**《程序设计基础》Project报告**

**（CST11103）**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | **史芳裘** | **学 号** | **2022xxxx** |
| **年级、专业、班级** | **HS学院2022级Phy拔Shit班** | | |
| **上课时间** | **2021-2022学年第1学期** | | |
| **Project成绩** | **代码成绩**  （70%） | **报告成绩**  （30%） | **综合成绩** |
|  |  |  |
| **评语** |  | | |
| **评价教师** |  | | |

**《程序设计基础》Project任务书**

|  |  |
| --- | --- |
| 名 称 | ATM模拟程序的设计与实现 |
| 类 型 | □验证性 □设计性 ■综合性 |
| 内 容 | 运用面向对象程序设计思想，基于命令行界面设计并实现一个ATM模拟程序，模拟常见的ATM功能。 |
| 要 求 | ATM模拟程序能够完成ATM的主要功能，包括：  ①显示欢迎词及提示信息；②用户插卡，ATM验证用户账号及密码有效性，输入错误3次即被锁卡；③余额查询：初始余额为10000元；④取款功能：每次取款余额为100的整数倍，有单笔和单日金额限制；⑤转账功能：可将本账户中的存款转入其它账户，转入账户账号需两次输入确认；⑥修改密码：密码为6位数字，新密码需要两次输入确认；⑦退卡。  设计实现命令行界面，界面应友好、方便操作。参考界面如图1所示。程序所涉及到的用户资料、银行帐户、存取款记录等信息保存在数据文件中。其中银行账户的格式如下：  账号 19位数字  姓名  身份证 18位字符  密码 6位数字  余额    图1 ATM模拟程序参考界面 |
| 任务时间 | 2022年11月27日至2023年1月8日 |

**代码评分标准（满分10分）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学习态度** | **程序功能** | **程序Bug** | **程序界面** | **异常处理** | **得分** |
| 抄袭、被抄袭 | -- | -- | -- | -- | **0** |
| 自己完成编码  无创新意识 | 没有完成实验要求的所有功能 | -- | -- | -- | **5** |
| 自己完成编码  有一定的创新意识 | 基本完成实验要求的所有功能 | 很多 | 一般 | 无 | **6** |
| 自己完成编码  有一定的创新意识 | 完成实验要求的所有功能 | 较多 | 一般 | 无 | **7** |
| 自己完成编码  有较强的创新意识 | 完成实验要求的所有功能，并有所扩展 | 较少 | 良好 | 有 | **8** |
| 自己完成编码  有较强的创新意识 | 完成实验要求的所有功能，并有所扩展 | 极少 | 友好 | 有 | **10** |

**报告评分标准（满分10分）**

|  |  |  |  |
| --- | --- | --- | --- |
| **文档规范性** | **文档条理性** | **表述准确性** | **得分** |
| 不规范，需求分析、系统设计、系统实现部分不完整。 | 描述缺少逻辑性，没有条理。 | 文字描述不准确。 | **3** |
| 较规范，有需求分析、系统设计、系统实现部分，但实质性描述较少。 | 描述具有一定的逻辑性，条理较清晰。 | 文字描述较准确，但缺少必要的类图、程序流程图等图表说明。 | **5** |
| 规范，有需求分析、系统设计、系统实现部分，描述较充分。 | 描述具有较好的逻辑性，条理清晰 | 文字描述较准确，所描述的类图、程序流程图基本准确、无大错。 | **7** |
| 规范，有需求分析、系统设计、系统实现部分，描述充分。 | 描述具有很好的逻辑性，条理清晰 | 文字描述准确，所描述的类图、程序流程图准确。 | **10** |

报告的主要内容：

1. 需求分析

拟解决的问题，分析软件在数据处理、数据存储和功能方面的需求。

制作ATM模拟程序，包含ATM主要功能：

1. 初始化程序：
   1. 初始化用户信息
      1. 程序初始化读用户信息文件，加载在程序临时内存
      2. 非空且格式必须全部正确，否则进行异常处理且不予登入
      3. 用户信息读写，包含账号、姓名、身份证、密码、余额及锁定状态，用csv文件储存(UserFile.csv)
   2. 初始化日取款限额信息
      1. 程序初始化读用户取款信息文件，加载在程序临时内存
      2. 程序处理当日用户取款信息，计算目标用户当日取款限额，储存于临时内存

当改变用户信息时，临时内存中用户信息重新写入csv用于保存

1. 只读插入卡信息用于读取账号：

用csv文件储存(CardFile.csv)，程序只读卡信息以获取账号

1. 核验账号密码正确性：
   1. 检测账户是否锁定
      1. 获取临时内存中用户信息的锁定状态，判断是否允许登入
   2. 核验密码正确性
      1. 程序获取键入密码，程序搜索账号对应密码并与键入密码比对，正确则进入主系统。
      2. 若输入密码错误次数超过三次，自动锁定账户，改写临时内存中用户信息的锁定状态，并立即重新写入csv文件保存
2. 主系统程序：
   1. 余额查询
      1. 调用临时内存中用户信息，输出余额信息
   2. 取款：
      1. 取款金额为100整数倍
      2. 单次取款限额5000，单日取款限额20000（依托时间日期程序实现）
      3. 取款信息读写，包含取款账号、存款金额及时间日期，用csv文件储存(WithdrawLog.csv)
   3. 存款：
      1. 存款信息只写，包含存款账号、存款金额及时间日期，用csv文件储存(DepositLog.csv)
   4. 转账
      1. 需两次输入目标账号以及转账金额
      2. 转账信息只写，包含转出、转入账号、转账金额及时间日期，用csv文件储存(TransferLog.csv)
   5. 修改密码
      1. 修改临时内存中用户信息的密码，修改后立即重新写入csv文件保存
   6. 退卡
      1. 退出主程序并将临时内存中用户信息重新写入csv文件保存
3. 时间日期程序：

获取当前时间日期，用于主系统程序输出时间日期以及取款程序比对当日取款限额

1. 信息储存程序：

用于将临时文件信息写入csv文件，用于主系统程序储存用户信息、取款、存款及转账信息

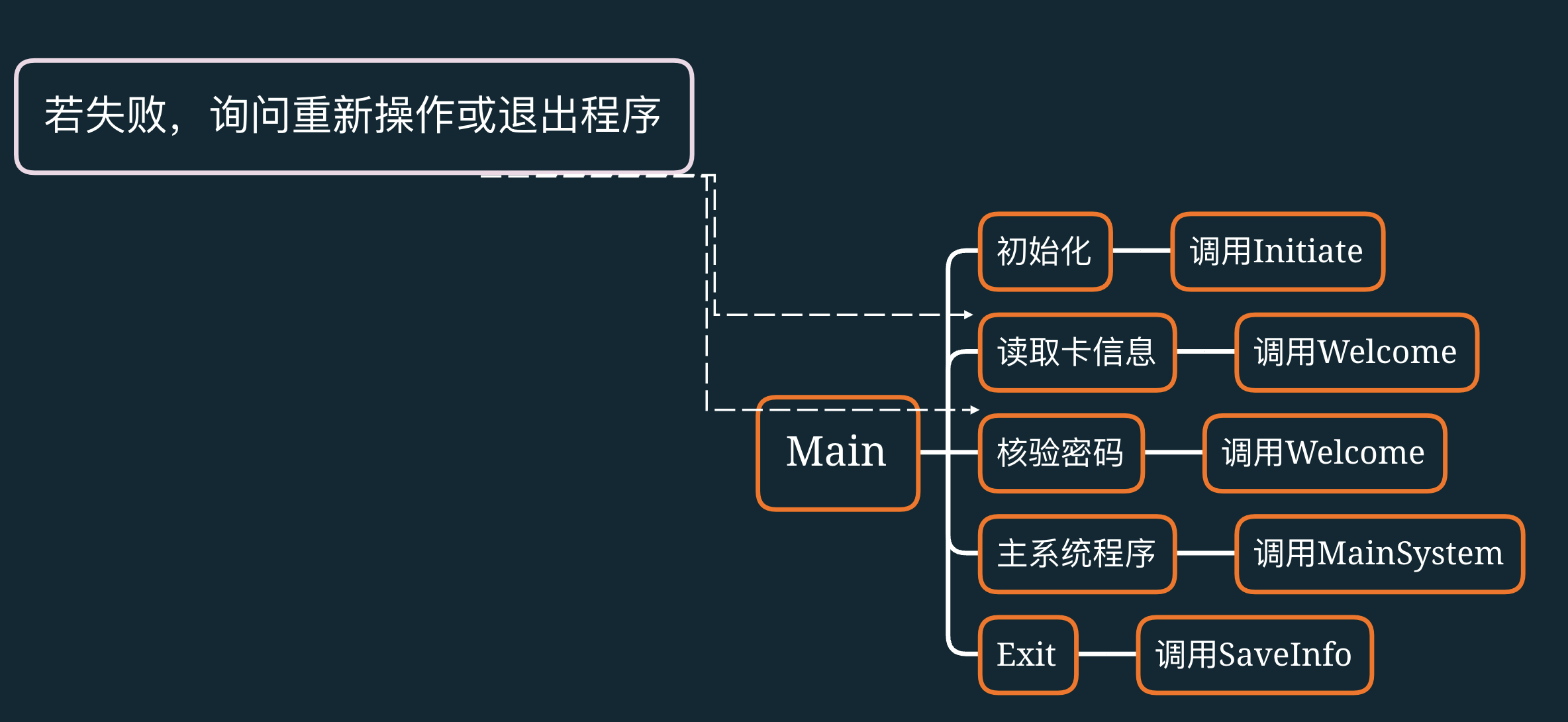
2. 系统设计

软件模块如何设计，这样设计的目的和优点是什么？

程序分为七个模块，分别为Main, Initiate, Welcome, MainSystem, SaveInfo, TimeInfo, User

各模块功能及运行流程如下：

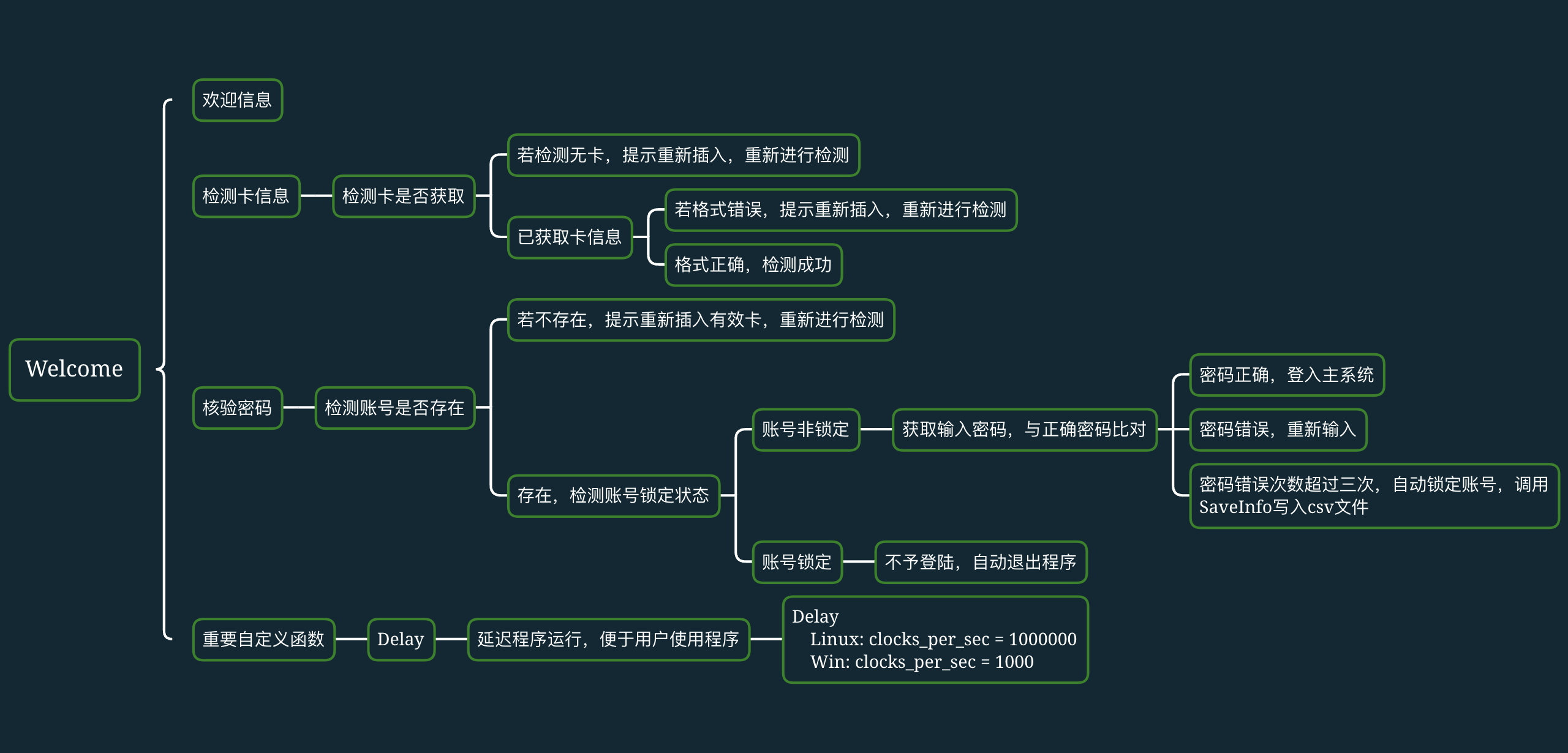
Main：程序总运行步骤，清晰呈现程序运行流程



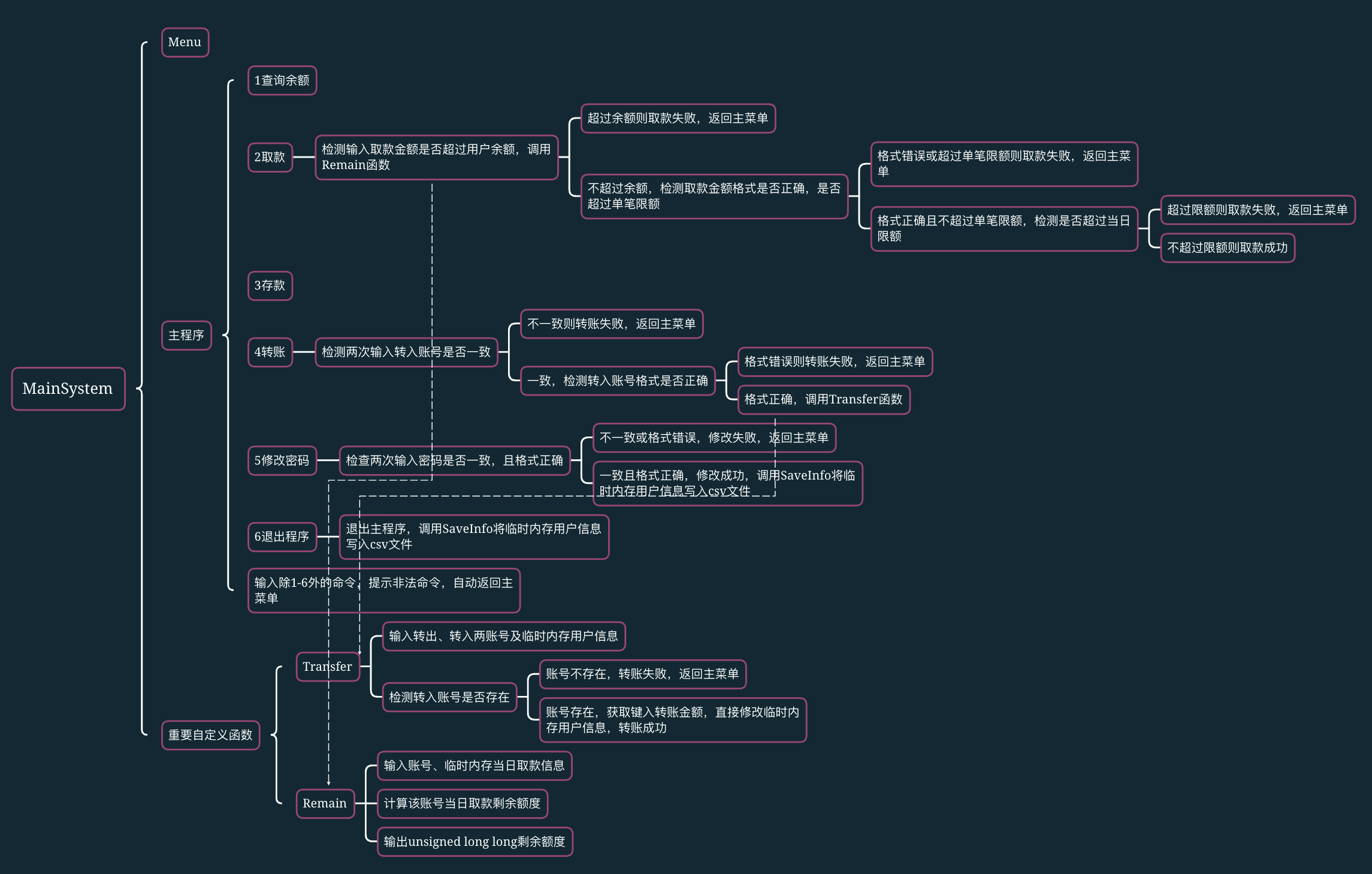
Initiate: 初始化操作，包含所有需要初始化的程序进程



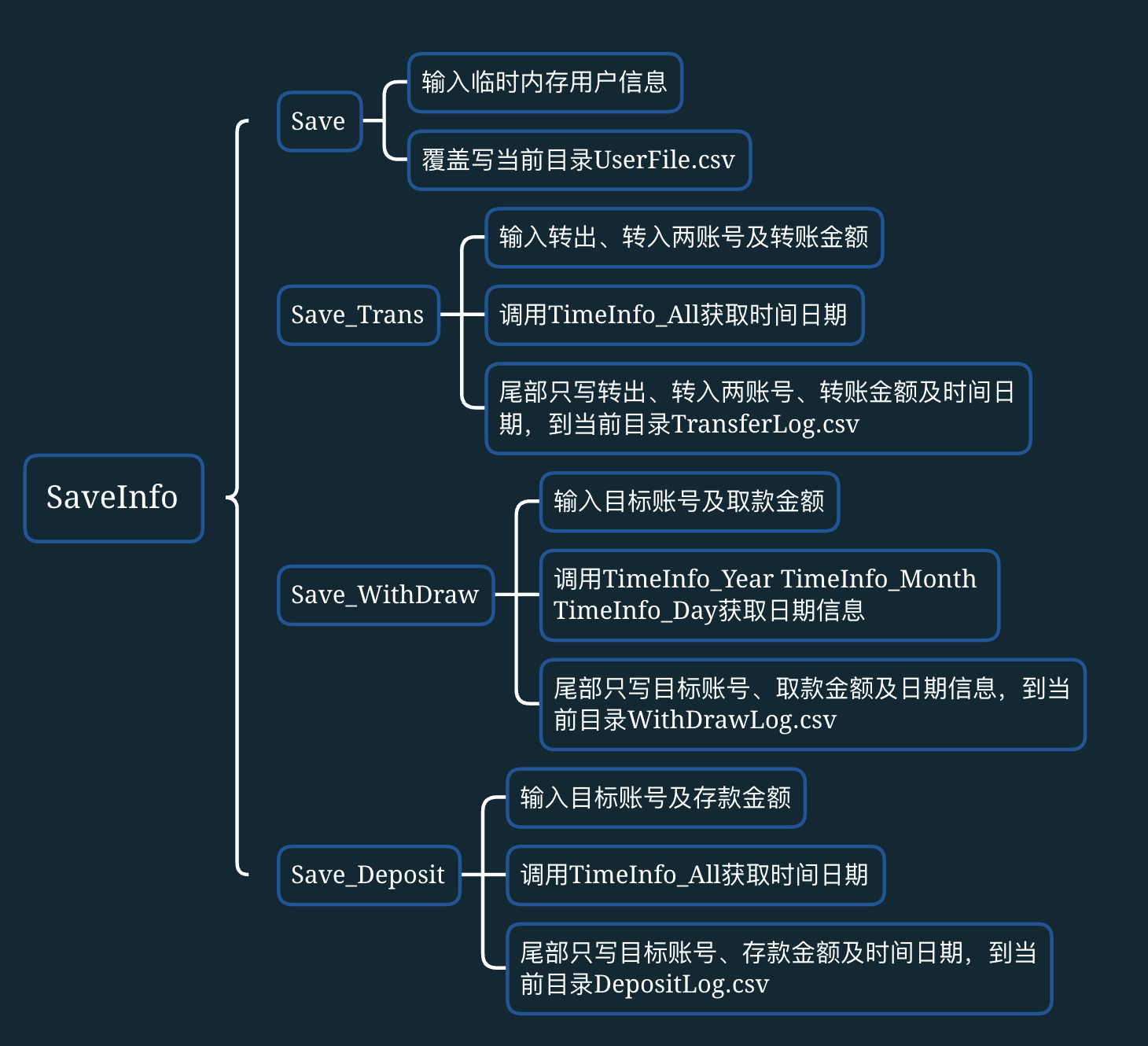
Welcome: 提示欢迎信息，主要处理初始化后、主程序前的插卡、登陆环节



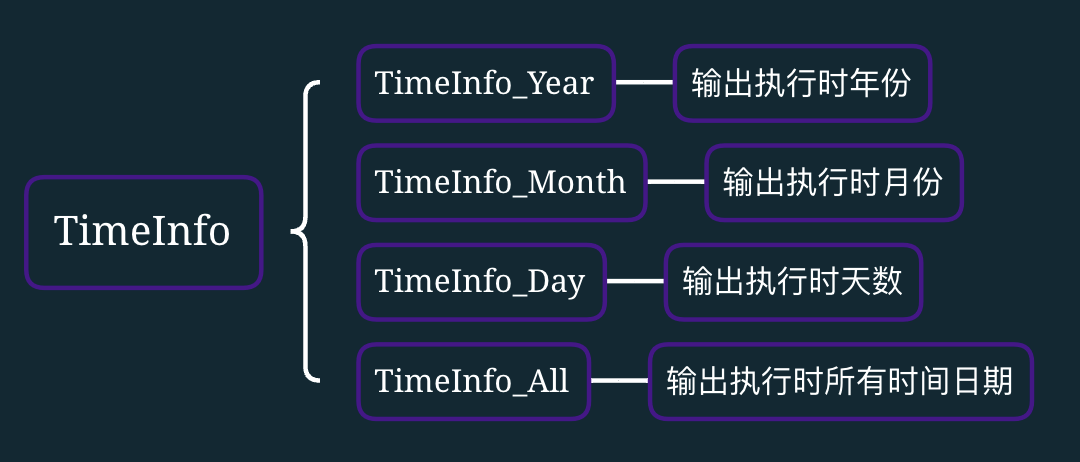
MainSystem: 主程序，包含登陆后进行查询余额、取存款、转账、修改密码、退出程序操作



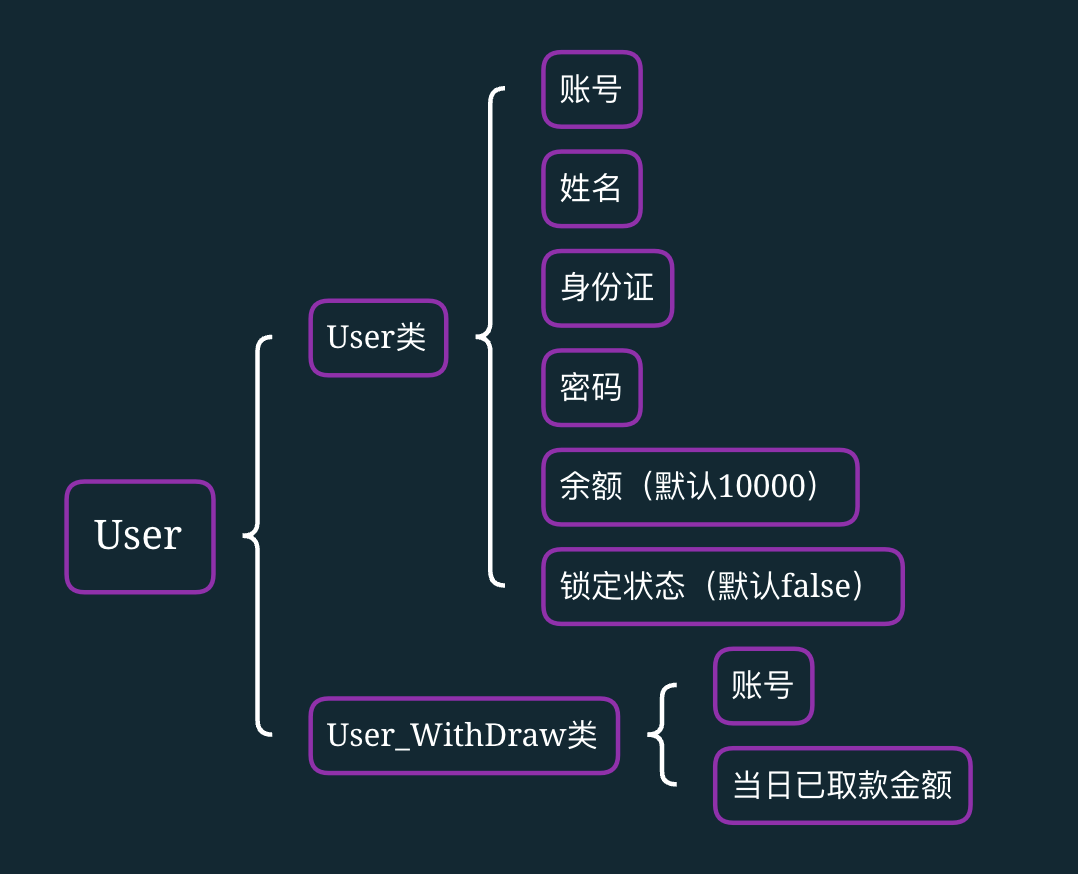
SaveInfo: 用于保存临时内存信息至csv文件，包含总共四种储存方式，便于主程序调用



TimeInfo: 获取时间信息，便于主程序以及存储程序调用



User: 类信息，包含User和User\_WithDraw类，便于储存至临时内存信息



分模块设计程序，便于清晰各模块功能，后期调优处理、异常处理及修复BUG时能快速找到目标节点，调试程序时能够便捷找到合适的断点。

Main程序仅用于调用其余程序，体现整个程序设计流程和运行步骤。

Initiate程序用于初始化csv文件，将本地目录下csv文件内容储存至类数据类型的vector数组中，成为程序运行中的临时内存信息，所有程序运行时操作都优先修改临时内存信息，在指定节点处会调用SaveInfo程序对本地csv文件内容进行修改，在结束程序时会最终修改csv文件，以确保程序运行时所有操作都被记录且保存。初始化用户信息时会对接收到的内容格式进行判断，若格式错误会进行异常处理，以提示用户，便于使用。初始化当日取款限额信息，便于在主程序进行取款操作时能对当日取款限额进行判断。

Welcome程序用于初始化后、主程序前的插卡检测账号以及核验密码正确性。密码输入错误超过三次后，会更改临时内存中用户信息的锁定状态，随后立即更新本地csv文件，以确保锁定状态保存到csv文件。

MainSystem程序用于主程序操作。键入1-6以选定目标服务，若键入其余按键，则提示未知命令并重新进入主程序。在各服务中若出现格式错误等问题则会进行异常处理，均会返回主菜单。在取存款、转账操作中，会按照图中所示调用SaveInfo程序对本地csv文件内容修改，以存储取存款、转账信息。选定退出程序后会自动调用SaveInfo程序对临时文件中用户信息覆盖写本地csv文件。

SaveInfo程序包含四种存储至csv文件方式，以便于主程序调用。

TimeInfo程序获取时间信息，便于主程序以及存储程序调用。

User程序包含两种类信息，便于储存至临时内存。

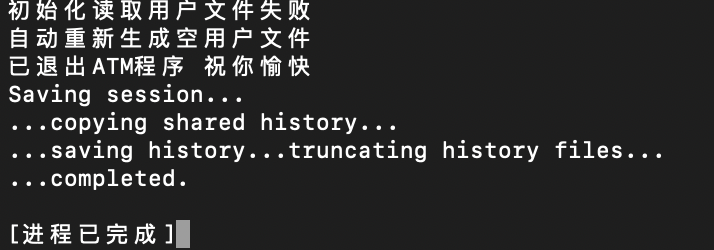
3. 系统实现

描述软件的实现效果，如何使用该软件？

软件使用时需至少包含可执行文件, UserFile.csv, CardFile.csv

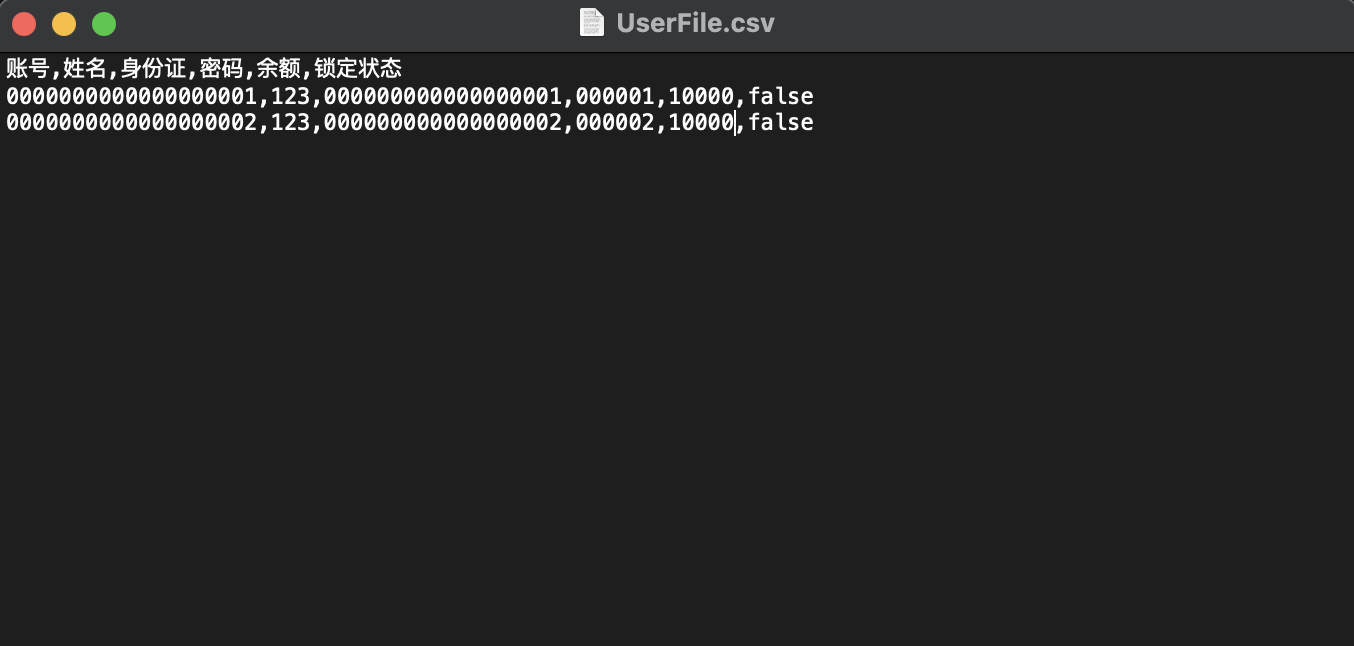
若缺失UserFile.csv则不能正常初始化：

会自动生成一个带有表头的空UserFile.csv文件, 并自动关闭程序





可手动填写UserFile.csv中信息，如下



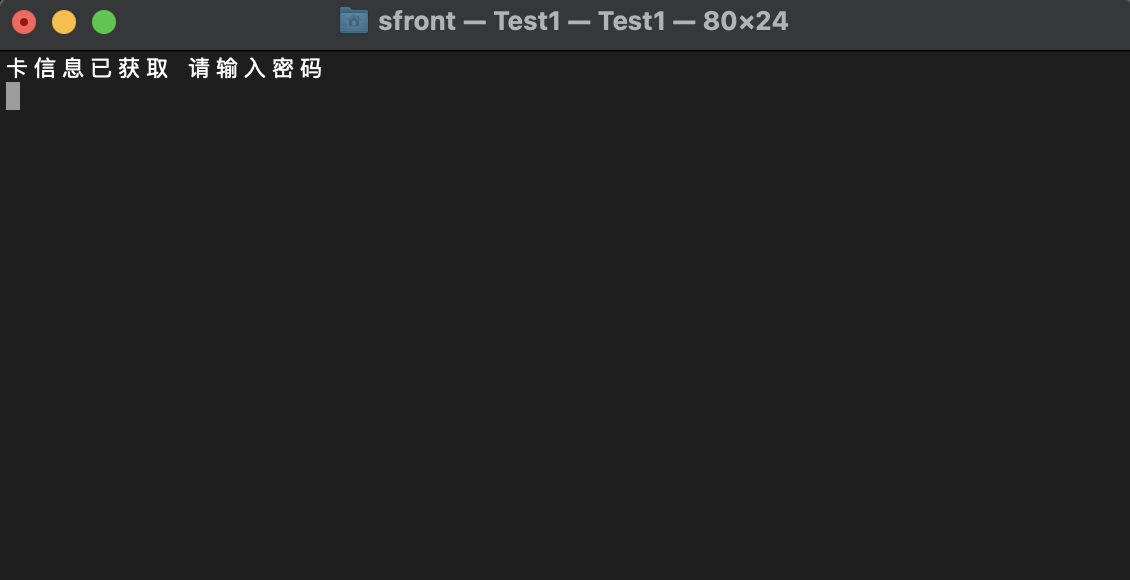
若含有UserFile.csv文件，缺失CardFile.csv文件

则会提示插入卡，并询问是否重新读取或退出程序

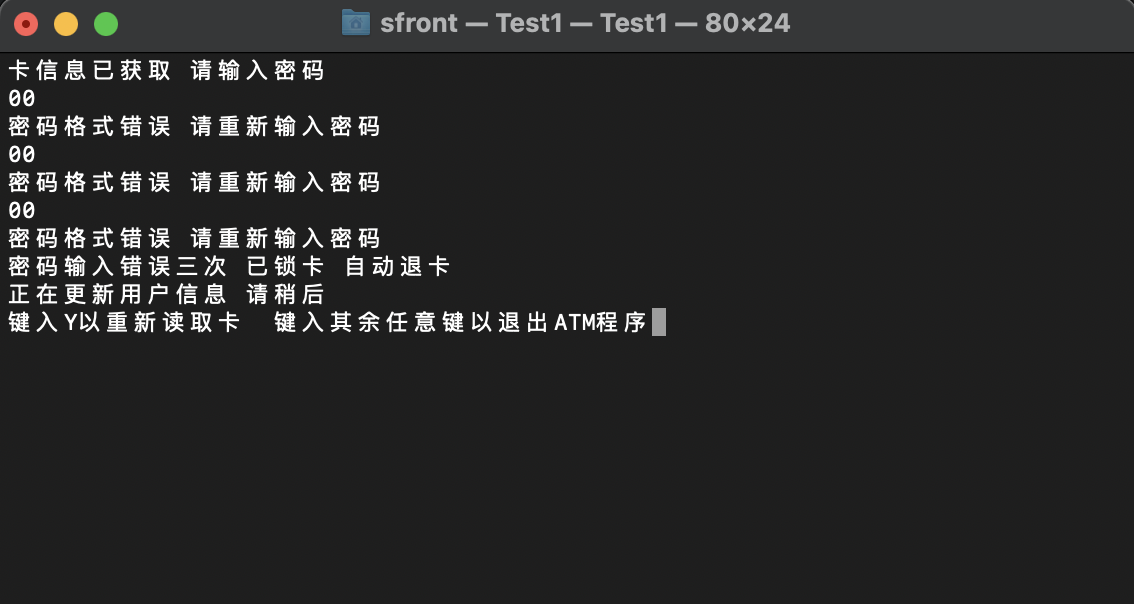


当两者均存在于同目录下，且格式正确，CardFile.csv文件中存储的账号在UserFile.csv能找到

则程序正常初始化，进入密码核验阶段



若输入密码错误将会提示再次输入，若输入错误次数超过三次，则会自动锁定账号

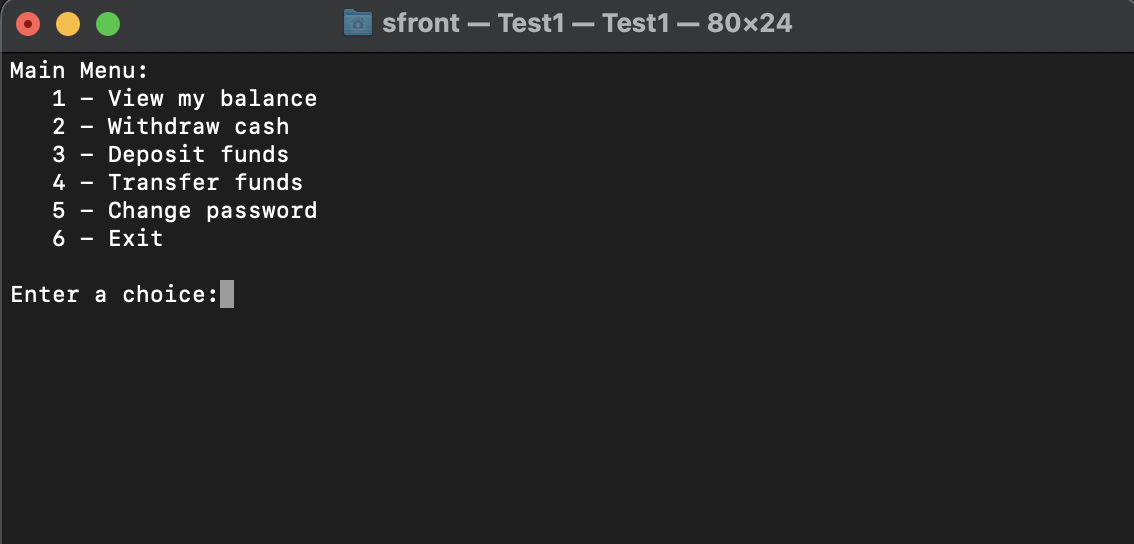


此时可查看UserFile.csv文件中该账号锁定状态为true，无法再登入



手动进行更改，将该账号锁定状态改为false则可正常输入密码

密码输入正确后将登入主程序

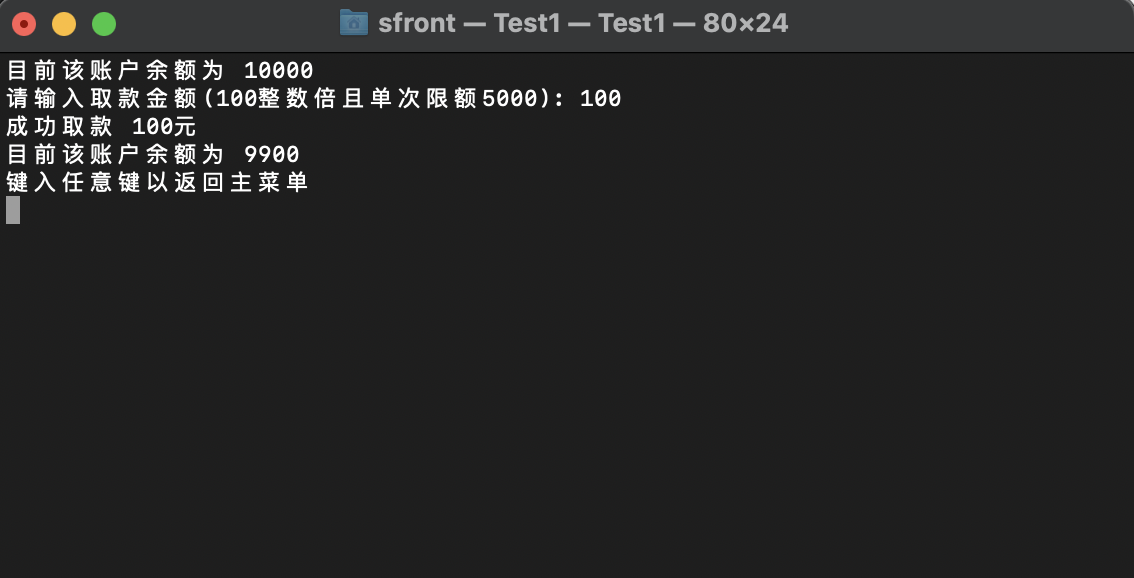


以下仅展示正确输入后结果，异常处理不再一一列出

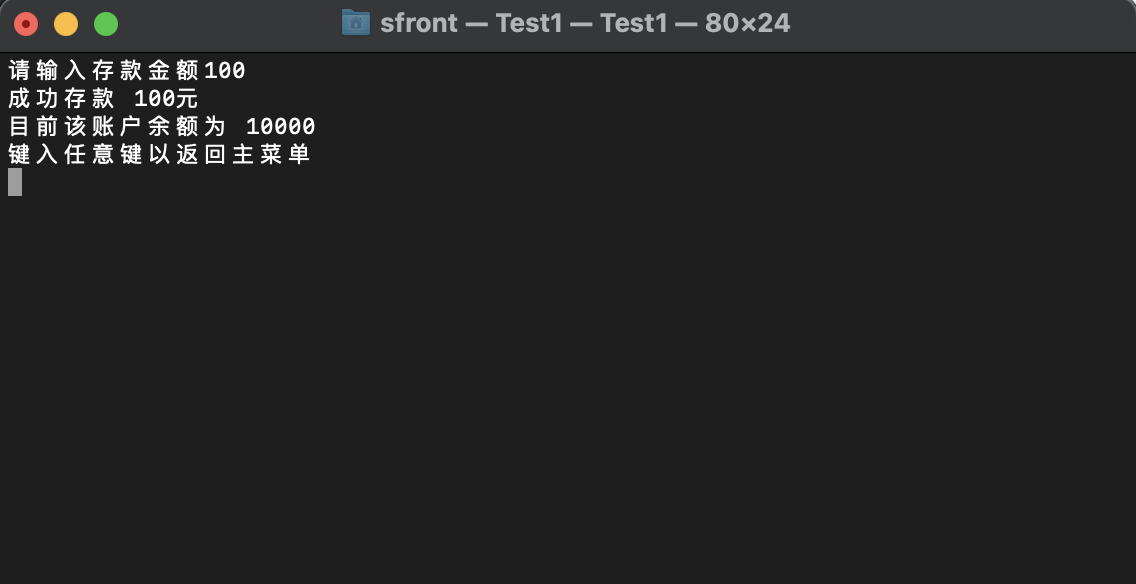
1. 查询余额



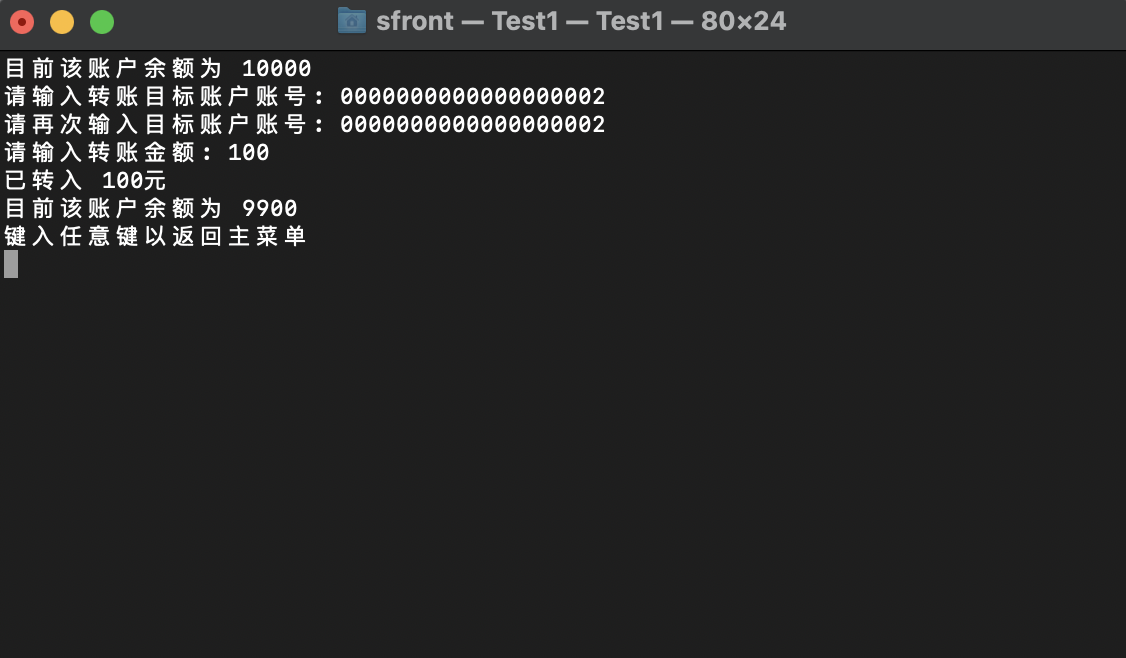
1. 取款



1. 存款



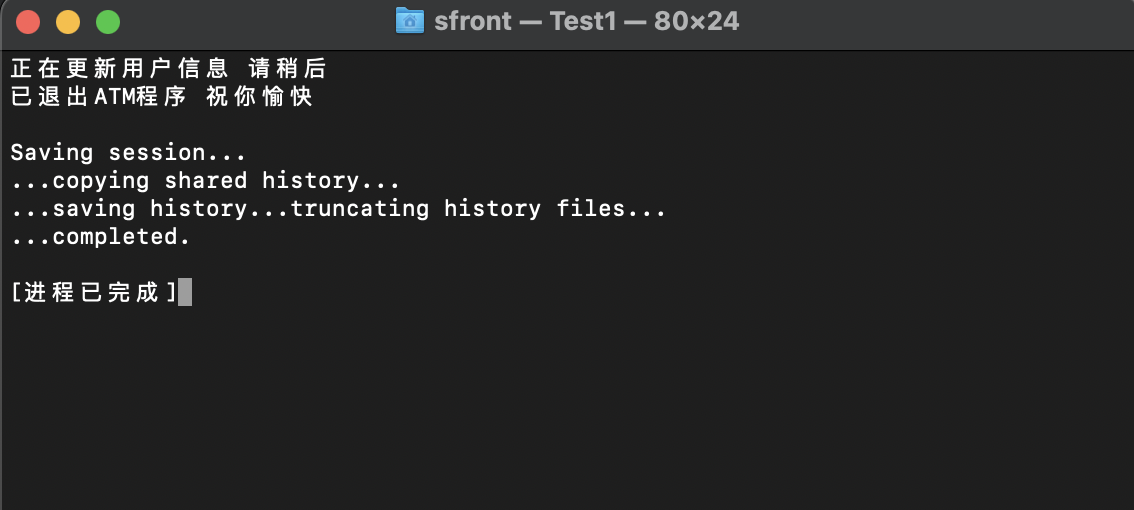
1. 转账



1. 修改密码



1. 退出



程序运行结束后可查看新生成的TransLog.csv, WithDrawLog.csv, DepositLog.csv, 此处不再展示

注: 演示时使用mac终端进行程序运行，最终上传可执行文件已修改为win端适用，运行效果与上述相同。

4. 总结

本项目重点在于

1. 程序进行读写csv文件，将csv文件内容与临时内存信息互通转换
2. 实现ATM各种主要服务
3. 各种异常处理的解决方案
4. 优化用户界面和操作方式

针对csv文件读写，参考相关网络文献[1]，最终整合修改创新出一套自定义简介的转换方式，相关内容可参考Initiate.cpp文件。

实现ATM的各种主要服务，主要借助User，SaveInfo等程序，由MainSystem程序提供，其中TimeInfo程序借助ctime.h头文件获取时间日期信息[2]，自定义不同函数进行时间日期信息的规范输出。

异常处理主要使用try-throw-catch语句进行解决，在后期修复BUG时也起到重要作用。

为优化用户界面，主要运用自定义函数Delay实现程序延迟运行[4]，以及system(“cls”)/system(“clear”) 进行清屏处理[3]，进一步优化用户界面，在程序运行时会有选择性键入的步骤，通过获取键入按键进行选择性处理，便捷用户操作方式。

5. 参考文献

列出在完成Project过程中查阅的中英文资料，并在报告正文中按照学术规范予以正确的标注。不得列入正文中没有引用的参考文献。

参考文献：

1. C++ Open 打开文件<http://c.biancheng.net/view/294.html>
2. C++日期&时间<https://www.runoob.com/cplusplus/cpp-date-time.html>
3. 深入理解system(“cls”)两种情况

<https://blog.csdn.net/weixin_49486457/article/details/124760735>

1. C++延迟处理的几种方法<https://www.cnblogs.com/findumars/p/8732305.html>