机功能之间的信息传递任务是通过基于面向对象设计的Apache开源工具类设计研发的,其传输层的应用协议是严格遵循HTTPS通讯协议的。最后,在对于本系统的测试工作中,本系统的测试人员也严格按照软件工程标准里的黑盒测试策略以及白盒测试策略进行,其测试结果均在理想范围内。

**1.2国内外发展状况**

掌中的“方寸之地”已然成为各银行的必争之地，招商银行更是在2018年底宣布完成全国网点“全面无卡化改造”项目，成为国内首家实现网点全面无卡化的银行。在技术层面打破银行间壁垒，在行业内率先向他行持卡人全面开放，完成了互联网式的虚拟账户搭建，推动平台的扩展经营。2016年，“饭票”、“影票”迁移到“招银金融云”上，云端扩容让“掌上生活”的“饭票”每分钟可以处理超过3万笔的订单，月消费也突破3亿元手机银行尽管发展较快，但是争夺流量入口的效果依然有限。微信、支付宝、京东等具有较强入口能力的APP都在逐渐平台化、重型化、生活化，并将金融功能嵌入其托起的业务场景中。而银行系，微众银行由于自身“基因”特点，平台化方向较为明显，是自身业务的主要承载渠道；招行提出了用手机APP替代银行卡的目标，转型意味也比较强；其他银行的手机银行尽管功能丰富，但是平台化的战略特点和方向并不十分清晰。

**1.3系统可行性研究**

**1.3.1经济可行性**

首先，在本系统的开发过程中所使用的开发环境和开发工具如IntellijIDEA、MySQL等均是开源软件，这在很大程度上节省了经济成本。其次，本系统对服务器的硬件要求不是很高，因此，开发完成后的系统能够在一般的服务器上运行。由于Java语言有跨平台的特性，因此运行系统的服务器可以选择Linux操作系统，这将大大节省系统的授权费用。

综上所述，本系统的开发与实现在经济上切实可行。

**1.3.2技术可行性**

首先，本系统使用集成开发工具intellij IDEA 进行开发。IDEA是JAVA语言集成开发环境，此工具对社区用户免费，是公认最好的Java集成开发环境。Java语言是现如今计算机编程语言占有率第一位的开发语言，很多的互联网公司和电商企业都使用Java作为后台开发语言，有很好的技术沉淀和技术的解决方法，Java语言诞生到现在，从刚开始的迎合C语言，到现在基本完全脱离C语言，自成一套一次开发到处运行的远景已然实现，现如今的多种开发框架如SSM、SpringBoot使得Java语言能够承担各种开发测需求，不管是桌面级应用还是企业及应用都使用简单、性能可靠，可以让用户和开发者满意，使用Java和SpringBoot框架进行开发，技术上困难有很多现行的解决方案可以借鉴，能够很好的锻炼学生的解决问题的能力。

数据库上，由于用不到分布式技术以及数据库的需求，考虑到使用的轻便和可靠的特点，所以采用MySQL8作为系统数据库。MySQL8是现在使用最多，最轻量的免费关系型数据库之一。本系统对于使用关系型数据库遇到的问题，相关网站有很多的解决方法，大大降低了重复造轮子的可能性，所以使用MySQL8可以很好的胜任系统对数据库的开发使用的要求。

综上所述，该手机银行系统的开发与实现在技术上完全可行。

**1.3.3社会可行性(法律可行性)**

“手机银行客户端系统”的开发，会给当今时间宝贵，线下实现手续繁琐的痛点用户群体带来生活，工作和投资的便利，受到普遍的社会接受。网站在开发过程中未侵害任何企业和个人的版权或所有权，且本系统在开发过程中使用的开源软件符合开源许可证要求。

综上所述，本系统的开发与实现在法律上切实可行。

**1.4本文的组织结构图**

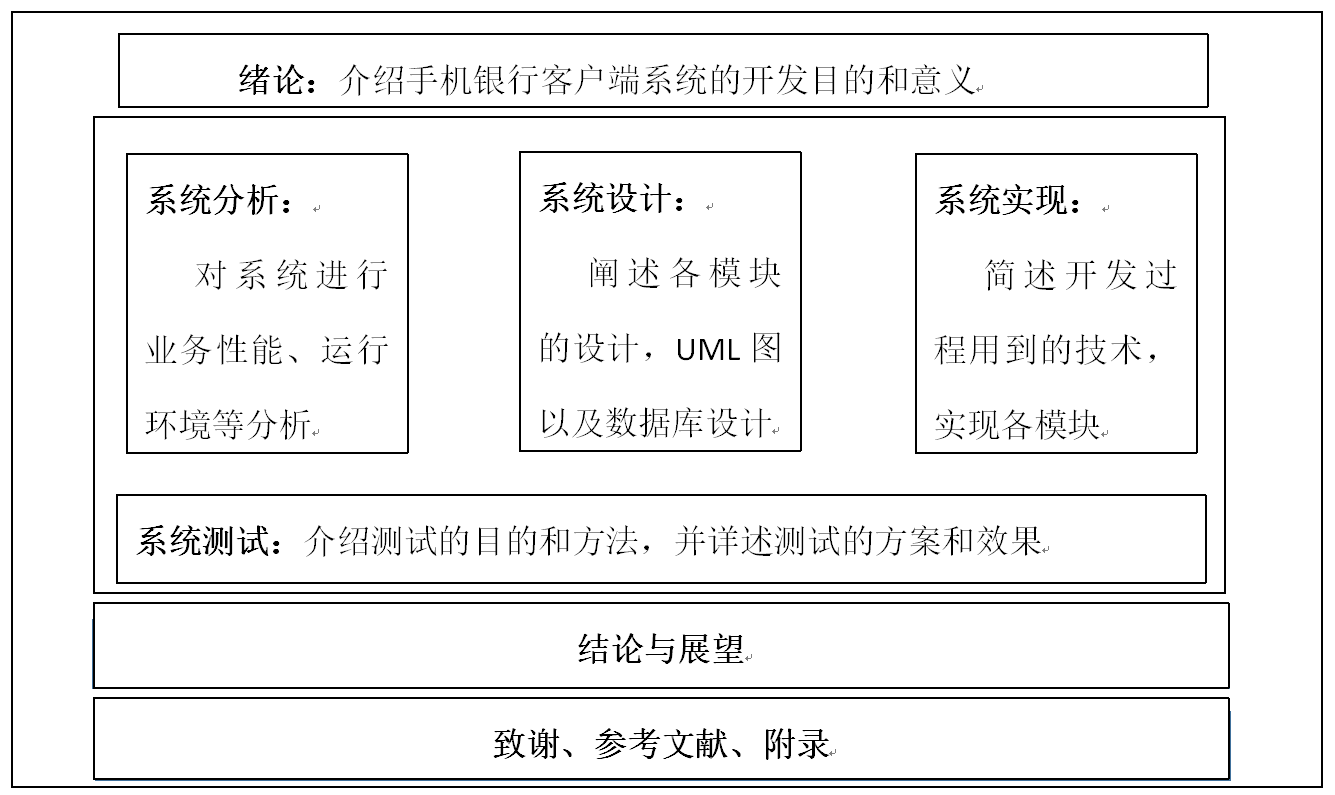


图1-1 组织结构图

**1.5本章小结**

本章节中，第一步阐述了“手机银行客户端系统”这个课题的来源及意义，第二步从系统的经济可行性、技术可行性、法律可行性三个大的方面对该系统进行可行性研究。最后，在进行本章任务时，查阅了相关的手机银行以及Java框架等技术文献，对该论文系统主题的国内外发展现状进行研究。本章作为基础的绪论部分，也为后续系统的设计和实现打下了理论基础。

**第2章 系统分析**

## 2.1系统业务需求分析

手机银行客户端系统根据业务属性进行垂直拆分，主要包含UI服务、组合服务、基础服务。

UI服务：主要面向前端用户层，提供服务包括但不限于网上银行整体页面设计优化，首页服务、功能服务、交易信息服务、用户个人中心页服务。

组合服务：针对UI服务提供不同场景的组合服务，包括但不限于提现服务、充值服务、理财购买服务、理财变现服务。

基础服务：针对UI服务和组合服务提供基础服务，基础服务遵循抽象、稳定、高可复用原则，按照功能划分为：

登录基础服务：用户登录（登录、验证）

注册基础服务：用户注册

开户基础服务：用户开户（信息验证）

绑卡基础服务：用户绑卡(信息验证)

交易记录服务：充值、提现、用户查看交易记录（交易记录、交易详情）

产品基础服务：申购，赎回

**公安部子系统：**接收虚拟银行系统的请求，根据虚拟银行系统发送的用户姓名、身份证号，完成连网核查及身份认证，返回认证结果给虚拟银行系统。所有接口均需支持并发访问。

**银行互联网核心子系统：**接收虚拟银行系统的绑卡、充值、提现、电子账户余额查询、申购、赎回请求，根据虚拟银行系统发送的信息，完成逻辑判断、动账等操作，返回请求结果给虚拟银行系统。所有接口均需支持并发访问。

**2.2系统性能需求分析**

根据上述的业务需求，并且结合实际情况和可能出现的性能需求，现对所需实现的系统在性能上提出了以下几个方面的要求：

1 精度

精度指的是系统的显示精度，如系统中出现的数字（金额、时间等）要完整的显示正确的位数或单位。

2 时间特性要求

（1）系统正常响应时间不能大于200ms；

（2）更新处理时间应小于500ms；

（3）数据的转换和传送时间应小于1s，对数据库需进行适当优化以提高效率。

**2.3 系统安全性需求分析**

系统的安全性高低对一个系统来说至关重要，首先招聘网站需要对用户的信息进行保密，还要对整个网站的安全有所保障。为了使系统拥有完善可靠的安全体系，本系统做出了如下几方面设计。

（1）将系统划分为不同功能的模块，并且设置不同的权限级别，防止一些非法用户的非法操作对网站进行破坏。

（2）对所有的登录用户进行身份验证，防止不法分子入侵本网站。

**2.4系统运行环境需求分析**

系统运行时对于环境的要求有：

**2.3.1 硬件环境最低需求**

（1） 服务器：

1. 处理器（CPU）：1.5GHz 或更快
2. 内存（RAM）：2GB以上
3. 硬盘空间：20GB以上
4. 接入互联网

（2） 客户端：

1. 处理器（CPU）：1.5GHz 或更快
2. 内存（RAM）：2GB以上
3. 分辨率在800\*600以上的显示器
4. 2M以上宽带
5. 键盘及鼠标等输入设备

**2.3.2 服务器软件环境需求**

（1）操作系统：Windows 10

（2）JDK版本：JDK 1.8

（3）服务器版本：Tomcat 9

（4）数据库版本：Mysql 8

**2.5系统数据流图**

数据流图有五种基本图形符号：

**->**（箭头），表示数据流：数据流是数据在系统内传播的路径，因此由一组成分固定的数据组成；

〇（圆或椭圆），表示加工：对数据流进行某些操作或变换。每个加工也要有名字，通常是动词，简明地描述完成什么加工；

= （双杠），表示数据存储：指暂时保存的数据，它可以是数据库文件或任何形式的数据组织；

□（矩形或方框），表示数据的源点或终点：是本软件系统外部环境中的实体(包括人员、组织或其他软件系统)，统称外部实体；

，表示对象：是系统中的实体。

**2.5.1 系统顶层数据流图。**

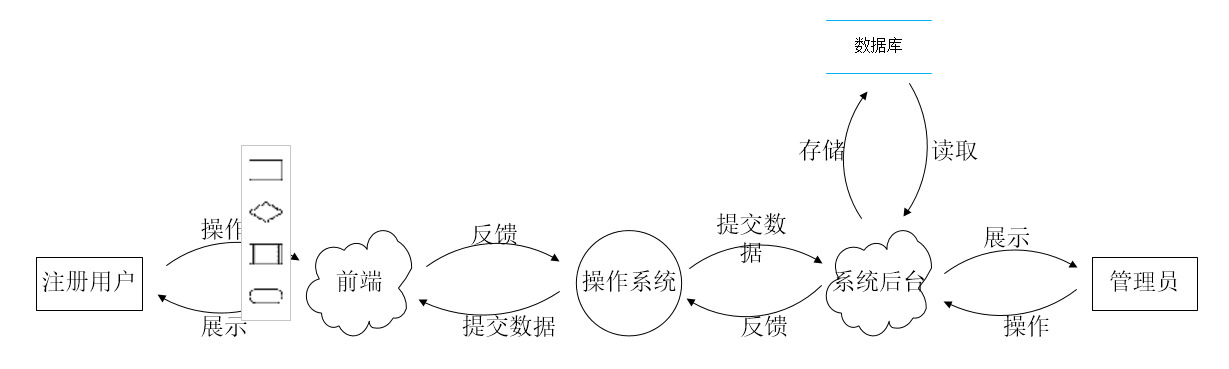


图2-1 系统顶层数据流图

**2.5.2 一层数据流图**

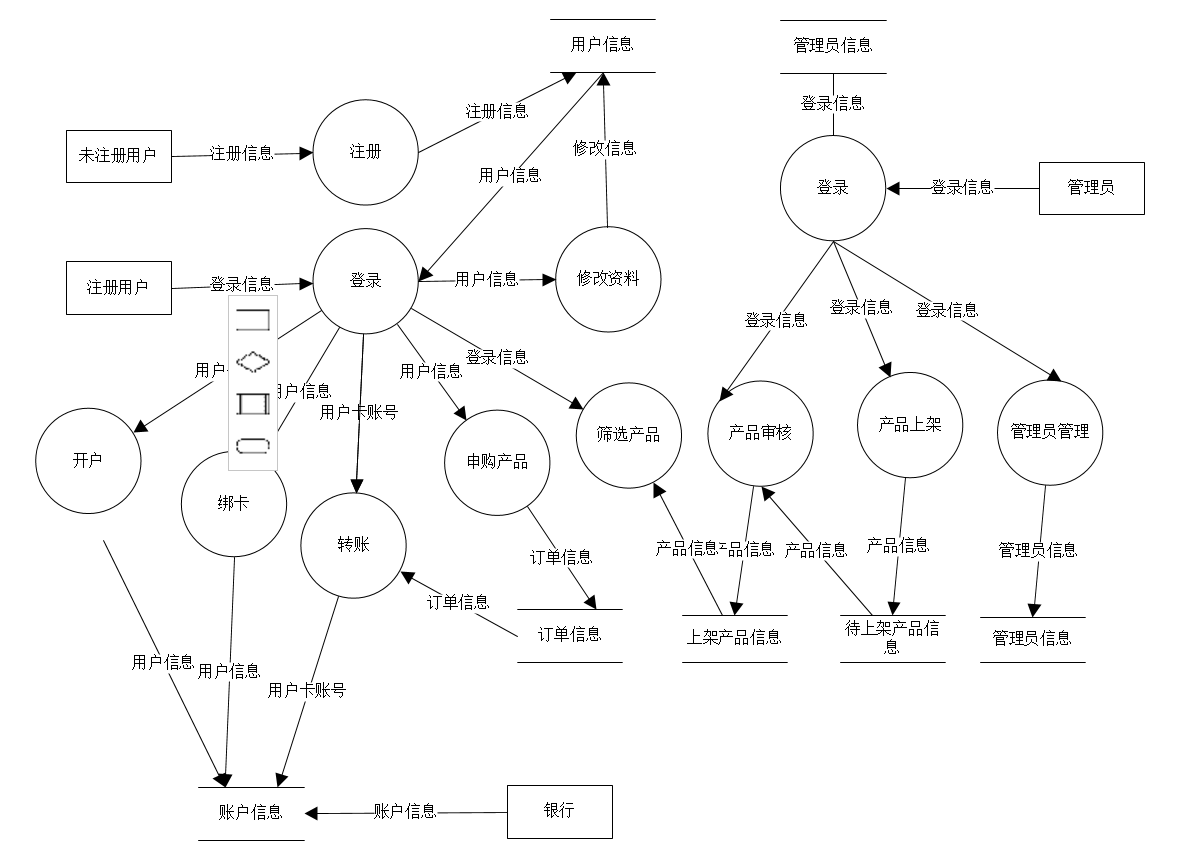


图2-2 系统一层数据流图

**2.5系统数据字典**

表2-1 数据项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 别名 | 数据类型 | 数值含义 |
| idCard | 用户身份证号 | 身份ID | varchar | 定长字符串 |
| cardNum | 系统中用户银行卡号 | 用户银行卡号 | varchar | 用户身份证对应的卡号 |
| financeProduct\_id | 产品的编号 | 创建产品ID | varchar | 唯一标识产品 |
| Status | 用户是否已注册 | 用户状态 | int | 0未注册 1注册 |
| role-type | 管理员权限 | 管理员权限 | int | 0产品管理员1审核员 2超管 |
| purchaseOrder\_id | 申购产品的编号 | 申购编号 | varchar | 购买记录的唯一标识 |
| order\_status | 订单状态 | 订单状态 | int | 0交易中，1交易完成 |
| complete\_time | 特指订单完成时间 | 完成时间 | datetime | 年月日T时分秒 |
| Tran\_id | 转账记录的编号 | 转账ID | varchar | 唯一标识转账记录 |

表2-2 数据结构

|  |  |  |
| --- | --- | --- |
| 数据结构名称 | 含义说明 | 组成 |
| 登陆用户 | 通过浏览器访问系统的已注册的人 | 用户名、密码、注册时间等可用用户信息 |
| 管理员 | 可以使用系统后台管理系统的用户 | 用户名、密码、权限 |
| 上架产品 | 用户上架的用于申购的理财产品 | 商品名、商品详细、价格、图片等 |
| 银行 | 用户管理用户的银行卡信息和动账交易 | 卡名称、银行类别等 |
| 订单 | 用户在线交易时生成的单据 | 编号、价格、用户、商品等 |
| 待上架产品 | 管理员上架产品到订单中待审核员通过后该产品才可以用于购买销售 | 产品的利率和图片，预期收益，上架时间和上架公司等 |

**2.6系统的用例图及用例描述**

**2.6.1用例图**



图2-3 手机银行客户端前台系统用例图

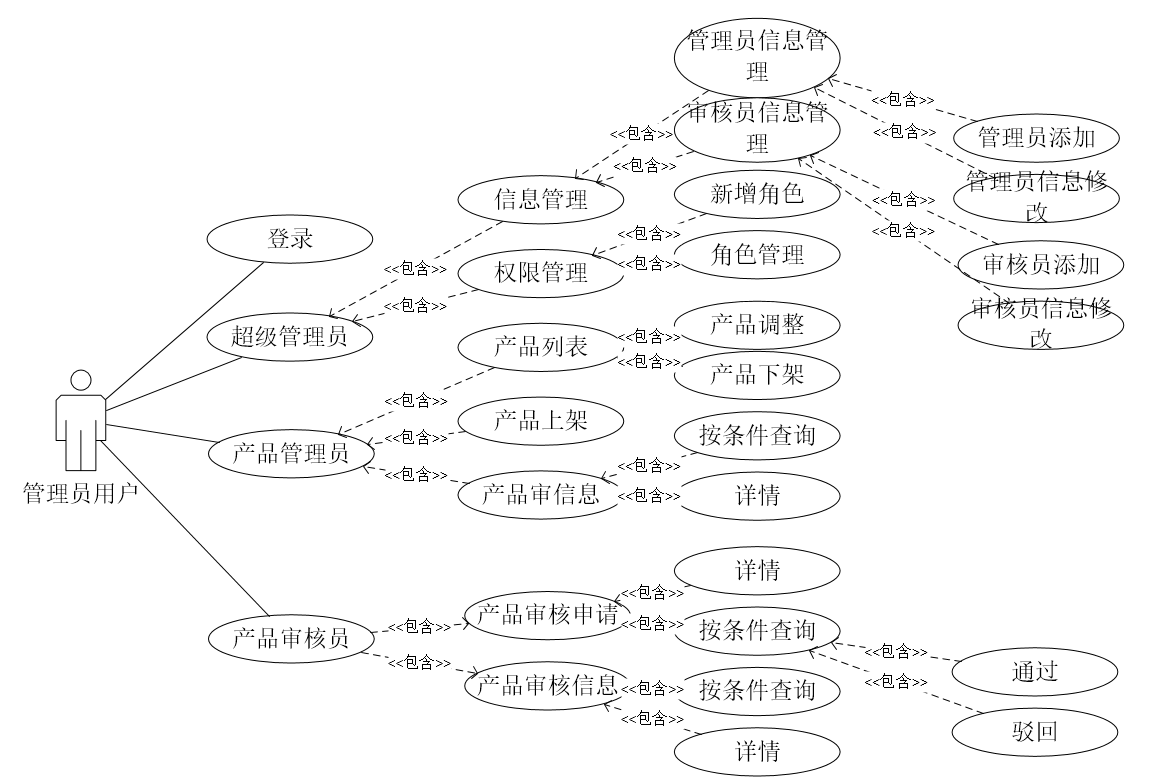


图2-4 手机银行客户端后管系统用例图

**2.6.2 用例描述**

1. 用户注册

当用户访问本网站时可以注册，注册的用例描述如表2-3所示。

表2-3 注册用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户注册 | |
| 用例ID | UC1 | |
| 参与者 | 普通用户 | |
| 描述 | 普通用户注册系统过程 | |
| 优先级 | 高 | |
| 前置条件 | 用户需在未注册状态 | |
| 后置条件 | 将用户注册信息保存到数据库中 | |
| 主事件流 | | |
| 用户 | | 系统 |
| 1．进入或转入用户注册画面 | |  |
| 2．用户输入电话号，密码，验证码 | |  |
|  | | 3．系统检查电话和密码格式是否合法 |
|  | | 4．系统检查电话号是否存在 |
|  | | 系统检查输入的密码以及验证码是否相匹配 |
|  | | 提示注册成功，转向用户界面 |
| 异常流 | | |
| 异常流 | | 系统中已经存在此电话号 |
| 系统 | | 给出提示信息“此电话号存在” |
| 异常流 | | 密码格式不合法 |
| 系统 | | 给出提示信息“请输入有数字和字母组成的八位以上的密码” |
| 异常流 | | 输入的密码以及验证码不相同 |
| 系统 | | 给出提示信息“输入的密码、验证码不相同” |

2）用户登录

注册完用户之后可以通过邮箱，密码登录，登录之后可以享受用户的基本权限，登录的用例描述如表2-4所示。

表2-4 登录用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户登录 | |
| 用例ID | UC2 | |
| 参与者 | 用户 | |
| 描述 | 用户登录系统过程 | |
| 优先级 | 高 | |
| 前置条件 | 用户需在未登录状态 | |
| 后置条件 | 将用户信息放入会话中 | |
| 主事件流 | | |
| 用户 | | 系统 |
| 1．进入或转入用户登录画面 | |  |
| 2．用户输入电话号和密码 | |  |
|  | | 3．系统检查电话号，密码是否正确 |
|  | | 4．读取用户的基本信息放入会话中 |
|  | | 5．转到主画面 |
| 异常流 | | |
| 异常流 | | 该用户未注册或是电话号码格式不合法 |
| 系统 | | 给出提示信息“请在输入正确位数的号码” |
| 异常流 | | 用户名或密码错误 |
| 系统 | | 给出提示信息“用户名或密码错误” |

3）管理员信息管理

管理员信息管理包括管理员查询，管理员添加。管理员查询可以通过用户名、用户编号进行不同程度的模糊查询，管理员信息管理用例描述如表2-5所示。

表2-5 管理员信息管理用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 管理员信息 | |
| 用例ID | UC3 | |
| 参与者 | 管理员 | |
| 描述 | 超管进行管理员管理 | |
| 优先级 | 高 | |
| 前置条件 | 超管主动进入管理员界面 | |
| 后置条件 | 成功管理用户 | |
| 主事件流 | | |
| 管理员 | | 系统 |
| 1．单击信息管理链接 | |  |
|  | | 2．显示信息管理主画面，下拉选框中包括用户查询，用户添加功能 |
| 可选事件流 | | |
| 用户查询 | | |
| 管理员 | | 系统 |
| 1．输入id，点击查询 | | 2．显示符合条件的用户及用户信息 |
| 3．输入用户名，点击查询 | | 4．显示符合条件的用户及用户信息 |
| 用户添加 | | |
| 11.输入添加用户的基本信息，点击提交 | | 12.提示添加成功 |

4）产品管理

产品管理包括产品查询，产品添加。产品查询可以关键字、ID及时间进行不同程度的查询，产品管理用例描述如表2-6所示。

表2-6 产品管理用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 产品管理 | |
| 用例ID | UC3 | |
| 参与者 | 管理员 | |
| 描述 | 管理员进行产品管理 | |
| 优先级 | 高 | |
| 前置条件 | 管理员主动进入产品管理界面 | |
| 后置条件 | 成功管理产品 | |
| 主事件流 | | |
| 管理员 | | 系统 |
| 1．单击产品管理链接 | |  |
|  | | 2．显示产品管理主画面，下拉选框中包括产品查询，产品添加功能 |
| 可选事件流 | | |
| 查询产品 | | |
| 管理员 | | 系统 |
| 1．输入关键字，点击查询 | | 2．显示符合条件的产品及产品详情 |
| 3．输入ID，点击查询 | | 4．显示符合条件的产品及产品详情 |
| 发布产品 | | |
| 7.输入标题及所添加内容，点击上架。 | | 8.提示上架产品成功 |

**2.7本章小结**

本章的主要内容是系统的需求分析，以上内容对系统的基本功能需求进行了分析，介绍了系统完整体验所需的硬件及软件要求。本章展示了系统的基本数据流图和一层数据流图，编写了数据字典，介绍了系统在开发和使用过程中涉及的部分易歧义的数据项。在本章的最后，展示了系统的用例图，介绍系统的用例关系，详细进行了用例描述，为后续步骤做好准备。

**第3章 系统设计**

## 3.1系统功能分析

**3.1.1 系统功能概述**

如下表3-1实现的主要功能有：

表3-1主要的功能模块

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 功能模块 | 主要功能点 | 功能描述 |
| 1 | 注册 | 手机号注册 | 验证手机号是否正确；  验证手机号是否已注册 |
| 2 | 设置密码 | 密码8-16位，需字母+数字+字符至少两种，错误提示密码错误 |
| 3 | 手机验证码 | 必填，不正确时提示验证码不正确，并保留已输入账号、密码 |
| 4 | 登录 | 手机号登录 | 验证手机号是否正确 |
| 5 | 输入密码 | 必填 |
| 6 | 手机验证码 | 必填，不正确时提示验证码不正确，并保留已输入账号密码 |
| 7 | 开户 | 输入姓名 | 必填 |
| 8 | 输入身份证号 | 需验证身份证号长度是否正确 |
| 9 | 绑卡成功跳转绑卡 |  |
| 10 |
| 11 | 绑卡 | 输入姓名 | 均必填  判断四个要素是否匹配 |
| 12 | 输入身份证号 |
| 13 | 输入银行卡号 |
| 14 | 输入银行预留手机号 |
| 15 | 手机验证码 | 必填，不正确时提示验证码不正确，并保留已输入信息 |
| 16 | 充值 | 选择扣款账号 | 用户通过选择框选择扣款账号 |
| 17 | 输入充值金额 | 必填；  大于0 |
| 18 | 手机验证码 | 必填，不正确时提示验证码不正确，并保留已填信息 |
| 19 | 余额查询 | 显示信息 | 开户和绑卡均完成后即可查询余额；  显示电子卡账户账号；  显示电子卡账户余额；  显示所有绑定卡账号；  显示所有绑定卡可用余额 |
| 20 | 提现 | 选择提现卡 | 开户和绑卡均完成，且二类户账户（电子卡账户）可用余额大于0即可提现 |
| 21 | 输入收款卡 | 收款卡号默认为当前账户主卡，可选择收款卡 |
| 22 | 手机验证码 | 必填，不正确时提示验证码不正确，并保留已输入信息 |
| 23 | 转账 | 显示可用余额 | 开户和绑卡均完成后即可转账；  脱敏显示可用余额（当二类户账户可用余额大于0时默认回显二类户可用余额，当二类户账户可用余额为0时默认回显主卡可用余额及账号）； |
| 24 | 输入收款人卡号 | 必填 |
| 25 | 输入收款人开户行 | 根据收款人账号自动匹配开户行，匹配错误用户可手动输入 |
| 26 | 选择转出卡号 | 一类户转一类户 |
| 27 | 输入转出金额 | 必填，用户无法输入大于当日限额的数字 |
| 28 | 手机验证码 | 必填，不正确时提示验证码不正确 |
| 29 | 交易记录 | 查看交易记录 | 开户和绑卡均完成后即可查询交易记录；  交易记录根据交易时间倒序排列； |
| 30 | 产品申购 | 输入买入金额 | 必填，金额不可小于最低购买金额，不可大于最大购买金额 |
| 31 | 手机验证码 | 必填，不正确时提示验证码不正确，并保留已输入信息 |
| 32 | 产品赎回 | 已购产品到期自动赎回 |  |
| 33 | 银行卡 | 显示已绑定卡 | 显示开户行，卡号显示前6位后4位 |
| 34 | 后管系统 | 产品上架、产品调整 | 必填产品相关信息，通过后可在app里查看，下架后查看已经停售 |

**3.1.2系统功能模块设计**

针对上述功能，客户端系统划分为2个子系统，并包括10个主要模块。子系统包括前台用户子系统和后台管理子系统。其中前台用户子系统含有注册登录模块、开户绑卡模块、理财模块、充值提现模块和查看交易记录模块等；后台管理子系统包括登录模块、管理员上架模块、产品审核模块、管理员管理模块、产品调整模块等。

其中，前台主要模块为理财模块和充值提现模块，理财模块包含理财产品的检索和展示，详情、在线买入赎回等功能，充值提现模块是供用户进行商品的购买之前必须进行的转账和提现等操作，方便线上进行交易的基础。

## 3.2系统功能模块结构图

基于以上的功能模块设计分析，本系统包括的功能模块结构图如下：

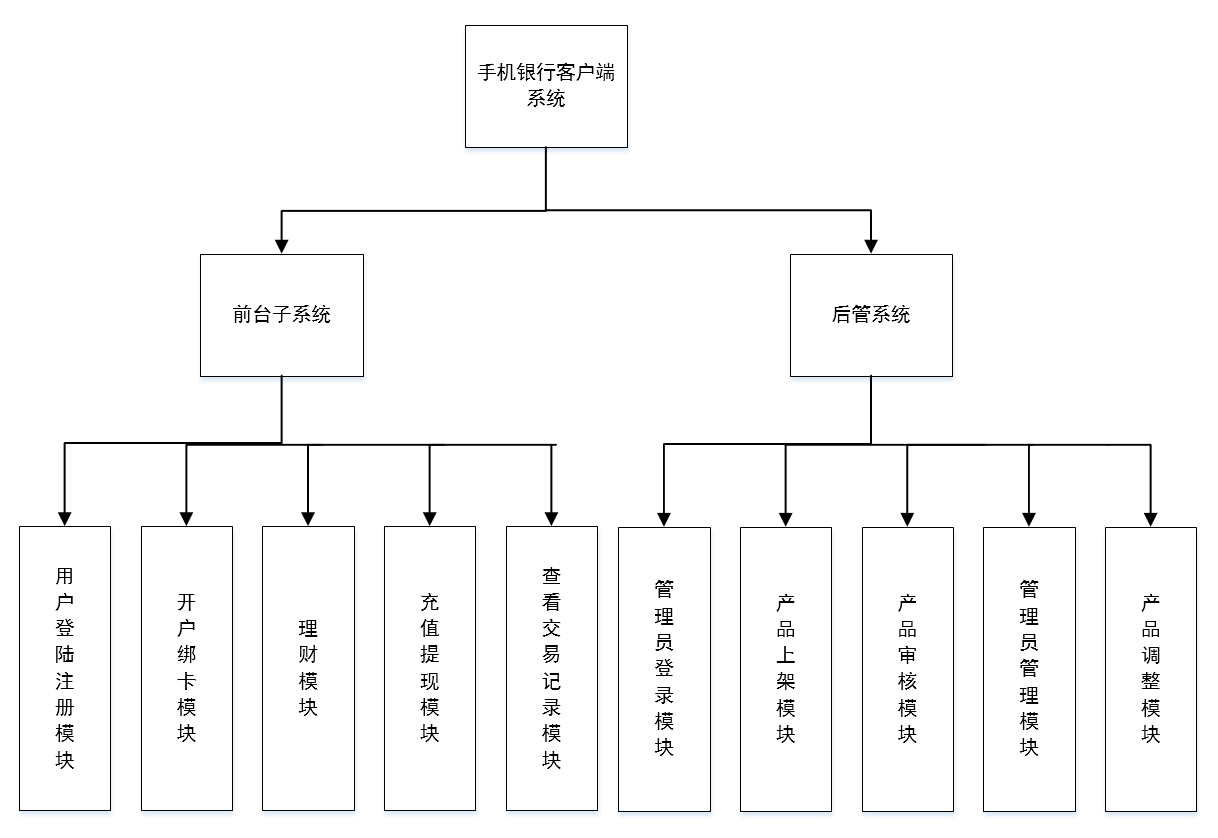


图3-1 手机银行客户端系统功能模块结构图

## 3.3系统主要功能模块流程图

3.3.1身份认证（开户）



图3-2 开户模块流程图

持卡人身份认证（绑卡）



图3-3 绑卡模块流程图

充值



图3-4 充值模块流程图

提现：



图3-2 提现模块流程图

电子账户余额查询



图3-5 余额模块流程图

产品申购



图3-6 理财模块流程图

产品赎回



图3-7 理财模块流程图

## 3.4系统类模型

## 3.5数据库设计

**3.5.1数据库实体关系图（E-R图）设计**

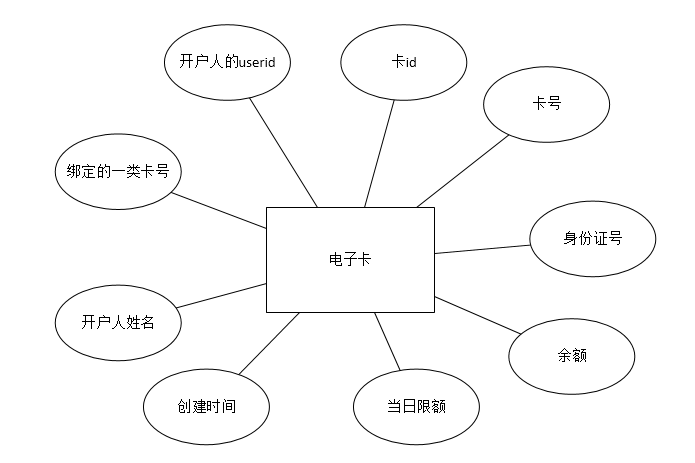


图3-9 电子卡的ER图

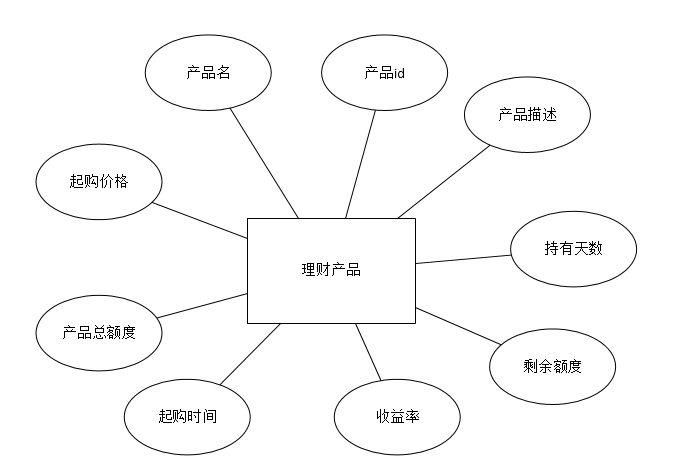


图3-10 理财产品的ER图

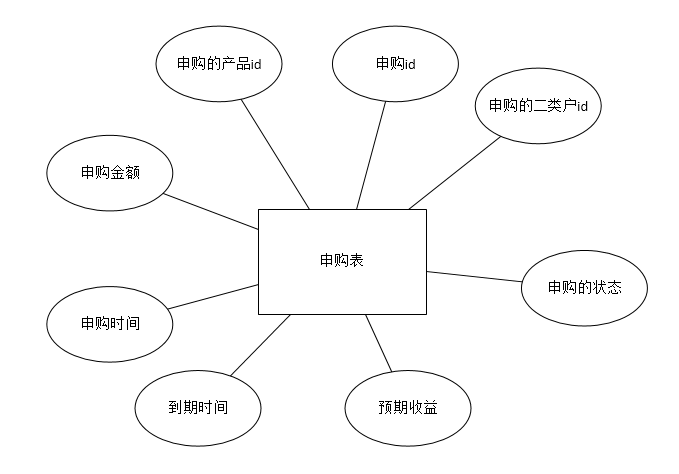


图3-11 申购表的ER图

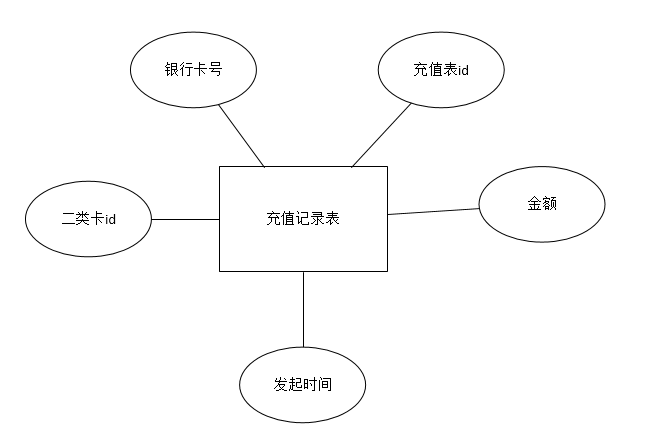


图3-12 充值记录的ER图

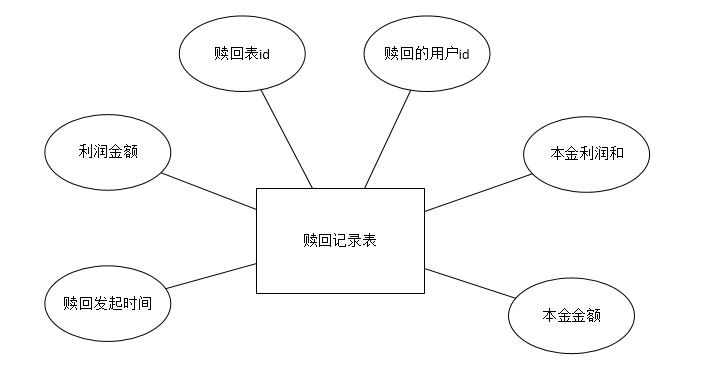


图3-13 赎回记录的ER图

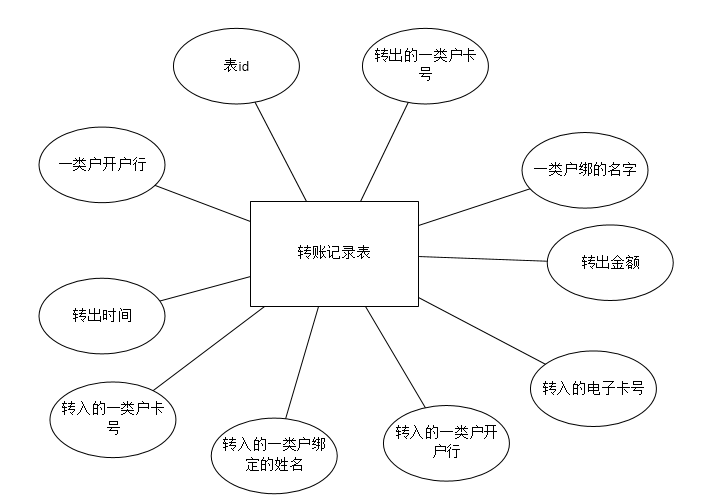


图3-14 转账记录表的ER图

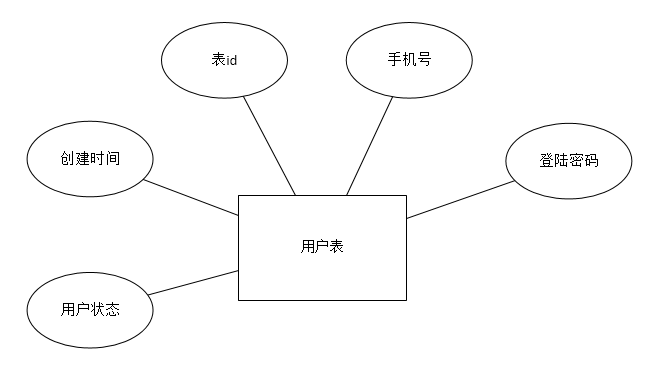


图3-14 用户表的ER图

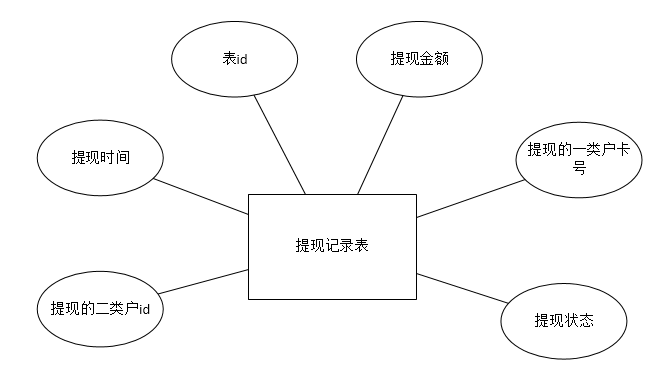


图3-15 提现记录表的ER图



图3-16 产品申请上架的ER图

**3.5.2数据库总体关系设计图**



图3-17 系统总体的ER图

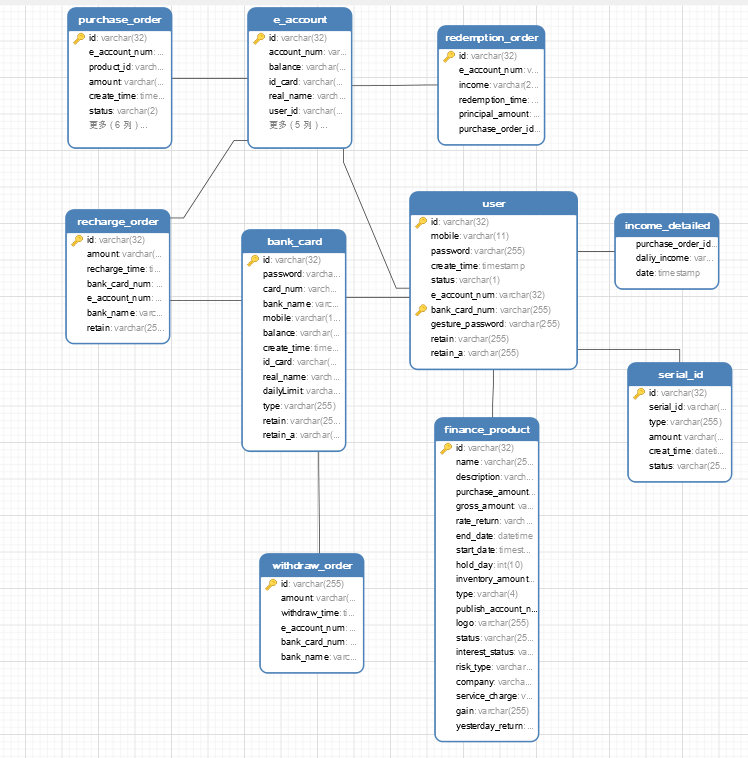


图3-18 总体关系设计图

**3.5.3数据库表设计**

表3-1 User表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | String | 9 | not |  | yes |  | 用户编号 |
| mobile | String | 50 | not |  |  |  | 用户电话 |
| createTime | Date | 50 | not |  |  |  | 创建时间 |
| password | String | 50 | not |  |  |  | 用户密码 |
| status | String | 50 |  |  |  |  | 用户状态 |

表3-2 bankCard表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| realName | string |  | not |  |  |  | 开户人姓名 |
| bancard\_id | string | 50 | not |  | yes |  | 表的编号 |
| mobile | string | 50 | not |  | yes |  | 银行预留手机号码 |
| password | string | 50 | not |  | yes |  | 密码 |
| cardNum | string | 50 | not |  |  |  | 银行卡号 |
| bankName | string | 50 | not |  |  |  | 银行名称 |
| balance | varchar | 50 | not |  |  |  | 银行卡余额 |
| createTime | datetime | 1 | not | 0 |  |  | 银行卡开户时间 |
| idCard | string |  | not |  |  |  | 身份证号18位 |

表3-3 eAccount表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| accountNum\_id | String | 9 | not |  | yes |  | 二类卡编号 |
| accountNum | String | 50 | not |  |  |  | 该二类户卡号 |
| balance | String | 255 |  |  |  |  | 二类户余额 |
| idCard | String | 9 | not |  |  |  | 身份证号18位 |
| realName | String | 9 | not |  |  |  | 开户人的真实姓名 |
| userId | String |  | not |  |  |  | 开户人的id |
| bankCardNum | String | 255 |  |  |  |  | 绑的一类户的银行卡号 |
| createTime | Date | 50 | not |  |  |  | 该二类户创建时间 |
| dailyLimit | String | 50 | not |  |  |  | 单日限额 |

表3-4 financeProduct表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| financeProduct\_id | string | 11 | not |  | yes |  | 产品编号 |
| name | string | 9 | not |  |  |  | 产品名 |
| description | string | 9 | not |  |  |  | 产品描述 |
| purchaseAmount | string | 5 | not | 1 |  |  | 起购价格 |
| grossAmount | string | 9 | not |  |  |  | 产品总额度，固定值 |
| rateReturn | BigDecimal | 11 |  |  |  |  | 收益率 |
| startDate | Date | 11 |  |  |  |  | 开始购买时间 |
| holdDay | Integer | 11 |  |  |  |  | 持有天数，固定值 |
| inventoryAmount | string | 11 |  |  |  |  | 剩余总额度 |

表3-5 purchaseOrder 表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| purchaseOrder\_id | String | 9 | not |  | yes |  | 申购订单编号 |
| eAccountId | String | 9 | not |  |  |  | 申购的二类户的id |
| productId | String | 9 | not |  |  |  | 申购的产品id |
| amount | String | 9 |  |  |  |  | 申购金额 |
| createTime | Date | 9 | not |  |  |  | 申购发起时间 |
| status | String | 1 | not |  |  |  | 申购状态 |
| income | String | 9 |  |  |  |  | 申购的预期收益-利润 |
| endTime | Date | 9 |  |  |  |  | 到期时间 |

表3-6 rechargeOrder表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | string | 9 | not |  | yes |  | 充值记录表编号 |
| amount | string | 9 | not |  |  |  | 充值金额 |
| string | string | 9 | not |  |  |  | 充值发起时间 |
| bankCard | string | 5 | not |  |  |  | 银行卡号 |
| eAccountId | string | 9 |  |  |  |  | 二类户账号的id |

表3-7 redemptionRecord表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | int | 9 | not |  | yes |  | 赎回记录编号 |
| purchaseId | int | 9 | not |  |  | yes | 赎回的用户的id |
| income | text |  | not |  |  |  | 利润金额 |
| redemptionTime | varchar | 50 | not |  |  |  | 赎回发起的时间 |
| principalAmount | datetime |  | not |  |  |  | 本金金额 |
| totalAmount | varchar | 50 | not |  |  |  | 本金利润总和 |

表3-8 transferRecord表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | String | 9 | not |  | yes |  | 转账记录编号 |
| payAccount | String | 9 | not |  |  | yes | 转出的一类户的卡号 |
| payName | String |  | not |  |  |  | 转出的人的一类户绑的名字 |
| payBank | String | 50 | not |  |  |  | 转出的一类户的开户行 |
| payTime | date | 50 | not |  |  |  | 转出时间 |
| amount | String |  | not |  |  |  | 转账金额 |
| payeeName | String | 1 | not |  |  |  | 转入方的一类户绑的名字 |
| payeeBank | String | 50 |  |  |  |  | 转入方的一类户的开户行 |
| payeeAccount | String |  |  |  |  |  | 转入方的一类户的卡号 |
| eAccountId | String |  |  |  |  |  |  |

表3-9 WithdrawOrder表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | String | 9 | not |  | yes |  | 提现记录编号 |
| amount | String | 9 | not |  |  |  | 提现金额 |
| withdrawTime | date |  | not |  |  |  | 提现时间 |
| eAccountId | String | 50 | not |  |  |  | 提现的二类户的id |
| backCard | String | 9 | not |  |  | yes | 提现的一类户的卡号 |
| status | String |  | not |  |  |  | 提现的状态 |

表3-10 internetcore\_serialId表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | String | 9 | not |  | yes |  | 流水号id |
| serialId | String | 9 | not |  |  | yes | 流水号 |
| type | String |  | not |  |  |  | 利润金额 |
| amount | String | 50 | not |  |  |  | 交易的金额 |
| creatTime | date |  | not |  |  |  | 交易的时间 |
| status | String | 50 | not |  |  |  | 交易状态 |

表3-11 police\_ serialId表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | String | 9 | not |  | yes |  | 流水号 |
| serialId | String | 9 | not |  |  | yes | 赎回的用户的id |
| creatTime | date |  | not |  |  |  | 本金金额 |
| status | String | 50 | not |  |  |  | 本金利润总和 |

表3-11 police\_ serialId表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| id | String | 9 | not |  | yes |  | 开户流水表编号 |
| serialId | String | 9 | not |  |  | yes | 赎回的用户的id |
| creatTime | date |  | not |  |  |  | 本金金额 |
| status | String | 50 | not |  |  |  | 本金利润总和 |

## 3.6动态建模

在动态建模机制中，以消息来完成对象之间的交互，用状态图、时序图和活动图来描述系统的行为。

时序图是一种交互图用于描述执行系统功能的各个角色之间相互传递消息的顺序关系，显示跨越多个对象的系统控制流程，强调的是时间和消息的次序，用来说明系统的动态情况[19]。

**3.6.1 用户管理设计时序图**

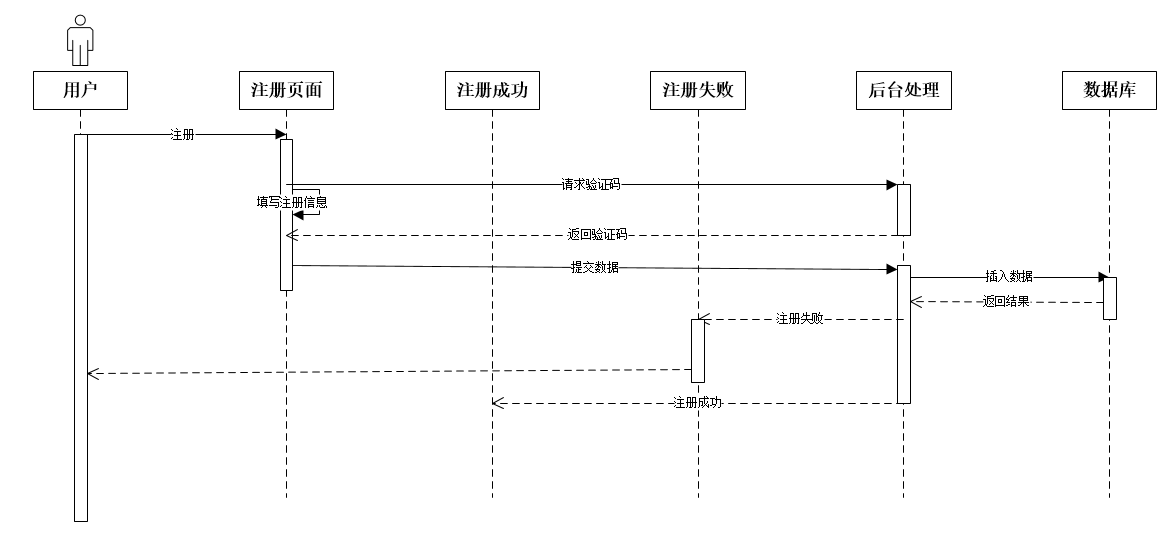


图3-21 注册账号时序图

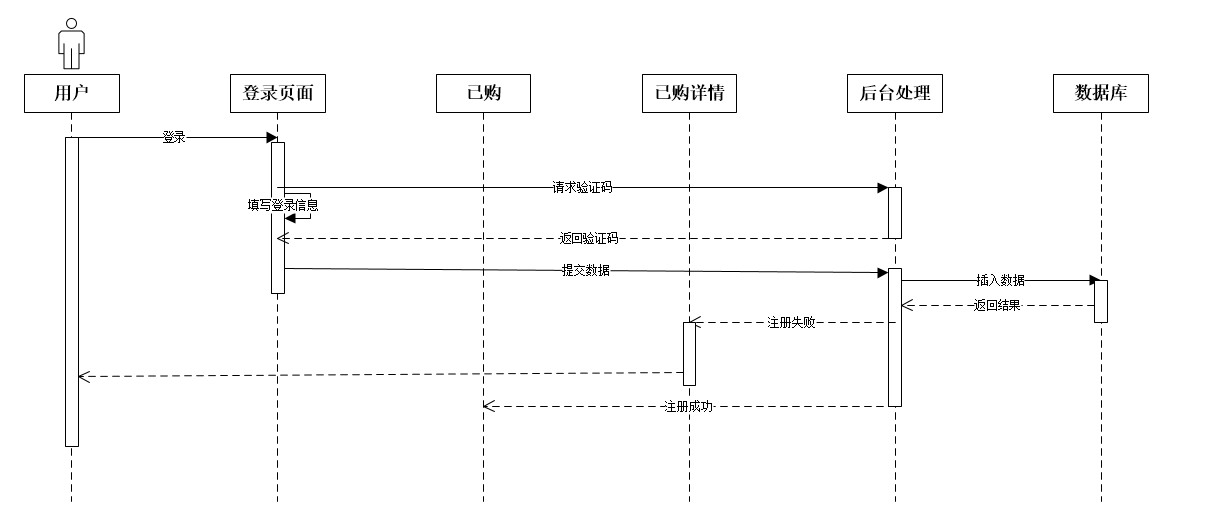


图3-2 已购时序图

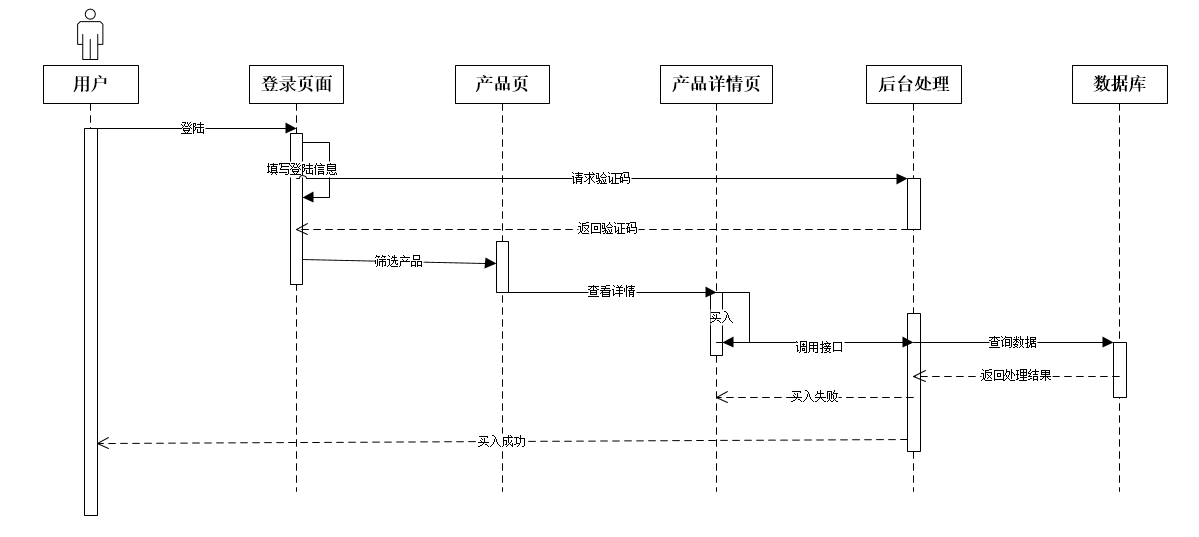


图3-23 申购时序图

**3.7 本章小结**

本章对手机银行客户端系统进行设计。首先分析了系统功能需求，根据需求将系统划分成三个模块，每个模块都包括自己对应的功能。根据划分好的模块绘制了系统功能模块图和每个模块的流程图，接下来绘制来类图。之后根据分步E-R图到总体E-R图设计了数据库以及数据库对应的表的信息。最后本章根据系统需求进行了系统的动态建模，包括了时序图和活动图。

**第4章 系统实现**

**4.1 本系统开发过程中技术特色说明**

本系统主要使用SpringBoot技术开发，所涉及的技术包括Angular、Ionic、Socket、Xml、 Spring、多线程、跨域访问等。

Angular+Ionic是实现手机APP的常用前端框架，专注业务的开发，能够有效的提高开发的效率。ionic是一个基于angular的移动端的ui组件库,基于angular，而且集成了cordov，有强大的cli工具丰富的ui组件的优势。支持开发跨平台（目前支持：Android、iOS，）的原生App应用。

SpringBoot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Boot致力于在蓬勃发展的快速应用开发领域（rapid application development）成为领导者。

SpringBoot的特点：

1. 创建独立的Spring应用程序

2. 嵌入的Tomcat，无需部署WAR文件

3. 简化Maven配置

4. 自动配置Spring

5. 提供生产就绪型功能，如指标，健康检查和外部配置

6. 绝对没有代码生成和对XML没有要求配置

SpringBoot的优点：

SpringBoot可以支持你快速的开发出restful风格的微服务架构

自动化确实方便，做微服务再合适不过了，单一jar包部署和管理都非常方便。只要系统架构设计合理，大型项目也能用，加上nginx负载均衡，轻松实现横向扩展

SpringBoot要解决的问题如下，精简配置是一方面, 另外一方面是如何方便的让spring生态圈和其他工具链整合(比如redis, email, elasticsearch)从动态网页技术为普及之前就作为网页开发的基础技术。

Socket两个跨计算机进程，需要通过网络对接起来通讯，通常也称作"套接字",用于描述IP地址和端口 , 是一个通信链的句柄。应用程序通常通过“套接字”先服务器发出请求或者应答网络请求。在Internet上的主机一般运行了多个服务软件 ， 同时提供集中服务。每种服务都打开一个Socket，并绑定一个端口，不同的端口对应于不同的服务。套节字之间的连接过程可以分为三个步骤：服务器监听，客户端请求，连接确认。

Spring是一个开源框架，它由Rod Johnson创建。它是为了解决企业应用开发的复杂性而创建的。Spring使用基本的JavaBean来完成以前只可能由EJB完成的事情。然而，Spring的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何Java应用都可以从Spring中受益。

多线程指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。java中的具体实现方法继承thread类，启动服务器端进行端口的监听，同时进行业务代码的处理。

**4.2欢迎引导页的设计与实现**

1）引导页的效果如图4-1所示：

图4-1 引导页



图4-2 引导页



图4-3 引导页

**4.3前台子系统的设计与实现**

**4.3.1 用户注册模块的设计与实现**

1）用户注册模块的功能

注册模块：实现用户注册功能，用户必须要使用手机号和密码登陆系统后才能使用系统进行其他操作，手机号和密码需要使用系统的注册模块进行注册。

2）用户注册模块的实际效果

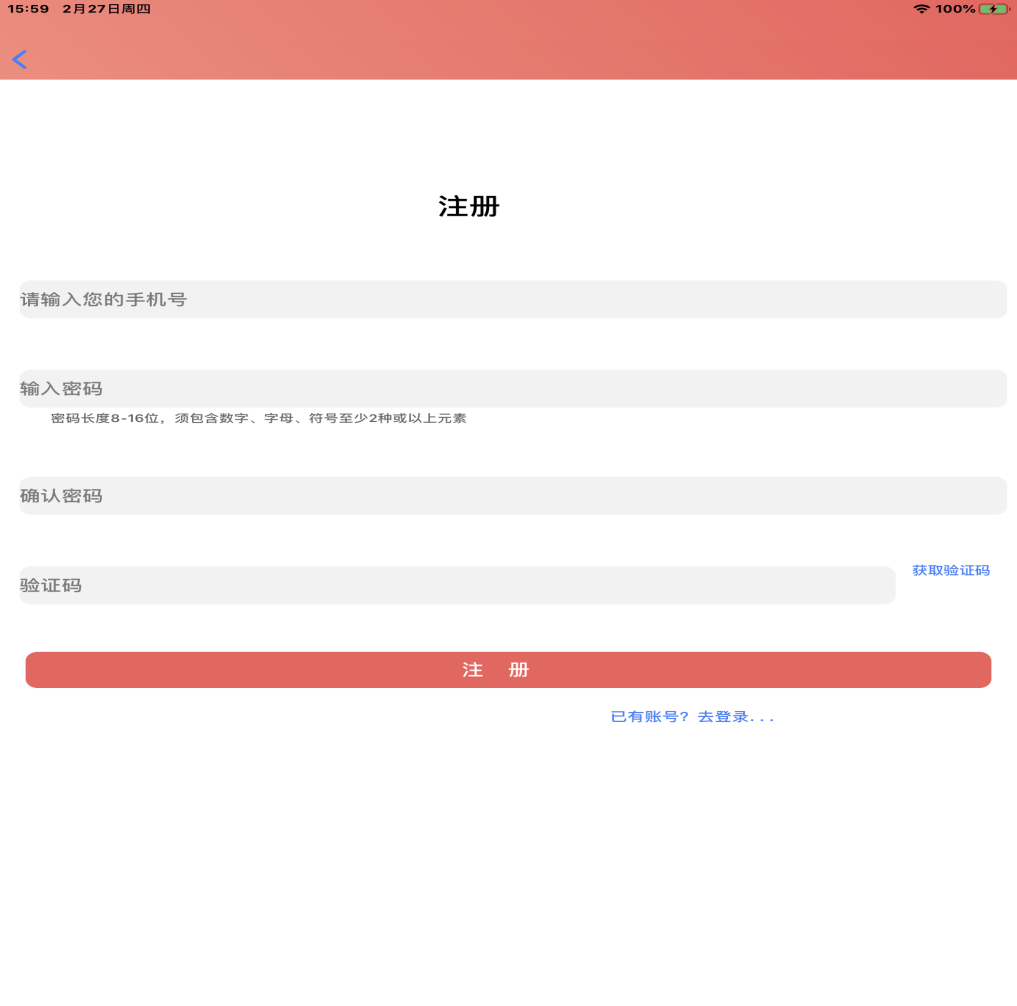


图4-4 注册界面

在此页面，用户需要按要求填写的信息，信息包括手机号、密码，确认密码和验证码，用户正确输入后点击注册按钮，注册数据会经过验证提交到后台并写入数据库，如果用户填写错误系统将不能提交注册。

**4.3.2 用户登录模块的设计与实现**

1）用户登录模块的功能

用户登录模块：登录模块实现的是登录功能，用户需要使用注册好的手机号和密码进行登录，登录是用户开始操作的前提，用户只有在登录状态下才可以浏览理财产品等。

2）用户登录模块的实际效果

用户登录界面如图4-5所示：



图4-5 登录界面

用户在完成注册后可以使用注册的手机号、密码和验证码登录系统，如果用户名或密码错误系统会提示用户重新输入。用户登录成功后会跳转到系统的首页，用户进入到登录状态，便可以使用系统导航栏的快捷按钮进入到理财，我的，已购模块等进行操作跳转进入首页的效果图：

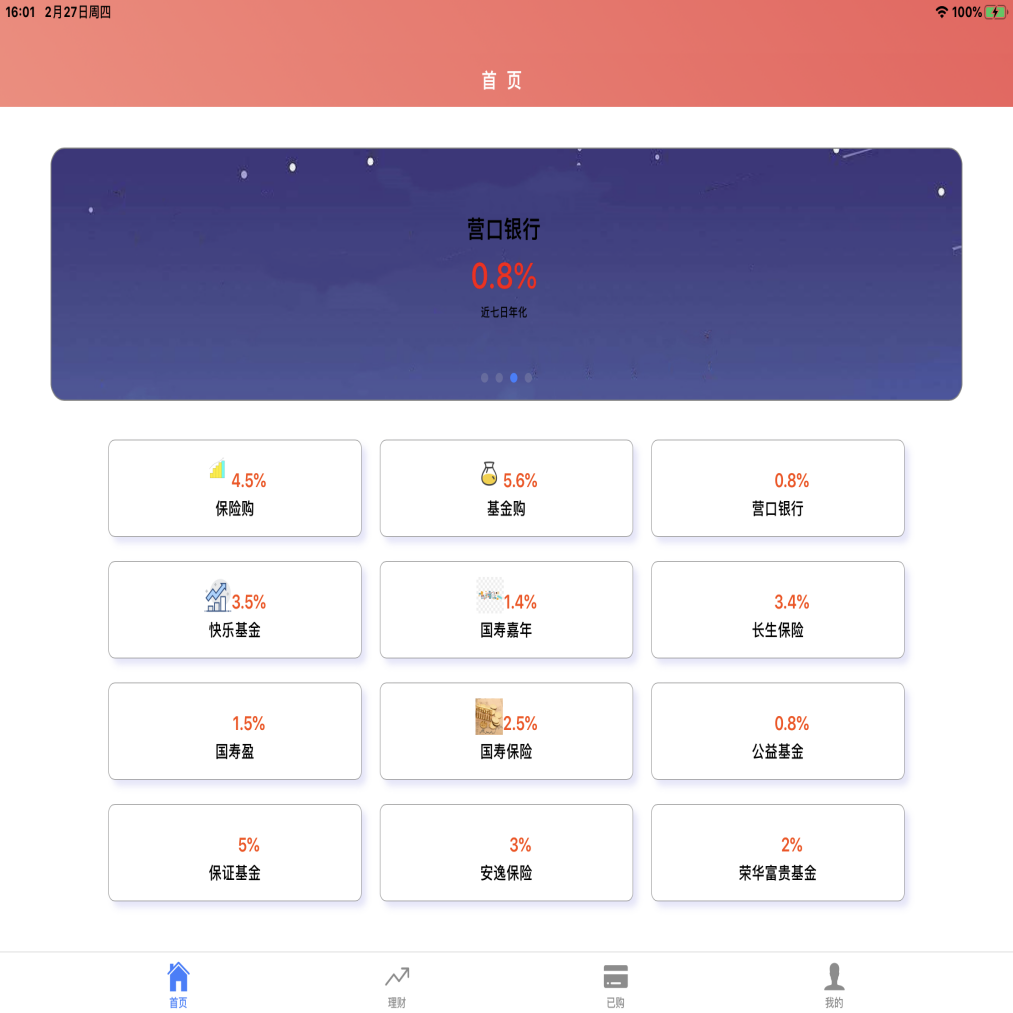


图4-6 首页

**4.3.3 申购赎回模块的设计与实现**

1）申购赎回模块的功能

申购赎回模块：申购赎回模块实现的是产品的申购和赎回的功能，用户需要使用注册好的手机号和密码进行登录，以及在开户和绑卡的状态下进行的操作登录是用户开始操作的前提，用户只有在开户状态下才可以购买理财产品等。

2）申购赎回模块的实际效果

申购赎回模块界面如图4-7所示

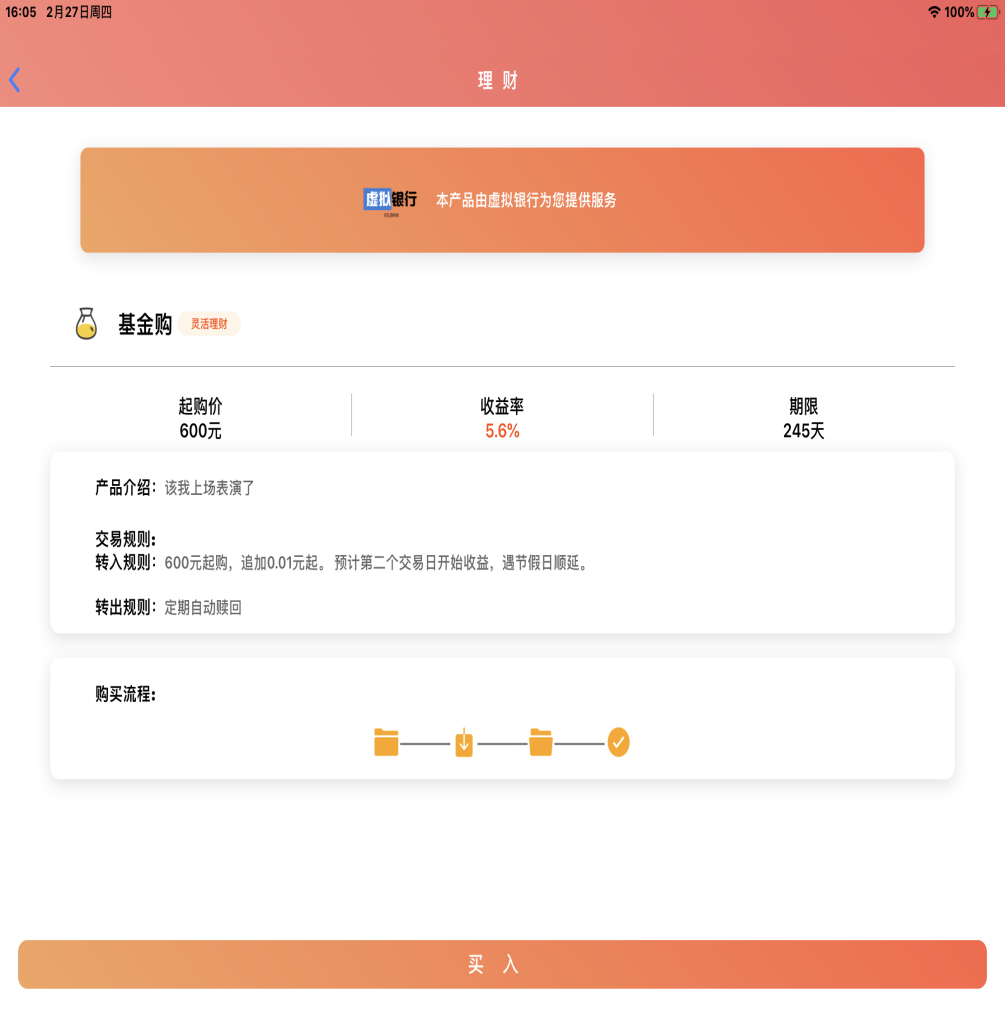


图4-7 申购赎回页

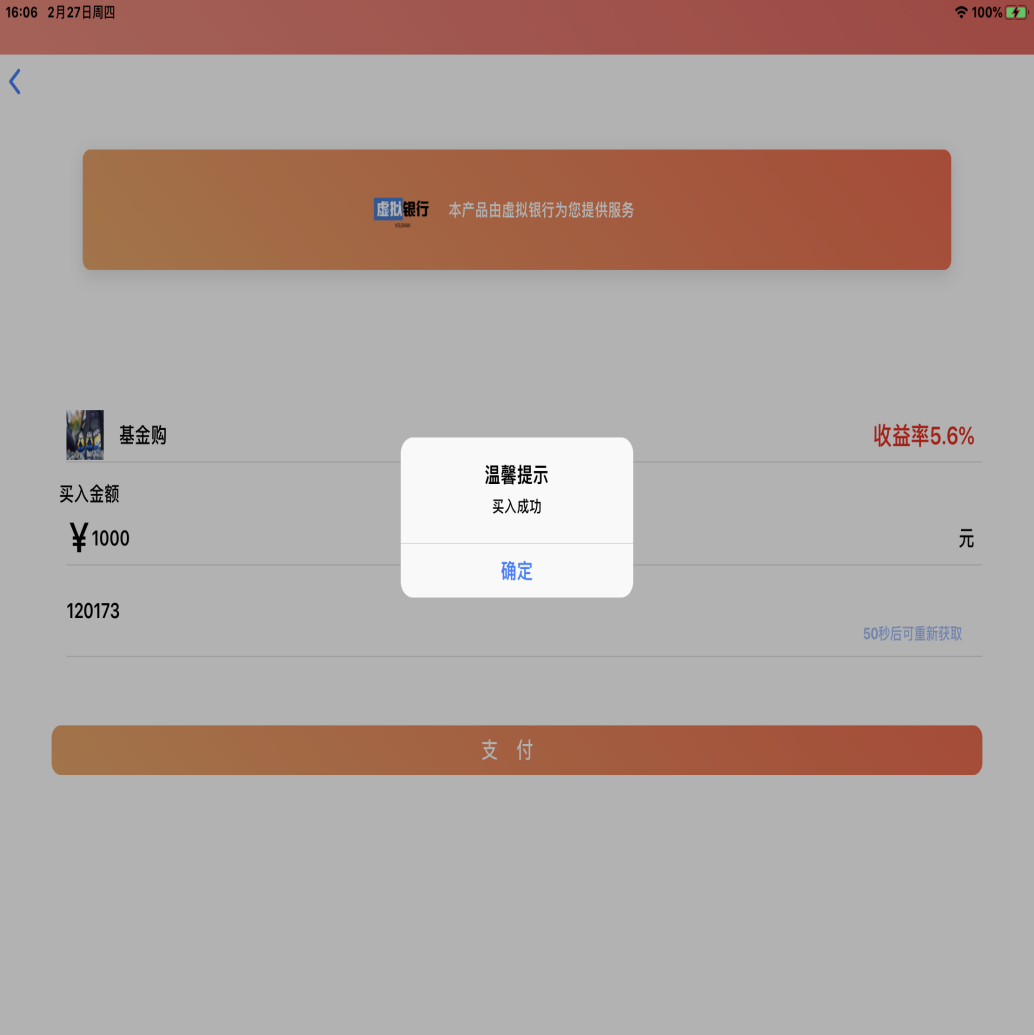


图4-8 申购赎回页



图4-9 申购赎回页

申购主要进行理财产品的购买和查看，可以查看历史收益率，以及产品的详情如起购日期和历史的收益率，最低起购额以及持有的时间等，赎回即对已经购买的产品进行赎回。

**4.3.4 理财及筛选产品模块的设计与实现**

1）理财及筛选产品模块的功能

理财及筛选产品模块实现的是产品信息的查看的功能，以及按照指定的持有日期和利率筛选产品。用户需要使用注册好的手机号和密码进行登录，登录是用户开始操作的前提，用户只有在登录进入系统下才可以查看和筛选理财产品等。

2）理财及筛选产品模块的实际效果

筛选产品界面如图4-10所示



图4-10 理财及筛选产品页



图4-11 理财及筛选产品页

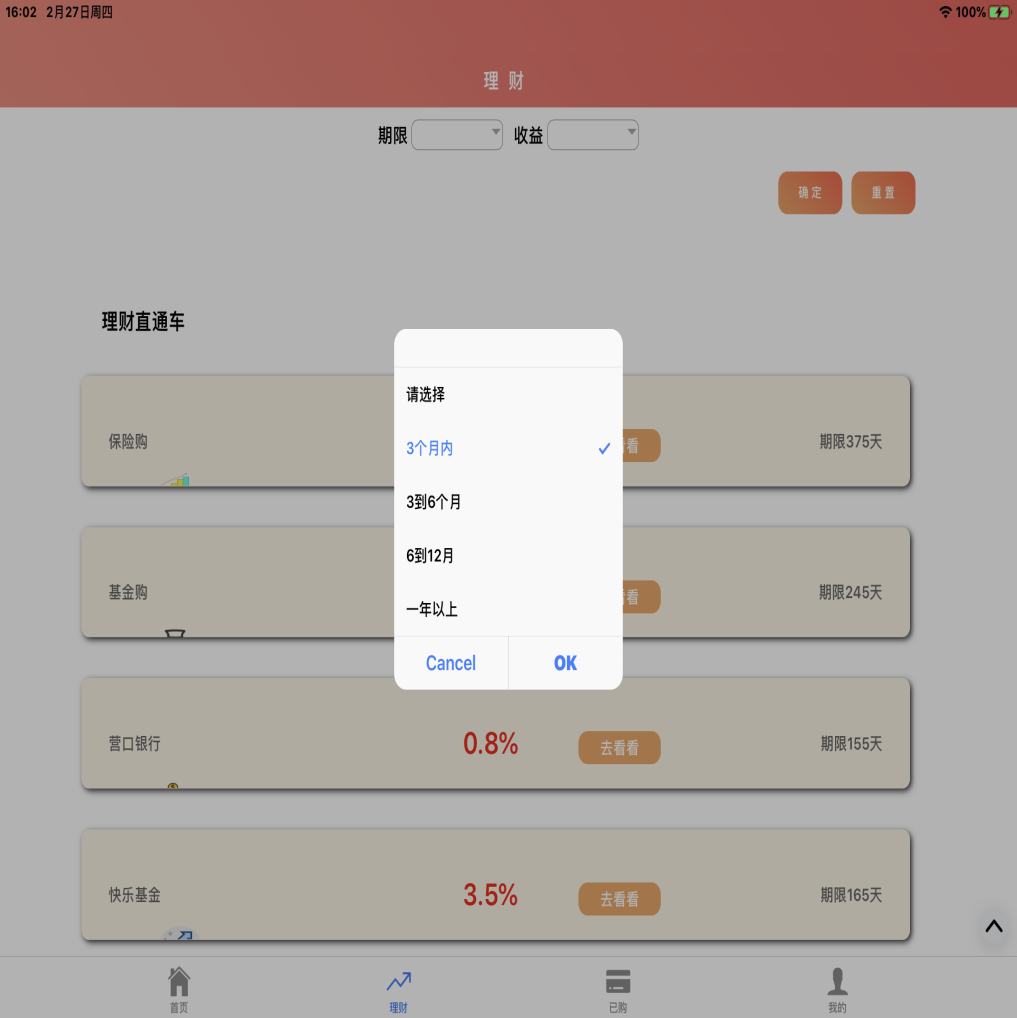


图4-12 理财及筛选产品页

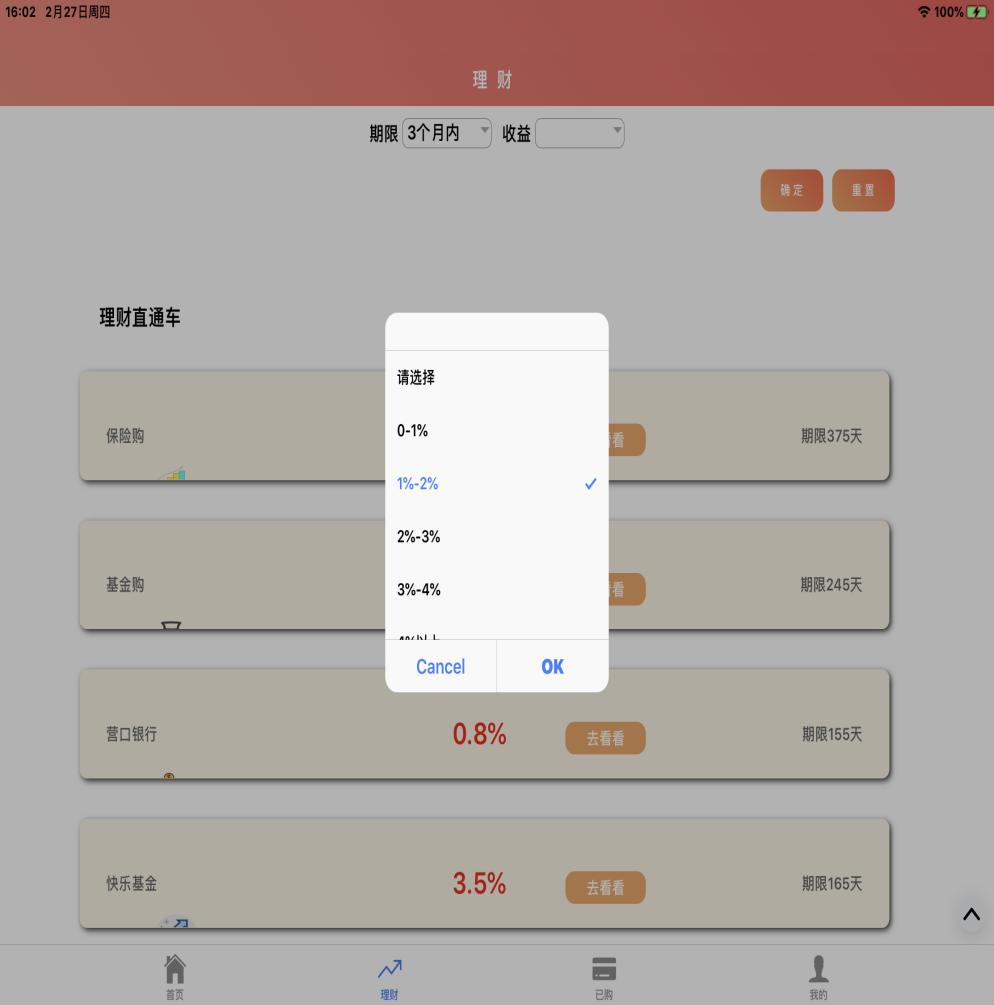


图4-13 理财及筛选产品页

用户可以对已经绑卡的资金进行投资和理财，进行申购时，用户输入起购额并且日期符合判断的要求的在账户里进行扣钱，买入的符合产品的规定而且值账户的大于买产品的前既可以申购成功，并且现有产品的收益和风险。如果进行赎回，用户失去产品的所有权，不再享受收益和承担亏损的风险，同时用户的账户转进买卖产品的钱同时加上利息。

**4.3.5 充值提现模块的设计与实现**

1）充值提现模块的功能

充值提现模块实现的是二类卡和一类卡中资金进行转移的功能，可以验证码核对成功后。用户需要使用转移的卡号和姓名进行比对，已经绑卡是用户开始操作的前提，用户只有在绑卡状态下下才可以输入正确的卡号并且核对成功。

2）充值提现模块的实际效果

充值提现界面如图4-14所示

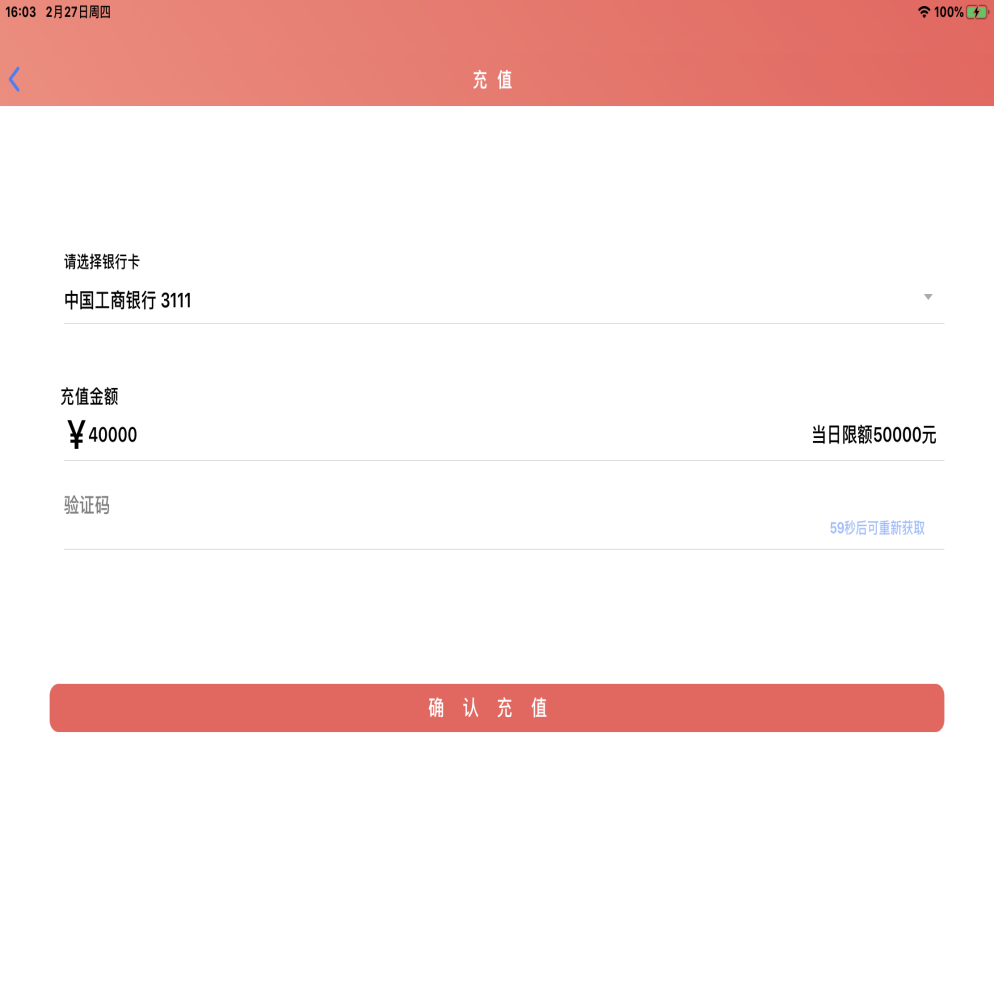


图4-14 充值提现页

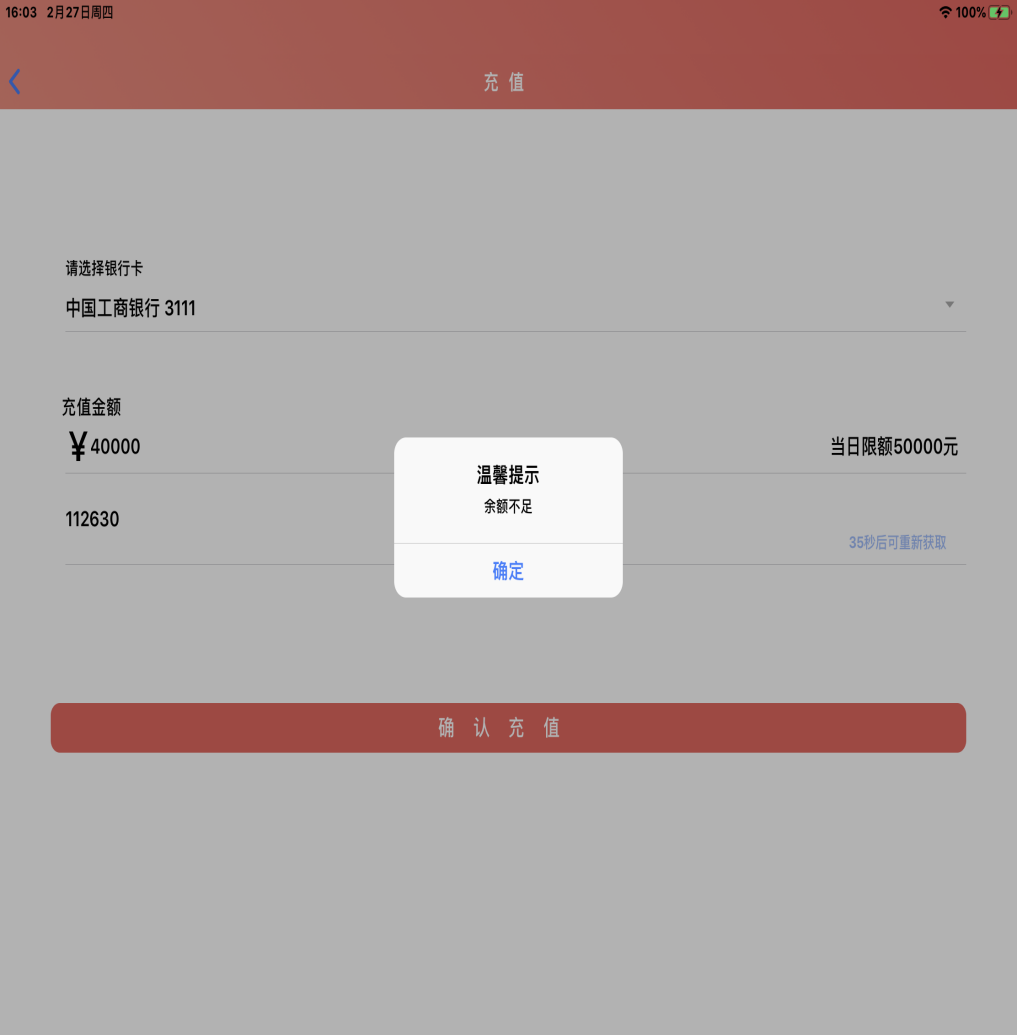


图4-15 充值提现页

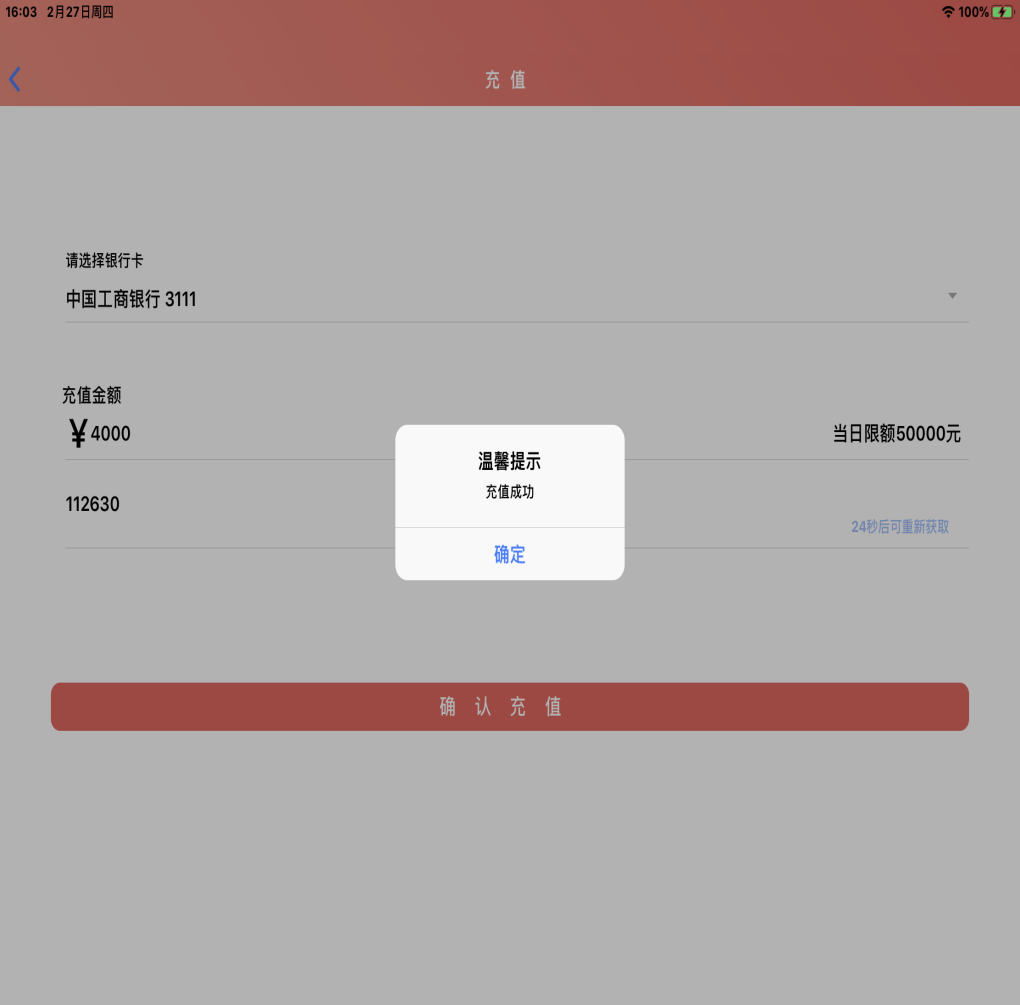


图4-16 充值提现页

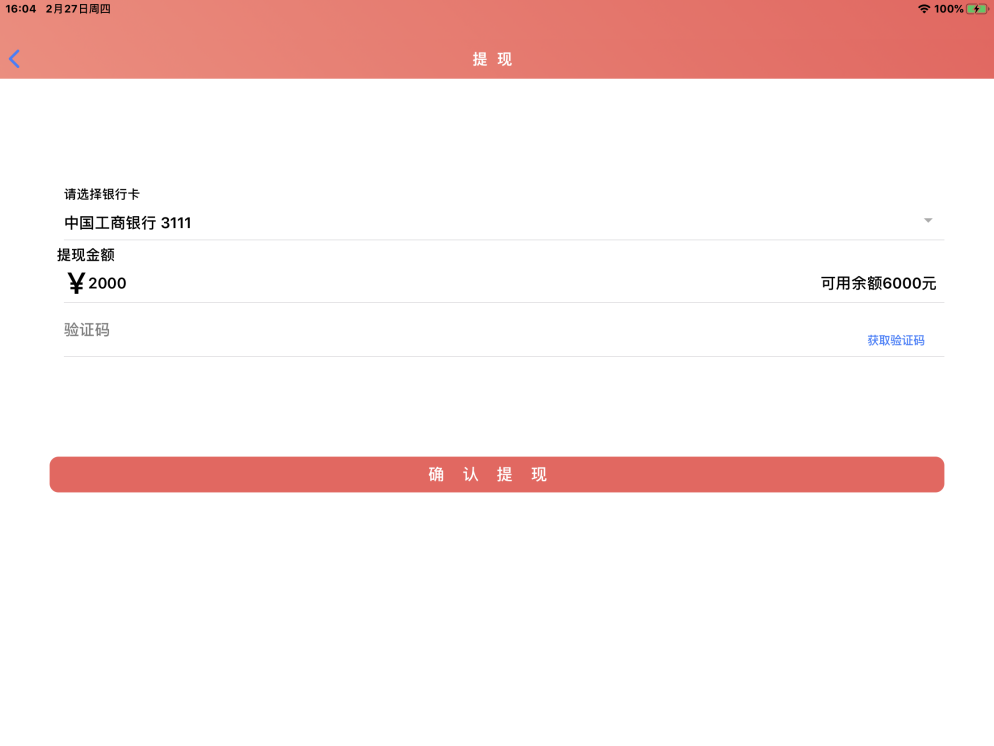


图4-17 充值提现页



图4-18 充值提现页

进入我的

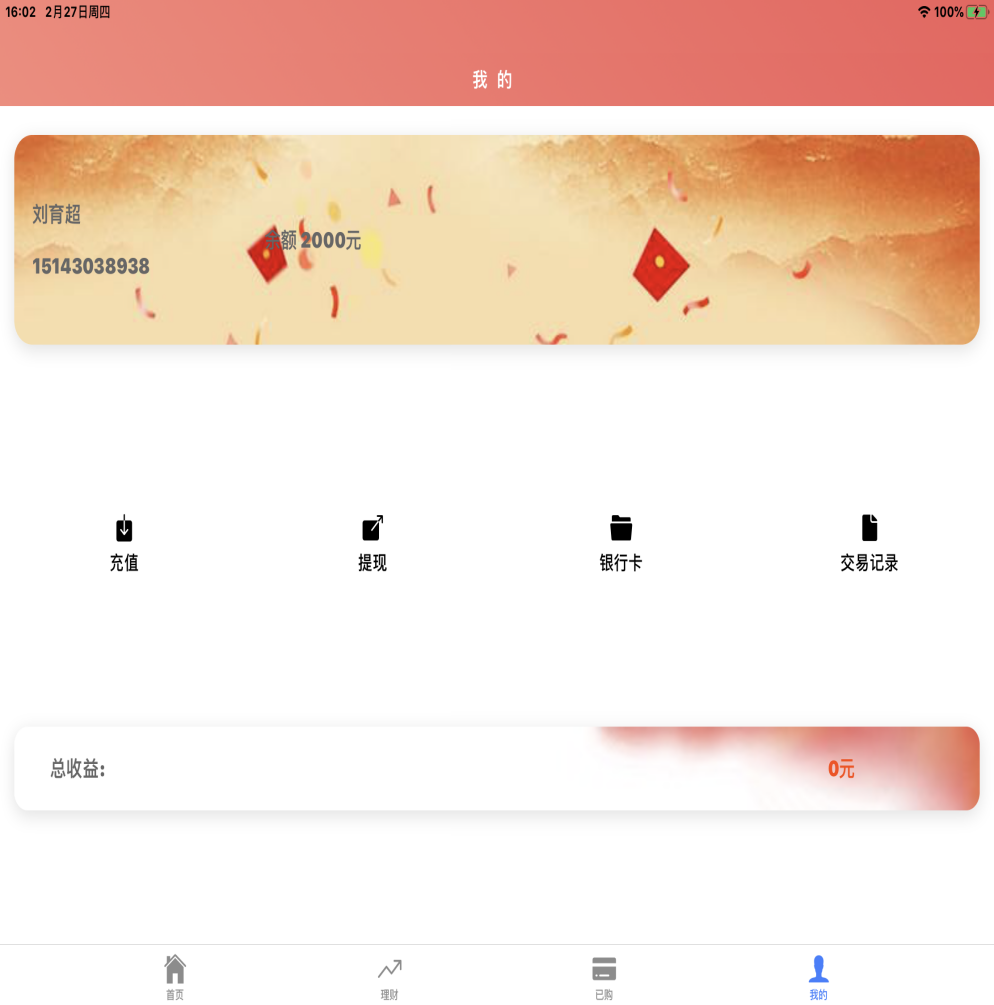


图4-19 我的页



图4-20 我的页

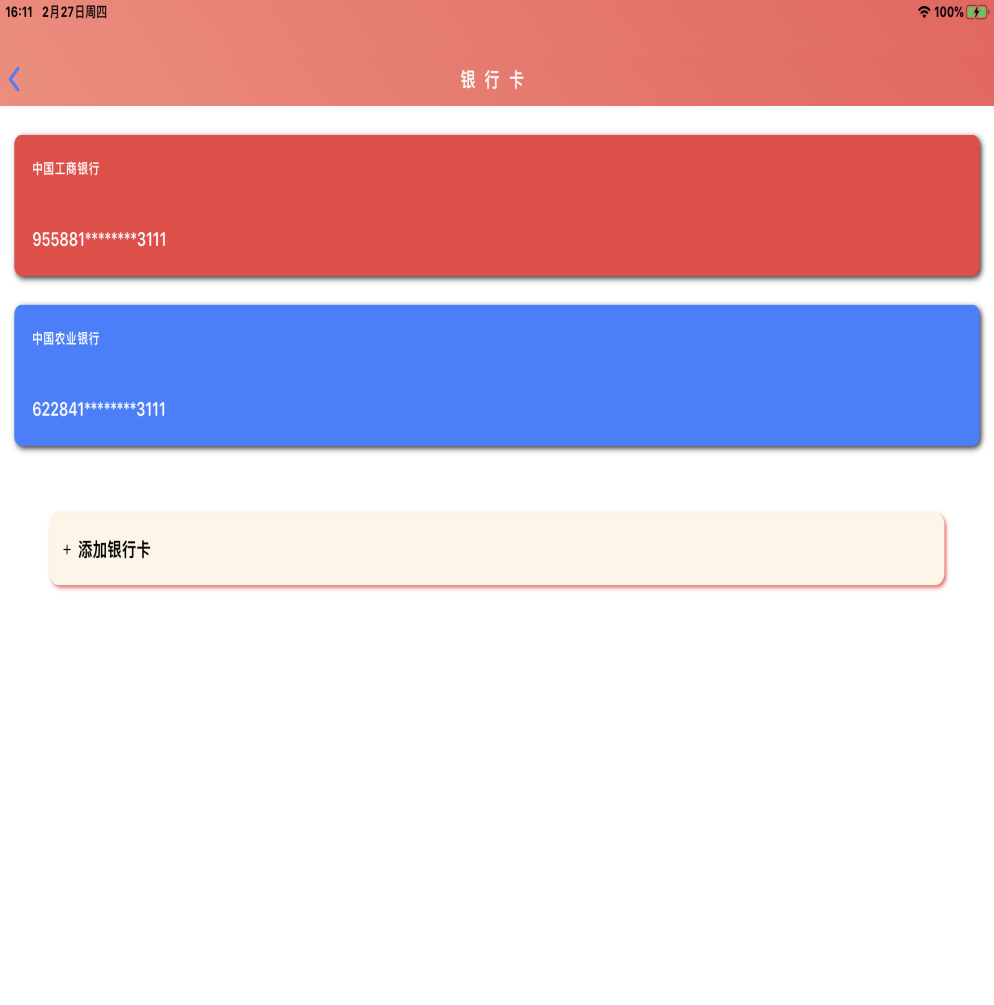


图4-21 我的页



图4-22 我的页



图4-23 我的页面

**4.4 后管系统的设计与实现**

后管系统主要作用是进行理财产品的上架和审核以及调整，分为三类典型的用户：他们分别是超级管理员和产品管理员和产品审核员，分别负责产品管理员和产品审核员的管理以及产品信息的填写和申请上架、审核，产品通过和驳回，他们的效果图如下：

**4.4.1 后管系统登录效果如图所示：**



图4-24 后管系统登录界面

**4.4.2 后管系统通过上架产品效果如图所示：**

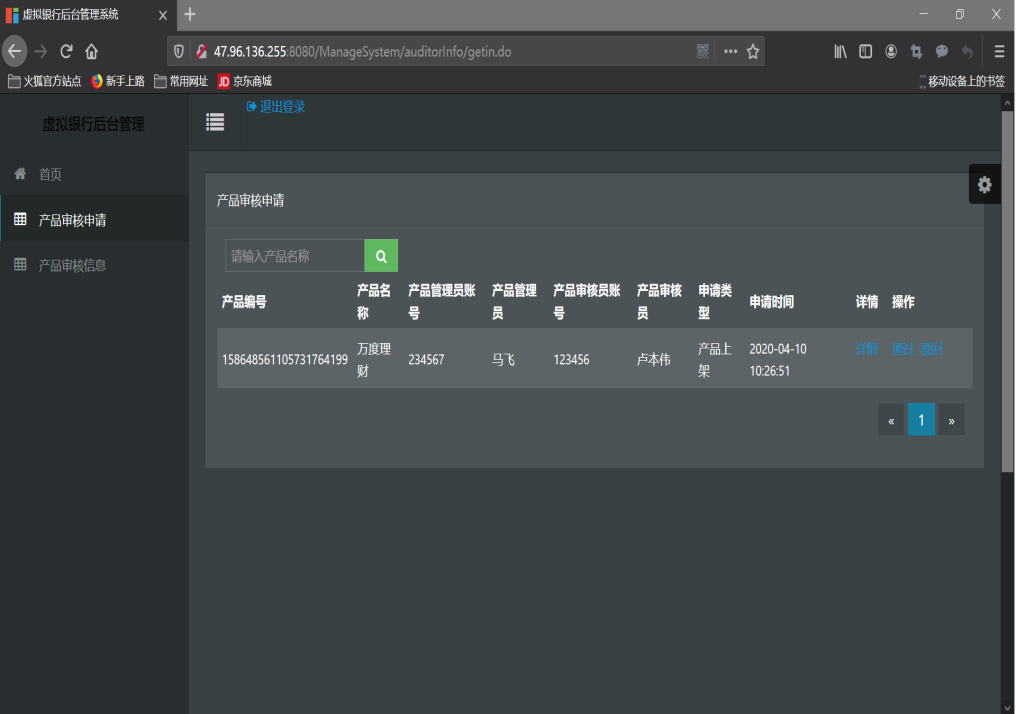


图4-25 后管通过产品页面

**4.4.3 后管系统上架产品效果如图所示：**

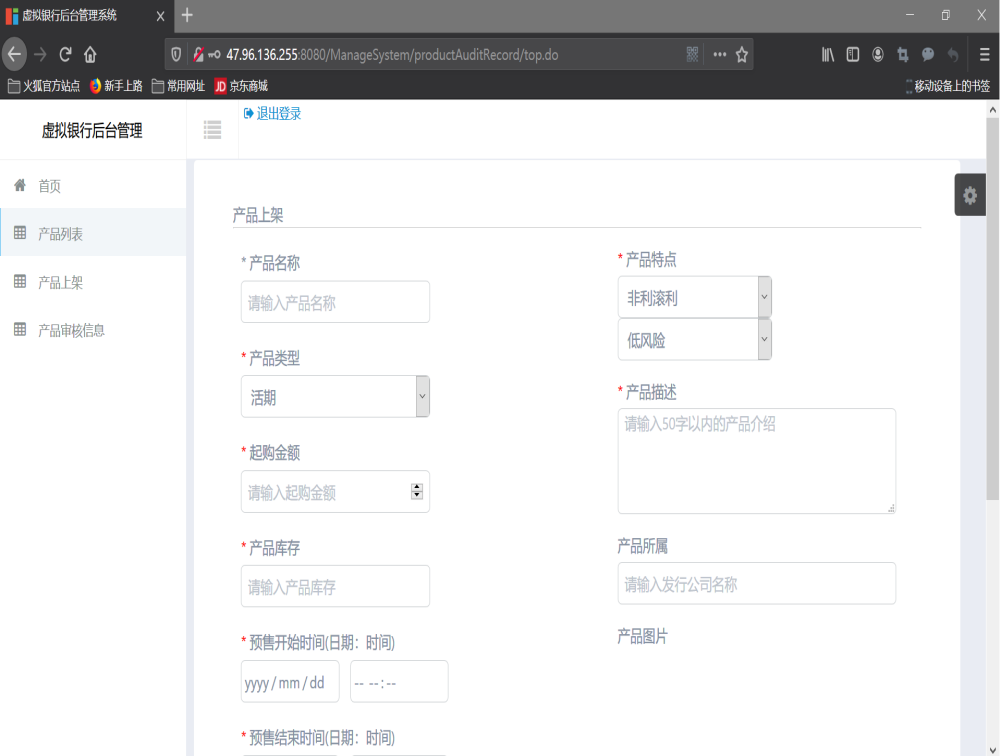


图4-26 后管上架产品页面

**4.4.4 后管系统产品审核结果效果如图所示：**



图4-27 后管产品审核结果页面

**4.4.5 后管系统产品列表效果如图所示：**

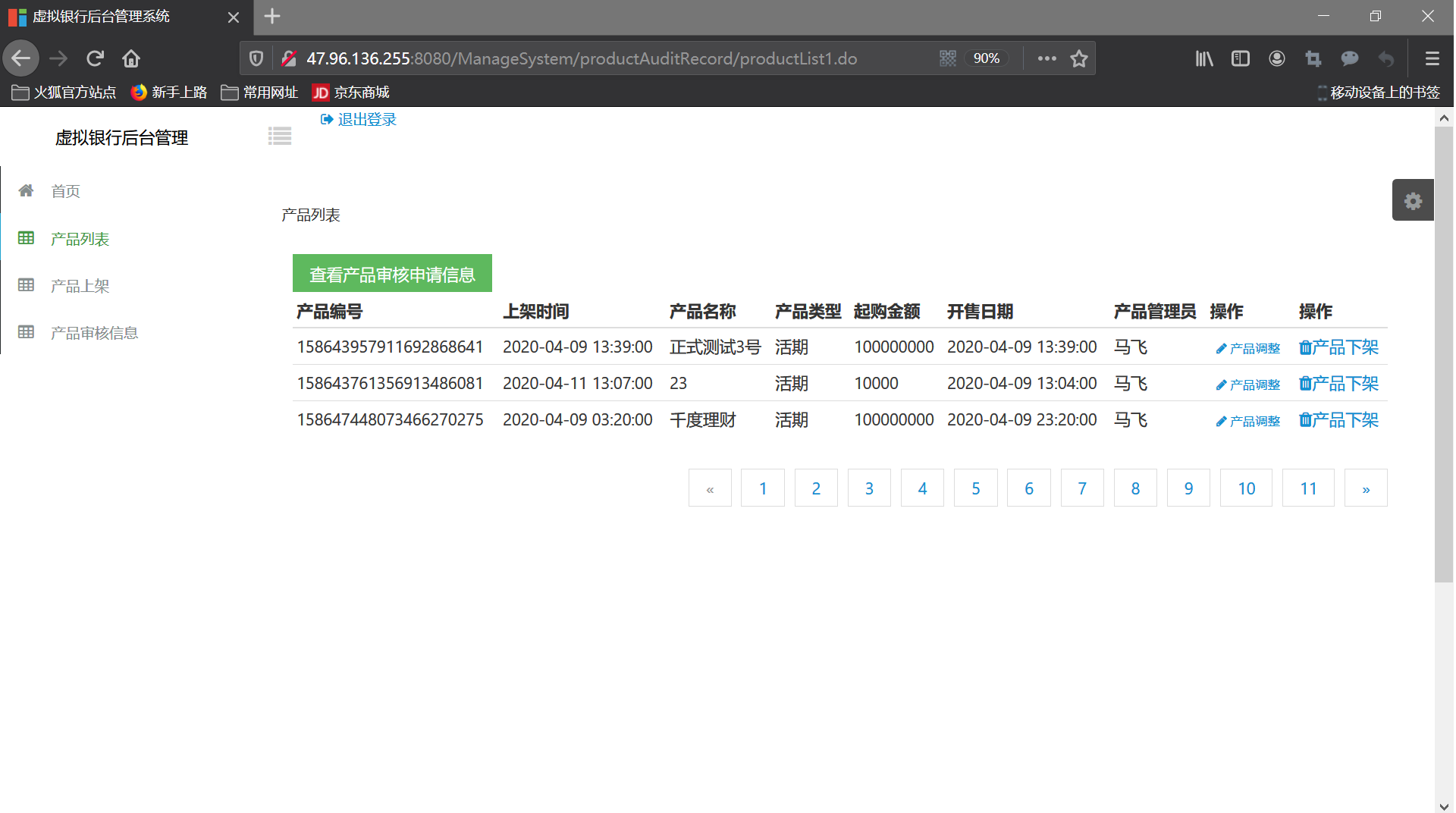


图4-28 后管产品列表结果页面

**4.5 安全功能的设计与实现**

充值提现，余额和和申购赎回等动账交易，都需要核心系统之间进行交互，而且一般不再局域网内，所以所有的交易和访问必须要考虑安全，手机银行客户端采用前后端访问加密和验签进行安全控制，系统之间交互采用MD5算法加密进行验签，确保所有的动账交易都安全的进行，能够有效的控制异常的访问和黑客的非法攻击，从而造成不可估计的损失。

**4.5.1加解密的设计与实现**

1. 加解密和验签模块的功能

加解密在于对传输的数据做一些位置的变化，字母的的替换等，双方可以利用实现商量好的密钥进行揭秘，从而确保数据的保密性和安全性。e

1. 关键代码的实现

//加解密工具类

public class CryptoUtils {

/\*\*

\* 加解密秘钥 偏移量

\*/

public interface CryptoParams {

/\*\*

\* 加密密钥

\*/

String key = "e2b8b25fad71409e";

/\*\*

\* 偏移量

\*/

String iv = "9e69e212bb37b587";

}

static String format= "";

Long time;

public static void main(String[] args) {

String time = String.valueOf(System.currentTimeMillis());

System.out.println(time+"wo");

System.out.println(encrypt("{\"key\":\"liu\"}"));

}

public static String encrypt(String data) {

try {

String key = CryptoParams.key;

String iv = CryptoParams.iv;

Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");

int blockSize = cipher.getBlockSize();

byte[] dataBytes = data.getBytes();

int plaintextLength = dataBytes.length;

if (plaintextLength % blockSize != 0) {

plaintextLength = plaintextLength + (blockSize - (plaintextLength % blockSize));

}

byte[] plaintext = new byte[plaintextLength];

System.arraycopy(dataBytes, 0, plaintext, 0, dataBytes.length);

SecretKeySpec keySpec = new SecretKeySpec(key.getBytes(), "AES");

IvParameterSpec ivSpec = new IvParameterSpec(iv.getBytes());

cipher.init(Cipher.ENCRYPT\_MODE, keySpec, ivSpec);

byte[] encrypted = cipher.doFinal(plaintext);

String resultStr = new BASE64Encoder().encode(encrypted);

resultStr = resultStr.replaceAll("\\+", "%2B");

return resultStr;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

public static String decrypt(String data) {

try {

if (data == null) {

return null;

} else if ("".equals(data)) {

return "";

} else {

data = data.replaceAll("%2B", "+");

String key = CryptoParams.key;

String iv = CryptoParams.iv;

byte[] encrypted1 = new BASE64Decoder().decodeBuffer(data);

Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");

SecretKeySpec keySpec = new SecretKeySpec(key.getBytes(), "AES");

IvParameterSpec ivSpec = new IvParameterSpec(iv.getBytes());

cipher.init(Cipher.DECRYPT\_MODE, keySpec, ivSpec);

byte[] original = cipher.doFinal(encrypted1);

String originalString = new String(original).trim();

return originalString;

}

} catch (Exception e) {

return data;

}

}

}

**4.5.2验签的设计与实现**

1. 验签模块的功能

验签的原理在于接收方检查接受的数据的真实性，没有在中间别黑客修改和恶意 的访问

1. 关键代码的实现

//生成时间戳

String timeStamp = String.valueOf(System.currentTimeMillis());

//生成签名并MD5加密

String preAutograph = timeStamp + "&" + serialId + "&" + realName + "&" + idCard + "&" + userMobile;

String autograph = DigestUtils.md5DigestAsHex(preAutograph.getBytes());

//调用函数传数据和签名并且接收数据和签名

stringMap1 = checkIdCardAndNameClient.openAccountClient(userMobile, realName, idCard,timeStamp,autograph,serialId);

//接收数据和签名

String status1 = stringMap1.get("status");

String timeStamp3 = stringMap1.get("timeStamp2");

String autograph3 = stringMap1.get("autograph2");

String s2 = timeStamp3 + "&" + status1;

String backAutograph = DigestUtils.md5DigestAsHex(s2.getBytes());

//验证签名

if (!StringUtils.equals(autograph3,backAutograph)){

Throw new TrainingException(Constants.Code.VERIFICATION\_CODE\_ERROR,Constants.Meaning.SYSTEM\_EXCEPTION);

}

**4.6 本章小结**

以上内容就是手机银行客户端系统的详细设计过程，系统分为前端和三个子系统，虚拟银行子系统、公安核心系统和互联网核心子系统，分别包含相应的模块。还突出了安全功能模块的设计与实现原理。本章详细介绍了本系统的各个页面的设计方式和结果，并介绍了用户和管理员在各自的子系统进行相应操作的基本流程，并且阐述了实现过程。

**第5章 系统测试**

**5.1 测试的目的**

软件测试是程序的一种执行过程，目的是尽可能发现并改正被测试软件中的错误，提高软件的可靠性。它是软件生命周期中一项非常重要且非常复杂的工作，对软件可靠性保证具有极其重要的意义。在目前形式化方法和程序正确性证明技术还无望成为实用性方法的情况下，软件测试在将来相当一段时间内仍然是软件可靠性保证的有效方法。软件工程的总目标是充分利用有限的人力和物力资源，高效率、高质量地完成软件开发项目。不足的测试势必使软件带着一些未揭露的隐藏错误投入运行，这将意味着更大的危险让用户承担。过度测试则会浪费许多宝贵的资源。到测试后期，即使找到了错误，然而付出了过高的代价。E.W.Dijkstra的一句名言说明了这一道理：“程序测试只能表明错误的存在，而不能表明错误不存在。”可见，测试是为了使软件中蕴涵的缺陷低于某一特定值，使产出、投入比达到最大。

在软件业较发达的国家。软件测试不仅成为软件开发的一个有机组成部分，而且在软件开发的系统工程中占据着相当大的比重。以美国的软件开发和生产的平均资金投入为例，通常是：“需求分析”和“规划确定”各占百分之三，“设计”占百分之五，设计占百分之五，编程占百分之七，测试占百分之十五，投产和维护占百分之六七十。测试在。软件开发中的地位不言而喻。 软件测试工程师和软件卡发工程师就像两兄弟，缺一不可，国内开发工程师招聘还是比较容易的，但在做一些大型项目时需要大量软件测试人员，因为目前高校里没有专业的专业，只能招聘后在进行培训，这就大大增加了企业的成本，所以企业还是希望有一批专业培训的人员能直接上岗。 软件测试是一个系列过程活动，贯穿于软件项目的整个生命过程，随着市场对软件质量的的要求不断提高，软件测试将变得越来越重要，

**5.2 测试的方法**

**5.2.1 黑盒测试**

系统测试属于黑盒测试，不接触代码，对已经集成好的软件系统进行彻底的测试，来验证软件系统的正确性和性能等是否满足需求规约中所指定的要求。系统测试应按照测试计划进行，其输入、输出和其他动态运行行为应与软件的需求规约进行对比。

功能测试主要是根据产品的需求规格说明书和测试需求列表对产品的各功能进行验证，根据功能测试用例逐项测试，验证产品的功能实现是否符合产品的需求规格。功能测试在测试工作中所占的比例最大，也叫黑盒测试。首先对各个模块进行黑盒测试，在各种条件下运行模块是否能正常运行。

**5.2.2 单元测试**

单元测试是针对软件的最小模块进行正确性检验的测试工作。是验证软件单元的实现是否和该单元的说明完全一致的相关联的测试活动组成的。根据软件单元的说明文档编写测试用例，对重要的接口、局部数据结构、边界条件、独立路径和错误处理路径，通过代码检查或执行测试用例有效地进行测试[30]。

对于Java语言的单元测试，本论文借助Junit单元测试包来完成，Junit是一个非常强大的单元测试包，可以对一个/多个类的单个/多个方法测试，还可以将不同的TestCase组合成TestSuit，使测试任务自动化，Junit设计的非常小巧，但是功能非常强大，并且整个框架设计良好，易扩展[31]。

**5.3 系统测试**

**5.3.1黑盒测试方案和效果**

1）前台登录模块测试

前台登录测试数据如表5-1所示，测试结果如图5-1、5-2所示。

表5-1前台测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能模块 | 前台登录 | | | | |
| 测试目的 | 验证是否输入合法的信息，允许合法登录，阻止非法登录 | | | | |
| 测试数据 | 电话号码15143038938密码123456 验证码 22222 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 输入电话号码，密码，验证码点击“登录”。 | 电话号码15143038938密码=123456验证码222222 | “在电话号码地址中包括11位多了空格” | 系统提示”电话号码输入不正确” | 正常 |
| 2 | 输入电话号码，密码，验证码点击“登录””。 | 电话号码15143038938密码=123456验证码222222 | “用户名或密码错误” | 系统提示“验证码或密码错误” | 正常 |
| 3 | 输入电话号码，密码，验证码点击“登录””。 | 电话号码15143038938密码=123456验证码222222 | 进入系统 | 进入系统 | 正常 |

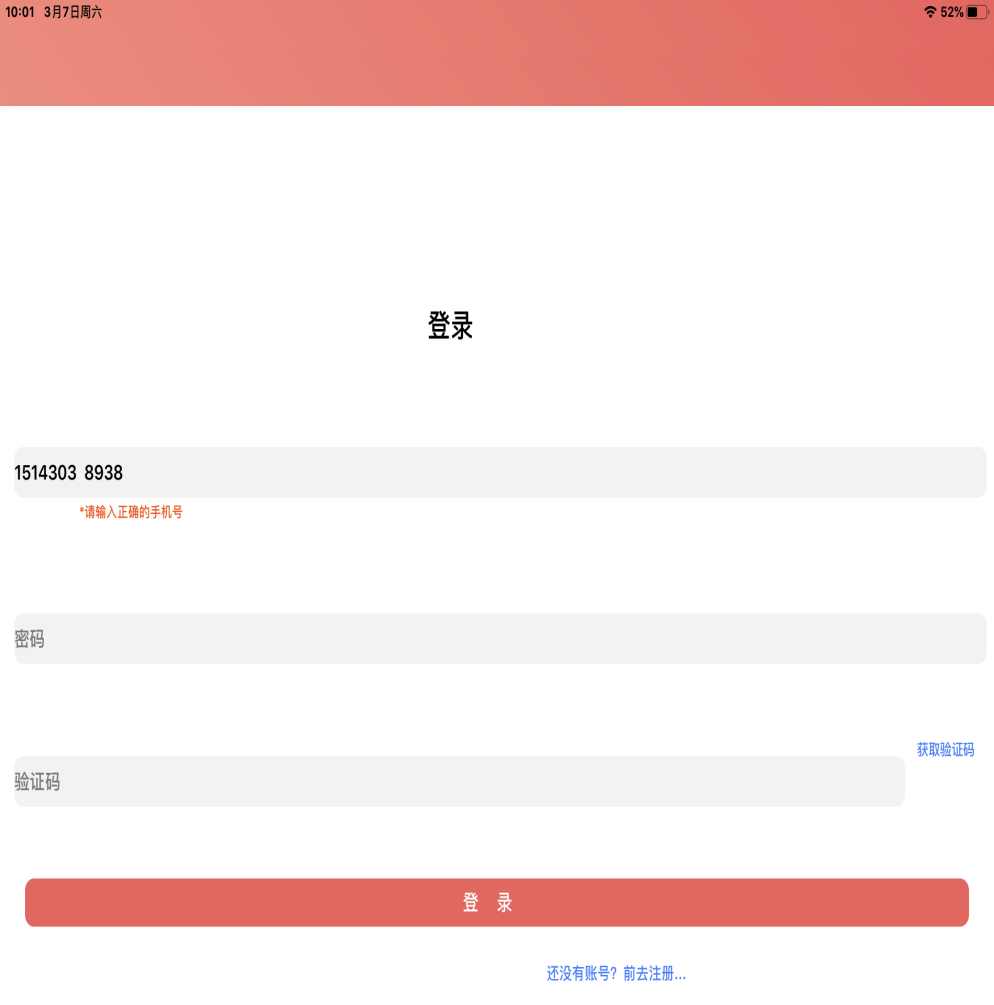


图5-1 手机号格式不准确结果

2）未开户理财界面中的产品查询功能测试

测试数据如表5-2所示，测试结果如图5-2、5-3所示。

表5-2测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能模块 | 界面中的关键字检索 | | | | |
| 测试目的 | 验证跳过开户，浏览对应的理财产品，弹出提示开户的提示框 | | | | |
| 测试数据 | 不进行开户进入产品页 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 在登陆的基础上不进行开户” | 姓名  身份证 | 弹出提示开户的提示框 | 弹出提示开户的提示框 | 正常 |
| 2 | 在按照正常流程进行开户” | 姓名  身份证 | 没有提示框，能够正常进行查询产品 | 没有提示框，能够正常进行查询产品 | 正常 |



图5-2 弹出提示开户结果



图5-3 进行开户结果

3）申购赎回模块功能测试

后台测试数据如表5-3所示，测试结果如图5-5、5-6、5-7、5-8所示。

表5-3后台测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能模块 | 申购赎回 | | | | |
| 测试目的 | 验证未绑卡申购赎回或者超过余额操作的结果 | | | | |
| 测试数据 | 未绑卡进入app界面 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 未开户进入首页中 |  | 出现您尚未绑卡时候绑卡 | 出现您尚未绑卡时候绑卡 | 正常 |
| 2 | 未开户进入已购中 |  | 出现您尚未绑卡时候绑卡 | 出现您尚未绑卡时候绑卡 | 正常 |
| 3 | 未开户进入申购中 |  | 权限不足尚未绑卡 | 权限不足尚未绑卡 | 正常 |



图5-4 未绑卡申购的测试结果



图5-5 未绑卡查询已购测试结果



图5-6 未绑卡查询已购测试结果

1. 充值提现模块功能测试，测试数据如表5-4所示。

表5-4后台测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能模块 | 充值提现模块 | | | | |
| 测试目的 | 确定充值提现超过限额以及未绑卡的情况 | | | | |
| 测试数据 | 省份下拉列表、市下拉列表 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 在转账和充值金额输入超过余额的数量 |  | 金额不足 | 金额不足 | 正常 |
| 2 | 在转账和充值金额输入超过单日限额的数量 |  | 理财剩余额度不足 | 理财剩余额度不足 | 正常 |



图5-7 充值提现测试结果

1. 后台我要上架调整下架理财产品功能测试

表5-5后台测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能模块 | 上架调整下架理财产品 | | | | |
| 测试目的 | 验证上架理财产品，审核员通过后，app产品页面出现该商品及对应信息且只有一个该商品，调整或下架对应信息显示 | | | | |
| 测试数据 | 输入上汽集团产品名，利率等基本信息修改或下架相关产品 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 在商品标题框中输入数据 | 上汽集团以及利率和期限和等信息 | 审核员通过后发布成功，且app显示单个该商品及对应信息 | 发布成功，且显示单个该商品及对应信息 | 正常 |
| 2 | 在商品描述框中输入数据 | 开售时间和结束时间在现在之后 | 审核员通过调整后，可以看到显示产品为待售状态 | 可以看到显示产品为待售状态 | 正常 |
| 3 | 在类别下拉框中选择数据 | 下架 | 审核员通过下架后，该产品不再显示 | 审核员通过下架后，该产品不再显示 | 正常 |

**5.4 本章小结**

经过上述的一系列测试，发现了部分bug，但已修改成功，手机银行客户端系统达到了系统要求的效果，表明了系统在开发过程中严格遵照了需求分析等文档，严格按照规范完成网站的设计和开发。实现了美观友好的交互界面，完善的功能，易于上手的操作流程，可以满足用户充值转账，申购投资理财等的基本要求，为用户提供良好服务。

**结 论**

银行首先就是为了满足其金融需求，而最基础的金融需求就是账户查询、转账（是大额转账）等交易，目前客户选择办理业务的主要渠道就是手机银行

但是目前手机银行的功能，依然偏格式化，缺乏温度和情感。其实围绕账户查询、转账和支付，可以建立有温度的互动和连接。比如账户查询，能否做些人性化的优化？改变当前单调、冰冷的数字展示？能否增加一些温情的问候和关怀？基础交易，比如转账交易，一般是大额，客户需要反馈机制，但是反馈的话术可以更有温度，更接地气，而不是冰冷的、格式化的模板和数字，能否用转账来构建起有趣的互动和和有温度的交流？比如来账时的红包雨，往账时候的小火箭等等，对于转账是否有规律，比如定期的还信用卡，还房贷，或者给孩子的生活费，这里面大有文章可以挖掘。再比如客户的登录时间，是否有迹可循，而消息推送需要遵循客户的登录节点，而不要盲目节点推送引发客户不满

把复杂留给后台和系统，把简单留给前台和客户。在简化交易的同时，要强化情感的连接。互联网的本质和初心是打破时空边界，拉近心灵距离，激发情感共鸣。手机银行也要在优化基础交易功能上，进一步增加手机银行平台的互动性、趣味性和情感化，更加注重社交化、游戏化、场景化，使手机银行真正实现由“金融工具”向“生活服务平台”的转变，让服务真正实现“不仅能听得见、看得见，还能感觉到” 过去在连接场景、触达客户方面，手机银行上已经做了比较多的工作，但是行内的分支机构及客户经理的作用尚未完全发挥，过去不论是业务线上化、网络化还是智能化、数字化，更多是总行在推动，未来要给分支机构赋能，把手机银行真正做成平台，网金搭台，分行唱戏，就像淘宝平台，发挥商家的作用。

客户经理与客户在线上的关系，需要建立一个有效的触达和持久的连接，不能开户过后就完了，再转用微信联系，可以从小处入手，建立客户经理与客户之间沟通互动的平台和圈子，如果没有太强的实力，可以不用像招行那样花大力气引进内容生产者，手机银行搭平台，给行员、客户提供展示的平台和机会，给分支机构赋能，给客户经理赋能，给客户自身赋能。

**参考文献**

[1]陈惠. 手机银行系统设计与实现[D].江西财经大学,2019.

[2]马俊. 基于J2EE的商业银行客户关系管理系统的设计与实现[D].电子科技大学,2013.

[3]韩凇. 手机银行系统的设计与实现[D].山东大学,2013.

[4]赵剑. 基于Android的手机银行系统的设计与实现[D].西安电子科技大学,2014.

[5]胡达. 手机银行支付安全加密系统的设计与实现[D].湖南大学,2016.

[6]陈鑫. 基于iOS平台的移动银行系统软件设计与实现[D].广西师范大学,2017.

[7]徐皓. 某商业银行智能终端手机银行的设计与实现[D].吉林大学,2014.

[8]倪春云. 基于J2EE平台的商业银行网上银行系统的设计与实现[D].电子科技大学,2014.

[9]顾朋生. 新型移动支付安全策略及技术研究[D].宁波大学,2012.

[10]江晓玲. 网上企业银行系统的设计与实现[D].电子科技大学,2012.

[11]Gutierrez F. Pro Spring Boot[M]. Apress, 2016.

[12]Webb P, Syer D, Long J, et al. Spring boot reference guide[J]. Part IV. Spring Boot features, 2013, 24.

[13]Walls C. Spring Boot in action[M]. Manning Publications, 2016.

[14]Gu M, Ye H, Yang Y. Design and Implementation of Affordable Housing Qualification Management System[J]. 2019.

**致 谢**

六月，总是阳光灿烂，但也总要曲终人散。今年六月，我们拒绝伤感。花儿谢了芬芳，迎来硕果飘香。毕业带来别离，我们走向辉煌。毕业了，挥挥手，祝朋友一路走好！。朋友再见，有缘千里来相会。同学再会，功成名就庆功酒。一月毕业设计以来，我从查资料选题到调研分析、撰写需求分析、概要设计、详细设计、编写代码、测试、到最后完成论文的写作，终于完成了本次毕业设计的全部工作，心里充满感激！

整个毕业设计完成的过程中，遇到过各式各样的问题与错误，精力很多的纠结和彷徨，每个问题的求解背后总是遇到很多的热心人得到的各种帮助和指教。

在这里我要感谢我的论文指导老师谷钰老师，从毕业设计一开始，就制定了很好的执行计划和方向，不断让我们学习做事的方法和思维，能够在恰当的地方给与适当的指导。对我而言，不仅完成了一个毕业的任务，更是在自己的心中形成了良好的学习习惯和思维，遇到问题不仅是请教老师，更多的是自己查阅资料进行完善和精进，在这个路上老师不断的交给我们心法和解决问题的而逻辑，同时团结了同学之间的友谊和团队协作的能力，带着老师的教诲，我一定会更加努力，在学术上用心钻研。衷心祝愿老师身体健康，工作顺利。

感谢培养教育我的学校长春工业大学，学校里浓厚的学术氛围，舒适的学习环境及母校对我的关怀和帮助我将终生难忘。祝母校蒸蒸日上，勇创辉煌。

感谢四年中陪伴在我身边的同学，朋友，感谢他们为我提出有益的建议和意见 ，感谢他们在我有困难时及时伸出援助之手，感谢他们对我毕业设计的帮助，希望大家前程似锦。

感谢我的父母对我的关心和理解。他们无微不至的关怀，孜孜不倦的教诲，使我受益终身，愿父母笑逐颜开，身体健康！最后，衷心地感谢在百忙之中评阅论文和参加答辩的各位专家、老师！