Topic: Online Banking

Group no: MLB_08.02_7

Campus: Malabe

Submission Date: 20/05/2020

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21166860	Sandamal P.G.E.J	0712984706
IT21182600	Dilsara W.N	0761732149
IT21182464	Arachchi R.P.N	0766217123
IT21166624	Nawod G.A.D	0719011062
IT21181610	Wijesinghe K.S.K	0719277356

Introduction

Online banking is one of the most widely used methods today. This has made it possible to carry out daily activities with ease and efficiency. Achieving this service regardless of age is a unique feature. An automated set of instructions can also be used to properly select the required service and proceed with the required task correctly. It also allows you to withdraw money, transfer money to another account, and track account balances correctly. It also has the ability to open a new account and modify user data as needed, as well as automated troubleshooting.

Much of the world today is changing to an online system. Therefore, the purpose of these is to allow a lot of work to be done in less time. Among them, the online banking system has a special place. The uniqueness here is that it is able to carry out all financial activities without any insecurity. These systems are designed to be able to safely reset their personal data even when they have been forgotten

Exercise 1:

1) User Requirements

- Registered user can log in to the system using his/her credentials.
- Unregistered user in an online Banking system needs to first register providing details such as NIC, Name, mobile number, account number, email, address
- A registered customer receives a user ID and ID will be receive through email.
- Customer details are stored in the system.
- If there are some mistakes in the registration details, he/she can request or inform to the system admin.
- The Register User can customize their user profile.
- The registered user can upload profile picture.
- The registered user can create Nick Name.
- The Registered User can view the status of the Saving Account details
- The Registered User can open new Saving Account or Fixed Account.
- The Register user can make payments
- The Register user can request credit card or debit card or both.
- The Register User Can Apply Loan and check Loan details.
- The Register user can file a complain
- The Register user can apply Job vacancy.
- An Unregister user can apply job vacancy.
- A Register User can choose the Account balance, Money transaction and bill payment from the Personal Banking.
- when money transaction user has to provide valid details
- Payment details and account details has to be validated.
- System has to update Account details, and also keep a record of the transaction details

2) Identified Nouns

- User
- Registered User
- Unregistered User
- System
- NIC
- Name
- Mobile number
- Account Number
- Email
- Address
- Customer
- User ID
- Inform
- System admin
- User profile
- Profile picture
- Nick Name
- Saving account
- Fixed account
- Payments
- Credit card
- Debit card
- Loan
- Complain
- Vacancy
- Account balance
- Money transaction
- Bill payment
- Personal banking
- Account Details

Identified classes

- User
- Registered user
- Unregister User
- Saving Account
- Fixed Account
- Bill Payments
- Debit Cards
- Credit Cards
- Loan
- Money Transaction
- Complain
- Vacancies

Reasons to reject other nouns

1.Attribute

- NIC
- Name
- Email
- Address
- Account number
- User ID
- Mobile number
- Account Balance

2.Redundant

- User
- Inform
- User profile
- Account details
- Nick name
- Personal banking

3.Out of Scope

- System
- System admin
- Profile picture

Exercise 1:

CRC Cards for the online Banking system

User	
Responsibility	**Collaborate**
Store Details of User	
Register Details	
Login	Registered User
Edit User Details	
Make Payments and Transaction	Payments, money transaction
Check Account Details	Saving Accounts, Fixed Account
Make Complains	Complain

Registered User	
Responsibility	**Collaborate**
Display login	
Validate Login Details	
Update Account Data	

Unregistered User	
Responsibility	**Collaborate**
Display Sign up	
Create User Account	
Generate Login Details	

Saving Account	
Responsibility	**Collaborate**
Display Account Details	
Store Account Activity	
Calculate Account Balance	Payment, Money Transactions, Loan
Update Account Data	

Fixed Account	
Responsibility	**Collaborate**
Display Account Details	
Store Account Activity	
Update Account Data	

Debit Cards	
Responsibility	**Collaborate**
Validate Payments	
Make Payments	Saving Account

Credit Cards	
Responsibility	**Collaborate**
Validate Payments	

Make payments	Loan

Loan	
Responsibility	**Collaborate**
Calculate Interest	
Update Loan Details	
Display Loan Details	

Money Transaction	
Responsibility	**Collaborate**
Make payment	Saving Account
Validate transactions	

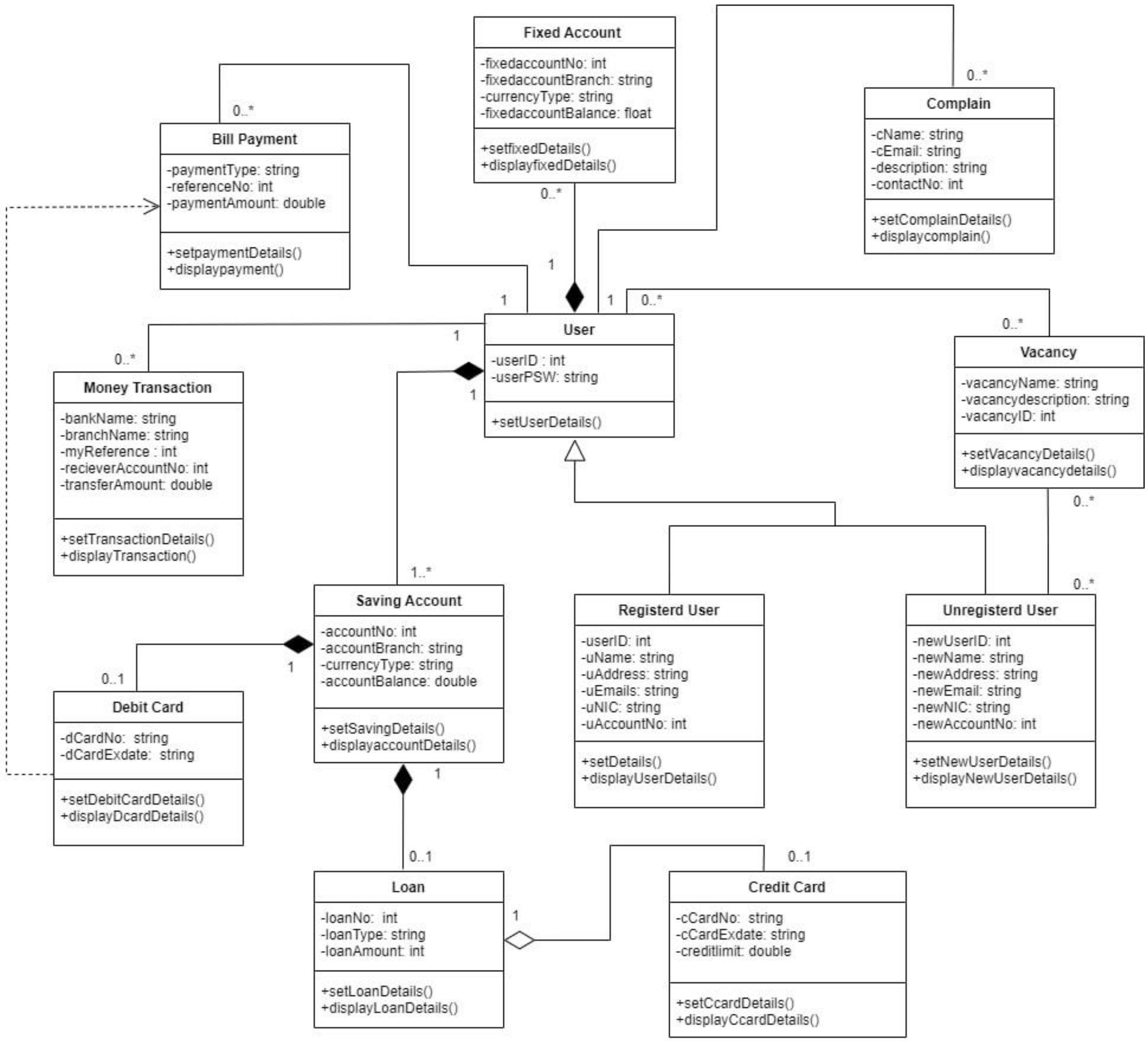
Bill Payment	
Responsibility	**Collaborate**
Keep record all payment	
Validate Payments	

Vacancies	
Responsibility	**Collaborate**
Display Vacancy details	
Update Vacancy details	

Complain	
Responsibility	**Collaborate**
Keep record of all complain	
Responds Complains	

Exercise 2:

Class diagram for the Online Banking System



C++ Coding

```
#include<iostream>
#include<iomanip>
#include<string>

using namespace std;

//Registered User class
class Registered_user
{
   protected:
               int userID;
               string uName;
               string uEmail;
               string uAddress;
               string uNIC;
               int accountNo;

   public:
             void setDetails(int pUserID, string pName, string pEmail, string  pAddress, string  pNIC, int pAccountNo);
             void displayUserDetails();
};

//Unregistered User class
class Unregistered_user
{
   private:
               int newUserID;
               string newName;
               string newEmail;
               string newAddress;
               string newNIC;
               string  newUserPSW;
               int newAccountNo;

   public:
               void setnewuserdetails(int newuserid, string newname, string neweamail, string newaddress, string
               newnic, string newpsw, int newaccountno);
               void displaynewuserdetails();
};

//  Debit card class
class Debit_Card
{
   protected:
               string dCardNo;
               string dCardExdate;
   public:
               void setDebitCardDetails(string dn,string de);
               void displayDcardDetails();
};
```

```
// Loan class
class Loan
{
   private:
            int loanNo;
            string loanType;
            double loanAmount;
   public:
            void setLoanDetails(int ln,string loantype,double la);
            void displayLoanDetails();
};

//  Credit card class
class Credit_Card
{
   private:
            string cCardNo;
            string cCardExdate;
            double creditlimit;
   public:
            void setCcardDetails(string cn,string ce,double climit);
            void displayCcardDetails();
};

 //  Saving Account class
class Saving_Account: public Debit_Card, public Loan, public Credit_Card
{
   protected:
            int saccountNo;
            string accountBranch;
            string currencyType;
            double accountBalance;
   public:
            void setSavingDetails(int accountNo, string abranch,string ctype,double abalance);
            void displayaccountDetails();
};

//  Fixed Account class
class Fixed_Account
{
   protected:
            int fixedaccountNo;
            string fixedaccountBranch;
            string currencyType;
            float fixedaccountBalance;
   public:
            void setfixedDetails(int faccountno,string faccountbranch,string currancytype,float
            fixedaccountbalance);
            void displayfixedDetails();

};
```

```
// Bill Payment  class
class Bill_Payment
{
   protected:
            string paymentType;
            int referenceNo;
            double paymentAmount;
   public:
            void setpaymentDetails(string ptype, int ref, double payamount);
            void displaypayment();
};

//  Money Transaction class
class Money_Transaction
{
   protected:
            string bankName;
            string branchName;
            int myReference;
            int recieverAccountNo;
            double transferAmount;
   public:
            void setTransactionDetails(string bname, string brname, int myref, int raccountno, double tramount);
            void displayTransaction();
};

//  Complain class
class Complain
{
   protected:
            string cName;
            string cEmail;
            string description;
            int contactNo;
   public:
            void setComplainDetails(string cname, string cemail, string des,int cn);
            void displaycomplain();
};

//  User class
class User: public Registered_user, public Saving_Account, public Fixed_Account, public Bill_Payment,
 public Money_Transaction, public Complain
{
protected :
   int userID;
   string userPSW;
public:
   void setUserDetails(int pUserID, string uPSW);
};

//  Vacancy class
class Vacancy
{
   private:
```

```
            string vacancyName;
            string vacancydescription;
            int vacancyID;
    public:
            void setVacancyDetails(string vname, string vdes, int vid );
            void displayvacancyDetails();
};

// Method implementation

// Registered user class method implementation
void Registered_user::setDetails(int pUserID, string pName, string pEmail, string  pAddress, string  pNIC, int
pAccountNo)
{
    userID = pUserID;
    uName=pName;
    uEmail=pEmail;
    uAddress= pAddress;
    uNIC= pNIC;
    accountNo =pAccountNo;
}

void Registered_user::displayUserDetails()
{
    cout << "\nLogin Success!!" <<endl<<endl;
    cout <<  "User ID: " << userID<<endl <<endl;
    cout << "Name: "  <<uName <<endl <<endl;
    cout << "Email: " <<uEmail <<endl <<endl;
    cout << "Address: " <<uAddress <<endl <<endl;
    cout << "NIC : " <<uNIC <<endl <<endl;
    cout << "Account No: " <<accountNo <<endl <<endl;
}

// User class method implementation
void User::setUserDetails(int pUserID, string uPSW)
{
    userID=pUserID;
    userPSW=uPSW;
}

// Unregistered user class method implementation
void Unregistered_user::setnewuserdetails(int newuserid, string newname, string newemail, string newaddress,
string newnic, string newpsw, int newaccountno)
{
    newUserID = newuserid;
    newUserPSW = newpsw;
    newName = newname;
    newEmail = newemail;
    newAddress = newaddress;
    newNIC = newnic;
    newAccountNo = newaccountno;
}

void Unregistered_user::displaynewuserdetails()
```

```
{
   cout<<endl<< "Congratulations, Your account has been successfully created. :)"<<endl<<endl;

   cout<<  "Account Details!!"<<endl<<endl;
   cout << "User ID: "<<newUserID<<endl<<endl;
   cout << "User Password: "<<newUserPSW<<endl<<endl;
   cout << "User Name: "<<newName<<endl<<endl;
   cout << "User NIC: "<<newNIC<<endl<<endl;
   cout << "User Address: "<<newAddress<<endl<<endl;
   cout << "User Email: "<<newEmail<<endl<<endl;
   cout << "User Account No: "<<newAccountNo<<endl<<endl;
}

// Saving Account class method implementation
 void Saving_Account::setSavingDetails( int accountNo,string abranch,string ctype,double abalance)
{
   saccountNo = accountNo;
   accountBranch = abranch;
   currencyType = ctype;
   accountBalance = abalance;
}

void Saving_Account::displayaccountDetails()
{
   cout << "\n\nSummary of Saving  account\n" << endl << endl;
   cout << "Account No: "<<saccountNo<< endl;
   cout << "Branch: " <<accountBranch << endl;
   cout << "Currency Type: " <<currencyType << endl;
   cout << "Account Balance: "<<fixed << setprecision(2) << accountBalance << endl;
}

// Debit Card class method implementation
void Debit_Card::setDebitCardDetails(string dn,string de)
{
   dCardNo = dn;
   dCardExdate = de;
}

void Debit_Card::displayDcardDetails()
{
   cout<< "\nYour Debit Card Details Here!!\n " <<endl;
   cout<< "Card Number: " <<dCardNo<<endl;
   cout<< "Card Expire Date: "  <<dCardExdate<<endl;
}

// Loan class method implementation
void Loan::setLoanDetails(int ln,string loantype,double la)
{
   loanNo = ln;
   loanType = loantype;
   loanAmount = la;
}

void Loan::displayLoanDetails()
```

```
{
   cout << "\nYour Loan Details Here!! \n" << endl << endl;
   cout << "Your Loan Number: "<<loanNo << endl;
   cout << "Loan Type: "<<loanType<< endl;
   cout << "Loan Amount: "<< fixed << setprecision(2) << loanAmount << endl;
}

// Credit card class method implementation
void Credit_Card::setCcardDetails(string cn, string ce, double climit)
{
      cCardNo = cn;
      cCardExdate = ce;
      creditlimit = climit;
}

void Credit_Card::displayCcardDetails()
{
   cout << "\nYour Credit Card Details!!\n"  << endl << endl;
   cout << "Credit Card No: "<<cCardNo << endl;
   cout << "Credit Limit Per Month: "<< fixed << setprecision(2) << creditlimit << endl;
}

// Fixed account class method implementation
void Fixed_Account::setfixedDetails(int faccountno,string faccounntbranch,string currancytype,float
fixedaccountbalance)
{
   fixedaccountNo = faccountno;
   fixedaccountBranch = faccounntbranch;
   currencyType = currancytype;
   fixedaccountBalance = fixedaccountbalance;
}

void Fixed_Account::displayfixedDetails()
{
   cout << "\nSummary of Fixed Deposit" << endl << endl;
   cout << "Account No: "<<fixedaccountNo << endl;
   cout << "Branch: "<<fixedaccountBranch << endl;
   cout << "Currency Type: "<<currencyType << endl;
   cout << "Account Balance: "<<fixed << setprecision(2) << fixedaccountBalance << endl;
}

// Bill payment class method implementation
 void Bill_Payment::setpaymentDetails(string ptype, int ref, double payamount)
 {
   paymentType = ptype;
   referenceNo = ref;
   paymentAmount = payamount;
 }

 void Bill_Payment::displaypayment()
 {
   cout << "\nCongratulations!! Payment Successfull\n";
   cout << "\nPayment Type: "<<paymentType<<endl;
   cout << "\nReference No: "<<referenceNo<<endl;
   cout << "\nPayment Amount: "<<fixed << setprecision(2) << paymentAmount << endl;
 }


// Money transaction class method implementation
void Money_Transaction::setTransactionDetails(string bname, string brname, int myref, int raccountno, double
tramount)
 {
   bankName = bname;
   branchName = brname;
   myReference = myref;
   recieverAccountNo = raccountno;
   transferAmount = tramount;

 }

 void Money_Transaction::displayTransaction()
 {
   cout << "\nCongratulations!! Transaction Successfull\n";
   cout << "\nBank Name: "<<bankName<<endl;
   cout << "\nBranch Name: "<<branchName<<endl;
   cout << "\nMy Reference: "<<myReference<<endl;
   cout << "\nReciever Account No: "<<recieverAccountNo<<endl;
   cout << "\nTransfer Amount: "<<fixed << setprecision(2) << transferAmount << endl;
 }

// Complain class method implementation
 void Complain::setComplainDetails(string cname, string cemail, string des, int cn)
 {
   cName = cname;
   cEmail = cemail;
   description = des;
   contactNo = cn;
 }

 void Complain::displaycomplain()
 {
   cout << "\nYour Complain here!!\n"<<endl;
   cout << "Name: "<<cName<<endl;
   cout << "Email:  "<<cEmail<<endl;
   cout << "Contact No: "<<contactNo<<endl;
   cout << "Complain :\n"<<description<<endl;
 }

// Vacancy class method implementation
 void Vacancy::setVacancyDetails(string vname,string vdes, int vid)
{
   vacancyName = vname;
   vacancydescription = vdes;
   vacancyID = vid;
}

void Vacancy::displayvacancyDetails()
{
   cout << "\nVacancy ID: "<<vacancyID<<endl;
   cout<< "\nVacany Name: "<<vacancyName<<endl;
   cout << "\nVacancy Description:      "<<vacancydescription<<endl;
   cout << "\n\n                    Apply Now!!        \n\n" <<endl;
}
 // Main Program
int main ()
{
   int uID;
   // Multipurpose Variables
   string pw,td,br;
   char nu;
   int x,mn,ra,mr,a=0;
   double amount;

   while (a != -1){
   cout << "Enter UserID: ";
   cin>> uID;

   cout << "Enter User Password: ";
   cin >> pw;

   User u1,u2,u3,u4,u5; //Create 5 User Objects
   Vacancy v1,v2,v3; //Create 3 Vacancy Objects

   // User Accounts
   if (uID == 1001 && pw == "Janith@123")
   {
           u1.setUserDetails(1001, "Janith@123");
           u1.setDetails(1001, "P.G.E.J Sandamal",  "sandamal@gmail.com", "No/123/2, Galnawa Rd,
           Kirulapana. ", "200015285697",120018001);
           u1.displayUserDetails();
           cout<< "\nDo you want another service: (Y/N): ";
           cin>>nu;
           switch (nu)
           {
                   case 'Y':
                   case 'y':
                           cout<< "Select: \n1.Saving Account Details\n2.Fixed Account Details\n3.Transaction
                           & Bill Payment\n4.Do you Have any complain? \n\n";
                           cin>>x;

                           if(x==1)
                           {
                                   u1.setSavingDetails(120018001, "Kirulapana", "LKR", 56500.35);
                                   u1.setDebitCardDetails( "4565 4895 1265 4856" , "03 / 25");
                                   u1.setLoanDetails(2000525352, "Education Loan", 800000.00);
                                   u1.setCcardDetails( "4585 8956 7845 5656" , "04/23" ,25000.00);
                                   u1.displayaccountDetails();
                                   u1.displayDcardDetails();
                                   u1.displayLoanDetails();
                                   u1.displayCcardDetails();
                           }
                           else if (x == 2)
```

```
{
    u1.setfixedDetails(350060001, "Kirulapana" , "LKR" , 10000000.00);
    u1.displayfixedDetails();
}
else if ( x==3)
{
    cout << "\n\nselect Bill payment(1) Or Money Transaction (2) : ";
    cin >> x;

    switch (x)
    {
        case 1:
            {
                cin.ignore();
                cout<< "Select Service Provider Name: ";
                getline(cin,td);
                cout<< "Enter Mobile Number: ";
                cin >> mn;
                cout<< "Enter Amount: ";
                cin >> amount;

                u1.setpaymentDetails(td,mn,amount);
                u1.displaypayment();
                break;
            }
        case 2:
        {
            cin.ignore();
            cout<< "Enter Bank Name:  ";
            getline(cin,td);
            cin.ignore();
            cout<< "Enter Branch Name:  ";
            getline(cin,br);
            cout<< "Enter  my Reference No:  ";
            cin>>mr;
            cout<<  "Enter Account Number: ";
            cin >> ra;
            cout<< "Enter Amount: ";
            cin >> amount;

            u1.setTransactionDetails(td,br,mr,ra,amount);
            u1.displayTransaction();
            break;
        }
    }
}
else if( x == 4)
{
    cin.ignore();
    cout<< "Enter Your Name: ";
    getline(cin,br);
    cin.ignore();
    cout<< "Enter Your Email: ";
    getline(cin,pw);
```

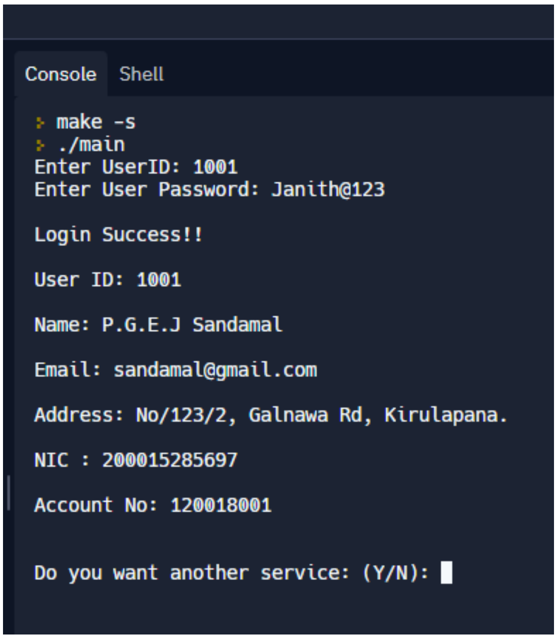
```
                cout<< "Enter your contact No: ";
                cin>>mn;
                cin.ignore();
                cout<< "\nEnter your complain here: \n"<<endl;
                getline(cin,td);

                 u1.setComplainDetails(br,pw,td,mn);
                 u1.displaycomplain();
            }
    }
 a=-1;
}
else if (uID == 1002 && pw == "Pahasara@123")
{
    u2.setUserDetails(1002, "Pahasara@123");
    u2.setDetails(1002, "R.P.N Arachchi",  "pahasra@gmail.com",  "Arachchi Niwasa, Allewela Rd, Hakmana. " , "994035512V", 120018002);
    u2.displayUserDetails();
    cout<< "\nDo you want another service: (Y/N): ";
    cin>>nu;
    switch (nu)
    {
        case 'Y':
        case 'y':
            cout<< "Select: \n1.Saving Account Details\n2.Fixed Account Details\n3.Transaction & Bill Payment\n4.Do you Have any complain? \n\n";
            cin>>x;

            if(x==1)
            {
                u2.setSavingDetails(120018002, "Hakmana",  "LKR", 6500.35);
                u2.setDebitCardDetails( "4885 4845 1565 4886", "03 / 23");
                u2.setLoanDetails(2000525353, "Home Loan", 1500000.00);

                u2.displayaccountDetails();
                u2.displayDcardDetails();
                u2.displayLoanDetails();
                cout<< "\n\nNo active Credit Cards Details in Here!!   Apply Now!!\n " <<endl;
            }
            else if (x == 2)
            {
                u2.setfixedDetails(350060002, "Matara", "LKR", 1000000.00);
                u2.displayfixedDetails();
            }
            else if ( x==3)
            {
                cout << "\n\nselect Bill payment(1) Or Money Transaction (2) : ";
                cin >> x;

                switch (x)
                {
                    case 1:
                        {
                            cin.ignore();
                            cout<< "Select Service Provider Name: ";
                            getline(cin,td);
                            cout<< "Enter Mobile Number: ";
                            cin >> mn;
                            cout<< "Enter Amount: ";
                            cin >> amount;

                            u2.setpaymentDetails(td,mn,amount);
                            u2.displaypayment();
                            break;
                        }
                    case 2:
                    {
                        cin.ignore();
                        cout<< "Enter Bank Name:  ";
                        getline(cin,td);
                        cin.ignore();
                        cout<< "Enter Branch Name:  ";
                        getline(cin,br);
                        cout<< "Enter  my Reference No:  ";
                        cin>>mr;
                        cout<< "Enter Account Number: ";
                        cin >> ra;
                        cout<< "Enter Amount: ";
                        cin >> amount;

                        u2.setTransactionDetails(td,br,mr,ra,amount);
                        u2.displayTransaction();
                        break;
                    }
                }
            }
            else if( x == 4)
            {
                cin.ignore();
                cout<<  "Enter Your Name: ";
                getline(cin,br);
                cin.ignore();
                cout<< "Enter Your Email: ";
                getline(cin,pw);
                cout<< "Enter your contact No: ";
                cin>>mn;
                cin.ignore();
                cout<< "\nEnter your complain here: \n"<<endl;
                getline(cin,td);

                 u2.setComplainDetails(br,pw,td,mn);
                 u2.displaycomplain();
            }
    }
 a=-1;
}
else if (uID == 1003 && pw == "Dinuka@123")
```

```
{
    u3.setUserDetails(1003, "Dinuka@123");
    u3.setDetails(1003, "G.A.D Nawod", "dinuka@gmail.com", "No/129 New Kandy Rd, Panadura.",
    "200054268795",120018003);
    u3.displayUserDetails();
    cout<< "\nDo you want another service: (Y/N): ";
    cin>>nu;
    switch (nu)
    {
        case 'Y':
        case 'y':
            cout<<  "Select: \n1.Saving Account Details\n2.Fixed Account Details\n3.Transaction
            & Bill Payment\n4.Do you Have any complain? \n\n";
            cin>>x;

            if(x==1)
            {
                u3.setSavingDetails(120018003, "Panadura",  "LKR", 156500.45);
                u3.setDebitCardDetails( "4525 4855 7265 9856", "02 / 25");
                u3.setLoanDetails(2000525352, "Personal Loan", 80000.00);
                u3.setCcardDetails( "4515 8556 7885 5656", "04/25",50000.00);
                u3.displayaccountDetails();
                u3.displayDcardDetails();
                u3.displayLoanDetails();
                u3.displayCcardDetails();

            }
            else if (x == 2)
            {
                cout<< "\n\nNo Fixed Deposit Account!\n\n"<<endl;
            }
            else if ( x==3)
            {
                cout << "\n\nselect Bill payment(1) Or Money Transaction (2) : ";
                cin >> x;

                switch (x)
                {
                    case 1:
                        {
                            cin.ignore();
                            cout<< "Select Service Provider Name: ";
                            getline(cin,td);
                            cout<< "Enter Mobile Number: ";
                            cin >> mn;
                            cout<< "Enter Amount: ";
                            cin >> amount;

                            u3.setpaymentDetails(td,mn,amount);
                            u3.displaypayment();
                            break;
                        }
                    case 2:
                    {
                        cin.ignore();
                        cout<< "Enter Bank Name:  ";
                        getline(cin,td);
                        cin.ignore();
                        cout<< "Enter Branch Name:  ";
                        getline(cin,br);
                        cout<< "Enter  my Reference No:  ";
                        cin>>mr;
                        cout<< "Enter Account Number: ";
                        cin >> ra;
                        cout<< "Enter Amount: ";
                        cin >> amount;

                        u3.setTransactionDetails(td,br,mr,ra,amount);
                        u3.displayTransaction();
                        break;
                    }
                }
            }

            else if( x == 4)
            {
                cin.ignore();
                cout<< "Enter Your Name: ";
                getline(cin,br);
                cin.ignore();
                cout<< "Enter Your Email: ";
                getline(cin,pw);
                cout<< "Enter your contact No: ";
                cin>>mn;
                cin.ignore();
                cout<< "\nEnter your complain here: \n"<<endl;
                getline(cin,td);

                 u3.setComplainDetails(br,pw,td,mn);
                 u3.displaycomplain();
            }
    }
 a=-1;
}
else if (uID == 1004 && pw == "Naveen@123")
{
    u4.setUserDetails(1004, "Naveen@123");
    u4.setDetails(1004, "W.N Dilsara",  "naveen@gmail.com", "No/225 Gehan Rd, Thalangama North,
    Koswatta", "993520411V",120018004);
    u4.displayUserDetails();
    cout<< "\nDo you want another service: (Y/N): ";
    cin>>nu;
    switch (nu)
    {
        case 'Y':
        case 'y':
            cout<< "Select: \n1.Saving Account Details\n2.Fixed Account Details\n3.Transaction
            & Bill Payment\n4.Do you Have any complain? \n\n";
```

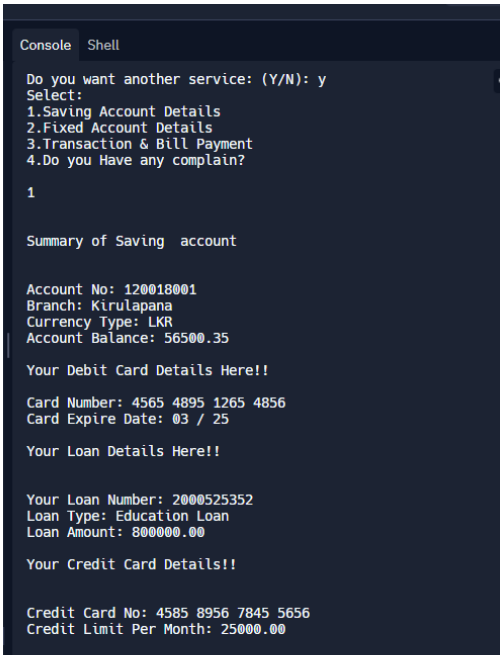
```
cin>>x;

if(x==1)
{
        u4.setSavingDetails(120018004, "Battaramulla", "LKR", 31507.85);
        u4.setDebitCardDetails( "4865 5895 1225 4896", "03 / 26");
        u4.setLoanDetails(2000525352, "Education Loan", 800000.00);
        u4.setCcardDetails( "4995 8985 7847 5546", "11/25",15000.00);

        u4.displayaccountDetails();
        u4.displayDcardDetails();
        u4.displayLoanDetails();
        u4.displayCcardDetails();
}
else if (x == 2)
{
        u4.setfixedDetails(350060004, "Malabe",  "LKR", 5000000.00);
        u4.displayfixedDetails();
}

else if ( x==3)
{
        cout << "\n\nselect Bill payment(1) Or Money Transaction (2) : ";
        cin >> x;

        switch (x)
        {
                case 1:
                        {
                                cin.ignore();
                                cout<< "Select Service Provider Name: ";
                                getline(cin,td);
                                cout<< "Enter Mobile Number: ";
                                cin >> mn;
                                cout<< "Enter Amount: ";
                                cin >> amount;

                                u4.setpaymentDetails(td,mn,amount);
                                u4.displaypayment();
                                break;
                        }

                case 2:
                {
                        cin.ignore();
                        cout<< "Enter Bank Name:  ";
                        getline(cin,td);
                        cin.ignore();
                        cout<< "Enter Branch Name:  ";
                        getline(cin,br);
                        cout<< "Enter  my Reference No:  ";
                        cin>>mr;
                        cout<< "Enter Account Number: ";
                        cin >> ra;
                        cout<< "Enter Amount: ";
                        cin >> amount;

                        u4.setTransactionDetails(td,br,mr,ra,amount);
                        u4.displayTransaction();
                        break;
                    }
                }
            }
            else if( x == 4)
            {
                cin.ignore();
                cout<< "Enter Your Name: ";
                getline(cin,br);
                cin.ignore();
                cout<< "Enter Your Email: ";
                getline(cin,pw);
                cout<< "Enter your contact No: ";
                cin>>mn;
                cin.ignore();
                cout<< "\nEnter your complain here: \n"<<endl;
                getline(cin,td);

                 u4.setComplainDetails(br,pw,td,mn);
                 u4.displaycomplain();
            }
    }
 a=-1;
}
else if (uID == 1005 && pw ==  "Kithmini@123")
{
    u5.setUserDetails(1005, "Kithmini@123" );
    u5.setDetails(1005 , "K.S.K Wijesinghe",  "kithmini@gmail.com", "Kithmini niwasa,  bandaththara Rd, Matara.", "9938960411V",120018005);
    u5.displayUserDetails();
    cout<< "\nDo you want another service: (Y/N): ";
    cin>>nu;
    switch (nu)
    {
        case 'Y':
        case 'y':
            cout<<  "Select: \n1.Saving Account Details\n2.Fixed Account Details\n3.Transaction & Bill Payment\n4.Do you Have any complain? \n\n" ;
            cin>>x;

            if(x==1)
            {
                u5.setSavingDetails(120018005, "Matara",  "LKR", 5100.35);
                u5.setDebitCardDetails( "4885 4895 5265 4876", "03 / 25");
                u5.setLoanDetails(2000525352, "Education Loan", 800000.00);
                u5.setCcardDetails( "4585 8956 7856 5856", "04/23",35000.00);

                u5.displayaccountDetails();
                u5.displayDcardDetails();
                u5.displayLoanDetails();
                u5.displayCcardDetails();
        }
        else if (x == 2)
        {
                cout<< "\n\nNo Fixed Deposit Account!\n\n"<<endl;
        }
        else if ( x==3)
        {
                cout << "\n\nselect Bill payment(1) Or Money Transaction (2) : ";
                cin >> x;

                switch (x)
                {
                        case 1:
                                {
                                        cin.ignore();
                                        cout<< "Select Service Provider Name: ";
                                        getline(cin,td);
                                        cout<< "Enter Mobile Number: ";
                                        cin >> mn;
                                        cout<< "Enter Amount: ";
                                        cin >> amount;

                                        u5.setpaymentDetails(td,mn,amount);
                                        u5.displaypayment();
                                        break;
                                }
                        case 2:
                        {
                                cin.ignore();
                                cout<< "Enter Bank Name:  ";
                                getline(cin,td);
                                cin.ignore();
                                cout<< "Enter Branch Name:  ";
                                getline(cin,br);
                                cout<< "Enter  my Reference No:  ";
                                cin>>mr;
                                cout<< "Enter Account Number: ";
                                cin >> ra;
                                cout<< "Enter Amount: ";
                                cin >> amount;

                                u5.setTransactionDetails(td,br,mr,ra,amount);
                                u5.displayTransaction();
                                break;
                        }
                }
        }
        else if( x == 4)
        {
                cin.ignore();
                cout<< "Enter Your Name: ";
                getline(cin,br);
                cin.ignore();
                cout<< "Enter Your Email: ";
                getline(cin,pw);
                cout<< "Enter your contact No: ";
                cin>>mn;
                cin.ignore();
                cout<< "\nEnter your complain here: \n"<<endl;
                getline(cin,td);

                 u5.setComplainDetails(br,pw,td,mn);
                 u5.displaycomplain();
            }
    }
a=-1;
}
//Create new user account
else
{
    cout << endl<< "You entered an incorrect user ID or password." << endl<<endl;
    cout << "New to Online Banking? Create an Account (Y/N): ";
    cin >> nu;
    switch (nu)
    {
        case 'Y':
        case 'y':
            {
                long account;
                string name;
                string email;
                string password;
                string address;
                string nic;

                cout<<endl<<endl;
                cout<< "Enter Your Account No : ";
                cin>> account;
                cin.ignore();
                cout<< "Enter Your Name: ";
                getline(cin, name);
                cout<<  "Enter Your NIC No: ";
                cin>> nic;
                cin.ignore();
                cout<< "Enter Your Address: ";
                getline(cin, address);
                cout<< "Enter Your Email: ";
                cin>> email;
                cin.ignore();
                cout<< "Enter Your Password: ";
                getline(cin, password);

                Unregistered_user newUser;

                newUser.setnewuserdetails(1006,name,email,address,nic,password,
                account);
```

```
                    newUser.displaynewuserdetails();
                    a=-1;
                    break;
                }
            case 'N':
            case 'n':
                {
                    cout<< "\n\n"<<endl;
                     break;
                }
            default:
                cout<<endl<< "Invalid Input";
                break;
        }
  }
}

//Vacancies
  cout<< "\n\n Vacancies Available in Here !!\n\tsee more(enter 1): ";
  cin>>x;
  if(x==1)
  {
        Vacancy v1,v2,v3;
        v1.setVacancyDetails( "Bank Trainee", "\n\n5 > C Passes or Credit Passes in O/L exam.including C pass (Minimum) in English Language \nAge limi: 18-24 years\n", 10001);
        v2.setVacancyDetails( "Office Guard", "\n\n5 > C Passes or Credit Passes in O/L exam. \nAge limi: 45-65 years\n", 10002);
        v3.setVacancyDetails( "Area Manager", "\n\nCompleted CIMA (Up to Operational-Level)\nCompleted CIM (UK)  (up to Diploma-Level)\nCompleted IABF  (at IBSL -Institute of #Bankers of Sri-Lanka)\nAAT-Passed Finalist.\nAge limi: below 24\n", 10003);

        v1.displayvacancyDetails();
        v2.displayvacancyDetails();
        v3.displayvacancyDetails();
  }
        return 0;
}
```

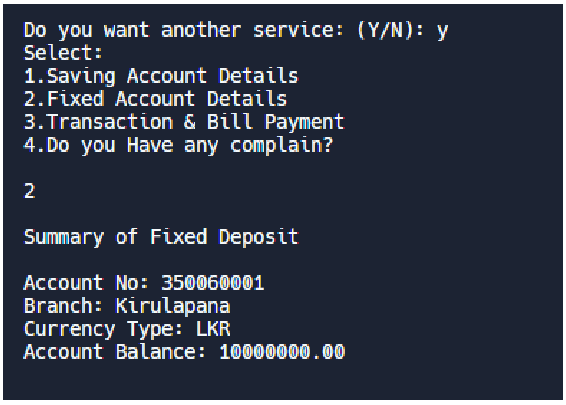
Sample Outputs



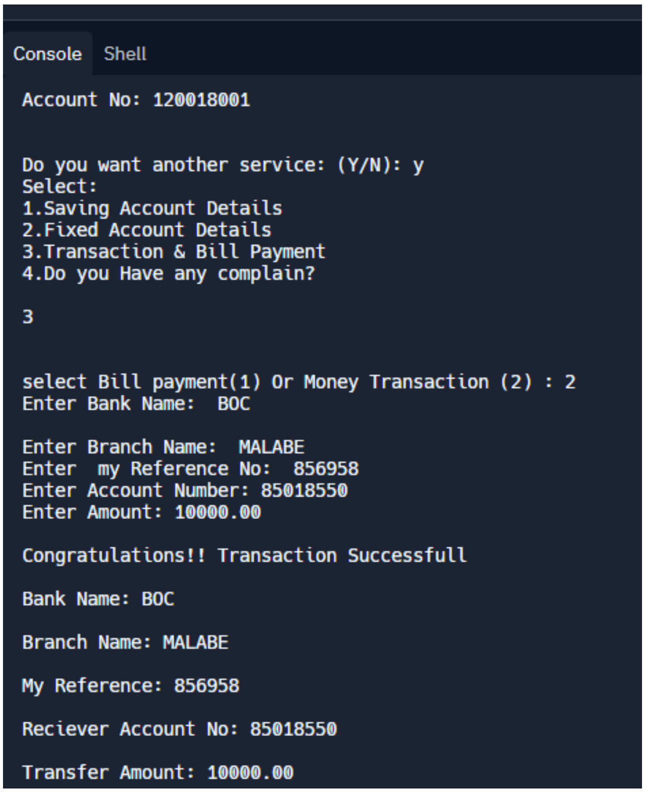
```
› make -s
› ./main
Enter UserID: 1001
Enter User Password: Janith@123

Login Success!!

User ID: 1001

Name: P.G.E.J Sandamal

Email: sandamal@gmail.com

Address: No/123/2, Galnawa Rd, Kirulapana.

NIC : 200015285697

Account No: 120018001


Do you want another service: (Y/N): █
```



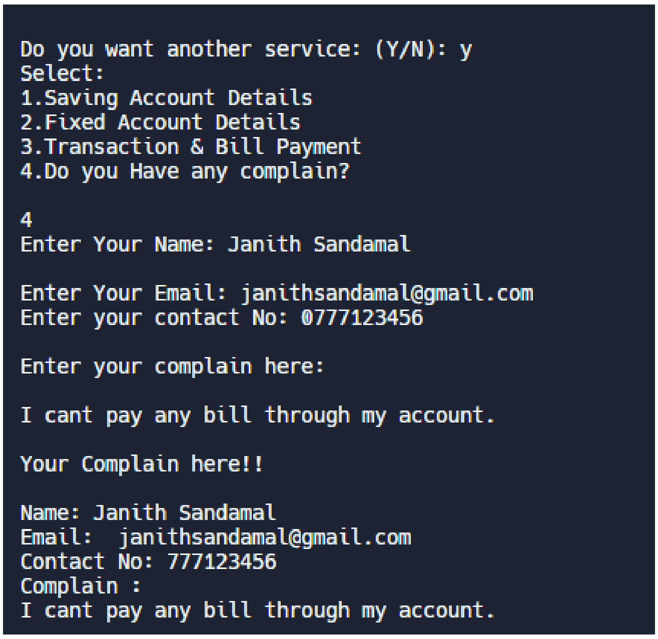
```
Do you want another service: (Y/N): y
Select:
1.Saving Account Details
2.Fixed Account Details
3.Transaction & Bill Payment
4.Do you Have any complain?

1


Summary of Saving  account


Account No: 120018001
Branch: Kirulapana
Currency Type: LKR
Account Balance: 56500.35

Your Debit Card Details Here!!

Card Number: 4565 4895 1265 4856
Card Expire Date: 03 / 25

Your Loan Details Here!!


Your Loan Number: 2000525352
Loan Type: Education Loan
Loan Amount: 800000.00

Your Credit Card Details!!


Credit Card No: 4585 8956 7845 5656
Credit Limit Per Month: 25000.00
```



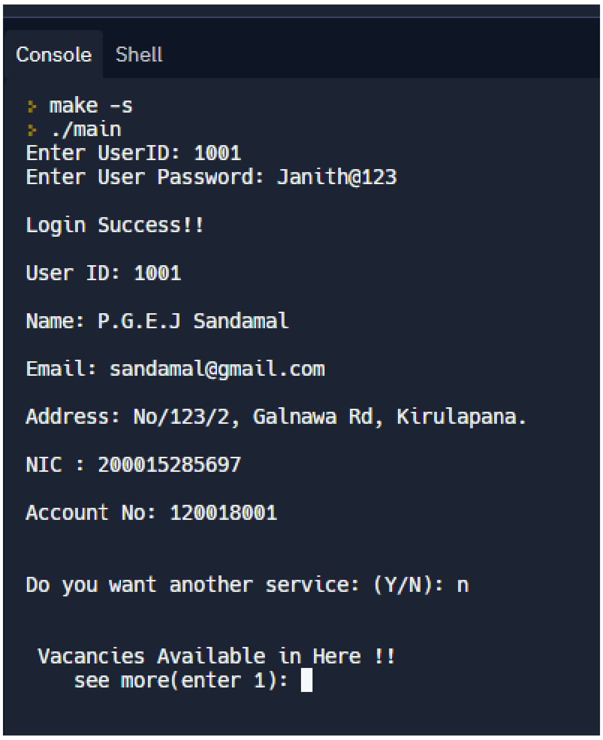
```
Do you want another service: (Y/N): y
Select:
1.Saving Account Details
2.Fixed Account Details
3.Transaction & Bill Payment
4.Do you Have any complain?

2

Summary of Fixed Deposit

Account No: 350060001
Branch: Kirulapana
Currency Type: LKR
Account Balance: 10000000.00
```



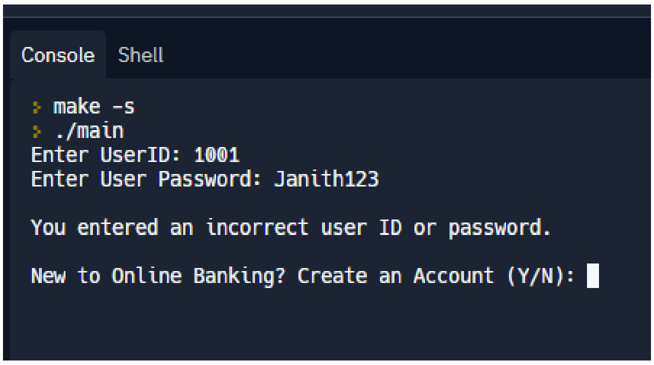
```
Account No: 120018001


Do you want another service: (Y/N): y
Select:
1.Saving Account Details
2.Fixed Account Details
3.Transaction & Bill Payment
4.Do you Have any complain?

3


select Bill payment(1) Or Money Transaction (2) : 2
Enter Bank Name:  BOC

Enter Branch Name:  MALABE
Enter  my Reference No:  856958
Enter Account Number: 85018550
Enter Amount: 10000.00

Congratulations!! Transaction Successfull

Bank Name: BOC

Branch Name: MALABE

My Reference: 856958

Reciever Account No: 85018550

Transfer Amount: 10000.00
```



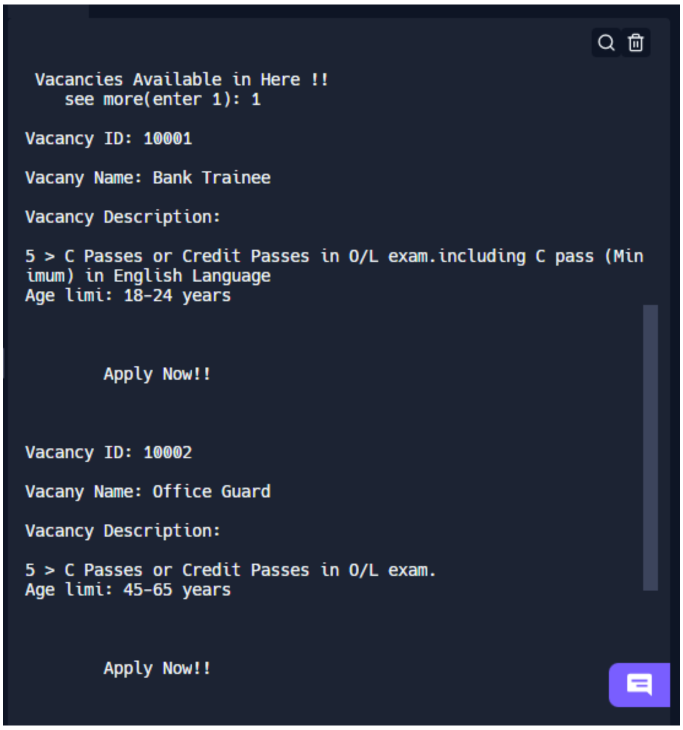
```
Do you want another service: (Y/N): y
Select:
1.Saving Account Details
2.Fixed Account Details
3.Transaction & Bill Payment
4.Do you Have any complain?

4
Enter Your Name: Janith Sandamal

Enter Your Email: janithsandamal@gmail.com
Enter your contact No: 0777123456

Enter your complain here:

I cant pay any bill through my account.

Your Complain here!!

Name: Janith Sandamal
Email:  janithsandamal@gmail.com
Contact No: 777123456
Complain :
I cant pay any bill through my account.
```



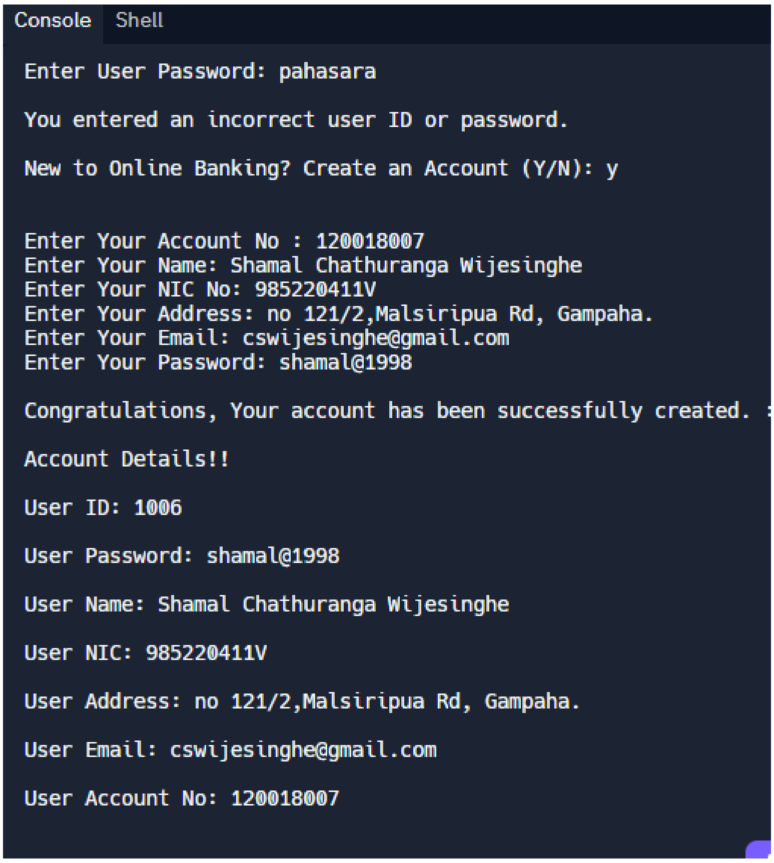
```
> make -s
> ./main
Enter UserID: 1001
Enter User Password: Janith@123

Login Success!!

User ID: 1001

Name: P.G.E.J Sandamal

Email: sandamal@gmail.com

Address: No/123/2, Galnawa Rd, Kirulapana.

NIC : 200015285697

Account No: 120018001


Do you want another service: (Y/N): n


 Vacancies Available in Here !!
    see more(enter 1):
```



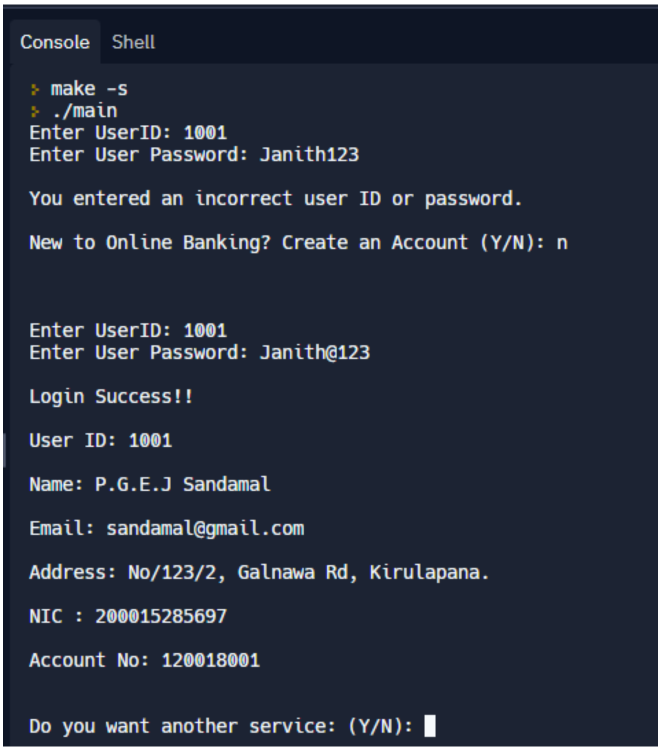
```
> make -s
> ./main
Enter UserID: 1001
Enter User Password: Janith123

You entered an incorrect user ID or password.

New to Online Banking? Create an Account (Y/N):
```



```
 Vacancies Available in Here !!
    see more(enter 1): 1

Vacancy ID: 10001

Vacany Name: Bank Trainee

Vacancy Description:

5 > C Passes or Credit Passes in O/L exam.including C pass (Min
imum) in English Language
Age limi: 18-24 years


        Apply Now!!


Vacancy ID: 10002

Vacany Name: Office Guard

Vacancy Description:

5 > C Passes or Credit Passes in O/L exam.
Age limi: 45-65 years


        Apply Now!!
```



```
Enter User Password: pahasara

You entered an incorrect user ID or password.

New to Online Banking? Create an Account (Y/N): y


Enter Your Account No : 120018007
Enter Your Name: Shamal Chathuranga Wijesinghe
Enter Your NIC No: 985220411V
Enter Your Address: no 121/2,Malsiripua Rd, Gampaha.
Enter Your Email: cswijesinghe@gmail.com
Enter Your Password: shamal@1998

Congratulations, Your account has been successfully created. :

Account Details!!

User ID: 1006

User Password: shamal@1998

User Name: Shamal Chathuranga Wijesinghe

User NIC: 985220411V

User Address: no 121/2,Malsiripua Rd, Gampaha.

User Email: cswijesinghe@gmail.com

User Account No: 120018007
```



```
> make -s
> ./main
Enter UserID: 1001
Enter User Password: Janith123

You entered an incorrect user ID or password.

New to Online Banking? Create an Account (Y/N): n


Enter UserID: 1001
Enter User Password: Janith@123

Login Success!!

User ID: 1001

Name: P.G.E.J Sandamal

Email: sandamal@gmail.com

Address: No/123/2, Galnawa Rd, Kirulapana.

NIC : 200015285697

Account No: 120018001


Do you want another service: (Y/N):
```

Individual Contributions

Sandamal P.G.E.J - IT21166860

- Identifying the relationships between User, unregistered user and registered User and other classes
- Created the CRC Card for User, Registered user and Unregistered user classes.
- Created the class diagram User, Registered user and Unregistered user.
- implemented the coding for the User, Registered user and Unregistered user class.

Dilsara W.N - IT21182600

- Created the class diagram and CRC for Saving and Fixed Account
- Create the classes and code the Saving and Fixed Account
- Combined header files and main.cpp files and documented the entire code.
- Completed the rest of document

Arachchi R.P.N- IT21182464

- Created and documented the user requirements.
- Created the class diagram for Money transaction and Bill payment.
- Created the CRC card for Money transaction and Bill payment
- Created the classes and coded the Money transaction and Bill payment.

Nawod G.A.D - IT21166624

- Created and documented the introduction.
- CRC cards for Credit Card, Debit Card and Loan.
- Drawing of class diagram for Credit Crad, Debit Card and Loan with relationships to other classes
- Coding of Credit Card, Debit Card and Loan classes

Wijesinghe K.S.K - IT21181610

- Designed the class diagram for the Complain and Vacancy.
- Designed the CRC Cards for the Inquiry Complain and Vacancy classes.
- I have implemented the codes for the Complain and Vacancy.
- Identified and documented nouns in user requirements description.