**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN CẤU TRÚC DỮ LIỆU**

**---🙡🙣---**

Logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**PROJECT 1**

Nhóm gồm các thành viên : 48.01.104.116 – Liêu Lâm Tài

: 49.01.104.032 – Nguyễn Đức Điềm

: 49.01.103.063 – Nguyễn Hoàng Đại Phú

: 49.01.103.002 – Dương Thị Hoài Anh

Giảng viên hướng dẫn: ThS. Nguyễn Đỗ Thái Nguyên

Mã lớp học phần : 2411COMP101604

TP.HỒ CHÍ MINH - Năm 2024

**Mục lục**

[1 Giới thiệu đề tài 1](#_Toc146962748)

[1.1 Yêu cầu 1](#_Toc146962749)

[1.2 Chức năng 1](#_Toc146962750)

[2 Giải quyết và thiết kế 4](#_Toc146962751)

[3 Phân công công việc 7](#_Toc146962752)

[4 Sản phẩm 8](#_Toc146962753)

[5 Tham khảo 19](#_Toc146962754)

[6 Link video demo 19](#_Toc146962755)

# Giới thiệu đề tài

## 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 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

## Chức năng

❖ Các chức năng chính của chương trình bao gồm:  
**1. Đăng nhập 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.

A screen shot of a computer

Description automatically generated

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

**A menu with yellow text

Description automatically generated**

**a. Xem danh sách Tài khoản:** Hiển thị toàn bộ thông tin danh sách TheTu có trong file  
TheTu.txt.  
**b. 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.  
**c. 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  
**d. Mở khóa Tài Khoản:** cho phép mở khóa theo ID những tài khoản đã bị khóa do đăng  
nhập sai quá 3 lần.  
**2. Đăng nhập 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 screen shot of a computer screen

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.  
  
- 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.

# Giải quyết và thiết kế

**Person.h**

class Person

{

private:

std::string \_strID, \_strPassword;

bool \_bIsAuth;

public:

Person(std::string strID = "", std::string strPassword = "", bool bIsAuth = false);

Person(const Person& person);

~Person();

std::string getID();

std::string getPassword();

bool getIsAuth();

void setID(std::string strID);

void setPassword(std::string strPassword);

void setIsAuth(bool bIsAuth);

virtual bool login() = 0;

};

**User.h**

class User : public Person

{

private:

std::string \_strFirstName, \_strLastName, \_strCurrencyName, \_strAccountStatus;

double \_dBalance;

public:

User(std::string strID = "", std::string strPassword = "", std::string strFirstName = "No", std::string strLastName = "Name", std::string strCurrencyName = "VND", std::string strAccountStatus = "unlocked", double dBalance = 0.0);

User(const User& user);

~User();

std::string getFirstName();

std::string getLastName();

std::string getCurrencyName();

std::string getAccountStatus();

void setFirstName(std::string strFirstName);

void setLastName(std::string strLastName);

void setCurrencyName(std::string strCurrencyName);

void setAccountStatus(std::string strAccountStatus);

void setBalance(double dBalance);

double getBalance();

int checkNumbersOfUsers();

bool login() override;

void viewAccountInfo();

bool withdrawMoney();

bool transferMoney();

void viewTransactionDetails();

void changePassword();

};

**Admin.h**

class Admin : public Person

{

public:

Admin(std::string strID = "", std::string strMaPin = "");

Admin(const Admin& admin);

~Admin();

void inputID();

void inputPassword();

int checkNumbersOfAdmins();

bool login() override;

bool checkAccountDuplicate(std::string strID);

void addUserAccount();

void viewAccountList();

void removeAccount();

void unlockAccount();

};

**Array.h**

class Node {

friend class Array;

std::string \_strData;

Node\* \_pNext;

public:

Node(std::string strData = "");

Node(const Node& node);

~Node();

};

class Array {

Node\* \_pHead;

Node\* \_pTail;

int \_iSize;

public:

Array(int iSize = 0);

Array(const Array& array);

~Array();

int getSize();

void push\_back(std::string strAdd);

std::string Pop();

std::string Peek(int i);

bool isEmpty();

};

**DataTime.h**

std::string getTime();

**tools.h**

extern HANDLE hConsole;

extern CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

extern WORD defaultWordAttributes;

extern int width;

extern int height;

void initRandomSeed();

int getRandNum();

void alignTextPositionWidth(std::string text, float n);

void alignTextPositionHeight(float n);

void getDefaultAttribute();

void resetColor();

void textColor(int n);

**UI.h**

class UI

{

public:

UI();

~UI();

void renderWelcomeScreen();

void renderGoBackToMainMenu\_USER(User &usr);

void renderLoginScreen(int choice = 0);

void renderMenuScreen\_USER(User &usr);

void renderAccountDetail\_USER(User &usr);

void renderWithDrawScreen\_USER(User &usr);

void renderTransferMoneyScreen\_USER(User &usr);

void renderTransactionDetailsScreen\_USER(User &usr);

void renderChangePasswordScreen\_USER(User &usr);

void renderGoBackToMainMenu\_ADMIN(Admin &admin);

void renderMenuScreen\_ADMIN(Admin &admin);

void renderAccountList\_ADMIN(Admin &admin);

void renderAddAccount\_ADMIN(Admin &admin);

void renderRemoveAccount\_ADMIN(Admin &admin);

void renderUnlockAccount\_ADMIN(Admin &admin);

};

# Phân công công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Họ tên | Công việc được giao | Khó khăn trong khi thực hiện | Hướng giải quyết | Mức độ hoàn thành |
| Liêu Lâm Tài  (48.01.104.116) | Viết chương trình  Kiểm chứng lại chương trình | Không tìm thấy file  Compiled nhiều file cùng 1 lúc bị xung đột thư | Tìm hiểu trên mạng rồi tìm cách để file vào đúng đường dẫn  Bị xung đột thư viện thì check lại từng file | 100% |
| Nguyễn Đức Điềm  (49.01.104.032) | Kiểm chứng lại chương trình  Video demo  bug/debug | Khó khăn chưa sửa lỗi được *LinkedList* | Tham khảo *learn.microsoft.com, LinkedList<T> Class, Microsoft [3]* | 100% |
| Nguyễn Hoàng Đại Phú  (49.01.103.063) | Viết chương trình  Xây dựng danh sách liên kết  Video demo | Ban đầu chưa xây dựng được danh sách liên kết còn sử dụng thư viện STL | Tham khảo giáo trình cấu trúc dữ liệu và thuật toán [1],[2] | 100% |
| Dương Thị Hoài Anh  (49.01.103.002) | Thiết kế giao diện  Kiểm chứng lại chương trình  Viết báo cáo | Ban đầu chưa thiết kế được giao diện | Tham khảo các giao diện có trên Internet | 100% |

# Sản phẩm

***Màn hình chính của máy ATM yêu cầu:*** *đăng nhập với tư cách quản trị viên hoặc người dùng (Chọn 0 để tiến hành đăng nhập với tư cách người dùng hoặc chọn 1 để đăng nhập với tư cách ADMIN).*

***A screen shot of a computer

Description automatically generated***

***Màn hình đăng nhập của user:*** *Tài khoản người dùng được lưu ở file TheTu.txt .Tiến hành nhập ID và mã Pin. Với tài khoản mới tạo mã PIN sẽ mặc định là 123456.* A screenshot of a computer

Description automatically generated

***Khi đăng nhập không đúng mã PIN hoặc sai ID màn hình sẽ hiển thị.***

A screenshot of a computer

Description automatically generated

***Khi đăng nhập sai quá 3 lần tài khoản sẽ bị khóa****: Dữ liệu sẽ được cập nhật vào file ID.txt (49011030021234.txt). Lúc này cần liên hệ ADMIN để mở lại tài khoản.*

***A screenshot of a computer

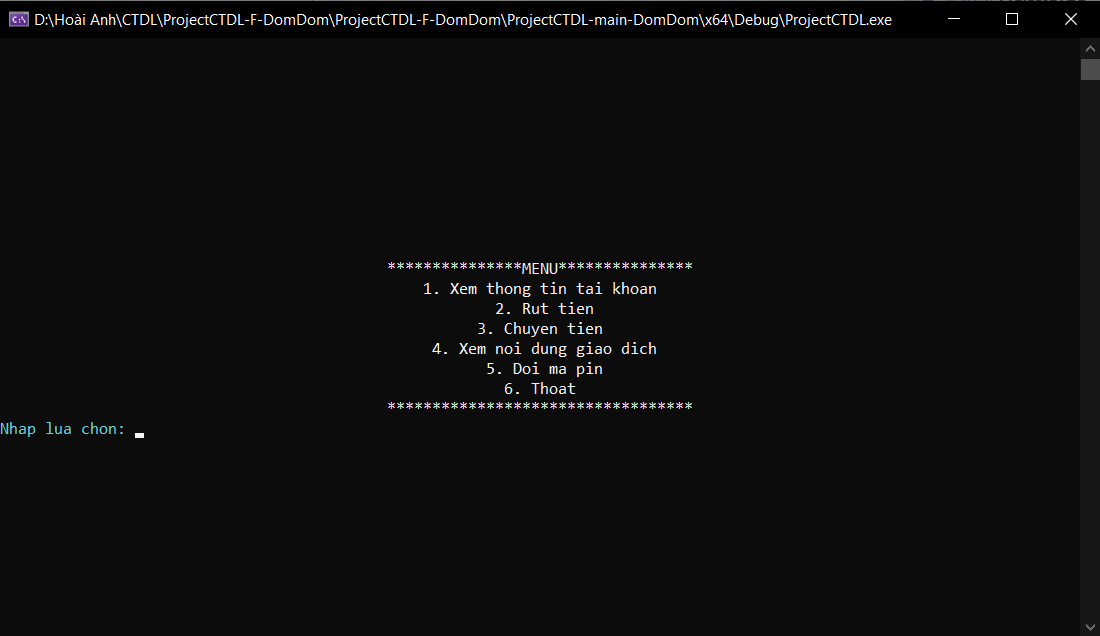
Description automatically generated***

***Khi đăng nhập thành công hệ thống sẽ yêu cầu người dùng tiến hành đổi mã PIN:*** *Lúc này mật khẩu mới sẽ được cập nhật trong file TheTu.txt .Sau đó tiến hành đăng nhập lại.*

A computer screen with white text

Description automatically generated

***Khi đăng nhập thành công hệ thống sẽ đưa người dùng đến Menu của User:*** *Ở đây người dùng có thể chọn một số từ số 1 đến 6 để phục vục nhu cầu của mình*



***Chọn số 1 hệ thống đưa người dùng đến màn hình thông tin tài khoản:*** *Thông tin tài khoản người dùng được lưu ở file ID.txt (49011030021234.txt). Chọn 1 để quay trở về Menu chính hoặc 0 để thoát khỏi chương trình.* A screen shot of a computer

Description automatically generated

***Chọn số 2 hệ thống đưa người dùng đến màn hình rút tiền:*** *Khi rút tiền xong người file ID.txt (49011030021234.txt) sẽ được cập nhập dữ liệu mới. Nếu giá trị muốn rút (<50000) sẽ phải nhập lại.*

A computer screen with text on it

Description automatically generated

*Nếu ghi đúng giá trị, việc rút tiền thành công.*

A computer screen with text and images

Description automatically generated

***Chọn số 3 hệ thống đưa người dùng đến màn hình chuyển tiền:*** *Người dùng cần chuyển tiền từ 50000 trở lên thì hệ thống mới chập nhận yêu cầu.* A screenshot of a computer

Description automatically generated

***Khi chuyển tiền thành công:*** *Dữ liệu người chuyển (49011030021234.txt) và người nhận (49011030025678.txt) sẽ được cập nhật dữ liệu mới.*

***A screenshot of a computer

Description automatically generated***

***Chọn số 4 hệ thống đưa người dùng đến màn hình lịch sử giao dịch:*** *Ở đây người dùng có thể xem lịch sử giao dịch của mình từ trước đến nay. Dữ liệu được lưu ở file LichSuID.txt(LichSu49011030021234.txt).* SA computer screen with white text

Description automatically generated

***Chọn số 5 hệ thống đưa người dùng có thể tiến hành đổi mật khẩu của mình:*** *Dữ liệu sẽ được cập nhật vào file TheTu.txt. Sau đó người dùng tiến hành đăng nhập lại với mật khẩu mới.* *A computer screen shot of a black screen

Description automatically generated*

***Chọn số 6 để thoát khoải chương trình***

*A screenshot of a computer

Description automatically generated*

***Màn hình đăng nhập của 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.*

***A screen shot of a computer

Description automatically generated***

***Màn hình ADMIN khi đăng nhập thành công:*** *Ở đây ADMIN có thể chọn một số từ số 1 đến 5 để phục vục nhu cầu của mình*

A screen shot of a computer

Description automatically generated

***Chọn số 1 để xem danh sách tài khoản của tất cả người dùng :*** *Dữ liệu được lưu ở file TheTu.txt.*

***A computer screen with white text

Description automatically generated***

***Chọn số 2 để thêm tài khoản cho người dùng:*** *Khi thêm tài khoản mới thành công dữ liệu được cập nhật vào file TheTu.txt với mật khẩu mặc định là 123456.*

A screenshot of a computer

Description automatically generated

***Chọn số 3 để xóa tài khoản của người dùng:*** *Khi xóa thành công tài khoản chứa ID yêu cầu file TheTu.txt sẽ cập nhật dữ liệu mới, tài khoản được yêu cầu xóa sẽ biến mất.*

A screenshot of a computer

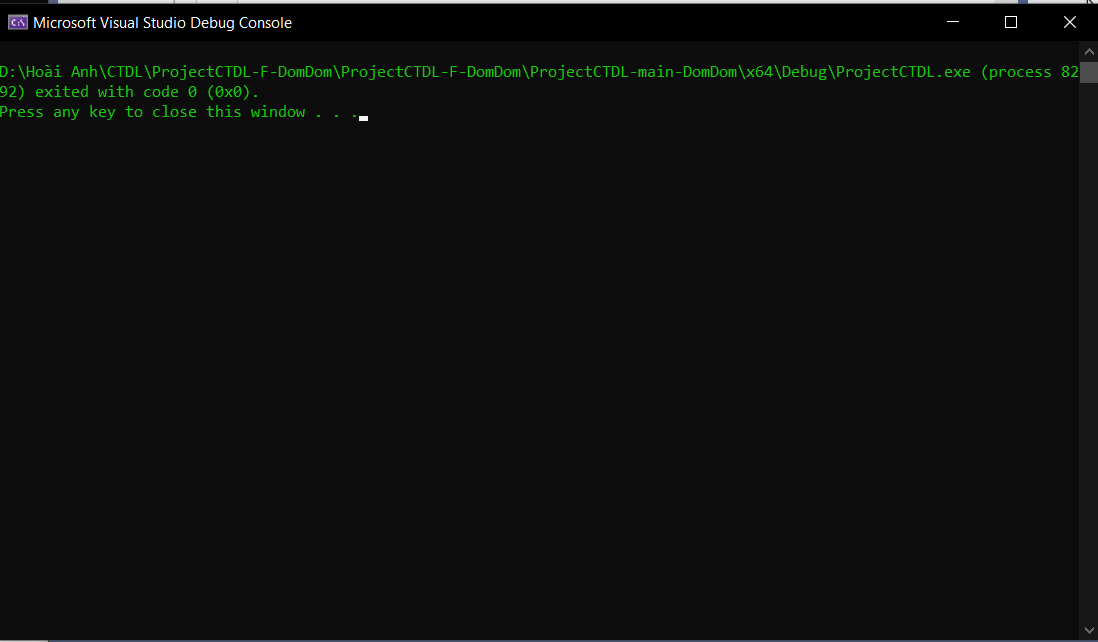
Description automatically generated

***Chọn số 4 để mở các tài khoản đã bị khóa:*** *Lúc này file ID.txt (49011030021234.txt) đã cập nhật dữ liệu mới. Tài khoản chứa ID trên đã được mở khóa và có thể sử dụng lại như bình thường.*

A screenshot of a computer

Description automatically generated

***Chọn số 5 để thoát khỏi chương trình***

****

# Tham khảo

*[1] N. V. Hưng, "Cấu trúc dữ liệu," NXB Đại học Sư Phạm, TP.HCM.*

*[2] learn.microsoft.com, LinkedList<T> Class, Microsoft*

[*https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1?view=net-8.0*](https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1?view=net-8.0)

# Link video demo

***Link video demo (Youtube) :*** [Link video demo project 1 nhom Đom Đóm](https://youtu.be/DjVD_9zsdvg?si=dKCDkwgC8n5Kdxki)