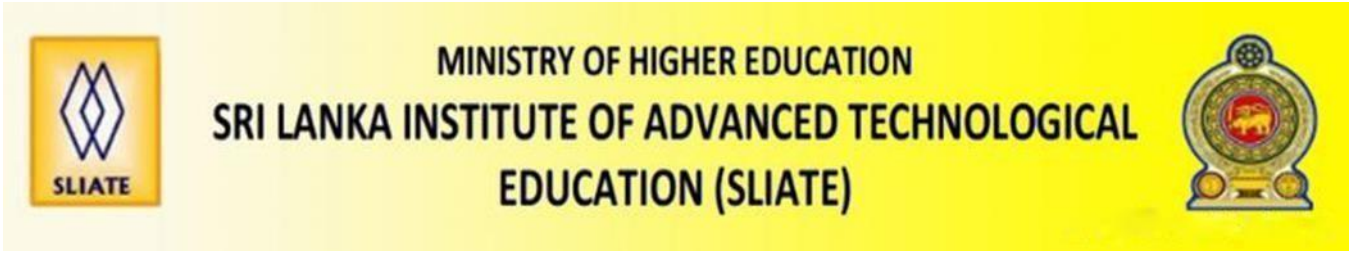


Sri Lanka Institute of Advanced Technological Education.
Labuduwa.

Higher National Diploma in Information Technology

2018

Project Document

Banking Management System

Project Details

Project Supervisor:

Lecturer:

Mr.C. Samarasekara

Project By:

M.E. Rathnayaka
GAL/IT/2018/F/0146

Abstract

After completing our internal Higher National Diploma in Information Technology belong in Sri Lanka Institute of Advanced Technological Education in Labuduwa, we were established the Banking Management System which users can be log easily.

This Banking Management System is specially designed national bank.
In this banking management system online process is available. Generally, there are four main topics in development of this system. The first main topic is **Introduction**. Under this topic purpose of the project, scope of the project, Minimum Hardware, Software Requirements. The second topic is **General Description**. It consists User Characteristics, General Constraints, Review of Related literature, Assumptions and Dependencies of the project. The Third topic is **System analysis**. Under this topic Functional Requirements, Nonfunctional Requirements, Product Features, Project Feasibility (technical, operational, legal, economic), ER Diagram, User Case Diagram, Class Diagram, Activity Diagram. The fourth topic is **Design Introduction**. Under this topic Logging Function, update function, Data Structures, are discussed. The fifth topic is **Software testing**. The sixth topic is **System Implementation**. Under this topic System Implementation Overall Description are discussed. The sixth topic is **Evaluation**. The last topic is **Conclusions**. Under this topic Conclusions, References are discussed.

Acknowledgement

In performing this project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We sincerely appreciate the inspiration; support and guidance of all those people who have been conduct in making this project a success.

At this juncture we feel deeply honored in expressing our sincere thanks to senior lecturer of the Information Technology department lecturer Sir C. Samarasekara of supervisor and all the other lecturers of Information Technology department for making this project available at right time and providing valuable insights leading to the successful completion.

We would also like to thank all the HNDIT students of 19th batch for their critical advice and guidance without which this project would not have been possible.

Last but not the least we place a deep sense of gratitude to our family members and our friends who have been constant source of inspiration during the preparation of this project work.

Finally, we would like to wish this valuable place will be eminent, efficacious and knowledgeable students can be made for my mother land.

Content

CHAPTER 01

Introduction

The project entitled “Bank management System” is a computerized telecommunication device that provides the customers of financial institution with the customers of a financial transaction is a public space without the need for a human clerk or bank taller.

Thousands of bank performs millions of transaction every day and thousands of users used banking system in day to day life. As we know that if number of users increases us need more banks and more staff it means increasing manual work also we put more amount of money in bank it is more risky and not much secure. If we developed advanced computerized based banking system so there is no need to open more branches as well the manpower is reduce and maximum information and stored automatically in banking server.

In my project I provide the security question when customer login with account to prevent the fraud and provide the best security in the bank management system.

Goals and objectives

1. Create an easy to understand user friendly environment.
2. Save cost.
3. Save time.
4. Efficiency

Scope of the product

- The proposed system is highly computerized in which the data related to user accounts will be secured high with high accuracy that even reduced the machine damage and human made errors and this existing system is highly efficient to offer best service to the customers as well as bank because it has user friendly access that customers less time when compare with a normal banking system.

- When the data is entered it will check for its validity. Appropriate message are provided as when needed so that the user will not be in a maze of instant.

- The data entry screen is design such a way that all the data manipulates can be performed, it also provide record viewing facilities.

- In the below fig this project is use for online banking system, the user can register first and then login. When user login successfully they will perform the operation like money withdraw, money transfer, deposit.

- Admin has all authority to handle all the user account and transactions in a sequence to avoid unauthorized user.

- Customer can update his data like address, contact number….etc. User can transfer money, deposit money, withdraw and check account balance through banking system.

HARDWARE REQUIREMENTS

- Pentium (R)Dual-core CPU 2.53GHz
- Operating system-windows platform
- 4.00 GB RAM 250 GB Hard Disk.
- VGA 512MB or higher.
- Mouse.
- Keyboard.

SOFTWARE REQUIREMENTS

- SQLite: It will be used as the Database Management Software to hold the data of the payroll system. It is required in the operating computer.
- Microsoft Word 2013: It will be used to prepare the Documentation and the User Manual for the system.
- Star UML: It will be used to prepare the diagrammatic explanations like Data Flow Diagrams, System Flowcharts.
- An operating system strong enough to handle the powerful applications. E.g. Windows 10

CHAPTER 02

General Description

Customer Perspective

This software is totally self-contained and works relatively as efficient as other packages related to the subject. It provides simple database rather than complex ones for high requirements and it provides good and easy graphical user interface to both new, as well as experienced users of the computers.

User Characteristics

The system will be used in the Trade Center. The administrator will be the main user. Given the condition that not all the users are computer – literate. Some users may have to be trained on using the system. The system is also designed to be user- friendly. It uses a Graphical User Interface (GUI)

- No pre knowledge of JAVA
- No pre knowledge of database management.
- Should no English.
- Should be able to use and Do according to the graphical user interface

General Constraints

This project works well in PC's having SQLite database installed properly.

Assumptions and Dependencies

It is assumed that the Banking will have enough trained staff to take care of the system. It is assumed that several compatible computers will be available before the system is installed and tested.

Review of Related literature

Literature Review is the reviewing from the before research that can be made by other people. The reviewing is including the process that relate to the Banking Management System. The source of research can be referring to the journal, paper work, websites and reference books.

CHAPTER 03

System analysis

This chapter mainly focuses on the study on the business and understand the domain knowledge. After getting the domain knowledge we can understand the system expected user's functional and nonfunctional requirements of the proposed system.

PRODUCT FEATURES

There are many different users who will be using this system: User who will be acting as the administrator

- An admin can view Customer List, transaction list
- Customer can withdraw, transaction, and deposit
- Admin and customer can register
- Can access all other user's features

Security

Customer can access this software only after entering the appropriate user name and password.

Different users have different views and User privileges.

System Attributes

Maintainability - The database is self-maintained.

Flexibility - It Is Easy to Update and Modify the Data When Needed.

Validation - An Error Message Is Displayed If Entry Is Wrong.

Project Feasibility

Technical Feasibility

For our feasibility analysis we will have to determine how the system is regularly works and how patients and administrators understand it.

Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. So in that case we can gave answers for that, the new process is happening totally automatically.

Human Factors

In order for this project to be successful we will have to determine what kind of impact it will make on our users.

Legal Feasibility

Not only for the Admin but also all users can log in to the system entering valid username and password.

Economic Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the medical center that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.in that case we can see there's some things to look, both cost/ benefit analysis, actually in that case the medical center can have lot of cost benefits through the system.

Diagrams

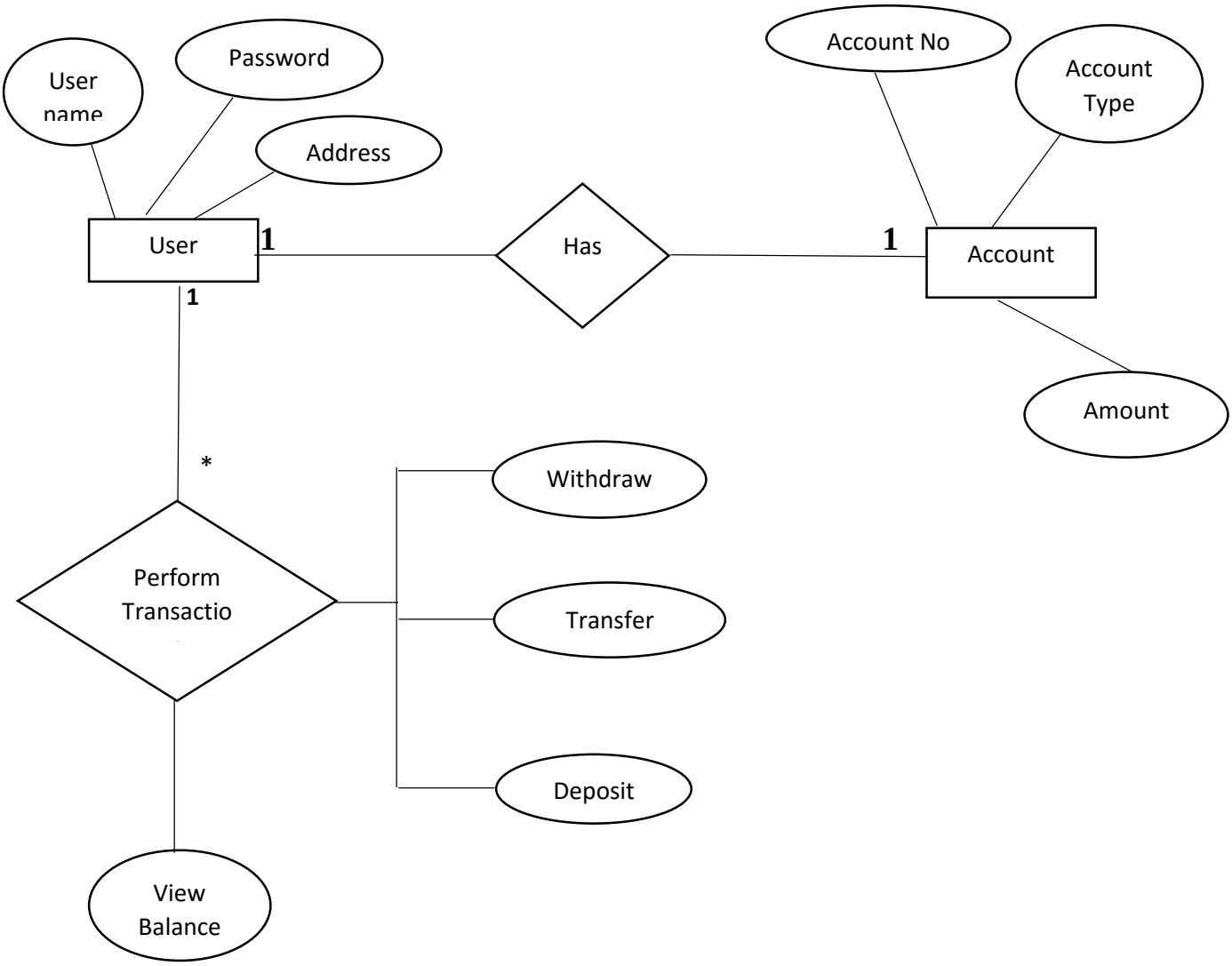
ER Diagram

Use Case Diagram

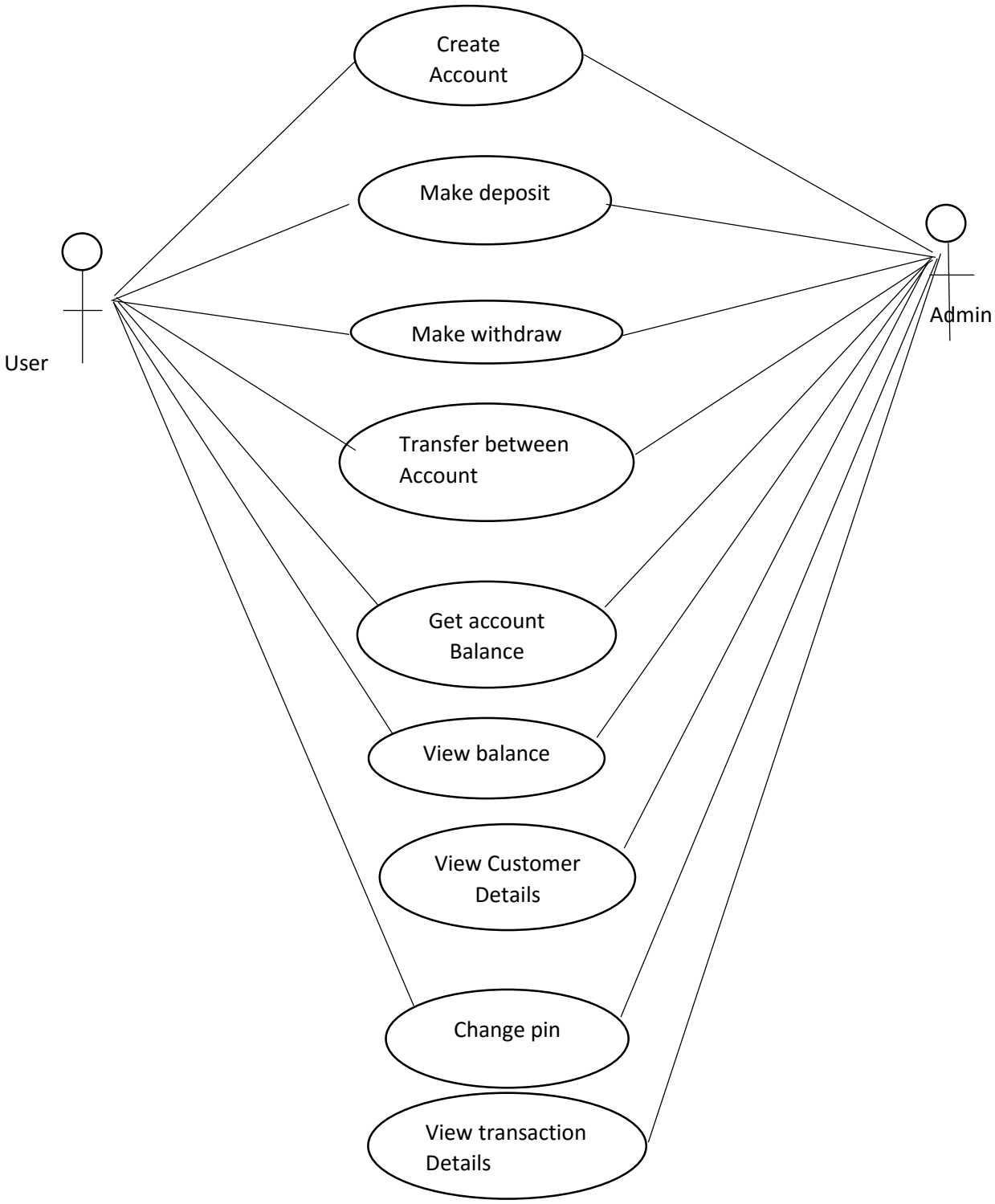
Class Diagram

Activity Diagram

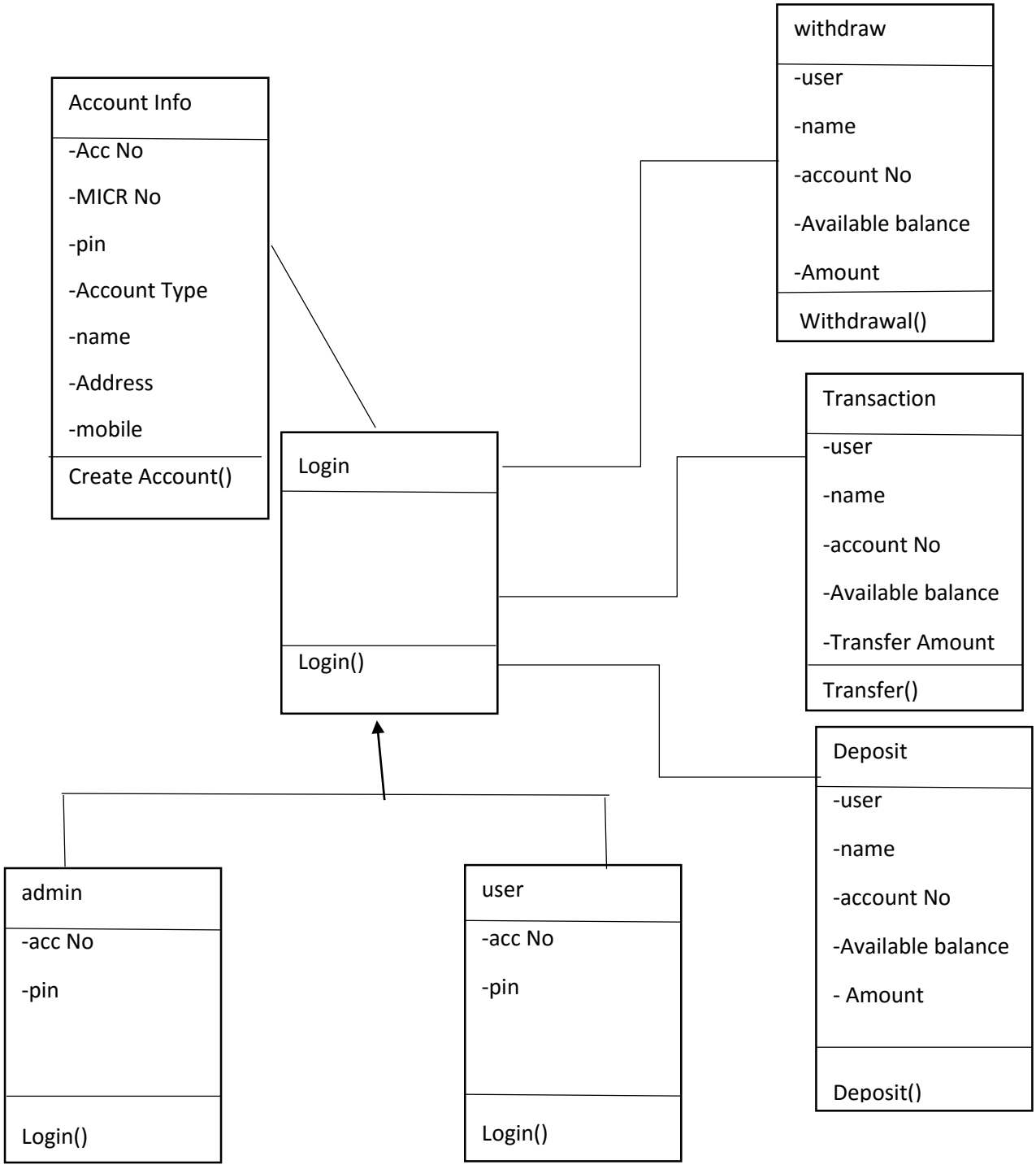
ER Diagram



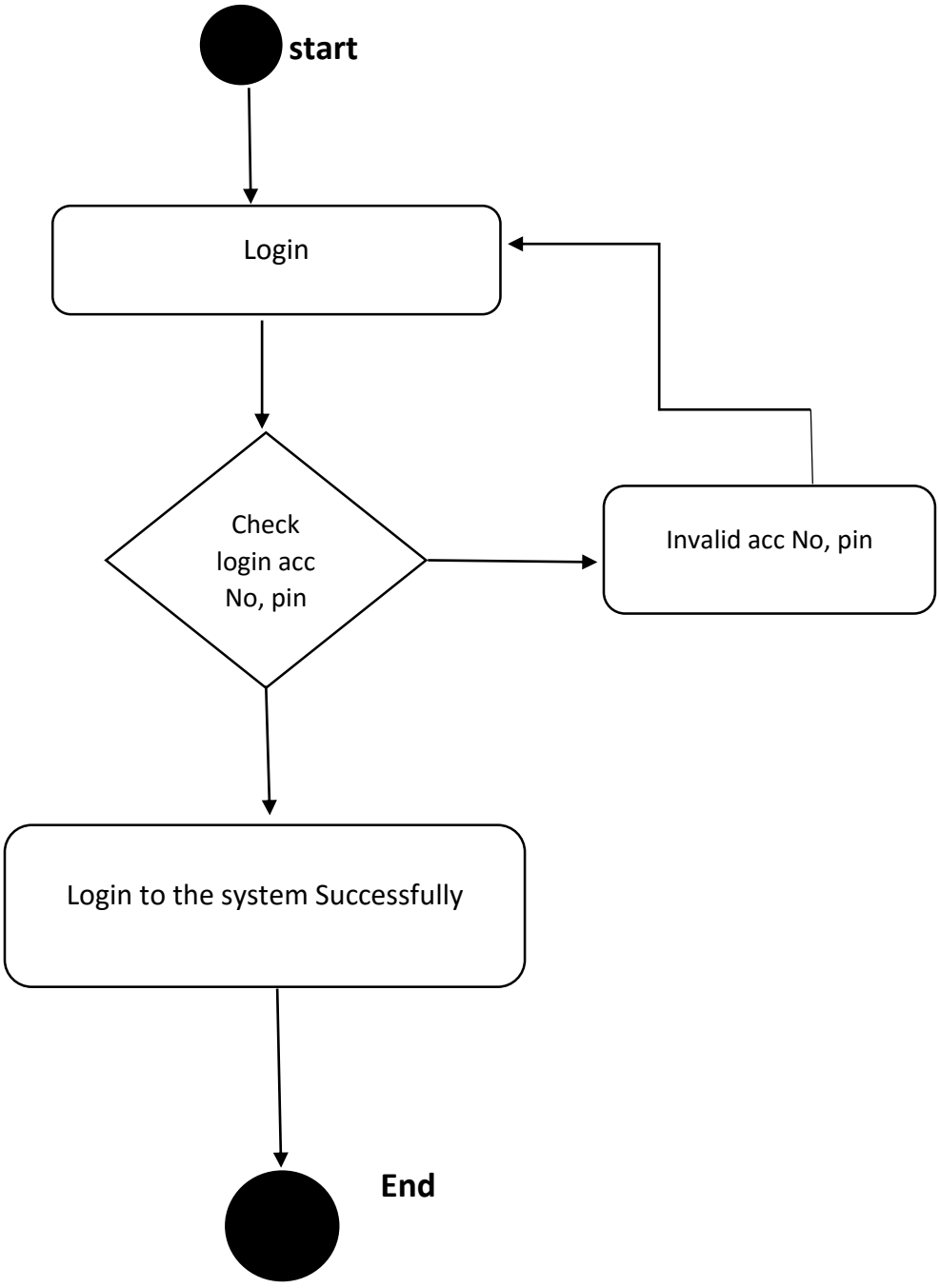
Use Case Diagram



Class Diagram



Activity Diagram



Functional Requirements

1. Admin, user login to the System
2. User should allow to fill the registration form.
3. User can transfer money, withdraw and deposit
4. User can change pin number.
5. Updating the user details.

Nonfunctional Requirements

Performance Requirements

Performance requirements are:

- Reliability.
- Performance Requirements.
- Availability.
- Maintainability.
- Security.

Security

Admin is the primary who will have permissions to access the website. Provides data security than manual system because only admin can access all the details of system and other user like staff of this banking system, they have limitation of accessing the system, so data will secure.

Accuracy

Storage functions are also should be efficient and the system should provide correct information.

Reliability

The system should run without any failure and response time taken should be less.

CHAPTER 04

Design Introduction

Design chapter will discuss how to model the system development process using a process model and how to translate problem domain into a solution domain using a design technique.

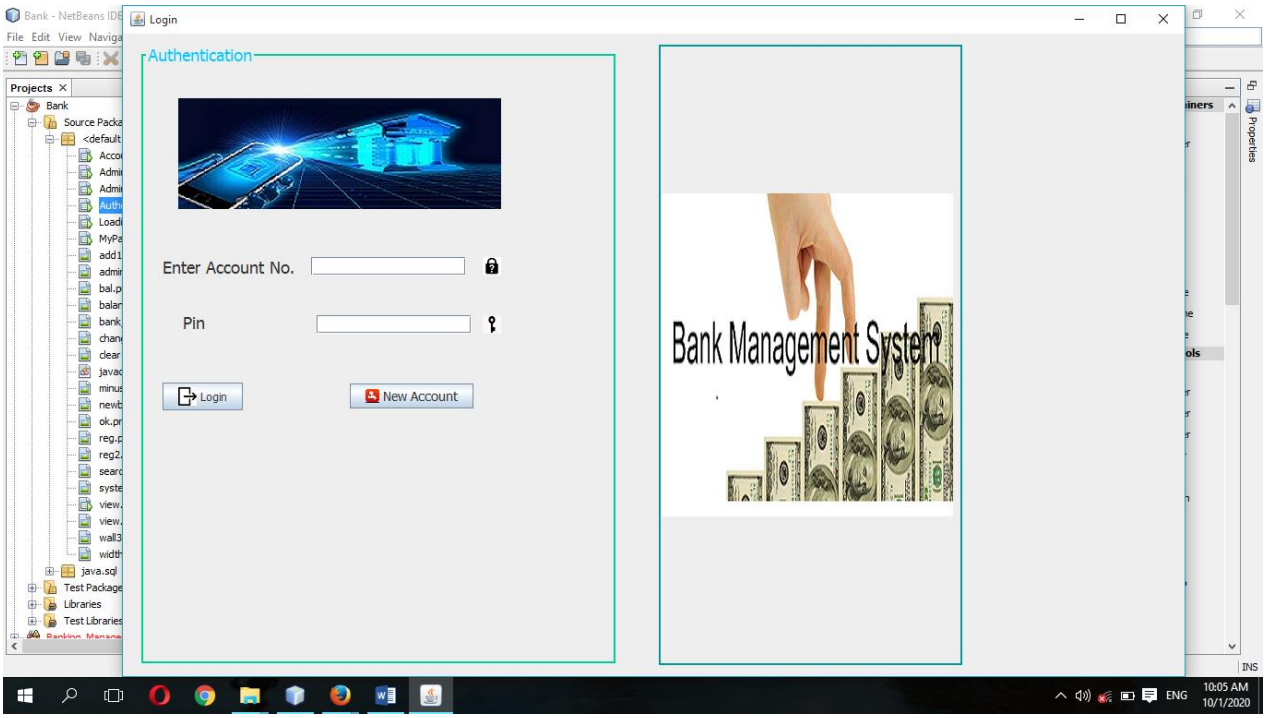
LOGGIN FUNCTION

These functions will log in a user based on a username and password being matched in a SQLite database. Values would be captured from a form and then passed to the main function.

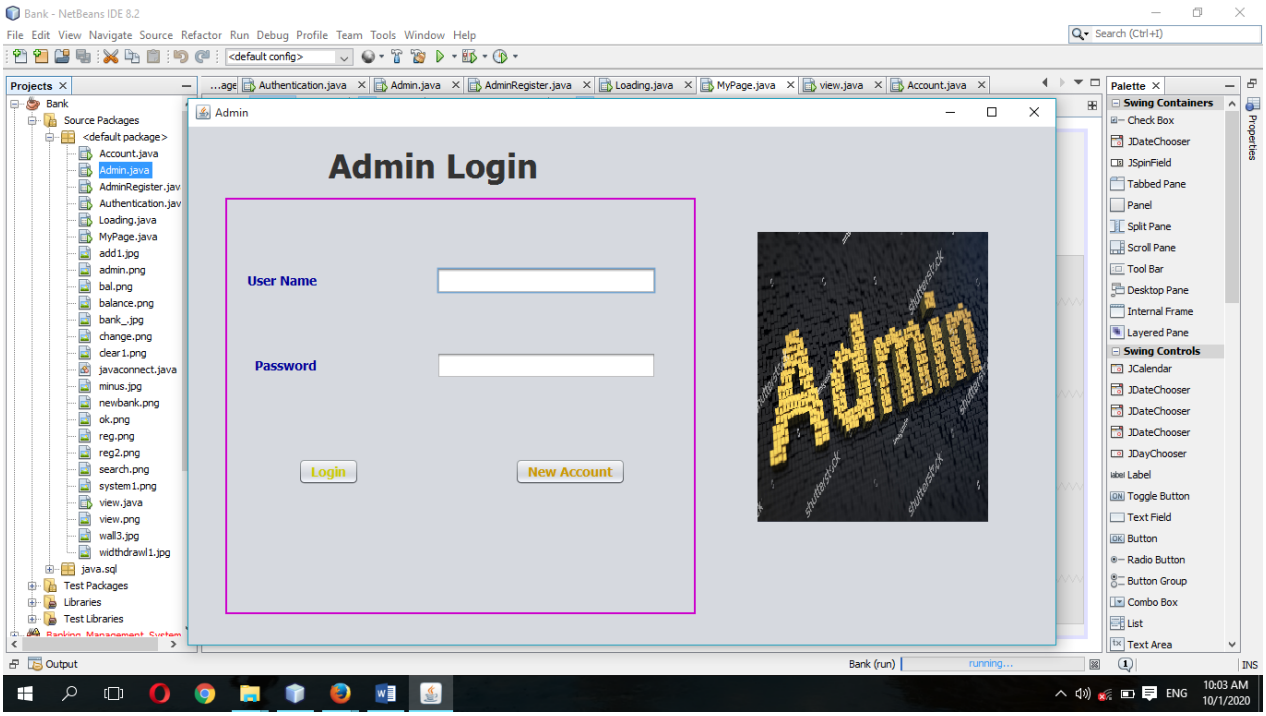
There are four logins in this system.

1. Admin login
2. User login

User Login



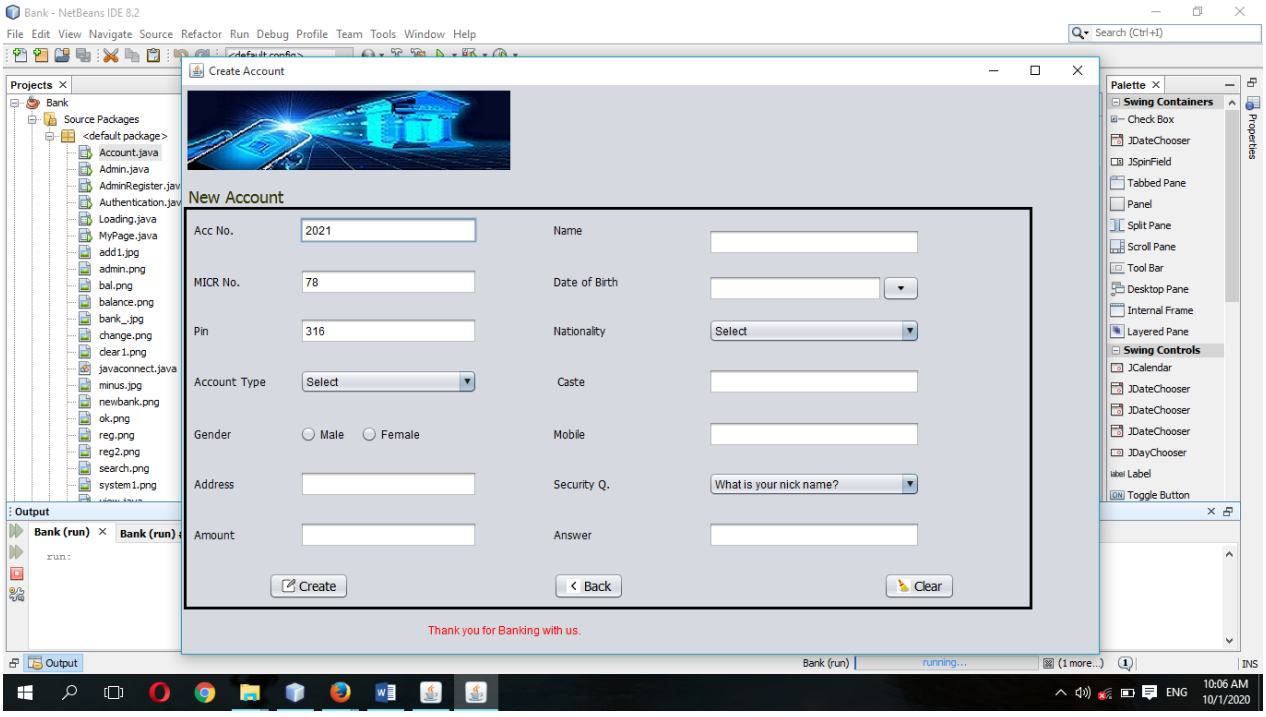
Admin Login


Bank - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Bank
Source Packages
<default package>
Account.java
Admin.java
AdminRegister.jav
Authentication.jav
Loading.java
MyPage.java
add1.jpg
admin.png
bal.png
balance.png
bank_.jpg
change.png
clear1.png
javaconnect.java
minus.jpg
newbank.png
ok.png
reg.png
reg2.png
search.png
system1.png
view.java
view.png
wall3.jpg
widthdrawl1.jpg
java.sql
Test Packages
Libraries
Test Libraries
Authentication.java
Admin.java
AdminRegister.java
Loading.java
MyPage.java
view.java
Account.java
Admin
Admin Login
User Name
Password
Login
New Account
Palette
Swing Containers
Check Box
JDateChooser
JSpinField
Tabbed Pane
Panel
Split Pane
Scroll Pane
Tool Bar
Desktop Pane
Internal Frame
Layered Pane
Swing Controls
JCalendar
JDateChooser
JDateChooser
JDateChooser
JDayChooser
Label
Toggle Button
Text Field
Button
Radio Button
Button Group
Combo Box
List
Text Area
Properties
Output
Bank (run)
running...
INS
ENG
10:03 AM
10/1/2020

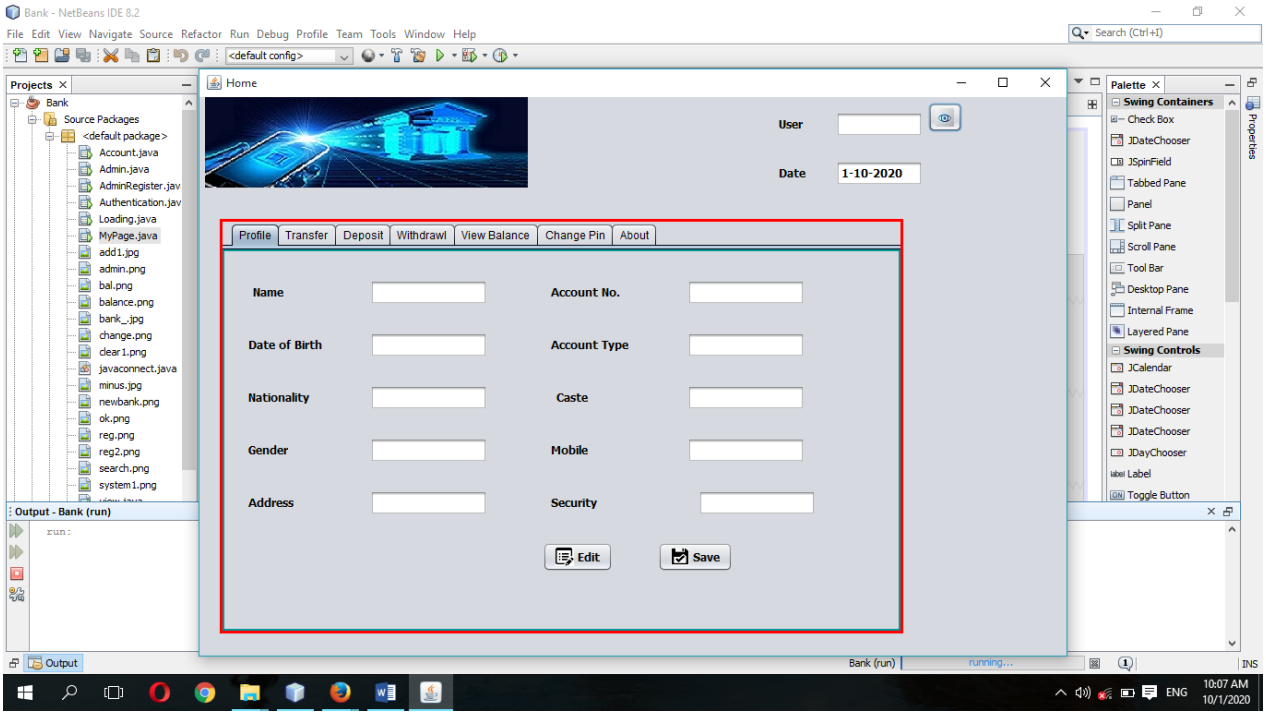
Update function

Update function is used to add data of user this system.

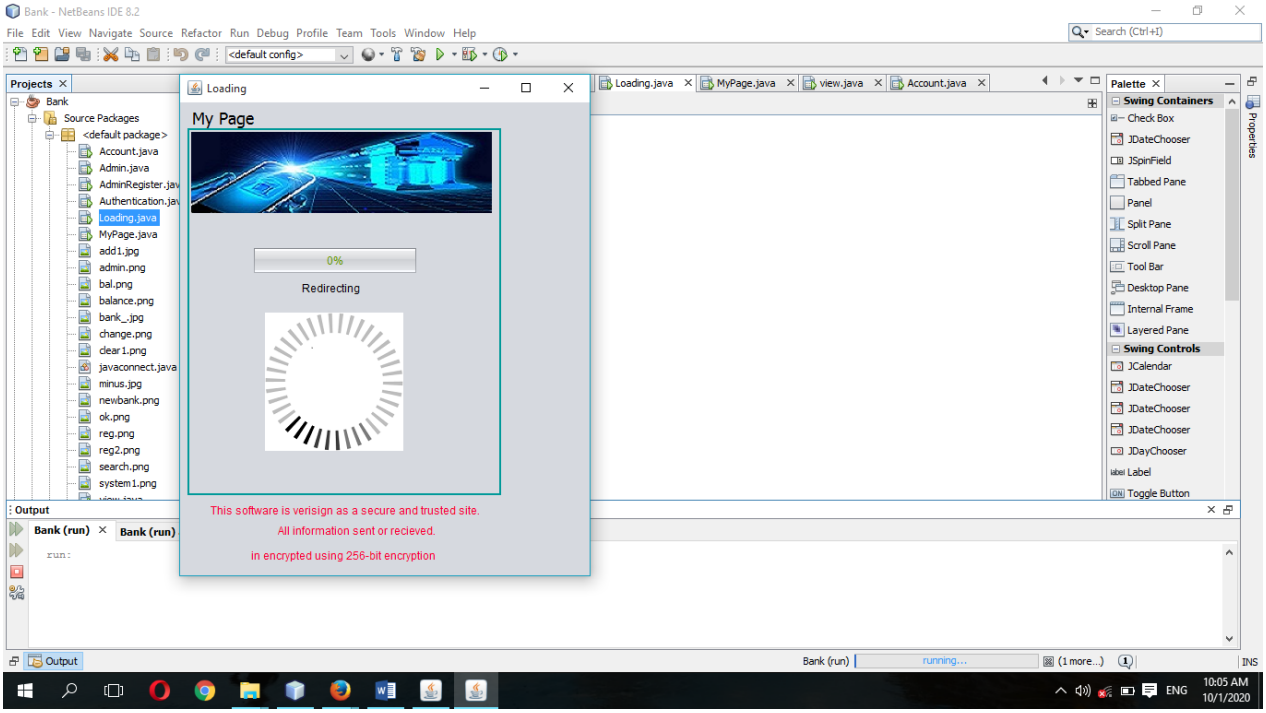
Patient Registration Form (my profile)



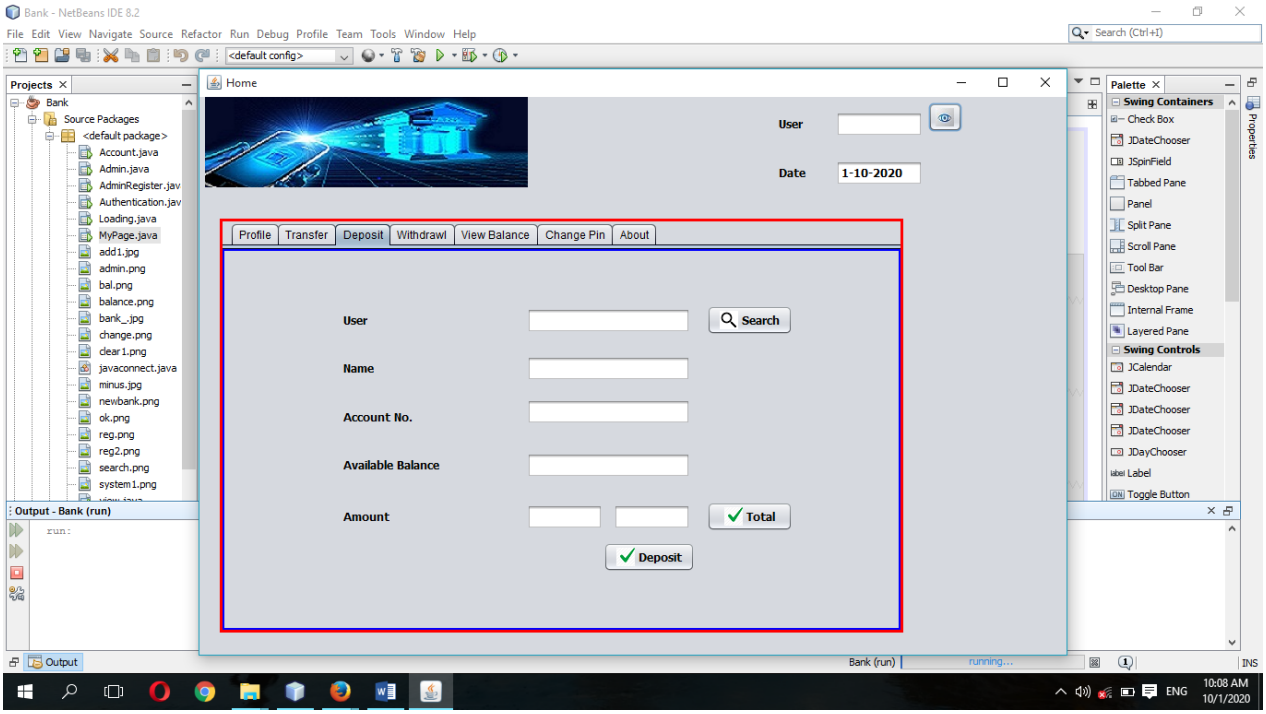
Edit patient profile form (my profile)



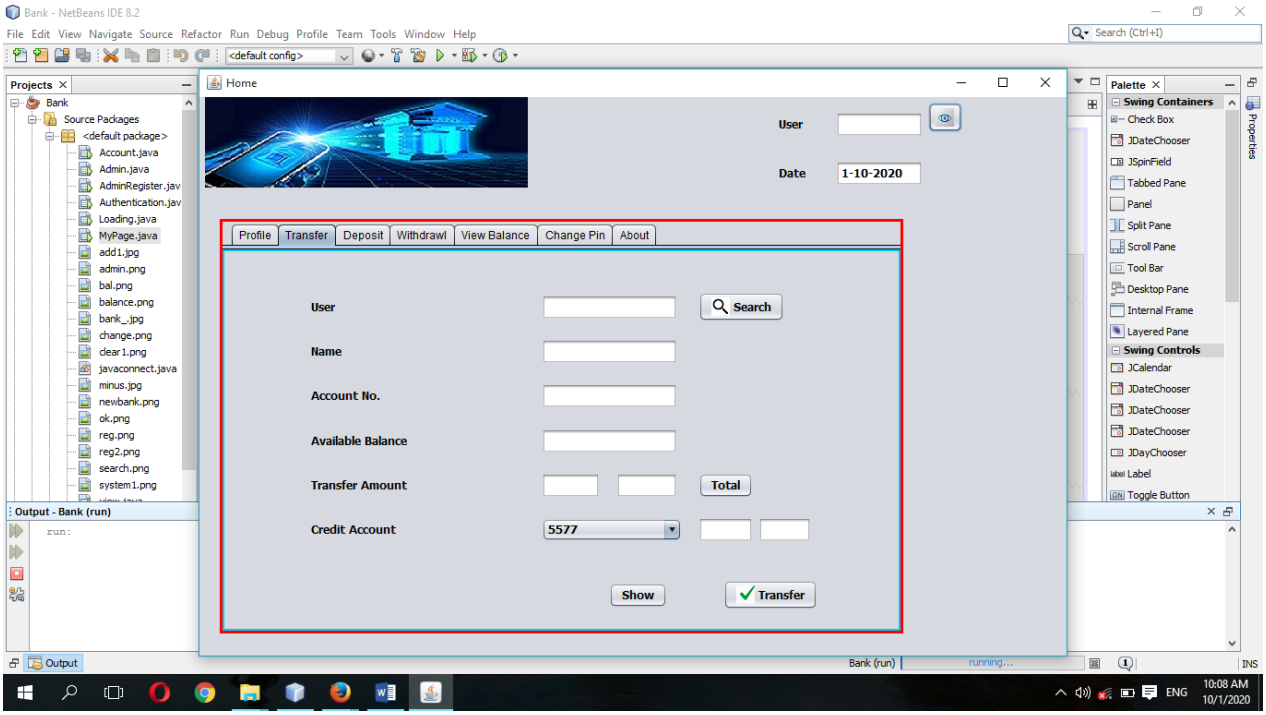
Loading page



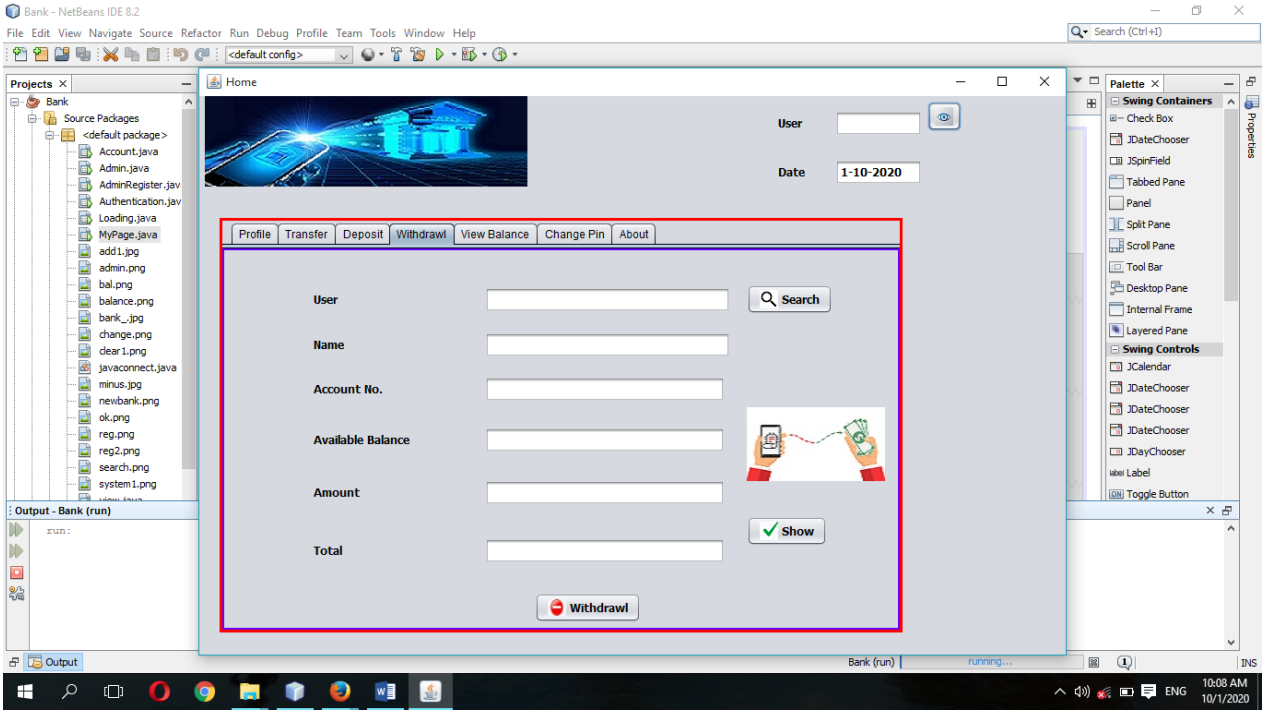
Deposit



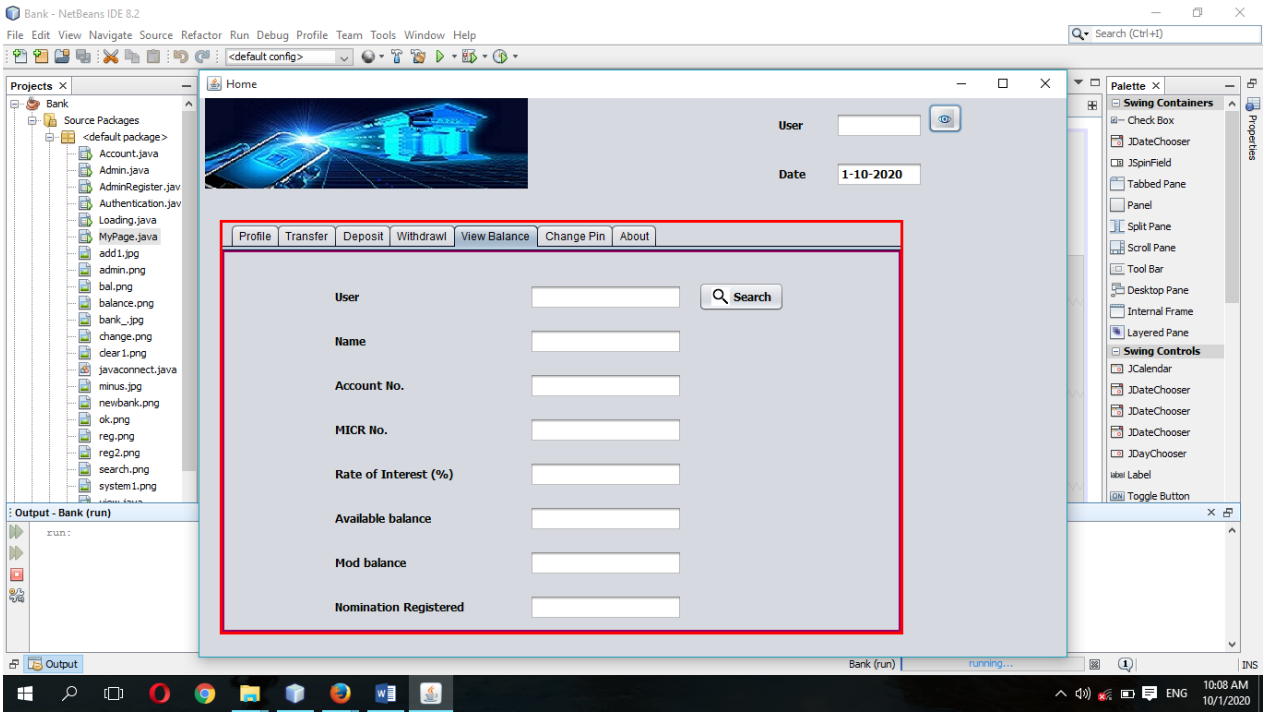
Transfer



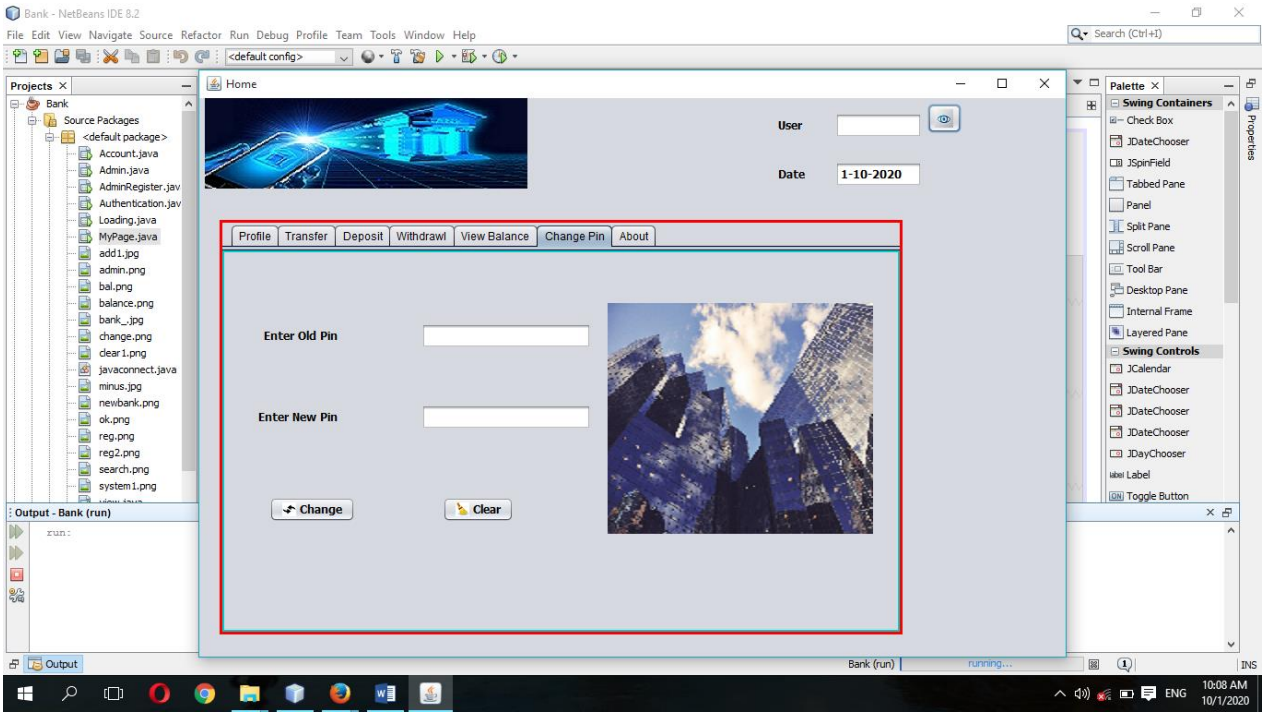
Withdrawal



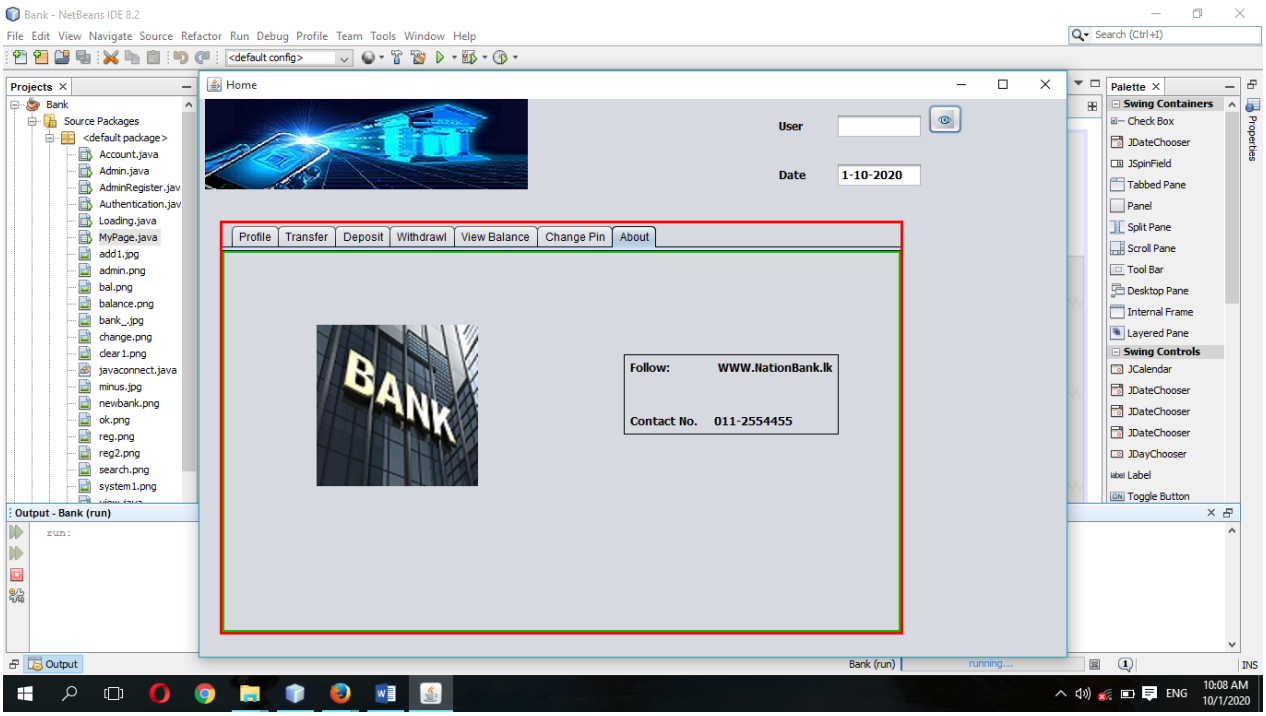
View Balance



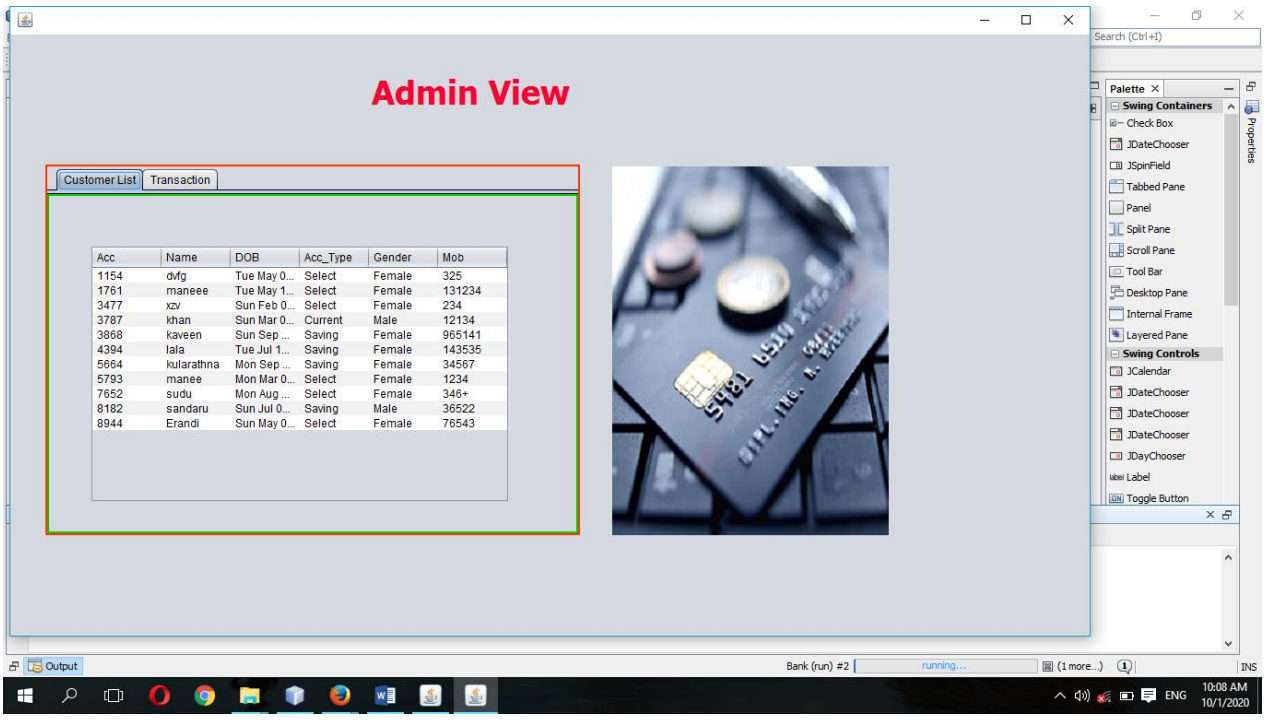
Change pin



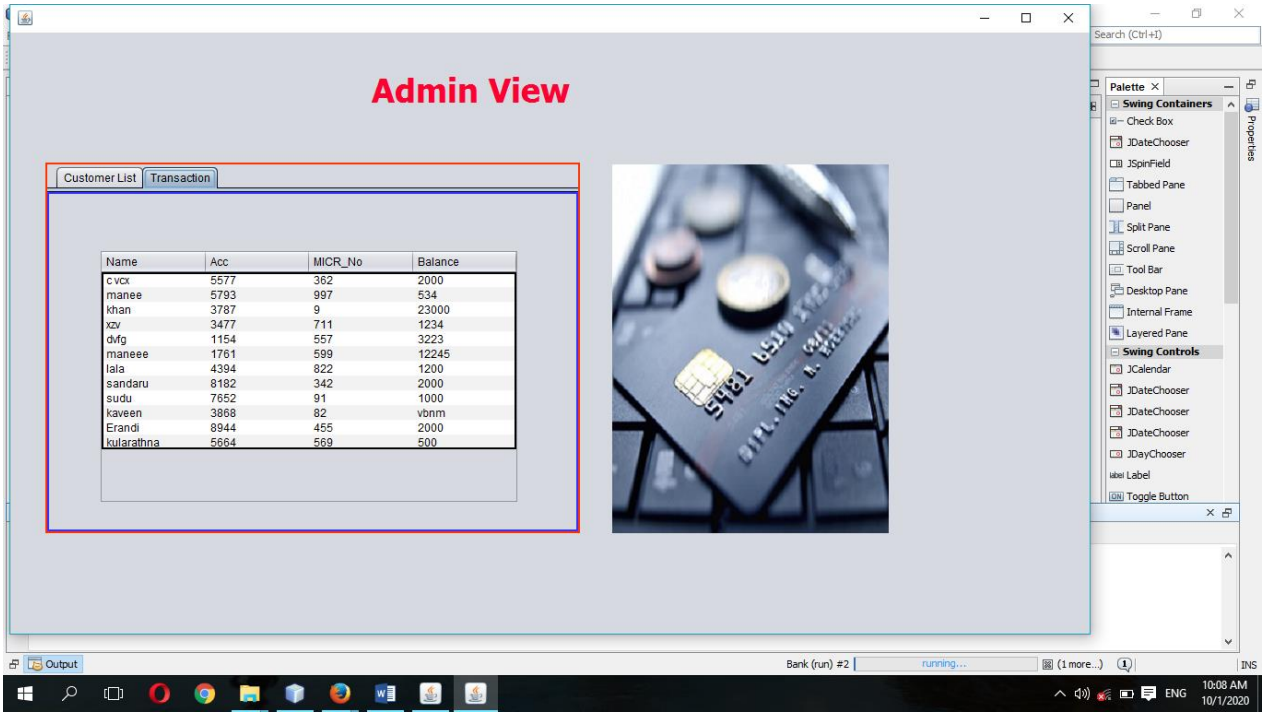
About



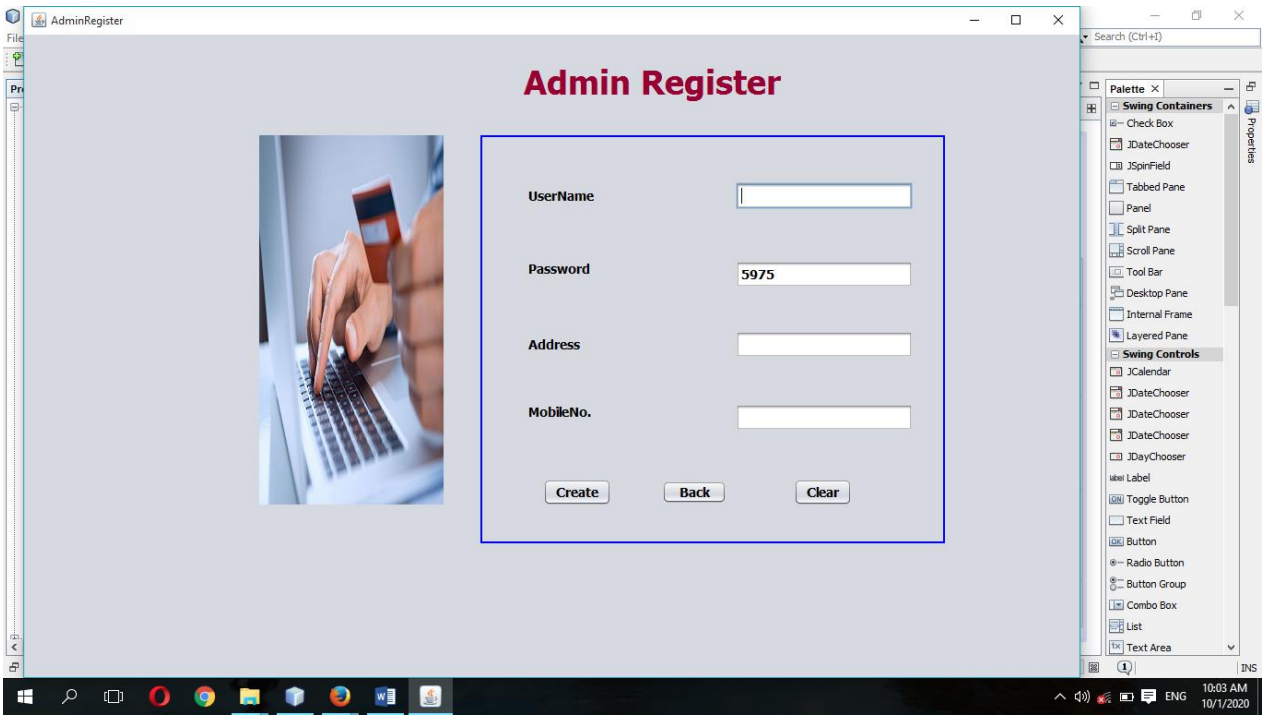
Admin view customer list



Admin view transaction list

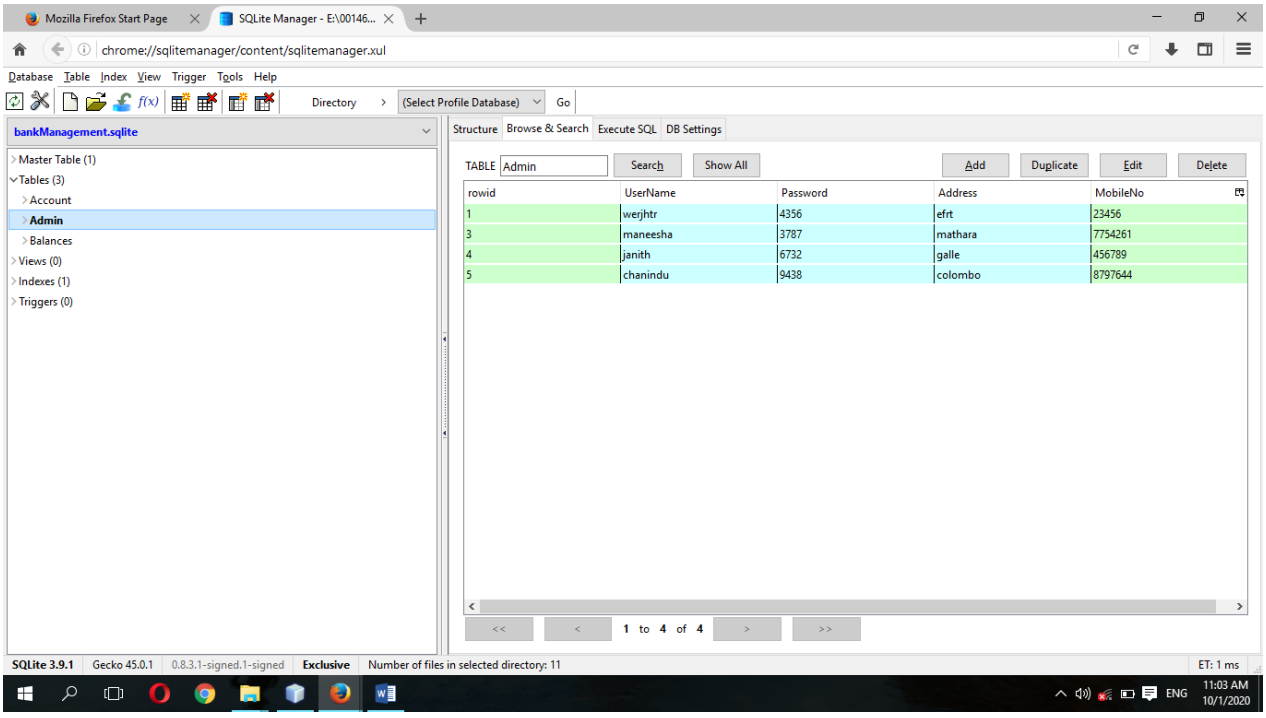


Admin Register Form



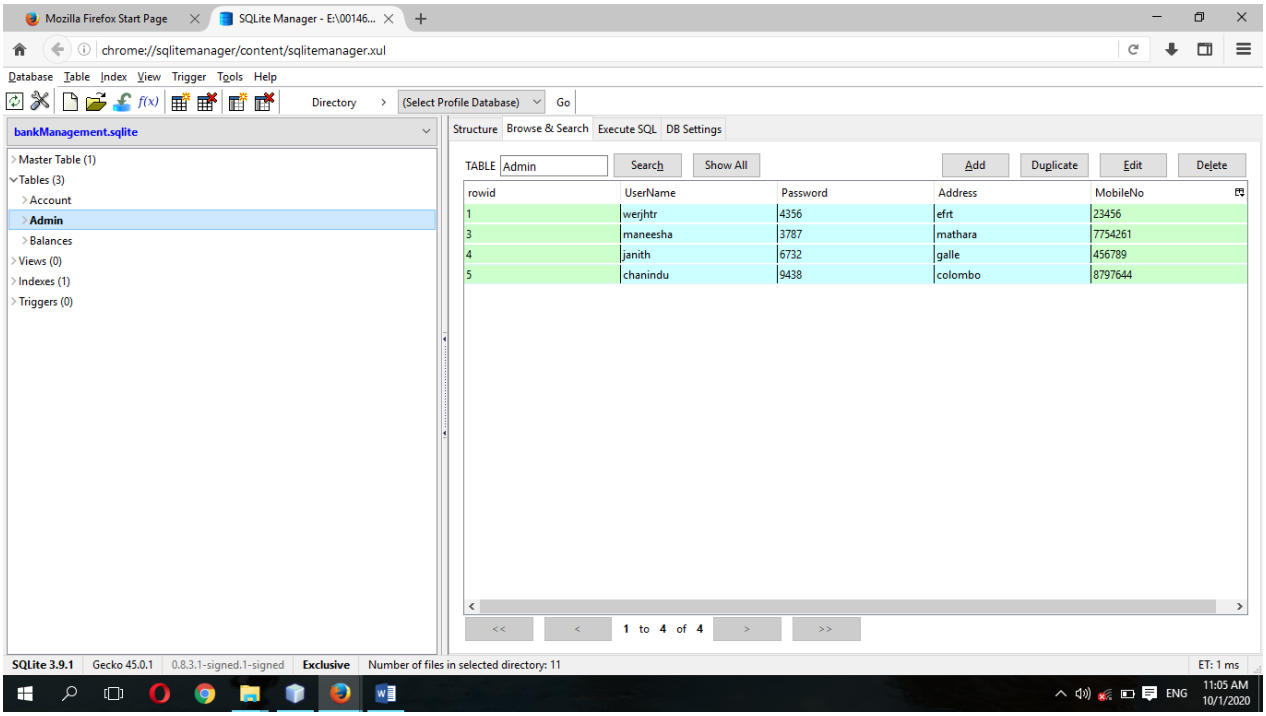
Data Structures

There are three table in this system which stores data.

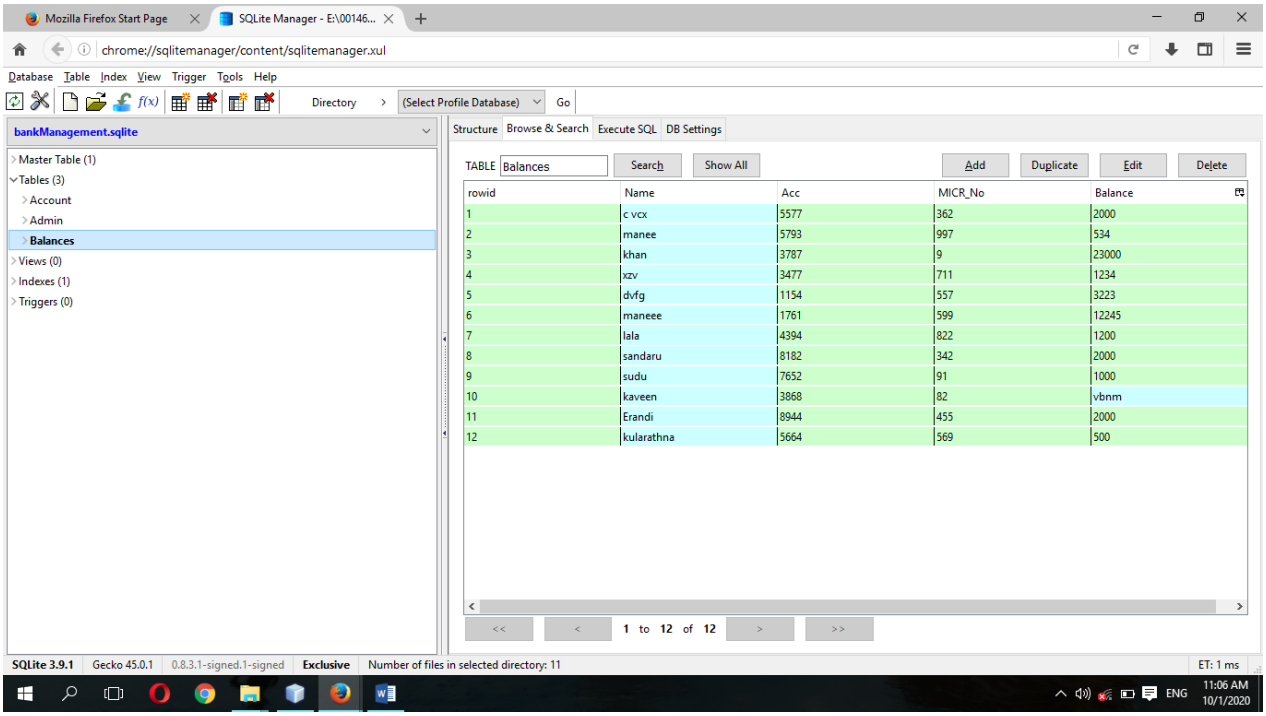


They are thoroughly normalized and designed in such a way to increase the efficiency of the system and remove data redundancy. The data structures of the system are presented below.

Admin Database

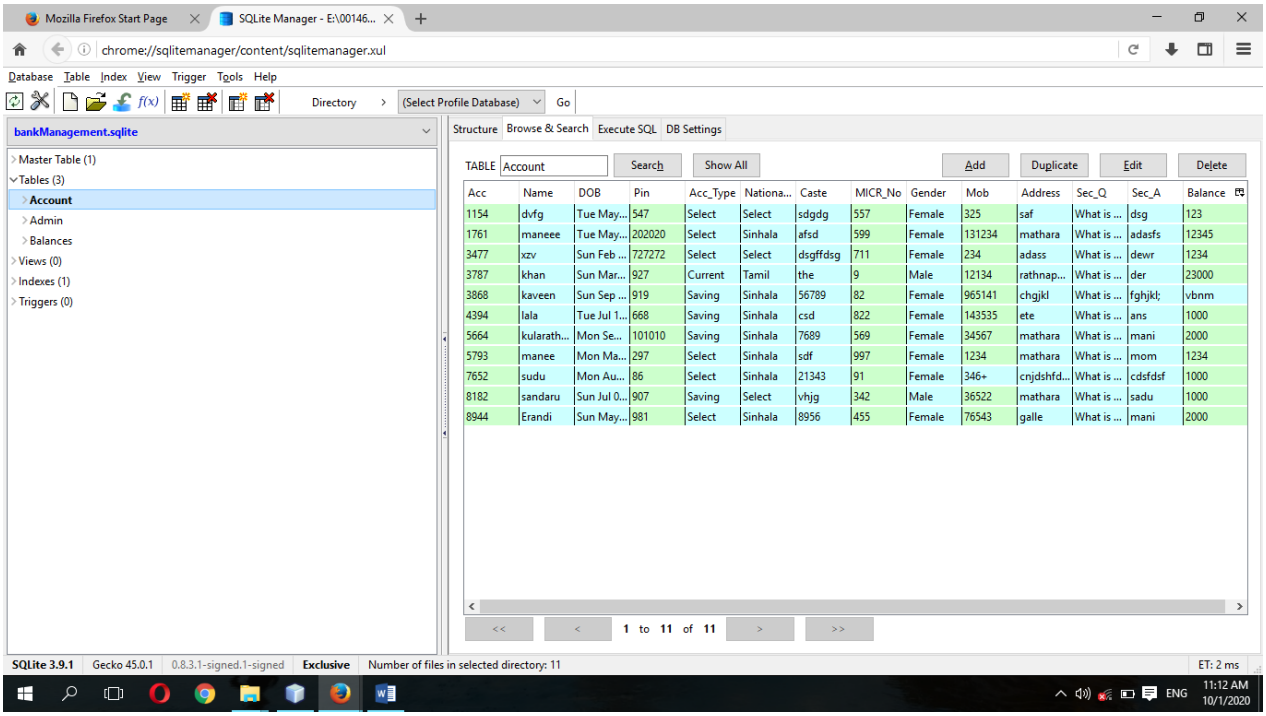


Balances Database



Account

The account table stores the details about the user. Account number is used as the primary key and it is increased automatically.



CHAPTER 06

System Implementation

User Login Code

```
String sql="select * from Account where Acc=? and Pin=? " ;
      try{
      pst=conn.prepareStatement(sql);
      pst.setString(1, jTextField1.getText());
      pst.setString(2, jTextField2.getText());
      rs=pst.executeQuery();
      if(rs.next()){
         setVisible(false);
         Loading ob=new Loading();
         ob.setUpLoading();
         ob.setVisible(true);
         rs.close();
         pst.close();

      }
      else{
         JOptionPane.showMessageDialog(null, "Incorrect Credential");


      }

      }catch(Exception e){
         JOptionPane.showMessageDialog(null, e);
```

```
    }finally{
      try{
      rs.close();
      pst.close();
      }catch(Exception e){

      }
    }
  }
```

Registration

```
 Random ra=new Random();
        jTextField1.setText(""+ra.nextInt(10000+1));
  }

  public void RandomMICR() {
  Random ra=new Random();
        jTextField2.setText(""+ra.nextInt(1000+1));
  }

  public void RandomPIN() {
  Random ra=new Random();
        jTextField3.setText(""+ra.nextInt(1000+1));
  }

  public void Bal(){
  String sql="insert into Balances(Name,Acc,MICR_No,Balance)values(?,?,?,?)";
  try{
  pst=conn.prepareStatement(sql);
  pst.setString(1, jTextField4.getText());
  pst.setString(2, jTextField1.getText());
  pst.setString(3, jTextField2.getText());
  pst.setString(4, jTextField10.getText());
```

```
pst.execute();

}catch(Exception e){
JOptionPane.showMessageDialog(null, e);
}
}
private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   setVisible(false);
   Authentication ob=new Authentication();
   ob.setVisible(true);

}

private void jTextField9ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   jTextField1.setText("");
   jTextField2.setText("");
   jTextField3.setText("");
   jTextField4.setText("");
```

```
        jTextField5.setText("");
        jTextField6.setText("");
        jTextField7.setText("");
        jTextField9.setText("");
        jTextField10.setText("");



    }


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String sql="insert into
Account(Acc,Name,DOB,Pin,Acc_Type,Nationality,Caste,MICR_No,Gender,Mob,Address,Sec_Q,Sec_A,Bal
ance)values (?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
        try{
        pst=conn.prepareStatement(sql);
        pst.setString(1, jTextField1.getText());
        pst.setString(2, jTextField4.getText());
        pst.setString(3, jXDatePicker1.getDate().toString());
        pst.setString(4,jTextField3 .getText());
        pst.setString(5,(String) jComboBox1.getSelectedItem());
        pst.setString(6,(String) jComboBox2.getSelectedItem());
        pst.setString(7,jTextField6.getText());
        pst.setString(8,jTextField2.getText());


        jRadioButton1.setActionCommand("Male");
        jRadioButton2.setActionCommand("Female");
        pst.setString(9, buttonGroup1.getSelection().getActionCommand());
        pst.setString(10,jTextField7.getText());
        pst.setString(11,jTextField5.getText());
        pst.setString(12,(String) jComboBox3.getSelectedItem());
        pst.setString(13,jTextField9.getText());
```

```
        pst.setString(14,jTextField10.getText());
        pst.execute();
        JOptionPane.showMessageDialog(null, " Contrtz\n Account has been created");
        Bal();
        } catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
        }
    }
```

Edit Profile

```
jTextField5.setEditable(true);
        jTextField6.setEditable(true);
        jTextField7.setEditable(true);
        jTextField10.setEditable(true);
        jTextField11.setEditable(true);
        jTextField12.setEditable(true);
```

Save

```
try{
        String
value1=jTextField5.getText();
        String
value2=jTextField6.getText();
        String value3=jTextField7
.getText();
        String
value4=jTextField10.getText();
        String
value5=jTextField11.getText();
        String
value6=jTextField12.getText();
```

```
        String
value7=jTextField3.getText();

        String sql="update Account set
Nationality='"+value1+"',Gender='"
+value2+"',Address='"+value3+"',Ca
ste='"+value4+"',Mob='"+value5+"',
Sec_Q='"+value6+"' where
Name='"+value7+"'";

pst=conn.prepareStatement(sql);
        pst.execute();
        pst.close();

JOptionPane.showMessageDialog(n
ull, "Profile Updated");
        }catch(Exception e){

JOptionPane.showMessageDialog(n
ull, e);



        }
    }
```

Change Password

```
        try{
            String value1=jTextField1.getText();
            String value2=jTextField42.getText();
            String sql="update Account set Pin='"+value2+"' where Name='"+value1+"'";
```

```
            pst=conn.prepareStatement(sql);
            pst.execute();
            JOptionPane.showMessageDialog(null, "Pin successfully Changed");



        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
```

Admin Login Code

```
String sql="select * from Admin
where UserName=? and
Password=?";
        try{

conn=javaconnect.connecrDb();

pst=conn.prepareStatement(sql);
            pst.setString(1,
jTextField1.getText());
            pst.setString(2,
jTextField2.getText());
            rs=pst.executeQuery();
            if(rs.next()){
                setVisible(false);
                rs.close();
                pst.close();
                conn.close();
                view ob=new view();
                ob.setVisible(true);
            }
            else{
```

```
JOptionPane.showMessageDialog(n
ull, "Incorrect Credential");


        }

      }catch(Exception e){

JOptionPane.showMessageDialog(n
ull, e);

      }finally{
         try{
         rs.close();
         pst.close();
         conn.close();
         }catch(Exception e){

         }
      }
   }
```

Withdrawal

```
 try{
        String
a1=jTextField27.getText();
        String
a2=jTextField32.getText();
        String sql="update Balances
set Balance='"+a2+"' where
Name='"+a1+"'";
```

```
pst=conn.prepareStatement(sql);
        pst.execute();

JOptionPane.showMessageDialog(n
ull, "Withdraw Successful");
      }catch(Exception e){

JOptionPane.showMessageDialog(n
ull, e);
      }

   }
```

Deposit

```
try{
        String
value1=jTextField13.getText();
        String
value2=jTextField18.getText();
        String sql="update Balances
set Balance='"+value2+"' where
Name='"+value1+"'";

pst=conn.prepareStatement(sql);
        pst.execute();

JOptionPane.showMessageDialog(n
ull, "Successfully Deposited");
        jTextField13.setText("");
        jTextField14.setText("");
        jTextField15.setText("");
```

```
            jTextField16.setText("");
            jTextField17.setText("");
            jTextField18.setText("");

        }catch(Exception e){

JOptionPane.showMessageDialog(n
ull, e);

        }
    }
```

Transfer

```
public void TransferC(){
        try{
            String value1=(String)
jComboBox1.getSelectedItem();
            String
value2=jTextField26.getText();
            String sql="update Balances
set Balance='"+value2+"' where
Acc='"+value1+"'";

pst=conn.prepareStatement(sql);
            pst.execute();

JOptionPane.showMessageDialog(n
ull, "Succesfully Transfered");




        }catch(Exception e){
```

```
JOptionPane.showMessageDialog(n
ull, e);
      }
   }

   public void TransferD(){
      try{
         String
value1=jTextField19.getText();
         String
value2=jTextField24.getText();
         String sql="update Balances
set Balance='"+value2+"' where
Name='"+value1+"'";

pst=conn.prepareStatement(sql);
         pst.execute();
      }catch(Exception e){

JOptionPane.showMessageDialog(n
ull, e);
      }

   }
```

View Balance

```
String sql="select * from Balances
where Name=?";
      try{

pst=conn.prepareStatement(sql);
```

```
        pst.setString(1,
jTextField33.getText());
        rs=pst.executeQuery();
           if(rs.next()){
               String
add1=rs.getString("Name");

jTextField34.setText(add1);

               String
add2=rs.getString("Acc");

jTextField35.setText(add2);

               String
add3=rs.getString("MICR_No");

jTextField36.setText(add3);

               String
add4=rs.getString("Balance");

jTextField38.setText(add4);
               jTextField37.setText("4
%");
               jTextField39.setText("Rs
0.00");

jTextField40.setText("No");

                }
      }catch(Exception e){
```

```
JOptionPane.showMessageDialog(n
ull, e);



      }
   }
```

Loading code

```
public Loading() {
        super("Loading");
       initComponents();
        th=new Thread((Runnable)this);
   }

   public void setUpLoading(){
   setVisible(false);
   th.start();
   }

    public void run(){
    try{
    for(int i=0;i<=200;i++){
    s=s+1;
    int
m=jProgressBar1.getMaximum();
    int v=jProgressBar1.getValue();
    if(v<m){

jProgressBar1.setValue(jProgressBar
1.getValue()+1);
    }else{
```

```
i=201;
setVisible(false);
MyPage ob=new MyPage();
ob.setVisible(true);
}
Thread.sleep(50);

}

}catch(Exception e){



}
}
```

CHAPTER 5

Software Testing

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits. The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out. There are two major type of testing they are

1. White Box Testing
2. Black Box Testing

White Box Testing

White box sometimes Called 'Glass box testing' is a test case design uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests were made on the system.

a) All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed.
b) All logical decisions were checked for the truth and falsity of the values.

Black box Testing

Black box testing focuses on the functional requirements of the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. It identifies

1. Interface errors
2. Performance in data structure
3. Performance errors
4. Initializing and termination errors

Test Cases

If you click login button without entering all required values for user name and password an error message is displayed.

CHAPTER 5

OVERALL DESCRIPTION

The more types of users for the banking Management System are:

• Admin

• Users

The following table describe general user characteristics that will affect the functionality of the software product.

Type of User	User Characteristic	How the user characteristic and technical expertise affect system functionality
Admin	Good understanding to banking management system.	User interface with less input steps. Easy to learn.
Users	Will not have any formal training to use the system.	GUI interface may be easier to learn than text interface. Provide system help. Provide appropriate error messages for invalid user

CHAPTER 07

Evaluation

According to the project proposal we hope to design this project several phrases. Now we think we have reached near the project proposal. When the database system is completed built in implementation phase, the database system is then evaluated by using real data and also through is connectivity with interface that is develop. The purposed of evaluating the database system is to ensure it perform as designed and meet its specification. Maintenance involves monitoring the database. It is important to correct errors and improve the future implementation of the database system.

CHAPTER 08

Conclusion

This project is based on PC application. This system can be used in any medical Centre. All the tasks related to their business environment can be easily and efficiently done by using this system. This system helps to channel doctors easily and it also helps to make further arrangements regarding the business. This system can be used easily by any person.

References

- www.wikipedia.com
- www.youtube.com
- https://code.visualstudio.com/docs/languages/csharp
- https://www.learncs.org/
- https://www.tutorialspoint.com/csharp/
- https://stackoverflow.com/