**“Online” Banking System Program**

**Name : Daniel Juanda**

**Student ID : 2001586262**

About :

A program that provide its user several features, such Checking balance, Funds Transfer, Payments, Mini Statements(which provide user’s history transactions made), and Cancel.

Separated into 2 program. Client and Admin, Client’s Task is to provide user with read-only tasks, such checking balance, fund transfer, etcs. Admin’s task is to create account and it’s specific file

**Client.cpp**

main() : Program run from main and all method will be the hub of these function.

main\_Menu() : this function shows up at first, which give user to prompt user, choice 1 is login\_Page() and choice number 2 is forgot\_Password().

login\_Page() : this function run if user selected choice ‘1’ from main\_Menu(). In this function user will be prompted to input username and password based on stored data file.

Login\_Validation() : this function serve as validator, a bridge between login\_Page() and account\_Page(), it is accept 2 parameters string username and password and return bool data-type.

forgot\_Password() : If user selected choice ‘2’ from main\_Menu(). User will be directed forgot\_Password() function, here Program will provide user with random- generated-number ticket number for user to contact CS later.

account\_Page() : If username & password input acquired from login\_Page() are true, then proceed to this function. At Here, user will be prompted to 1-6 choices available to choose.

check\_Balance() : this selection choice check and display user’s balance by reading instance file.

Fund\_Transfer() : this function let the user to transfer fund to another registered account user.

Select\_Payment() : select\_Payment function contain choices to user, user may select available choice provided.

Exit\_Select() : terminate program as the function called.

Buy\_Voucher() : choice available in select\_Payment(), this function allow user to buy ‘prepaid voucher’ which directly deduct user’s balance.

**Admin.cpp**

Admin() : this another console app is only for write-only user’s ID, username, password and other details. Show available menu for user to select.

Login\_page() : this function prompt user to enter admin’s username and password.

Add\_Account() : this selection allows admin to create account data.

Search\_Data() : this choice will provide the record of saved account IDs such full-name, address, username, and balance;

Exit\_Select() : program will terminate if user select this choice.

**AccountData.h**

Header file which contains classes needed for .cpp files and also contains some of header files for .cpp files.

**Problem overcomed**

* Designing program. At first when wanted to start this program, Author find trouble which part or where to start at first, but it is become easier after author decided to make a visual diagram (hierarchy chart) everything is organized by creating Chart for it.
* Solving logical problems. In order to create something run as expected it is mandatory to have a good problem-solving skills to make a program works to do certain specific assigned-job. Author find it hard at first, because maybe author still not get used to it, but throughout the project work, it is becoming ease.
* Creating multiple username and password login data in file C++, biggest problem, which stuck this program development progress for weeks. But after doing some research through books, forum and other sources. It is now solved.

**HIERARCHY CHART**

**Coding Section**

**Final Project**

*//Daniel Juanda // 2001586262 // // //  
//HUMAN MALE*#include **<iostream>**#include **<iomanip>**#include **<cstdlib>**#include **<conio.h>**#include **<sstream>**#include **"accountData.h" // call header file accountData.h which contain struct and link to here.  
  
void** main\_Menu();  
**bool** login\_validation(string usrName, string pin);  
**void** login\_page();  
**void** forgot\_password();  
**void** clearScreen();  
**void** account\_Page();  
**void** check\_Balance();  
**void** fund\_Transfer();  
**void** select\_Payment();  
**void** buy\_Voucher();  
**int** exit\_Select();  
string myID,myPass;  
  
**int** main2()  
{  
 clearScreen();  
 main\_Menu(); *// call main\_menu();* **return** 0;  
}  
  
**void** main\_Menu()  
{  
 **int** choice;  
 cout << **"-------------------------------------------------------------"** << endl;  
 cout << **" WELCOME TO "** << endl;  
 cout << **" BANK CENTRAL AFRICA "** << endl;  
 cout << **" ONLINE BANKING "** << endl;  
 cout << **"-------------------------------------------------------------"** << endl;  
  
  
 cout << **"1.\t LOGIN"** << endl;  
 cout << **"2.\t FORGOT PASSWORD"** << endl;  
 cout << **"Enter your choice (1-2)"** << endl;  
 cin >> choice;  
 **switch** (choice) {  
 **case** 1: login\_page(); *// if user input '1' then call function login\_page* **break**;  
 **case** 2: forgot\_password(); *// if user input '2' then call function forgot\_password* **break**;  
 **default** :main\_Menu();  
  
 }  
}  
  
**void** login\_page()  
{  
 string username, password;  
 clearScreen(); *// call clear screen* **bool** validateLogin; *// set initial value of validateLogin as false  
 // declare string variables as input for argument login\_Validatio* **do**{ *// do while syntax .. .* cout << **"Enter Username : \n"**; *// prompt user to input username* cin >> username; *// store nameLogin string* cin.ignore();  
 cout << **"Enter Password : \n"**; *// prompt user to input username* cin >> password; *// store passLogin string* cin.ignore();  
 validateLogin = login\_validation(username,password);  
  
 **if**(validateLogin) *// if validateLogin accept true value then proceed* {  
 clearScreen(); *// clear screen* account\_Page(); *// go to account\_Page function* }  
 **else** cout << **"INVALID INPUT. ENTER CORRECT USERNAME/PASSWORD!.\n"**; *//display if login input is false* }**while**(!validateLogin); *// keep repeating loop while login input isn't correctly entered.*}  
  
**void** forgot\_password()  
{  
 string username,id;  
 **bool** goOn=**false**; *// initial value set to false* **int** choice; *// variable to store user's choice* clearScreen(); *// call clearScreen function* **const int** MIN\_VALUE = 1000000; *// set min value of random-number generated* **const int** MAX\_VALUE = 9999999; *// max value of random-numer generated* **int** ticketNumber; *// declare ticket\_Number variable* **unsigned** seed = (**unsigned int**) time(0); *// seed generated by time* srand(seed);  
  
 ticketNumber = (rand() % (MAX\_VALUE - MIN\_VALUE + 1)) + MIN\_VALUE; *// formula to generate random-number* stringstream ss;  
 ss << ticketNumber;  
 string ticket;  
 ss >> ticket;  
 cout << **"--------------------------------------------\n"**; *// display* cout << **" WE'VE MADE YOU A TICKET \n"**; *// display* cout << **" Ticket Number : "** << ticketNumber << endl; *// display* cout << **" STATUS : OPEN \n"**;  
 cout << **" PLEASE KEEP THIS TICKET. \n"**; *// display* cout << **"--------------------------------------------\n"**; *// display* cout << **"ENTER USERNAME : "**;  
 cin >> username;  
 cout << **"ENTER ACCOUNT ID : "**;  
 cin >> id;  
 fstream make;  
 make.open(**"ticket\_"**+ticket+**".txt"**,ios::out);  
 make << **"USERNAME :"**<< username << endl  
 << **"ACCOUNT ID : "**<< id << endl  
 << **"TICKET NUMBER : "**<< ticket << endl;  
  
 **do** { *// do start* cout << **"Enter 1 to Main Menu"** << endl; *// display choice 1* cout << **"Enter 2 to Exit"** << endl; *// display choice 2* cin >> choice; *// prompt user to enter choice;* **switch** (choice){ *// switch statement start* **case** 1 : { *// case 1* goOn = **true**; *//* clearScreen(); *// clear screen func* main\_Menu(); *// go to main\_Menu()* **break**; *// break statement if case 1 fulfill* }  
 **case** 2 :{ *// case 2* cout << **"Exit Now."**; *// display exit* **break**; *// break statement ifcase 2 fulfill* }  
  
 **default** : { *// default switch* cout <<**"INVALID INPUT. ENTER 1 OR 2 ONLY!."**; *// display* **break**; *// break* }  
 }  
  
 }**while** (goOn);  
}  
  
**bool** login\_validation(string usrName, string pass) {  
 AccountData user;  
 **bool** valid = **false**;  
 string getUsrname, getPass, getID;  
 ifstream data;  
 data.open(**"dataLogin.txt"**);  
 **while** (data >> getUsrname >> getPass >> getID) {  
 **if** (usrName == getUsrname && pass == getPass) {  
 valid = **true**;  
 getPass = myPass;  
 **break**;  
 }  
 }  
 myID = getID;  
 **return** valid;  
  
}  
  
**void** clearScreen() *//function to clear the screen*{  
 cout << string( 100, **'\n'** );  
}  
  
**void** account\_Page(){ *// start func account\_Page()* clearScreen();  
 AccountData user;  
 **double** balance;  
 string firstName,lastName,username,address;  
 ifstream userAccount;  
 userAccount.open(myID+**".txt"**);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
 user.setFirstName(firstName);  
 user.setLastName(lastName);  
 user.setUserName(username);  
 user.setAddress(address);  
 user.setBalance(balance);  
 clearScreen();  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"Hi, "** << firstName << endl;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << **"---------------------------------------------------\n"**;  
  
 cout << myPass;  
 cout << **"1. CHECK BALANCE \n"**;  
 cout << **"2. FUNDS TRANSFER \n"**;  
 cout << **"3. SELECT PAYMENTS \n"**;  
 cout << **"4. LOG OUT\n"**;  
 cout << **"5. EXIT \n"**;  
  
 cout << **"Enter your choice: \n"**;  
 **int** select;  
 cin >> select;  
 **while**(select<0||select>6) {  
 cout << **"INVALID INPUT"**;  
 cin >> select;  
 }  
 **switch**(select){  
 **case** 1: check\_Balance();  
 **break**;  
 **case** 2: fund\_Transfer();  
 **break**;  
 **case** 3: select\_Payment();  
 **break**;  
 **case** 4: clearScreen();main\_Menu();  
 **break**;  
 **case** 5: exit\_Select();  
 **break**;  
 **default**:account\_Page();  
 }  
 userAccount.close();  
}  
  
**void** check\_Balance() {  
 string firstName,lastName,username,address;  
 AccountData user;  
 **double** balance;  
 fstream userAccount;  
 userAccount.open(myID+**".txt"**,ios::in);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
  
 clearScreen();  
 cout << **"---------------------------------------------------\n"**;  
 cout << **" YOUR BALANCE \n "**;  
 cout << **"ACCOUNT ID : "** << myID;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << **"\t\t"** << **"$"**<< balance << endl;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"PRESS Y TO RETURN MAIN MENU.. PRESS N TO EXIT PROGRAM\n"**;  
 **int** key = getch();  
 **if**(key==**'y'**||key==**'Y'**){  
 account\_Page();  
 }  
 **else** {  
 exit\_Select();  
 }  
  
}  
*// end func account\_Page()***void** fund\_Transfer() {  
 clearScreen();  
 AccountData user;  
 **double** amount, balance,destBalance;  
 **int** c;  
 string firstName, lastName, username, address;  
 fstream userAccount;  
 string myFileName = myID + **".txt"**;  
 userAccount.open(myFileName, ios::in);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
 userAccount.close();  
 string destID, destFirstName, destLastName, destUserName, destAddress;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **" FUND TRANSFER \n"**;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"ENTER AMOUNT TO TRANSFER : "** << endl;  
 cin >> amount;  
 **while** (amount >= balance) {  
 cout << **"INSUFFICIENT FUND!"**;  
 cin >> amount;  
 }  
  
 clearScreen();  
 cout << **"ENTER DESTINATION ACCOUNT ID : "** << endl;  
 cin >> destID;  
 ifstream dest;  
 string destFileName = destID + **".txt"**;  
 dest.open(destFileName);  
 dest >> destFirstName >> destLastName >> destUserName >> destAddress >> destBalance;  
 string destFullName = destFirstName + **" "** + destLastName;  
 **if** (dest.is\_open()) {  
 dest.close();  
 cout << **"---------------------------------------------------\n"**;  
 cout << **" FUND TRANSFER \n"**;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"ACCOUNT ID RECEIVER : "** << destID << endl;  
 cout << **"ACCOUNT NAME : "** << destFullName << endl;  
 cout << **"TRANSFER AMOUNT : $"** << amount << endl;  
  
 cout << **"DO YOU WISH TO PROCEED ?"** << endl;  
 cout << **"PRESS Y TO CONTINUE..."** << endl;  
 cout << **"PRESS N TO CANCEL"**;  
 c = getch();  
 **if** (c == **'y'** || c == **'Y'**) {  
 balance-=amount;  
 userAccount.open(myFileName, ios::out);  
 userAccount << firstName << endl  
 << lastName << endl  
 << username << endl  
 << address << endl  
 << balance << endl;  
  
 destBalance += amount;  
 fstream destNew;  
 destNew.open(destFileName,ios::out);  
 destNew << destFirstName << endl  
 << destLastName << endl  
 << destUserName << endl  
 << destAddress << endl  
 << destBalance << endl;  
 destNew.close();  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"TRANSACTION SUCCESS\n"**;  
 cout << **"TRANSFER DETAILS "** << endl;  
 cout << endl;  
 cout << **"RECEIVER NAME : "** << destFullName << endl;  
 cout << **"AMOUNT TRANSFERED : "** << amount << endl;  
 cout << **"---------------------------------------------------\n"**;  
  
 } **else if** (c == **'n'** || c == **'N'**) {  
 fund\_Transfer();  
 }  
 } **else** {  
 cout << **"ACCOUNT ID NOT FOUND!"** << endl;  
 }  
 cout << **"ENTER 1 TO MENU"** << endl;  
 cout << **"ENTER 2 TO EXIT "** << endl;  
 **int** enter;  
 cin >> enter;  
 **while** (enter < 0 || enter > 2) {  
 cout << **"ENTER 1 OR 2 ONLY!"**;  
 cin >> enter;  
 }  
 **switch** (enter) {  
 **case** 1:  
 account\_Page();  
 **break**;  
 **case** 2:  
 exit\_Select();  
 **break**;  
 **default**:  
 cout << **"ERROR"**;  
 **break**;  
 }  
}  
**void** select\_Payment(){  
 clearScreen();  
 clearScreen();  
 **int** choice;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **" BCA ONLINE PAYMENT"** << endl;  
 cout << endl;  
 cout << endl;  
 cout <<**"----------------------------------------------------\n"**;  
 cout << **"1. PREPAID VOUCHER\n"**;  
 cout << **"2. PAY ELECTRICITY\n"**;  
 cout << **"3. GO-PAY\n"**;  
 cout << **"Enter choice : "**;  
 cin >> choice;  
 **switch**(choice){  
 **case** 1: buy\_Voucher();  
 **break**;  
 **case** 2: cout <<**"UNDER MAINTENANCE"**;  
 **break**;  
 **case** 3: cout << **"UNDER MAINTENANCE"**;  
 **break**;  
 **default**:select\_Payment();  
 }  
}  
  
**void** buy\_Voucher(){  
 clearScreen();  
 **int** choice;  
 string phoneNum;  
 **double** amount = 0, balance = 0;  
 cout << **"---------------------------------------------------\n"**;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << **"SELECT OPERATOR : \n"**;  
 cout << **"1.TELKOMSEL"** << endl;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"ENTER CHOICE: "**<< endl;  
 cin >> choice;  
 **if**(choice==1){  
 clearScreen();  
 cout << **"1. $2\n"**;  
 cout << **"2. $5\n"**;  
 cout << **"3. $10\n"**;  
 cout << **"SELECT CHOICE : \n"**;  
 cin >> choice;  
 **switch**(choice){  
 **case** 1: amount=2;  
 **break**;  
 **case** 2 : amount=5;  
 **break**;  
 **case** 3: amount=10;  
 **break**;  
 **default**:buy\_Voucher();  
 }  
  
 cout << **"ENTER MOBILE PHONE NUMBER : \n"**;  
 cin >> phoneNum;  
  
 clearScreen();  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"PREPAID VOUCHER DETAILS : "**<<endl;  
 cout << endl;  
 cout << endl;  
 cout << **"MOBILE PHONE : "** << phoneNum << endl;  
 cout << **"AMOUNT : $"** << amount << endl;  
 cout << **"$"** << amount << **" WILL BE DEDUCTED FROM YOUR ACCOUNT"**<< endl;  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"PRESS Y TO CONFIRM.. PRESS N TO CANCEL"** << endl;  
 **int** yn=getch();  
 **if**(yn==**'y'**||yn==**'Y'**){  
 clearScreen();  
 string firstName, lastName, username, address;  
 string myFileName = myID + **".txt"**;  
 fstream userAccount;  
 userAccount.open(myFileName,ios::in);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
 userAccount.close();  
  
 userAccount.open(myFileName, ios::out);  
 balance -= amount;  
 userAccount << firstName << endl  
 << lastName << endl  
 << username << endl  
 << address << endl  
 << balance<< endl;  
 userAccount.close();  
 cout << **"---------------------------------------------------\n"**;  
 cout << **"TRANSACTION SUCCESS"** << endl;  
 cout << endl;  
 cout << **"MOBILE PHONE NUMBER : "** << phoneNum << endl;  
 cout << **"$"** << amount << **"HAS BEEN DEDUCTED FROM YOUR ACCOUNT"** << endl;  
 cout << endl;  
 cout << **"PRESS Y TO RETURN MAIN PAGE.. PRESS N TO EXIT PROGRAM\n"**;  
 **int** key = getch();  
 **if**(key==**'y'**||key==**'Y'**){  
 account\_Page();  
 }  
 **else**{  
 exit\_Select();  
 }  
 }  
 **else if**(yn==**'n'**||yn==**'N'**){  
 account\_Page();  
 }  
  
  
  
  
 }  
  
}  
  
  
**int** exit\_Select(){  
 cout << **"Exit Now..."**;  
 **return** 0;  
  
}

**Admin.cpp**

//Daniel Juanda // 2001586262 // // //  
//HUMAN MALE  
  
#include <iostream>  
#include <iomanip>  
#include <cstdlib>  
#include <conio.h>  
#include <sstream>  
#include "accountData.h" // call header file accountData.h which contain struct and link to here.  
  
void main\_Menu();  
bool login\_validation(string usrName, string pin);  
void login\_page();  
void forgot\_password();  
void clearScreen();  
void account\_Page();  
void check\_Balance();  
void fund\_Transfer();  
void select\_Payment();  
void buy\_Voucher();  
int exit\_Select();  
string myID,myPass;  
  
int main2()  
{  
 clearScreen();  
 main\_Menu(); // call main\_menu();  
  
  
 return 0;  
}  
  
void main\_Menu()  
{  
 int choice;  
 cout << "-------------------------------------------------------------" << endl;  
 cout << " WELCOME TO " << endl;  
 cout << " BANK CENTRAL AFRICA " << endl;  
 cout << " ONLINE BANKING " << endl;  
 cout << "-------------------------------------------------------------" << endl;  
  
  
 cout << "1.\t LOGIN" << endl;  
 cout << "2.\t FORGOT PASSWORD" << endl;  
 cout << "Enter your choice (1-2)" << endl;  
 cin >> choice;  
 switch (choice) {  
 case 1: login\_page(); // if user input '1' then call function login\_page  
 break;  
 case 2: forgot\_password(); // if user input '2' then call function forgot\_password  
 break;  
 default :main\_Menu();  
  
 }  
}  
  
void login\_page()  
{  
 string username, password;  
 clearScreen(); // call clear screen  
 bool validateLogin; // set initial value of validateLogin as false  
 // declare string variables as input for argument login\_Validatio  
 do{ // do while syntax .. .  
 cout << "Enter Username : \n"; // prompt user to input username  
 cin >> username; // store nameLogin string  
 cin.ignore();  
 cout << "Enter Password : \n"; // prompt user to input username  
 cin >> password; // store passLogin string  
 cin.ignore();  
 validateLogin = login\_validation(username,password);  
  
 if(validateLogin) // if validateLogin accept true value then proceed  
 {  
 clearScreen(); // clear screen  
 account\_Page(); // go to account\_Page function  
 }  
 else  
 cout << "INVALID INPUT. ENTER CORRECT USERNAME/PASSWORD!.\n"; //display if login input is false  
  
 }while(!validateLogin); // keep repeating loop while login input isn't correctly entered.  
  
}  
  
void forgot\_password()  
{  
 string username,id;  
 bool goOn=false; // initial value set to false  
 int choice; // variable to store user's choice  
 clearScreen(); // call clearScreen function  
 const int MIN\_VALUE = 1000000; // set min value of random-number generated  
 const int MAX\_VALUE = 9999999; // max value of random-numer generated  
 int ticketNumber; // declare ticket\_Number variable  
 unsigned seed = (unsigned int) time(0); // seed generated by time  
  
 srand(seed);  
  
 ticketNumber = (rand() % (MAX\_VALUE - MIN\_VALUE + 1)) + MIN\_VALUE; // formula to generate random-number  
 stringstream ss;  
 ss << ticketNumber;  
 string ticket;  
 ss >> ticket;  
 cout << "--------------------------------------------\n"; // display  
 cout << " WE'VE MADE YOU A TICKET \n"; // display  
 cout << " Ticket Number : " << ticketNumber << endl; // display  
 cout << " STATUS : OPEN \n";  
 cout << " PLEASE KEEP THIS TICKET. \n"; // display  
 cout << "--------------------------------------------\n"; // display  
 cout << "ENTER USERNAME : ";  
 cin >> username;  
 cout << "ENTER ACCOUNT ID : ";  
 cin >> id;  
 fstream make;  
 make.open("ticket\_"+ticket+".txt",ios::out);  
 make << "USERNAME :"<< username << endl  
 << "ACCOUNT ID : "<< id << endl  
 << "TICKET NUMBER : "<< ticket << endl;  
  
 do { // do start  
  
 cout << "Enter 1 to Main Menu" << endl; // display choice 1  
 cout << "Enter 2 to Exit" << endl; // display choice 2  
 cin >> choice; // prompt user to enter choice;  
  
 switch (choice){ // switch statement start  
  
 case 1 : { // case 1  
 goOn = true; //  
 clearScreen(); // clear screen func  
 main\_Menu(); // go to main\_Menu()  
 break; // break statement if case 1 fulfill  
 }  
 case 2 :{ // case 2  
 cout << "Exit Now."; // display exit  
 break; // break statement ifcase 2 fulfill  
 }  
  
 default : { // default switch  
 cout <<"INVALID INPUT. ENTER 1 OR 2 ONLY!."; // display  
 break; // break  
 }  
 }  
  
 }while (goOn);  
}  
  
bool login\_validation(string usrName, string pass) {  
 AccountData user;  
 bool valid = false;  
 string getUsrname, getPass, getID;  
 ifstream data;  
 data.open("dataLogin.txt");  
 while (data >> getUsrname >> getPass >> getID) {  
 if (usrName == getUsrname && pass == getPass) {  
 valid = true;  
 getPass = myPass;  
 break;  
 }  
 }  
 myID = getID;  
 return valid;  
  
}  
  
void clearScreen() //function to clear the screen  
{  
 cout << string( 100, '\n' );  
}  
  
void account\_Page(){ // start func account\_Page()  
 clearScreen();  
 AccountData user;  
 double balance;  
 string firstName,lastName,username,address;  
 ifstream userAccount;  
 userAccount.open(myID+".txt");  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
 user.setFirstName(firstName);  
 user.setLastName(lastName);  
 user.setUserName(username);  
 user.setAddress(address);  
 user.setBalance(balance);  
 clearScreen();  
 cout << "---------------------------------------------------\n";  
 cout << "Hi, " << firstName << endl;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << "---------------------------------------------------\n";  
  
 cout << myPass;  
 cout << "1. CHECK BALANCE \n";  
 cout << "2. FUNDS TRANSFER \n";  
 cout << "3. SELECT PAYMENTS \n";  
 cout << "4. LOG OUT\n";  
 cout << "5. EXIT \n";  
  
 cout << "Enter your choice: \n";  
 int select;  
 cin >> select;  
 while(select<0||select>6) {  
 cout << "INVALID INPUT";  
 cin >> select;  
 }  
 switch(select){  
 case 1: check\_Balance();  
 break;  
 case 2: fund\_Transfer();  
 break;  
 case 3: select\_Payment();  
 break;  
 case 4: clearScreen();main\_Menu();  
 break;  
 case 5: exit\_Select();  
 break;  
 default:account\_Page();  
 }  
 userAccount.close();  
}  
  
void check\_Balance() {  
 string firstName,lastName,username,address;  
 AccountData user;  
 double balance;  
 fstream userAccount;  
 userAccount.open(myID+".txt",ios::in);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
  
 clearScreen();  
 cout << "---------------------------------------------------\n";  
 cout << " YOUR BALANCE \n ";  
 cout << "ACCOUNT ID : " << myID;  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << "\t\t" << "$"<< balance << endl;  
 cout << "---------------------------------------------------\n";  
 cout << "PRESS Y TO RETURN MAIN MENU.. PRESS N TO EXIT PROGRAM\n";  
 int key = getch();  
 if(key=='y'||key=='Y'){  
 account\_Page();  
 }  
 else {  
 exit\_Select();  
 }  
  
}  
// end func account\_Page()  
  
  
void fund\_Transfer() {  
 clearScreen();  
 AccountData user;  
 double amount, balance,destBalance;  
 int c;  
 string firstName, lastName, username, address;  
 fstream userAccount;  
 string myFileName = myID + ".txt";  
 userAccount.open(myFileName, ios::in);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
 userAccount.close();  
 string destID, destFirstName, destLastName, destUserName, destAddress;  
 cout << "---------------------------------------------------\n";  
 cout << " FUND TRANSFER \n";  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << "---------------------------------------------------\n";  
 cout << "ENTER AMOUNT TO TRANSFER : " << endl;  
 cin >> amount;  
 while (amount >= balance) {  
 cout << "INSUFFICIENT FUND!";  
 cin >> amount;  
 }  
  
 clearScreen();  
 cout << "ENTER DESTINATION ACCOUNT ID : " << endl;  
 cin >> destID;  
 ifstream dest;  
 string destFileName = destID + ".txt";  
 dest.open(destFileName);  
 dest >> destFirstName >> destLastName >> destUserName >> destAddress >> destBalance;  
 string destFullName = destFirstName + " " + destLastName;  
 if (dest.is\_open()) {  
 dest.close();  
 cout << "---------------------------------------------------\n";  
 cout << " FUND TRANSFER \n";  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << "---------------------------------------------------\n";  
 cout << "ACCOUNT ID RECEIVER : " << destID << endl;  
 cout << "ACCOUNT NAME : " << destFullName << endl;  
 cout << "TRANSFER AMOUNT : $" << amount << endl;  
  
 cout << "DO YOU WISH TO PROCEED ?" << endl;  
 cout << "PRESS Y TO CONTINUE..." << endl;  
 cout << "PRESS N TO CANCEL";  
 c = getch();  
 if (c == 'y' || c == 'Y') {  
 balance-=amount;  
 userAccount.open(myFileName, ios::out);  
 userAccount << firstName << endl  
 << lastName << endl  
 << username << endl  
 << address << endl  
 << balance << endl;  
  
 destBalance += amount;  
 fstream destNew;  
 destNew.open(destFileName,ios::out);  
 destNew << destFirstName << endl  
 << destLastName << endl  
 << destUserName << endl  
 << destAddress << endl  
 << destBalance << endl;  
 destNew.close();  
 cout << "---------------------------------------------------\n";  
 cout << "TRANSACTION SUCCESS\n";  
 cout << "TRANSFER DETAILS " << endl;  
 cout << endl;  
 cout << "RECEIVER NAME : " << destFullName << endl;  
 cout << "AMOUNT TRANSFERED : " << amount << endl;  
 cout << "---------------------------------------------------\n";  
  
 } else if (c == 'n' || c == 'N') {  
 fund\_Transfer();  
 }  
 } else {  
 cout << "ACCOUNT ID NOT FOUND!" << endl;  
 }  
 cout << "ENTER 1 TO MENU" << endl;  
 cout << "ENTER 2 TO EXIT " << endl;  
 int enter;  
 cin >> enter;  
 while (enter < 0 || enter > 2) {  
 cout << "ENTER 1 OR 2 ONLY!";  
 cin >> enter;  
 }  
 switch (enter) {  
 case 1:  
 account\_Page();  
 break;  
 case 2:  
 exit\_Select();  
 break;  
 default:  
 cout << "ERROR";  
 break;  
 }  
}  
void select\_Payment(){  
 clearScreen();  
 clearScreen();  
 int choice;  
 cout << "---------------------------------------------------\n";  
 cout << " BCA ONLINE PAYMENT" << endl;  
 cout << endl;  
 cout << endl;  
 cout <<"----------------------------------------------------\n";  
 cout << "1. PREPAID VOUCHER\n";  
 cout << "2. PAY ELECTRICITY\n";  
 cout << "3. GO-PAY\n";  
 cout << "Enter choice : ";  
 cin >> choice;  
 switch(choice){  
 case 1: buy\_Voucher();  
 break;  
 case 2: cout <<"UNDER MAINTENANCE";  
 break;  
 case 3: cout << "UNDER MAINTENANCE";  
 break;  
 default:select\_Payment();  
 }  
}  
  
void buy\_Voucher(){  
 clearScreen();  
 int choice;  
 string phoneNum;  
 double amount = 0, balance = 0;  
 cout << "---------------------------------------------------\n";  
 cout << endl;  
 cout << endl;  
 cout << endl;  
 cout << "SELECT OPERATOR : \n";  
 cout << "1.TELKOMSEL" << endl;  
 cout << "---------------------------------------------------\n";  
 cout << "ENTER CHOICE: "<< endl;  
 cin >> choice;  
 if(choice==1){  
 clearScreen();  
 cout << "1. $2\n";  
 cout << "2. $5\n";  
 cout << "3. $10\n";  
 cout << "SELECT CHOICE : \n";  
 cin >> choice;  
 switch(choice){  
 case 1: amount=2;  
 break;  
 case 2 : amount=5;  
 break;  
 case 3: amount=10;  
 break;  
 default:buy\_Voucher();  
 }  
  
 cout << "ENTER MOBILE PHONE NUMBER : \n";  
 cin >> phoneNum;  
  
 clearScreen();  
 cout << "---------------------------------------------------\n";  
 cout << "PREPAID VOUCHER DETAILS : "<<endl;  
 cout << endl;  
 cout << endl;  
 cout << "MOBILE PHONE : " << phoneNum << endl;  
 cout << "AMOUNT : $" << amount << endl;  
 cout << "$" << amount << " WILL BE DEDUCTED FROM YOUR ACCOUNT"<< endl;  
 cout << "---------------------------------------------------\n";  
 cout << "PRESS Y TO CONFIRM.. PRESS N TO CANCEL" << endl;  
 int yn=getch();  
 if(yn=='y'||yn=='Y'){  
 clearScreen();  
 string firstName, lastName, username, address;  
 string myFileName = myID + ".txt";  
 fstream userAccount;  
 userAccount.open(myFileName,ios::in);  
 userAccount >> firstName >> lastName >> username >> address >> balance;  
 userAccount.close();  
  
 userAccount.open(myFileName, ios::out);  
 balance -= amount;  
 userAccount << firstName << endl  
 << lastName << endl  
 << username << endl  
 << address << endl  
 << balance<< endl;  
 userAccount.close();  
 cout << "---------------------------------------------------\n";  
 cout << "TRANSACTION SUCCESS" << endl;  
 cout << endl;  
 cout << "MOBILE PHONE NUMBER : " << phoneNum << endl;  
 cout << "$" << amount << "HAS BEEN DEDUCTED FROM YOUR ACCOUNT" << endl;  
 cout << endl;  
 cout << "PRESS Y TO RETURN MAIN PAGE.. PRESS N TO EXIT PROGRAM\n";  
 int key = getch();  
 if(key=='y'||key=='Y'){  
 account\_Page();  
 }  
 else{  
 exit\_Select();  
 }  
 }  
 else if(yn=='n'||yn=='N'){  
 account\_Page();  
 }  
  
  
  
  
 }  
  
}  
  
  
int exit\_Select(){  
 cout << "Exit Now...";  
 return 0;  
  
}