华中科技大学

软件工程课程设计

银行信息处理系统

院系 计算机科学与技术学院

班级 CS1607

学号 U201614702

姓名 樊俊超

指导老师 冯琪

目录

[1问题定义及可行性分析 3](#_Toc531295761)

[1.1问题定义 3](#_Toc531295762)

[1.2可行性分析 4](#_Toc531295763)

[1.2.1技术上可行性 4](#_Toc531295764)

[1.2.2经济上可行性 5](#_Toc531295765)

[1.2.3管理上可行性 6](#_Toc531295766)

[1.2.4分析结果 6](#_Toc531295767)

[2需求分析 7](#_Toc531295768)

[2.1业务需求 7](#_Toc531295769)

[2.2功能需求 7](#_Toc531295770)

[2.3数据字典 9](#_Toc531295771)

[2.4性能需求 10](#_Toc531295772)

[2.5数据流图 11](#_Toc531295773)

[2.6状态转化图 14](#_Toc531295774)

[3概要设计与详细设计 15](#_Toc531295775)

[3.1系统数据结构设计 15](#_Toc531295776)

[3.2系统功能模块图 17](#_Toc531295777)

[3.3核心模块流程图 18](#_Toc531295778)

[4测试报告 21](#_Toc531295779)

[5项目管理 26](#_Toc531295780)

[6体会 28](#_Toc531295781)

# 1问题定义及可行性分析

## 1.1问题定义

最早的银行业发源于西欧古代社会的货币兑换业。最初货币兑换商只是为商人兑换货币，后来发展到为商人保管货币，收付现金、办理结算和汇款，但不支付利息，而且收取保管费和手续费。随着工商业的发展，货币兑换商的业务进一步发展，他们手中聚集了大量资金。货币兑换商为了谋取更多的利润，利用手中聚集的货币发放贷款以取得利息时，货币兑换业就发展成为银行了。

我国银行业资产规模、税后利润逐年大幅增长，2011年中国银行业所实现利润占全球银行业总利润的近三分之一。我国银行业规模发展迅速，但是利率市场化加快、内外竞争加剧、盈利增速下滑背景下，银行金融机构必须在业务结构、资源配置以及区域布局上均做出相应战略性调整。而随着银行业竞争的不断加剧，银行业金融机构愈来愈重视对行业发展环境与市场需求的跟踪研究，特别是对银行业务发展环境和客户需求趋势变化的深入研究。正因为如此，我们急需设计一套基于发展实情和民众意愿的银行信息管理系统，来吸引更多的用户。

银行信息管理系统，是利用银行原有存储信息，经过安全检查，通过管理员操作权限和用户意愿进行信息操作和存储的计算机软件系统，它具备高效性，安全性，易操作性。一个银行用户在系统操作界面通过个人意愿进行个人财产存取，转账，贷款，换算外汇等操作，某些涉及银行本身利益的操作需要加入高级权限。

## 1.2可行性分析

可行性分析是通过对项目的主要内容和配套条件，如市场需求、资源供应、建设规模、工艺路线、设备选型、环境影响、资金筹措、盈利能力等，从技术、经济、工程等方面进行调查研究和分析比较，并对项目建成以后可能取得的财务、经济效益及社会环境影响进行预测，从而提出该项目是否值得投资和如何进行建设的咨询意见，为项目决策提供依据的一种综合性的系统分析方法。可行性分析应具有预见性、公正性、可靠性、科学性的特点。

### 1.2.1技术上可行性

技术可行性评价是通过原有系统和欲开发系统的系统流程图和数据流图，对系统进行比较，分析新系统具有的优越性，以及对设备、现有软件、用户、系统运行、开发环境、运行环境和经费支出的影响，然后评价新系统的技术可行性。

技术可行性包括以下几个方面：硬件，系统软件，应用软件，技术人员等方面。本管理系统开发环境为python3环境，Python是一种计算机程序设计语言。是一种动态的、面向对象的脚本语言，最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越来越多被用于独立的、大型项目的开发。基于python语言的简洁性、易读性以及可扩展性，使得基于python开发的本系统能具备高效地处理效率，

系统采用linux系统。系统管理人员要求：系统分析人员，开发人员，数据库管理人员，系统测试人员。银行信息管理系统已经有很多银行进行过相关开发和研究，有很多成功案例，因而技术上是可行的。

### 1.2.2经济上可行性

经济可行性是指可以使用的资源的可能性（资源包括人力资源、自然资源和资金条件）。任何一项公共政策都存在一个争取公共经济资源的问题。经济可行性包括两个方面的内容：一是某一备选方案占有和使用经济资源的可能性，进而实现政策目标的可能性；二是实施某一政策方案所需花费的成本和取得的收益。政府的财政资源是有限的，任何政策方案占有和使用的经济资源也是有限的。因此，任何一项公共政策都存在一个争取公共经济资源的问题。一般说来，“公共政策的经济资源的占有量与其政策目标的期望值成正比例关系。”当然，这还涉及到一个成本效益问题。如果某一方案的成本大于收益，显然这项政策是不可行的。

由于本系统所需设备资源仅一台电脑，开发和测试人员全由我一人担当，但设计出来的系统具备获益的可能，所以在经济上是可行的。

### 1.2.3管理上可行性

主要是管理人员对开发信息系统是否支持，现有管理制度是否科学，规章制度是否齐全，原始数据是否正确等。银行信息管理系统比较完善的规章制度和管理方法为系统的建设提供了保障，一个软件设计的好坏，关键在于产品的可靠，稳定，便于维护，系统采用模块化程序设计方法，既便于系统功能的各种组合和修改，又便于未参与开发的技术维护人员进行补充、维护。

### 1.2.4分析结果

为了开发本管理系统，开发设计人员参考了银行领域的现有的管理系统，通过对现有管理系统的实际操作，来确定管理系统的优点和可能存在的弊端，从而在开发系统时进行一定的优化处理。

前景根据以上分析，我们认为开发此系统的条件已经具备，可以开始进行开发。并且我们相信此软件经济实惠，肯定具有良好的前景。.我们将会逐步完善该软件,尽可能地让满足客户的需求。让客户对自己的资产信息有更为全面细致的认识。同时保障用户的投资财产安全，在贷款和外汇方面给予一定的帮助，同时保障用户个人信息不受不法分子的关注。因此本银行信息管理系统是可行的。

# 2需求分析

根据用户的需求，给出系统分析的结果，明确给出系统实现后的功能定义，角色定义等，设计系统的逻辑关系，系统接口以及达到系统性能的措施及解决办法等。采用结构化分析方法时要有系统的详细的数据流图和核心数据的数据字典，必要时进一步画出系统的E-R图和状态转换图。采用面向对象分析方法时画出系统详细的用例模型和对象模型，必要时进一步写出系统的动态脚本或画出系统的事件跟踪图。

## 2.1业务需求

业务需求表示组织或客户高层次的目标。业务需求通常来自项目投资人、购买产品的客户、实际用户的管理者、市场营销部门或产品策划部门。业务需求描述了组织为什么要开发一个系统，即组织希望达到的目标。

此次开发银行信息管理系统，一是为了设计一个可以弥补日益更迭下用户对银行功能的需求的管理系统；再者是为了吸引更多的银行新用户，为银行带来更大的经济利润和客户来源。

## 2.2功能需求

功能需求（functional requirement）规定开发人员必须在产品中实现的软件功能，用户利用这些功能来完成任务，满足业务需求。在该银行信息管理系统中，针对用户可能的需要，我们在基础的银行功能上设计实现了更多的银行功能。其中基础银行功能包括开户，销户，存款，取款，转账，密改，挂失，解挂失，查询等功能；除此之外，本系统添加了外汇兑换，信用贷款等高级功能。

1. 开户：根据用户输入信息生成新卡号并建立新用户档案，根据身份证及用户姓名更新信息库数据。
2. 销户：用户登陆成功后，根据客户意愿销掉该银行卡。
3. 存款：用户登陆成功后，更新用户该卡余额信息。
4. 取款：用户登陆成功后，根据余额情况更新用户该卡余额信息。
5. 转账：只接受本行转账，用户登录成功后，寻找对方卡号，若存在，根据实际情况更新两卡余额信息
6. 密改：修改用户一张银行卡的密码
7. 挂失：将银行卡状态设为挂失状态，此状态下，无法进行任何操作，默认丢失
8. 解挂失：将银行卡状态设置为使用状态，此状态下，可以更新卡内信息。
9. 查询：查询一张银行卡内余额，若有贷款，提示偿还贷款，可优先用卡内余额抵消部分或全部。
10. 外汇：提供当前外汇汇率，提供存取外汇选择，并更新卡内余额
11. 贷款：为未贷过款的行卡提供贷款服务。
12. 个人：由管理员操作，输出一张身份证对应的所有行内卡信息。
13. 资料：输出所有卡片信息。

## 2.3数据字典

数据字典（data dictionary）是对于数据模型中的数据对象或者项目的描述的集合，这样做有利于程序员和其他需要参考的人。分析一个用户交换的对象系统的第一步就是去辨别每一个对象，以及它与其他对象之间的关系。这个过程称为数据建模，结果产生一个对象关系图。当每个数据对象和项目都给出了一个描述性的名字之后，它的关系再进行描述（或者是成为潜在描述关系的结构中的一部分），然后再描述数据的类型（例如文本还是图像，或者是二进制数值），列出所有可能预先定义的数值，以及提供简单的文字性描述。这个集合被组织成书的形式用来参考，就叫做数据字典。

|  |
| --- |
| 名字：card  别名：银行卡  描述：该系统操作对象，拥有自己唯一标识的属性卡号  定义：card=卡号+密码+余额+贷款+挂失标志，为class类  位置：存储到文件里以python字典读取 |

|  |
| --- |
| 名字：person  别名：用户  描述：该系统部分功能的使用者，拥有自己唯一标识的属性身份证号  定义：person=姓名+身份证号+联系方式+银行卡，为class类  位置：存储到文件里以python字典读取 |

|  |
| --- |
| 名字：users  别名：用户信息  描述：负责记录所有银行卡的数据  定义：以字典形式存在，键：值为银行卡号：class类  位置：存储到文件里以python字典读取 |

|  |
| --- |
| 名字：islock  别名：银行卡挂失标志  描述：作为银行卡是否丢失的判断标志  定义：False/True，分别表示未挂失和已挂失  位置：存储到文件里以python字典方式读取 |

## 2.4性能需求

1. 精度：软件主要用于管理，不是科学计算，要求计算的精度不是很苛刻，所以输入/输出数据精度的要求不是很高，由于python语言的特点，可以在某些时候显示浮点数。
2. 时间特性要求：软件运行的响应时间要求不超过1~2秒。
3. 灵活性：软件具有升级功能，以满足用户的需求

## 2.5数据流图

绘制数据流图的目的是为了方便开发人员与用户的交流，以及为系统的设计提供依据。数据流图的绘制过程通常是采用自顶向下，逐层分解，直到功能细化为止，形成若干层次的数据流图。银行信息管理系统的数据流图如下所示：

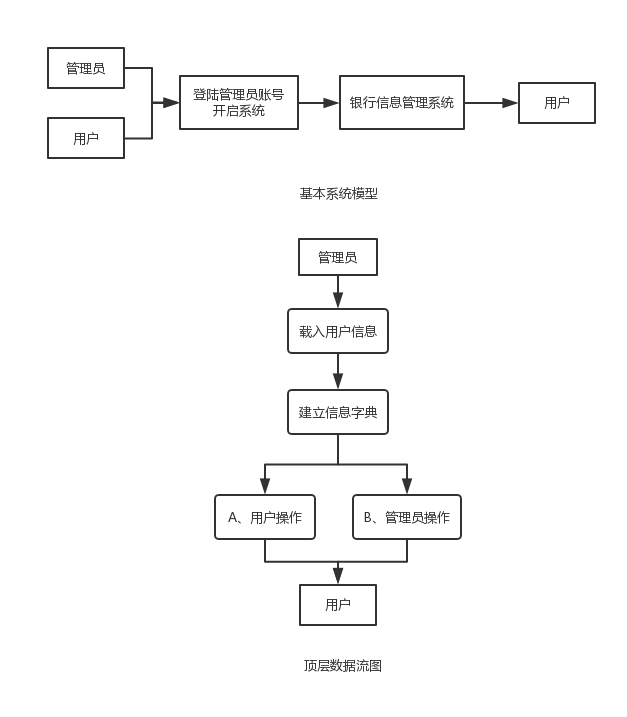


图2.5-1

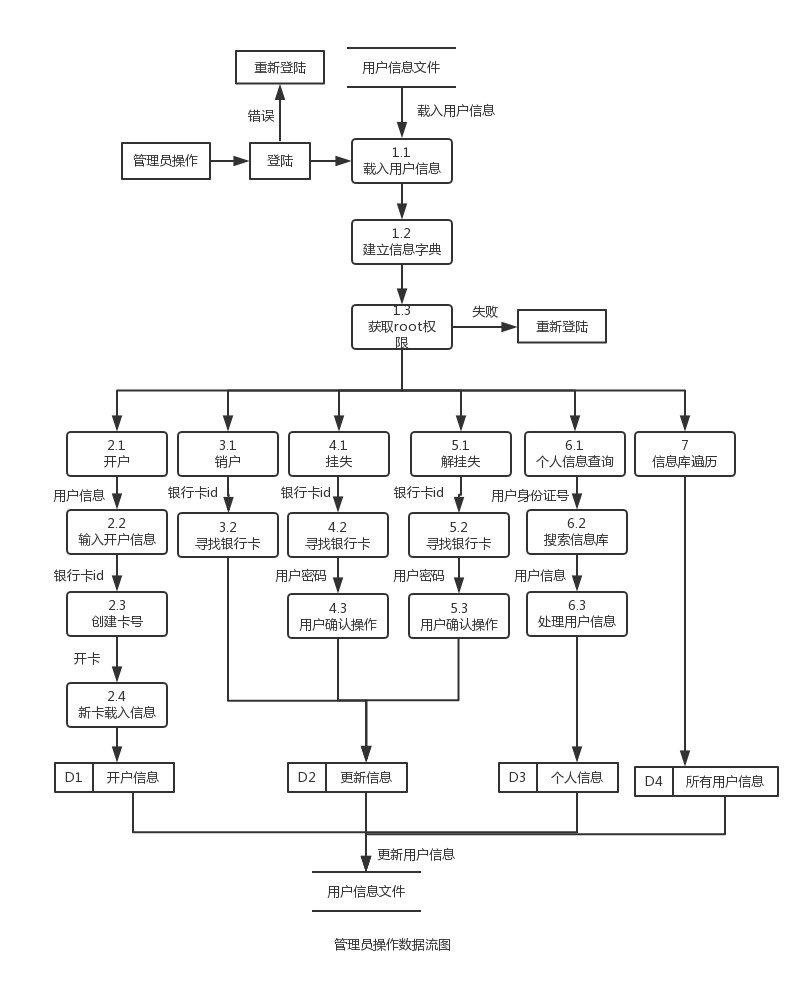


图2.5-2

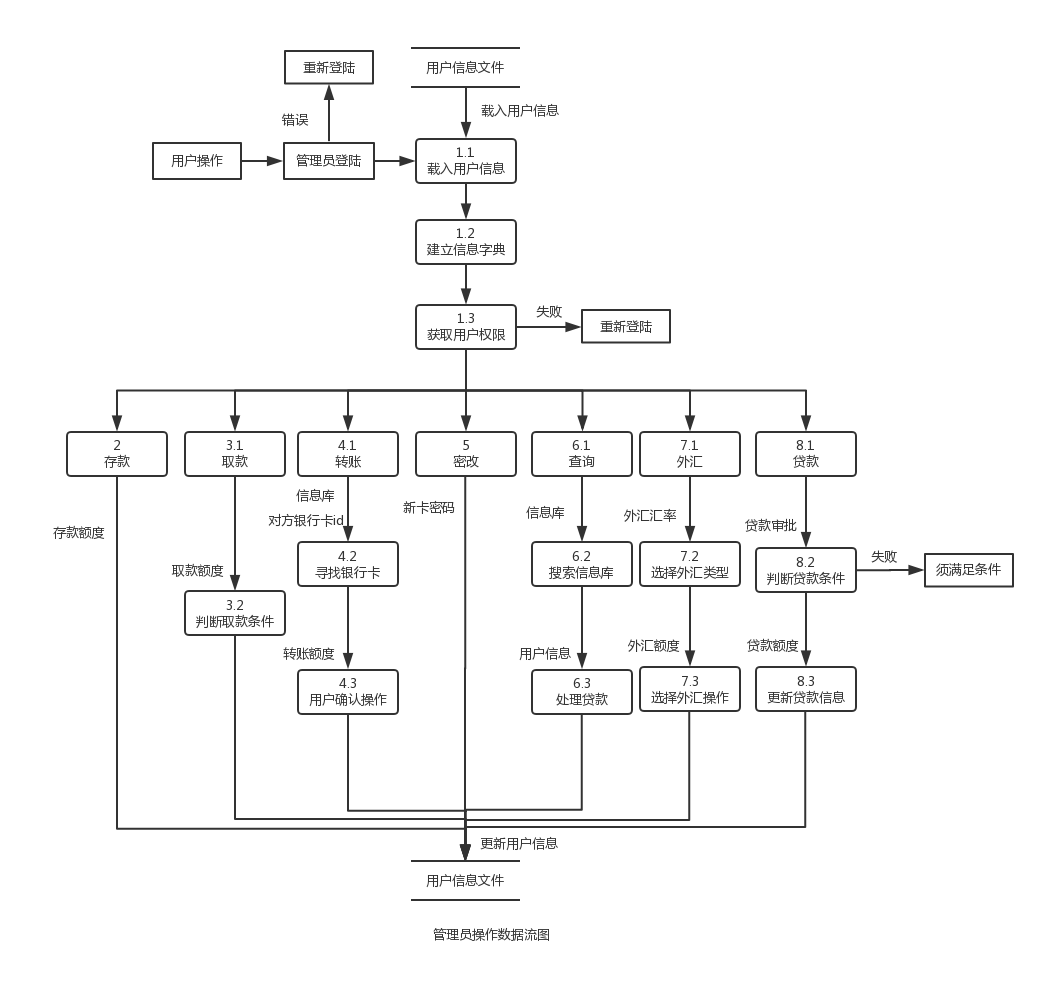


图2.5-3

## 2.6状态转化图

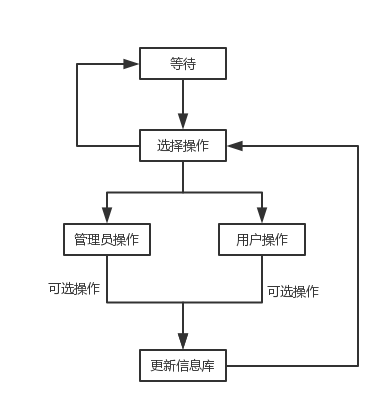


图2.6-1

# 3概要设计与详细设计

以数据为中心的思想，分析系统的数据结构设计，系统的命名规格定义等内容。

设计出系统的功能模块图或系统结构图。

结构化设计时采用过程模型工具（例如流程图、盒图或PAD图等）设计核心模块内容或算法描述。采用面向对象方法时给出核心类的核心服务的详细内容或算法描述，设计出类之间的关联，指明继承关系。

采用数据库存储数据时给数据库表的设计及表间关联，采用文件存储数据时给出文件的设计结构。

给出系统核心的数据结构。

## 3.1系统数据结构设计

class Person():

def \_\_init\_\_(self,name,credit\_card,iphone,card):

self.name=name #名字

self.credit\_card=credit\_card #身份证号

self.iphone=iphone #联系方式

self.card=card #对应银行卡

class Card():

def \_\_init\_\_(self,card\_id,password,money):

self.card\_id=card\_id #卡号

self.password=password #密码

self.money=money #余额

self.loan=0 #贷款

self.islock=False #判断标志，是否已被挂失

系统的数据主要为字符串数据和整型数据，而且对于不同数据在同一数据信息结构中存在对应的关系，如：每一个银行卡编号唯一对应一位用户及其相关信息，身份证号也唯一对应用户的姓名等。所以，可以用键值对来保存数据，而python自有的数据结构-字典，字典的键可以是数字，字符串或者元组，而键可以是字符串，数字，元组，列表或者字典，而唯一的键对应唯一的值，所以利用字典和类建立用户-银行卡基本信息数据结构。

用户-银行卡基本信息数据结构定义person类和card类，以唯一的银行卡号为键，对应的值为person类，在person类中，除了对person的描述外还有对应的card类信息。在person类中，描述为用户姓名，联系方式，身份证号，其中身份证号具有唯一标识性，利用这一特点可以实现某些需要唯一标识的系统功能。在card类中，描述为卡号，密码，余额，贷款，挂失标志，卡号为唯一性表示，由系统在开卡时随机生成。

## 3.2系统功能模块图

本部分全部以图描述系统大致操作过程，主要内容为系统操作流程图和系统功能划分图。

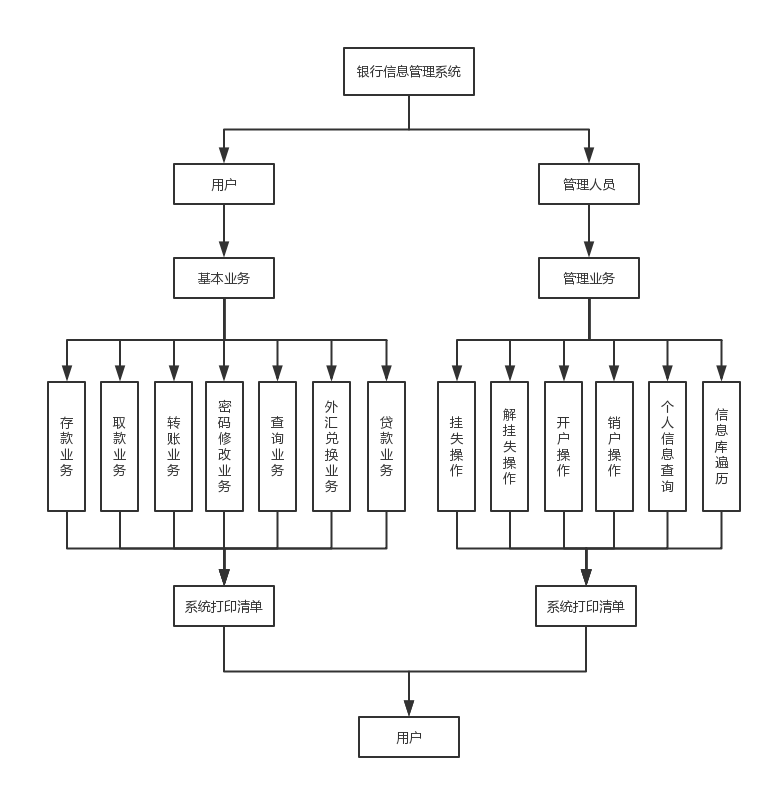


图3.2-1系统功能划分图

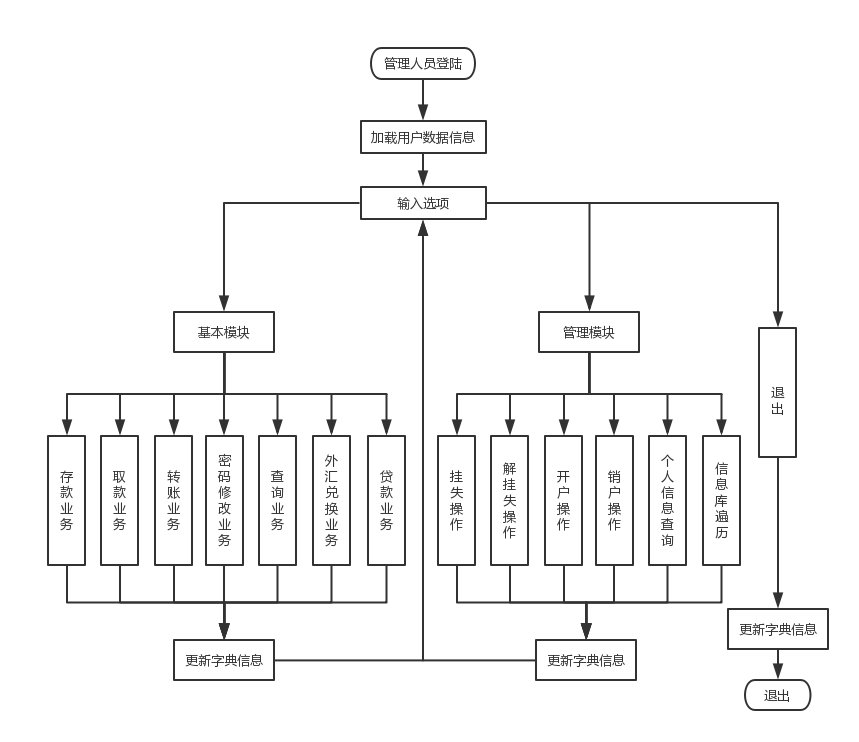


图3.2-2系统操作流程图

## 3.3核心模块流程图

系统主要功能已经在图3.2-1中给出，在3.3中，给出核心模块外汇兑换流程图，开户操作流程图，如下。

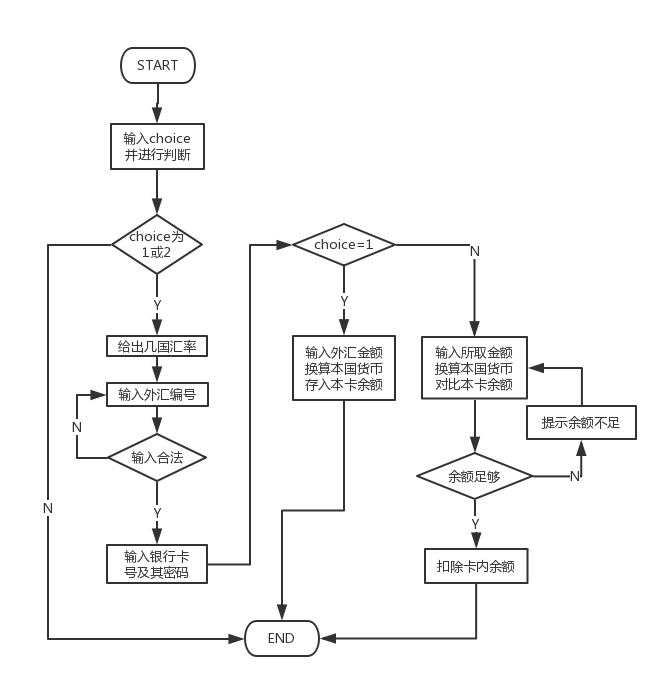


图3.3-1外汇兑换流程图

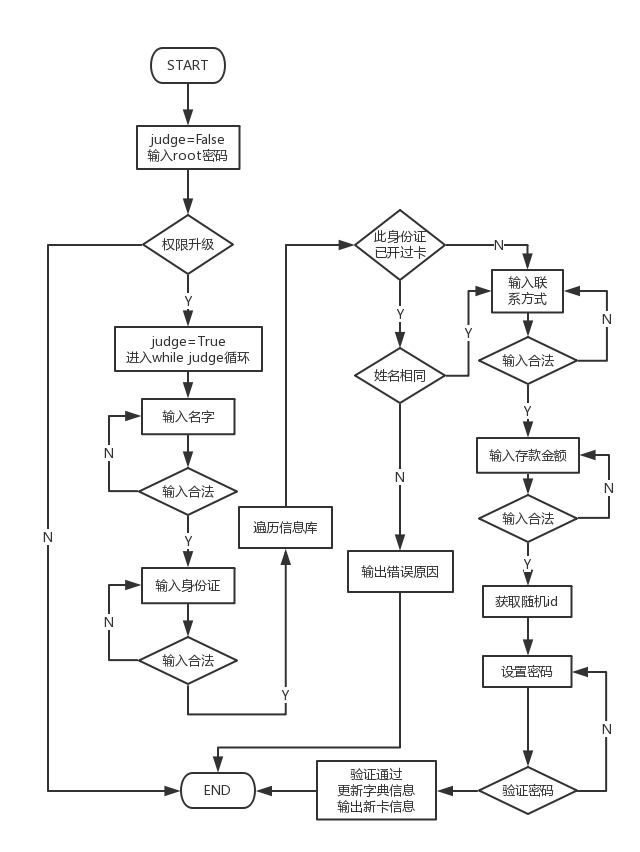


图3.3-2开户操作流程图

# 4测试报告

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，是以用户的角度，从输入数据与输出数据的对应关系出发进行测试的，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

由于篇幅有限，我决定从开户开始，完整演绎一张银行卡的开户，存款，取款，贷款，存取外汇，挂失，解挂失；然后进行root权限的个人信息查询查看此卡。具体实现过程在下面由图给出并给出一定的文字说明。



图4.1开户

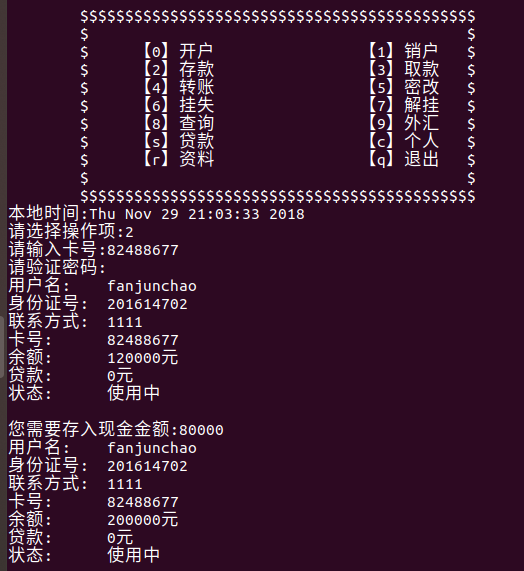


图4.2存款

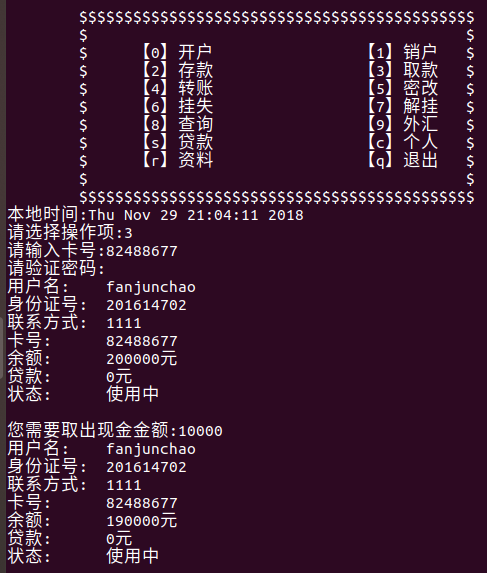


图4.3取款



图4.4挂失

在挂失此卡后，进行任何操作都会提示卡被锁无法操作，只有在输入密码解挂失后才能进行卡操作，如下图尝试进行存取外汇操作。



图4.5尝试操作

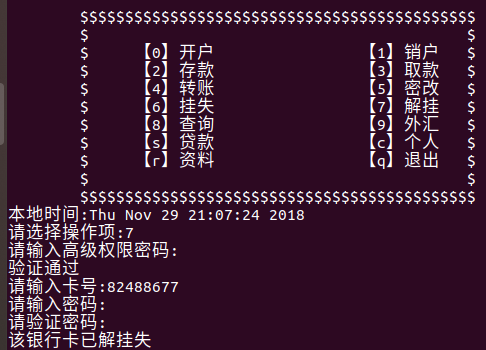
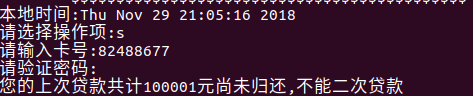


图4.6解挂失操作



图4.6存外汇操作

如上图，该银行卡已进行过一次贷款操作，接下来进行二次贷款，显示信息如下：



# 5项目管理

项目管理是管理学的一个分支学科 ，对项目管理的定义是：指在项目活动中运用专门的知识、技能、工具和方法，使项目能够在有限资源限定条件下，实现或超过设定的需求和期望的过程。项目管理是对一些成功地达成一系列目标相关的活动（譬如任务）的整体监测和管控。这包括策划、进度计划和维护组成项目的活动的进展。

本程序采用功能点技术估算软件规模及工作量，根据系统的设计：输入的项数为25，其中简单级为14项，平均级为8项，复杂级为3项；输出项数为32，其中简单级为20，平均级为11，复杂级为1项；查询项数为40项，其中简单级为16项，平均级为12项，复杂级为12项；主文件项数为1项，其中简单级为0项，平均级为0项，复杂级为1项；外部接口数为5项简单级为3项，平均级为2项，复杂级为0项。根据功能点UFP计算公式，计算出本程序的UFP=446。

根据对系统技术因素的分析，数据通信F1=2，分布式数据处理F2=0，性能标准F3=3，高负荷的硬件F4=0，高处理率F5=1，联机数据输入F6=0，终端用户效率F7=4，联机更新F8=0，复杂的计算F9=0，可重用性F10=0，安装方便F11=2，操作方便F12=3，可移植性F13=5，可维护性F14=5。根据技术复杂因子TCF的计算公式估算系统的TCF为0.9。

所以系统的功能点数FP=446\*0.9=401.4。

对于系统工作量的估算，运用静态单变量模型中的Kemerer模型计算，综合上述FP数据，

根据获得了的项目历史生产率数据，其中平均生产率为80FP/PM，平均生产工资为60元/ FP，平均工资成本为2000元/人。

所以工作量的估算为E=41.56PM，6人完成，工期为7个月，总成本的估算为83120元

项目人员分配为民主分权式。软件资源小粒度复用，开发环境为Python3，linux，联机环境。硬件资源为计算机1台。

项目任务有：功能定义4天，课设准备7天，功能定义3天，原型开发以模块开发为主系统分为3个模块，代码测试7天，文档写作25天等。系统gnatt图见图5-1

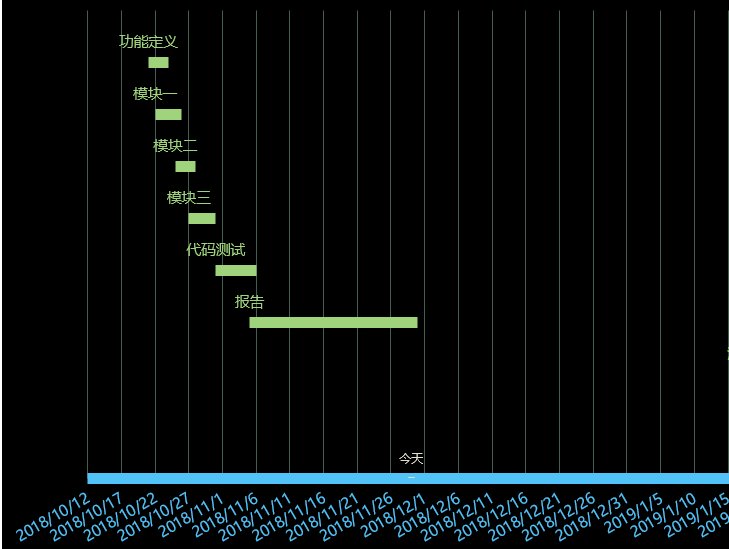


图5.1

# 6体会

一开始选题目时，我并没有想好要做什么，最后想起第一次作业留的有关银行的作业，感觉对银行的基本职能还是挺熟悉的，就想着做银行信息管理系统吧，也决定用python语言写。

从开始写代码，到测试代码，到找bug，修复bug；这么多天过去，果然还是报告最难写。尤其在需求设计阶段花费的时间较多。但通过写报告，我也了解到一个软件是如何创建的。设计软件不是一个简单的事情，它需要多方面人员的配合，从多角度，广领域进行分析，参考社会上已有的模板，配合全面的专业知识，这些都是设计软件不可或缺的。

这次软工课设最遗憾的恐怕就是我没有运用到数据库技术，之前并没有学过，错过了这次测试能力的机会。