# UML课程实践报告

**学号：2016118083**

**姓名：黄茂杰**

## 0. 系统描述

系统名称：ATM机业务

主要功能：

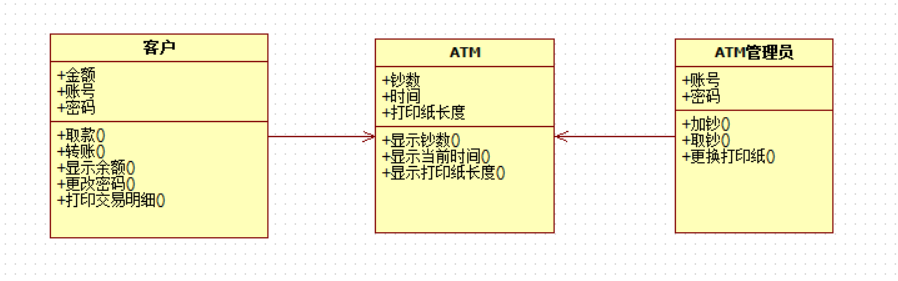
1. 取款
2. 转账
3. 查询余额
4. 更改密码
5. 打印交易明细
6. 加钞
7. 取钞
8. 显示钞数

预期用户：ATM客户和ATM管理员

1. 业务建模

### 1.1 领域模型

领域类图：描述领域概念



组织结构图（可选）：描述系统用户角色

术语表：定义领域内重要业务概念

（1）客户：利用ATM进行存款、转账等操作的使用人员

（2）ATM：又称[自动柜员机](https://baike.baidu.com/item/%E8%87%AA%E5%8A%A8%E6%9F%9C%E5%91%98%E6%9C%BA/3432290)，是指银行在不同地点设置一种小型机器，利用一张信用卡大小的胶卡上的磁带记录客户的基本户口资料（通常就是[银行卡](https://baike.baidu.com/item/%E9%93%B6%E8%A1%8C%E5%8D%A1/2321159)），让客户可以透过机器进行提款、[存款](https://baike.baidu.com/item/%E5%AD%98%E6%AC%BE/943968)、转账等银行柜台服务，大多数客户都把这种自助机器称为自动提款机。

（3）ATM管理员：对ATM机进行管理的管理人员。

（4）取款：从ATM中获得部分现金，并且此时银行卡内的余额相应减少的操作。

（5）转账：将银行卡内的余额转给其他账号的操作。

（6）显示余额：显示银行卡目前余额的操作。

（7）更改密码：将目前密码更改为其他密码的操作。

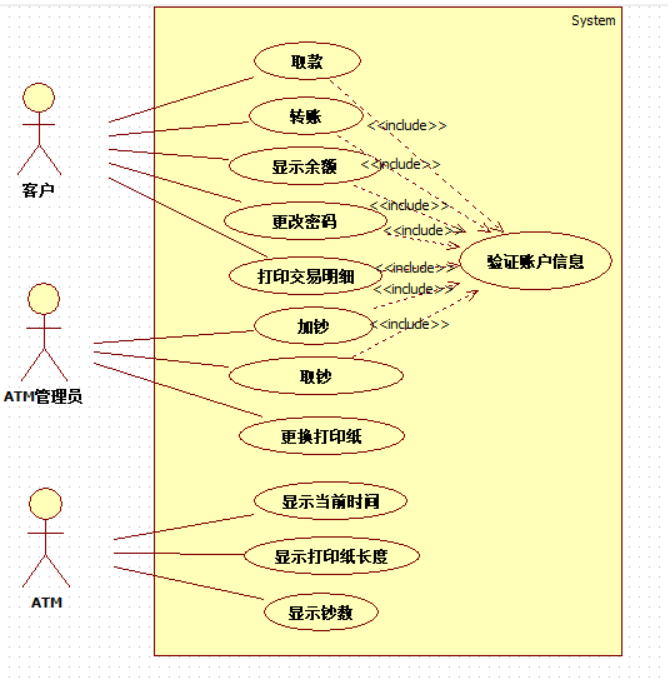
（8）打印交易明细：将本次进行操作的内容、时间打印出来。

（9）显示钞数：显示ATM内部当前钞数的操作

（10）加钞：打开ATM，向ATM机内部增加钞票的操作。

（11）取钞：打开ATM，将ATM内部钞数取出的操作。

## 1.2 用例模型



|  |  |  |
| --- | --- | --- |
| **用例编号** | UC1 | |
| **用例名称** | 验证账户信息 | |
| **用例概述** | 验证账户身份信息 | |
| **前置条件** | ATM机空闲 | |
| **后置条件** | ATM机显示“取款成功” | |
| **基本事件流** | 步骤 | 活动 |
|  | 1 | 插卡 |
|  | 2 | 客户或ATM管理员输入账号信息 |
|  | 3 | 验证密码是否匹配 |
| **备选事件流** | 3a | 密码错误，提示客户重新输入密码 |

|  |  |  |
| --- | --- | --- |
| **用例编号** | UC2 | |
| **用例名称** | 取款 | |
| **用例概述** | 客户使用ATM取钱 | |
| **前置条件** | ATM验证账户信息成功 | |
| **后置条件** | ATM机显示“取款成功” | |
| **基本事件流** | 步骤 | 活动 |
|  | 1 | 选择取款 |
|  | 2 | 客户输入或选择取款金额 |
|  | 3 | 检查客户取款金额是否合法（余额是否足够） |
|  | 4 | 客户在规定时间内从出钞口取走钞票 |
|  | 5 | 客户选择退卡 |
|  | 6 | 客户取卡 |
| **备选事件流** | 3a | 客户余额不足，提示客户重新输入金额 |
|  | 4a | 1分钟内未取走钞票，ATM进行吞钞 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **用例编号** | UC3 | |
| **用例名称** | 转账 | |
| **用例概述** | 客户使用ATM进行转账 | |
| **前置条件** | ATM验证账户信息成功 | |
| **后置条件** | ATM机显示“转账成功” | |
| **基本事件流** | 步骤 | 活动 |
|  | 1 | 客户选择转账 |
|  | 2 | 客户输入转账账号 |
|  | 3 | 检查账号信息 |
|  | 4 | 客户选择或输入转账金额 |
|  | 5 | 转账金额小于账户余额，进行转账 |
|  | 6 | 客户选择退卡 |
|  | 7 | 客户取卡 |
| **备选事件流** | 3a | 被转账账号不存在或账号冻结或挂失，提示客户重新输入转账账号 |
|  | 4a | 转账金额大于账户余额，提示客户重新输入转账金额 |

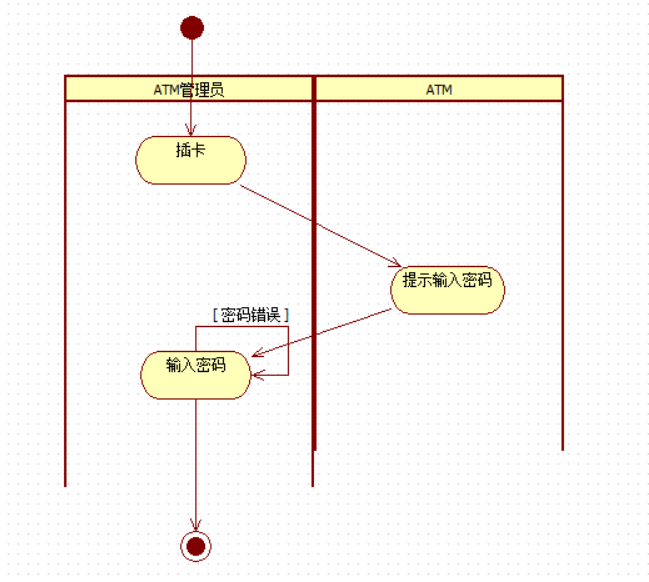
|  |  |  |
| --- | --- | --- |
| **用例编号** | UC4 | |
| **用例名称** | 加钞 | |
| **用例概述** | ATM机将钞票放入ATM内 | |
| **前置条件** | ATM验证账户信息成功 | |
| **后置条件** | ATM机显示“加钞成功” | |
| **基本事件流** | 步骤 | 活动 |
|  | 1 | ATM管理员用钥匙打开ATM机 |
|  | 2 | ATM管理员将钞票放入ATM机内 |
|  | 3 | ATM机更新钞数 |

|  |  |  |
| --- | --- | --- |
| **用例编号** | UC5 | |
| **用例名称** | 显示钞数 | |
| **用例概述** | ATM机显示当前机内所拥有的钞票数量 | |
| **前置条件** | ATM验证账户信息成功 | |
| **后置条件** | ATM机进入空闲状态 | |
| **基本事件流** | 步骤 | 活动 |
|  | 1 | ATM管理员选择显示钞数 |
|  | 2 | ATM显示钞数 |

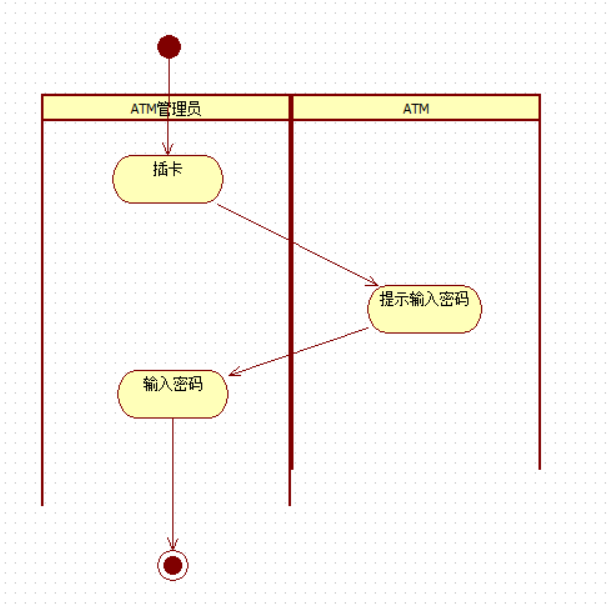
## 1.3 业务流程模型

1. 验证账户信息

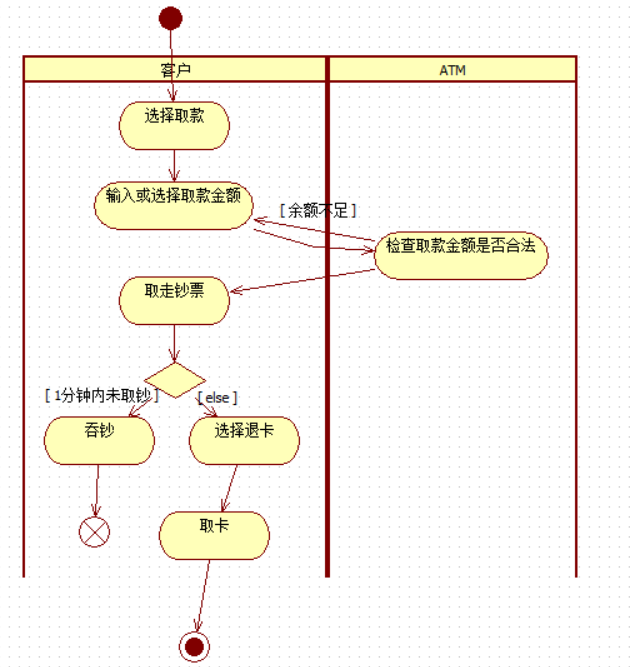
（1）客户验证账户信息



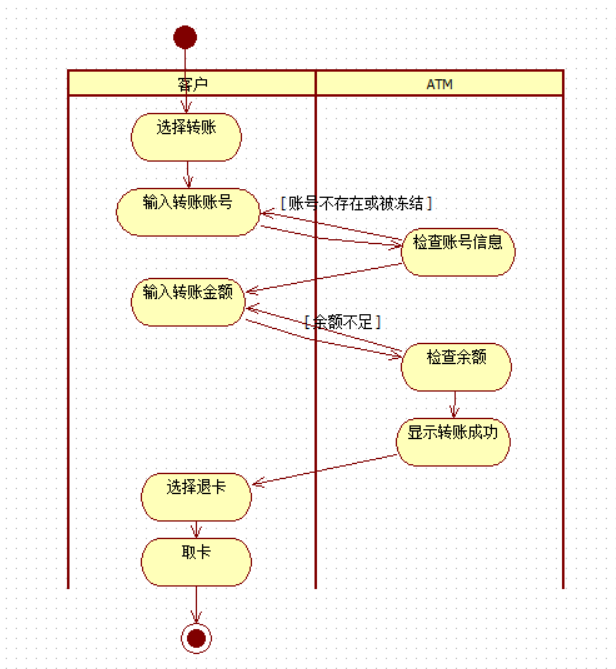
（2）ATM管理员验证账户信息



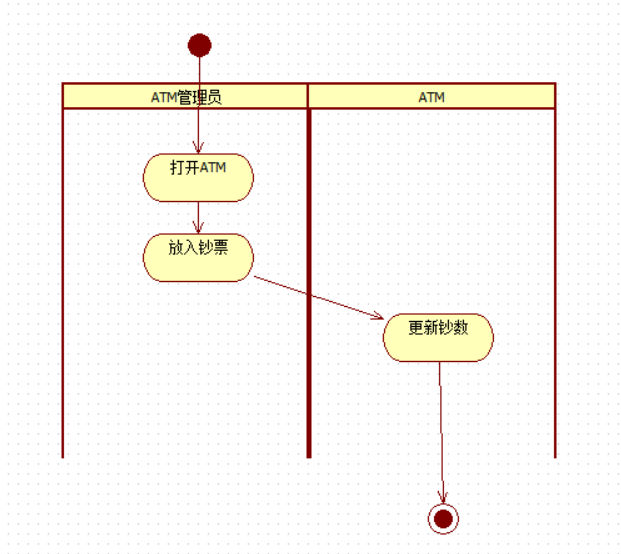
1. 取款



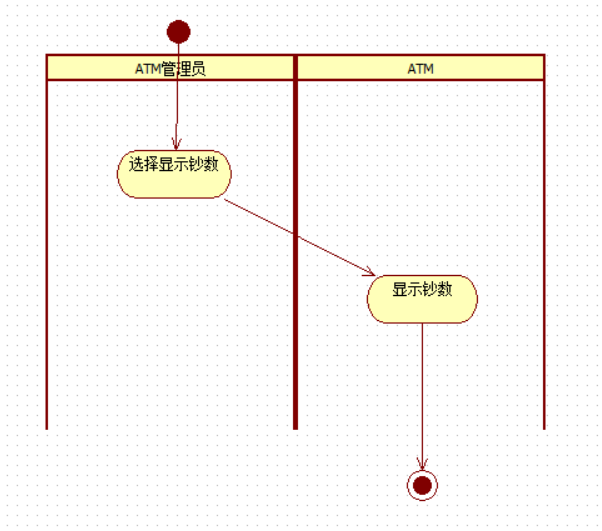
1. 转账



1. 加钞



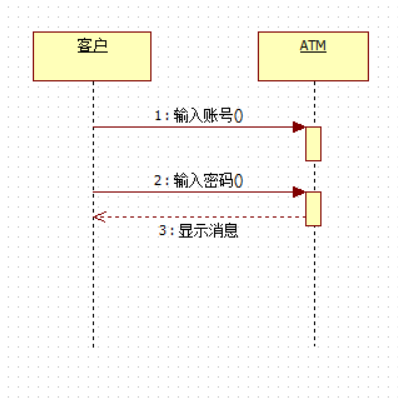
1. 显示钞数



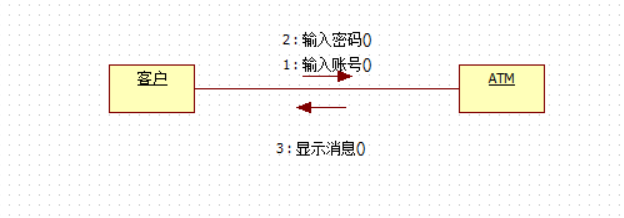
## 2. 分析

### 2.1 用例实现

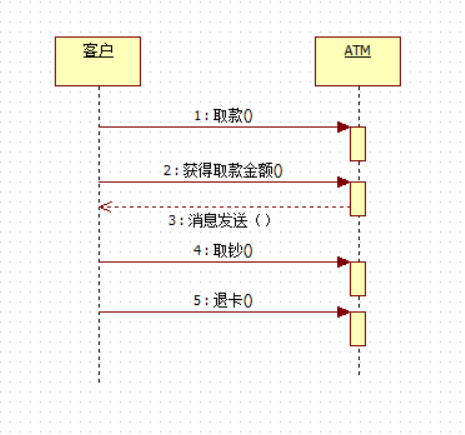
1. 验证账户信息（客户）
2. 序列图



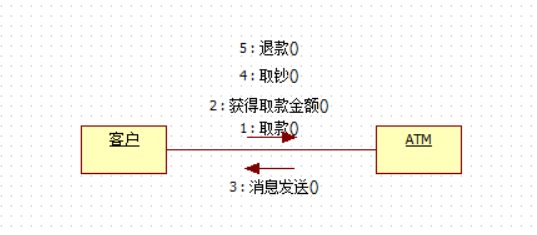
(2)协作图



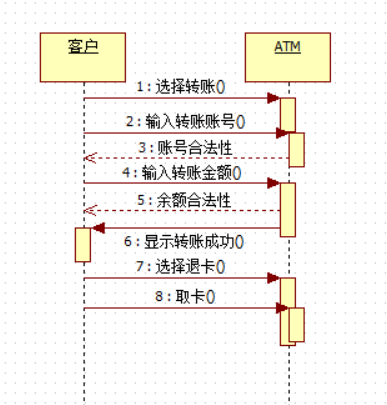
1. 取款
2. 序列图



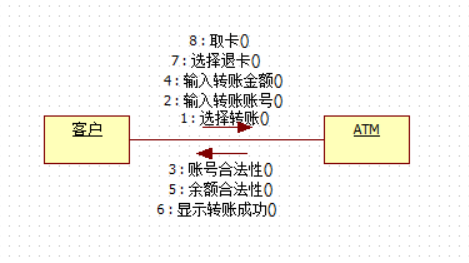
1. 协作图



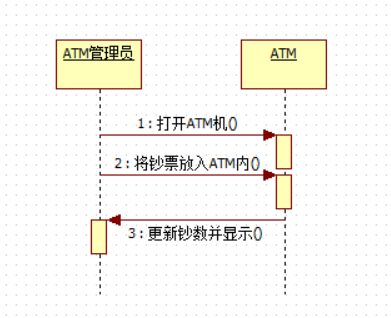
1. 转账
2. 序列图



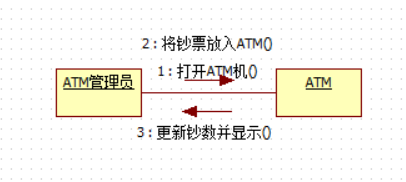
1. 协作图



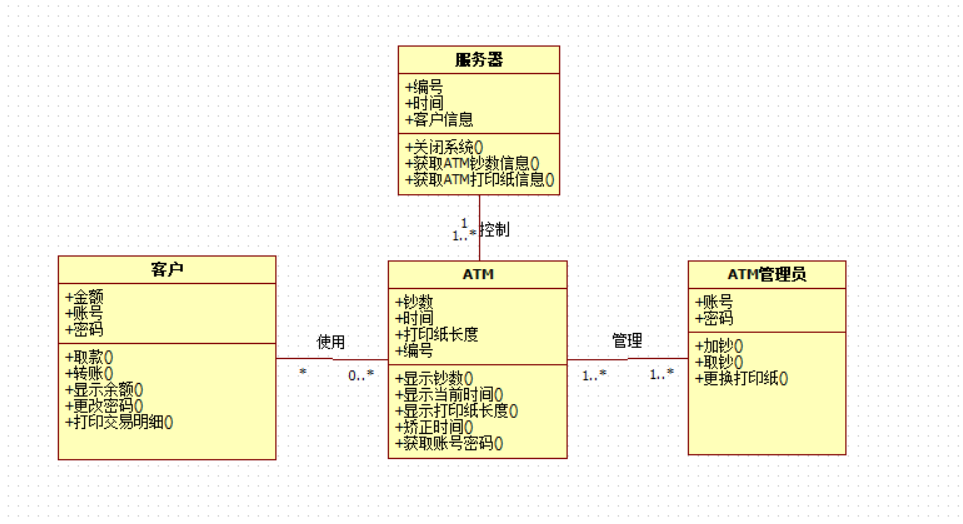
1. 加钞
2. 序列图



1. 协作图



### 2.2 分析类图



### 系统体系结构描述

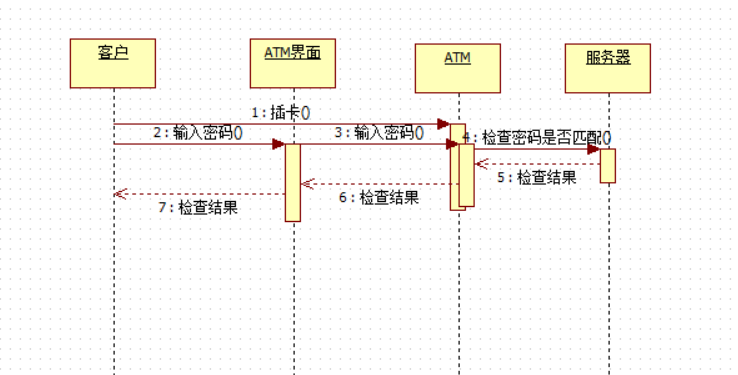
该系统为C/S（Client/Server）结构，即客户机和服务器结构。这是以[数据库服务器](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E6%9C%8D%E5%8A%A1%E5%99%A8)为中心、以客户机（ATM）为网络基础、在信息系统软件支持下的两层结构模型。这种体系结构中，用户操作模块布置在客户机（ATM）上，[数据存储](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AD%98%E5%82%A8)在服务器上的数据库中。客户机（ATM）依靠服务器获得所需要的网络资源，而服务器为客户机提供网络必须的资源。目前大多数信息系统是采用Client/Server结构。

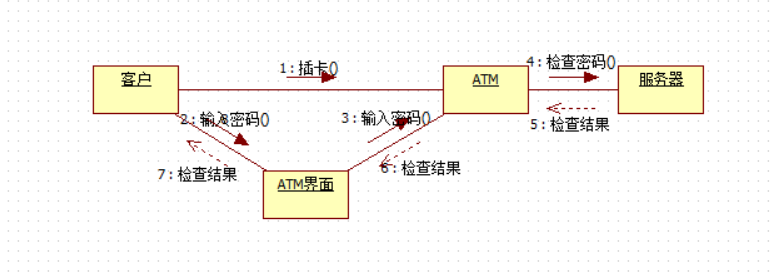
## 3. 设计

### 3.1 对象设计

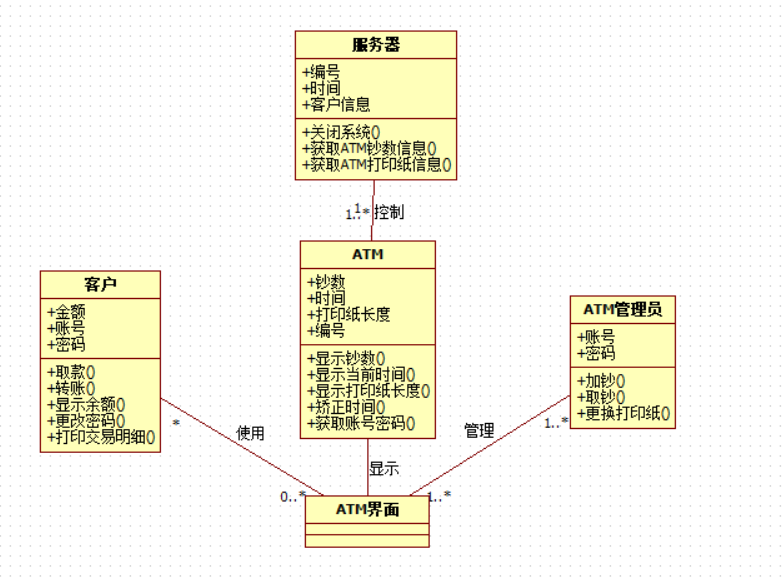
1.交互图：

（1）验证账户信息（客户）

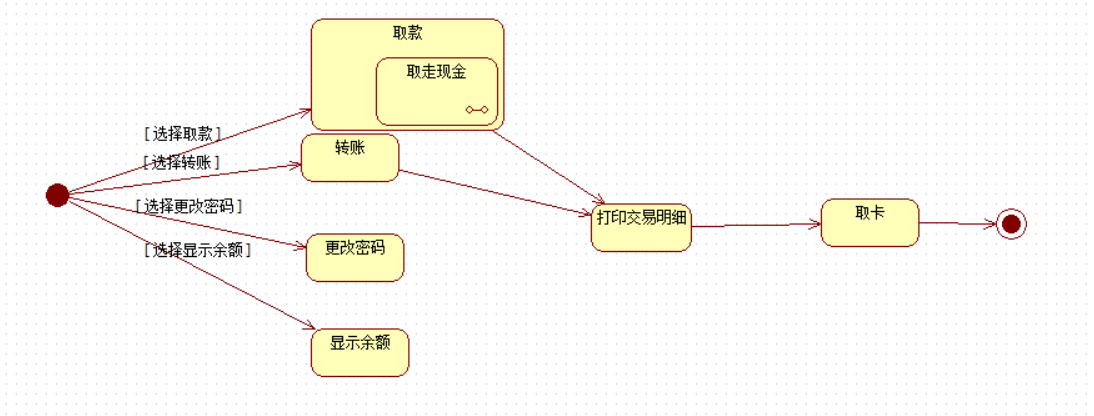




2.设计类图：



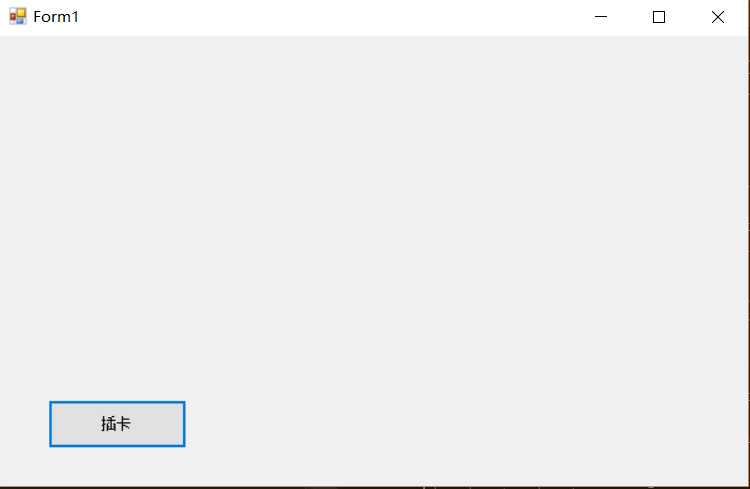
3.状态图：

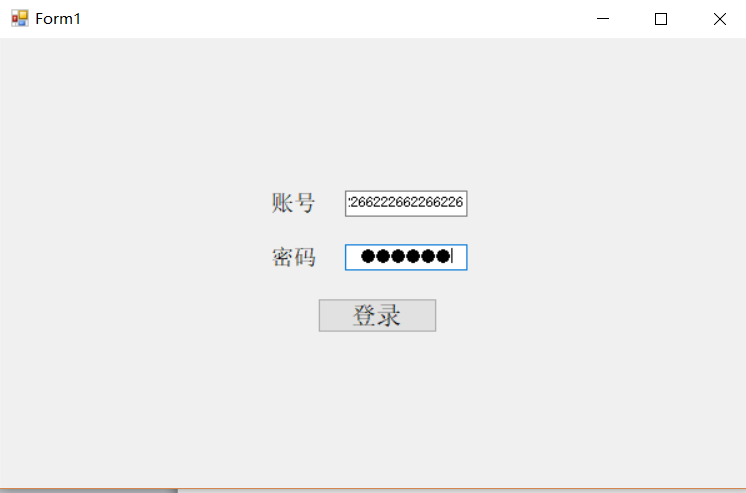


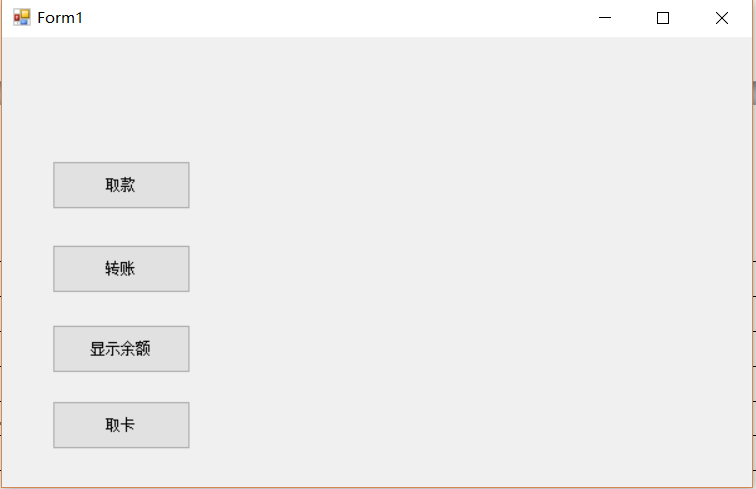
### 3.2 用户界面设计

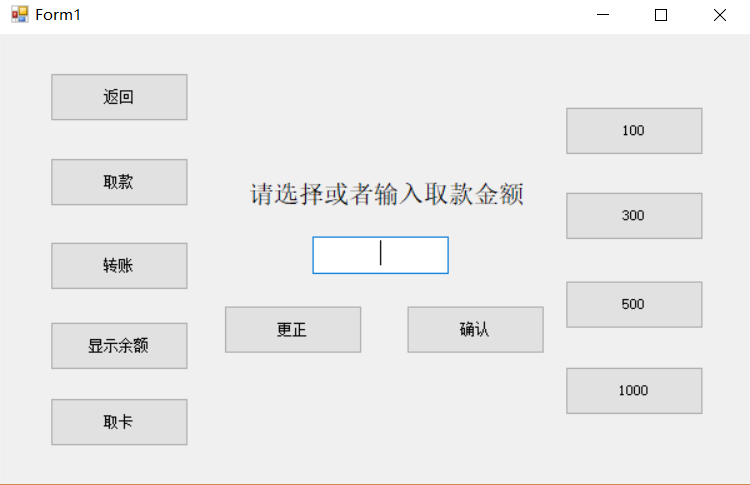
实现语言：C#

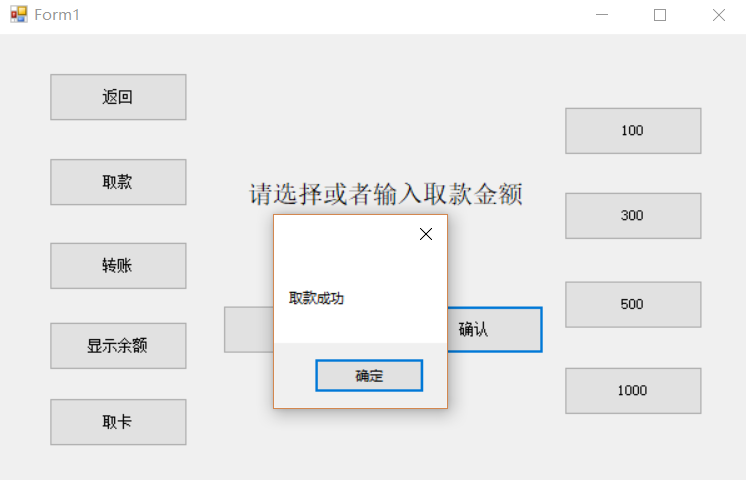
实现平台：Visual Studio 2017

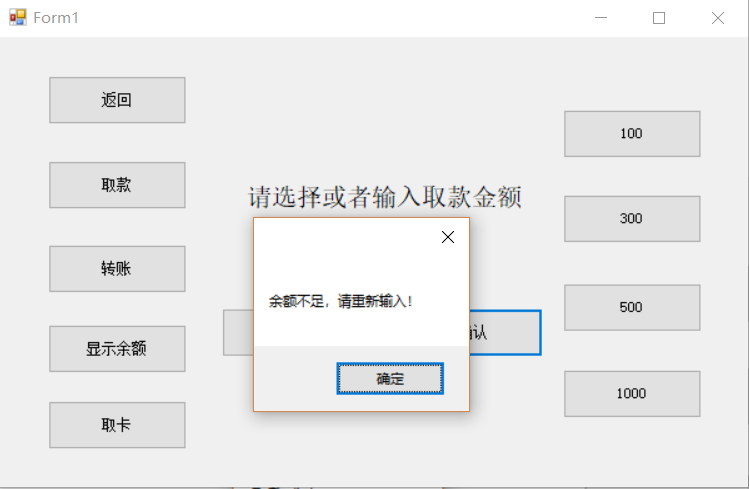


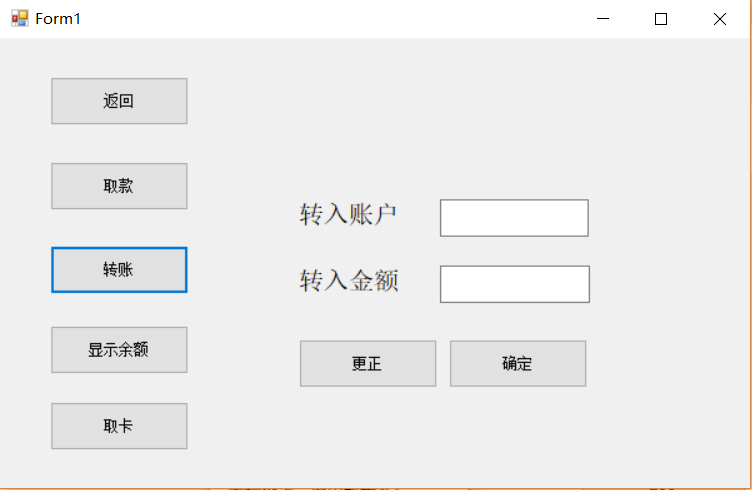














### 3.3 数据库设计

SQL Server数据库：BankAccount

单一表：AccountInformation

表结构

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 允许NULL值 |
| 账号 | nvarchar(50) |  |
| 密码 | nvarchar(50) |  |
| 姓名 | nchar(10) | 是 |
| 身份证号 | nvarchar(50) | 是 |
| 金额 | Money |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 账号 | 密码 | 姓名 | 身份证号 | 金额 |
| 001 | 123456 | 张三 | NULL | 10000.0000 |
| 002 | 123456 | 李四 | NULL | 1000.0000 |

## 4. 实现

1.源代码清单：

namespace ATM\_simulator

{

public partial class Form1 : Form

{

Form1(){}

resetButton\_Click(){}

insertButton\_Click(){}

withdrawButton\_Click(){}

removeButton\_Click(){}

checkButton\_Click(){}

LoginButton\_Click(){}

returnButton\_Click(){}

transferButton\_Click(){}

oneHundredButton\_Click(){}

threeHundredButton\_Click(){}

fiveHundredButton\_Click(){}

thousandButton\_Click(){}

displayButton\_Click(){}

transferResetButton\_Click(){}

transferCheckButton\_Click(){}

resetPassword(){}

}

public class DatabaseOp（）

{

DatabaseOp(){}

OpenConn(){}

CloseConn(){}

GetDataReaderValue(){}

GetDataSetValue(){}

GetDataViewValue(){}

GetDataTableValue(){}

ExecuteNonQuery(){}

ExecuteNonQueryCount(){}

ExecuteScalar(){}

SqlServerRecordCount(){}

IsNumber(){}

GetSafeValue(){}

}

}

2.测试数据和结果

|  |  |
| --- | --- |
| 测试数据 | 测试结果 |
| 对账号001取款100 | 取款成功，账号001余额变为9900 |
| 账号001向账号002转账100 | 转账成功，账号001余额变为9900，账号002余额变为1100 |

3.对系统的评价和分析

（1）系统耦合度仍然较高，需要进一步优化，提高内聚度

（2）数据库是本地数据库，没有实现云数据库