#include <iostream>

#include <iomanip>

#include <fstream>

#include <string>

#include <cstring>

using namespace std;

//function for account system

bool Detect(string);

void Login(string&, string&, double&, bool&, bool&);

void Registration();

//function for banking system

void BankMenu(string, double);

void Deposit(double&, double&);

void WithDraw(double&, double&);

int main()

{

string userID, password, rewriteText, detect\_userID, detect\_password, detect\_userBalance;

int choice, line = 0, changesLine = 0, detectLine = 1;

double balance = {};

bool isLogin = false, exit = false;

//Account System

while (exit != true)

{

cout << "\n\n\tWelcome to KAC BANK!\n"

<< "====================================\n"

<< "\t MENU\n\n"

<< "| 1. LOGIN |\n"

<< "| 2. REGISTER |\n"

<< "| 3. EXIT |\n"

<< "\nPlease enter your choice: ";

cin >> choice;

cout << endl;

if (cin.fail())

{

cin.clear();

cin.ignore();

system("cls");

continue;

}

switch (choice)

{

case 1:

Login(userID, password, balance, exit, isLogin);

break;

case 2:

Registration();

break;

case 3:

cout << "Thank you for using our service!\n\n";

exit = true;

break;

default:

system("cls");

cout << "Please select according the menu !";

break;

}

}

//Banking System

exit = false;

double initialDeposit = 0.0, userDeposit = 0.0, userWithdraw = 0.0;

initialDeposit = balance;

while (exit != true)

{

BankMenu(userID, balance);

cin >> choice;

if (cin.fail())

{

cin.clear();

cin.ignore();

system("cls");

continue;

}

switch (choice)

{

case 1:

Deposit(balance, userDeposit);

break;

case 2:

WithDraw(balance, userWithdraw);

break;

case 3:

exit = true;

break;

default:

system("cls");

cout << "Please select according the menu !";

}

}

//Receipt (Modify)

ofstream receiptFile("receipt.txt");

receiptFile << "KAC BANK RECEIPT" << endl;

receiptFile << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

receiptFile << "Name: " << userID << endl;

receiptFile << "Initial Deposit: RM" << initialDeposit << endl;

receiptFile << "Deposit: RM" << userDeposit << endl;

receiptFile << "Wtihdraw: RM" << userWithdraw << endl;

receiptFile << "Balance: RM" << balance << endl;

receiptFile.close();

/\*Rewrite file\*/

//Detect the user's account location(lines)

ifstream inputFile("account.txt");

while (inputFile >> detect\_userID >> detect\_password >> detect\_userBalance) {

line++;

if (detect\_userID == userID && detect\_password == password) {

changesLine = line;

system("cls");

}

}

inputFile.close();

//Stores user's account and balance from account file to temp file

ifstream readAccountFile("account.txt");

ofstream writeAccountFileToTempFile("accountTemp.txt");

if (readAccountFile && writeAccountFileToTempFile) {

while (getline(readAccountFile, rewriteText))

{

if (detectLine == changesLine) {

writeAccountFileToTempFile << userID << ' ' << password << ' ' << balance << endl;

}

else

{

writeAccountFileToTempFile << rewriteText << '\n';

}

detectLine++;

}

}

readAccountFile.close();

writeAccountFileToTempFile.close();

//Stores user's account and balance from temp file to account file

ifstream readTempFile("accountTemp.txt");

ofstream writeTempFileToAccountFile("account.txt");

if (readTempFile && writeTempFileToAccountFile) {

while (getline(readTempFile, rewriteText))

{

writeTempFileToAccountFile << rewriteText << '\n';

}

}

cout << "Thank you for using our service!\n\n";

}

//Use for detect whether there's a space inside the string (Account System)

bool detectSpace(string check)

{

bool detect = false;

for (char character : check) {

if (character == ' ') {

detect = true;

return detect;

}

}

}

//Login system (Account System)

void Login(string& userID, string& password, double& balance, bool& exit, bool& isLogin)

{

bool valid = false;

string id, pass;

double accountBalance;

system("cls");

cout << "\n\n\t\tPlease enter the username and password: \n"

<< "username: ";

cin >> userID;

cout << "password: ";

cin >> password;

ifstream inputFile("account.txt");

while (inputFile >> id >> pass >> accountBalance) {

if (id == userID && pass == password) {

balance = accountBalance;

valid = true;

system("cls");

}

}

inputFile.close();

if (valid == 1) {

isLogin = true;

exit = true;

system("cls");

}

else {

system("cls");

cout << "Login error, Username/Password is invalid";

}

}

//Register an account (Account System)

void Registration()

{

bool detect = false;

string r\_userID, r\_password;

double initialDeposit;

system("cls");

cin.ignore();

//Register username

while (true)

{

cout << "\n\n\nEnter the username: \n";

getline(cin, r\_userID);

cout << "Enter the password: \n";

getline(cin, r\_password);

if (detectSpace(r\_userID) == 1 || detectSpace(r\_password) == 1)

{

system("cls");

cin.clear();

cout << "Username/Password cannot have spacebar\n"

<< "Please enter again the username and password: ";

continue;

}

else

{

system("cls");

cin.clear();

break;

}

}

//Register password

while (true) {

cout << "\nPlease enter number for your initial deposit\n\n"

<< "Initial deposit: \n";

cin >> initialDeposit;

if (cin.fail())

{

cin.clear();

cin.ignore();

system("cls");

continue;

}

else

{

if (initialDeposit < 500) {

system("cls");

cout << "Initial deposit cannot below than RM500";

continue;

}

else {

break;

}

}

}

//Save the username and password into account file

ofstream openFile("account.txt", ios::app);

openFile << r\_userID << ' ' << r\_password << ' ' << initialDeposit << endl;

system("cls");

cout << "Registration is successful! \n\n";

openFile.close();

}

//Print out Bank Menu (Banking System)

void BankMenu(string userID, double balance)

{

cout << "\n\n\tWelcome " << userID << endl

<< "====================================\n"

<< "\t MENU\n\n"

<< "| 1. Deposit |\n"

<< "| 2. Withdraw |\n"

<< "| 3. Exit |\n"

<< "Current balance: RM" << setprecision(2) << fixed << balance << endl

<< "\nPlease enter your choice: ";

}

//Deposit amount of money to user's account (Banking System)

void Deposit(double& balance, double &deposit)

{

double amount;

system("cls");

while (true) {

cout << "\n\nPlease insert the amount to deposit: ";

cin >> amount;

if (cin.fail())

{

cin.clear();

cin.ignore();

continue;

}

else if (amount < 0)

{

system("cls");

cout << "Negative is not allowed";

continue;

}

else

{

system("cls");

balance += amount;

deposit += amount;

break;

}

}

}

//Withdraw amount of money from user's account (Banking System)

void WithDraw(double& balance, double& withdraw)

{

double amount;

system("cls");

while (true)

{

cout << "\n\nPlease insert the amount to withdraw: ";

cin >> amount;

if (cin.fail())

{

cin.clear();

cin.ignore();

continue;

}

else if (balance < amount)

{

system("cls");

cout << "Your balance is not enough!";

continue;

}

else if (amount < 0)

{

system("cls");

cout << "Negative is not allowed";

continue;

}

else

{

system("cls");

balance -= amount;

withdraw -= amount;

break;

}

}

}