



長春工業大學

C#程序設計 實訓報告

項目組：_____第二項目組

專 業：_____軟件技術

學 院：_____應用技術學院

指導教師：_____聞喆

實訓時間：_____2017年7月4日-2017年7月22日

應用技術學院

模拟 ATM 自动柜员机管理系统

第一部分 团队构成及分工

第二部分 系统开发进度计划

第三部分 项目需求分析

第四部分 项目概要设计

第五部分 项目详细设计

第六部分 项目实施

第七部分 项目测试

第八部分 代码清单

第九部分 参考文献

第十部分 实训总结

目 录

1. 团队构成及分工	1
2. 系统开发进度计划.....	2
3. 项目需求分析	3
3.1 导言	3
3.1.1 目的	3
3.1.2 定义及缩写.....	3
3.2 系统定义	3
3.2.1 项目来源及背景.....	3
3.2.2 用户特点.....	4
3.2.3 项目目标.....	4
3.2.4 应用环境.....	4
3.3 功能需求	5
3.3.1 功能描述.....	5
3.3.2 用例图.....	5
3.4 数据描述	7
3.5 性能需求	7
3.5.1 界面需求.....	7
3.5.2 响应时间需求.....	8
3.5.3 可靠性需求.....	8
3.5.4 开发性需求.....	8
3.5.5 可扩展性需求.....	8
3.5.6 系统安全性需求.....	8
4. 项目概要设计	8
4.1 导言	8
4.1.1 目的	8
4.1.2 定义及缩写.....	9
4.2 功能设计	9
4.2.1 功能模块设计.....	9
4.2.2 模块详述.....	9
4.2.3 模拟 ATM 自动柜员机管理系统模块详述.....	10
4.3 数据设计	11
4.3.1 数据概念结构设计.....	11
4.3.2 数据逻辑结构设计.....	14
5 项目详细设计	15
5.1 导言	15
5.1.1 目的	15
5.1.2 定义及缩写.....	16
5.2 界面设计	16
5.2.1 模拟 ATM 自动柜员机管理系统登录界面设计	16
5.2.2 账户开户注册界面设计	16
5.2.3 用户操作界面设计.....	17
5.3 功能详细设计.....	18

5.3.1 模拟 ATM 自动柜员机管理系统功能活动图.....	18
5.3.2 功能活动图.....	19
6. 项目实施	19
6.1 导言	19
6.1.1 目的	19
6.1.2 缩写及定义.....	19
6.2 编码格式规范.....	20
6.2.1 缩进规则.....	20
6.2.2 断行规则.....	20
6.3 命名规范	20
6.3.1 类/接口	20
6.3.2 控件	20
6.3.3 方法	20
6.4 功能实现	21
6.4.1 模拟 ATM 自动柜员机管理系统窗体功能实现.....	21
6.4.2 模拟 ATM 自动柜员机管理系统窗体功能实现.....	22
7. 测试	26
7.1 登录功能测试.....	26
7.2 注册功能测试.....	27
7.3 模拟 ATM 自动柜员机管理系统功能测试.....	28
8. 代码清单	29
8.1 模拟 ATM 自动柜员机管理系统模块	29
8.1.1 模拟 ATM 自动柜员机管理系统功能	29
9. 参考文献	39
10. 实训总结	40

1. 团队构成及分工

项目组：				
职能	成员	项目角色	任务分工	成绩
组长	马进	项目经理	设计和编码	
组员	纪骁宵	界面设计人员	界面设计	
	姜佳新	数据库设计人员	数据库设计	
	李冰	需求分析人员	需求分析	
	刘帅 ₂₀₁₆₄₄₅₄	测试人员	软件调试及测试	

指导教师签字：

年 月 日

2. 系统开发进度计划

序号	任务名称	开始时间	完成时间	持续时间
1	确定题目	7. 4	7. 5	1
2	寻找资料	7. 5	7. 6	1
3	需求分析	7. 7	7. 14	2
4	概要设计	7. 10	7. 12	2
5	详细设计	7. 13	7. 14	1
6	实现	7. 3	7. 3	1
7	测试	7. 11	7. 14	3
8	文档编写	7. 10	7. 14	4

3. 项目需求分析

3.1 引言

3.1.1 目的

相比国外，国内的 ATM 机起步较晚，我国的 ATM 是伴随着银行卡业务的成长而逐步发展起来的，为了满足持卡人在银行营业时间外也能够用卡存，取款，银行投资并安装了 ATM，以此提供 24 小时的自助服务。ATM 是社会发展过程中电子技术与银行业务完美结合的产物之一，信息技术的创新推动者 ATM 向更加智能化和人性化的方向发展。现在的 ATM 机使用方便，界面更加友好，为广大客户提供了一种便捷的资助服务方式，不但增强了银行业务的灵活性，而且提升了银行的整体思想。对用户来说，自动柜员机（ATM）的诞生，使原来必须到银行柜台办理的账务查询、转账、充值等业务，可以在大街小巷的固定取款机前进行办理。

本文档的预期读者是：

1. 开发人员：马进
2. 项目管理人员：纪骁宵，姜佳新，李冰
3. 测试人员：刘帅20164454

3.1.2 定义及缩写

账号，在银行中，事物应用的单个账号。每个顾客可以拥有多个账号。用户，拥有银行的一个或多个账号的人。可以是一个人或多个人，或者是公司。相同的人，拥有不同的银行账号被认为是不同的落户。

ATM，ATM 是 Automatic Teller Machine 的缩写，意为自动取款机。是一种高度精密的机电一体化设备，利用磁卡或智能 IC 卡储存用户信息并通过加密键盘输入密码然后通过银行内部网络验证并进行各种交易的金融自助设备。

模拟 ATM 自动柜员管理系统：Simulation of ATM teller management system

3.2 系统定义

3.2.1 项目来源及背景

随着时代的进步，生活节奏的加快，越来越多的的人进入银行。在银行业务量爆炸式增长的时候，ATM 已经开始悄悄的走入人们的生活。ATM 机作为银行服务的扩展终端，大大的提高了银行业务办理能力，24 小时为客户提供基础的银行服务，使得银行客户在面对长长的队伍时更加主动的选择。ATM 设备已经成为我们日常生活中不

可短缺的重要设备。因此我决定学习了解 ATM 设备的工作原理，探究 ATM 机的工作机制，并自行模拟 ATM 系统功能。

在市场经济的蓬勃发展和人们日益繁忙的条件下，现有的银行系统往往需要客户在办理手续时等待很长的时间，这不仅会浪费很多宝贵的时间，也会使得银行的业务人员十分的繁忙，需要很大的人力和财力。基于这样的情况，ATM 取款机系统的开发就显的十分的重要！它可以减少银行的业务处理压力，尽量节省人们的时间，并且可以有效解决用户信息和资金信息的繁杂问题。

3.2.2 用户特点

主要实现模拟 ATM 自动柜员管理，为用户提供充足的了解和快捷的存取和查询手段。例如：存取方便、安全性高、到账速度快、方便可靠、保密性好、不受时间限制。这些优点能够及大地提高业务流程的效率，也是经济不断发张的重要条件。用软件模拟 ATM 自动柜员管理系统还可以通过功能强大的 Internet 网及时的读者发布最新动态

3.2.3 项目目标

本项目设计的目标如下：

1. 界面设计美观友好，信息查询灵活、方便、快捷、准确、数据存储安全可靠
2. 实现取款等功能，完成基本的流程操作。
3. 对用户操作信息进行增加、修改、查询。
4. 对输入的数据，系统进行严格的数据检验
5. 系统最大限度地实现了易维护性和易操作性
6. 系统运行稳定、安全可靠
7. 历练了我们的团结合作的能力
8. 训练综合运用所学的理论知识和方法进行独立分析和解决问题的能力；
9. 培养独立的 C#语言与算法设计方法，具备初步的独立分析和设计能力；
10. 学习软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；

3.2.4 应用环境

系统运行硬件环境：

操作系统：Windows10

系统运行软件环境：

开发语言：C#

数据库:SQL Server2016

开发工具包: Visual Studio 2017

文档编写: Adobe Acrobat Pro DC 2017

3.3 功能需求

3.3.1 功能描述

对本系统的各项功能做简要描述:

用户登录: 包括对 账号、密码的输入规范, 不正确的次数。

用户取款: 取款面额需为100的倍数, 且取款次数有限制, 期间不透支。

用户存款: 放入金额, 最多一百张。

用户转账: 需输入转账账号, 确定无误后按确定。

查询余额: 显示可用余额、账户余额。

修改密码: 需要输入原密码, 输入新密码, 并且确认新密码。

退 卡: 请及时取回您的卡。

查询记录: 查询出关于本账号的转账, 取款, 存款记录。

3.3.2 用例图

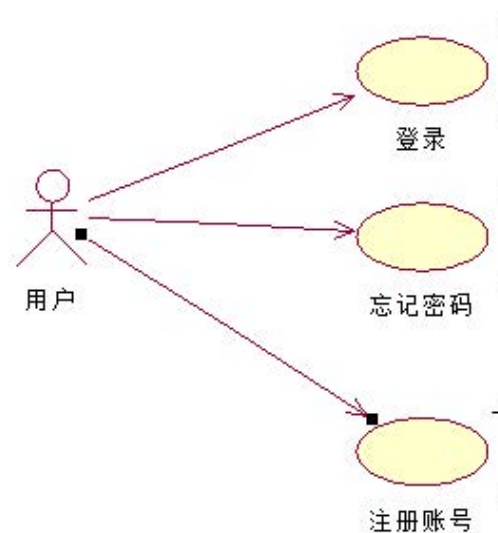


图 3-1 用户登录用例图

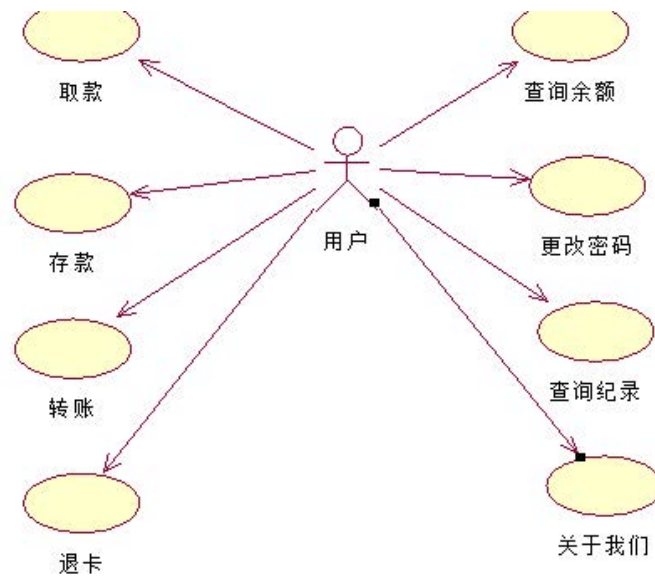


图 3-2 用户使用用例图

1.用户登录用例

名称：用户登录

参与执行者：系统用户、登录窗

前置条件：数据库已经连接

活动的基本过程：

1. 用户在登录窗体中输入银行卡账号、账户密码，选择权限
 2. 用户点击“登录”按钮
 3. 系统查询数据库中的用户表，判断此帐号是否存在，账号和密码是否吻合
 4. 如果帐号和密码吻合，系统将根据该帐号所对应的操作权限跳转到相应的窗体
- 后置条件：**用户已登录，用户可以操作相应的页面

2. 用户 ATM 使用用例

名称：ATM 使用

参与执行者：窗体

前置条件：数据库已经连接

活动的基本过程：

1. 用户进入主界面窗口
2. 点击所需要的功能，进行工作
3. 系统查询数据库，将结果显示在新窗体上

后置条件：无

3.4 数据描述

1. 用户名信息

表 3-1 用户名信息

名称	用户名
描述	是用户使用本系统的用户名信息
定义	是汉字、字母或它们的组合，长度为 8 位
位置	登录窗体、注册窗体、用户存取功能窗体

2. 密码信息

表 3-2 密码信息

名称	密码
描述	是用户登录本系统的密码信息
定义	是汉字、字母或它们的组合，且不能为空
位置	登录窗体、注册窗体、用户存取功能窗体

3.5 性能需求

3.5.1 界面需求

系统的界面要求如下：

1. 页面内容：主题突出、操作方便、术语和行文格式统一、规范、明确。菜单设置和布局合理，传递的信息准确。

2. 菜单结构摆放合理，方便用户使用。
3. 技术环境：页面大小适中，控制字体和版面布局

3.5.2 响应时间需求

当用户登录，进行任何操作的时候，系统应该及时地进行反应，反应的时间在 5 秒以内。系统应该及时地进行反应，反应的时间在 5 秒以内。系统应能检测出各种非正常情况，如与设备的通信中断，无法连接数据库服务器等，以避免出现长时间等待甚至无响应。

3.5.3 可靠性需求

系统应保证 7*24 小时不脱机，保证顾客可以随时在客户端登录。

3.5.4 开发性需求

开发基于的平台要考虑向上兼容性，如操作系统，数据库等要考虑更高版本的兼容性。

3.5.5 可扩展性需求

当需求发生变化时系统应具有一定的适应能力，要求系统能够为将来的变更提供以下支持：能够在系统变更用户界面和数据库设计，甚至在更换新的 DBMS 后，系统的现有设计和编码能够最大程度的重用，以保护现阶段的投资和保证软件系统能够在较少后续投入的情况下适应系统的扩展和更新。在设计中最好列出针对变更所需要重新设计的模块部分

3.5.6 系统安全性需求

系统有严格的权限管理功能，各功能模块需要相应的权限方能进入。系统需能够防止各类误操作可能造成的数据丢失，破坏。防止不法用户盗取重要信息。

4. 项目概要设计

4.1 引言

4.1.1 目的

该文档的目的是描述模拟 ATM 自动柜员管理系统的概要设计，其主要内容包括：

1. 系统功能设计
2. 系统界面设计

3. 数据设计

本文档预期读者：

- 1. 开发人员 ： 马进
- 2. 项目管理人员 ： 纪骁宵， 姜佳新， 李冰
- 3. 测试人员 ： 刘帅20164454

4.1.2 定义及缩写

账号，在银行中，事物应用的单个账号。每个顾客可以拥有多个账号。用户，拥有银行的一个或多个账号的人。可以是一个人或多个人，或者是公司。相同的人，拥有不同的银行账号被认为是不同的落户。

ATM，ATM 是 Automatic Teller Machine 的缩写，意为自动取款机。是一种高度精密的机电一体化设备，利用磁卡或智能 IC 卡储存用户信息并通过加密键盘输入密码然后通过银行内部网络验证并进行各种交易的金融自助设备。

模拟 ATM 自动柜员管理系统：Simulation of ATM teller management system

4.2 功能设计

4.2.1 功能模块设计

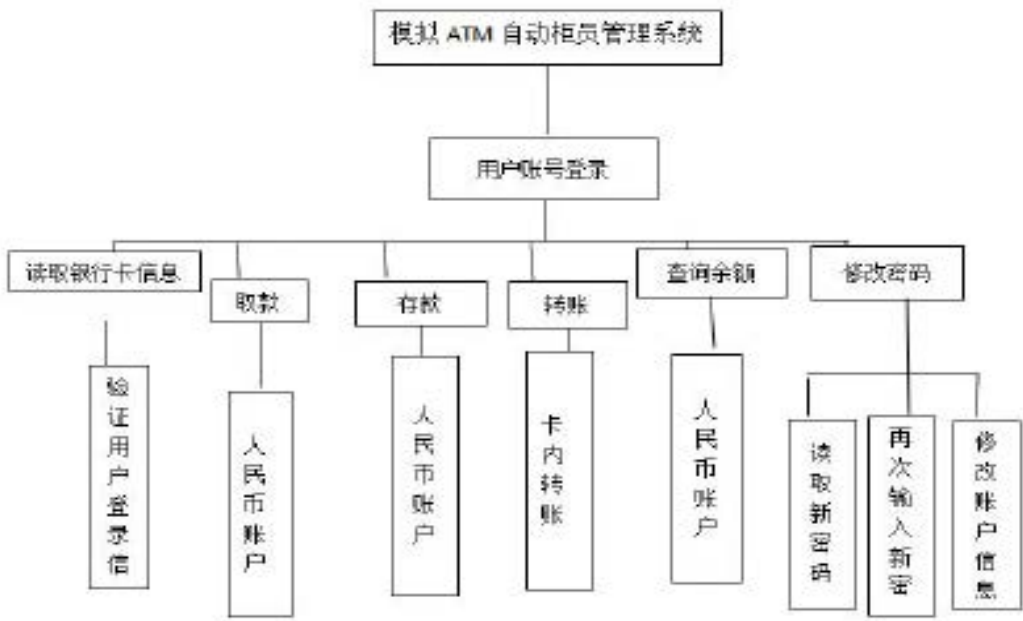


图 4-1 ATM 自动柜员管理系统功能模块图

4.2.2 模块详述

用户首先进行用户账号登录，进入输入密码界面，密码输入成功后进入主界面，主界面共有取款，存款，转账，退卡，修改密码，查询余额，帮助，关于我们，等功能

4. 2. 模拟ATM自动柜员机管理系统模块详述

取款功能：

用户选择取款，系统会提供面值的按钮，用户还可自己输入，完成后会显示自己的信息，然后确定是否打印凭条。

存款功能：

用户选择存款，会出现用户的账号，需要用户手动输入金额，存款成功后，确定是否打印凭条。

转账功能：

用户选择转账功能，输入转账的账号和所要转的金额，转账成功后，确定是否打印凭单。

修改密码功能：

用户选择修改密码，需要输入原密码，新密码，并且确认新密码。如果两次新密码不一样，则修改失败。

忘记密码功能：

用户可以凭借密保问题来重新设置密码

查询订单功能：

可以查询到关于本账号存款取款转帐的信息。

关于我们功能：

介绍本产品的相关人员。

冻结账号：

输入三次错误直接冻结账号。

解锁账号：

回答密保问题才可以。

4.3 数据设计

4.3.1 数据概念结构设计

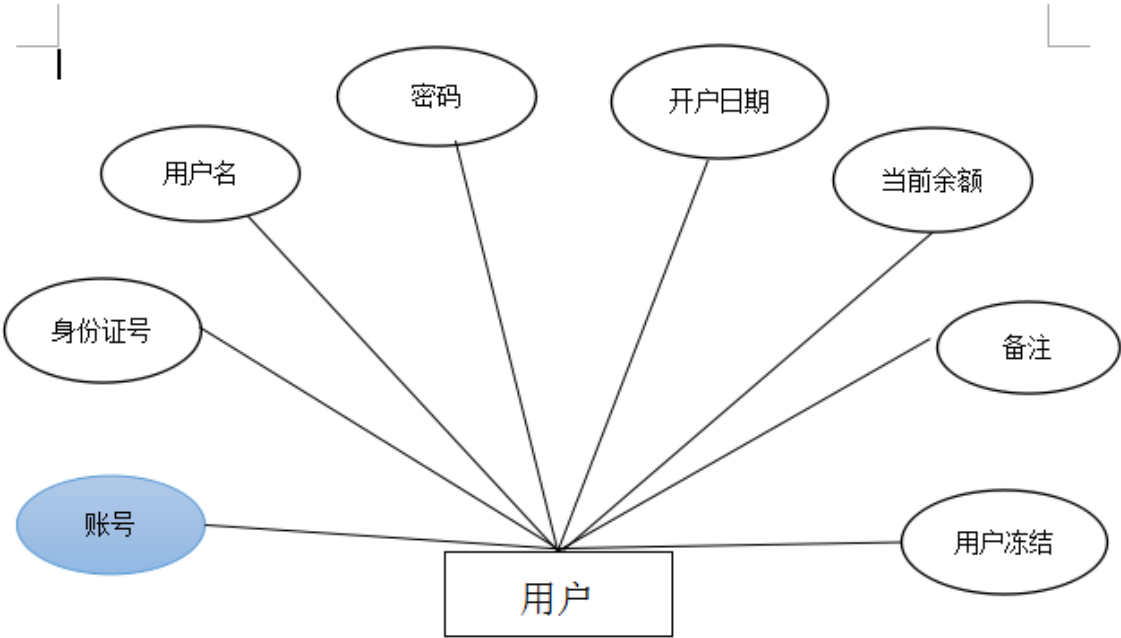


图 4-1 用户信息实体与属性关系图

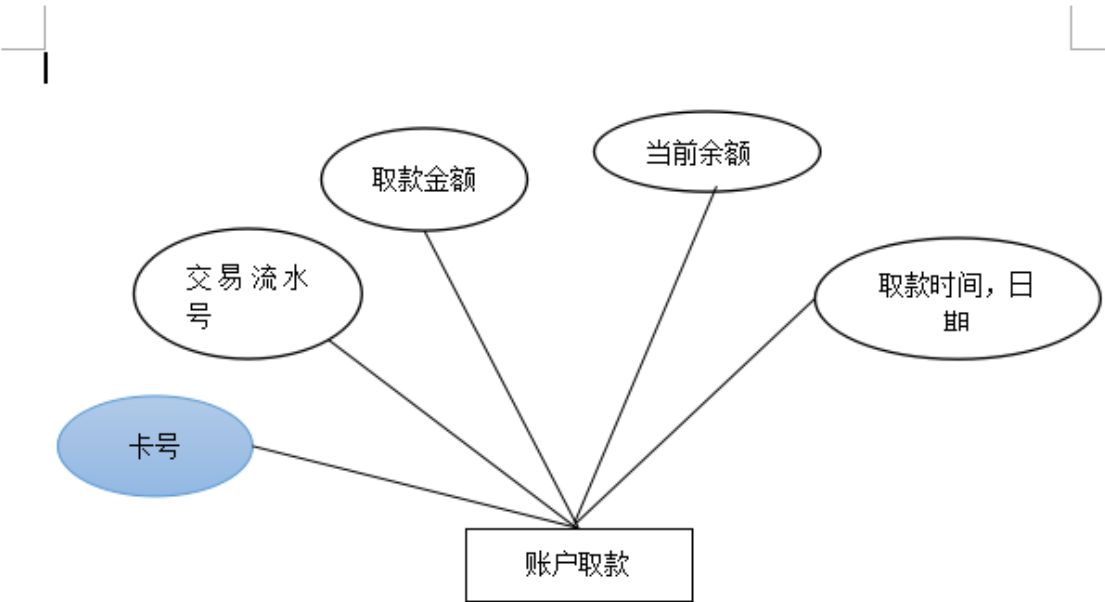


图 4-2 用户取款实体与属性关系图

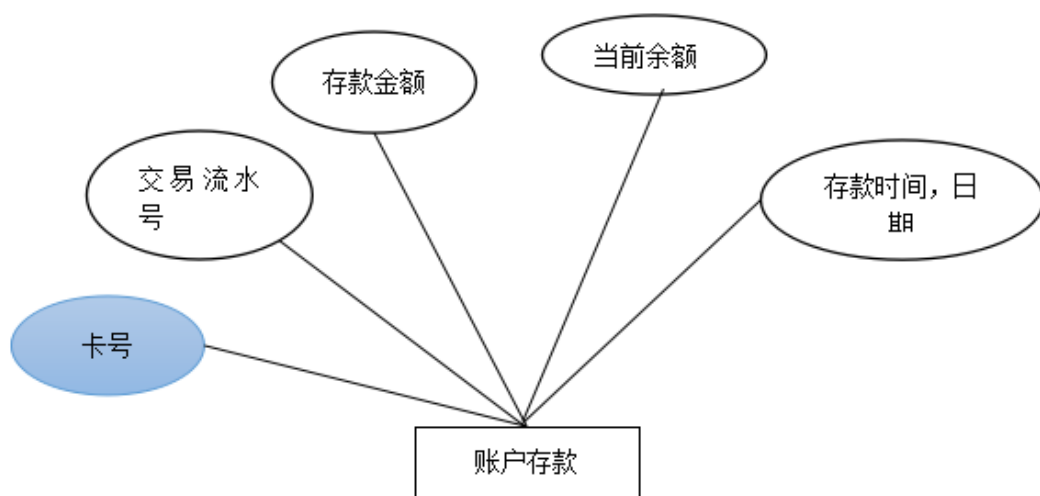


图 4-3 用户存款实体与属性关系图

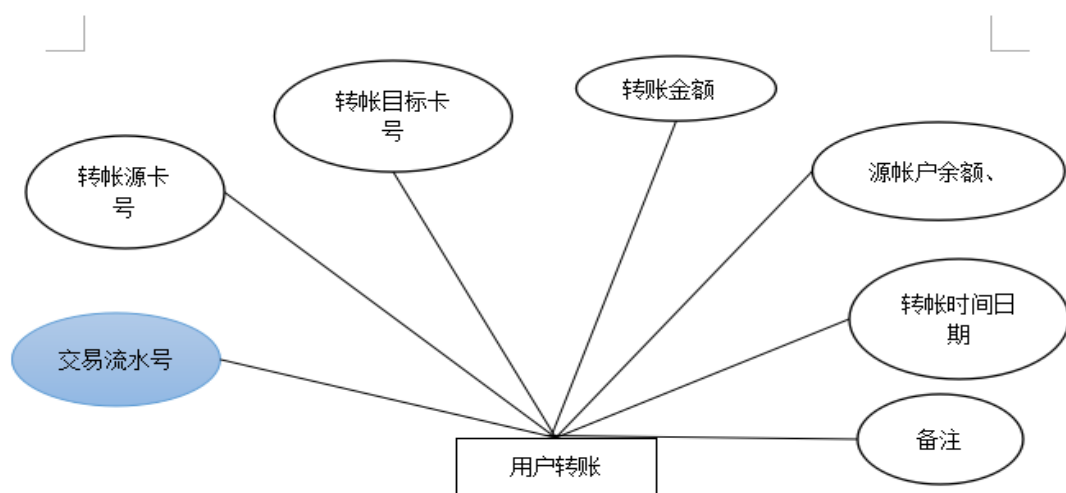


图 4-4 用户转账实体与属性关系图

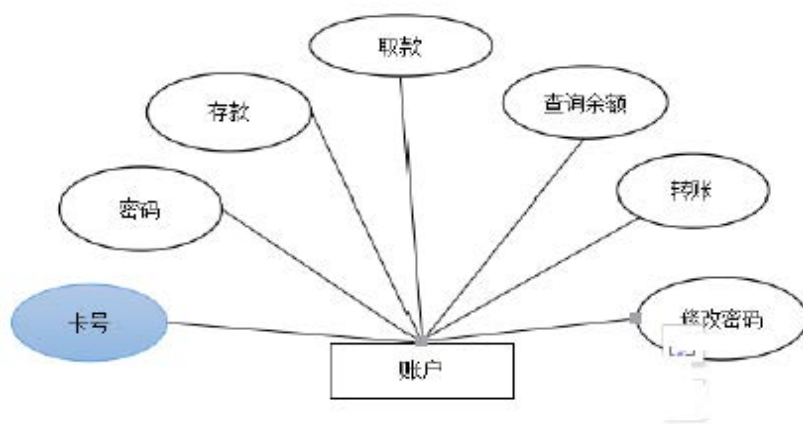


图 4-5 用户信息实体与关系属性图

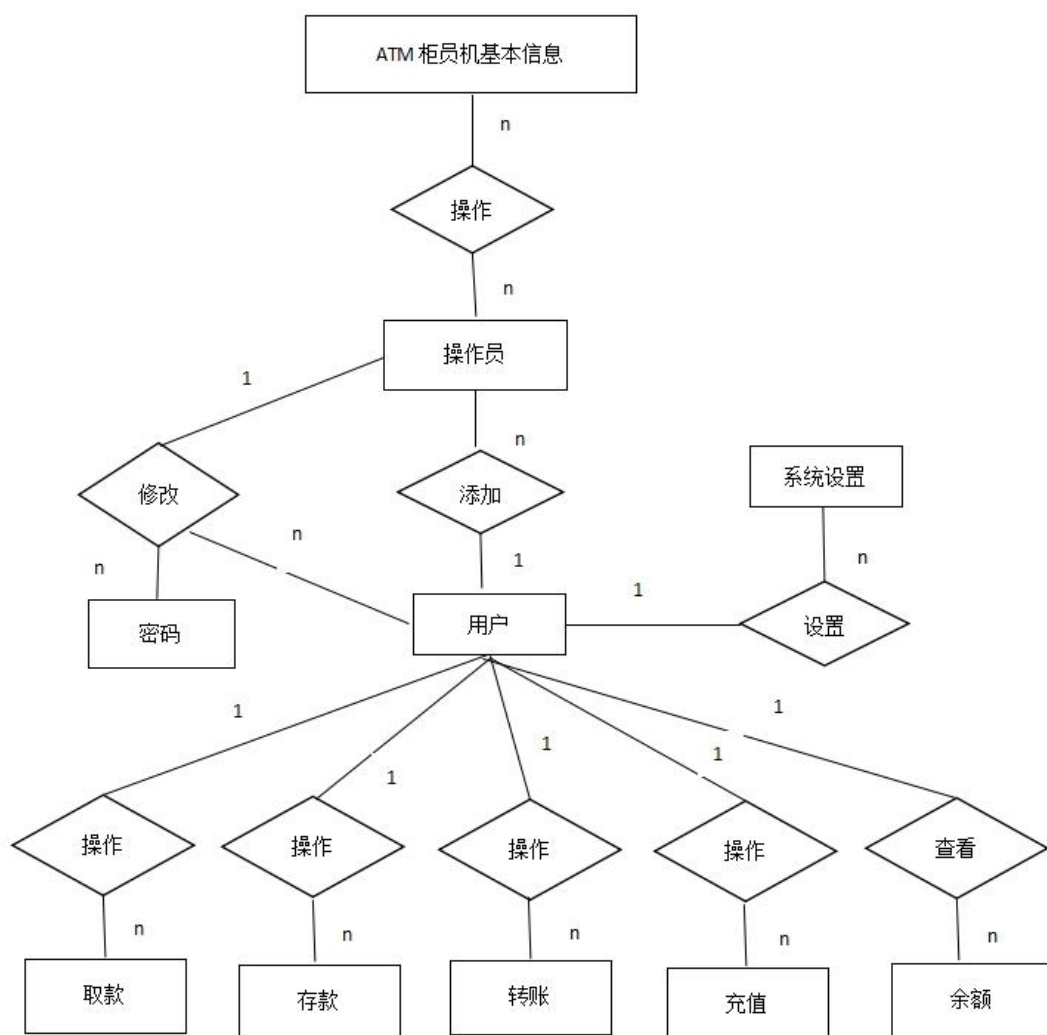


图 4-4 用户、账户、管理员信息 E-R 图

4.3.2 数据逻辑结构设计

表 4-1 用户信息表

字段	数据类型	长度	允许空	注释	备注
CardID	char	8		卡号	主键
UserID	char	18	×	持卡人身份证	
UserName	varchar	20	×	持卡人姓名	
password	char	6	×	密码	
StartTime	DateTime	8	×	开户日期	
Balance	float	4	×	当前余额	
Frozen	bit	1	×	是否冻结	
Content	varchar	100	√	备注	

表 4-2 存款记录表

字段	数据类型	长度	允许空	注释	备注
TradeID	int	4		交易流水号	主键
CardID	char	8	×	卡号	
Cash	float	4	×	存款金额	
Balance	float	4	×	当前金额	
Time	DateTime	8	×	存款日期时间	
Content	varchar	100	√	备注	

表 4-3 取款记录表

字段	数据类型	长度	允许空	注释	备注
TradeID	int	4		交易流水号	主键
CardID	char	8	×	卡号	
Cash	float	4	×	取款金额	
Balance	float	4	×	当前余额	
Time	DateTime	8	×	取款时间日期	
Content	varchar	100	√	备注	

表 4-4 转帐记录表

字段	数据类型	长度	允许空	注释	备注
TradeID	int	4		交易流水号	主键
SourceCardID	char	8	×	转帐源卡号	
TargetCardId	Char	8	×	转帐目标卡号	
Cash	float	4	×	转帐金额	
Balance	float	4	×	源帐户余额	
Time	DateTime	8	×	转帐时间日期	
Content	varchar	100	√	备注	

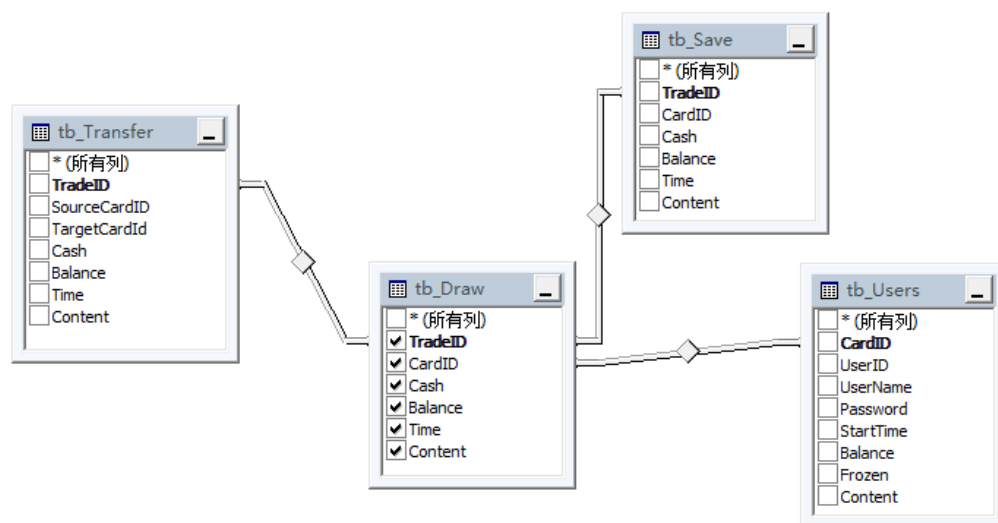


图 4-5 模拟 ATM 自动柜员管理系统系统数据表关系图

5 项目详细设计

5.1 导言

5.1.1 目的

该文档描述的模拟 ATM 自动柜员管理系统的详细设计，其主要内容包括：
系统中各子模块功能的详细设计

本文档预期读者：

1. 开发人员：马进
2. 项目管理人员：纪骁宵，姜佳新，李冰
3. 测试人员：刘帅20164454

5.1.2 定义及缩写

账号，在银行中，事物应用的单个账号。每个顾客可以拥有多个账号。用户，拥有银行的一个或多个账号的人。可以是一个人或多个人，或者是公司。相同的人，拥有不同的银行账号被认为是不同的落户。

ATM，ATM 是 Automatic Teller Machine 的缩写，意为自动取款机。是一种高度精密的机电一体化设备，利用磁卡或智能 IC 卡储存用户信息并通过加密键盘输入密码然后通过银行内部网络验证并进行各种交易的金融自助设备。

模拟 ATM 自动柜员管理系统：Simulation of ATM teller management system

5.2 界面设计

5.2.1 模拟 ATM 自动柜员机管理系统登录界面设计

系统登录界面是本系统的首页，也是使用者进入系统的入口。本页设计由用来选择权限的 6 个按钮、设置文字 1 个组合框、用来输入银行卡账号和账户密码的 2 个文本框组成，简洁大方。

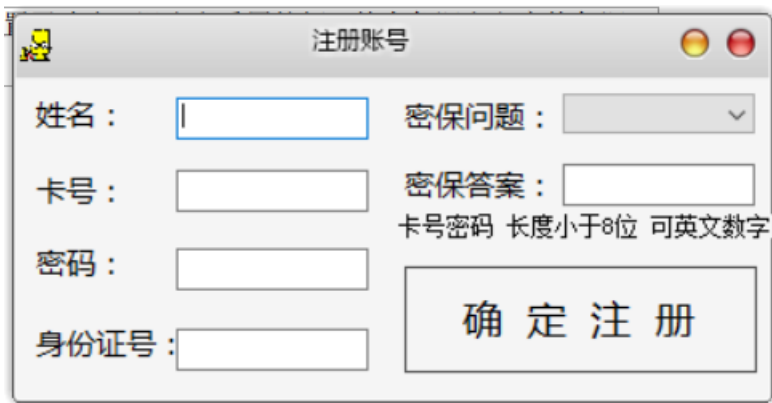


图 5-1 系统登录界面

5.2.2 账户开户注册界面设计

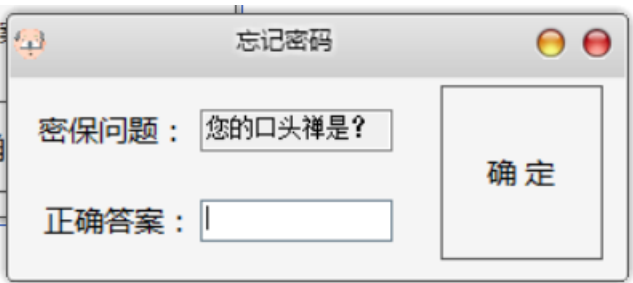
账户注册界面主要是为了没有在本行开户的用户提供方便，为了用户注册时的安全做出了保障。本界面提供了用来输入账号、账号密码、姓名、证件号、手机号和

预存金额的 6 个文本框，并设置了确定、退出和重置按钮。基本色调和主窗体色调一致，风格统一。



The '注册账号' (Register Account) window contains the following fields and controls:

- 姓名 (Name): Text input field.
- 卡号 (Card Number): Text input field.
- 密码 (Password): Text input field.
- 身份证号 (ID Number): Text input field.
- 密保问题 (Security Question): Dropdown menu.
- 密保答案 (Security Answer): Text input field.
- 卡号密码 长度小于8位 可英文数字 (Card Number Password length less than 8 bits, can be English and numbers): Text label.
- 确定注册 (Confirm Registration): Button.



The '忘记密码' (Forgot Password) window contains the following fields and controls:

- 密保问题 (Security Question): Text input field with the placeholder text '您的口头禅是?' (What is your catchphrase?).
- 正确答案 (Correct Answer): Text input field.
- 确定 (Confirm): Button.

图 5-2 用户开户注册界面

5.2.3 用户操作界面设计

用户操作平台是用户登录成功后跳转到的界面。左侧设置了取款、存款、转账、退卡四个按钮，右侧设置了查询余额、修改密码、查询记录、关于我们四个按钮。用户可根据需要进行相应选择。基本色调和主窗体色调一致，风格统一。



The '主页面' (Main Page) window displays the following elements:

- Top Bar: 主页面 (Main Page) title and window controls.
- Left Column: 取款 (Withdraw), 存款 (Deposit), 转账 (Transfer), 退卡 (Return Card) buttons.
- Right Column: 查询余额 (Check Balance), 修改密码 (Change Password), 查询记录 (Check Record), 关于我们 (About Us) buttons.
- Center: A banner with the text '身份证银行卡不能放在一起' (ID card and bank card cannot be placed together) and an image of a hand inserting a card into an ATM.
- Bottom: A row of five numbered tabs: 1, 2, 3, 4, 5.

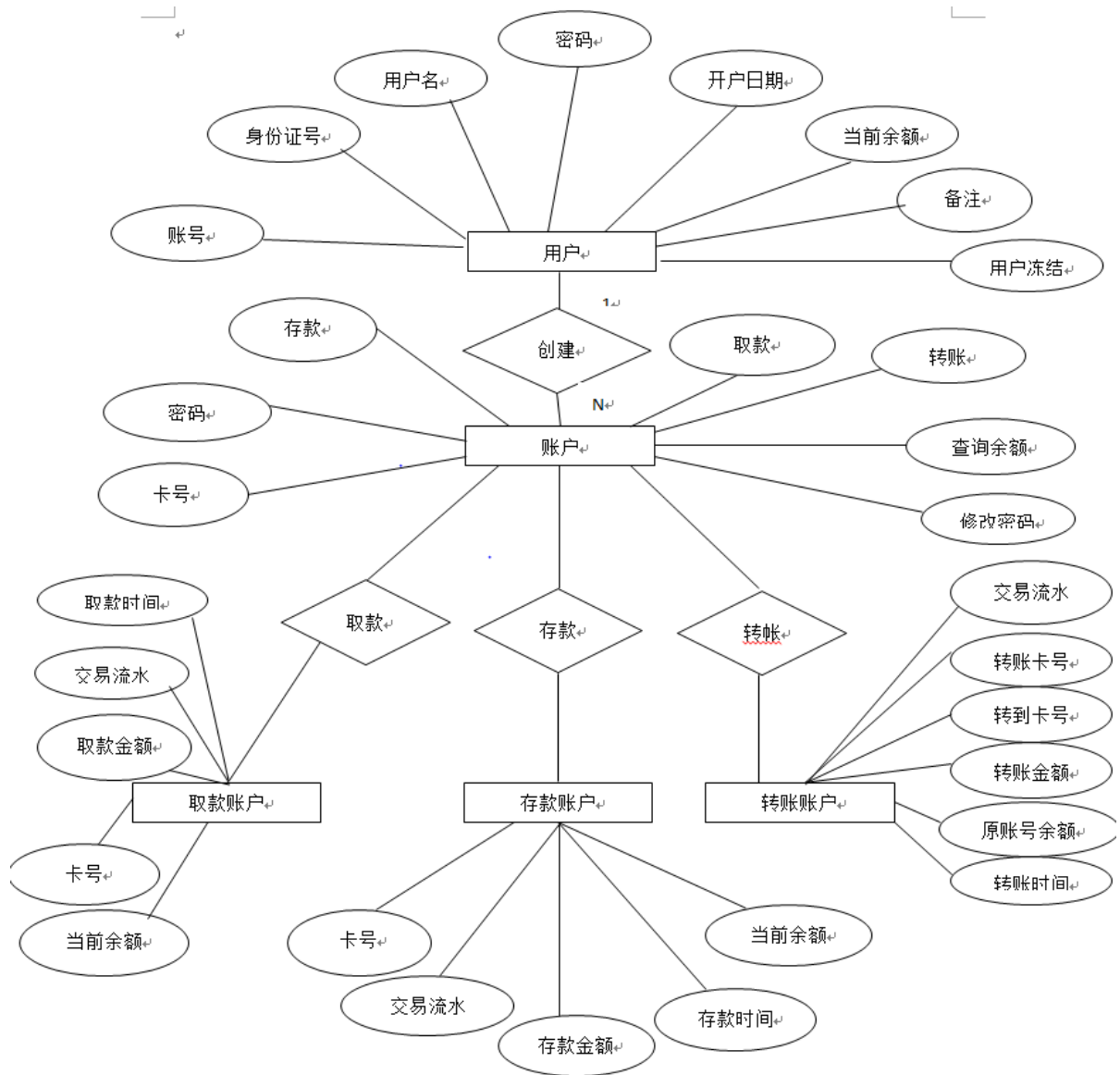
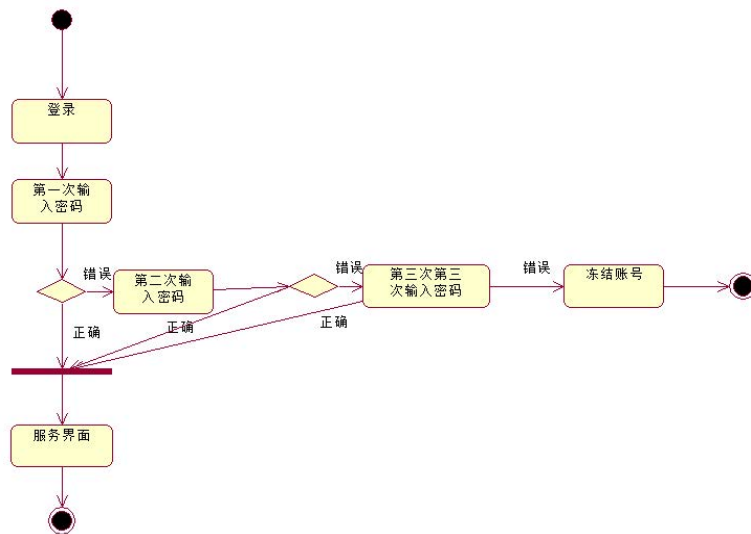


图 5-2 E-R图

5.3 功能详细设计

5.3.1 模拟 ATM 自动柜员机管理系统功能活动图



5-4用户登录

5.3.2 功能活动图

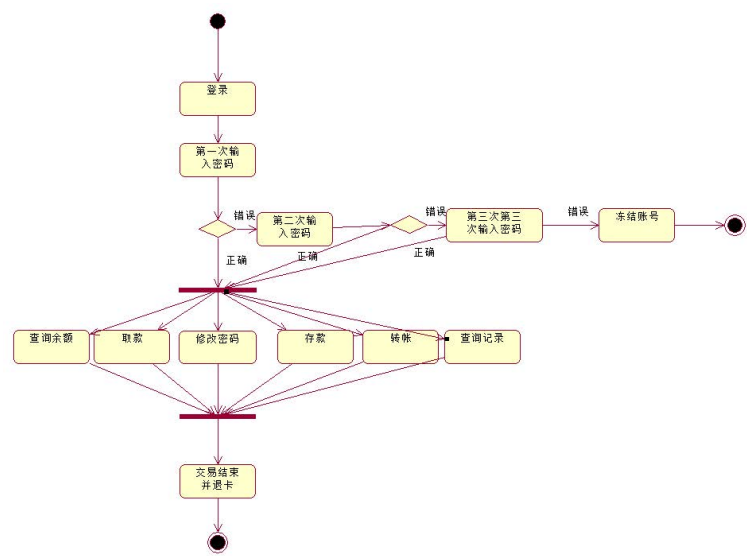


图 5-4 各功能活动图

6. 项目实施

6.1 导言

6.1.1 目的

本文档编写的目的是描述模拟 ATM 自动柜员机管理系统编码规范和各个模块的实现，主要说明：

- 1. 编码规范
- 2. 命名规范

本文档预期读者

- 1. 开发人员：马进
- 2. 项目管理人员：纪骁宵，姜佳新，李冰
- 3. 测试人员：刘帅20164454

6.1.2 缩写及定义

账号，在银行中，事物应用的单个账号。每个顾客可以拥有多个账号。用户，拥有银行的一个或多个账号的人。可以是一个人或多个人，或者是公司。相同的人，拥有不同的银行账号被认为是不同的落户。

ATM，ATM 是 Automatic Teller Machine 的缩写，意为自动取款机。是一种高度

精密的机电一体化设备，利用磁卡或智能 IC 卡储存用户信息并通过加密键盘输入密码然后通过银行内部网络验证并进行各种交易的金融自助设备。

模拟 ATM 自动柜员管理系统：Simulation of ATM teller management system

6.2 编码格式规范

6.2.1 缩进规则

4 个空格为一个单位。

6.2.2 断行规则

在逗号后换行。分号后换行。

一行的代码超过屏幕的长度时换行。

6.3 命名规范

6.3.1 类/接口

名字应该能够标识事物的特性。

名字尽量不使用缩写，除非它是众所周知的。

名字可以有两个或三个单词组成，但通常不应多于三个。

在名字中，所有单词第一个字母大写。或名词短语命名类. 少用缩写。

6.3.2 控件

Form 缩写 frm。

TextBox 缩写 txt。

Button 缩写 btn。

ComboBox 缩写 cbox。

DataSet 缩写 dst。

DatetimePicke 缩写 dtp。

ErrorProvider 缩写 errorPr。

ListBox 缩写 lb。

GroupBox 缩写 gbox。

PictureBox 缩写 picbox。

TreeView 缩写 trv。

6.3.3 方法

必须为动词，

6.4 功能实现

6.4.1 模拟 ATM 自动柜员机管理系统窗体功能实现

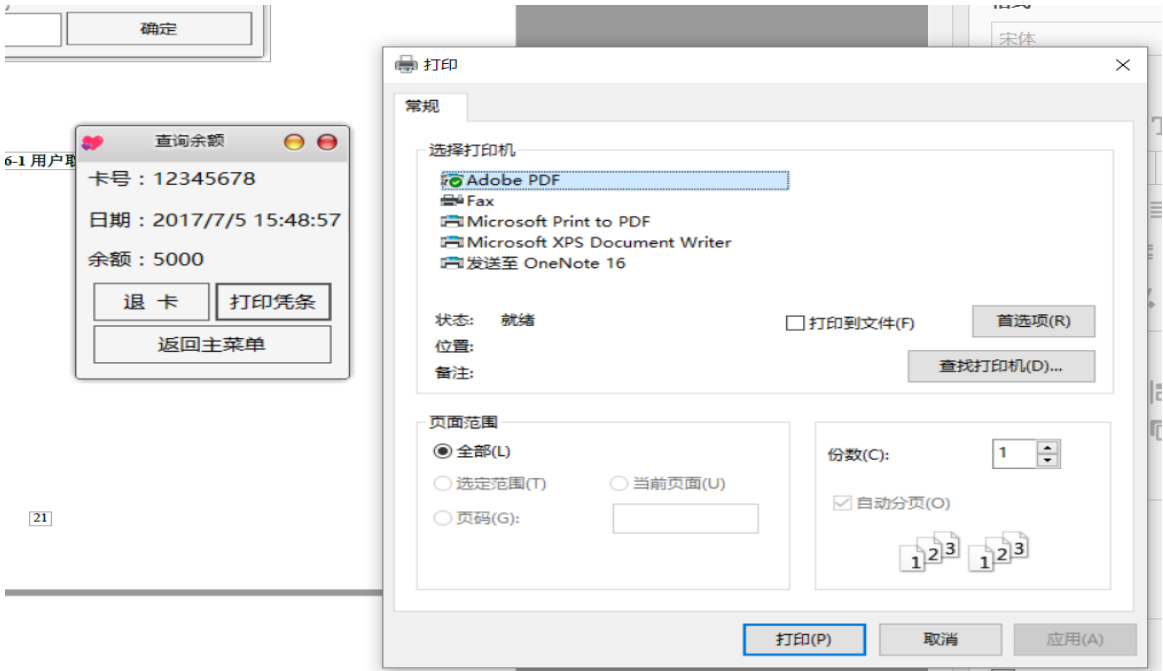
在此窗体，可以存款、取款、充值、转账、查询余额、修改密码

1. 用户取款窗体

在用户取款窗体可以输入取款金额或按固定的金额按钮，点击“确定”按钮进行取款，或按返回和退卡。



图 6-1 用户取款



6.4.2 模拟 ATM 自动柜员机管理系统窗体功能实现



图 6-2 用户存款



图 6-3 取款明细



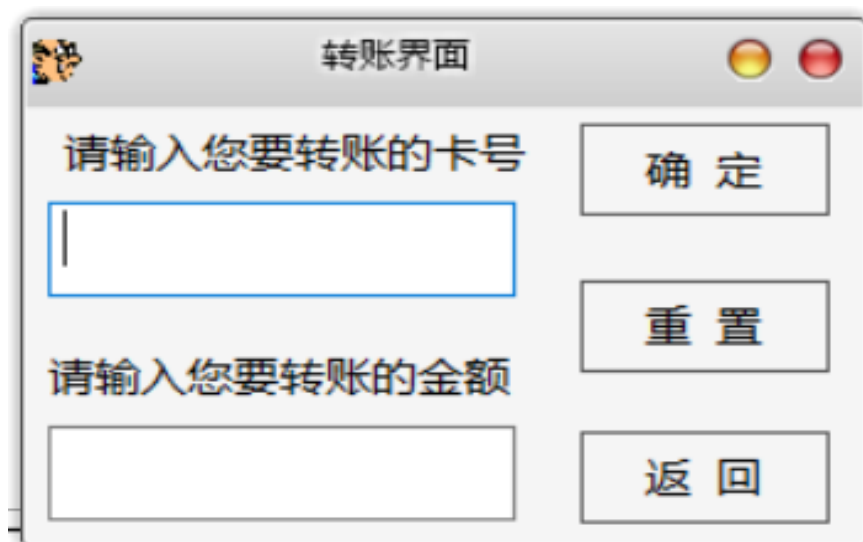
A screenshot of a software window titled "打印凭条" (Print Receipt). The window has a standard Windows-style title bar with a printer icon on the left and three colored buttons (yellow, red, green) on the right. The main content area displays the following text in a simple black font:

- 交易流水号 : 4369
- 存款卡号 : 12345678
- 存款金额 : 2000
- 当前余额 : 5000
- 存款日期 : 2017/7/5 15:45:54
- 备注 : 无

At the bottom of the window, there are three buttons arranged in two rows:

- Top row: "退 卡" (Withdraw Card) and "打印凭条" (Print Receipt).
- Bottom row: "返回主菜单" (Return to Main Menu).

图 6-4 用户存款



A screenshot of a software window titled "转账界面" (Transfer Interface). The window has a title bar with a small icon on the left and three colored buttons (yellow, red, green) on the right. The main content area contains two input sections, each with a label and a text box:

- Section 1: Label "请输入您要转账的卡号" (Please enter the card number you want to transfer), followed by a text box with a vertical cursor.
- Section 2: Label "请输入您要转账的金额" (Please enter the amount you want to transfer), followed by an empty text box.

To the right of these input fields are three buttons arranged vertically:

- "确 定" (Confirm)
- "重 置" (Reset)
- "返 回" (Return)

图 6-5 用户转账

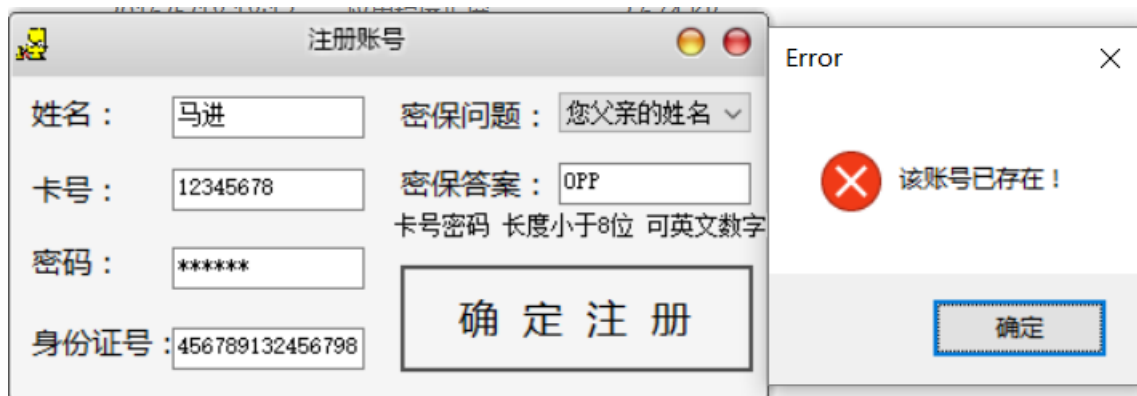


图 6-6 用户注册

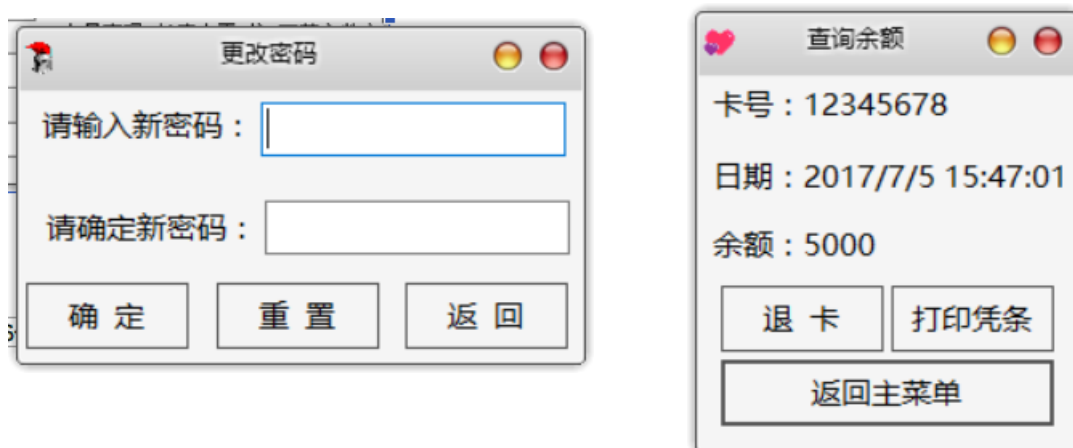


图 6-7 用户修改密码

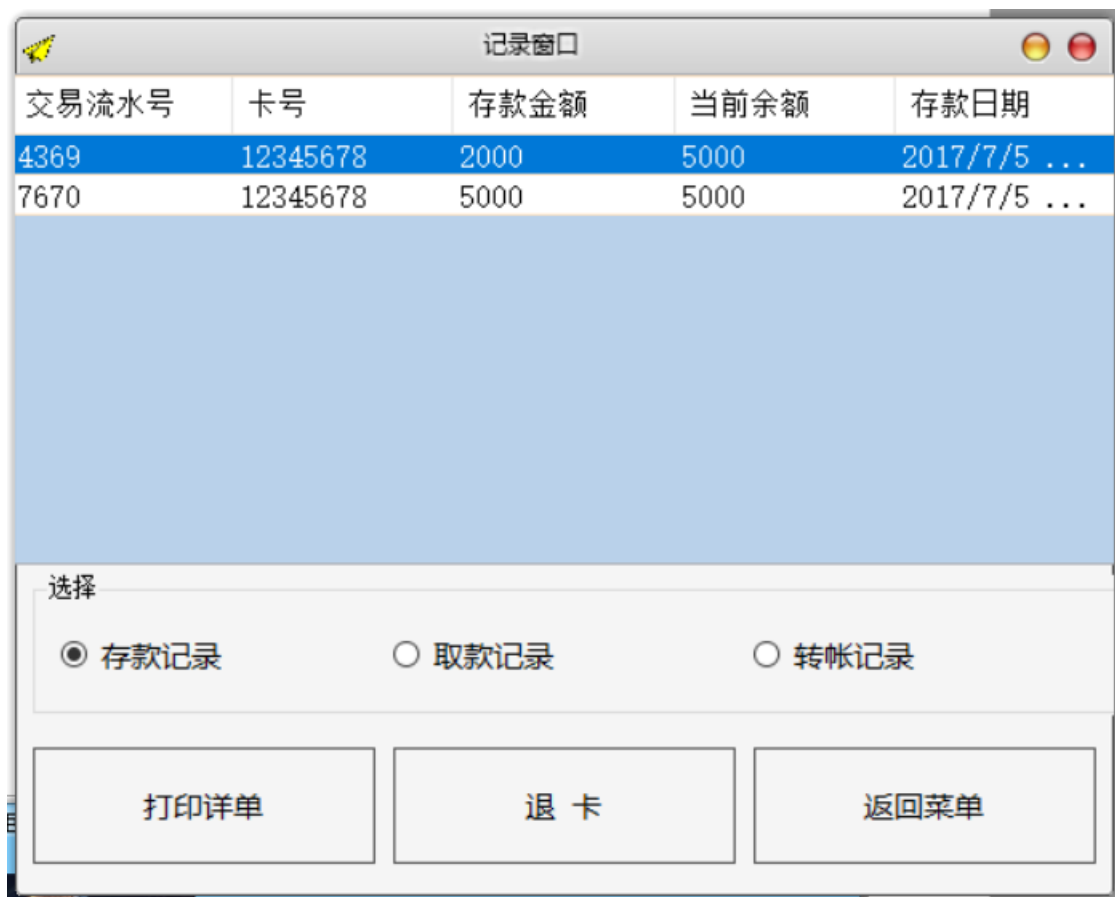


图 6-8 用户查询

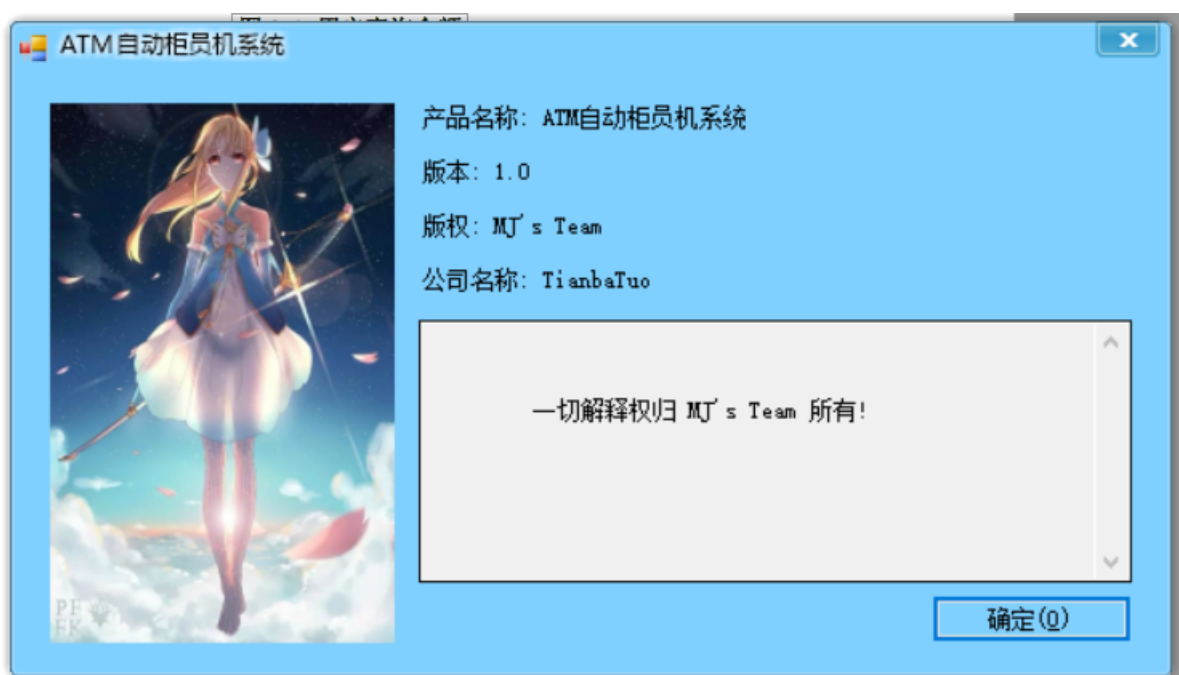


图 6-9 用户帮助

7. 测试

7.1 登录功能测试

1. 正常登录

测试编号	T1.1
测试项目	登录窗体
测试子项目	登录
测试目的	正常登录
预置条件	1 安装和部署业务
输入数据	1 用户“12345678”，密码“000000”
测试步骤	1 进入登录窗体 2 录入用户名“马进”，密码“000000” 3 点击登录
预期结果	1 用户能成功登录业务窗体
实际结果	1 用户能成功登录业务窗体
结论	功能正确

2. 用户名为空

测试编号	T1.2
测试项目	登录窗体
测试子项目	登录
测试目的	用户为空
预置条件	1 安装和部署业务
输入数据	2 用户“12345677”，密码“111111”
测试步骤	1 进入业务登录页面 2 录入管理员“□”，密码“111111” 3 点击登录
预期结果	1 弹出对话框，提示信息“用户名不能为空”
实际结果	1 弹出对话框，提示信息“用户名不存在，请重新输入”
结论	存在问题，可能是由于没有验证用户名是否为空的判断代码

7.2 注册功能测试

1. 注册成功

测试编号	T2.1
测试项目	注册窗体
测试子项目	注册成功
测试目的	成功注册
预置条件	1 注册账户业务
输入数据	1 新用户名“马进”，账号“12345678”，密码“000000”，身份证号“123465789123456789”，密码问题“您的口头禅是？”，答案“OPP”
测试步骤	1 进入注册窗体 2 新用户名“纪骁宵”，账号“12345677”，密码“111111”，身份证号“741852963741852963”，密码问题“您的口头禅是？”，答案"OPP" 3 点击注册
预期结果	1 显示文本框“注册成功”
实际结果	1 显示文本框“注册成功”
结论	功能正确

2. 注册失败

测试编号	T2.2
测试项目	注册窗体
测试子项目	注册失败
测试目的	未成功注册
预置条件	1 安装和部署业务
输入数据	1 新用户名“姜佳新”，
测试步骤	1 进入注册窗体 2 新用户名“姜佳新”，账号“12346572”，密码“666666”，身份证号“231083199705057118543” 点击注册

预期结果	1 显示提示框，提示密码应该为 6 位，身份证号应该为 18 位，预存金额应小于 1000 元
实际结果	1 显示提示框，提示密码应该为 6 位，身份证号应该为 18 位，预存金额应小于 1000 元
结论	功能正确

7.3 模拟 ATM 自动柜员机管理系统功能测试

取款功能测试

取款成功

测试编号	T3.3
测试项目	取款窗体
测试子项目	取款
测试目的	成功取款
预置条件	取款业务
输入数据	1 取款金额“100”
测试步骤	1. 进入取款窗体 2. 输入取款金额“100” 3. 显示账号，取款金额，时间，可用余额
预期结果	1 提示取款成功
实际结果	1 提示取款成功
结论	取款成功



取款失败

测试编号	T3.3
测试项目	取款窗体
测试子项目	取款
测试目的	钱未取出
预置条件	取款业务
输入数据	1 取款金额 “3000”
测试步骤	1. 进入取款窗体 2. 输入取款金额 “3000” 3. 显示账号，取款金额，时间，可用余额
预期结果	1 提示取款失败
实际结果	1 提示取款失败
结论	取款失败

8. 代码清单

8.1 模拟 ATM 自动柜员机管理系统模块

8.1.1 模拟 ATM 自动柜员机管理系统功能

登录界面代码：

1. 登录按钮代码

```
private void Btn_Login_Click(object sender, EventArgs e)
{
    if (sender.Equals(Btn_ApplyID))
    {
        Hide();
        new Frm_CardID().Show();
    }
    else
    {
        if (txt_Account.Text.Length != 8)
        {
            MessageBox.Show("卡号长度应为8!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```


#region 判断存在身份证号

1 个引用

static public bool IsExistUserID(string UserID)

```
{
    Con = new SqlConnection("server=.;database=db_ATM;integrated security=true");
    Con.Open();
    Com = new SqlCommand("select CardID from tb_Users where UserID='" + UserID + "'",
        Con);
    if (Com.ExecuteReader().HasRows)
    {
        End(false);
        return true;
    }
    else
    {
        End(false);
        return false;
    }
}
```

#endregion

#region 判断存在密码

1 个引用

static public bool IsExistPWS(string CardID,string Password)

```
{
    Con = new SqlConnection("server=.;database=db_ATM;integrated security=true");
    Con.Open();
    Com = new SqlCommand("select CardID from tb_Users where Password='" + Password +
        "'and CardID='" + CardID + "'", Con);
    if (Com.ExecuteReader().HasRows)
    {
        End(false);
        return true;
    }
    else
    {
        End(false);
        return false;
    }
}
```

#endregion

#region 保存账号信息

1 个引用

static public bool SaveNewUser(Model.Users User)

{

Start();

DataRow Res = Set.Tables[0]

.Rows.Find(User.CardID);

if (Res != null)

{

MessageBox.Show("该卡号已经存在!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);

return false;

}

DataRow Ac = Set.Tables[0].NewRow();

Ac["CardID"] = User.CardID;

Ac["UserID"] = User.UserID;

Ac["UserName"] = User.UserName;

Ac["Password"] = User.Password;

Ac["StartTime"] = User.StartTime

Ac["Balance"] = User.Balance;

Ac["Frozen"] = User.Frozen;

Ac["Content"] = User.Content;

Ac["Question"] = User.Question;

Ac["Answer"] = User.Answer;

Set.Tables[0].Rows.Add(Ac);

End(true);

return true;

}

#endregion

#region 保存新纪录

1 个引用

static public void SaveNewData(Model.Transfer Transfer)

{

Start();

DataRow Ac = Set.Tables[0].NewRow();

Ac["TradeID"] = Transfer.TradeID;

Ac["SourceCardID"] = Transfer.SourceCardID;

Ac["TargetCardID"] = Transfer.TargetCardID;

Ac["Cash"] = Transfer.Cash;

Ac["Balance"] = Transfer.Balance;

Ac["Time"] = Transfer.Time;

Set.Tables[0].Rows.Add(Ac);

End();

}

#endregion

```

#region 获取密保答案
1 个引用
static public string GetAnswer(string CardID)
{
    Start();
    DataRow Res = Set.Tables[0].Rows.Find(CardID);
    return Res["Answer"].ToString();
}
#endregion

#region 获取冻结状态
1 个引用
static public bool GetFrozen(string CardID)
{
    Start();
    DataRow Res = Set.Tables[0].Rows.Find(CardID);
    return Convert.ToBoolean(Res["Frozen"]);
}
#endregion

#region 获取用户余额
1 个引用
static public float GetUserMoney(string CardID)
{
    Start();
    DataRow Res = Set.Tables[0].Rows.Find(CardID);
    return Convert.ToSingle(Res["Balance"]);
}
#endregion

#region 得到数据源
1 个引用
static public DataSet GetDataSet()
{
    Ada = new SqlDataAdapter("select * from tb_Transfer where SourceCardID='" +
        Model.Users.CID + "' or TargetCardID='" + Model.Users.CID + "'",
        "server=.;database=db_ATM;integrated security=true");
    Set = new DataSet();
    Ada.Fill(Set);
    return Set;
}
#endregion

```

(参数) string CardID

```

#region 更改用户密码
1 个引用
static public void SaveNewPWS(string CardID,string NewPWS)
{
    Start();
    DataRow Res = Set.Tables[0].Rows.Find(CardID);
    Res["Password"] = NewPWS;
    End(true);
}
#endregion

#region 更改用户余额
1 个引用
static public void ChangeUserMoney(string CardID,float DoMoney,bool AddOrReduce)
{
    Start();
    DataRow Res = Set.Tables[0].Rows.Find(CardID);
    float OldMoney = Convert.ToSingle(Res["Balance"]);
    if (AddOrReduce)
    {
        OldMoney += DoMoney;
    }
    else
    {
        OldMoney -= DoMoney;
    }
    Res["Balance"] = OldMoney.ToString();
    End(true);
}
#endregion

#region 得到交易流水账号
static int NewTradeID;
2 个引用
static public void GetNewTradeID()
{
    NewTradeID = new Random().Next(1000, 9999);
    if (TradeID.IsExistTradeID(NewTradeID))
    {
        GetNewTradeID();
    }
}
3 个引用
static public int GetTradeID()
{
    GetNewTradeID();
    DAL.TradeID.AddNewTradeID(NewTradeID);
    return NewTradeID;
}

#endregion
}

```

1 个引用

```
private void Btn_Sure_Click(object sender, EventArgs e)
```

```
{
```

```
    Model.Users User = new Users(txt_CardID.Text, txt_UserID.Text, txt_UserName.Text,  
        txt_Password.Text, System.DateTime.Now, 0, false, null, cbox_PWSQuestion.Text,  
        txt_PWSAnswer.Text);
```

```
    if (BLL.Login.ExistCardID(txt_CardID.Text))
```

```
    {
```

```
        MessageBox.Show("该账号已存在!", "Error", MessageBoxButtons.OK,  
            MessageBoxIcon.Error);
```

```
    }
```

```
    else if (BLL.Login.ExistUserID(txt_UserID.Text))
```

```
    {
```

```
        MessageBox.Show("该身份证号码已存在!", "Error", MessageBoxButtons.OK,  
            MessageBoxIcon.Error);
```

```
    }
```

```
    else if (BLL.ApplyID.InfoIsOk(User))
```

```
    {
```

```
        BLL.ApplyID.SaveNewUser(User);  
        this.Close();
```

```
    }
```

```
}
```

```
private void Btn_Sure_Click(object sender, EventArgs e)
```

```
{
```

```
    if (txt_NewPWS.Text.Length != 6)
```

```
    {
```

```
        MessageBox.Show("密码长度为6位!", "Error", MessageBoxButtons.OK,  
            MessageBoxIcon.Error);
```

```
    }
```

```
    else if (txt_NewPWS.Text != txt_ReNewPWS.Text)
```

```
    {
```

```
        MessageBox.Show("密码输入不一致!", "Error", MessageBoxButtons.OK,  
            MessageBoxIcon.Error);
```

```
    }
```

```
    else
```

```
    {
```

```
        if (CardID == null) CardID = Model.Users.CID;  
        BLL.ChangePWS.SaveNewPWS(CardID, txt_NewPWS.Text);  
        BLL.Login.ChangeAccountFrozen(CardID, false);  
        this.Close();
```

```
    }
```

```
}
```

```

private void Btn_Sure_Click(object sender, EventArgs e)
{
    if (txt_Result.Text == "")
    {
        MessageBox.Show("金额不许为空!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if ((DoMoney = Convert.ToInt32(txt_Result.Text)) % 100 != 0)
    {
        MessageBox.Show("金额必须为100整数倍!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if (DoMoney > (UserMoney = BLL.DrawMoney.GetUserMoney(Model.Users.CID)))
        {
            MessageBox.Show("所取金额已大于卡里余额!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else if (DoMoney > 1000000)
        {
            MessageBox.Show("所取金额已大于ATM机余额!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            BLL.DrawMoney.ChangeUserMoney(Model.Users.CID, DoMoney, false);
            Model.Draw DRAW = new Model.Draw(BLL.DrawMoney.GetTradeID(), Model.Users.CID, DoMoney, DateTime.Now,
            null, UserMoney - DoMoney);
            BLL.DrawMoney.SaveNewData(DRAW);
            MessageBox.Show("取款成功!", "OK", MessageBoxButtons.OK);
            CloseKey = false;
            this.Close();
            new Frm_PrintDrawM() { Draw = DRAW }.Show();
        }
    }
}

// 1151用
private void Btn_Bring_Click(object sender, EventArgs e)
{
    Form A = null;
    if (sender.Equals(Btn_About))
    {
        A = new Frm_About();
    }
    if (sender.Equals(Btn_Bring))
    {
        A = new Frm_BringMoney();
    }
    if (sender.Equals(Btn_Change))
    {
        A = new Frm_ChangePWS();
    }
    if (sender.Equals(Btn_Draw))
    {
        A = new Frm_DrawMoney();
    }
    if (sender.Equals(Btn_SelectData))
    {
        A = new Frm_SelectData();
    }
    if (sender.Equals(Btn_Save))
    {
        A = new Frm_SaveMoney();
    }
    if (sender.Equals(Btn_Select))
    {
        A = new Frm_SelectMoney();
    }
    Time.Stop();
    Hide(); A.Show();
}

```



```

private void Btn_Sure_Click(object sender, EventArgs e)
{
    if (txt_Account.Text == "")
    {
        MessageBox.Show("卡号不能为空!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (txt_Account.Text.Length != 8)
    {
        MessageBox.Show("卡号长度应为8!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (!BLL.Login.ExistCardID(txt_Account.Text))
    {
        MessageBox.Show("该账号不存在!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (txt_Money.Text == "")
    {
        MessageBox.Show("金额不许为空!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if ((DoMoney = Convert.ToInt32(txt_Money.Text)) % 100 != 0)
    {
        MessageBox.Show("金额必须为100整数倍!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (txt_Account.Text == Model.Users.CID)
    {
        MessageBox.Show("不能给自己转账!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (DoMoney > (UserMoney = BLL.DrawMoney.GetUserMoney(Model.Users.CID)))
    {
        MessageBox.Show("所取金额已大于卡里余额!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (DoMoney > (UserMoney = BLL.DrawMoney.GetUserMoney(Model.Users.CID)))
    {
        MessageBox.Show("所取金额已大于卡里余额!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (DoMoney > 1000000)
    {
        MessageBox.Show("所取金额已大于ATM机余额!", "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    else
    {
        BLL.DrawMoney.ChangeUserMoney(Model.Users.CID, DoMoney, false);
        BLL.DrawMoney.ChangeUserMoney(txt_Account.Text, DoMoney, true);
        Model.Transfer TRANSFER = new Model.Transfer(BLL.DrawMoney.GetTradeID(), Model.Users.CID,
            txt_Account.Text, DoMoney, UserMoney - DoMoney, DateTime.Now, null);
        BLL.BringMoney.SaveNewData(TRANSFER);
        MessageBox.Show("转账成功!", "OK", MessageBoxButtons.OK);
        CloseKey = false;
        this.Close();
        new Frm_PrintBringM() { Transfer = TRANSFER }.Show();
    }
}

```

```

1 个引用
private void rBtn_Save_CheckedChanged(object sender, EventArgs e)
{
    if (rBtn_Save.Checked)
    {
        dgv_Save.Visible = true;
        dgv_Save.AutoGenerateColumns = false;
        dgv_Save.Font = new Font(new FontFamily("宋体"), 11);
        dgv_Save.DataSource = BLL.SelectData.GetSaveInfo().Tables[0];
        dgv_Bring.Visible = dgv_Draw.Visible = false;
    }
}

```

```

1 个引用
private void rBtn_Draw_CheckedChanged(object sender, EventArgs e)
{
    if (rBtn_Draw.Checked)
    {
        dgv_Draw.Visible = true;
        dgv_Draw.AutoGenerateColumns = false;
        dgv_Draw.Font = new Font(new FontFamily("宋体"), 11);
        dgv_Draw.DataSource = BLL.SelectData.GetDrawInfo().Tables[0];
        dgv_Bring.Visible = dgv_Save.Visible = false;
    }
}

```

```

1 个引用
private void rBtn_Bring_CheckedChanged(object sender, EventArgs e)
{
    if (rBtn_Bring.Checked)
    {
        dgv_Bring.Visible = true;
        dgv_Bring.AutoGenerateColumns = false;
        dgv_Bring.Font = new Font(new FontFamily("宋体"), 11);
        dgv_Bring.DataSource = BLL.SelectData.GetTransferInfo().Tables[0];
        dgv_Draw.Visible = dgv_Save.Visible = false;
    }
}

```

```

6 个引用
private void Btn_Five_Click(object sender, EventArgs e)
{
    if (sender.Equals(Btn_One))
    {
        txt_Result.Text = "100";
    }
    if (sender.Equals(Btn_Three))
    {
        txt_Result.Text = "300";
    }
    if (sender.Equals(Btn_Five))
    {
        txt_Result.Text = "500";
    }
    if (sender.Equals(Btn_Ten))
    {
        txt_Result.Text = "1000";
    }
    if (sender.Equals(Btn_Twl))
    {
        txt_Result.Text = "2000";
    }
    if (sender.Equals(Btn_Fiveth))
    {
        txt_Result.Text = "5000";
    }
}

```

9. 参考文献

1. 徐孝凯 贺佳英 著 《数据库基础与 SQL Server 应用开发》2011 年 4 月 清华大学出版社
2. [美]Jeffer R.Shapiro 著, 周之、黄玫译, 《SQL Server 2008 考大全》, 清华大学出版社, 2013 年 6 月。
3. (英)Robin Dewson 著 董明 译 《SQL Server 2008 基础教程》2010 年 5 月 人民邮电出版社
4. 姚永一, SQL Server 数据库实用教程, 电子工业出版社, 2010.
5. 高云, 崔艳春, SQL Server 2008 数据库技术实用教程, 北京: 清华大学出版社, 2011
6. 何玉洁, 梁琦, 数据库原理与应用 (第二版), 机械工业出版社, 2011
7. 隆华软件工作室 编著, 《SQL Server 2008 序设计》, 北京: 清华大学出版社, 2013 年 10 月
8. 刘甫迎, 刘光会, 王蓉.C#程序设计教程 (第 2 版) 北京: 电子工业出版社, 013.
9. 李兰友, 杨晓光. Visual C#.NET 程序设计[M]. 北京: 北方交通大学出版社, 2011.
10. 王珊, 萨师煊. 数据库系统概论[M]. 北京: 高等教育出版, 2008.
11. 高云, 崔艳春, SQL Server 2008 数据库技术实用教程, 北京: 清华大学出版社, 2011
12. 郭胜等. C#.NET 程序设计教程[M]. 北京: 清华大学出版社, 2012.
13. 明月创作室. Visual C#编程精彩百例[M]. 北京: 人民邮电出版社, 2010.
14. [美] John Sharp Jon Jagger. 《Microsoft Visual C#.NET 程序设计》 北京: 北京大学出版社, 2012.
15. 王昊亮, 李刚等. Visual C#程序设计教程[M]. 北京: 清华大学出版社, 2010.

10. 实训总结

小组总结

转眼间为期两个星期的实训就结束了，但是这些实训过程仍历历在目。

两个星期的实训让我学到了许多东西，不仅在理论上对IT等领域有了全新的认识，在实践能力上也有了一定的提高。此外，在实训期间我学会了如何把所学的知识在应用在实践中，让实践与理论相结合，真正做到学以致用。这次实训对于我以后学习、找工作真是受益菲浅，在短短的两个星期中让我初步从理性回到感性的重新认识，也让我初步的认识这个社会，对于以后做人所应把握的方向也有所启发!相信这些宝贵的经验会成为我今后成功的重要的基石。

作为一名大一学生，经过一年多的在校学习对专业知识也有了一定的认识和理解。在校期间一直都是忙于理论知识的学习，所以到了实训地方对每一个环节都相当的抽象。这次实训使我了解一个相关的运营操作程序，增强感性认识，并从中进一步了解、巩固与深化已经学过的理论知识，了解运作方式，将我们所学到的专业知识和具体实践相结合，以提高我们的专业综合素质和能力。

在实训的过程中我感受最深有以下几点：

其一、实训是让每个同学检验自己的综合能力。要想做好任何事，除了自己平时要有一定的功底外，我们还需要一定的实践动手能力，操作能力，因此，每个同学都应该多在实践中提高自己的能力。

其二、此次实训，我深深体会到了积累知识的重要性。俗话说：“要想为事业多添一把火，自己就得多添一捆材”。在这个竞争如此激烈的社会中，只有努力充实自己才能够站得住不被淘汰。

其三、实训是让每个同学确定自己的未来方向。以前缺乏实践，无法认识社会企业的需求，因此，对自己的未来也十分迷茫，但通过这次实训我们体会到了我们这一行的艰辛和干这一行所需要必备怎样的能力，只有我们确定了自己未来的方向、定下目标才能给自己定位，并努力提升自己来让自己适任职位。

最后，感谢老师对我的照顾和帮助。

签字：

年 月 日

个人总结

经过此次的 C#实训，使我明白小组合作的重要性。很多东西也许自己一个人做起来有些难度，但经过与小组的其他成员的讨论请教就可以解决。而且，当工作太多时，可以进行分工，加快工作的效率。

我们未来的职业，合作很重要在项目小组中我们不应该把自己看做一个个体，我们是小组的组成成员。小组成员之间的合作是否紧密会直接关系到这个项目的最终成果。在数据库的设计中架构尤为重要！！如果前面的数据库框架没有建好，那么后面的同学在创建数据库、表以及约束的时候，是很容易出错的。到最后数据库的错误就会越累积越多。在后面的四个用例中几乎每个都与 select 查询语句有关。因此，我们必须很熟悉查询语句的用法。尤其是连接查询和联合查询，这两种中的联合查询我就不是很会用。我们应该拿起大一的课本好好复习关于查询的语句。最后就是，我发现如果每个用例都参与的话，比我们光看书要来的有用多了。

因此我们以后如果想要让自己的技能提高还是需要多多的做一些案例。实践总会比光理论要来的有效。当然，最好就是理论与实践相结合。这样就需要我们多看书，上课时要认真听讲，老师在做案例是要尽量跟着敲。而在课外我们更不能因为所谓的休息而太过于放松自己，要经常看看书，将书上的例子打出来运行看看。

我懂得了实训的目的，就是让我们查找缺点补足自己。

“千里之行，始于足下”，这是一个短暂而又充实的实训，我认为对我走向社会起到了一个桥梁的作用，过渡的作用，是人生的一段重要的经历，也是一个重要步骤，对将来走上工作岗位也有着很大帮助。

离我们步入社会也没多久时间了，不管面临的是继续深造，还是就业的压力，我想我们都应该充分利用好这一段时间，充实、完善自我。

组员签字：

年 月 日

个人总结

通过这两周的实训，虽然实训的时间不长，但是总体上收获挺大的，实训对于我来说是必须要学会熟练操作的。可是自己也不会多些，有的知识点只是了解点，但我也没有去研究，实训时突然间觉得自己真的有点无知，虽然现在去看依然可以解决问题，但要浪费许多时间，这一点是我必须在以后的学习中加以改进的地方，同时也要督促自己在学习的过程中不断的完善自我。另外一点,也是在实训中必不可少的部分，就是要学会团队合作，互相帮助。所谓”当局者迷,旁观者清”，有些东西感觉自己做的是时候明明没什么错误，偏偏程序运行时就是有错误，有的时候没有思路，小组内几个人讨论讨论就会出现各种的想法。所以说，团队合作是很重要的一点，这在以后的工作或生活中也是很关键的。

此次实训，我深深体会到了积累知识的重要性。在实训当中我们遇到了不少难题，例如三层架构的问题之前并没有认真听，实训的时候做起来很费劲，但是我们问老师和同学，慢慢地对三层架构有了了解，也可以自己做出一部分来，问题得到了解决。两个星期的实训结束了，收获颇丰，同时也更深刻的认识到要做一个合格的程序员并非我以前想像的那么容易，最重要的还是细致严谨。社会是不会要一个一无是处的人的，所以我们要更多更快地从一个学生向工作者转变，总的来说通过这次实训我又对#有了进一步的了解和学习，也把以前没有及时弄懂的那些知识点做了补救。自己还是要多多的练习，有些不懂的问题不能略过，要把他弄懂，弄会。还是要把之前学的在认真的看一遍，多动手练习。

学生时代只是单纯的学习知识，而社会实践则意味着继续学习，并将知识用于实践，在这次的实训中我感受到了责任，作为一名大学生，更应该把学习作为保持工作积极性的重要性，在以后走上岗位以后我将积极的响应单位号召，结合工作实际，不断学习理论，技能知识和社会知识，将理论付诸了实践才能实现理论自身的价值。

组员签字:

年 月 日

个人总结

通过这两个星期的实训，我们小组出现了很多问题，在这之前我们 c#学的不是特别好，在做的过程中，我们遇到了很多难题，但是我们通过讨论，问老师。解决了大部分的问题。这次实训按照老师的要求，我们在小组之间都做了分工，我主要负责界面设计和代码功能实现，纪骁宵主要负责数据库设计，报告填写，画图，姜佳新主要负责整理报告。这次实训中我们既有有点也有不足的地方，首先我们的优势主要体现在

一：从选题上我们选择了最适合我们的情况，这样会使我们做起来比较轻松些。

二：当我们没有思路时，在一起互相讨论，这使我们有了很多新想法。

三：在分工上明确。

我们也有不足：

一：对c#的知识没有学习的特别透彻，对三层架构的了解也不是很多。

二：会出现很多不会的问题，这些问题都是以前不会，并且没有去研究的

通过这次实训，我们小组对 c#的知识有了更进一步的学习，也让我们认识到自己的不足之处，在做程序的过程中，我们在很多的地方，例如：三层架构的知识上，还有一些连接数据库的代码上，都出现了很多问题，这让我们意识到知识是很重要的，我们在这些地方的欠缺，也是我们需要注意并且改正的地方。在实训过程中，我们对以前不懂，不会的知识。也进行了学习，在我们的努力下弄明白了很多以前不同的问题同时也感受到理论实习是实战的基础，在下学期上课的时候更应该好好听课，有了扎实的理论才能更好的应用于实践中去，每学期结束的实训，实训是对我们这学期专业知识的一种检验，学到了很多在课堂上学不到的知识，也学到了在课堂上学不到的技能，又增长了见识，为我们进一步走向社会打下坚实的基础，也为以后走向工作岗位奠定了基础

通过这次实训，也让我们明白了团队合作的意义。

组员签字：

年 月 日

个人总结

在这次实训中，我对C#语言有了一个更深的了解认识，也对这个学期学的知识得到巩固，还尝试运行编程，每次运行程序成功，让我对下面的项目就充满信心。通过自己与同学合作编写程序，最终把最初的理论知识转化基本技能这次的实训，使我对C#语言的学习产生浓厚的兴趣。通过一周得实训我学到了不少的知识，了解了一些怎么找错误的方法。提高了注重团队学习精神。

从实训中我了解了ADO.NET的结构构成，熟悉了ADO.NET的组件的用法，以及Command和Connection的使用方法，了解了使用ADO.NET 进行事务处理。学会了怎么建立数据，怎么和window窗体建立连接。也学会了熟练应BindingSource和BindingNavigator控件并且熟练使用DataAdapter和DataReader。知道了创建一个具有添加、删除、修改等功能的SQL Server数据库应用程序。也知道了怎么使用TableAdapter配置向导功能。

在实训中，我认识到自己还有很多的知识没学好，基础知识没理清，而且许多东西还要去翻书，去上网搜索。而且遇到一些小错误运行不出来，就会烦躁不安，觉得有些自暴自弃或者抱怨项目的变态，以后要克服，尽量保持一颗良好的心态，学好C#语言，也学好用C#语言编写一个按要求的系统。

总之，这一切都成为我记忆里面的一个篇章，更是在C#语言编程上一个里程碑的。在今后也要有一套学习知识的系统，遇到问题自己能通过相关途径自行解决，这样不断能把问题解决，更能从中获得经验，以此来储备自己的能力。

组员签字:

年 月 日