

1 实验目的

通过迭代式开发，深入掌握 C 语言的文件、链表、结构体、动态内存管理等技术，开发实现一个计费管理软件。同时在开发中使用了 C++，Qt 编写图形页面，对面向对象，GUI，有了初步认识。

2 系统功能与描述

2.1 总体结构

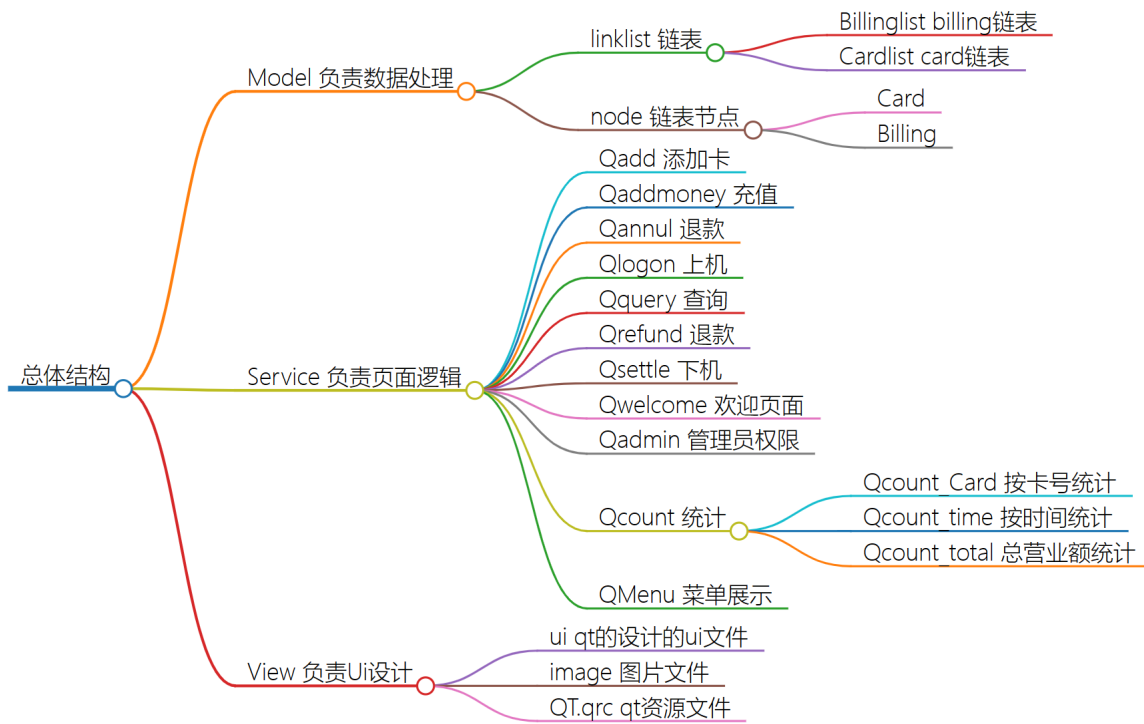


图 1: 总体结构

如图 1为该程序总体结构，分为 Model，Service，View 三层结构

2.2 基础功能介绍

2.2.1 菜单

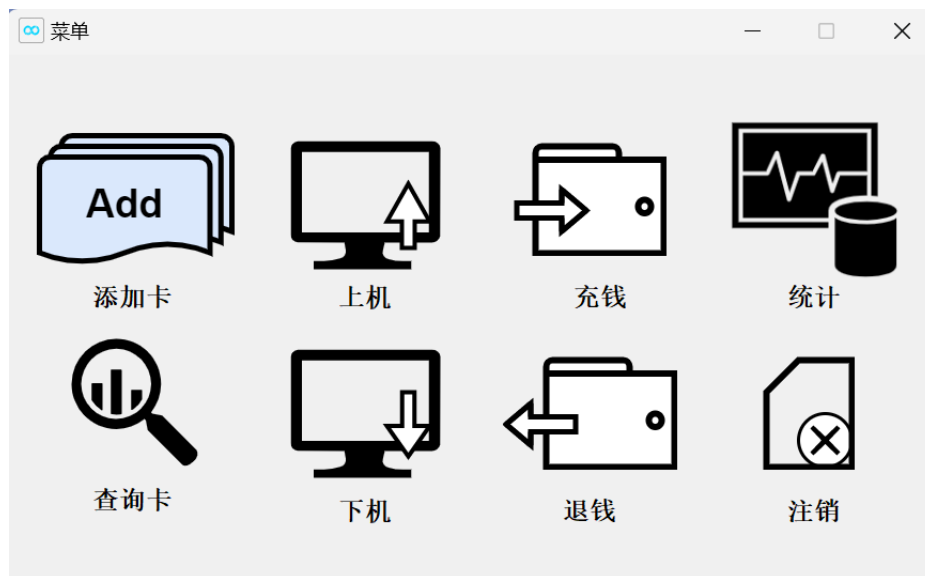


图 2: 主菜单

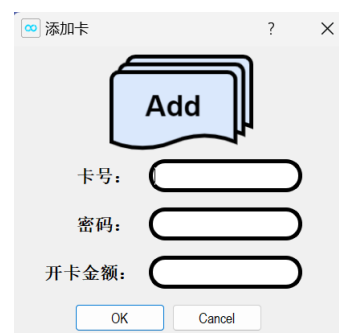
程序运行后进入主页面，如图 2，用户可以根据自己需求点击相应按钮进入该功能的页面。目前提供的功能有添加卡，上机，下机，查询卡，充钱，退钱，统计，注销六个功能

2.2.2 添加卡

功能介绍 如右图，用户根据指引输入卡号，密码，开卡金额后点击 OK 就可以添加卡。

返回结果

1. 成功，则返回 3a
2. 卡号与已申请卡号重复，则返回 3b
3. 输入开卡金额非正，则返回 3c
4. 输入卡号或密码过长，则返回 3d。



(a) 成功页面

(b) 卡号重复页面

(c) 金额非正页面

(d) 输入过长页面

2.2.3 查询卡



图 4: 查询卡

功能介绍 如 4，用户根据指引输入卡号，通过切换精确与模糊查找找到自己的结果

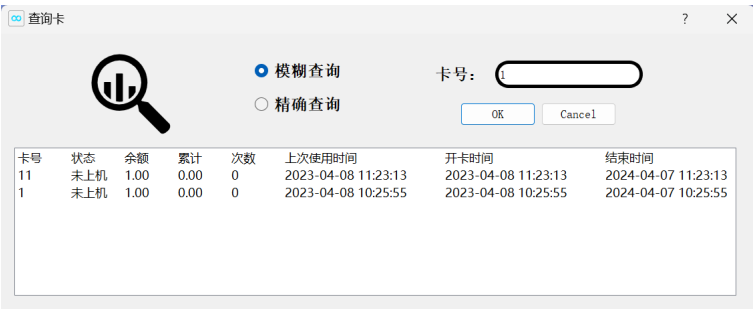


图 5: 模糊查找

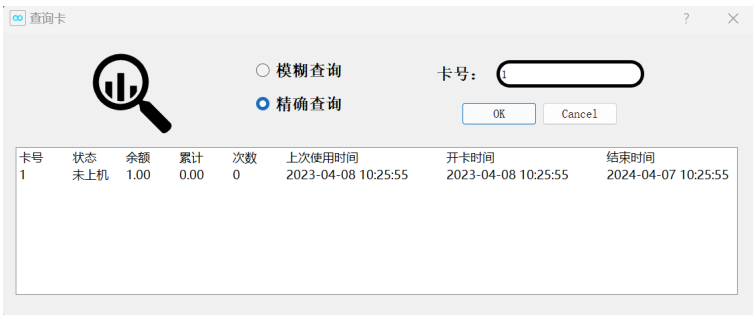


图 6: 精确查找

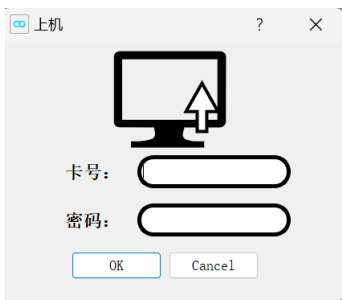
返回结果

- 1. 模糊查找如图 5，输入较短卡号序列就可以返回所有包含该卡号序列所有卡的信息
- 2. 精确查找如图 6，输入精确卡号序列才可以返回该卡的信息

2.2.4 上机

功能介绍 如右图，用户根据指引输入卡号，密码后点击 OK 就可以上机，该卡会标记为已上机
返回结果

- 1. 成功，则返回 7a
- 2. 卡号与密码不匹配页面，则返回 7b
- 3. 已上机再重复上机，则返回 7c
- 4. 如果该卡金额过少，则返回 7d。



2.2.5 下机

功能介绍 如右图，用户根据指引输入卡号，密码后点击 OK 就可以下机，下机时该卡会标记为已下机，同时扣除费用。

返回结果

- 1. 成功，则返回 8a，提示下机后的余额
- 2. 卡号与密码不匹配页面，则返回 8b
- 3. 已下机再重复下机，则返回 8c
- 4. 如果该卡金额过少，则返回 8d。

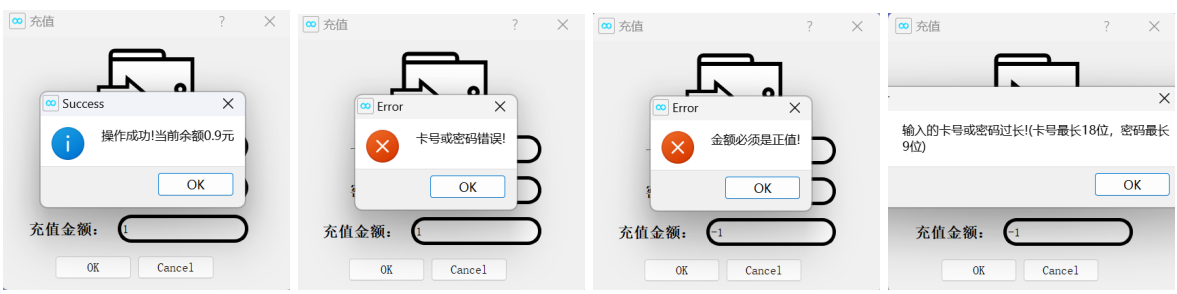
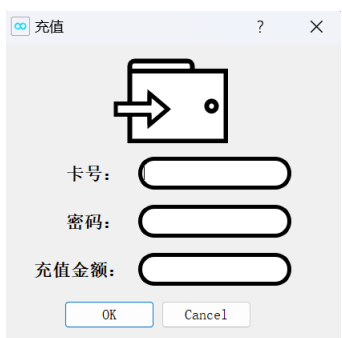


2.2.6 充值

功能介绍 如右图，用户根据指引输入卡号，密码，充值金额后点击 OK 就可以充值。

返回结果

- 1. 成功，则返回 9a
- 2. 卡号与密码不匹配，则返回 9b
- 3. 输入充值金额非正，则返回 9c
- 4. 输入卡号或密码过长，则返回 9d。



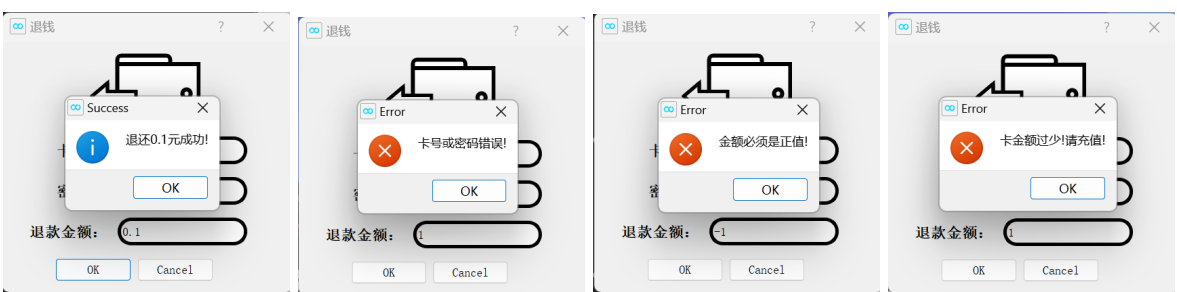
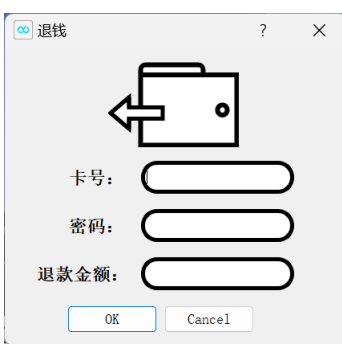
(a) 成功页面 (b) 卡号与密码不匹配页面 (c) 金额非正页面 (d) 输入过长页面

2.2.7 退款

功能介绍 如右图，用户根据指引输入卡号，密码，退款金额后点击 OK 就可以退款。

返回结果

- 1. 成功，则返回 10a
- 2. 卡号与密码不匹配，则返回 10b
- 3. 输入退款金额非正，则返回 10c
- 4. 退款金额不足，则返回 10d



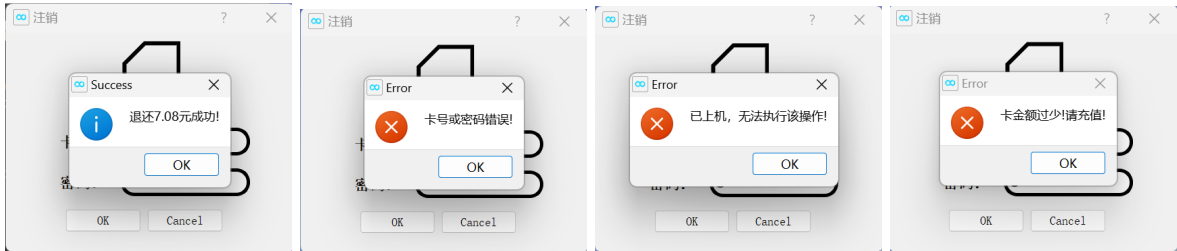
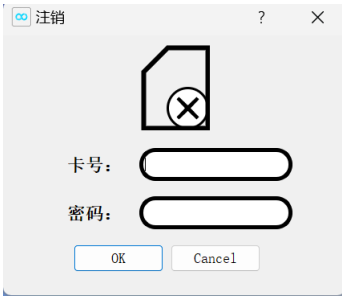
(a) 成功页面 (b) 卡号与密码不匹配页面 (c) 金额非正页面 (d) 退款金额不足

2.2.8 注销

功能介绍 如右图，用户根据指引输入卡号，密码后点击 OK 就可以注销该卡。

返回结果

1. 成功，则返回 11a
2. 卡号与密码不匹配，则返回 11b
3. 注销时处于上机状态，则返回 11c
4. 注销时金额过少，则返回 11d



(a) 成功页面

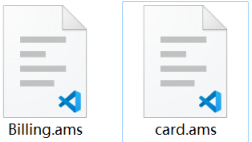
(b) 卡号与密码不匹配页面

(c) 注销时处于上机

(d) 注销时金额小于 0

2.2.9 可以进行保存文件

功能介绍 如右图，每次操作后，会自动生成 card.ams 与 billing.ams 文件，两个文件为二进制文件



2.3 拓展功能

2.3.1 权限管理

功能介绍 如右图，统计服务必须得到管理员权限，账号，密码写在 global.h 文件，账号默认为 NaCl，密码为 qwert12345.

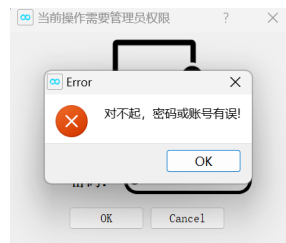
返回结果

1. 成功，则返回 12a
2. 帐号与密码不匹配，则返回 12b





(a) 成功页面

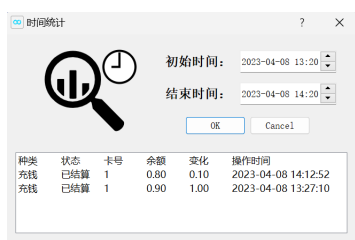


(b) 帐号与密码不匹配页面

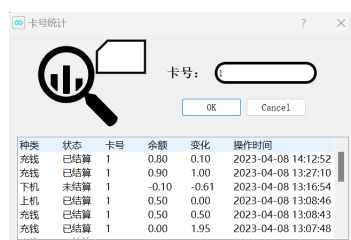
2.3.2 统计

功能介绍 如右图，为统计提供三种统计返回结果

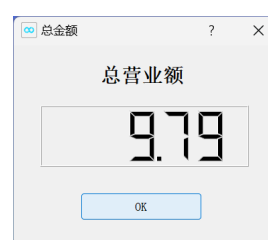
1. 如 13a，按时间节点统计消费记录
2. 如 13b，按卡号统计统计消费记录
3. 如 13c，总金额统计



(a) 按时间统计



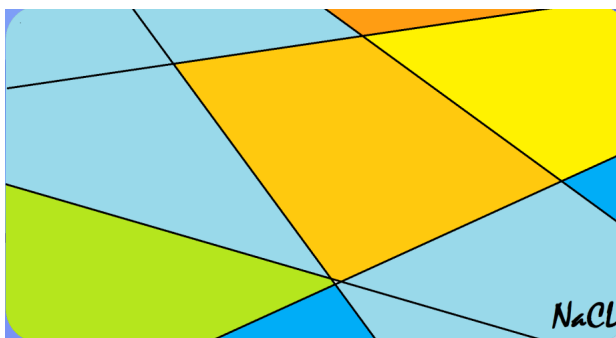
(b) 按卡号统计



(c) 总金额统计

2.3.3 定时启动页面

功能介绍 点下程序后会有个 2s 的启动页面展示，该窗口可以自由拖动，四周使用圆角，更加美观。



2.3.4 采用图形页面

1. 相比 CLI 程序，更加美观，更加方便用户操作

2. 该程序使用了 Qt 进行 GUI 编写，支持 Windows，Linux 跨平台运行

3 典型算法分析

3.1 添加卡流程分析

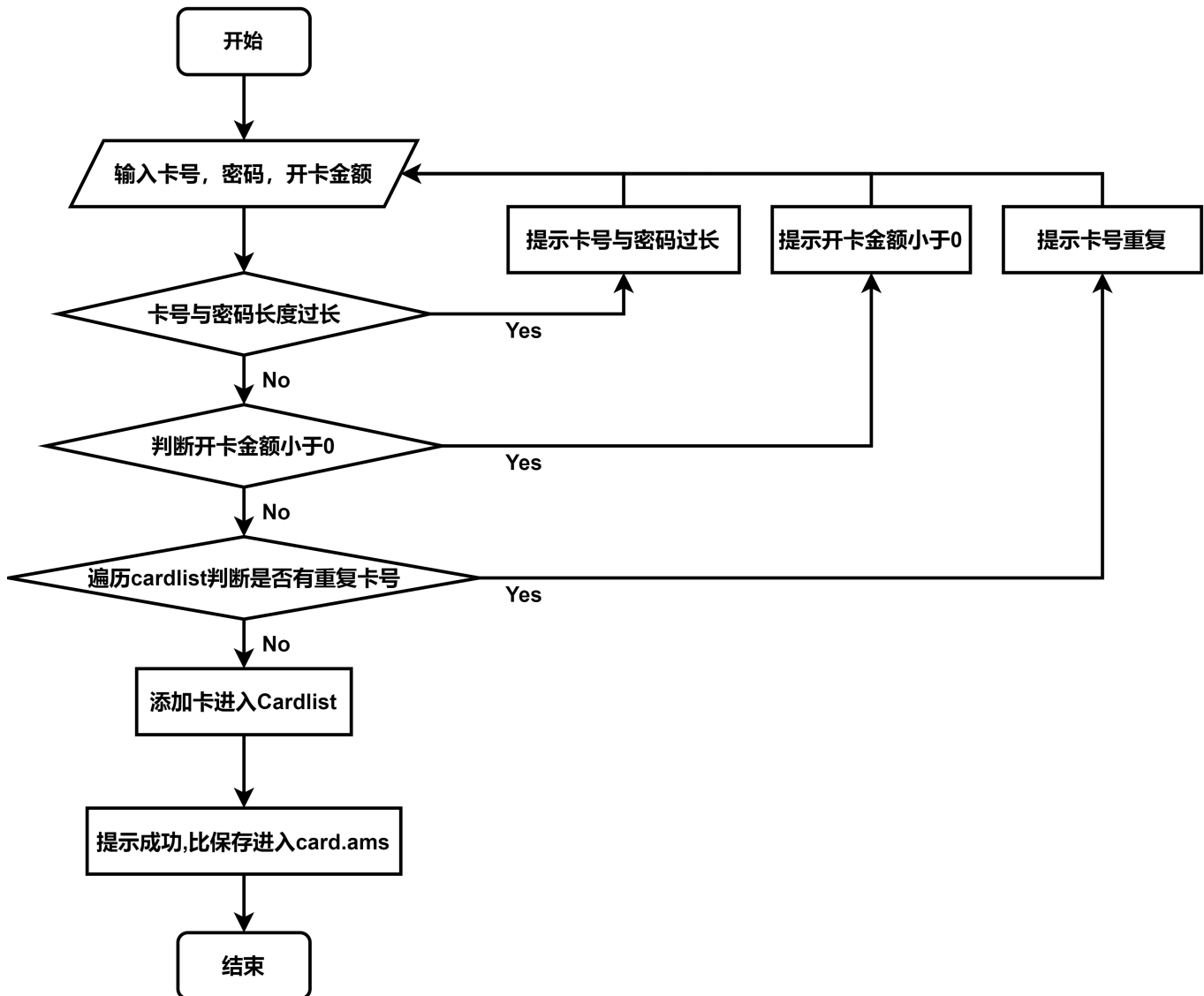


图 14: 添加卡流程图

如图 14为添加卡的流程图，首先判断是否存在输入非数字，长度过长问题，然后链表链表 cardlist 判断是否有重复卡号，这些处理通过后才能完成添加卡，并保存到 card.ams，这些处理提高了软件的鲁棒性，防止意外输出导致程序出错。

3.2 统计流程分析

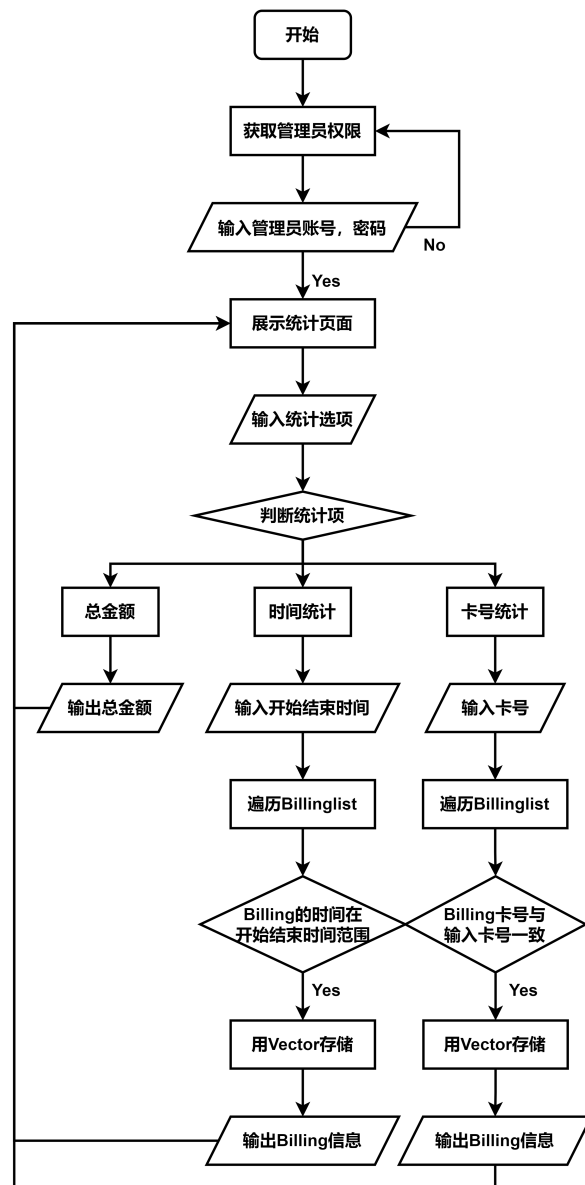


图 15: 统计流程图

如图 15为统计的流程图，首先获取管理员权限，再进入统计页面，选择所需功能，如果是按时间统计，则输入开始结束后时间后，通过遍历 Billinglist，如果 Billing 的时间在其之内，则通过 Billing.toString() 函数转换成 string，再存放在 vector<string> 内，最后将 vector<string> 内容输出，如果是按卡号统计，如果 Billing 的卡号与输入相等，转换成 string，存入 vector<string> 输出，如果是总金额统计，则直接输出总盈利金额。

3.3 上机流程分析

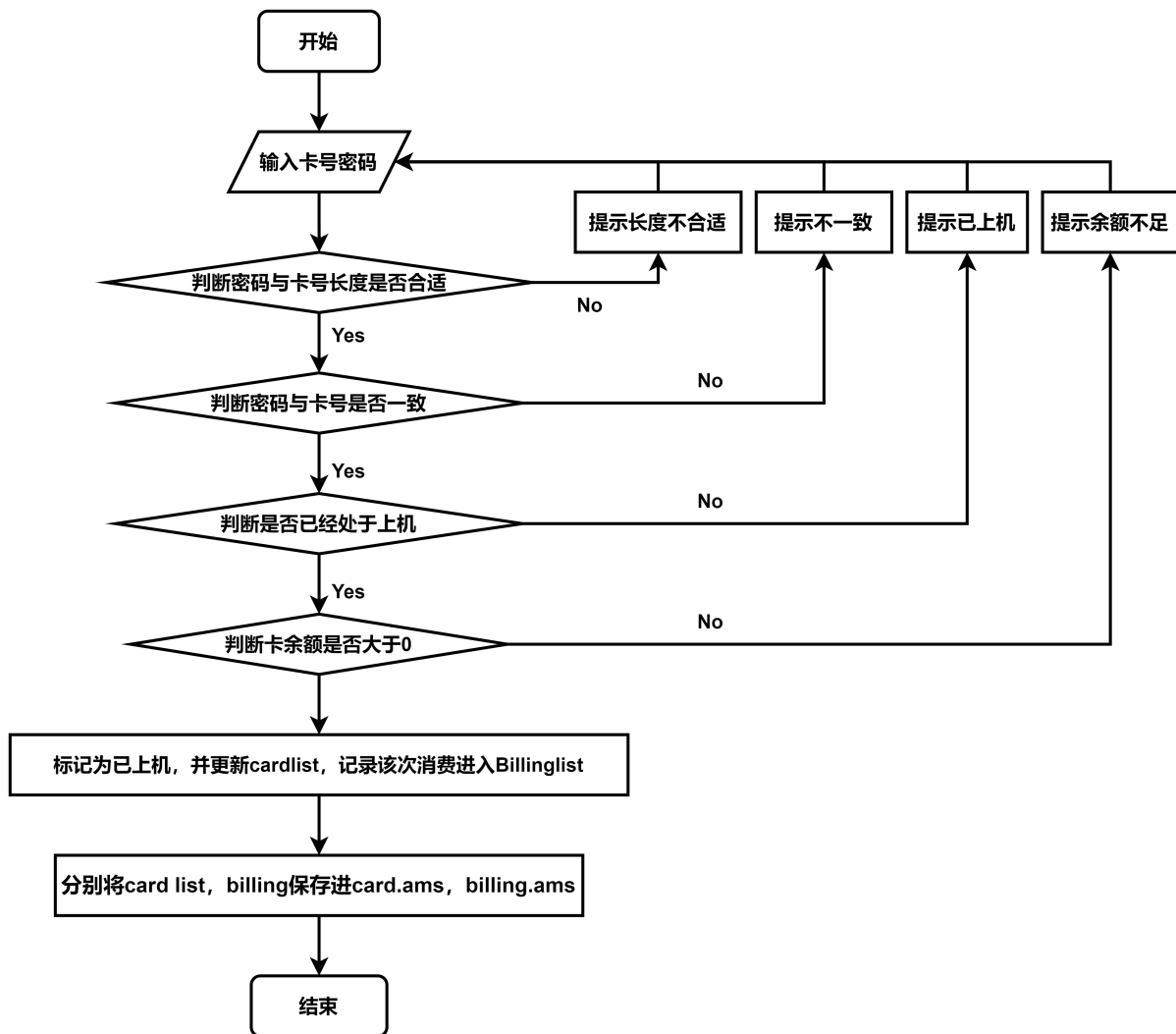


图 16: 上机流程图

如图 16为上机的流程图，首先判断是否存在输入长度过长问题，然后遍历链表 cardlist 判断是否该卡是否存在，该卡密码是否正确，再判断是否已经处于下机状态，是否金额足够，通过后把该卡标记为上机，并分别保存卡信息与消费到 card.ams，billing.ams.

3.4 下机流程分析

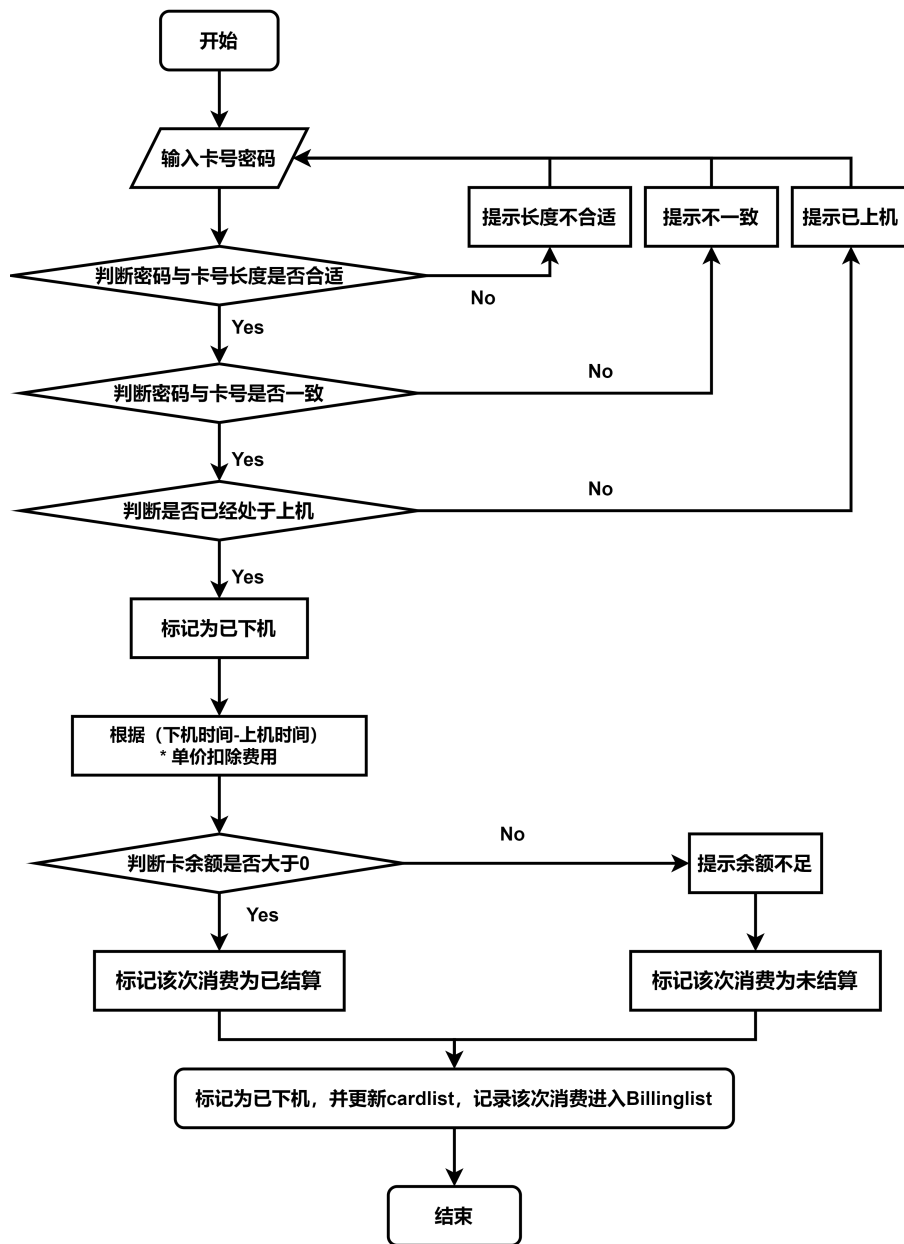


图 17: 下机流程图

如图 17为下机的流程图，首先判断是否存在输入长度过长问题，然后遍历链表 cardlist 判断是否该卡是否存在，该卡密码是否正确，再判断是否已经处于上机状态，通过后把该卡标记为下机，同时扣除费用（下机费用-上机费用）* 每秒金额，如果扣除后余额小于等于 0，就提示余额不够，标记该次消费为未结算，并分别保存卡信息与消费到 card.ams，billing.ams.

3.5 充值流程图

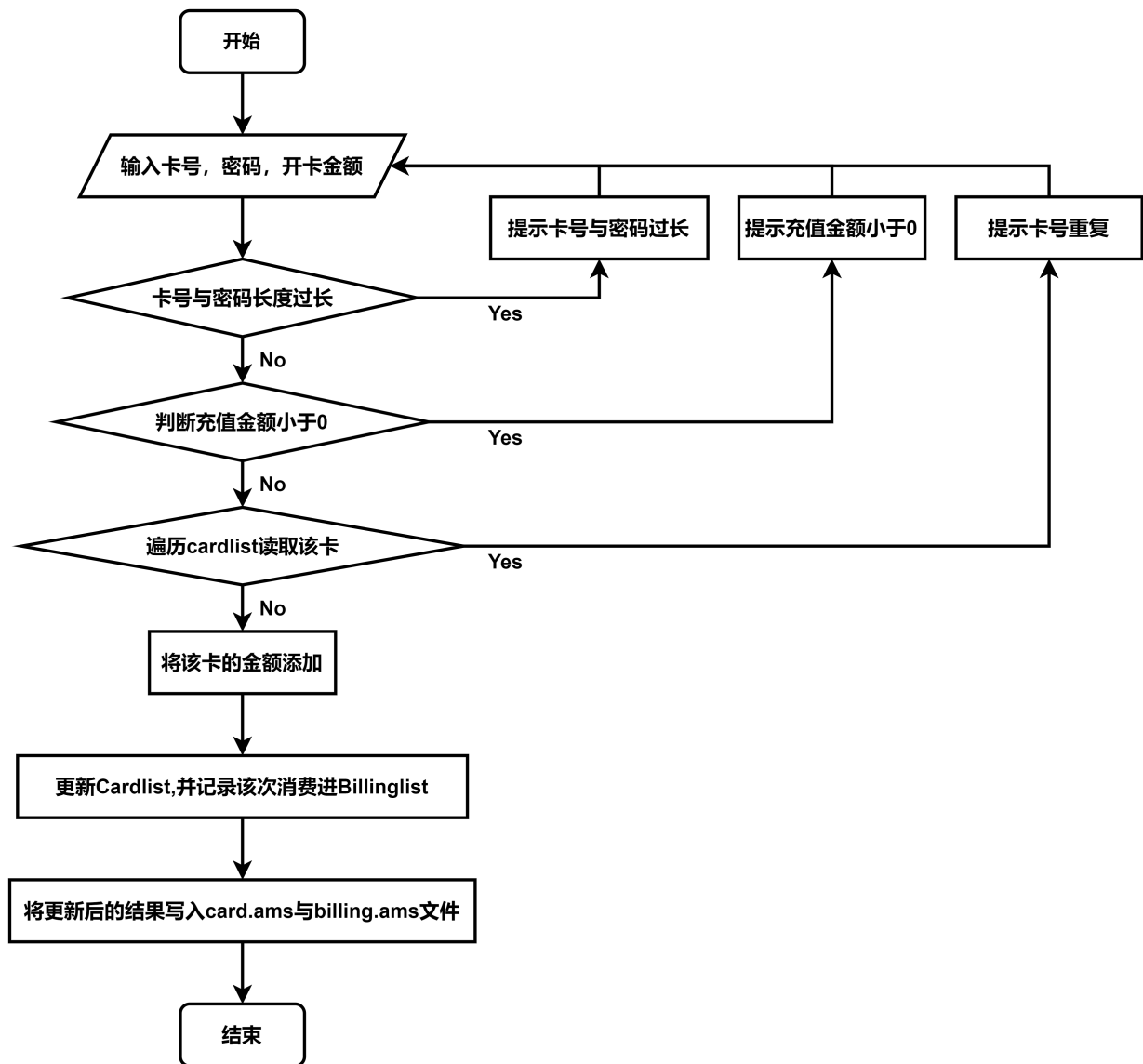


图 18: 充值流程图

如图 18为充值的流程图, 首先判断是否存在输入非数字, 长度过长问题, 然后遍历链表 cardlist 判断是否该卡密码是否正确, 再增加该卡金额, 更新 Cardlist, billing, 将更新结果写入 card.ams, billing.ams.

3.6 退款流程图

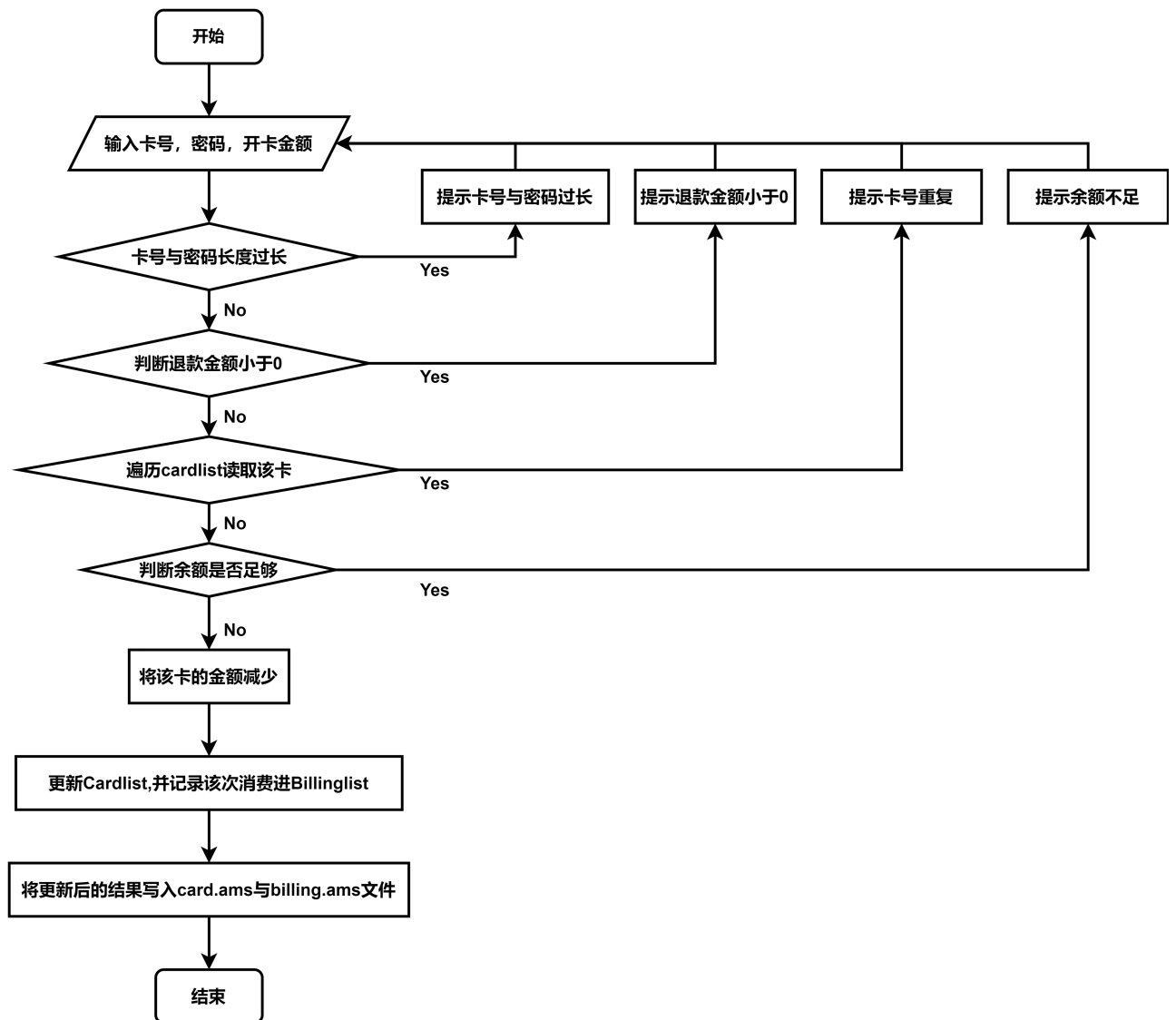


图 19: 退款流程图

如图 19为退款的流程图, 首先判断是否存在输入非数字, 长度过长问题, 然后遍历链表 cardlist 判断是否该卡密码是否正确, 再判断该卡是否余额足够, 减少该卡金额, 更新 Cardlist, billing, 将更新结果写入文件

3.7 注销流程图

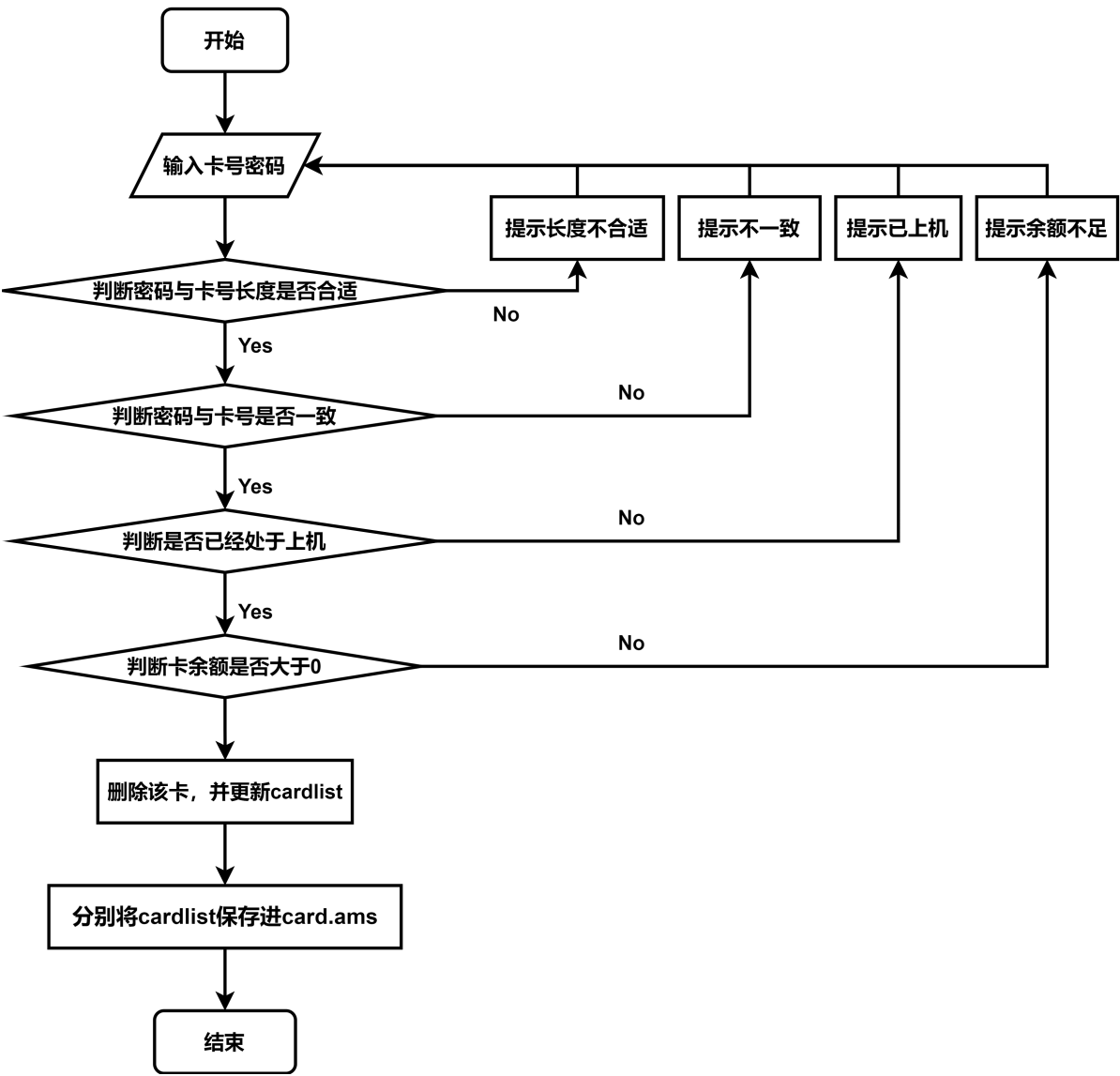


图 20: 注销流程图

如图 20为注销的流程图，首先判断是否存在输入长度过长问题，然后遍历链表 cardlist 判断是否该卡是否存在，该卡密码是否正确，再判断是否已经处于下机状态，是否余额足够，通过后把删除该卡，并分别保存卡信息到 card.ams.

4 开发难点与体会

4.1 多文件编写

4.1.1 难点

虽然 VS 提供非常方便的功能，但是不利于我们学习 C++/C 的编译步骤。使用 VSCode 分文件编写 C++ 程序并不像 VS 那样容易，不能直接简单的编译多个 C++ 文件。Makefile 编写复杂，不好使用。

4.1.2 解决办法

刚开始使用 Cmake 工具，通过编写以下 Cmakelist.txt 编译管理

```
cmake_minimum_required(VERSION 3.0)
project(dev)
set(EXECUTABLE_OUTPUT_PATH ${PROJECT_SOURCE_DIR}/bin)
aux_source_directory(src SRC_LIST)
set(SRC_LIST src/model/node/*.cpp
            src/model/linklist/*.cpp
            src/view/*.cpp
            src/service/service.cpp
            src/main.cpp)
include_directories(include)
add_executable(main ${SRC_LIST})
```

后来使用 Qt 后，通过 Qmake 编译管理多文件系统。

4.1.3 体会

- 相比单文件，这次多文件需要更加良好的编译管理工具。
- 同时熟悉了 C++/C 的编译过程，如图 21所示。

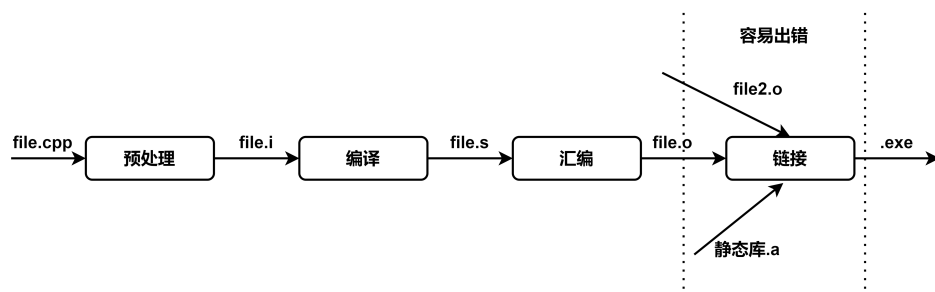


图 21: 编译过程

4.2 较难理解链表

4.2.1 难点

该次程序使用了链表作为存储数据结构，学习链表较难理解链表构建，编写链表容易出现野指针，空指针问题导致运行错误。

4.2.2 解决办法

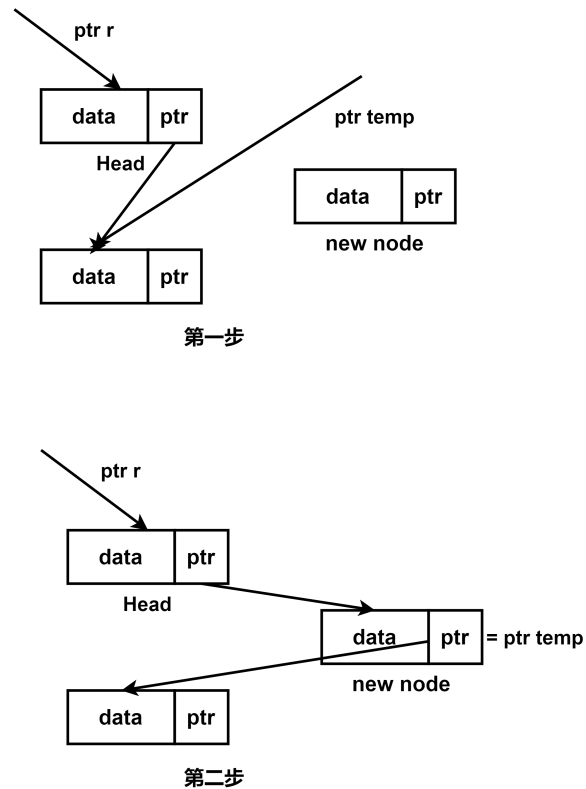


图 22: 链表添加

通过画一些简单的图进行辅助，看图编写程序。例如根据图 22，编写以下链表添加代码。

```
//第一步
Billing* r = head;
Billing* n = new Billing(billing), *temp = r->next;
//第二步
r->next = n;
n->next = temp;
```


4.2.3 体会

- 很多算法与数据结构知识点，单看文字表述概念可能比较枯燥，难以理解，所以有时我们可以通过边画图边编程来辅助我们理解每一步的过程
- 遇到一些运行错误可以使用 `printf`, `qDebug` 等函数进行锁定错误原因，从而改正错误。
- 遇到不懂的需要多看 B 站或者 Mooc 上的视频同时自己实操才能慢慢掌握。

4.3 GUI 编写

4.3.1 难点

该次程序多次需要跳转页面，在这个过程容易出现野指针，数据无法同步等问题，自己写的页面并不美观，同时总是存在各种小 bug.

4.3.2 解决办法

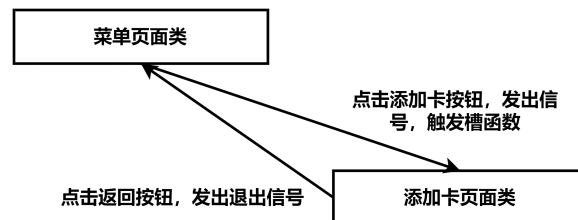


图 23: QT 跳转页面

如图 23。为页面跳转的方式，通过信号与槽来实现页面跳转，用 `connect` 绑定信号与槽实现页面跳转。每次退出后都及时退出该窗口，同时通过指针传递来实现不同页面信息的共享。对于页面不美观，我使用 `qss` 美化页面，同时自己绘制不少图片来形象展示功能。

4.3.3 体会

- 信号与槽是 QT 强大的工具，可以轻松实现类信息的共享。
- 如果只是目前不展示该页面，则将页面隐藏，而不是销毁。
- 使用指针传递来实现跳转页面的信息共享。
- 圆角页面感觉相比直角看起来更舒服。

5 实验总结

5.1 收获与成长

1. 多文件：这次实验我学习接触到多文件编写 c/c++，以前都是单文件，而这次是多文件，刚开始多文件直接在 vscode 上面跑总是显示错误，后来通过上网找到需要用 cmake 做编译管理工具，所以我自己编写了 CMakeList.txt，才能成功运行程序。
2. 程序结构：这次程序功能较多。需要一个比较好的设计，根据网上查找资料与学校课程，我大致采用 MVC 架构，Model 主要负责对于数据的处理，Service 负责联系 Model 与 View 数据交互，View 负责页面展示。这样设计可以方便实现从 CLI 程序到 GUI 程序过渡而不修改大量代码。
3. 链表：这次实验使用了链表，虽然这是个简单的数据结构，但是刚开始学确实很吃力，概念不理解，指针容易到处“乱窜”。后来在 b 站与 Mooc 上看了不少视频才逐渐理解链表，后来编写时边画草图边写才写出来。
4. GUI：该次程序也是我第一次写 GUI 程序，大一上写的基本都是 CLI 程序，输入输出处理简单，不需下太大功夫，而 GUI 程序需要简洁漂亮的画面来展示，需要设计页面的跳转，同时也需要对用户输入进行一定判断，防止意外导致程序崩溃，这也我感受到程序员应该“把简单留给用户，把复杂留给自己”。

5.2 反思与不足

1. 代码可重复利用性并不高：我觉得 QMenu, Qadd, Qaddmoney 等类的作用相似，结构差不多，所以应该可以设计一个父类包括所有共性功能，再通过继承来减少代码重用，但是一直想不出一个合适的父类。而且 linklist 中的 billinglist 与 cardlist 结构也类似，都为链表，以后打算学习用泛型编程来做一个 linklist 模板类来减少重复写两次链表。
2. GUI 页面不优美：我大部分采用 QT 自带的控件，一般都不好看。以后还要继续学习 QT，学习自定义定义控件，使画面更加多样好看，对用户友好。
3. 部分数据不灵活：每秒的扣钱数，管理员的账户密码都是写死在 global.h 里，这些数据只能在源码里修改。以后打算通过 json 确定配置，这样更加灵活方便。
4. 密码都是明文保存：容易泄露。明文密码容易导致个人信息的丢失，例如 CSDN 就出现明文密码泄露事件。所以密码需要使用一种加密算法进行加密。以后有机会接触一下加密算法。
5. 有些功能算法可以优化：比如字符串比较可以使用 KMP 等更好的算法来提高速度等，随着程序增大，更优的算法可以提高运行速度。

5.3 感想

计算机世界的水很深，需要我们不断学习新知识，多实践，时刻保持谦虚好学态度，相信我们终将在信息海洋中乘风破浪。

《程序设计综合实验》成绩评定表

班级：计算机类 m2201 班 姓名：龚锦程

学号：0122210880132

序号	评分项目	满分	实得分
1	学习态度认真、遵守纪律	10	
2	迭代开发进度合理, 提交结果正确	40	
3	代码规范、注释清晰、可读性好	10	
4	软件功能完善、运行正确	20	
5	验收情况良好	10	
6	报告规范、描述清晰准确	10	
		总得分	
评语：			

指导教师签名：

2023 年 月 日