TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

MÔN CẤU TRÚC DỮ LIỆU

---◄►---



**BÁO CÁO ĐỒ ÁN**

**Chương trình Mô phỏng cây ATM  
ID:PRO\_CTDL\_1**

**Nhóm gồm các thành viên:**

Châu Hoàn Thiện-49.01.103.077

Trần Ngọc Nhi-49.01.103.057

Nguyễn Thị Nhật Hạ-49.01.103.023

**Giáo viên hướng dẫn:** Ths.Võ Hoàng Quân

TP.HỒ CHÍ MINH - Năm 2024

Mục Lục

[**1 Giới thiệu đề tài** 1](#_Toc181483008)

[**1.1 Yêu cầu** 1](#_Toc181483009)

[**1.2 Chức năng** 2](#_Toc181483010)

[**2 Giải quyết và thiết kế** 5](#_Toc181483011)

[2.1 Các thư viện sử dụng trực tiếp trong chương trình mà không biên dịch 5](#_Toc181483012)

[2.1.1 Thư viện iostream 5](#_Toc181483013)

[2.2 Các class để quản lý các đối tượng cần thiết 8](#_Toc181483014)

[2.2.1 User.h 8](#_Toc181483015)

[2.2.2 Admin.h 9](#_Toc181483016)

[2.2.3 TheTu.h 9](#_Toc181483017)

[2.2.4 QuanLyUser.h 10](#_Toc181483018)

[2.3 File chính chạy chương trình ATM.cpp 12](#_Toc181483019)

[**3 Phân công nhiệm vụ** 12](#_Toc181483020)

[**4 Sản phẩm** 12](#_Toc181483021)

[**5 Tham khảo** 12](#_Toc181483022)

[**6 Link video demo** 12](#_Toc181483023)

# **1 Giới thiệu đề tài**

## **1.1 Yêu cầu**

Viết chương trình mô phỏng hoạt động của một máy ATM với các yêu cầu sau:

❖ Về thông tin lưu trữ: chương trình lưu trữ các thông tin về Admin, Thẻ từ, Tài khoản,

Lịch sử giao dịch:

1. Thông tin Admin là danh sách có tối thiểu 3 Admin lưu trong file Admin.txt gồm: user, pass.

2. Thông tin thẻ từ của tất cả user được lưu trong file TheTu.txt, trong đó mỗi thẻ từ bao gồm 2 thông tin (Danh sách có ít nhất 10 thẻ từ):

- Mã số tài khoản (ID): một chuỗi gồm 14 chữ số.

- Mã PIN: một chuỗi gồm 6 chữ số.

3. Thông tin tài khoản của mỗi user được lưu trong file [ID].txt, trong đó:

- ID là mã số tài khoản.

- Tên tài khoản, ví dụ: Nguyen Trung Kien.

- Số dư tài khoản, ví dụ: 100000.

- Loại tiền tệ, ví dụ: VND.

4. Thông tin lịch sử giao dịch của mỗi user được lưu trong file [LichSuID].txt, trong đó mô tả đầy đủ các thông tin: ID, loại giao dịch (rút tiền, chuyển tiền), số tiền giao dịch, thời gian giao dịch.Yêu cầu khác:

* Người dùng có thể lựa chọn thực hiện chức năng khác hoặc kết thúc chương trình sau khi hoàn thành mỗi thao tác/chức năng.
* Trong trường hợp có lỗi, hệ thống sẽ thông báo thông tin lỗi đến user, hủy chức năng đang thực hiện, và cho phép user chọn chức năng khác để thực hiện hoặc kết thúc chương trình.
* Mã hóa mã pin/ pass thành dấu \* khi đăng nhập.
* Dùng màu để thiết kế giao diện dễ nhìn.

❖ Yêu cầu khác:

- Người dùng có thể lựa chọn thực hiện chức năng khác hoặc kết thúc chương trình sau

khi hoàn thành mỗi thao tác/chức năng.

- Trong trường hợp có lỗi, hệ thống sẽ thông báo thông tin lỗi đến user, hủy giao dịch

đang thực hiện, và cho phép user chọn chức năng khác để thực hiện hoặc kết thúc

chương trình.

- Mã hóa mã pin/ pass thành dấu \* khi đăng nhập.

- Dùng màu để thiết kế giao diện dễ nhìn.

## **1.2 Chức năng**

Các chức năng chính của chương trình gồm:

1. Hiển thị menu lựa chọn chính lựa chọn chức năng đăng nhập Admin hay đăng nhập User và các menu con gồm:

A screenshot of a computer

Description automatically generated

* 1. Login Admin: Admin chỉ được phép truy cập đến các chức năng bên dưới khi họ nhập đúng thông tin user, pass, đang tồn tại trong file Admin.txt.
     1. Xem danh sách Tài khoản: Hiển thị toàn bộ thông tin danh sách TheTu có trong file TheTu.txt.
     2. Thêm Tài Khoản: Thêm một Tài khoản mới vào danh sách thẻ từ, ID không trùng và mã Pin mới mặc định là 123456. Nếu thêm thành công, tạo tự động file thông tin tài khoản [ID].txt và [LichSuID].txt đầy đủ các thông tin và ghi update lại dữ liệu cho file TheTu.txt.
     3. Xóa Thẻ: Xóa một Tài khoản theo ID. Nếu xóa thành công thì xóa file [ID].txt tương ứng và ghi update lại dữ liệu cho file TheTu.txt
     4. Mở khóa Tài Khoản: cho phép mở khóa theo ID những tài khoản đã bị khóa do đăng nhập sai quá 3 lần.

Menu chương Admin sau khi đăng nhập thành công như sau:

A black sign with yellow text

Description automatically generated

* 1. Login User: users chỉ được phép truy cập đến các chức năng bên dưới khi họ nhập đúng ID, mã PIN, và thông tin tài khoản của họ đang tồn tại trong hệ thống.

*Lưu ý: Nếu đăng nhập lần đầu tiên thì yêu cầu user phải đổi mã pin mặc định (mã pin mới không được giống mã pin cũ). Nếu đăng nhập thành công mới cho lựa chọn các chức năng trong hệ thống. Nếu không báo sai và nhập lại. Nếu nhập sai 3 lần thì thoát khỏi hệ thống và khóa tài khoản của user.*

Màn hình đăng nhập như sau:

*A black card with green and yellow text

Description automatically generated*

Menu chương trình chính sau khi đăng nhập thành công như sau:

A menu with yellow text

Description automatically generated

a. Xem thông tin tài khoản: Cho phép người dùng xem thông tin và số dư tài khoản của

mình từ file [ID].txt.

b. Rút tiền: user nhập số tiền cần rút, hệ thống xác nhận lại việc rút này trước khi trừ khỏi

tài khoản của họ số tiền tương ứng và lưu lại thông tin giao dịch của user.

- Biết rằng số tiền cần rút phải >= 50 ngàn, <= số dư – 50 ngàn (Tức trong tài khoản

phải được giữ lại 50 ngàn) và là bội số của 50 ngàn. Trong trường hợp nhập số tiền

sai sẽ báo người dùng nhập sai và sau đó xuất hiện lại giao diện nhập tiền cần rút.

Nếu rút tiền thành công phải cập nhật lại số dư trong file [ID].txt tương ứng.

- Giao dịch ghi lại lịch sử rút tiền.

c. Chuyển tiền: users nhập thông tin tài khoản sẽ chuyển đến, số tiền sẽ chuyển, xác nhận

việc này trước khi hệ thống cập nhật thông tin số tiền tương ứng trong các tài khoản

tương ứng và lại lưu thông tin giao dịch của user.

- Số tài khoản đến phải hợp lệ (tức tồn tại trong danh sách tài khoản) và số tiền cần

chuyển phải >= 50 ngàn, <= số dư – 50 ngàn và là bội của 50 ngàn. Nếu không hợp

lệ hãy thông báo người dùng và cho nhập lại.

- Giao dịch ghi lại lịch sử chuyển tiền: chuyển cho số tài khoản nào, tên gì và bao

nhiêu tiền, thời gian.

d. Xem nội dung giao dịch: Cho phép người dùng xem lại toàn bộ giao dịch của mình.

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

\* DANG NHAP USER \*

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

ID: 10014504501111

Pin: \*\*\*\*\*\*\*\*

\* \* \* \* \* \* \* \*\*MENU \* \* \* \* \* \* \* \* \* \*

1. Xem thong tin tài khoản

2. Rut tien

3. Chuyen tien

4. Xem noi dung giao dich

5. Doi ma Pin

6. Thoat

\* \* \* \* \* \* \* \* \* \* \*\*\* \* \* \* \* \* \* \* \* \* \*

User: user

Pin: \*\*\*\*\*\*\*\*

- Nếu có được nhận tiền từ số tài khoản khác thì cũng ghi nhận lại trong thông tin giao dịch.

e. Đổi mã Pin: Cho phép người dùng nhập lại mã pin cũ để xác nhận đúng chủ thẻ mới cho đổi mã pin. Nhập lại mã pin mới 2 lần để xác nhận chính xác mã pin mới (Mã pin mới không được giống mã pin cũ và mã pin mặc định). Nếu đổi mã pin thành công thì ghi update lại dữ liệu cho file TheTu.txt.

# **2 Giải quyết và thiết kế**

## 2.1 Các thư viện sử dụng trực tiếp trong chương trình mà không biên dịch

### 2.1.1 Thư viện iostream

Tác dụng:

Thư viện iostream trong C++ cung cấp các định nghĩa và công cụ để thực hiện các thao tác nhập/xuất dữ liệu qua dòng lệnh, bao gồm:

1. **Nhập dữ liệu**: sử dụng std::cin để nhập dữ liệu từ bàn phím.
2. **Xuất dữ liệu**: sử dụng std::cout để xuất dữ liệu ra màn hình.

Mục đính sử dụng:

Khai báo thư viện này với #include <iostream>để có thể dùng các đối tượng như cin, cout mà không cần khai báo thêm gì khác.

2.1.2 Thư viện conio.h

Tác dụng:

Thư viện conio.h là thư viện hỗ trợ nhập/xuất (console input/output) trên môi trường DOS và Windows. Thư viện này cung cấp một số hàm cơ bản cho việc xử lý màn hình và bàn phím, tuy nhiên, nó không phải là một thư viện chuẩn trong C/C++, nên không phải tất cả các trình biên dịch đều hỗ trợ.

* Nhập/Xuất ký tự: conio.h cung cấp các hàm cho phép nhập ký tự từ bàn phím ngay lập tức mà không cần nhấn Enter, chẳng hạn như getch() và getche().
* Quản lý màn hình console: Các hàm như clrscr() và gotoxy() giúp xóa màn hình và di chuyển con trỏ đến vị trí mong muốn trên console, hỗ trợ trong việc tạo giao diện điều khiển đơn giản.
* Phát hiện phím nhấn: Hàm kbhit() kiểm tra xem người dùng có nhấn phím nào không, hữu ích cho các chương trình cần xử lý đầu vào liên tục mà không chờ người dùng nhấn Enter.

Mục đích:

Thư viện này cho phép chương trình tương tác tốt hơn với người dùng thông qua các thao tác nhập xuất trực tiếp và kiểm soát màn hình.

2.2.3 Thư viện vector

Tác dụng:

Thư viện vector trong C++ cung cấp một cấu trúc dữ liệu mảng động. vector có khả năng tự động thay đổi kích thước khi thêm hoặc xóa phần tử, giúp quản lý mảng linh hoạt hơn so với mảng tĩnh và dslk.

* Thêm/Xóa phần tử: vector hỗ trợ các hàm push\_back() và pop\_back() để thêm hoặc xóa phần tử.
* Truy cập ngẫu nhiên: Có thể truy cập phần tử bằng chỉ số tương tự như mảng.

Mục đích:

Dùng vector khi cần một mảng có kích thước động và truy cập nhanh qua chỉ số.

2.2.4 Thư viện fstream

Tác dụng:

Thư viện fstream hỗ trợ thao tác với file, cho phép đọc, ghi và mở file trong C++.

* ifstream: Để đọc dữ liệu từ file.
* ofstream: Để ghi dữ liệu vào file.
* fstream: Hỗ trợ cả đọc và ghi.

Mục đích:

Sử dụng fstream để hỗ trợ làm việc với các file Admin.txt, HangHoa.txt...

2.2.5 Thư viện iomanip

Tác dụng:

Thư viện iomanip cung cấp các hàm để định dạng đầu ra, bao gồm:

* setw(): Đặt chiều rộng cho đầu ra.
* setprecision(): Thiết lập số chữ số thập phân.
* setfill(): Đặt ký tự lấp đầy khoảng trống.

Mục đích:

Dùng iomanip xuất bảng số liệu

2.2.6 Thư viện string

Tác dụng:

Các phương thức như substr, length, và find được sử dụng để thao tác với chuỗi, như trích xuất thông tin hoặc xác thực dữ liệu.

Mục đích:

Hỗ trợ thông tin từ trong các file .txt

2.2.7 Thư viện sstream

Tác dụng:

Thư viện sstream để xử lý và chuyển đổi dữ liệu từ chuỗi văn thành các kiểu dữ liệu tương ứng, giúp việc đọc và phân tích dữ liệu dễ dàng hơn.

* getline(): hỗ trợ đọc chuỗi có cả khoảng trắng
* clear():xóa lỗi

Mục đích:

Cùng với thư viện string xử lý các dữ liệu

2.2.8 Thư viện Windows.h

Tác dụng:

Thư viện Windows.h cung cấp các hàm hỗ trợ lập trình trên hệ điều hành Windows, bao gồm cả chức năng dừng chương trình tạm thời.

* Sleep(): Tạm dừng chương trình trong khoảng thời gian nhất định (đơn vị là mili giây).

Mục đích:

Dùng Windows.h khi cần các tính năng liên quan đến thời gian nghỉ.

2.2.10. Thư viện string

Tác dụng: Thư viện string cung cấp các lớp và hàm để làm việc với chuỗi ký tự trong C++. Nó cho phép bạn dễ dàng thao tác với chuỗi, như tạo chuỗi mới, nối chuỗi, so sánh chuỗi, và nhiều chức năng khác.Sleep(): Tạm dừng chương trình trong khoảng thời gian nhất định (đơn vị là mili giây).

Mục đích: Dùng string khi cần lưu trữ và xử lý dữ liệu dạng chuỗi, chẳng hạn như tên tài khoản, mã số tài khoản, và mã PIN.

2.2.11. Thư viện set

Tác dụng: Thư viện set cung cấp một cấu trúc dữ liệu cho phép lưu trữ các phần tử duy nhất và tự động sắp xếp chúng. Nó hỗ trợ các thao tác như thêm, xóa và kiểm tra sự tồn tại của phần tử.

Mục đích: Dùng set khi cần quản lý danh sách các tài khoản hoặc mã số tài khoản mà không cho phép trùng lặp, giúp dễ dàng thực hiện các phép toán tập hợp.

## 2.2 Các class để quản lý các đối tượng cần thiết

### 2.2.1 User.h

Lớp User đại diện cho một tài khoản người dùng trong hệ thống ATM. Mỗi tài khoản chứa thông tin cá nhân và tài chính của người dùng.Các thành phần của lớp Node:

* Thành viên:
  + id: Mã số tài khoản của người dùng, kiểu dữ liệu là string.
  + name: Tên tài khoản, kiểu dữ liệu là string.
  + balance: Số dư tài khoản, kiểu dữ liệu là double.
  + currency: Loại tiền tệ, kiểu dữ liệu là string.
  + isLocked: Trạng thái khóa tài khoản, kiểu dữ liệu là bool.
  + isFirstLogin: Trạng thái đăng nhập lần đầu, kiểu dữ liệu là bool.
  + pin: Mã PIN của tài khoản, kiểu dữ liệu là string.
* Hàm khởi tạo:
  + User(string id, string name, double balance, string currency, const string& pin, bool locked = false, bool firstLogin = true): Khởi tạo một đối tượng User với các tham số được cung cấp.
  + User(const string& id): Khởi tạo một đối tượng User với ID, số dư mặc định là 0, tài khoản không bị khóa và chưa đăng nhập lần đầu.
* Các phương thức:
  + bool loadFromFile(): Đọc dữ liệu tài khoản từ file và cập nhật các thuộc tính tương ứng.
  + void displayAccountInfo() const: Hiển thị thông tin tài khoản ra màn hình.
  + string getID() const: Trả về ID của tài khoản.
  + string getName() const: Trả về tên tài khoản.
  + double getBalance() const: Trả về số dư tài khoản.
  + string getCurrency() const: Trả về loại tiền tệ của tài khoản.
  + string getPIN() const: Trả về mã PIN của tài khoản.
  + bool isAccountLocked() const: Kiểm tra xem tài khoản có bị khóa hay không.
  + bool isFirstLoginStatus() const: Kiểm tra trạng thái đăng nhập lần đầu.
* Các hàm setter:
  + void setName(const string& newName): Cập nhật tên tài khoản.
  + void setBalance(double newBalance): Cập nhật số dư tài khoản.
  + void setCurrency(const string& newCurrency): Cập nhật loại tiền tệ.
  + void setPIN(const string& newPIN): Cập nhật mã PIN.
  + void lock(): Đặt trạng thái tài khoản thành bị khóa.
  + void unlock(): Mở khóa tài khoản.
  + void setFirstLoginStatus(bool status): Cập nhật trạng thái đăng nhập lần đầu.

### 2.2.2 Admin.h

Lớp Admin đại diện cho quản trị viên của hệ thống. Lớp này quản lý các chức năng như đăng nhập, thêm, xóa và quản lý tài khoản admin, và xóa tài khoản user

Các thành viên và phương thức chính:

1. Biến thành viên:

* username: Lưu trữ tên người dùng của quản trị viên, kiểu dữ liệu là string.
* password: Lưu trữ mật khẩu của quản trị viên, kiểu dữ liệu là string.

1. Constructor:

* Admin(string user = "", string pass = ""): Khởi tạo một đối tượng Admin với tên người dùng và mật khẩu mặc định.

1. Phương thức truy cập:

* bool login(): Thực hiện chức năng đăng nhập cho quản trị viên. Nếu đăng nhập thành công, trả về true, ngược lại trả về false. Hỗ trợ tối đa 3 lần thử.
* void showAdminMenu(): Hiển thị menu cho quản trị viên với các tùy chọn như xem danh sách tài khoản, thêm tài khoản, xóa tài khoản, mở khóa tài khoản, hiển thị tất cả quản trị viên, và quay lại màn hình đăng nhập.
* void showAllAdmins(): Hiển thị danh sách tất cả các quản trị viên trong file Admin.txt.
* void addAdmin(const string& username, const string& password): Thêm một quản trị viên mới vào file Admin.txt.
* void deleteAdmin(const string& username): Xóa một quản trị viên dựa trên tên người dùng từ file Admin.txt.
* void unlockAccount(const string& id): Mở khóa một tài khoản người dùng dựa trên ID, cập nhật trạng thái tài khoản trong file tương ứng.

### 2.2.3 TheTu.h

Lớp TheTu đại diện cho thẻ từ, chứa thông tin về mã ID và mã PIN. Lớp này cung cấp các phương thức để truy cập và lưu thông tin thẻ từ vào file.

Các thành viên của lớp TheTu:

1. Biến thành viên:

* id: Lưu trữ mã ID của thẻ từ, kiểu dữ liệu là string.
* pin: Lưu trữ mã PIN của thẻ từ, kiểu dữ liệu là string.

1. Hàm khởi tạo:

* TheTu(const string& id, const string& pin): Khởi tạo một đối tượng TheTu với mã ID và mã PIN được chỉ định.

1. Phương thức:

* string getID() const: Trả về mã ID của thẻ từ.
* string getPIN() const: Trả về mã PIN của thẻ từ.
* void setPIN(const string& newPin): Cập nhật mã PIN mới cho thẻ từ.
* void saveToFile(ofstream& outFile) const: Lưu thông tin của thẻ từ vào file thông qua đối tượng ofstream.

### 2.2.4 QuanLyUser.h

Lớp QuanLyUser đại diện cho quản lý người dùng trong hệ thống ATM, cho phép quản lý danh sách người dùng và thẻ từ, bao gồm các chức năng như thêm tài khoản, xóa tài khoản, cập nhật mã PIN, hiển thị danh sách người dùng và mở khóa tài khoản. Lớp này cũng chịu trách nhiệm đọc và ghi thông tin từ và vào các file liên quan.Các thành viên của lớp TheTu:

1. Biến thành viên:

* vector<User> danhSachUser: Danh sách lưu trữ thông tin của người dùng.
* vector<TheTu> danhSachTheTu: Danh sách lưu trữ thông tin của thẻ từ.

1. Hàm khởi tạo:

* QuanLyUser(): Khởi tạo một đối tượng QuanLyUser và tự động đọc dữ liệu người dùng và thẻ từ từ các file tương ứng.

1. Phương thức:

* vector<User> getDanhSachUser(): Trả về danh sách người dùng.
* bool kiemTraIDTrung(const string& id): Kiểm tra xem ID đã tồn tại trong danh sách người dùng hay chưa.
* void docDuLieuTuFile(): Đọc dữ liệu từ file TheTu.txt và tạo danh sách thẻ từ.
* void docDuLieuUser(): Đọc dữ liệu người dùng từ file [ID].txt và tạo danh sách người dùng dựa trên thẻ từ.
* void hienThiDanhSachUser(): Hiển thị thông tin danh sách người dùng đã tải từ file.
* void capNhatPIN(const string& id, const string& newPin): Cập nhật mã PIN cho thẻ từ dựa trên ID.
* void xoaTaiKhoan(const string& id): Xóa tài khoản người dùng và thẻ từ dựa trên ID.
* void unlockAccount(const string& id): Mở khóa tài khoản người dùng dựa trên ID.
* void updateAccountFile(const User& user): Cập nhật thông tin tài khoản vào file [ID].txt.
* void themTaiKhoan(const string& id, const string& ten, double soDu, const string& loaiTien): Thêm tài khoản mới vào hệ thống.
* void xemThongTinTaiKhoan(const string& id): Kiểm tra và hiển thị thông tin tài khoản dựa trên ID.

1. Phương thức hiển thị menu Admin:
   * static void displayMenuQuanLyTaiKhoan(int selectedOption): Hiển thị menu quản lý tài khoản.
     1. **Thêm tài khoản :** Cho phép quản trị viên nhập tên người dùng và mật khẩu mới để thêm vào danh sách tài khoản quản trị viên.
     2. **Xóa tài khoản :** Cho phép quản trị viên nhập tên người dùng và mật khẩu của tài khoản cần xóa. Nếu thông tin đúng, tài khoản sẽ bị xóa khỏi danh sách.
     3. **Đổi mật khẩu :** Cho phép quản trị viên nhập tên người dùng và mật khẩu mới để thay đổi mật khẩu cho tài khoản đã cho.
     4. **Hiển thị danh sách tài khoản :** Hiển thị tất cả các tài khoản quản trị viên hiện có trong danh sách.
     5. **Quay lại** : Thoát khỏi menu quản lý tài khoản và trở về menu trước đó hoặc kết thúc chương trình.
   * static int pickMenuQuanLyTaiKhoan(): Nhận tùy chọn của người dùng từ menu.
   * static void hienThiMenuQuanLyTaiKhoan(vector<Admin>& adminList): Thực hiện các chức năng quản lý tài khoản dựa trên lựa chọn của người dùng.

Tác dụng:

Lớp Admin là một lớp quan trọng trong hệ thống quản lý tài khoản, giúp quản trị viên thực hiện các chức năng quản lý tài khoản hiệu quả. Nó cung cấp nhiều phương thức tiện ích để thao tác với dữ liệu tài khoản và có khả năng tương tác với các thành phần khác trong hệ thống.

## 2.3 File chính chạy chương trình ATM.cpp

# **3 Phân công nhiệm vụ**

* Châu Hoàn Thiện: User.h, TheTu.h, ATM.cpp, Admin.h: setColor(), clearScreen(), login(), addAdmin(), deleteAdmin(), QuanLyUser.h: kiemTraIDTrung(), hienThiDanhSachUser(), updateAccountFile(), themTaiKhoan(), ghiLichSuRutTien(), rutTien(), ghiLichSuNhanTien(), xemNoiDungGiaoDich(), khoaTaiKhoan()
* Trần Ngọc Nhi: ATM.cpp, Admin: showAdminMenu(), showAllAdmins(), QuanLyUser.h: docDuLieuTuFile(), capNhatPIN(), xoaTaiKhoan(), xemThongTinTaiKhoan(), chuyenTien(), doiMaPIN(),
* Nguyễn Thị Nhật Hạ: ATM.cpp, Admin: unlockAccount(), QuanLyUser.h: getDanhSachUser(), docDuLieuUser(), unlockAccount(), dangNhapUser(), ghiLichSuChuyenTien(),

# **4 Sản phẩm**

Gửi kèm theo trong folder này

# **5 Tham khảo**

<https://github.com/VanHop/ATMSimulation>

<https://youtu.be/l4kjG8TbK70?si=1SWrpZHn-a4bgrcV>

<https://youtu.be/LdRHCa5gyUs?si=0REfu-_20SW6_HeU>

<https://youtube.com/shorts/9Xc0c1g1p8g?si=lTo8FqBclYO3wikN>

<https://www.studocu.com/vn/document/truong-dai-hoc-tai-chinh-marketing/cau-truc-du-lieu-va-giai-thuat/do-an-mon-cau-truc-du-lieu-va-giai-thuat/17816642?origin=search-results>

# **6 Link video demo**