



面向对象程序设计

(三级项目报告)

基于面向对象思想的 ATM 系统设计和实现

小组	23 级计算机科学与技术 5 班		第 1 组
小组成员	学号	姓名	所做工作
	202211090252	李济岑	整体项目框架设计及分工, QT 相关可视化界面制作, 拓展项目整体设计
	202311040113	姜枫	项目中 GUI 可视化界面的部分 QT 设计与功能实现, 展示视频制作
	202311040111	李梦晗	项目中 DOS 命令行界面的功能设计与实现, 及 BankCard 类的整体设计
	202311040109	闫硕	ATM 类的整体设计和文件读写, 讨论课 PPT 制作及项目流程图制作
	202311040110	武鑫沂	部分功能函数的实现, 及 PPT 框架编写
	202311040112	张泽一	部分功能函数的实现, 及报告框架编写
日期	2024. 4		

指导老师: 吴培良

摘要

[摘 要]

本文主要分为摘要、前言、目录、正文、结论与体会与附录六个部分。本报告主要记录了本小组在面向对象程序设计这门课程中三级项目 ATM 系统的具体设计方案、项目效果及心得体会。

ATM 系统是生活中常见的一种银行管理系统设计，具有广泛的应用。本小组尝试实现 ATM 系统的 DOS 命令行风格和 GUI 用户图形化风格，具有用户登录，用户注销，用户注册，余额查询，存取款以及密码修改等功能以保障系统的安全性和功能性。在实现过程中，采用了 C++ 和 QT 进行功能和图形化的实际开发。

拓展项目：本项目介绍了一个以 QT 实现的遥控器控制电视机系统。该系统通过使用 QT 的信号与槽机制进行信号的绑定和响应，实现了遥控器与电视机之间的稳定连接与高效交互。

[关键词] ATM C++ QT 人机交互 图形化界面 命令行

前言

计算机科学与技术的迅速发展使得自动化管理系统在各个领域中得到广泛应用。随着机械和业务自动化普及，ATM 银行管理系统成为一个常见的系。本论文旨在介绍并详细阐述一款由本文作者及其团队开发的基于 C++ 和 QT 的 ATM 管理系统。

本系统的设计灵感来源于对现代社会信息化需求的深刻理。该系统不仅具备基本的银行卡操作，如查询、取款、存款、修改密码、贷款等，还包含了用户登录、用户注销、用户注册等用户级功能，同时设计 Root 管理员账户。为了提升用户体验，我们还采用 QT 进行界面可视化设计，将系统设计为 GUI 和 DOS 两种风格的双系统。

通过这个项目，我们旨在展示在 C++ 编程环境下，如何构建一个功能丰富、高效稳定的 ATM 银行管理系统。同时，我们将深入讨论系统设计的各个方面，包括界面设计、数据结构选择、算法优化等。除此之外，我们还完成了拓展项目：遥控器控制电视机的制作和完善，经过小组通力合作，得到了巨大的提升。

本项目报告将分为多个部分，首先将介绍系统的背景和目的，然后阐述系统的设计与实现过程，包括功能模块划分、数据结构选择、算法设计等，并将展示系统的具体功能和效果。最后，我们将总结整个项目的经验和教训，在此基础上进行展望。

目录

面向对象程序设计.....	1
摘要	2
前言	3
正文	5
一、 研究内容的总体思路	5
（一）可行性分析.....	5
（二）需求分析.....	5
二、 系统设计	6
（一）选择窗口.....	7
（二）DOS 风格.....	8
（三）GUI 风格.....	9
（四）拓展项目	11
三、 系统实现与测试	11
（一）实现.....	11
（二）测试.....	18
结论与体会	27
附录	28

正文

一、研究内容的总体思路

（一）可行性分析

本项目的的主要工作是实现一个 ATM 系统的基础功能和人机交互。我们采用 C++面向对象思想，使用类对底层功能和数据进行设计和存储，保障了系统和用户信息的安全性及隐私性。同时采用 QT 进行 DOS 和 GUI 两种风格的窗体设计。

（二）需求分析

经过小组讨论，结合现实中 ATM 系统的实际功能，确定出一个 ATM 银行管理系统应具有如下需求，并为用户提供服务接口：

1. 用户认证：

系统应该安全的用户认证，可以验证用户的身份，使用合法的卡号和密码进行用户登录。同时应该能够注册、登录和注销账户。

2. 现金交易：

用户应该能够进行查询、存款、取款、贷款等金钱交易。系统应该能够处理不同面额的现金，并对不能处理的非法操作进行提示。

3. 界面设计：

用户界面应该友好、直观，易于操作。根据要求应该同时设计命令行界面和用户图形化界面，供用户自主选择和操作。

4. 可扩展性:

系统应该具备一定的可扩展性，以便系统在更新和维护时添加新或修改的功能或适应更多的用户。

5. 系统维护:

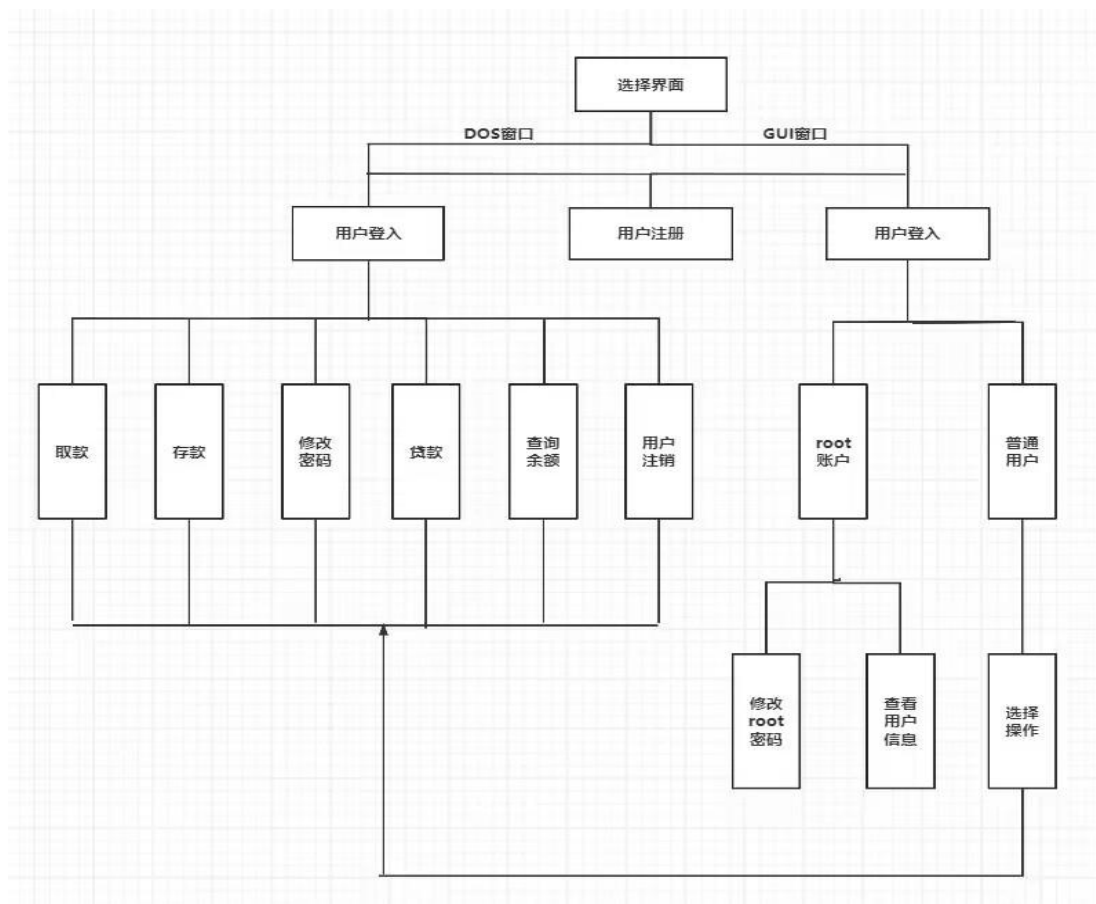
系统管理员应该能够拥有权限，并对全局系统进行系统维护和更新，以确保系统的稳定运行和安全性。

6. 多语言支持:

系统应该支持多种语言，以满足不同用户的需求。

二、 系统设计

系统的功能流程设计:



本系统以 C++面向对象的思想，通过以类进行封装函数和数据的方法实现系统。在设计中使用继承从抽象类中获取接口，利于此后的功能补充和维护。

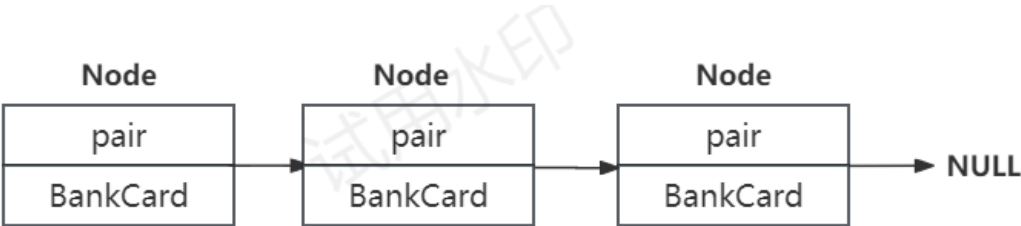
设计为可选择 DOS/GUI 风格的双系统，通过对两个系统的各自的功能和界面设计，整合为本项目的所属功能和实现。用户可通过偏好进行不同版本的系统选择。

在功能上，大体设计为通过输入卡号和密码与载入用户数据进行匹配登录，实现用户认证进入功能菜单选择。用户可选择各自需要的功能进行使用，同时支持中/英双语版本。

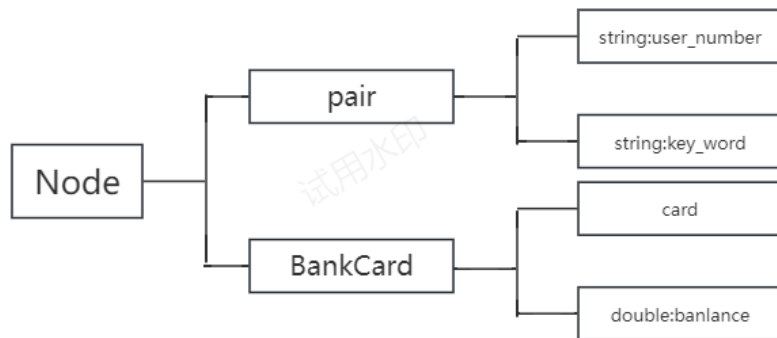
（一）选择窗口

设计一个选择窗口用于衔接 DOS 窗体和 GUI 窗体，在该窗体中即进行数据初始化。我们将读入的 csv 文件数据以 ATM 的私有成员 `map<pair<string, string>, BankCard> user_data` 进行存储，键存放卡号和密码组成的 `pair<string, string>`，值存放 `BankCard` 类，包含 `user_number`，`user_name`，`key_word`，`balance` 等变量。

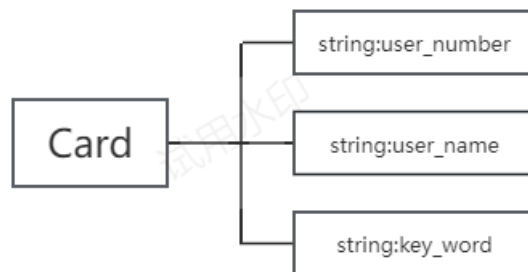
`user_data` 结构：



BankCard 结构:



Card 结构:

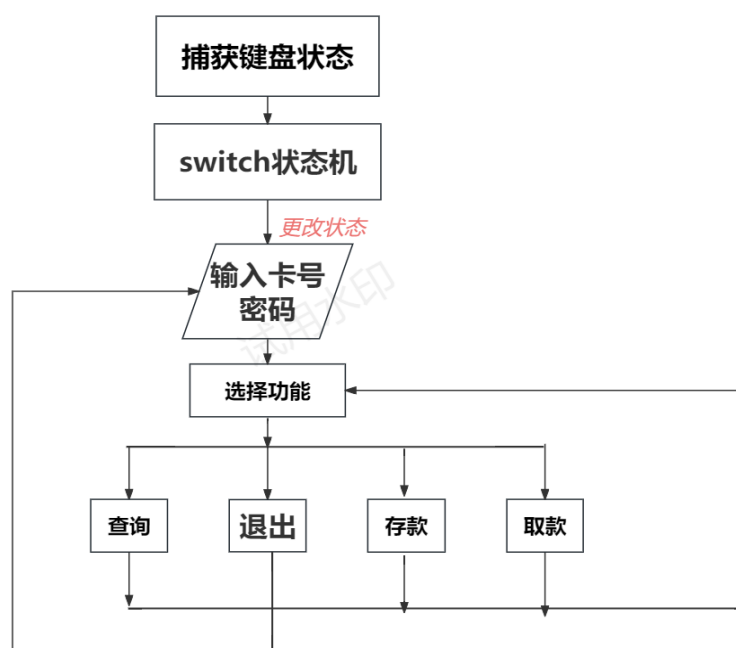


(二) DOS 风格

DOS 风格指的是命令行界面，通过文本命令与用户交互，引导用户进行登录及各种业务功能的选择并执行。该部分使用英文提示，以便实现多语言版本。系统将在退出时进行数据存储。

在该部分中，通过 QTextEdit 进行模拟 cmd 终端效果。因为 QT 不鼓励使用 `std::cin` 或者流类阻塞式输入，可能会导致用户界面无响应或者应用程序假死，从而影响用户体验。所以设置 Switch 状态机，通过检测回车以获取最新一行输入内容并更改状态机状态，达到循环实现各功能的效果。

交换机实现机理图：



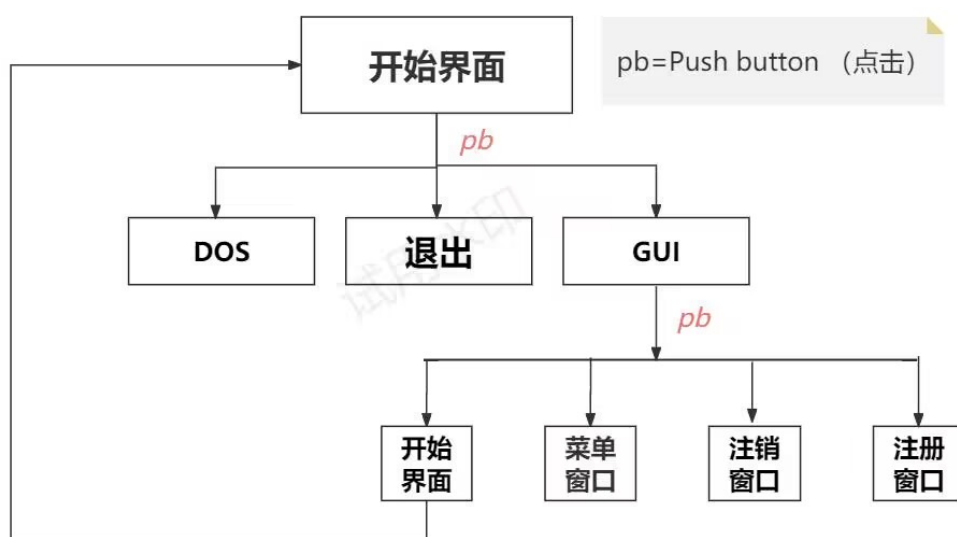
(三) GUI 风格

GUI 风格指的是图形用户界面，系统通过控件实现人机交互，在窗体进行操作的同时通过信号与槽实现与其他窗体的关联。通过控件的输入捕获，在窗体内部执行功能程序并响应用户反馈。

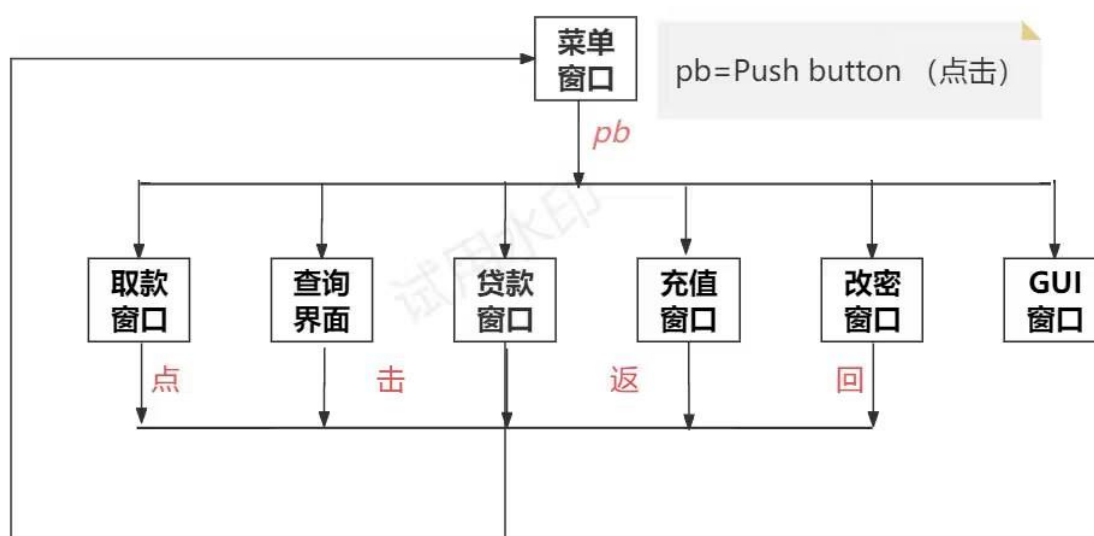
在该部分中大体分为两个部分。第一个部分由选择窗体及 DOS 窗体、GUI 登录窗体组成，通过控件响应实现窗体间的选择和跳转。用户在选择窗体中选择 DOS/GUI 系统，DOS 命令行在窗体内的 QTextEdit 中实现相关功能；GUI 登录窗体登陆后进入到相关窗体。

第二个部分由 GUI 登录后的菜单窗体及各个功能实现窗体组成，通过控件实现窗体的跳转及数据的操作。用户在登录后通过选择各功能控件进入各功能窗体，实现功能。

第一部分：



第二部分：



同时 GUI 部分还设计了 Root 管理员账户（默认卡号为 19 位 0，密码为 6 为 0 且管理员账户不可注销），在登陆时能进入管理员窗口，查看所有用户信息（只读模式），方便管理人员对数据的查看和管理。

（四）拓展项目

拓展项目要求制作一个单个窗体实现遥控器控制电视机，即在单个窗体中设计播放器并设计按钮对其进行控制。

经查询网络资料和多次讨论，我们小组决定以 QVideoWidget 作为播放器的载具，同时设计 QMediaPlayer 部件作为视频播放器和多个按键绑定作为键位。在实际播放方面，我们通过获取本地视频资源路径并将视频资源载入 QMediaPlaylist 作为播放列表。

同时设计 QTextEdit 部件作为快捷键提示文本，QCheckBox 控件进行控制收放，QSlider 作为视频播放进度条，QLineEdit 用于输入 URL 路径等控件，大致都通过获取键事件、鼠标事件或 QMediaPlayer 的变换实现变换和交互。

三、 系统实现与测试

注：本报告中因篇幅并不展示完整代码，详情见 DOS+GUI (QT) 文件夹或附录 Github 仓库。

（一）实现

一）选择窗口和数据初始化

1. 窗体设计：

相关函数：

```
two_class::two_class(QWidget *parent)
```

本项目统一使用 QT 的 ui 类进行窗体设计。在选择窗体中连接 DOS 和 GUI 两部分，作为将两部分连接成双系统的桥梁。设置 QPushButton 按键与两部分窗体绑定连接。

2. 数据初始化:

相关函数:

```
ATM::ATM(const string &path)
```

在选择窗口初始化时动态创建 ATM 类并通过其构造函数将 csv 文件中的数据读入并初始化存入临时数据容器 `map<pair<string, string>, BankCard>` 类型的 `user_data` 中。在选择两种系统时将动态创建的 ATM 变量以引用形式传入各自窗体类实现数据操作。用户的登录将不再读入文件，而是在 `user_data` 中进行查询。

二) DOS 风格

1. Switch 状态机:

相关函数:

```
void three_class::executeFunction(const QString &input)
```

状态机 (State Machine) 是一种抽象机制，用于描述系统在不同状态下的行为和状态之间的转换。它由状态、事件和转移组成，用于描述系统在接收到事件时从一个状态转移到另一个状态的行为。

在本项目的状态机中通过捕获回车进行状态的改变，其中状态为定义在窗体类内的枚举类型 `enum State` 元素。在每个状态下执行各自的状态处理和界面更新，同时更新下一个状态等待触发。即触发回车时，通过捕获新一行文本内容并执行上一个状态处理，同时通过新一行的文本进行状态更新。

2. 事件过滤器

相关函数：

```
bool three_class::eventFilter(QObject *obj, QEvent *event)
```

事件过滤器 (Event Filter) 是一种机制，允许在 Qt 应用程序中捕获和处理事件，包括鼠标事件、键盘事件、定时器事件等。本项目中通过覆写 QTextEdit 中的虚函数 eventFilter，实现捕获键盘在 QTextEdit 中输入的回车。

3. 重定向输出流

相关函数：

```
DOS::int_type DOS::overflow(int_type v)
```

将标准输出流 (stdout) 通过继承后复写，修改标准输出流的缓冲区，将原本输出到终端中的数据可以通过 std::cout 直接输出到 QTextEdit 上，更加便捷。

4. 具体功能实现

① 用户登录：

通过第一次捕获回车，触发初始化状态 State::Start 进入用户登录，状态修改为 State::EnterCarNumber 和 State::EnterPin 并获取新输入的卡号和密码，通过 ATM::userLogin 函数在 user_data 中进行匹配查找实现登录，登录失败则返回登陆界面。登录成功状态进入 State::Transaction。

② 选择业务：

通过选择查询(1)、取款(2)、充值(3)、修改密码(4)、退出(5)进入各自操作状态。查询调用 `BankCard::GetBalance` 函数并输出；取款调用 `BankCard::balanceWithdrawal` 函数，失败则警告并返回菜单；充值调用 `BankCard::balanceRecharge` 函数，失败则警告并返回菜单；修改密码调用 `three_class::isValidPassword` 函数检测密码合法性，并调用 `BankCard::changePassword` 函数进行修改。上述功能失败则返回菜单，退出则返回登陆界面。

③ 返回按钮：

点击返回按钮回到选择窗体并关闭 DOS 窗体，同时将数据写入文件进行保存。

三) GUI 风格

1. 窗体传递

相关函数：

如 `three_class::three_class(ATM &a, QWidget *parent)`

在选择窗体中我们已经读取了文件数据并载入到了 ATM 对象的 `user_data` 中，为了避免多次读取文件，我们将已经初始化好的 ATM 对象在新窗体初始化时作为参数给传入。同时为了避免新窗体内对 ATM 对象的操作无效，我们选择以引用传参的形式传递，后续才当窗体和功能窗体则传参具体登陆后的 `BankCard` 对象的引用。

2. Root 账户设计

相关函数：

```
one_class::one_class(ATM &a, QWidget *parent)
```

在该用户登录窗体中输入卡号和密码进行登录，在匹配成功 user_data 中的用户数据后检测该用户是否为 Root 账户，同时选择打开 Root 的 csv 窗口对所有用户数据进行检测。同理，注销时也通过检测是否为 Root 账户选择是否能够被注销。

3. 警告窗口

相关函数：

```
warning::warning(const QString &node, QWidget *parent)
```

为实现输出不合法或操作失败等信息的反馈，设置一个继承 QDialog 类的一个警告窗体。弹出警告窗体后父窗体不可操作直到警告窗体关闭，传入 QString 字符串用于在窗体内输出警告文本，不同时期的调用传入不同的警告文本以实现复用。

4. 具体功能实现

① 用户注册

用户注册的相关功能是在 one_class 类中实现的。用户通过点击注册按钮进入注册流程，one_class 类中的 user_register 函数将会创建一个 registration 对象，并显示出相应的注册窗口。新用户注册功能并不需要提前读取 csv 文件，因为在用户注册成功后，

ATM 系统会将新创建的用户信息写入到 csv 文件中，以方便下一次 ATM 上的读取操作。

② 用户登录

将用户输入的用户名和密码提取出来，并调用 ATM 类的 userLogin 函数进行用户身份校验，userLogin 函数首先会通过查找用户数据文来寻找用户名和密码是否正确，并返回相应的 BankCard 对象。如果银行卡对象的指针为 null，则表示用户名或密码错误，登录失败；否则就表示登录成功，将用户信息存储在 BankCard 指针中。

③ 用户注销

同用户登录，检测是否存在该用户，若存在则删去该 BankCard 指针指向的节点，否则则为用户不存在或密码错误。用户存在时检测是否为 Root 账户，Root 账户不可注销，弹出警告窗口。

② 选择业务

选择查询、取款、贷款、充值、修改密码、退出等按键进入各自窗体执行操作。查询调用 BankCard::GetBalance 函数并输出；取款调用 BankCard::balanceWithdrawal 函数，失败则弹出警告窗口并重新输入；充值调用 BankCard::balanceRecharge 函数，失败则弹出警告窗口并重新输入；修改密码限定输入框掩码属性，密码合法性，并调用 BankCard::changePassword 函数进行修改。上述功能失败则弹出警告窗口，退出则返回登陆界面。

四) 拓展项目

1. 载入播放资源

相关函数:

```
void test::setPlayer()
```

将本地视频资源播放地址进行访问，载入该目录中所有视频资源到 Loop 模式的 QMediaPlaylist 进行播放并初始化音量。

2. 事件过滤器

相关函数:

```
void test::keyPressEvent(QKeyEvent *event)
```

与 ATM 系统中的事件过滤器相同，用于捕获键盘按键并通过识别固定的快捷键调用相关功能按键的信号触发。该拓展项目快捷键包含:

- ① 空格：暂停/播放
- ② Tab/回车：选集
- ③ 左键：回退
- ④ 右键：快进
- ⑤ 上键：增加音量
- ⑥ 下键：减小音量
- ⑦ Ctrl+左键：上一集
- ⑧ Ctrl+右键：下一集

3. 爬取网络资源

相关函数：

```
void test::on_sure_clicked()
```

以 QT 中提供的与爬虫相关的 QnetworkReply 部件，以获取 QnetworkAccessManager 控件对 HTTP 协议的 URL 视频资源的 get 请求响应并进行指定路径下载。绑定下载完成的信号与播放该视频的槽函数实现播放后即刻播放。

（二）测试

一） 选择窗体

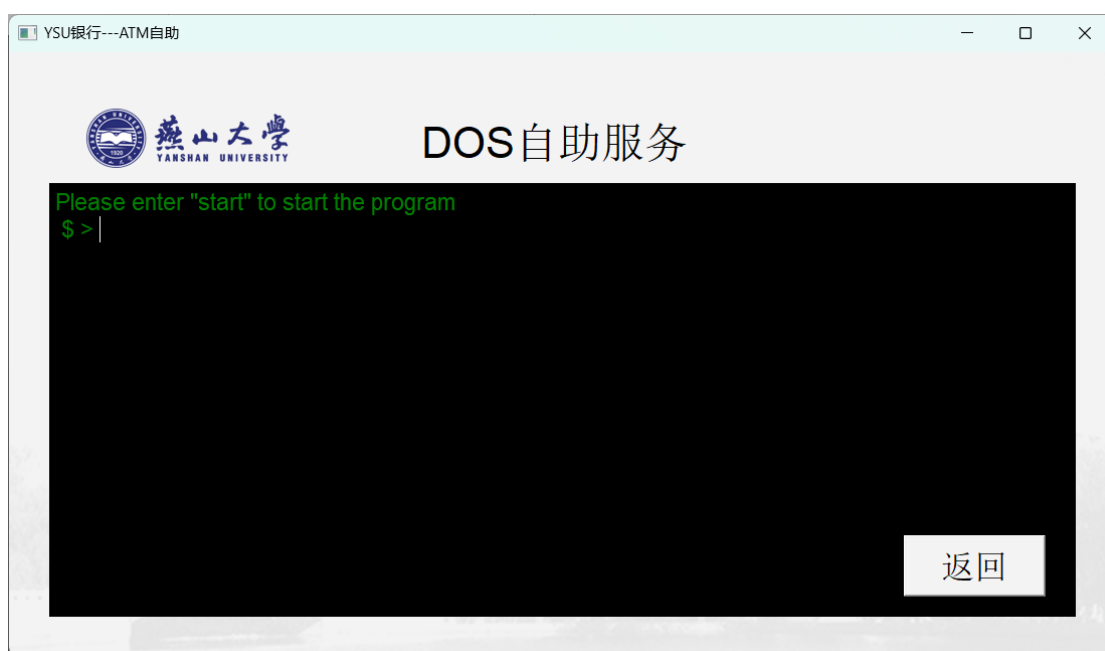
运行程序即弹出初始选择窗体。设置三个 QPushButton 按钮，分别对应 DOS 模式、GUI 模式、退出三种响应，通过 connect 连接按钮信号和对应槽函数分别实现 DOS 窗体、GUI 登录窗体、保存数据并退出的功能。



二) DOS 风格

以 "\$ >" 作为用户的命令提示符。输入 "start" 获取初始响应并分别输入卡号和密码，输入完成后会检测输入卡号和密码为键的 user_data 中是否存在相应的数据值并获取 BankCard 数据。

DOS 初始化界面:



登录失败，提示失败并需要重新输入:

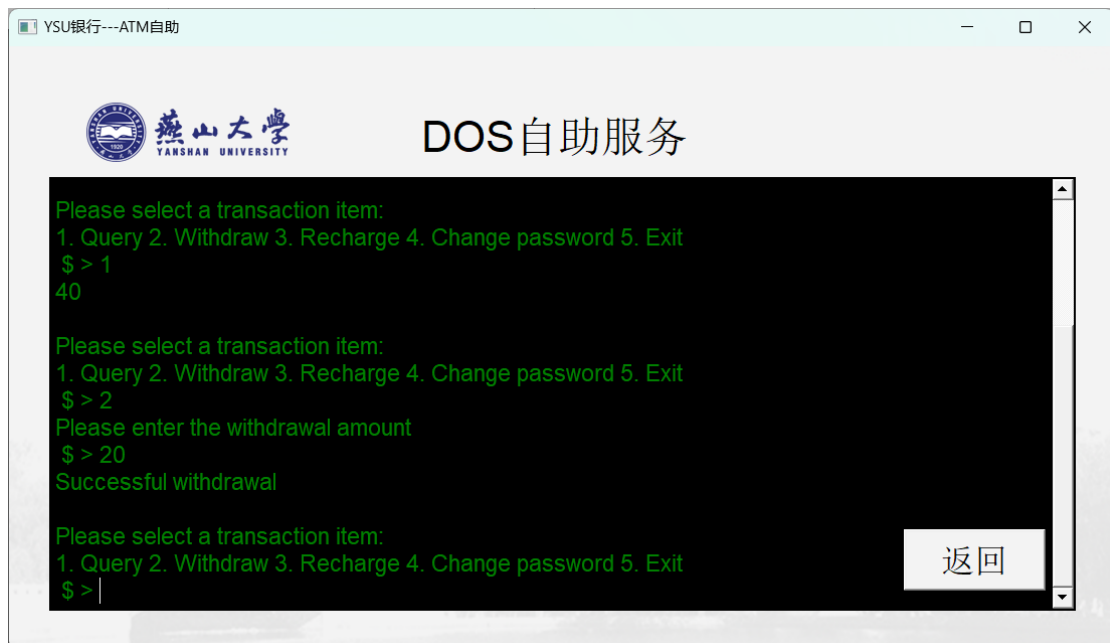


登录成功，提示成功并进入菜单：



登录成功后进入菜单界面，用户通过输入选项选择业务，分别为查询、取款、存款、修改密码、退出。选择退出后用户退出登录。

查询 & 取款：

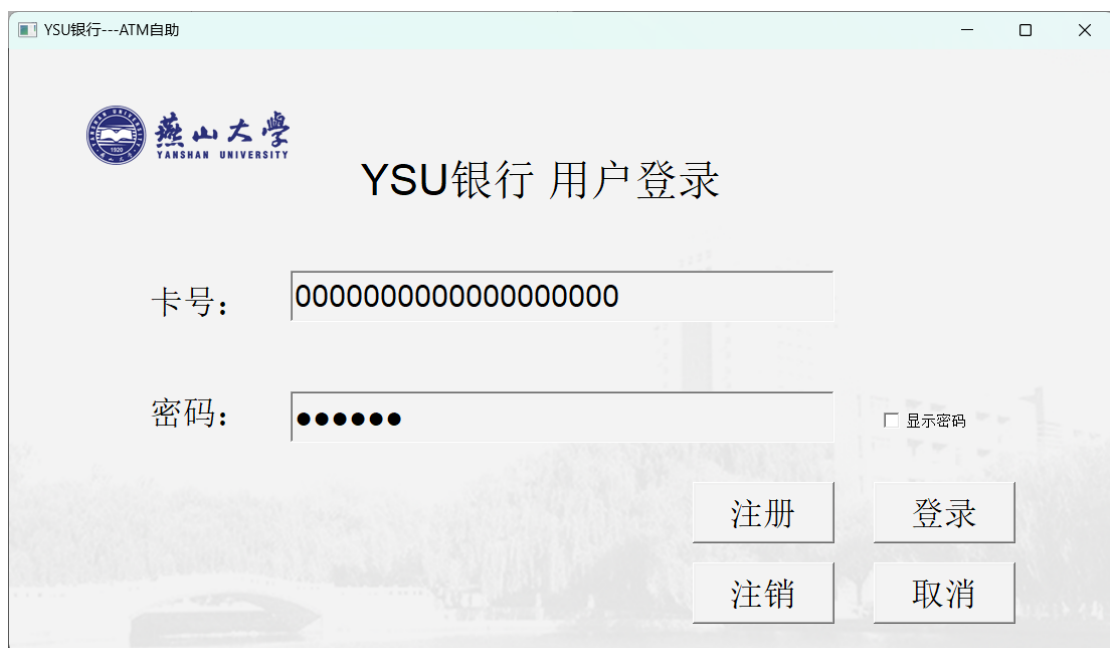


存款 & 修改密码：




三) GUI 风格

进入 GUI 用户登录界面，在该界面可选择用户注册、用户注册、用户注销等操作。两个 QLineEdit 输入框通过掩码属性加密，限定为卡号仅能输入 19 位数字，密码必须为 6 位数字和密码的组合并以密码模式输入不可见，可勾选 QCheckBox 显示密码。



注册界面：



The registration interface for YSU Bank is displayed within a window titled "YSU银行---ATM自助". It features the Yanshan University logo and name. The form includes three input fields: "请输入用户名:" (Please enter username), "请输入密码:" (Please enter password), and "请再次输入密码:" (Please re-enter password). At the bottom right, there are two buttons: "确认" (Confirm) and "返回" (Return).

注销界面（Root 账户不可注销）：



The user login/registration interface for YSU Bank is shown. It includes the Yanshan University logo and the title "YSU银行 用户登录". The form has input fields for "卡号:" (Card Number) and "密码:" (Password). A "显示密码" (Show Password) checkbox is located to the right of the password field. At the bottom, there are four buttons: "注册" (Register), "登录" (Login), "注销" (Logout), and "取消" (Cancel). A warning dialog box is overlaid on the form, titled "警告!" (Warning!), with the message "Root账户无法注销!" (Root account cannot be deleted!) and a "确定" (Confirm) button.

Root 账户登陆界面（默认为卡号密码全为 0）：



非 Root 账户登录界面：



除贷款功能外（暂未开通该功能）外，其他功能在输入不合法或越界数据时都会出现弹窗警告并重新输入，如余额不足或修改密码时旧密码不符或密码不合法：



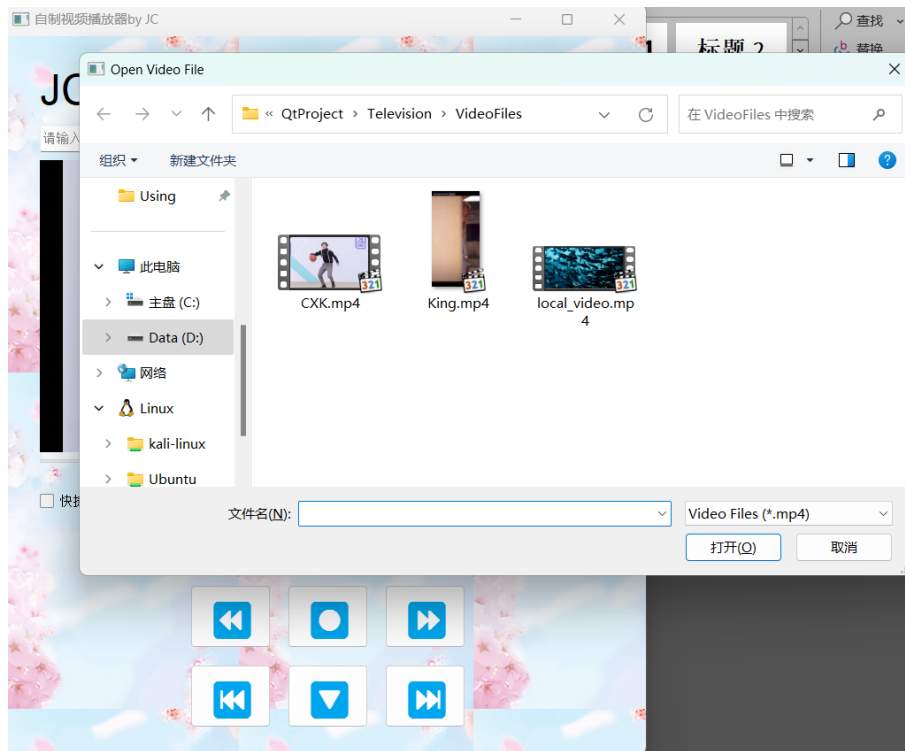
四） 扩展项目

运行项目时自动载入本地播放资源路径，将视频资源以 Loop 循环模式载入播放器的播放列表，同时可对播放资源进行选择或输入 URL，对网络上 HTTP 协议的视频资源进行爬取下载至本地进行播放。

本地资源播放：



本地资源选择：



网络资源下载：



同时本项目还通过信号与槽机制绑定了所有按键相对应的快捷键，同时实现键响应和鼠标响应：

结论与体会

（一）结论：

在本报告中，我们详细介绍了基于 C++ 和 Qt 框架开发的 ATM（自动取款机）可视化系统。该系统利用 C++ 语言的高效性和 Qt 框架的丰富功能，实现了一个用户友好、功能完善的 ATM 系统。ATM 系统是人们日常生活中稀松平常、随处可见的一种简单高效的业务系统，但仍然要兼顾功能性，安全性，人机交互与便于维护等诸多方面。通过开发这个 ATM 可视化系统，我们组成员深刻体会到了软件开发的乐趣和挑战。在这个过程中，不仅学会了如何使用 C++ 和 Qt 框架来实现具体的功能，还学到了团队合作、项目管理等方面的技能。

（二）心得体会：

在系统程序的设计和编写中，大家分工明确、各司其职，每个人都在为了小组共同的目标添砖加瓦，集思广益。尽管小组在整体进程中也是一路坎坷，但不妨碍我们各自攻坚克难，一步步完善项目。路漫漫其修远兮，吾将上下而求索，通过这次 ATM 系统的开发，我们不仅提升了自己的专业技能，更深刻体会到了团队合作的重要性。

附录

A 项目源代码:

https://github.com/LeeJc02/YSU_Project/tree/main/CPlusPlusProject

B 拓展项目源代码:

https://github.com/LeeJc02/Qt_Player