

模拟校园卡

一、课题内容和要求

模拟校园卡系统是模拟校园卡的消费记录查询系统，实现包括登录、计费、挂失、统计等功能，能够处理校园卡操作的软件平台，包括学生和管理员的登录验证、账户管理、上机记录的保存与查询，以及余额和费用的统计。应包含以下要求：

- (1) 登录模块：同学根据自己设定的密码登录，三次错误则冻结账户；
- (2) 管理员：管理员以密码登录系统，提供以下功能：充值、挂失、重置学生密码；
- (3) 学生：登录系统后，可以查询卡中余额，上机时间和地点，可以统计自己上机时间，上机费用等；
- (4) 每个学生 对应一个文本文件，比如：B23010101.txt 存放上机记录、卡号或学号、开始时间、结束时间、上机用时、是否上机、上机地点等，另外一个文件里包含所有学生的校园卡状态记录，上机记录表永久保存，用于事后查询和统计；
- (5) 提供菜单选择项，如角色选择、时间选择、功能选择等。

二、需求分析

模拟校园卡系统的功能框架图如图 1 所示。

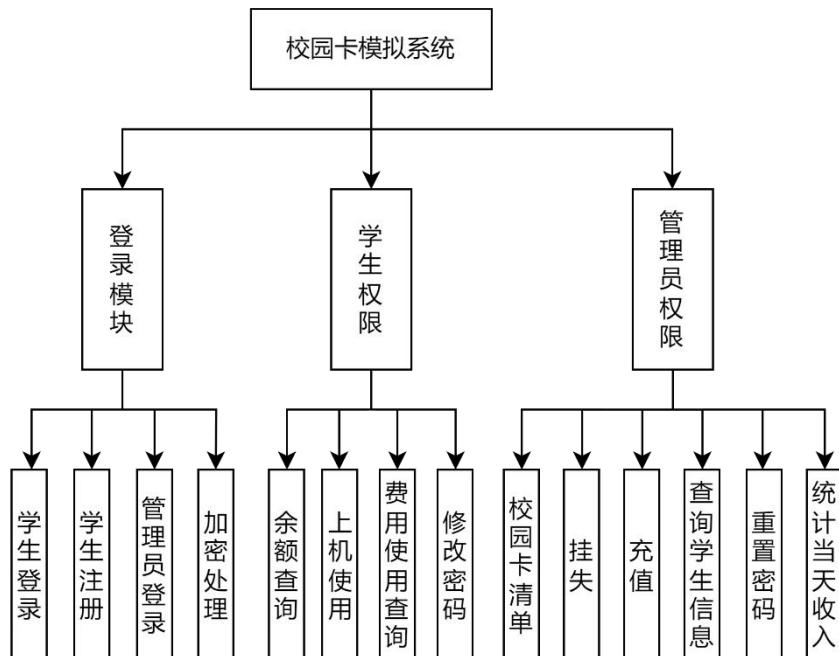


图 1 功能框架图

- (1) 提供可操作的主菜单：输出菜单，用于显示以从文件中加载的学生信息和若干个

可选的功能选项。根据客户输入的选项来运行不同的功能，运行不同的函数。

- (2) 提供学生上机充值信息：学生可以查询自己上机充值详细情况，管理员可以通过后台统计看到所有注册学生详细情况。
- (3) 提供安全检验：只有注册的学生卡可以进行登录上机，未经注册的校园卡不能登录，若校园卡密码输入错误达到三次，冻结该校园卡，管理员可以看到所有学生的信息，并进行重置学生密码、学生卡充值、注册、销卡、排序等操作。
- (4) 提供安全输入密码：密码进行输入时均以*显示，密码内容仍为键盘输入的内容，防止密码泄露。
- (5) 提供密码加密解密处理：将密码值后移 3 位进行加密处理，增强登录系统的安全性，防止密码被盗。
- (6) 提供统计当天收入：通过充值，上机的全部记录，管理员可查看当天的收入情况。
- (7) 提供用户输入的正确性检验：卡号为 8 位数字，密码为 6-12 位字母或数字，保证用户输入内容的正确性。

三、概要设计

1 主要存储结构

```
struct Card {  
    string ID;           // 校园卡号码，8 位数字  
    string name;         // 姓名  
    string password;     // 密码，6-12 位数字或字母  
    double recharge;     // 充值数额  
    double remain;       // 余额  
    int state;           // 状态，0 表示正常，1 表示挂失，2 表示冻结  
    Card* next;          // 下一个节点  
};  
  
struct Record {  
    string ID;           // 校园卡号码  
    string date;          // 日期  
    string time;          // 时间  
    string address;        // 地点  
    double money;          // 金额
```

```

int duration;           // 上机用时
char state;            // 状态, '1'表示上机中, '0'表示未上机
};

```

2 主要函数流程

(1) 学生注册函数(registerCar): 先利用 `findNode(Card* head, const string& id)` 判断是否找到重复节点, 再利用 `isValidID`, `isValidPassword` 检查卡号密码格式, 规定卡号为 8 位数字, 密码为 6-12 位字母或数字。学生注册函数流程图如图 2 所示。

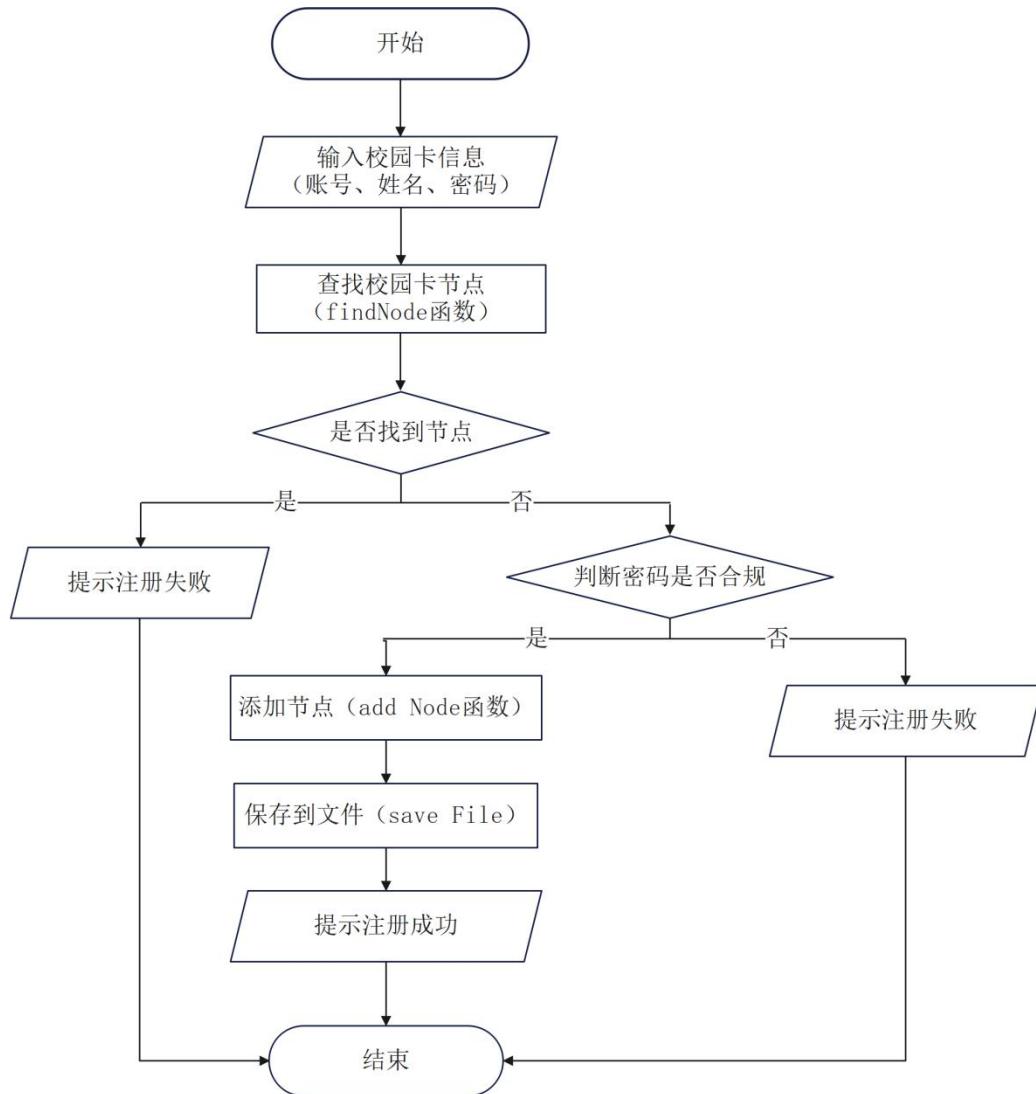


图 2 学生注册流程图

(2) 学生登录函数: 利用 `loginCard(Card** head)` 进行该操作, 在该函数中利用一个 `while` 循环将输入卡号密码次数限定在 3 次, 再利用一个 `if`, `else` 选择结构, 进行若密码正确进入学生菜单, 若密码错误冻结操作, `state` 值变为 2 表示冻结。学生登录输入密码时均以 * 显示, 防止密码泄露以保证密码安全性, 同时进行加密处理。学生登录函数流程图如

图 3 所示。

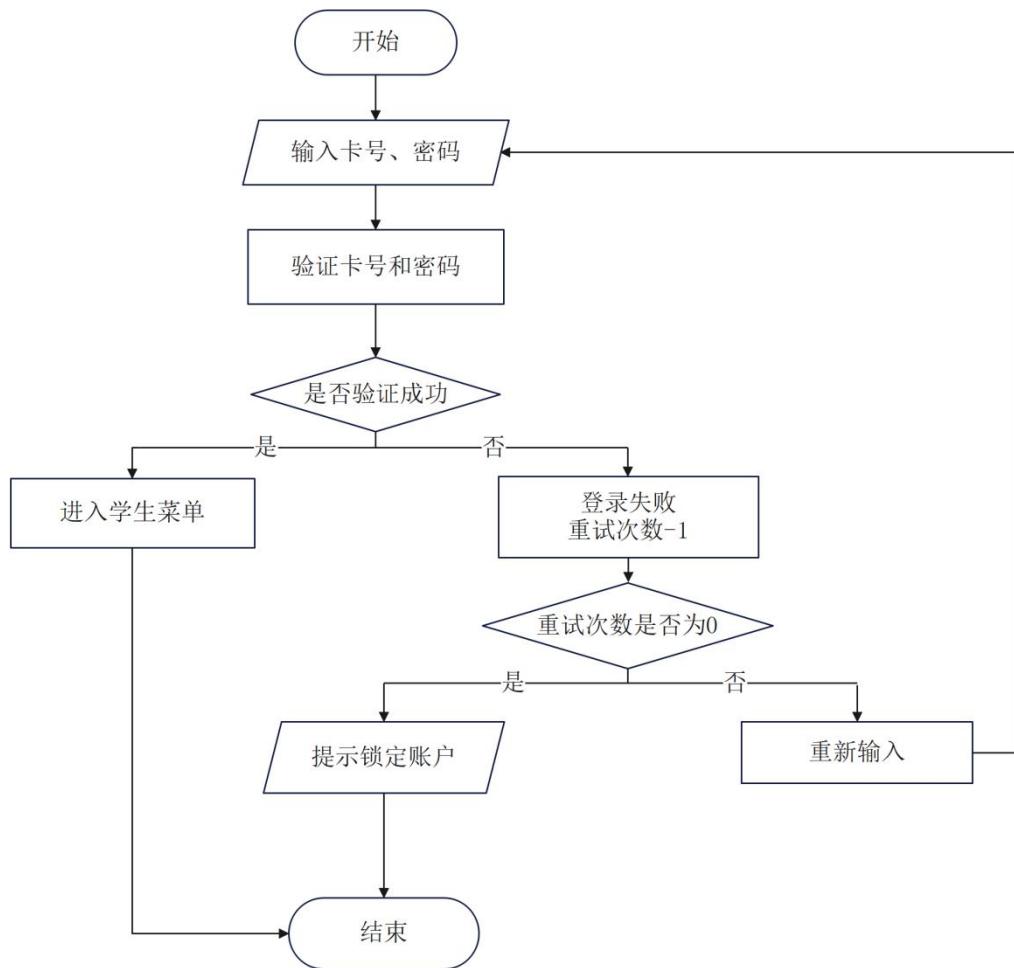


图 3 学生登录函数流程图

(3) 管理员登录函数(loginAdmin): 输入账号密码，在该函数中用一个 if 语句判断卡号与密码是否正确，正确则显示菜单，错误则提示登录失败。管理员登录输入密码时均以 * 显示，防止密码泄露以保证密码安全性，同时进行加密处理。本项目管理员账号为 zjh 密码为 zjh。管理员登录函数流程图如图 4 所示。

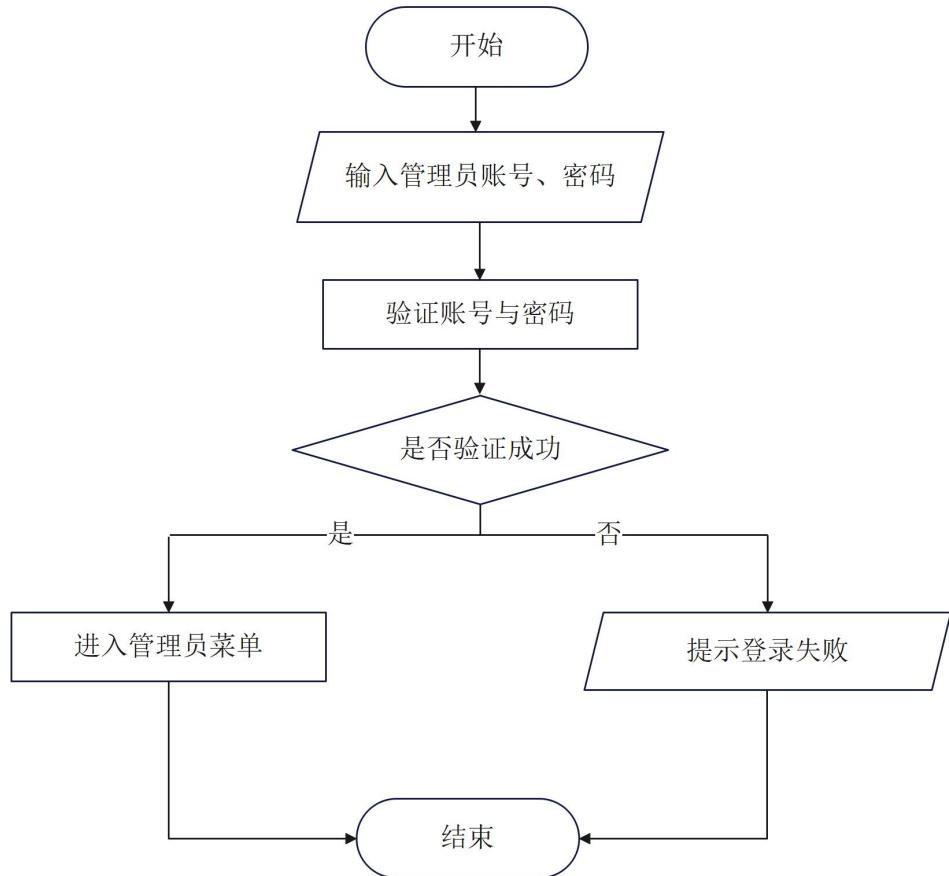


图 4 管理员登录函数流程图

(4) 加密解密函数: `encode` 函数用于对学生和管理员输入的密码进行加密处理, 采用将字符后移三位的加密方式, 如果移动后超出范围, 采用取余处理进行加密, 解密 `decode` 函数与之相反, 加密函数流程图如图 5 所示。

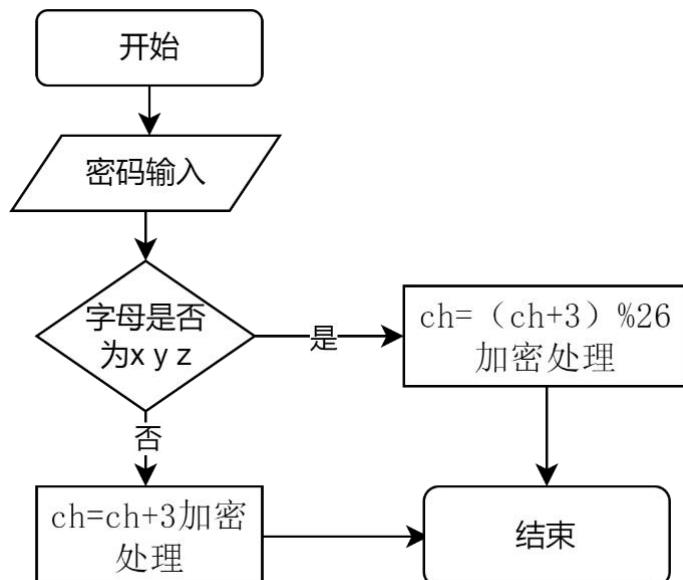


图 5 加密函数流程图

(5) 学生上机函数(expenseCard): 学生输入上机的耗时和地点，系统根据固定单价计算消费金额，并检查余额是否足够。如果余额充足，则扣除相应金额、保存余额更新记录并添加上机历史；否则提示余额不足并终止操作。学生上机函数流程图如图 6 所示。

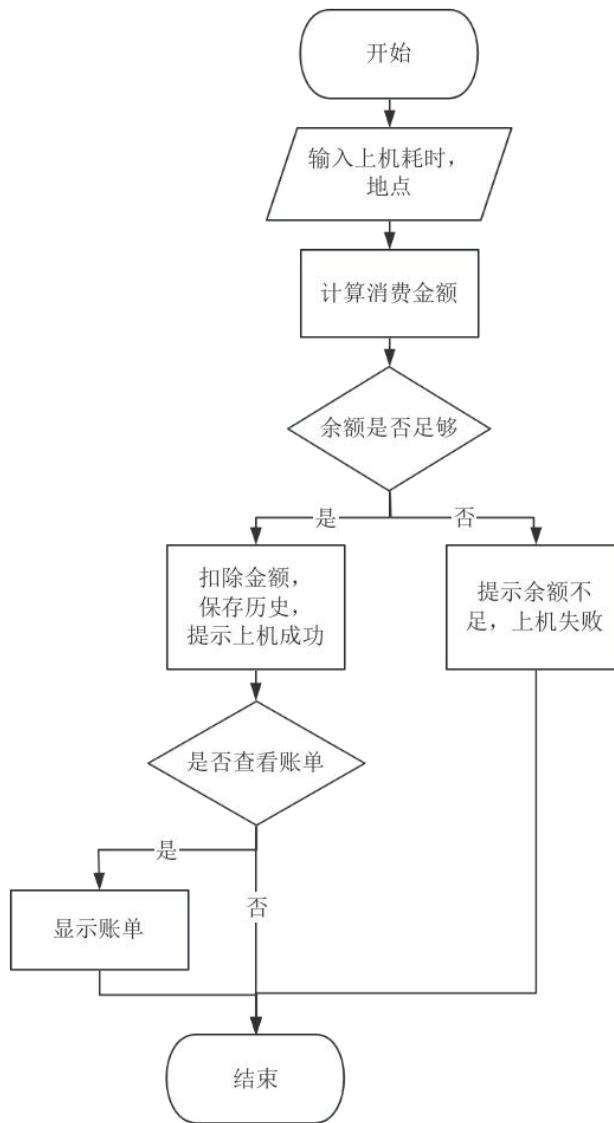


图 6 学生上机函数流程图

(6) 学生查询函数(searchCard): 学生登录后可以查询校园卡的详细信息（如卡号、姓名、充值金额、余额等），并选择查看充值记录或上机记录。查询历史记录时按类型筛选和展示，同时支持多次查询直到用户选择返回。学生查询函数流程图如图 7 所示。

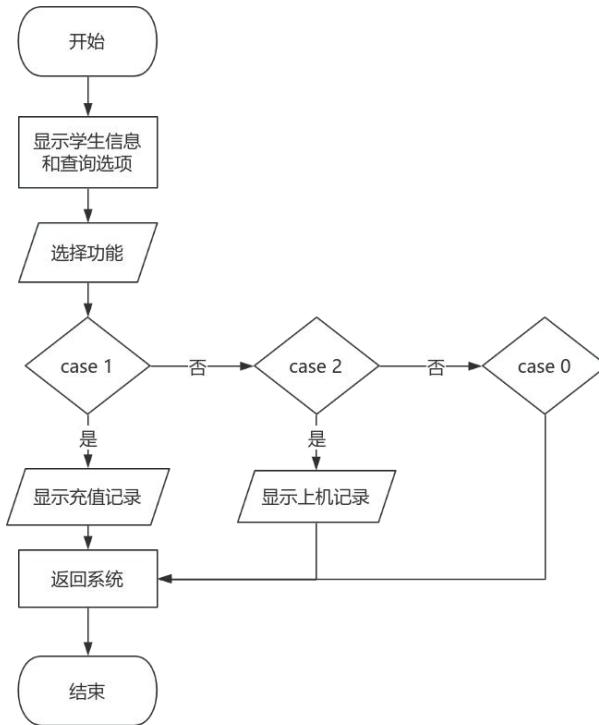


图 7 学生查询函数流程图

(7) 校园卡管理函数(manageMenu): 管理员登录成功后进入校园卡管理菜单, 可执行多种管理操作, 包括查看校园卡清单、挂失校园卡、查询校园卡、充值校园卡、重置密码等。管理员可以反复选择功能, 直到选择退出返回上级菜单。校园卡管理函数如图 8 所示。

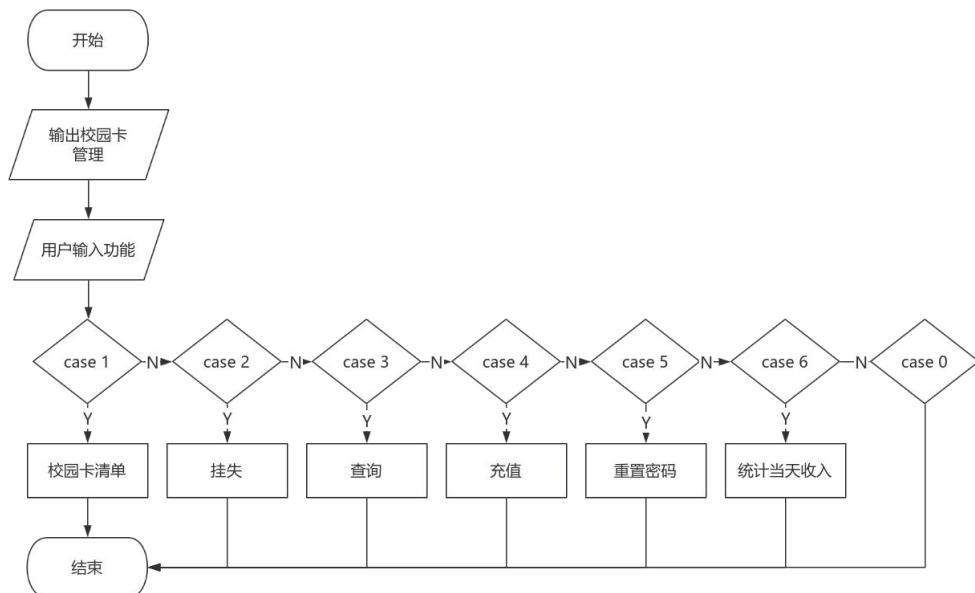


图 8 校园卡管理函数流程图

(8) 校园卡挂失函数(cancelCard): 在该函数内再利用 findNode 进行校园卡节点的寻

找是否存在此卡，再利用一个嵌套的 if, else 语句进行挂失操作，若挂失成功，校园卡状态为挂失，state=1。校园卡挂失函数流程图如图 9 所示。

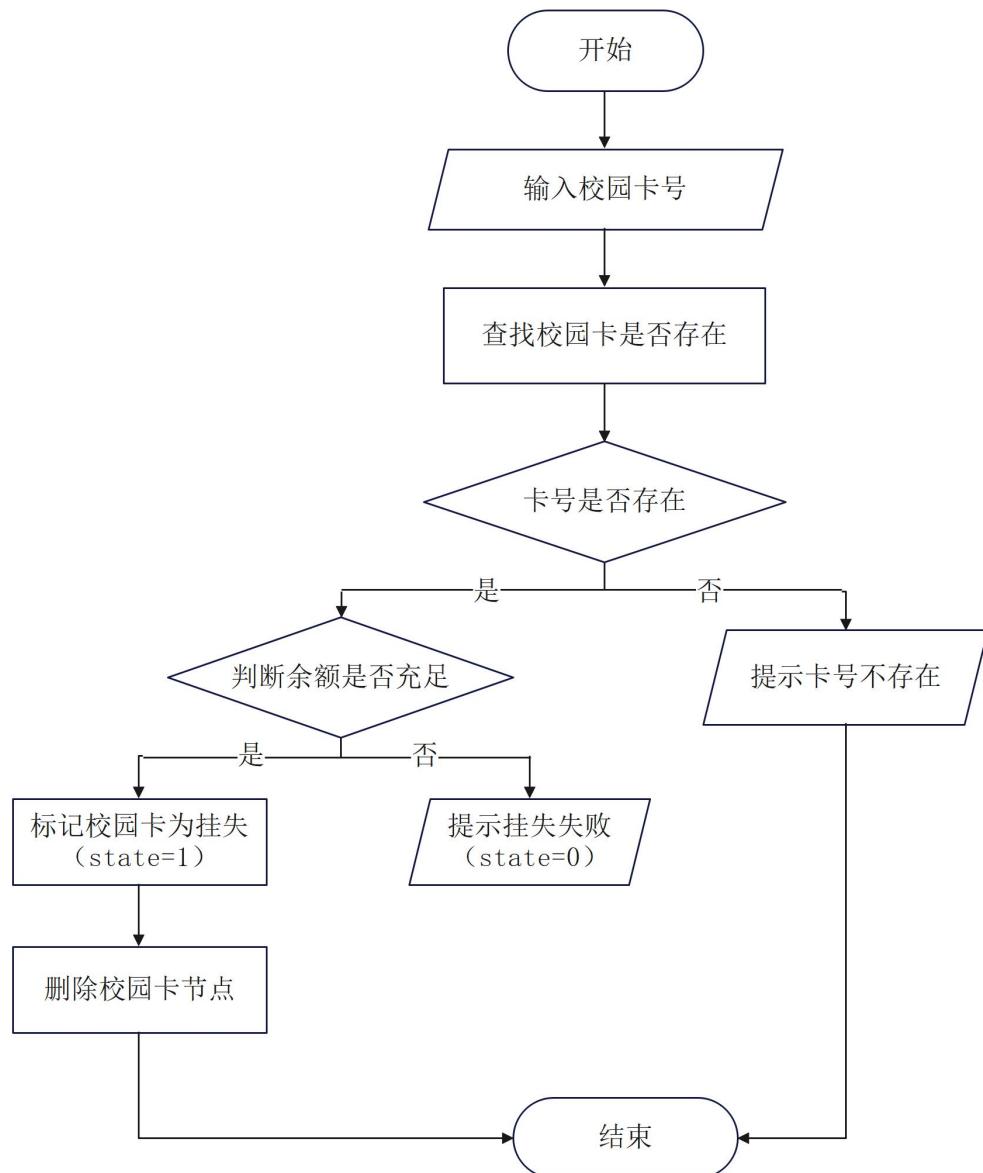


图 9 校园卡挂失函数流程图

(9) 管理员查询函数(findCard): 管理员输入校园卡号，系统查找对应校园卡信息。如果找到，则显示卡的详细信息（如卡号、姓名、余额等），并提供选项查询充值记录或上机记录；如果未找到，则提示卡号不存在。管理员查询函数流程图如图 10 所示。

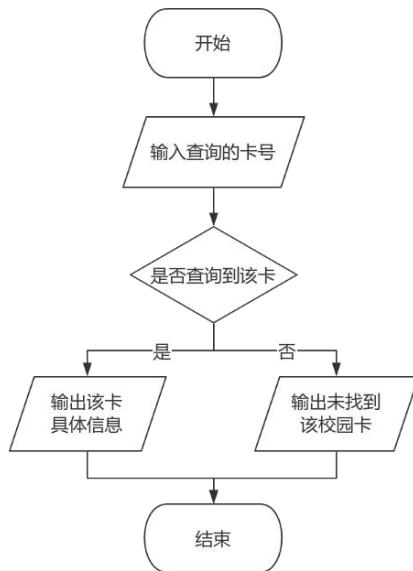


图 10 管理员查询函数流程图

(10) 余额充值函数(depositCard): 管理员为学生校园卡充值，输入卡号和充值金额后更新余额，同时保存充值记录以便查询。如果卡号不存在则提示错误；充值成功后更新校园卡信息并可以让用户选择是否查看余额变化记录。余额充值函数流程图如图 12 所示。

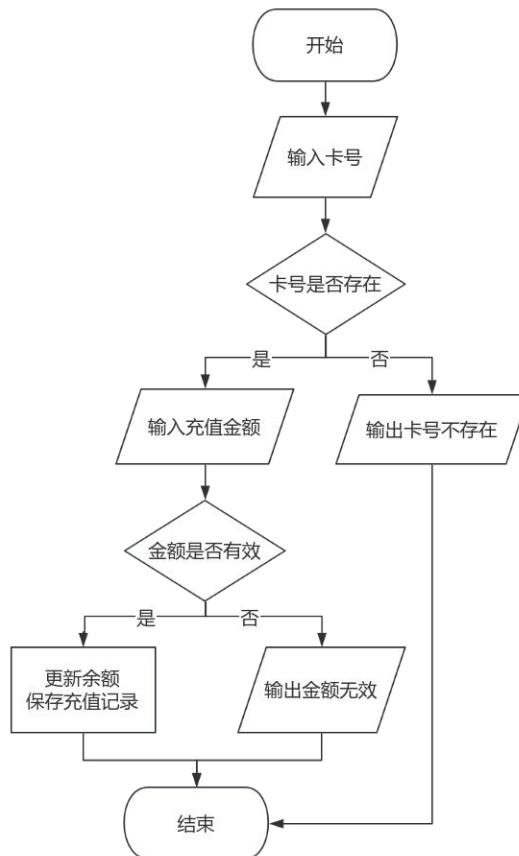


图 11 余额充值函数流程图

(11) 重置密码函数(resetPassword): 管理员输入校园卡号后, 将密码重置为默认初始值(123456), 并保存更新后的校园卡信息, 校园卡状态变成正常。如果卡号不存在则提示错误; 重置成功后显示成功提示信息。重置密码函数流程图如图 13 所示。

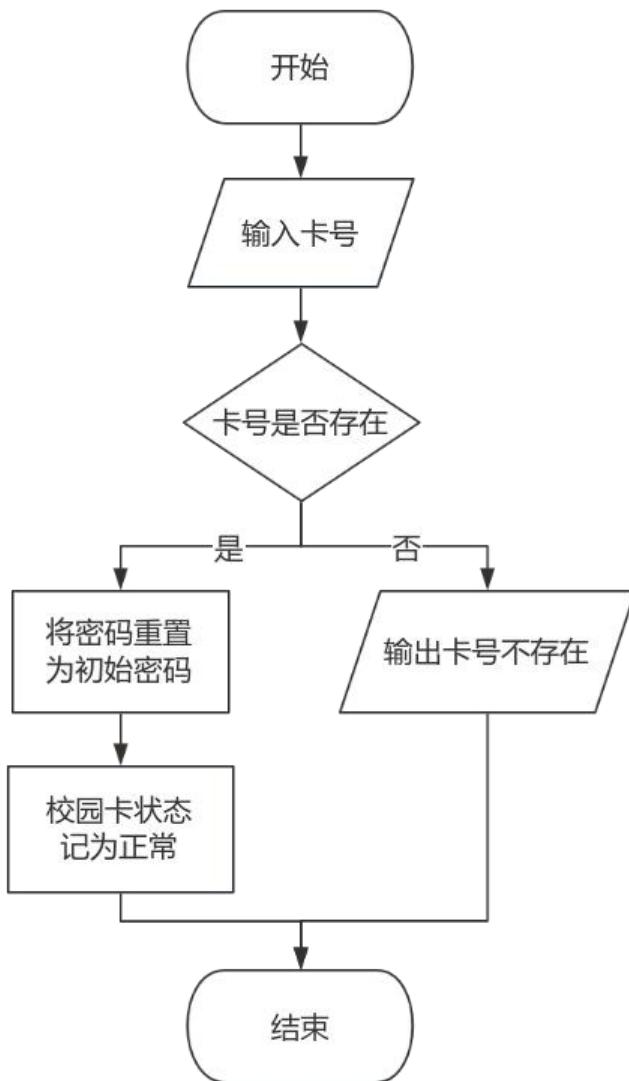


图 12 重置密码函数流程图

(12) 统计当天收入函数(calculateincome): 用于统计当天的收入信息, 通过遍历链表中的每张卡片, 读取对应卡片 ID 的文本文件, 检查文件中包含当天日期的充值记录, 并累加这些记录的金额, 最终输出当天的总收入。统计当天收入函数如图 14 所示

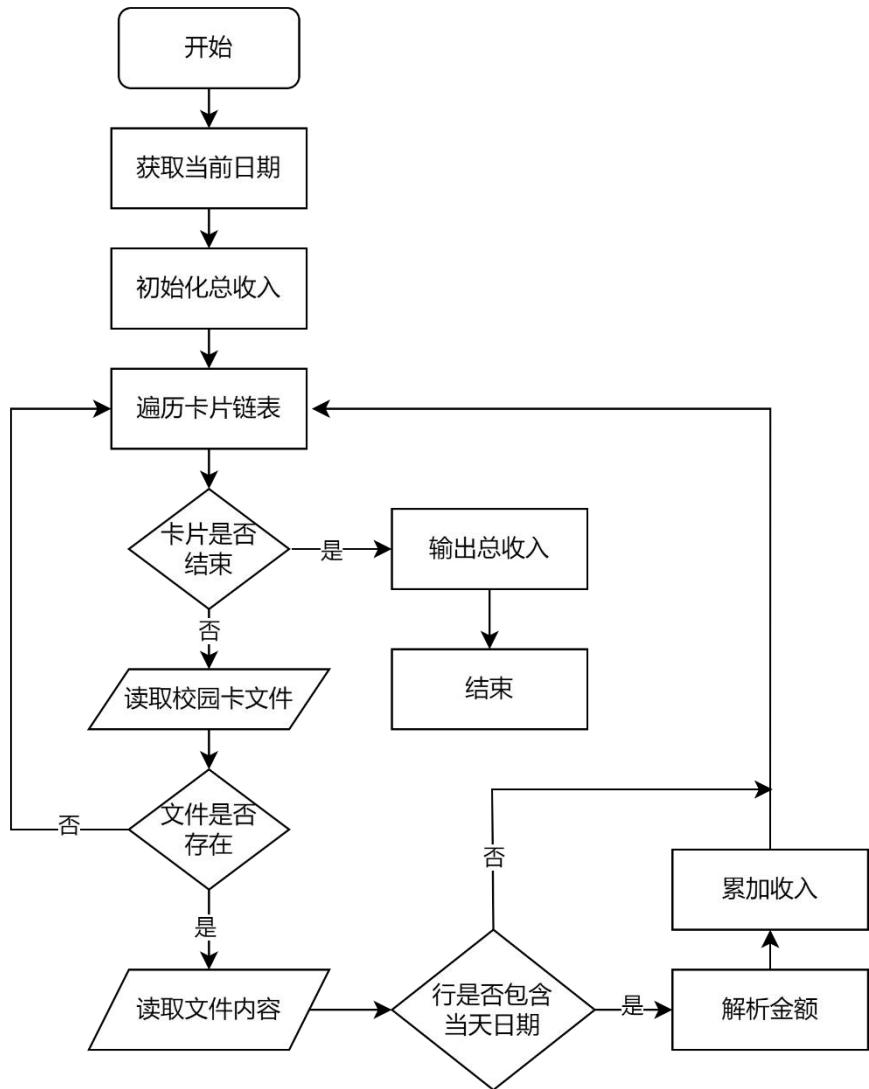


图 13 统计当天收入函数流程图

四、源程序代码见 cpp 文件

五、测试数据及其结果分析

运行后，程序的主菜单界面如图 14 所示，通过输入 1, 2, 3, 0 进行选择分别进入不同子菜单。



图 14 主菜单界面图

- (1) 学生登录，登录共有三次机会登录界面如图 15 所示，每次失败后会进行提醒。图 16 所示，三次失败后，该卡号会冻结如图 17 所示。

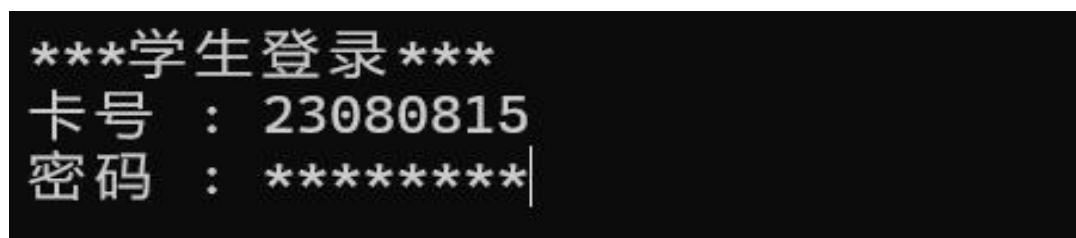


图 15 登录界面图

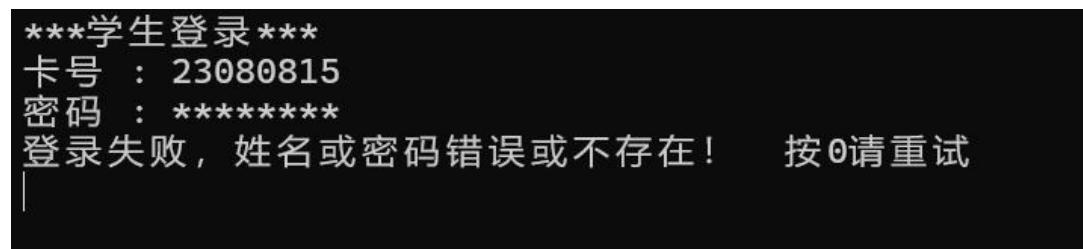


图 16 登录失败提示图



图 17 显示冻结图

- (2) 学生功能界面可进行上机，查询，修改密码操作，如图 18 所示

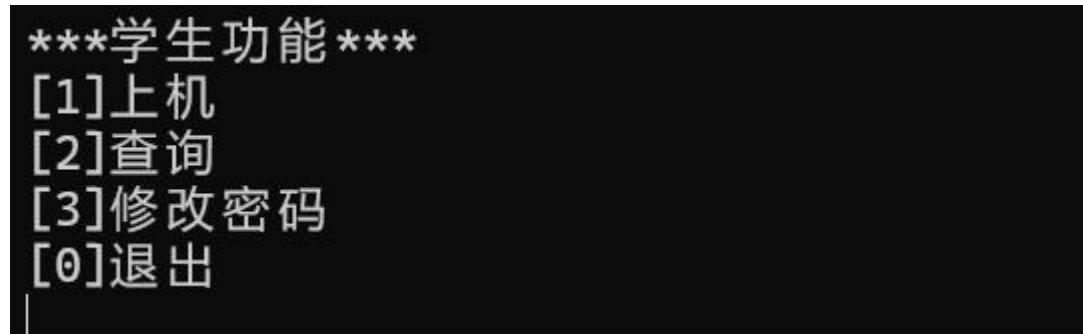


图 18 学生功能界面图

- (3) 学生上机，输入上机时间和地点，操作成功后可查看余额变化，如图 19 所示，若余额不足，则无法上机成功，如图 20 所示。

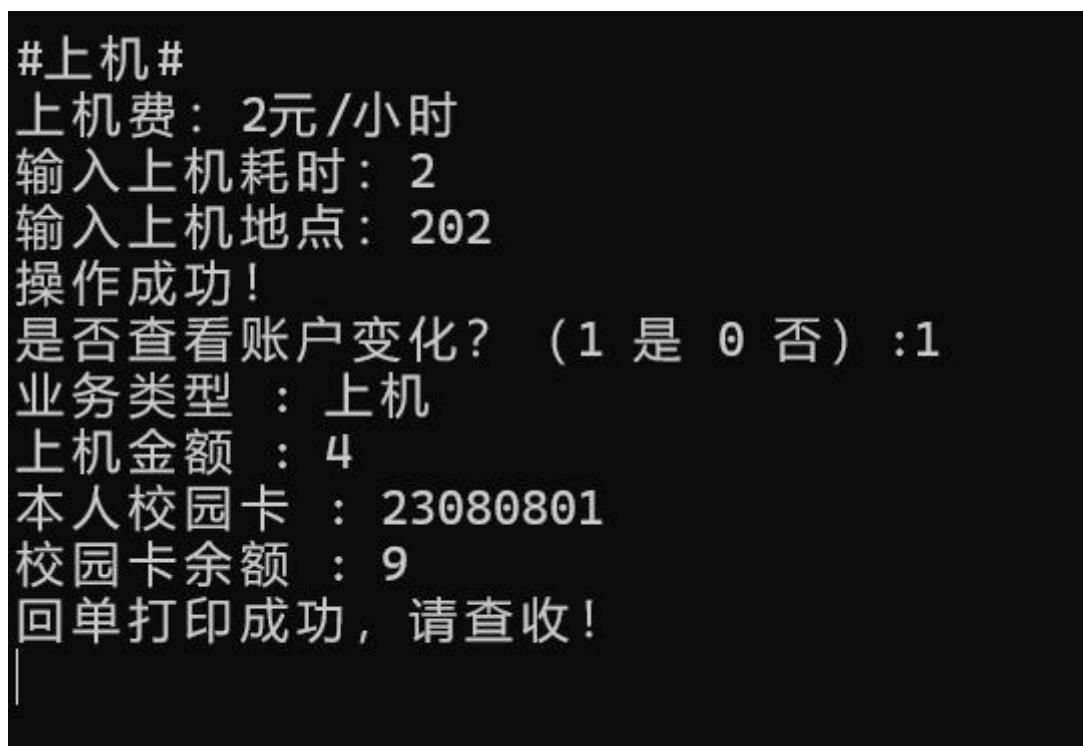


图 19 上机成功界面图

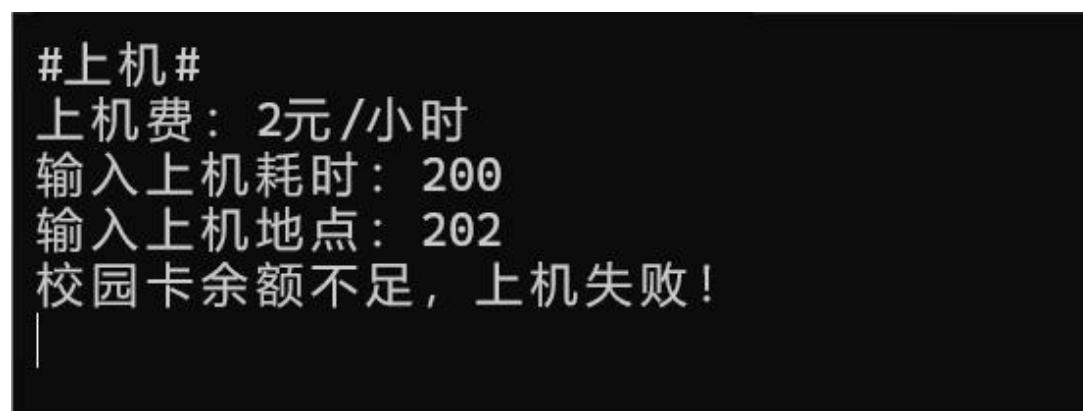


图 20 上机失败界面图

- (4) 学生查询，校园卡详情显示充值的金额和余额，可以选择充值记录如图 21 所

示和上机记录的详细查询，包括具体时间，地点，余额，使用时间，如图 22 所示。

```
***校园卡详情***  
卡号: 23080815  
姓名: zhengjh  
充值金额: 170  
余额: 146  
查询  
[1]充值记录  
[2]上机记录  
[0]返回  
1  
-----  
***记录***  
2024-12-19 14:19:38 充值 20 -  
-----
```

图 21 充值记录图

```
***校园卡详情***  
卡号: 23080815  
姓名: zhengjh  
充值金额: 170  
余额: 146  
查询  
[1]充值记录  
[2]上机记录  
[0]返回  
2  
-----  
***记录***  
2024-12-18 22:46:25 上机 12 103  
2024-12-19 09:47:14 上机 10 108  
2024-12-19 14:16:42 上机 2 108  
-----
```

图 22 上机记录图

- (5) 修改密码，学生可进行密码修改，密码修改需满足 6-12 位字母和数字的要求，如图 23 所示。

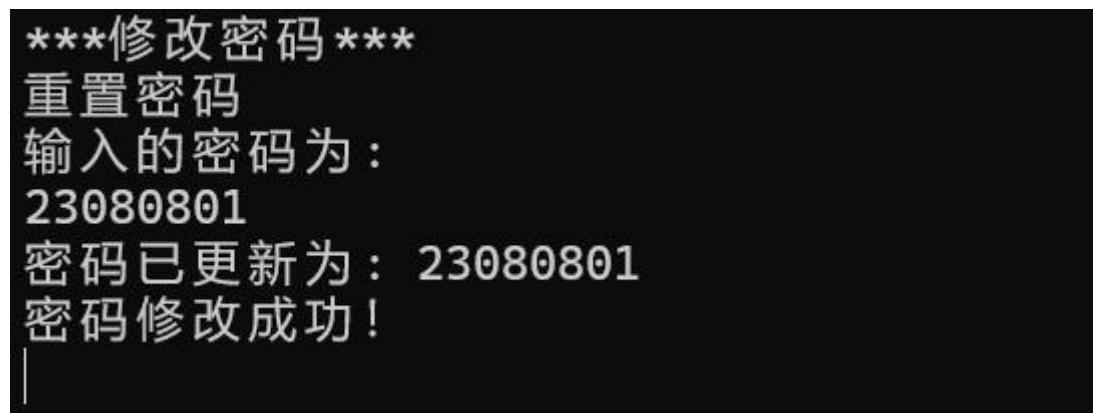


图 23 修改密码界面图

- (6) 学生注册，注册必须按照卡号为 8 位数字，密码为 6-12 位字母和数字的要求，如图 24 所示，若卡号格式不符合要求则会显示无效的卡号格式，如图 25 所示；若密码不符合格式要求则会显示无效的密码格式，如图 26 所示，则如果出现已注册过的卡号，则显示注册失败如图 27 所示。



图 24 注册成功界面图



图 25 卡号违规界面图

```
***注册 ***
***录入校园卡信息 ***
卡号(8位数字): 23080808
姓名: chenys
密码 (6-12位字母和数字) : 1
无效的密码格式!
```

图 26 密码违规界面图

```
***注册 ***
***录入校园卡信息 ***
卡号(8位数字): 23080801
姓名: wjx
密码 (6-12位字母和数字) : 23080801
注册失败
```

图 27 重复注册界面图

(7) 管理员登录界面, 账号:zjh 密码:zjh, 如图 28 所示

```
欢迎使用校园卡管理系统
Welcome to Campus Card Management System

[1]学生登录
[2]学生注册
[3]管理员登录
[0]退出系统
请选择： 3
-----
***欢迎进入管理员登录 ***
账号 :zjh
密码 : ***
```

图 28 管理员登录界面图

(8) 校园卡清单,会显示每个学生的卡号, 姓名, 余额, 状态, 如图 29 所示。

校园卡管理			
[1]校园卡清单			
[2]挂失			
[3]查询			
[4]充值			
[5]重置密码			
[6]统计当天收入			
[0]退出			
请选择： 1			
卡号	姓名	余额	状态
23080801	wangjx	21	正常
23080802	zhangy	82	正常
23080815	zhengjh	125	正常
23080816	liuzt	100	正常
23080814	zhaotw	50	正常
23080803	hongxy	0	正常

图 29 校园卡清单界面图

(9) 校园卡挂失，挂失后，校园卡状态会变成挂失状态，不可上机，如图 30 所示。

```
***挂失***
校园卡 :23080815
[-----]
校园卡挂失成功!
```

图 30 校园卡挂失界面图

(10) 校园卡查询，可以查询到卡号，姓名，密码，充值金额，余额，如图 31 所示。

```
***查询***
校园卡号 :23080801
卡号 23080801
姓名 wangjx
密码 23080801
充值金额 21
余额 9
```

图 31 校园卡查询界面图

(11) 校园卡充值,选择充值的卡号，并可查看余额变化，如图 32 所示。

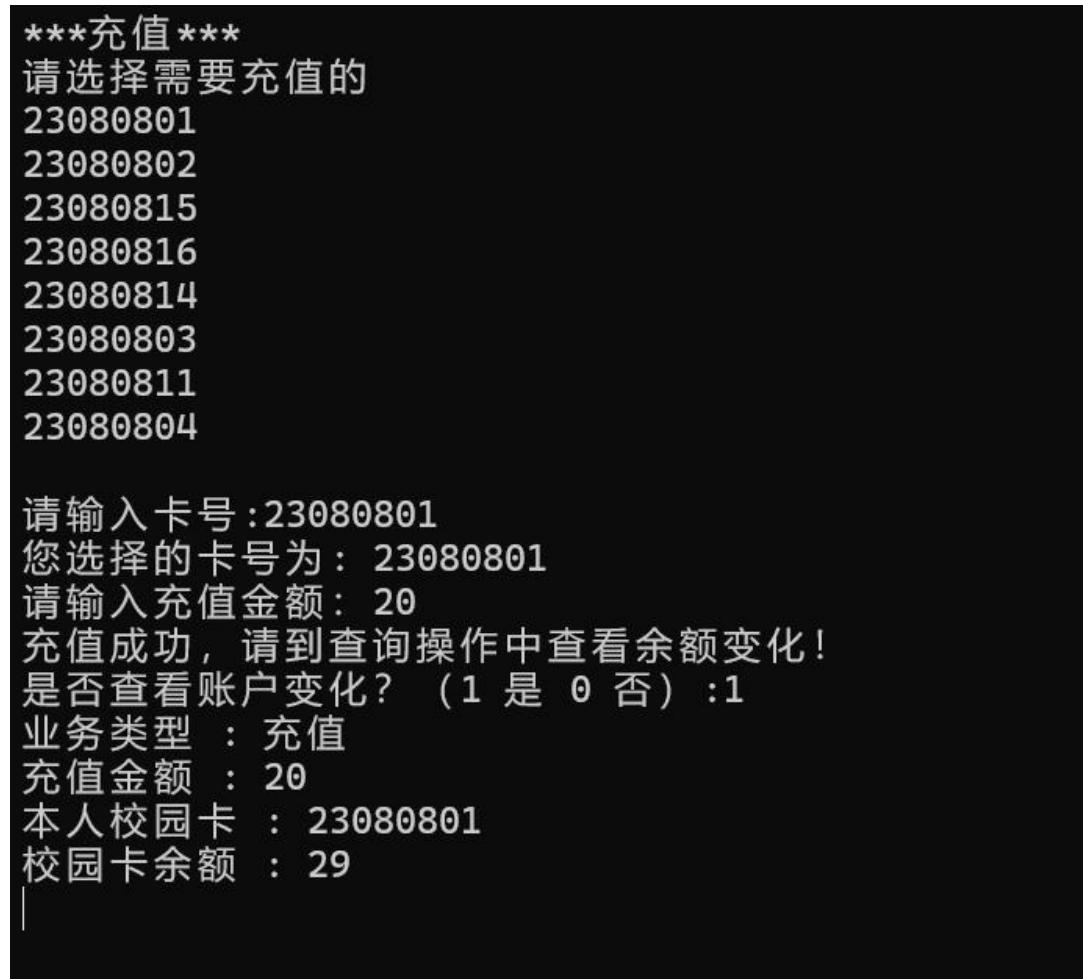


图 32 校园卡充值界面图

(12) 重置密码，密码会被重置为 123456。如图 33 所示。

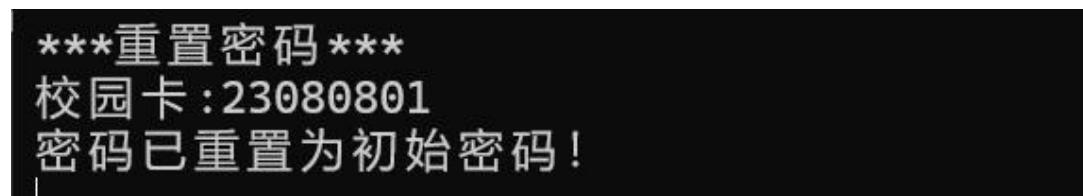


图 33 重置密码界面图

(13) 统计当天收入，可以查询到当天全部的充值收入，如图 34 所示。



图 34 统计当天收入界面图

(14) 退出系统，谢谢使用，请保管好您的校园卡，如图 35 所示。



图 35 退出系统界面图

结果分析如下。

(1) 学生登录：学生通过输入卡号和密码登录系统，系统验证卡号和密码是否匹配。登录成功后进入学生菜单，可选择上机、查询记录或修改密码。登录时输入错误密码最多允许 3 次，超出后锁定账户并退出。

(2) 学生注册：通过“学生注册”功能，学生可以录入校园卡的基本信息（如卡号、姓名、密码等）。注册完成后，校园卡会被记录在系统中，学生即可使用该卡完成上机操作，并查询个人记录（包括充值记录和上机记录）。如果输入的卡号已被注册，系统提示“注册失败，存在相同校园卡”。

(3) 学生上机：学生可以输入上机耗时和地点，系统根据耗时计算上机费用（按固定单价）。若余额充足，扣除相应金额并记录操作历史；若余额不足，则提示失败，操作终止。操作成功后可选择查看账户余额变化。

(4) 查询记录：学生可以查询个人校园卡的详细信息（如卡号、姓名、余额等），并选择查看充值记录或上机记录。记录会按时间顺序展示，支持多次查询直到返回。如果尝试查询不存在的记录，提示“没有记录”。

(5) 修改密码：学生可以选择修改密码，通过输入旧密码验证后设置新密码，确保密码安全性。密码输入以 * 显示，防止泄露。如果旧密码输入错误，提示“旧密码错误，修改失败”。

(6) 管理员登录：管理员通过输入账号和密码登录系统，系统验证管理员账号（默认账号 zjh，密码 zjh）。登录成功后进入校园卡管理菜单。密码输入以 * 显示，防止泄露。如果登录失败，提示“登录失败，账号或密码错误”。

(7) 管理员查询：管理员通过输入校园卡号，系统查找并显示对应校园卡的详细信息（如卡号、姓名、余额等）。管理员还可选择查看该卡的充值记录或上机记录。如果输入的卡号未注册，提示“未找到卡号”，并返回菜单。

(8) 校园卡清单：管理员可以查看所有已注册校园卡的信息，并选择排序方式：按卡号排序，按姓名排序，按余额排序。排序后可直接查看完整清单。

(9) 校园卡挂失：管理员输入校园卡号，将卡标记为挂失并从系统中移除。如果输入的卡号未注册，提示“未找到卡号”。

(10) 修改密码：学生可以在登录后修改密码，输入旧密码验证成功后设置新密码。密码输入和存储均经过加密处理，避免明文泄露。若旧密码错误或新密码不符合规范，系统会提示相应错误信息。

(11) 重置密码：管理员可通过输入校园卡号，将密码直接重置为默认初始值 123456。如果卡号未注册，提示“未找到卡号”并返回上级菜单。

(12) 余额充值：管理员输入校园卡号和充值金额，为卡充值。系统会更新余额并记录充值历史。如果卡号未注册，提示“未找到卡号”；若充值金额无效，提示重新输入金额。充值完成后，可选择查看余额变化。

(13) 注册校园卡：学生可以通过输入卡号、姓名、密码注册校园卡。注册完成后，系统会将新卡信息保存至文件中。如果卡号重复，提示“注册失败，存在相同校园卡”。

六、调试过程中的问题

(1) 在“校园卡管理菜单”中调用“管理员充值”函数时出现了 depositCard 未找到标识符，通过加一行 void depositCard(pCard* head, pCard card);使得调用可以正确运行。

(2) 关于 localtime 和 sprintf 的安全警告：这些警告是因为您使用的 localtime 和 sprintf 函数在 Visual Studio 中被认为是不安全的，因为它们可能导致缓冲区溢出。Visual Studio 建议使用它们的安全版本，如 localtime_s 和 sprintf_s。您可以通过在代码中定义 _CRT_SECURE_NO_WARNINGS 来禁用这些警告，或者使用建议的安全版本。

(3) 没有与参数列表匹配的 重载函数 "getline" 实例：开头没有定义#include <iostream>#include <string>

(4) 没有对密码输入进行隐藏：使用 inputPassword0 函数 cout 输出提示信息 "密码：

", 进入一个 while 循环, 使用 _getch() 函数从键盘获取单个字符 _getch() 函数来自 <conio.h> 头文件, 调用 #include <conio.h>, 它允许我们读取用户的即时输入而不显示在屏幕上。如果用户按下的是回车键 ('\r'), 则在屏幕上打印一个换行符并退出循环。如果用户按下的是退格键 ('\b'), 并且 password 字符串不为空, 则在屏幕上打印两个退格符来模拟退格效果, 并将 password 字符串中最后一个字符删除。如果用户输入的是其他字符, 则在屏幕上打印一个星号 (*), 并将该字符添加到 password 字符串中。

(5) 读取历史记录时读取失败: 原因在于时间用 [] 包含, 所以修改读取文件的方式, 只读取时间, 不读取 [], 使用如下代码进行读取

```
// 去除时间字段的方括号  
size_t endPos = line.find("]", 1);  
if (endPos != string::npos) {  
    timenow = line.substr(1, endPos - 1); // 提取时间  
    line.erase(0, endPos + 1); // 删除方括号及时间部分  
}
```

(6) 姓名密码太长, 显示出现问题: 为了保证系统严谨性, 设置输入内容的正确性检验, 检查卡号格式, 8 位数字, 检查密码格式, 6-12 位数字或字母。

(7) 链表处理存在逻辑错误, 导致异常数据被写入或读取: 在添加或删除链表节点时, 如果没有正确处理链表的尾部, 可能会导致多余的数据被写入文件。在处理链表时, 确保正确地管理节点的链接, 避免在文件写入和读取过程中出现逻辑错误。

(8) 程序输出的校园卡清单中多出了一行异常数据 (0 正常): 这可能是由于程序在读取或解析文件数据时, 错误地将某些数据字段解释为其他字段, 导致状态字段没有被正确读取或显示。检查并确保 loadFile 函数正确地解析文件中的每个字段。在 loadFile 函数中, 使用 input.fail() 来检查数据是否正确读取。同时, 确保 saveFile 函数正确地写入数据, 以便 loadFile 能够正确解析。

七、课程总结

在本次校园卡系统程序设计实验中, 我们团队设计并实现了一个校园卡管理系统, 包含注册、充值、消费、查询等功能模块。通过这次实验, 我们不仅提高了对 C++ 语言的应用能力, 也深刻体会到了编程的严谨性和团队合作的重要性。

最初, 我们面对需求时感到迷茫。如何用 C++ 实现链表管理用户信息? 如何通过文件读写保存用户数据? 这些问题对我们来说是全新的挑战。为此, 我们首先学习了 C++ 的核

心知识，包括类与对象、动态内存分配、文件流等，然后逐步将这些知识应用到项目中。

在程序设计中，我们使用链表实现了校园卡的动态管理，并通过封装类来组织功能模块，便于代码的扩展和维护。比如，我们用 Card 类表示校园卡，用链表存储用户数据，用文件流实现数据持久化，确保系统能够在多次运行中保持数据一致性。这种实践让我们深入理解了 C++ 的面向对象思想及其在实际项目中的应用。

实验过程中，我们深刻认识到编程的严谨性和细节的重要性。比如，在注册功能中，我们增加了对卡号和密码格式的校验，避免用户输入不合法的数据；在链表操作中，处理了空链表、重复卡号等边界情况；在充值功能中，我们格式化金额，确保结果精确到两位小数。这些改进极大提高了程序的健壮性。

此外，我们在调试中不断优化程序。例如，在消费功能中，我们严格判断余额是否充足，并将每一次消费记录写入日志文件，方便后续查询。在异常输入的处理上，我们通过清空缓冲区和异常捕获机制，确保程序在错误输入下依然能稳定运行。

这次校园卡系统设计实验让我们从迷茫到掌握了 C++ 在实际开发中的应用，不仅收获了技术上的进步，更感受到了团队合作的力量。实验中，C++ 语言的灵活性和逻辑性让我们深刻体会到编程的乐趣，而团队间的配合更让我们在解决每一个问题时都充满成就感。感谢学校和老师提供的支持，这次实验为我们以后的学习和实践打下了坚实基础！