

Acknowledgement

I take this opportunity to express my gratitude to everyone who helped me and guided me to complete the project successfully. I wish to thank my project supervisors Miss Sazna, lecturer in Esoft Metro Campus, ,and Miss.Shafia, lecture in Esoft Metro Campus who gave the technical guidance and constant supervision despite their busy schedule,also Mr.Udara Prabath for giving me the domain knowledge and providing necessary information and documents regarding the project.

Table of Contains

1. Acknowledgement	01
2. Table of Contains.....	2-5
3. Abstract.....	06
4. Introduction.....	7-8
3.1 Introduction.....	07
3.2 Project Overview.....	08
5. System Analysis.....	9-16
5.1 Background & Motivaion.....	09
5.1.1 Features.....	09
5.1.2 Advantages.....	09
5.1.3 Disadvantages.....	09
5.2 Problem in Brief.....	10
5.3 Objectives & Aims.....	10
5.3.1 Objectives.....	10
5.3.2 Aims.....	10
5.4 Proposed Solution.....	11
5.5 ER Diagram.....	12
5.6 Use Case Diagram.....	13
5.7 Class Diagram.....	14
5.8 Flowcharts.....	15-20
5.8.1 Add.....	15
5.8.2 Edit.....	16
5.8.3 Save.....	17
5.8.4 Delete.....	18

5.8.5 Search.....	19
5.8.6 Login.....	20
5.9 Resource Requirements.....	21
5.9.1 Functional Requirement.....	21
5.9.2 Non Functional Requirement.....	21
5.9.3 Requirement Gathering Techniques.....	21
5.9.4 Software Requirement.....	21
5.9.5 Hardware Requirement.....	21
6. Software Development Methodology.....	22-26
6.1 Development Process Models.....	22
6.1.1 Introduction.....	22
6.1.2 Traditional Models.....	22
6.1.3 Agile Models.....	22
6.2 What is SDLC?.....	22
6.2.1 Waterfall Models.....	22
6.2.2 Stages Diagram.....	23
6.2.3 Principal Staged of Waterfall Model.....	24
6.2.4 Table.....	25
7. Software Design.....	26-
7.1 Designs.....	26
7.1.1 Database Design.....	26
7.1.2 Interface Design.....	27
7.1.3 Java Netbeans.....	27
7.2 Software Explain.....	28
7.2.1 eBank Software.....	28
7.2.2 eClient Software.....	28
7.3 Software Interfaces & Codes.....	29
7.3.1 eBanking Software.....	29-111
7.3.1.1 Database Connection	

- 7.3.1.2 Login
- 7.3.1.3 Home Screen
- 7.3.1.4 Select Account Type
- 7.3.1.5 New Bank Account
- 7.3.1.6 Bank Account Search
- 7.3.1.7 Bank Account Details
- 7.3.1.8 Bank Account Details Print
- 7.3.1.9 New Client Account
- 7.3.1.10 Client Account Search
- 7.3.1.11 Client Account Details
- 7.3.1.12 Client Account Details Print
- 7.3.1.13 Money Deposit
- 7.3.1.14 Deposited Search
- 7.3.1.15 Deposited Details
- 7.3.1.16 Deposited Details Print
- 7.3.1.17 Money Transfer
- 7.3.1.18 Transaction Search
- 7.3.1.19 Transaction Details
- 7.3.1.20 Transaction Details Print
- 7.3.1.21 Employee Menu
- 7.3.1.22 New Employee
- 7.3.1.23 Employee Search
- 7.3.1.24 Employee Details
- 7.3.1.25 Employee Details Print
- 7.3.1.26 Bank Account Type Details
- 7.3.1.27 Reports
- 7.3.1.28 Report Print
- 7.3.1.29 About Bank
- 7.3.1.30 Help

7.3.1.31 Menu Bar	
7.3.1.32 Logout	
7.3.2 eClient Software	112-162
7.3.2.1 Login	
7.3.2.2 Welcome (Home Screen)	
7.3.2.3 Cash Deposit	
7.3.2.4 Money Transfer	
7.3.2.5 Deposited Details	
7.3.2.6 Deposited Details Print	
7.3.2.7 Transfaction Details	
7.3.2.8 Transaction Details Print	
7.3.2.9 Account Information	
7.3.2.10 Change Password	
7.3.2.11 Forgot Password	
7.3.2.12 About Us	
7.3.2.13 Contact Us	
7.3.2.14 Online Chat	
7.3.2.15 Menu Bar	
7.3.2.16 LogOut	
8. Development.....	163
8.1Future Scope.....	163
8.2Testing.....	163
8.3 Implementation.....	163
9.Evalution & Conclution.....	164-165
10. Reference.....	166
11. Note.....	167

Abstract

This project was developed to handle Banking System. It is a useful software for the clients and Employees in the Bank.

When going through the procedure of the client and employee keeping records of the accounts we identified several weakness such as customers and staff members hand writings not been clear and uncompleted data included in forms. It takes a long time to find account details and other information safety of the documents is very low. These mistakes could cost a Bank great deal.

By using this computer program I am hoping to introduce a new face of management system by using Java [Netbeans] which is complete computer programming language.

This software introduces reliable, safe, computer program for account details and other details keeping.

Normally Account details should be filled, updated and edited by the staff by hand by using this software the user can easily manipulate the whole account details, Transaction details, Client Details by just clicking some button.

Introduction

Information Communication Technology is widely utilized for many applications at present. Almost all the processes are being computerized to bring more efficiency and reliability in them. People have the tendency to think that machines are more reliable to handle sensitive and non-sensitive data. That is why most of the processing is done by machines. Sri Lanka, being a developing country, encourages automation of the processes of companies. Therefore, this Computer-base Bank Management System is built to make a contribution to the productivity enhancement of our country's Bank Management. Tea is a beverage consumed by numerous people throughout the world. This software was built to withstand the some hardship the clients and employees were facing. With this introduction of this product the client and employee will be able to resolve a lot of problem they have with the present documented system.

I hope used by this system in Bank Management System.

Project Overview

In the modern world technology field is very popular. So most systems must upgrade to smart type. Not only that nowadays people used to find easy way to doing works. They are little bit lazy to used paper work. As a solution that in this paper we describe about Online Bank Management System.

This system will important to employees of the bank and customers. They can find all the informations. The system allows them to make manual deposits, money transfers, account details and more. It will be easier more because there are 2 softwares named eBank system and eClient System. This system can generate all the information reports. Not only that this will help to the bank to do their works will more efficiency and more effectively.

System Analysis

❖ Background & Motivation

In the background of Bank management system all the manual and paper work which was done in the past would be removed by the use the computer technology different huge size register were used to maintain. The records of books and details about books issues etc. All these this would be software based and customize report would be generate paper work would be lesser. This project of ONLINE BANK MANAGEMENT SYSTEM of give use the complete information about the Customers, Bank Account Details, Transaction Details, Deposited Details, salary, employee details etc.

➤ Features

- Deposit Money.
- Transfer Money.
- Enter, Update, Delete data.
- Preparing Reports.

➤ Advantages

- Easy to Handel.
- More accuracy.
- Fast responding Easy to maintain.
- Large data can store within small space.

➤ Disadvantages

- Electricity problems.
- Expensive. Expert Needed.
- Lack of knowledge about system.
- More exposed due to electricity problem and virus attack.

Problem in Brief

Bank have any main process. In the Bank Management system they have more than 1000 Clients to give their service. Bank data entry operator update their system with this all data And data entry operator can't handle this much data from file base system. That's a wasting time, Low of accuracy and if they lost one file it is effected to entire process. That's why Bank Management System come forward as a solution.

❖ Objectives & Aims

➤ Aim

- Develop a Software tool to eBank for do their work easily.

➤ Objective

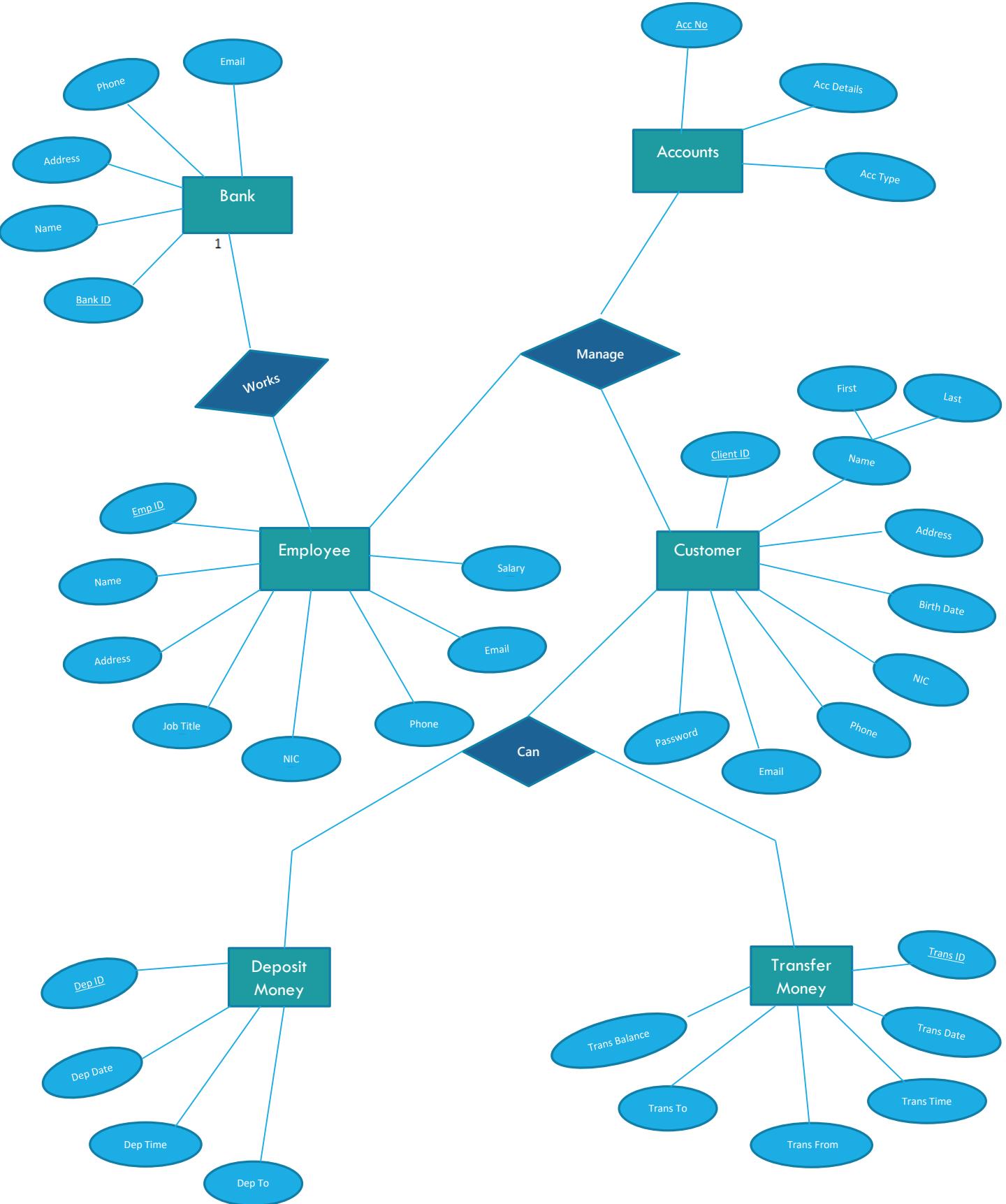
- To analyze the behavior of customers in the area
- To store the Account information and other information.
- To manage the clients details of the bank
- To provide service to money transfer and deposit at a time.
- To generate the reports about all the informations.
- To manage all the employee details.

Proposed Solution

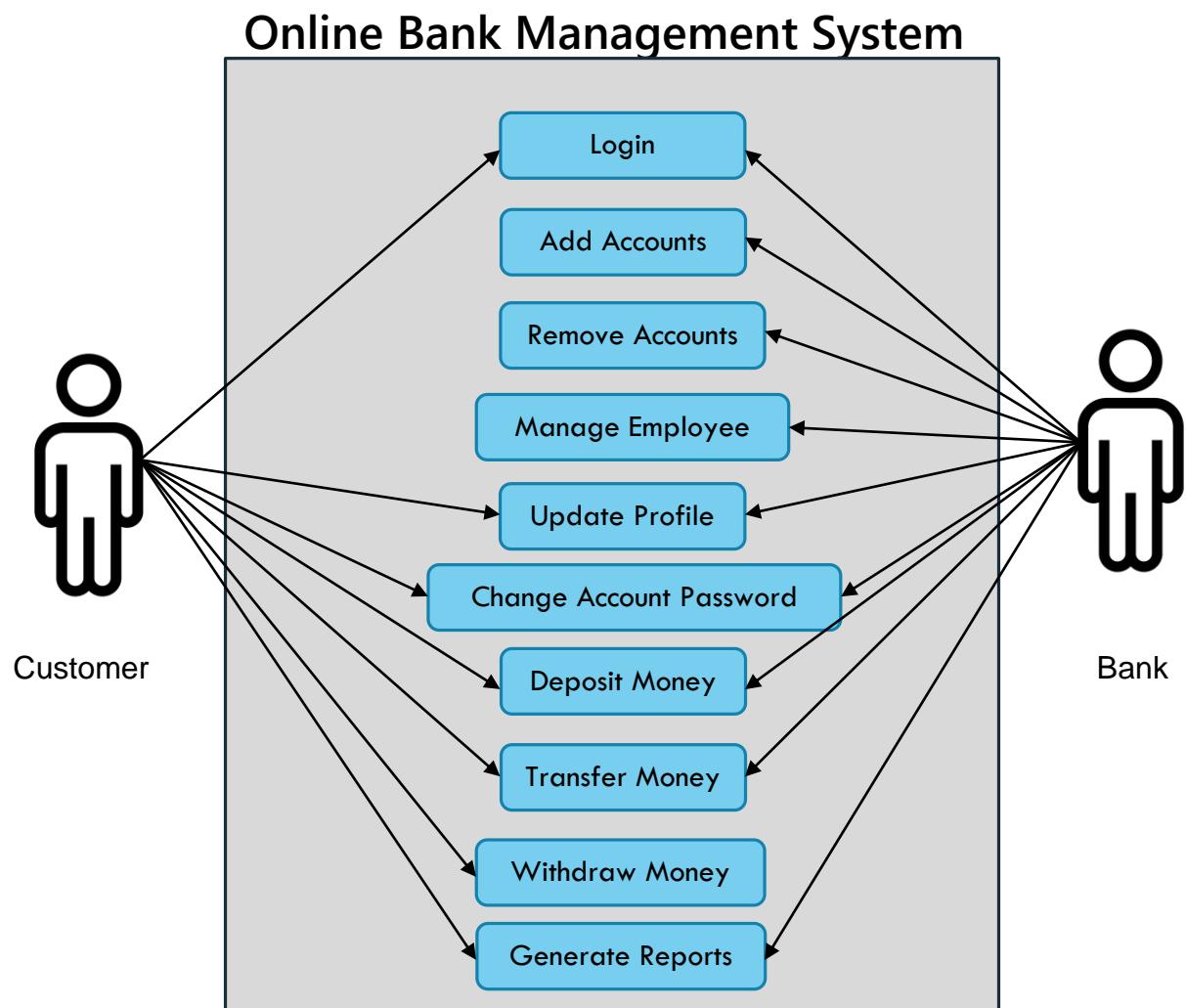
There are lots of problems in the file base system. So if the bank need to upgrade their system to modern technologies. Then they will can work more accuracy and give the valuable and trustful service to the client and customers. I decided to create this system under the below entities.

- **System operator.**
- **eBank System.**
- **eClient System**

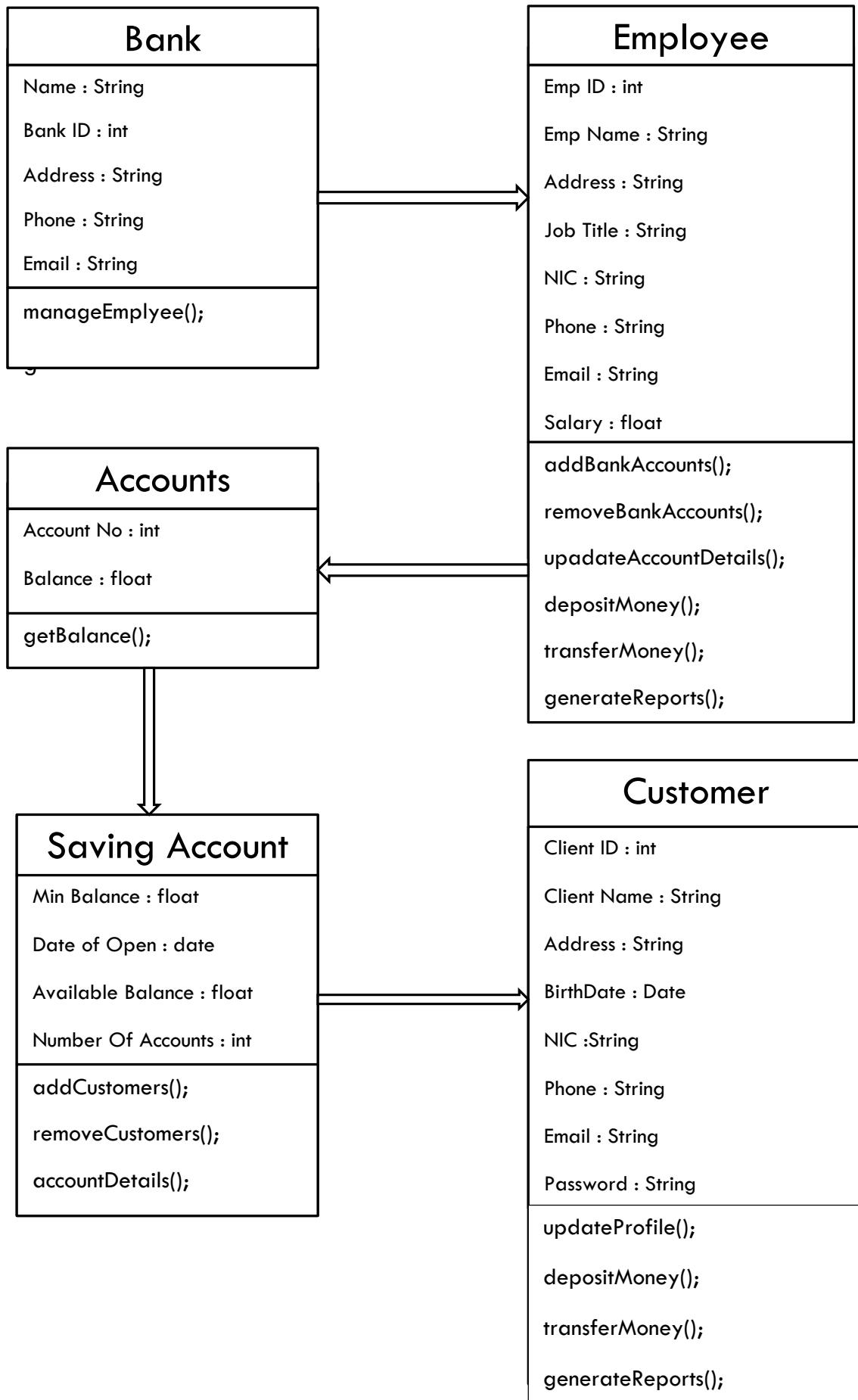
❖ ER Diagram



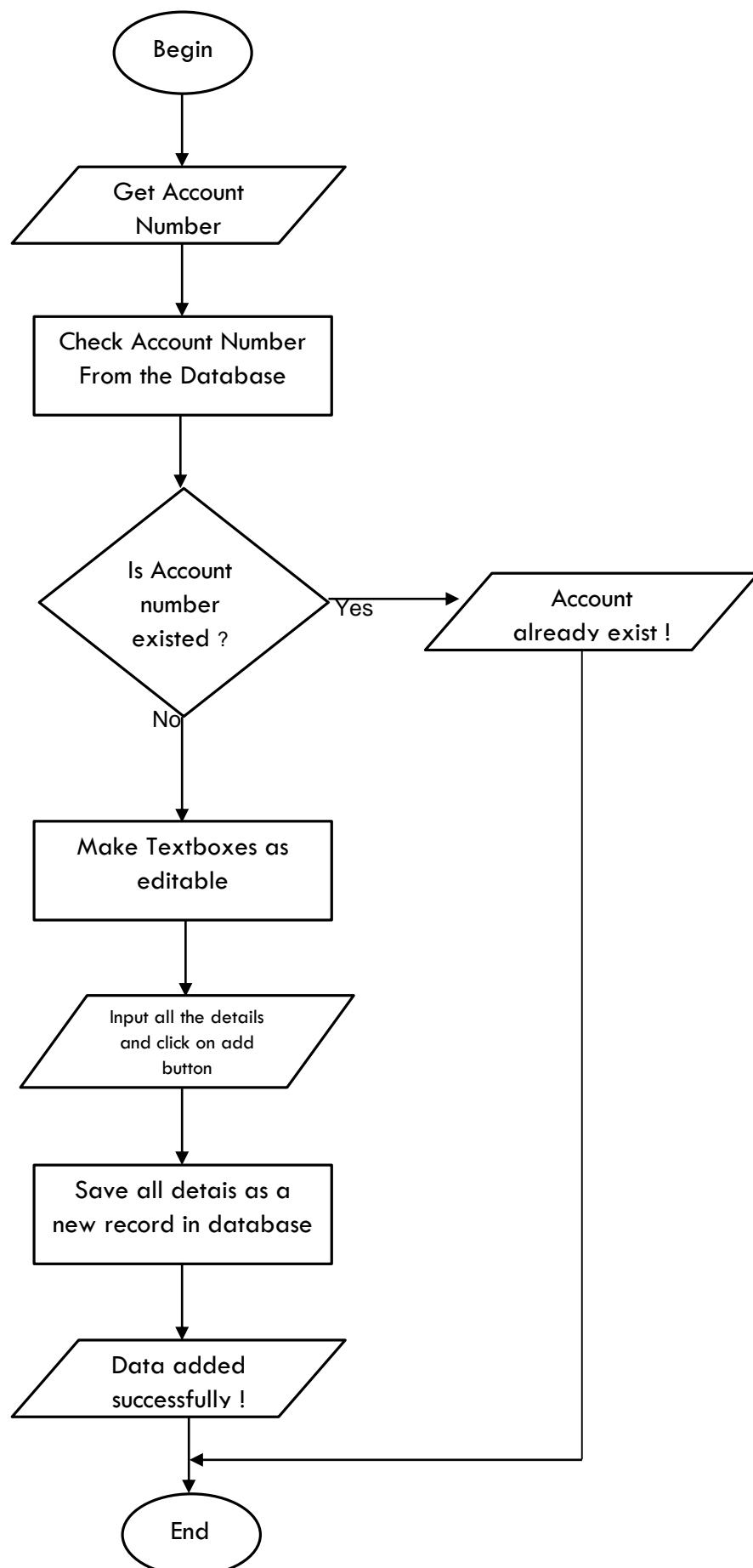
❖ Use Case Diagram



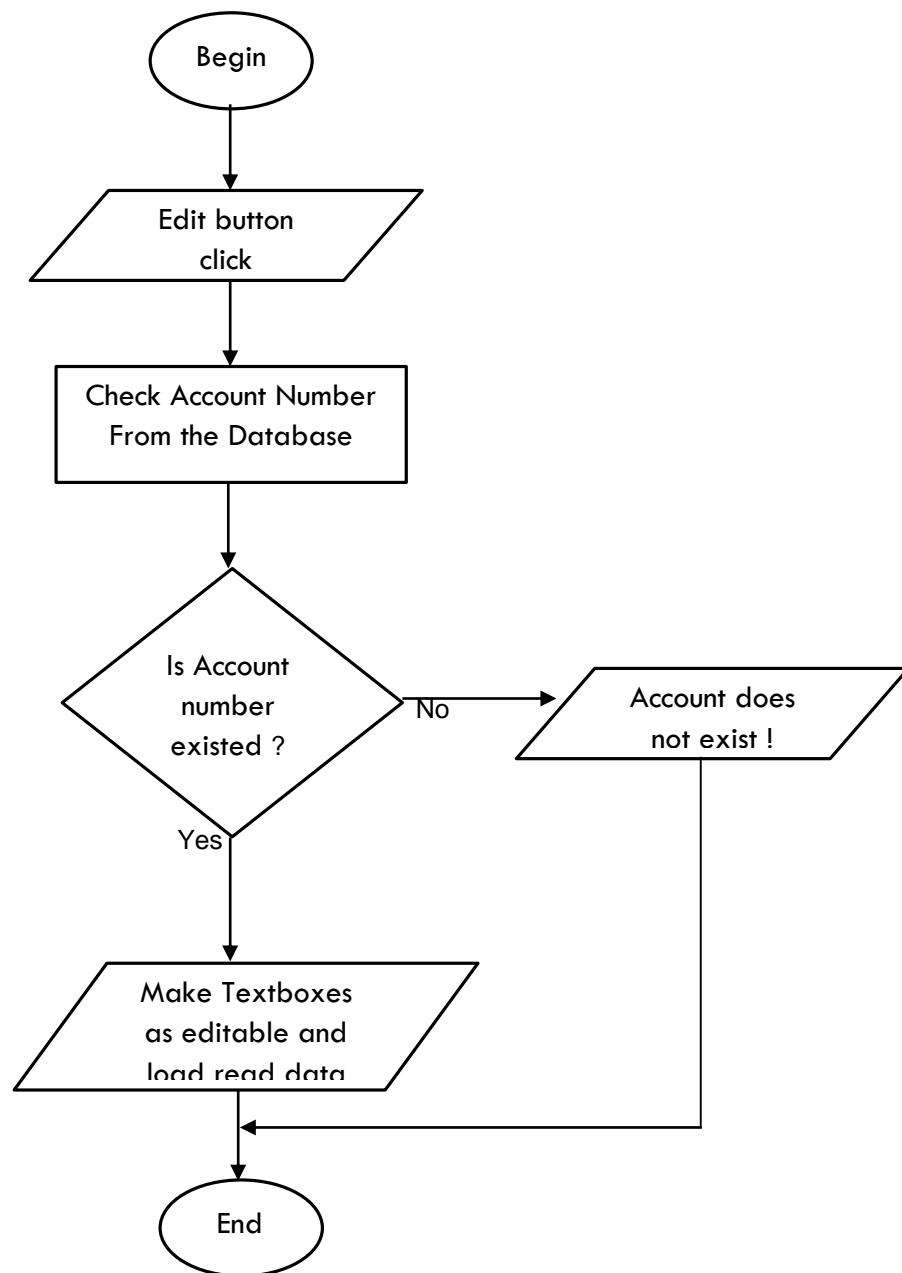
❖ Class Diagram



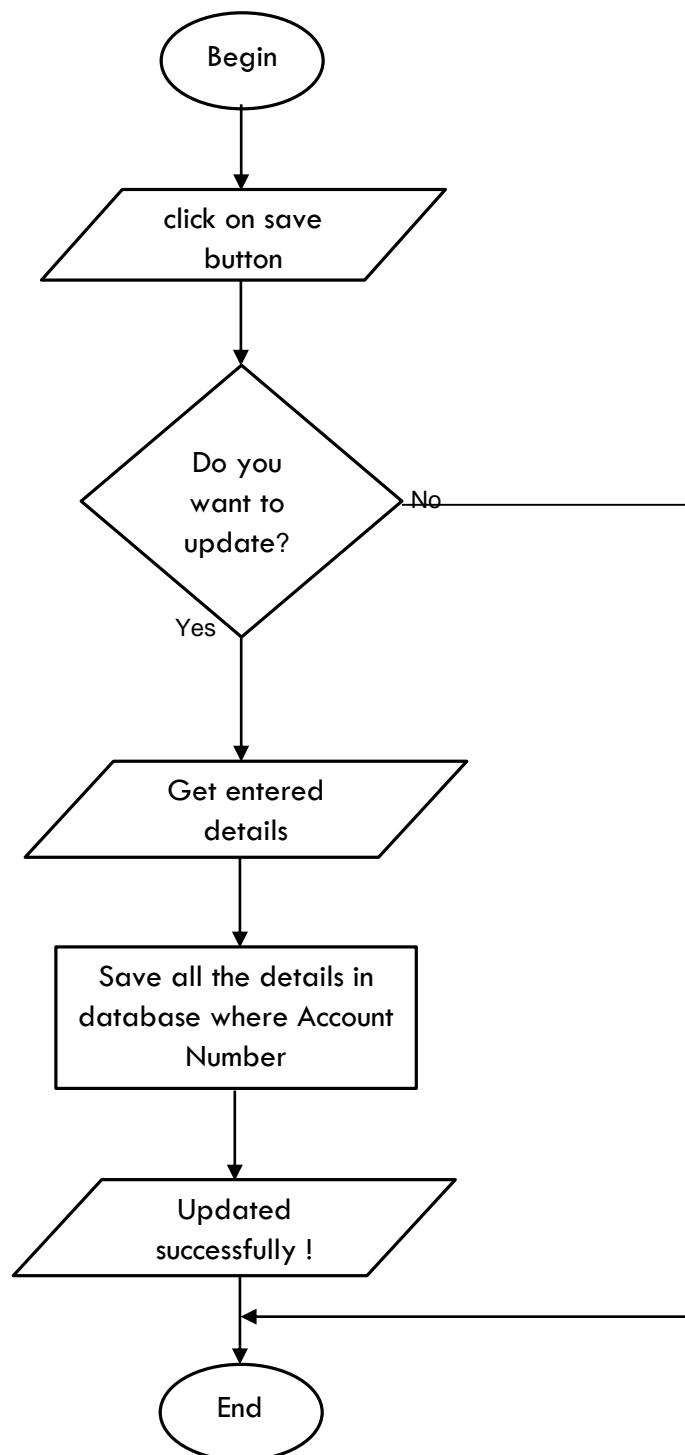
❖ Add Button



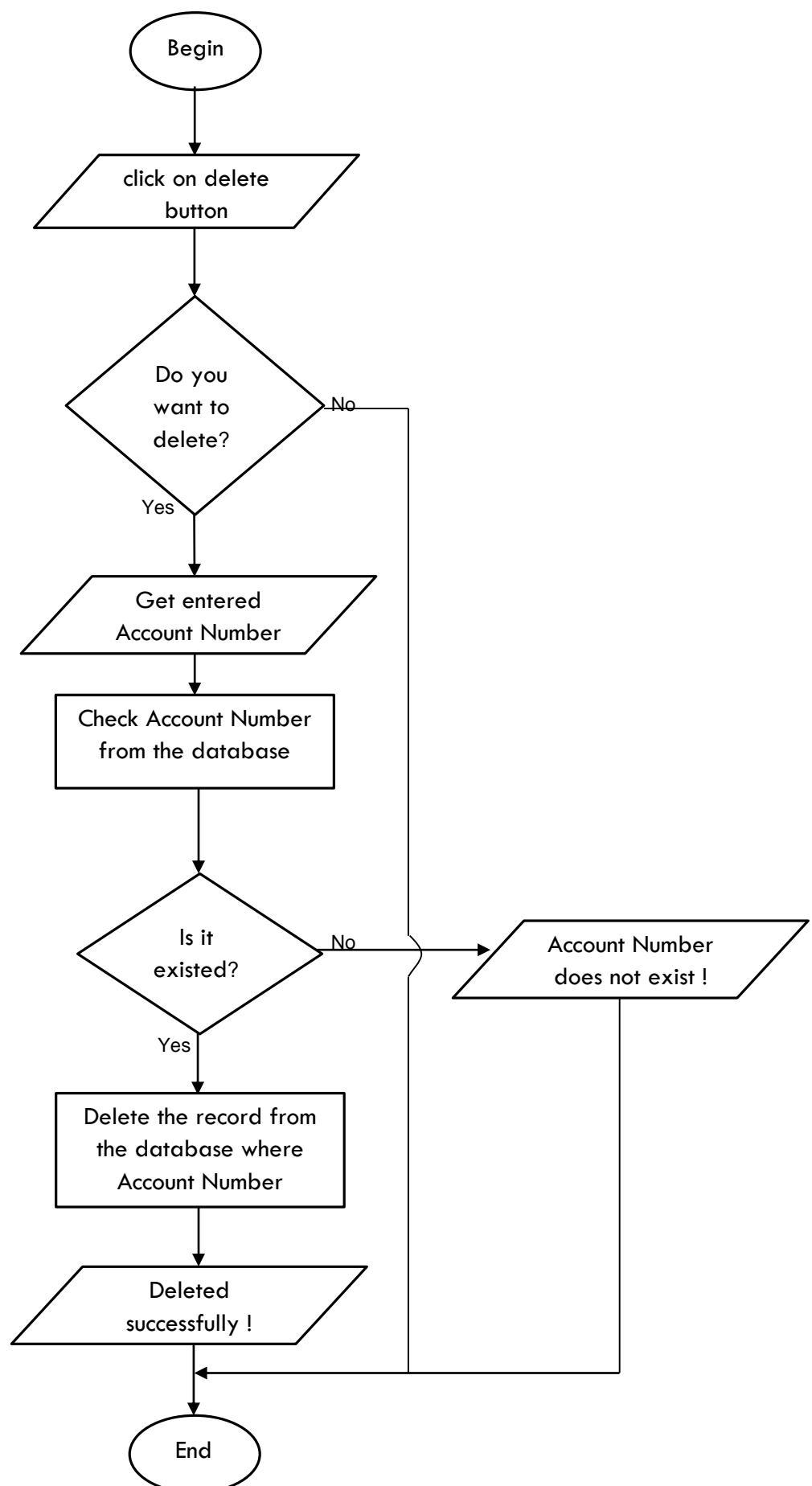
❖ Edit Button



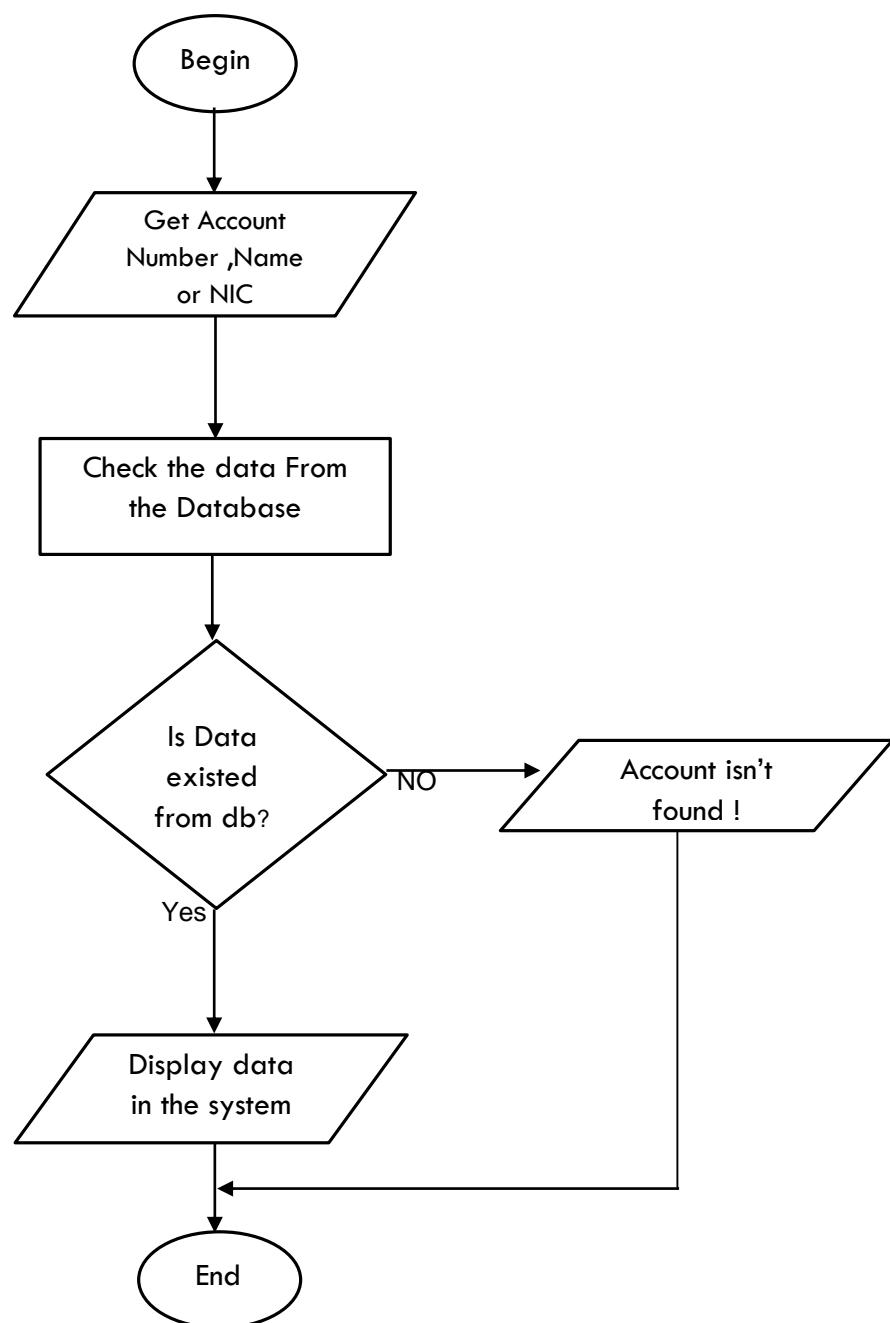
❖ Save Button



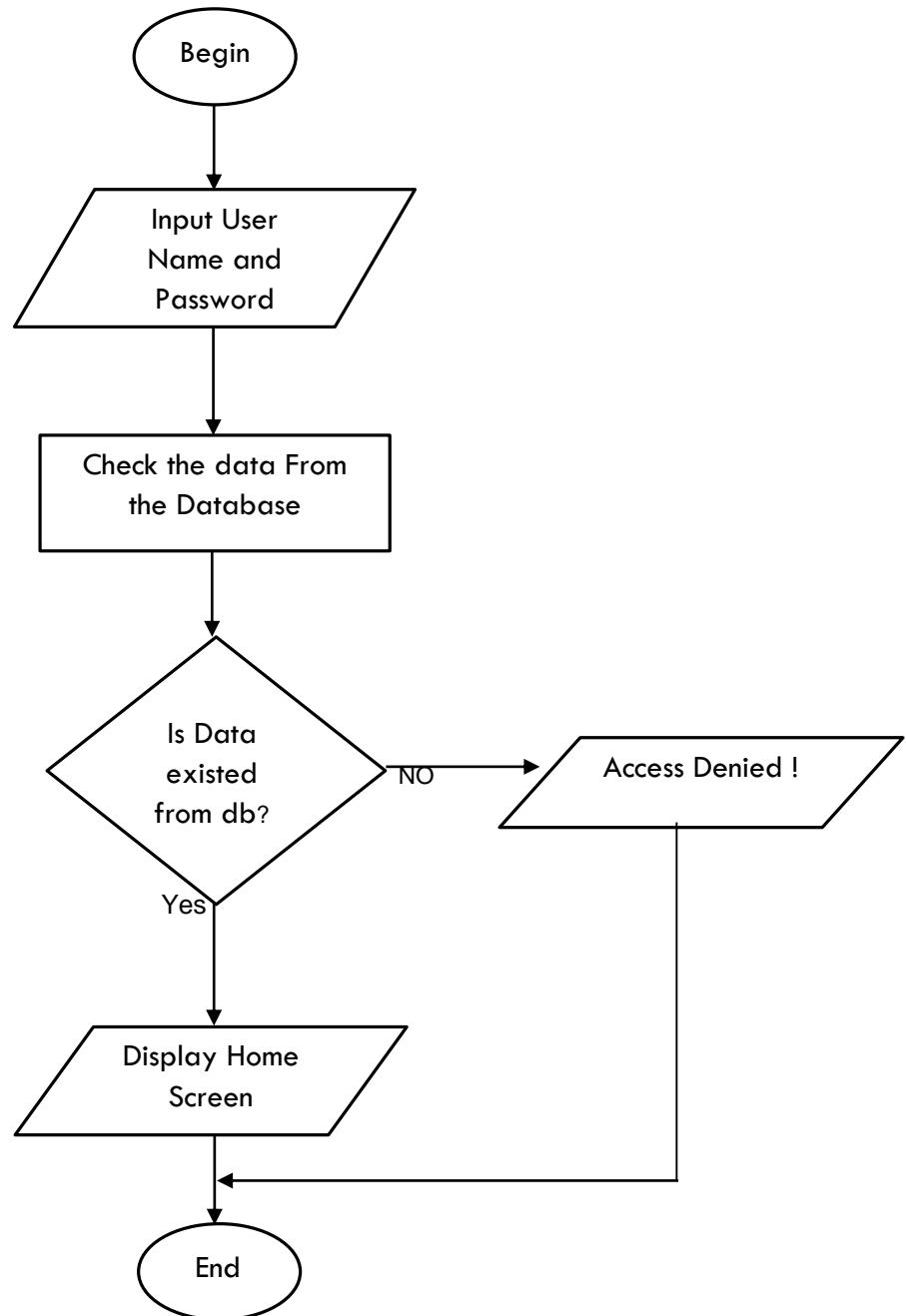
❖ Delete Button



❖ Search



❖ Login



Resource Requirements.

❖ Requirement of the system.

➤ Functional requirement.

- Store and received details.
- Report.
- Automatic report generation.

➤ Non Functional requirement.

- Simplicity and user friendliness.
- Information security and access control.
- Accuracy.
- Efficiency.
- Existence.
- Satisfaction.

➤ Requirement Gathering Techniques.

- One-on-One Interview.
- Group Interviews.
- Questionnaires.

➤ Software requirement.

- Java NetBeans.
- FreeMySQLHosing Service (Online Database)
- Adobe Photoshop or design software.

➤ Hardware Requirements.

- Computer or Laptop.
- Intel core i3 CPU 1.5 GHz processor.
- 2 GB or higher RAM.
- Windows 7 or higher version.

The Software Development Methodology.

The software process is a set activities associated with production of a software product. For the effective control of the software process. It is essential to have a phased development strategy that is, it consists of number of stages that produce one or more documents and program codes.

The software development process model is also called SDLC – Software Development Life Cycle, as the process is repeated when the software is needed to be changed while the software is in use.

There are different software process models used in industry and the choice of the model should be based on the product and project needs.

Some software process models are:

Traditional Development Process Models

- Waterfall Model
- Evolutionary Development Model

Agile Development Process Models

- Scrum

We used one of the traditional development process model, **Waterfall Model** to develop our software.

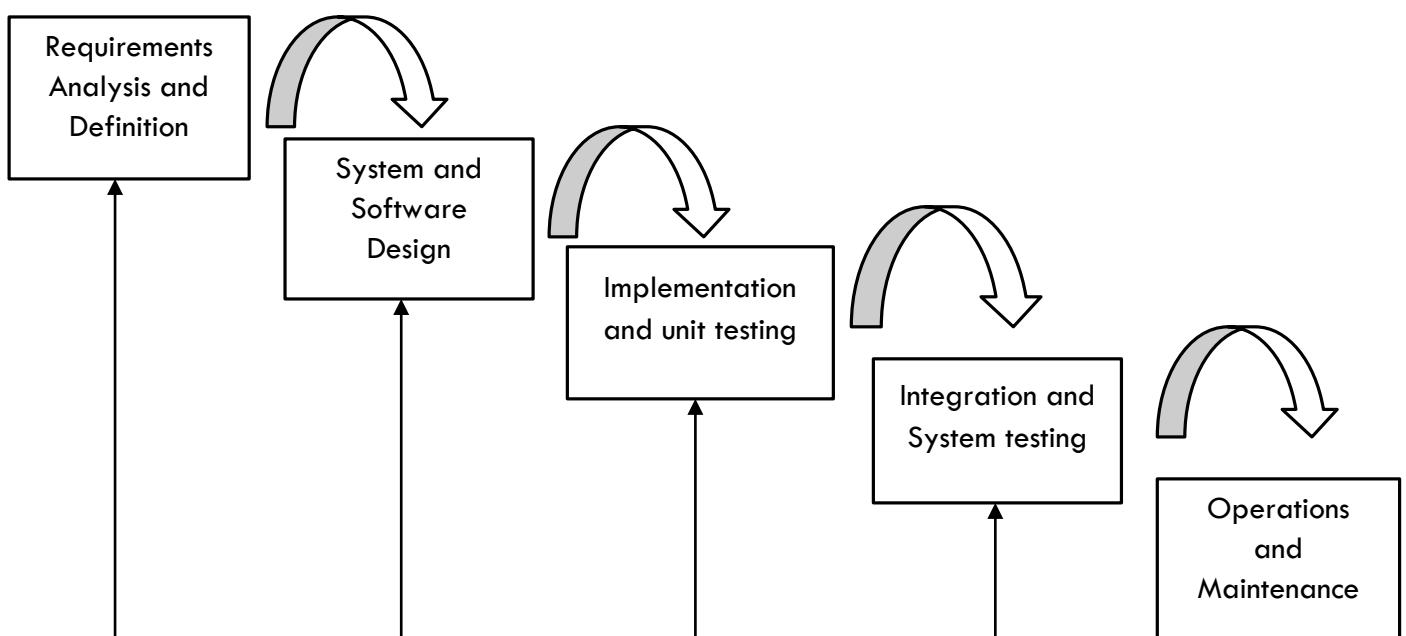
Waterfall Model

This first explicit model for software development process was derived from other engineering process. It offered a means of the development process more visible. Because of the cascade from one phase to another, this model is known as the "Waterfall Model".

The waterfall model is a sequential design process, used in software development process in which progress is seen as flowing steadily downward like a waterfall, through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production or Implementation and Maintenance.

The waterfall model is originating in the manufacturing and construction industries, highly structured physical environment in which after the fact changes are prohibitively costly, if not impossible. Because no formal software development methodologies existed all the time, this hardware oriented model was simply adapted for software development.

Principal stages of the waterfall model (Diagram)



Principal stages of the waterfall model

1. Requirement Analysis and Definition

The system services (functions), limitation and goals are established by consultation with system users. They are then defined in a manner which is understandable by both users and development team.

2. System and Software Design

The system design process partitions the requirements to either hardware or software system. It establishes overall system architecture. Software design involves representing the software system function in a form that may be transformed into one or more executable programs. Software designing can be done by using flow charts or pseudo code.

3. Implementation and Unit Testing

During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

4. Integration and System Testing

Individual program units or programs are integrated and tested as a complete system to ensure that software requirements have been met. After testing, Software system is delivered to be consumer.

5. Operation and Maintenance

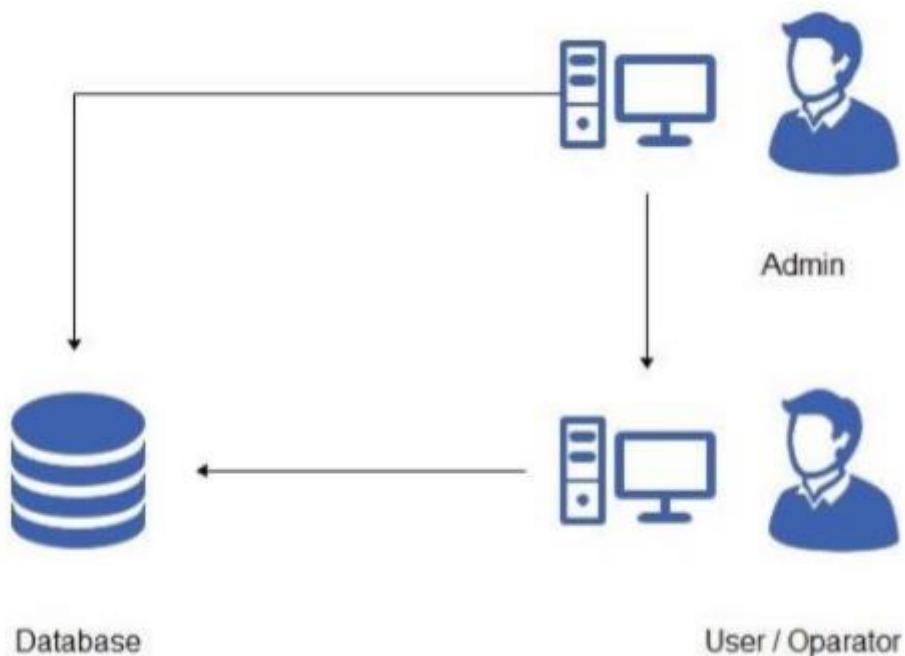
This is the longest life cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which are not discovered in earlier stages of the life cycle, improving the implementation of the system units and adding the system requirements as new requirements are discovered. Activities like staff training, file conversion, site preparing, preparing user manuals and possibly some parallel running also involve in the stage.

Table of the project

Stages	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Preparing Proposal								
Requirement Gathering								
Architecture Design								
Interface/Database Design and Coding								
Beta Release								
Bug Fixing / Final Release								

System Design.

An overall view of the system is presented together with illustrations as how each module interacts with each other. It is here that the written specifications are translated into relevant diagrams to represent the system. This system used ER diagram for system design



Database design

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The general objective is to make information access, easy quick, inexpensive and flexible for other users. During database design the following objectives are concerned,

- ✓ Data independence
- ✓ Accurate and integrating
- ✓ Privacy and security
- ✓ Performance
- ✓ Controlled Redundancy
- ✓ More information at low cost
- ✓ Recovery from failure
- ✓ Ease of learning and use

Interface Designs

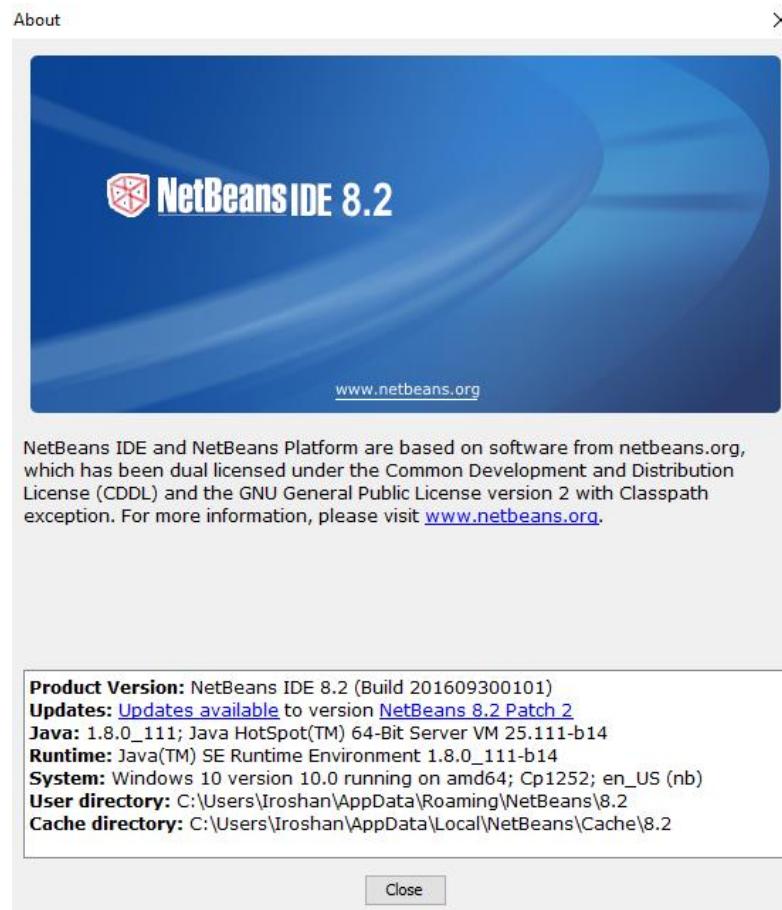
This system contains various Forms, Buttons, Textboxes, Labels, and Tables & Combo boxes.

There are three types of interface design available. But I use menu based interface (GUI). Because it has more advantages than other interfaces. Such as,

- ✓ User guidance and on-line help
- ✓ Speed of use
- ✓ Support for multiple skill levels
- ✓ Users easy to achieve the functions by selecting icon
- ✓ Users does not need that much of knowledge to access the system
- ✓ Users learning time is relatively short
- ✓ Easy to maintain
- ✓ Saving time wastage (command line)

❖ Java Netbeans.

I used java language and NetBeans for creating our software easy and good interface.



Softwares Explain

I Created two Softwares for the final project. First one is in the eBank and other one is for clients who registered in the eBank.

❖ eBank Software

This software can control bank Accounts, Client Accounts, Money Deposit, Money Transactions, Employees and all others.

❖ eClient Software

This software is for the clients. They can control their accounts with computers. They can transfer money, see account details, transaction details and others. It will be easy for manage their bank accounts.

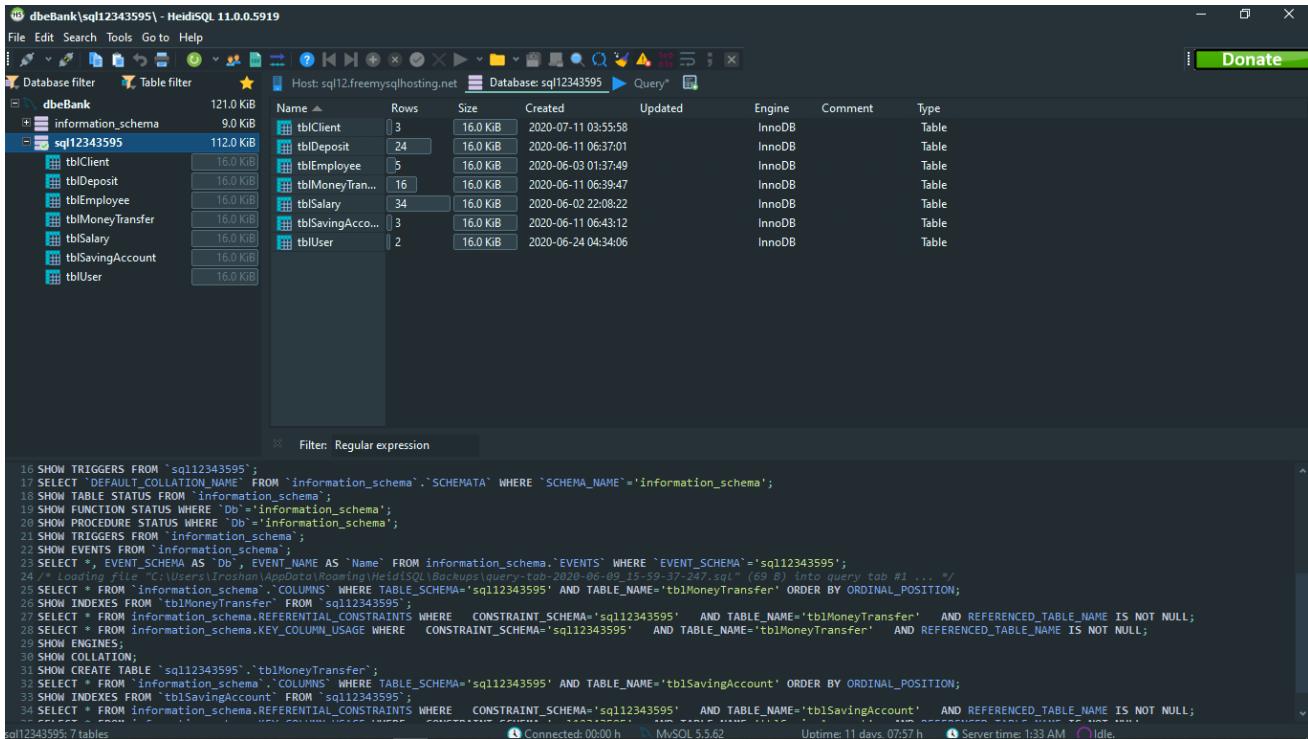
LOGO of the Bank



Software Interfaces & Codes

eBank Software

➤ Database Connection



Name	Rows	Size	Created	Updated	Engine	Comment	Type
tblClient	3	16.0 kB	2020-07-11 03:55:58		InnoDB		Table
tblDeposit	24	16.0 kB	2020-06-11 06:37:01		InnoDB		Table
tblEmployee	5	16.0 kB	2020-06-03 01:37:49		InnoDB		Table
tblMoneyTrans...	16	16.0 kB	2020-06-11 06:39:47		InnoDB		Table
tblSalary	34	16.0 kB	2020-06-02 22:08:22		InnoDB		Table
tblSavingAcco...	3	16.0 kB	2020-06-11 06:43:12		InnoDB		Table
tblUser	2	16.0 kB	2020-06-24 04:34:06		InnoDB		Table

//Database Connection

```
public class DBconnect {  
    public static Connection connect(){  
        Connection conn = null;  
        try {  
            Class.forName("com.mysql.jdbc.Driver");  
            conn=(Connection)DriverManager.getConnection("jdbc:mysql://sql12.freemy  
sqlhosting.net3306/sql12 343595" , "sql12343595","JJphbhNgQv");  
        } catch (Exception e) {  
    }  
}
```

```
        JOptionPane.showMessageDialog(null, e);

    }

    return conn;

}
```

➤ Login



//Login

```
public class Login extends javax.swing.JFrame {
    Connection conn;
    PreparedStatement pst;
    ResultSet rs;

    public Login() {
        initComponents();

        conn = DBconnect.connect();
    }

    public void access(){
        try {
            String sql = "SELECT * FROM tblUser WHERE User_Name=? && Password=?";
            pst = conn.prepareStatement(sql);
            pst.setString (1, tbUserName.getText());
            pst.setString (2, tbPass.getText());
        }
    }
}
```

```

        rs =pst.executeQuery();

        if (tbUserName.getText().equals("") || tbPass.getText().equals("")){
            JOptionPane.showMessageDialog(null, "Please Enter Your Details!");
        }
        else if (rs.next()) {

            Home hm = new Home();
            hm.setVisible(true);
            this.dispose();
        }
        else{
            JOptionPane.showMessageDialog(null, "Access Denied!");
            tbUserName.setText("");
            tbPass.setText("");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    access();
}
private void jButton1KeyPressed(java.awt.event.KeyEvent evt) {
    access();
}

```

➤ Home



//Home

```
public class Home extends javax.swing.JFrame {  
    private Object tbUserName;  
    Connection conn;  
    PreparedStatement pst;  
    PreparedStatement pst2;  
    ResultSet rs;  
  
    String status =null;  
    public Home() {  
        initComponents();  
        conn = DBconnect.connect();  
        lblUser.setText(Login.tbUserName.getText());  
        Login log = new Login();  
        log.setIcon();
```

```

        setExtendedState(JFrame.MAXIMIZED_BOTH);

        date();
        time();
    }

    //Date

    public void date(){
        Date d = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        lblDate.setText(sdf.format(d));
    }

    //Time

    public void time(){
        new Timer(0, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                Date d = new Date();
                SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss");
                lblTime.setText(sdf.format(d));
            }
        });
    }

    private void jLabel15MouseClicked(java.awt.event.MouseEvent evt) {
        AccountType account = new AccountType();
        account.setVisible(true);
        this.dispose();
    }
}

```

```

private void jLabel6MouseClicked(java.awt.event.MouseEvent evt) {
    Home hm = new Home();
    hm.setVisible(true);
    this.dispose();
}

private void lblHelpMouseClicked(java.awt.event.MouseEvent evt) {
    Help hp =new Help();
    hp.setVisible(true);
}

private void lblBankAccountsMouseClicked(java.awt.event.MouseEvent evt) {
    AccountTypeDetails ATD = new AccountTypeDetails();
    ATD.setVisible(true);
    this.dispose();
}

private void jLabel11MouseClicked(java.awt.event.MouseEvent evt) {
    newClientAccount NCA = new newClientAccount(); //NCA means object of
    new Client Account
    NCA.setVisible(true);
    this.dispose();
}

private void jLabel12MouseClicked(java.awt.event.MouseEvent evt) {
    ClientAccountSearch CAS = new ClientAccountSearch();
    CAS.setVisible(true);
    this.dispose();
}

private void jLabel18MouseClicked(java.awt.event.MouseEvent evt) {
    BankAccountSearch BAS = new BankAccountSearch();
    BAS.setVisible(true);
    this.dispose();
}

```

```

}

private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {
    Deposit DP = new Deposit();
    DP.setVisible(true);
    this.dispose();
}

private void jLabel21MouseClicked(java.awt.event.MouseEvent evt) {
    MoneyTransfer MT = new MoneyTransfer();
    MT.setVisible(true);
    this.dispose();
}

private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {
    DepositDetails DD = new DepositDetails();
    DD.setVisible(true);
    this.dispose();
}

private void jLabel24MouseClicked(java.awt.event.MouseEvent evt) {
    Transfactiondetails TD = new Transfactiondetails();
    TD.setVisible(true);
    this.dispose();
}

private void lblReportMouseClicked(java.awt.event.MouseEvent evt) {
    Reports rp = new Reports();
    rp.setVisible(true);
    this.dispose();
}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    AboutBank AB = new AboutBank();
    AB.setVisible(true);
}

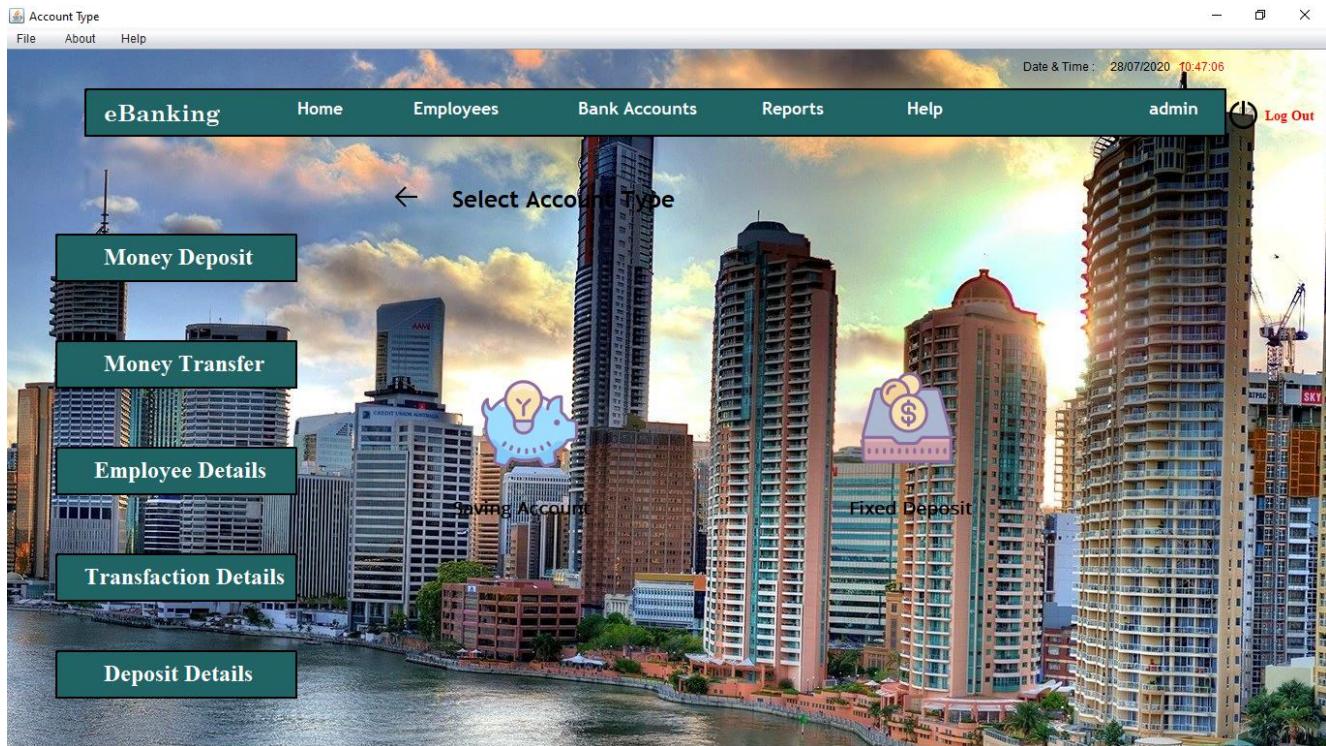
```

```
}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {

    Help hp = new Help();
    hp.setVisible(true);
}
```

➤ Account Type

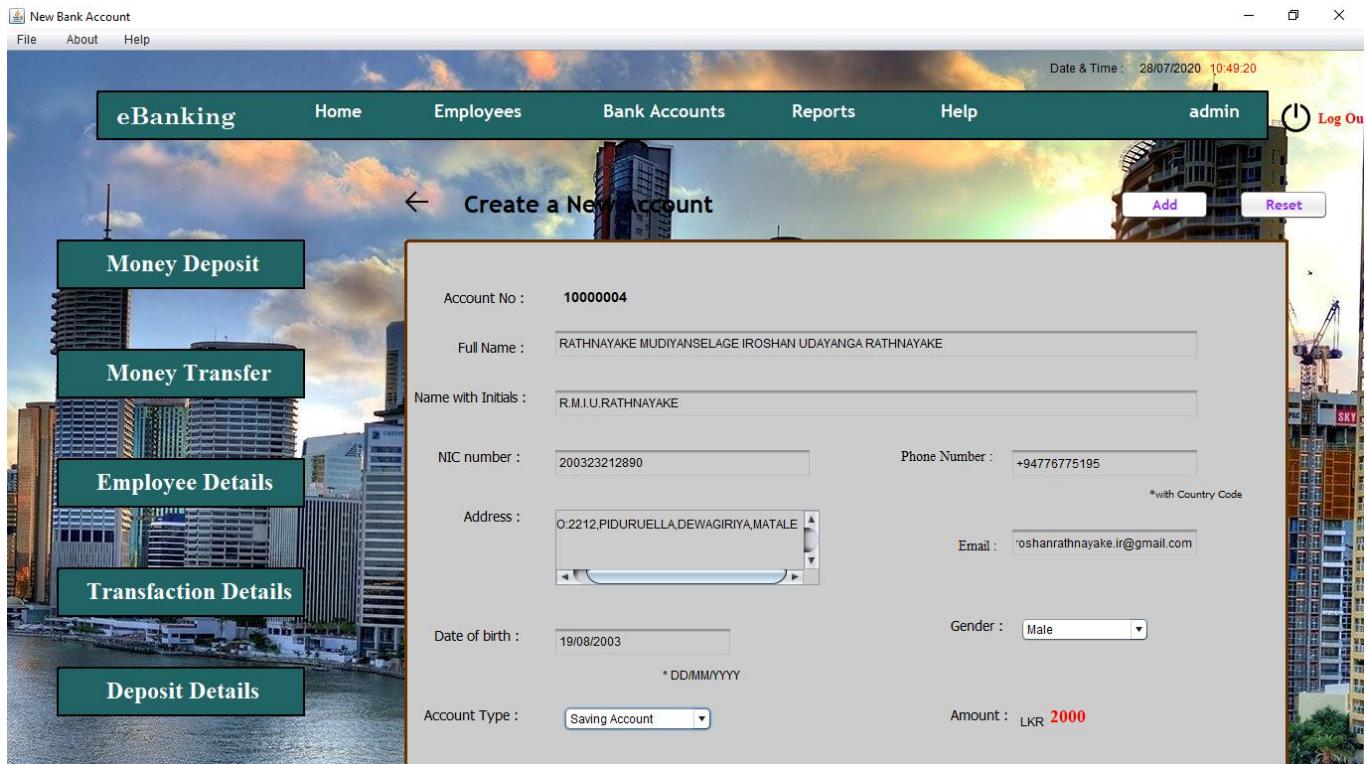


```
//Select Account Type
public AccountType() {
    initComponents();
    lblUser.setText(Home.lblUser.getText());
    Login log = new Login();
    log.setIcon();
    setExtendedState(JFrame.MAXIMIZED_BOTH);
}

//Saving Account
private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {
    NewAccount NA = new NewAccount();
    NA.setVisible(true);
    this.dispose();
}

//Fixed Deposit Account
private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
    JOptionPane.showMessageDialog(null, "This has not been implemented yet. It'll be coming soon!");
}
```

➤ New Bank Accounts (Saving Account)



```
//New Bank Account
public class NewAccount extends javax.swing.JFrame {
```

```
    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
```

```
public NewAccount() {
    initComponents();
    Login log = new Login();
    log.setIcon();
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    lblUser.setText(Home.lblUser.getText());
    conn = DBconnect.connect();
    accountNo();
    date();
    time();
```

```
}
```

```
//Auto detect Account Number
public void accountNo(){
```

```
try {
```

```

        String sql = "SELECT * FROM tblSavingAccount ORDER BY
Account_No DESC";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if(rs.next()){

            int id = rs.getInt("Account_No");
            int relID = id +1;

            lblID.setText(Integer.toString(relID));
        }

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

public void reset(){

    tbFName.setText("");
    tbNameWithInitials.setText("");
    tbAddress.setText("");
    tbFName.setText("");
    tbNameWithInitials.setText("");
    tbNIC.setText("");
    tbPhone.setText("");
    tbAddress.setText("");
    tbEmail1.setText("");
    tbBirth.setText("");
    cbGender.setSelectedItem(0);
    cbaccountType.setSelectedItem(0);

}

//Add Button
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {

    String fname = tbFName.getText();
    String uname = tbNameWithInitials.getText();
    String nic = tbNIC.getText();
    String phone = tbPhone.getText();
    String address = tbAddress.getText();
    String email = tbEmail1.getText();
}

```

```

String bd = tbBirth.getText();
String gen = cbGender.getSelectedItem().toString();
String acctype = cbaccountType.getSelectedItem().toString();
float amount = Float.parseFloat(lblAmount.getText());

if(fname.equals("") || uname.equals("") || nic.equals("") || phone.equals("") || address.equals("") || bd.equals("") || gen.equals(0) || acctype.equals(0) ){

    JOptionPane.showMessageDialog(null, "Please fill all the details!");
}

else{

    try {
        String sql      =      "INSERT      INTO      tblSavingAccount
        (User_Name,FullName,NIC,Address,BD,Gender,Account_Type,Email,Phone,A
        mount)VALUES("+uname+","+fname+","+nic+","+address+","+bd+","+gen
        +",""+acctype+","+email+","+phone+","+amount+");

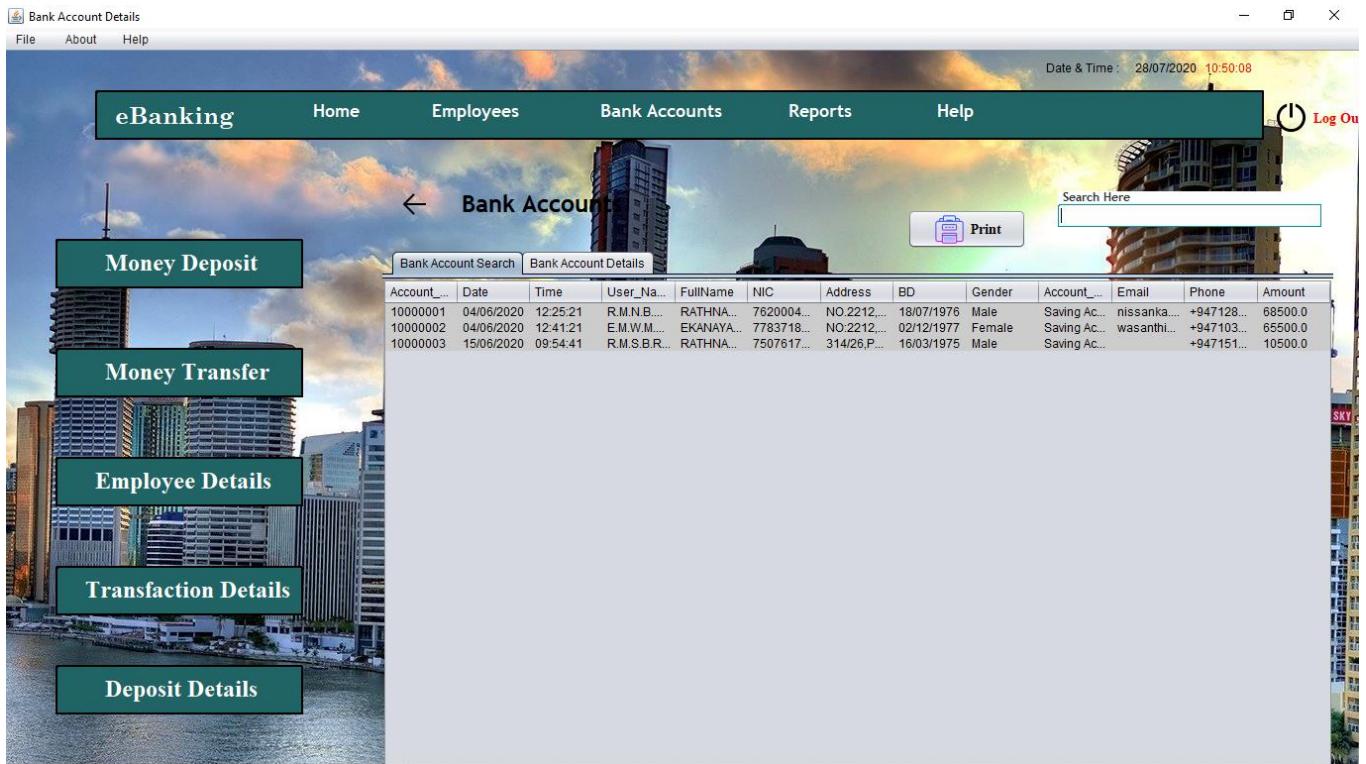
        pst =conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Account Added Successfully!");
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
reset();
}

//Reset Button
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    reset();
}

}

```

➤ Bank Account Search



```
public final class BankAccountSearch extends javax.swing.JFrame {
```

```
    Connection conn;  
    PreparedStatement pst;  
    PreparedStatement pst2;  
    ResultSet rs;
```

```
    public BankAccountSearch() {
```

```
        Login log = new Login();  
        log.setIcon();  
        initComponents();  
        setExtendedState(JFrame.MAXIMIZED_BOTH);  
        lblUser.setText(Home.lblUser.getText());  
        conn = DBconnect.connect();  
        tableLoad();  
        editBoxdisable();  
        date();  
        time();
```

```
}
```

```

//Table Load

public void tableLoad() {

    try {

        String sql = "SELECT
Account_No,User_Name,NIC,Address,BD,Gender,Account_Type,Email,Phone,
Amount FROM tblSavingAccount";

        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        tblAccount.setModel(DbUtils.resultSetToTableModel(rs));

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

//Search Accounts

public void search(String table , String srch) {

    try {

        String sql = "SELECT * FROM "+table+" WHERE User_Name LIKE '%" + srch
+ "%' || Account_No LIke '%" + srch + "%' || NIC LIKE '%" + srch + "%'";

        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        tblAccount.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);
    }
}

private void tbSearchKeyReleased(java.awt.event.KeyEvent evt) {
    search("tblSavingAccount", tbSearch.getText());
}

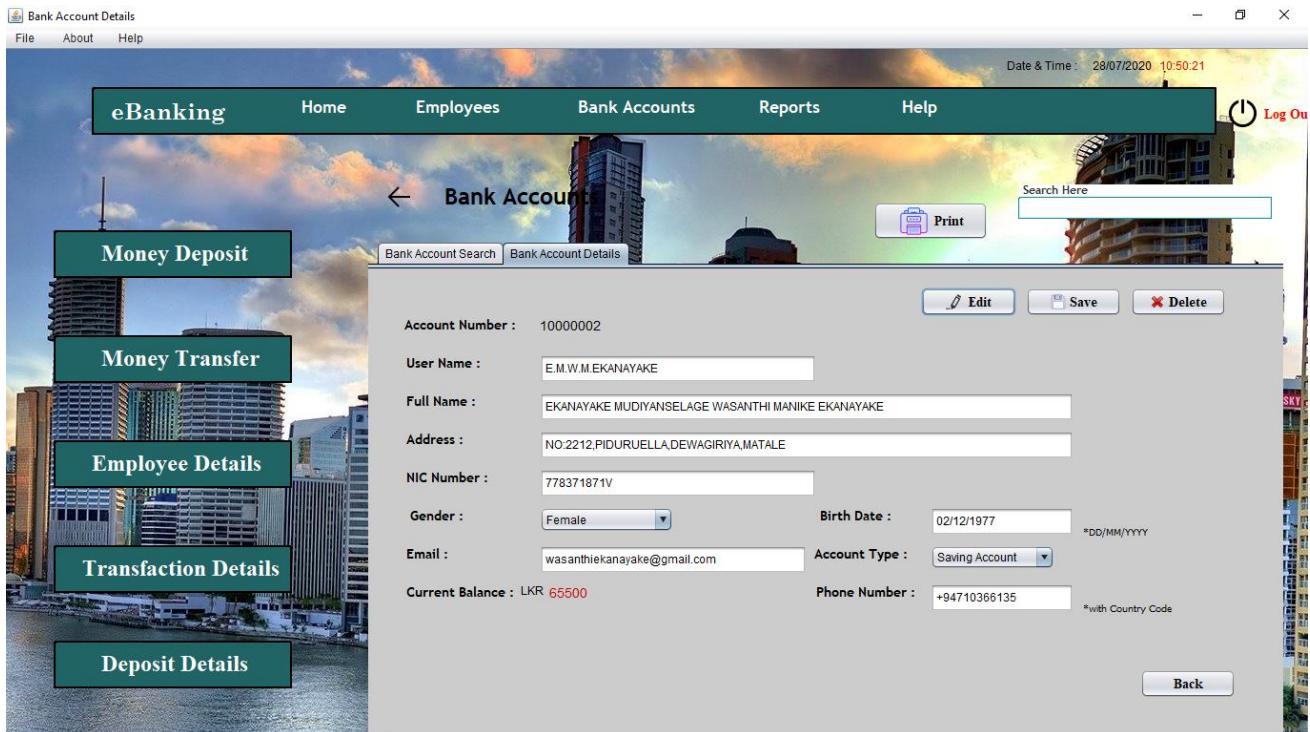
private void tblAccountMouseClicked(java.awt.event.MouseEvent evt) {

    tableData();
    JTpane.setSelectedIndex(1);

}

```

➤ Bank Account Details



//Table Data to Boxes

```
public void tableData(){
    int r = tblAccount.getSelectedRow();
    String AccNo = tblAccount.getValueAt(r, 0).toString();
    try {
        String sql = "SELECT * FROM tblSavingAccount WHERE Account_No ='" + AccNo + "'";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        if (rs.next()) {
            String id = rs.getString("Account_No");
            lblAccountNo.setText(id);
            String UName = rs.getString("User_Name");
            tbUname.setText(UName);
            String FName = rs.getString("FullName");
            tbFName.setText(FName);
            String nic = rs.getString("NIC");
            tbNIC.setText(nic);
            String Address = rs.getString("Address");
            tbAddress.setText(Address);
        }
    }
}
```

```

        String bd = rs.getString("BD");
        tbBD.setText(bd);
        String gen = rs.getString("Gender");
        cbGender.setSelectedItem(gen);
        String AccType = rs.getString("Account_Type");
        cbAccType.setSelectedItem(AccType);
        String email = rs.getString("Email");
        tbEmail.setText(email);
        String phone = rs.getString("Phone");
        tbPhone.setText(phone);
        String balance = rs.getString("Amount");
        lblBalance.setText(balance);

    }
} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);
}

}

//Make Boxes as Editable

public void editBoxEnable(){
    tbUname.setEditable(true);
    tbFName.setEditable(true);
    tbNIC.setEditable(true);
    tbAddress.setEditable(true);
    tbBD.setEditable(true);
    cbAccType.setEditable(true);
    cbGender.setEditable(true);
    tbEmail.setEditable(true);
    tbPhone.setEditable(true);

}

//Disable Boxes

public void editBoxDisable(){
    tbUname.setEditable(false);
    tbFName.setEditable(false);
    tbNIC.setEditable(false);
    tbAddress.setEditable(false);
    tbBD.setEditable(false);
    cbAccType.setEditable(false);
    cbGender.setEditable(false);
    tbEmail.setEditable(false);
    tbPhone.setEditable(false);
}

```

```

        }
    //Clear data in the boxes

    public void tbClear(){
        lblAccountNo.setText("");
        lblAccountNo.setText("");
        tbUname.setText("");
        tbFName.setText("");
        tbNIC.setText("");
        tbAddress.setText("");
        tbBD.setText("");
        cbAccType.setSelectedIndex(0);
        cbGender.setSelectedIndex(0);
        tbEmail.setText("");
        tbPhone.setText("");
        lblBalance.setText("");
    }

    //Edit Button

    private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
        editBoxEnable();
    }

    //Save Button

    private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
        int id = Integer.parseInt(lblAccountNo.getText());
        String uname = tbUname.getText();
        String fname = tbFName.getText();
        String address = tbAddress.getText();
        String nic = tbNIC.getText();
        String gen = cbGender.getSelectedItem().toString();
        String email = tbEmail.getText();
        String bd = tbBD.getText();
        String acctype = cbAccType.getSelectedItem().toString();
        String phone = tbPhone.getText();

        try {
            String      sql      ="UPDATE      tblSavingAccount      SET
User_Name='"+uname+"',FullName='"+fname+"',NIC='"+nic+"',Address=
"+address+",BD='"+bd+"',Gender='"+gen+"',Account_Type='"+acctype+"'
',Email='"+email+"',Phone='"+phone+"' WHERE Account_No='"+id+"'";

```

```

        String SQL2      ="UPDATE      tblClient      SET
        Client_Name='"+uname+"',FullName='"+fname+"',NIC='"+nic+"',Address
        ="'+address+"',BD='"+bd+"',Gender='"+gen+"',Account_Type='"+acctype
        +"',Email='"+email+"',Phone='"+phone+"' WHERE Account_No='"+id+"';

        pst = conn.prepareStatement(sql);
        pst2 = conn.prepareStatement(SQL2);
        pst.execute();
        pst2.execute();
        JOptionPane.showMessageDialog(null, "Updated Successfully!");

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
    tableLoad();
    tbClear();
    editBoxdisable();

}

//Delete Button

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {

    String id = lblAccountNo.getText();
    int check = JOptionPane.showConfirmDialog(null, "Do you want to delete?");

    if(check == 0){
        try {

            String sql   =   "DELETE   FROM   tblSavingAccount   WHERE
            Account_No='"+id+"';

            String SQL2 = "DELETE FROM tblClient WHERE Account_No='"+id+"';

            pst = conn.prepareStatement(sql);
            pst2 = conn.prepareStatement(SQL2);
            pst.execute();
            pst2.execute();
            JOptionPane.showMessageDialog(null, "Account Deleted!");

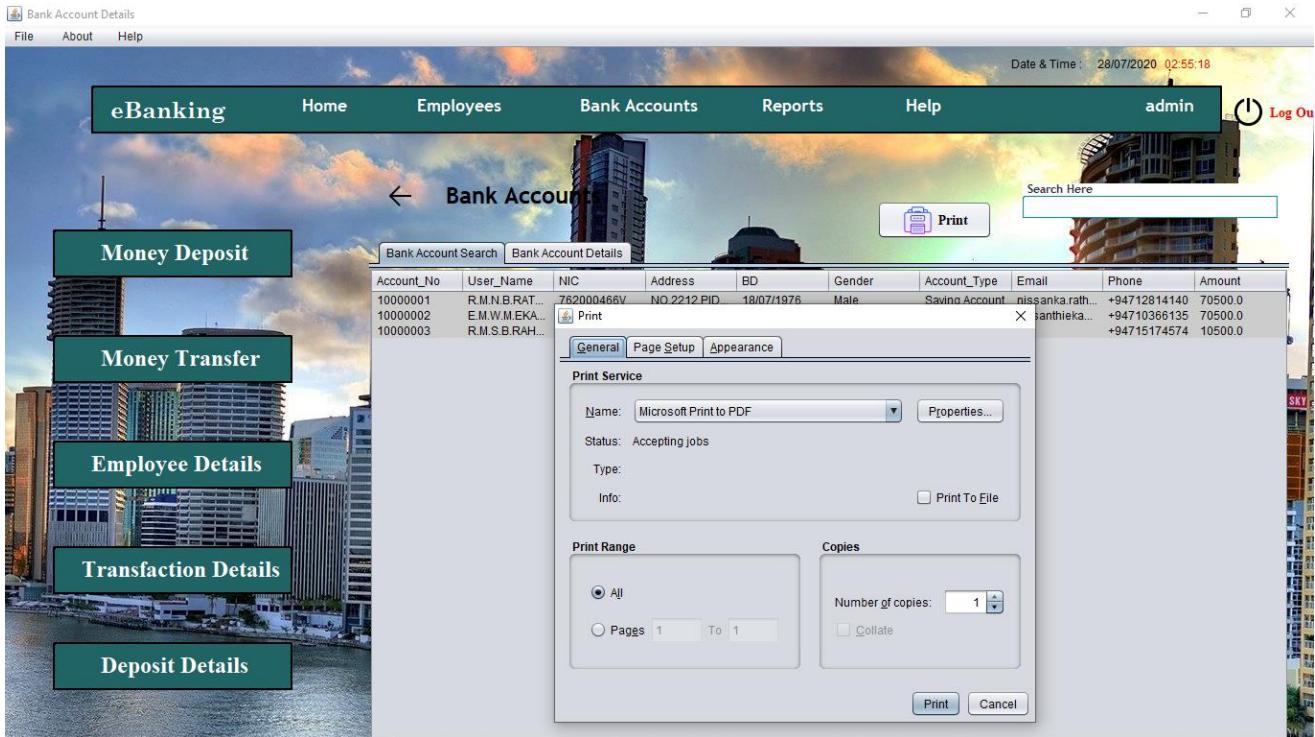
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
        tableLoad();
    }
}

```

```
//Back Button
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    tbClear();
    editBoxDisable();
    tableLoad();

}
```

➤ Bank Account Details Print

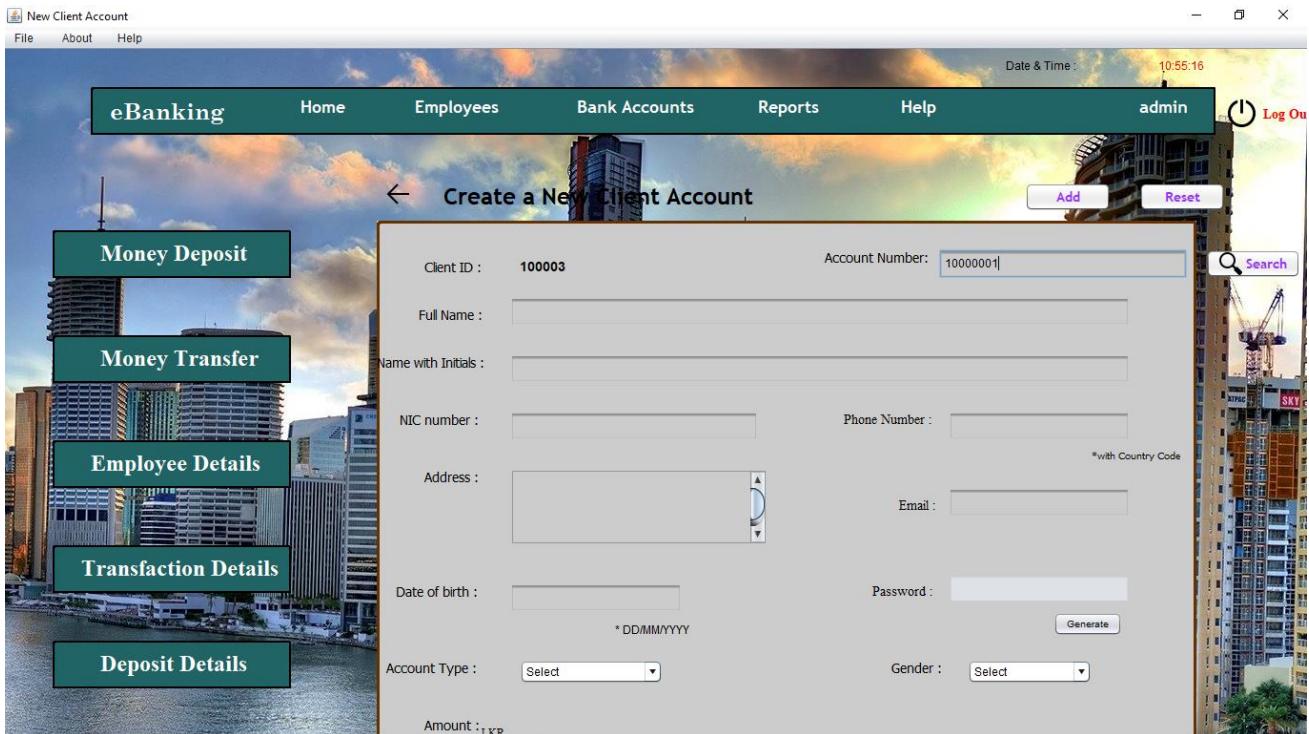


//Bank Account Details Print

```
public void print() {
    MessageFormat header = new MessageFormat("Bank Accounts Report");
    MessageFormat footer = new MessageFormat("-eBankONLINE-");

    try {
        tblAccount.print(JTable.PrintMode.NORMAL, header, footer);
    } catch (java.awt.print.PrinterException e) {
        System.err.format("Cannot Print", e.getMessage());
    }
}
//Print Button
```

➤ New Client Account



```
public class newClientAccount extends javax.swing.JFrame {
```

```
    Connection conn;  
    PreparedStatement pst;  
    PreparedStatement pst2;  
    ResultSet rs;  
    ResultSet rs2;
```

```
    public newClientAccount() {
```

```
        initComponents();  
        Login log = new Login();  
        log.setIcon();  
        setExtendedState(JFrame.MAXIMIZED_BOTH);  
        lblUser.setText(Home.lblUser.getText());  
        tbPassword.enable(false);  
        conn = DBconnect.connect();  
        clientID();  
        date();  
        time();
```

```
    }
```

```

//Automatically Detect Client ID

public void clientID(){

    try {
        String sql = "SELECT * FROM tblClient ORDER BY Client_ID DESC";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if (rs.next()){
            int id = rs.getInt("Client_ID");
            int relID = id + 1;
            lblClientID.setText(Integer.toString(relID));
        }

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

//Search Client Details using Account No

public void search(){

    int id = Integer.parseInt(tbAccNo.getText());

    try {

        String SQL2 = "SELECT * FROM tblClient WHERE Account_No = '"+id+"'";
        String sql = "SELECT * FROM tblSavingAccount WHERE Account_No='"+id+"'";

        pst = conn.prepareStatement(sql);
        pst2 = conn.prepareStatement(SQL2);
        rs = pst.executeQuery();
        rs2 = pst2.executeQuery();

        if(rs.next()){

            JOptionPane.showMessageDialog(null, "Account Already Exist!");
            tbAccNo.setText("");

        }

    } else{

```

```

        if(rs2.next()){
            String uname = rs.getString("User_Name");
            tbNameWithInitials.setText(uname);
            String fname = rs.getString("FullName");
            tbFName.setText(fname);
            String nic = rs.getString("NIC");
            tbNIC.setText(nic);
            String address = rs.getString("Address");
            tbAddress.setText(address);
            String bd = rs.getString("BD");
            tbBirth.setText(bd);
            String gen = rs.getString("Gender");
            cbGender.setSelectedItem(gen);
            String acctype = rs.getString("Account_Type");
            cbAccType.setSelectedItem(acctype);
            String email = rs.getString("Email");
            tbEmail.setText(email);
            String phone = rs.getString("Phone");
            tbPhone.setText(phone);
            float amount = rs.getFloat("Amount");
            lblAmount.setText(Float.toString(amount));
        }
        else {
            JOptionPane.showMessageDialog(null, "Create New Bank Account!");
            tbAccNo.setText("");
        }
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}

public void reset(){
    tbAccNo.setText("");
    tbFName.setText("");
    tbNameWithInitials.setText("");
    tbAccNo.setText("");
    tbPhone.setText("");
    tbAddress.setText("");
    tbBirth.setText("");
    tbPassword.setText("");
    tbEmail.setText("");
    cbAccType.setSelectedItem(0);
    cbGender.setSelectedItem(0);
    lblAmount.setText("");
}
}

```

```

//Search Button
private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    search();
}

//Password Generate Button

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    int length = 6;
    StringBuilder str = new StringBuilder();
    for (int i=0 ; i<length; i++){
        int num = randomNum();
        char ch = GenPass.charAt(num);
        str.append(ch);

    }
    tbPassword.setText(str.toString());
}

//Add Button

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {

    int accno = Integer.parseInt(tbAccNo.getText());
    String fname = tbFName.getText();
    String uname = tbNameWithInitials.getText();
    String nic = tbAccNo.getText();
    String phone = tbPhone.getText();
    String address = tbAddress.getText();
    String bd = tbBirth.getText();
    String pass = tbPassword.getText();
    String email= tbEmail.getText();
    String acctype = cbAccType.getSelectedItem().toString();
    String gen = cbGender.getSelectedItem().toString();
    float amount = Float.parseFloat(lblAmount.getText());

    if(tbAccNo.equals("")){
        JOptionPane.showMessageDialog(null, "Please fill all the details!");
    }
    else{
        if(pass.equals("")){
            JOptionPane.showMessageDialog(null, "Please generate the password!");
        }
    }
}

```

```

        }
    else{
        try {

            String      SQL2      =      "INSERT      INTO      tblClient
(Account_No,Client_Name,FullName,NIC,Address,BD,Gender,Passwor
d,Email,Phone,Amount)VALUES("+accno+",""+uname+",""+fname+",
"+nic+",""+address+",""+bd+",""+gen+",""+pass+",""+email+",
"+phone+",
"+amount+");

            pst2 = conn.prepareStatement(SQL2);

            pst2.execute();

            JOptionPane.showMessageDialog(null, "Added Successfully!");

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

        }
    }

}

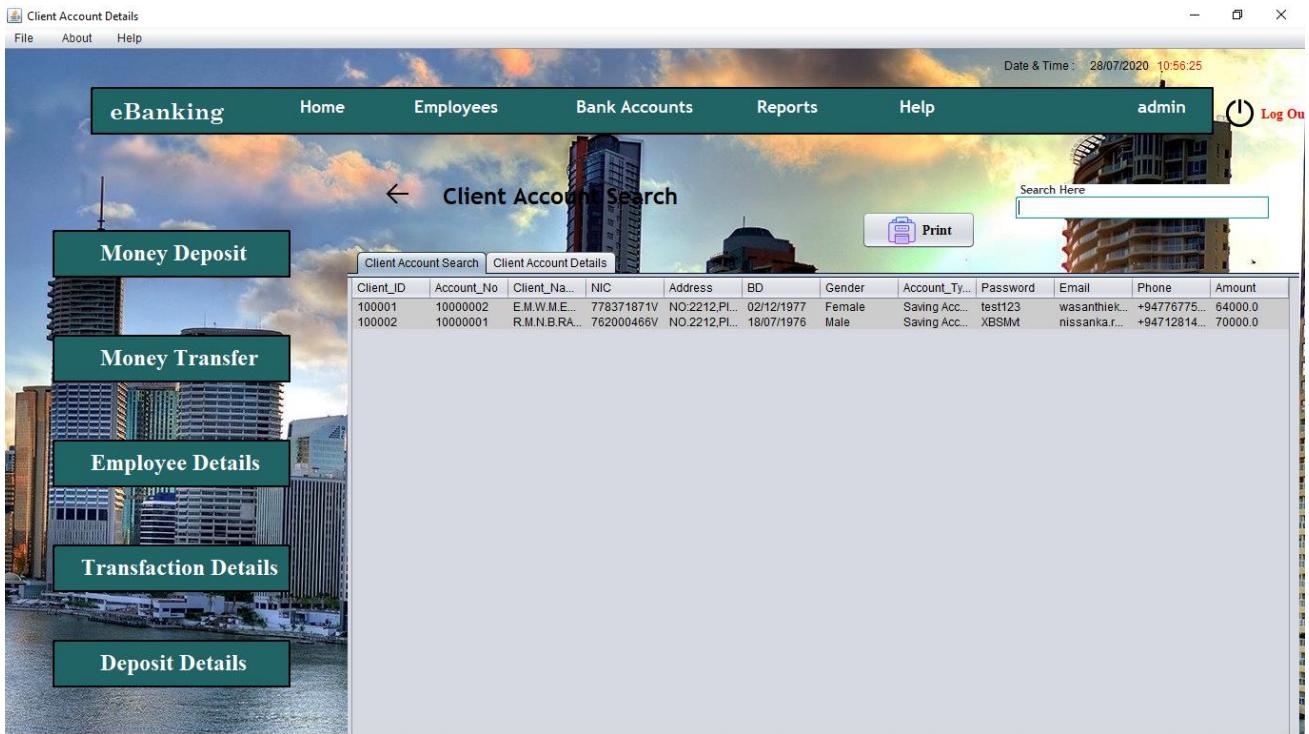
reset();

//Reset Button
private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    clientID();
    reset();

}

```

➤ Client Account Search



//Client Account Search

```
public class ClientAccountSearch extends javax.swing.JFrame {
    Connection conn = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public ClientAccountSearch() {
        initComponents();
        Login log = new Login();
        log.setIcon();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        lblUser.setText(Home.lblUser.getText());
        conn = DBconnect.connect();
        tableLoad();
        editBoxdisable();
        date();
        time();
    }
}
```

//Table Load

```
public void tableLoad(){
```

```
    try {
```

```

String sql = "SELECT
Client_ID,Account_No,Client_Name,NIC,Address,BD,Gender,Account_
Type,Password,Email,Phone,Amount FROM tblClient";

pst = conn.prepareStatement(sql);
rs = pst.executeQuery();
tblClient.setModel(DbUtils.resultSetToTableModel(rs));

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

//Search

public void search(){

String srch = tbSearch.getText();

try {

String sql = "SELECT * FROM tblClient WHERE Client_Name LIKE '%" +
srch + "%' || Client_ID Like '%" + srch + "%' || NIC LIKE '%" + srch + "%'";

pst = conn.prepareStatement(sql);
rs = pst.executeQuery();
tblClient.setModel(DbUtils.resultSetToTableModel(rs));

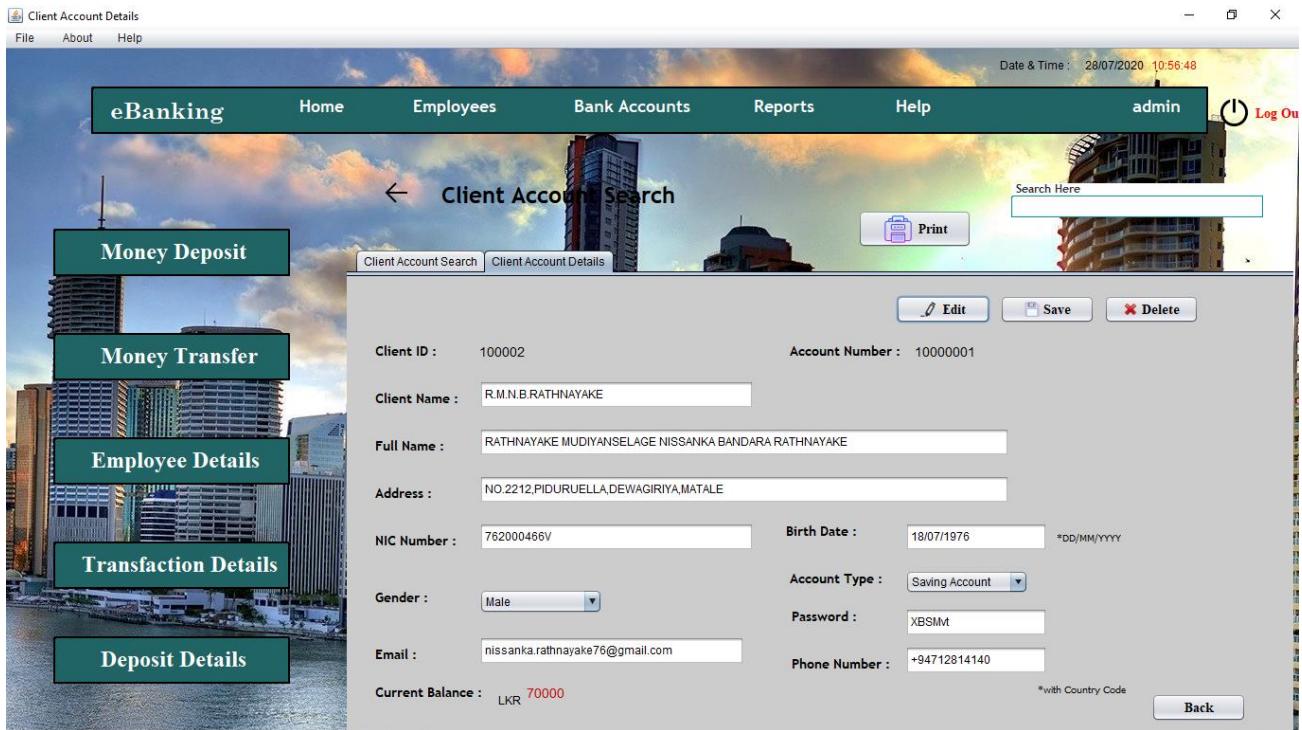
} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);
}
}

private void tbSearchKeyReleased(java.awt.event.KeyEvent evt) {
    search();
}

```

➤ Client Account Details



//Client Account Details

```
public void tableData(){
    int r = tblClient.getSelectedRow();
    String ClientID = tblClient.getValueAt(r, 0).toString();

    try {

        String sql = "SELECT * FROM tblClient WHERE Client_ID ='" + ClientID + "'";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if (rs.next()) {

            String id = rs.getString("Client_ID");
            lblClientID.setText(id);
            String accno = rs.getString("Account_No");
            lblAccountNo.setText(accno);
            String UName = rs.getString("Client_Name");
            tbCname.setText(UName);
            String FName = rs.getString("FullName");
            tbFName.setText(FName);
            String nic = rs.getString("NIC");
            tbNIC.setText(nic);
            String Address = rs.getString("Address");
        }
    }
}
```

```

        tbAddress.setText(Address);
        String bd = rs.getString("BD");
        tbBD.setText(bd);
        String gen = rs.getString("Gender");
        cbGender.setSelectedItem(gen);
        String AccType = rs.getString("Account_Type");
        cbAccType.setSelectedItem(AccType);
        String pass = rs.getString("Password");
        tbPassword.setText(pass);
        String email = rs.getString("Email");
        tbEmail.setText(email);
        String phone = rs.getString("Phone");
        tbPhone.setText(phone);
        String balance = rs.getString("Amount");
        lblBalance.setText(balance);
    }
} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);
}
}

```

//Make text boxes as editable

```

public void editBoxEnable(){
    tbCname.setEditable(true);
    tbFName.setEditable(true);
    tbNIC.setEditable(true);
    tbAddress.setEditable(true);
    tbBD.setEditable(true);
    cbGender.setEditable(true);
    cbAccType.setEditable(true);
    tbEmail.setEditable(true);
    tbPhone.setEditable(true);
}

```

//Disable text boxes

```

public void editBoxdisable(){
    tbCname.setEditable(false);
    tbFName.setEditable(false);
    tbNIC.setEditable(false);
    tbAddress.setEditable(false);
    tbBD.setEditable(false);
    cbGender.setEditable(false);
    cbAccType.setEditable(false);
}

```

```

tbPassword.setEditable(false);
tbEmail.setEditable(false);
tbPhone.setEditable(false);

}

//Clear data in the boxes

public void tbClear(){
    lblClientID.setText("");
    lblAccountNo.setText("");
    tbCname.setText("");
    tbFName.setText("");
    tbNIC.setText("");
    tbAddress.setText("");
    tbBD.setText("");
    cbGender.setSelectedIndex(0);
    cbAccType.setSelectedIndex(0);
    tbPassword.setText("");
    tbEmail.setText("");
    tbPhone.setText("");
    lblBalance.setText("");
}

//Edit Button
private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    editBoxEnable();

}

//Save Button

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    int id = Integer.parseInt(lblClientID.getText());
    String Cname = tbCname.getText();
    String Fname = tbFName.getText();
    String address = tbAddress.getText();
    String nic = tbNIC.getText();
    String bd = tbBD.getText();
    String gen = cbGender.getSelectedItem().toString();
    String acctype = cbAccType.getSelectedItem().toString();
    String pass = tbPassword.getText();
    String email = tbEmail.getText();
    String phone = tbPhone.getText();

    try {

```

```

        String      sql      =      "UPDATE      tblClient      SET
        Client_Name='"+Cname+"',FullName='"+Fname+"',NIC='"+nic+"',Addres
        s='"+address+"',BD='"+bd+"',Gender='"+gen+"',Account_Type='"+acctyp
        e+"',Password='"+pass+"',Email='"+email+"',Phone='"+phone+"' WHERE
        Client_ID = '"+id+"'";

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

        JOptionPane.showMessageDialog(null, "Account Updated
        Successfully!");

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
    tableLoad();
    tbClear();
    editBoxdisable();

}

//Delete Button

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    String id = lblClientID.getText();
    int check = JOptionPane.showConfirmDialog(null, "Do you want to
Delete?");

    if(check==0){
        try {

            String sql="DELETE FROM tblClient WHERE Client_ID='"+id+"'";
            pst = conn.prepareStatement(sql);
            pst.execute();
            JOptionPane.showMessageDialog(null, "Account Deleted Successfully!");

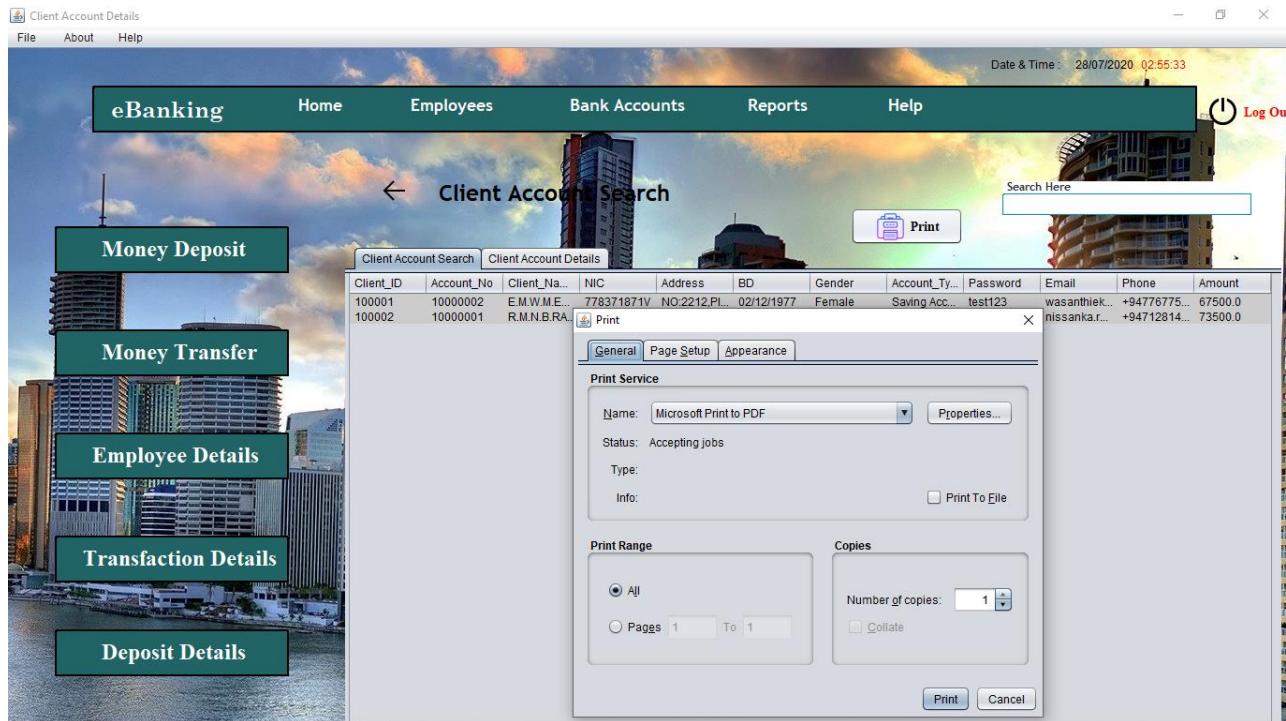
        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);
        }
    }
    tableLoad();
    editBoxdisable();
    tbClear();
}

```

```
//Back Button
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    tbClear();
    editBoxDisable();
    tableLoad();
}
private void tblClientMouseClicked(java.awt.event.MouseEvent evt) {
    JTpane.setSelectedIndex(1);
    tableData();
}
```

➤ Client Account Details Print



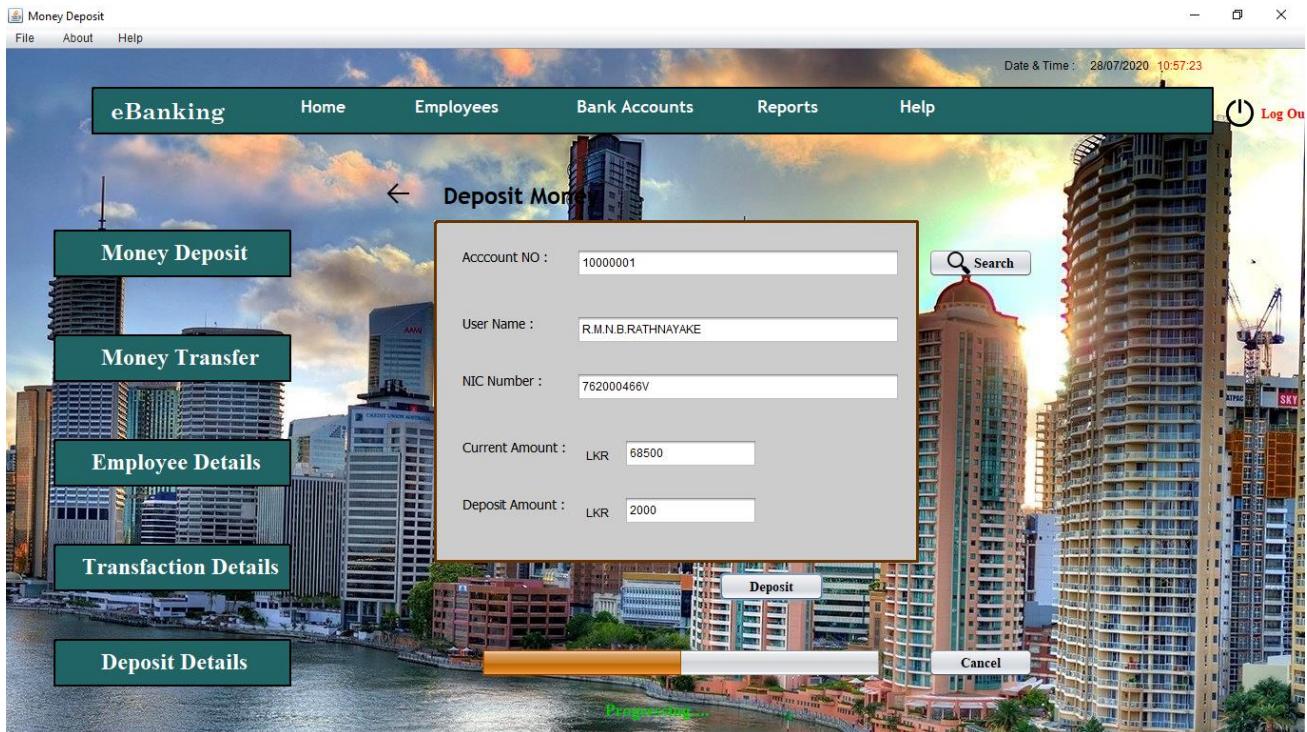
//Client Account Details Print

```
public void print(){
    MessageFormat header = new MessageFormat("Deposit Report");
    MessageFormat footer = new MessageFormat("-eBankONLINE-");

    try {
        tblClient.print(JTable.PrintMode.NORMAL, header, footer);
    } catch (java.awt.print.PrinterException e) {
        System.err.format("Cannot Print", e.getMessage());
    }
}

//Print Button
private void btnPrint3ActionPerformed(java.awt.event.ActionEvent evt) {
    print();
}
```

➤ Money Deposit



//Deposit Money

```
public class Deposit extends javax.swing.JFrame {  
  
    Connection conn;  
    PreparedStatement pst;  
    PreparedStatement pst2;  
    PreparedStatement pst3;  
    ResultSet rs;  
    progressThread th;  
  
    public Deposit() {  
        initComponents();  
        Login log = new Login();  
        log.setIcon();  
        setExtendedState(JFrame.MAXIMIZED_BOTH);  
        lblUser.setText(Home.lblUser.getText());  
        conn = DBconnect.connect();  
        btnCancel.hide();  
        date();  
        time();  
        ProgressBar.hide();  
    }  
}
```

```

//Search Account

public void search(){
    String srch = tbAccNo.getText();
    try {
        String sql = "SELECT * FROM tblSavingAccount WHERE
        Account_No='"+srch+"';

        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if (rs.next()){
            String name = rs.getString("User_Name");
            tbUserName.setText(name);
            String nic = rs.getString("NIC");
            tbNIC.setText(nic);
            String CAmount = rs.getString("Amount");
            tbCurrentAmount.setText(CAmount);

            rs.close();
            pst.close();
        }
        else{
            JOptionPane.showMessageDialog(null, "Invalid Input!");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

//Progress Bar

class progressThread extends Thread{

    JProgressBar proBar;

    progressThread(JProgressBar proBar){
        proBar = ProgressBar;
    }

    public void run(){
        int min = 0;
        int max =10;
    }
}

```

```

ProgressBar.setMaximum(min);
ProgressBar.setMaximum(max);
ProgressBar.setValue(0);

for(int i=min; i<=max; i++){
    ProgressBar.setValue(i);

    try {
        sleep(500);
    } catch (InterruptedException ex) {
        Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
    }

    if (i== max){
        JOptionPane.showMessageDialog(null, "Deposited Successfully!");
        ProgressBar.setValue(0);
        lblProgress.setText("");
        tbAccNo.setText("");
        tbAccNo.setEditable(true);
        tbUserName.setText("");
        tbNIC.setText("");
        tbCurrentAmount.setText("");
        tbDepositAmount.setText("");
        btnCancel.hide();
        ProgressBar.hide();
    }
}
}

//Search Button

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    search();
}

//Deposit Button

private void btnDepositActionPerformed(java.awt.event.ActionEvent evt) {

```

```

int check = JOptionPane.showConfirmDialog(null, "Are Details correct?");

if(check == 0){
String date = lblDate.getText();
String time = lblTime.getText();
String id = tbAccNo.getText();
String uname = tbUserName.getText();
String nic = tbNIC.getText();
String CAmount = tbCurrentAmount.getText();
String DAmount = tbDepositAmount.getText();
float value = Float.parseFloat(CAmount) + Float.parseFloat(DAmount);

try {
    String sql = "UPDATE tblSavingAccount SET Amount='"+value+"'
    WHERE Account_No='"+id+"'";

    String SQL2 ="UPDATE tblClient SET Amount='"+value+"'
    WHERE Account_No='"+id+"'";

    String SQL3 ="INSERT INTO tblDeposit
    Date,Time,Account_No,User_Name,NIC,Before_Amount,Deposit_Amount,Current_Amount
    VALUES
    ('"+date+"','"+time+"','"+id+"','"+uname+"','"+nic+"','"+CAmount+"','"+DAmount+"','"+value+"')";

    pst=conn.prepareStatement(sql);
    pst2 = conn.prepareStatement(SQL2);
    pst3 = conn.prepareStatement(SQL3);
    pst.execute();
    pst2.execute();
    pst3.execute();
    ProgressBar.show();
    lblProgress.setText("Progressing....");
    th = new progressThread(ProgressBar);
    th.start();
    btnCancel.show();
    btnCancel.enable();
    tbAccNo.setEditable(false);

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

```

```

//Cancel Button

private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {

    th.stop();

    int check = JOptionPane.showConfirmDialog(null, "Do you want to cancel
deposit?");

    if(check == 0){

        String id = tbAccNo.getText();
        String CAmount = tbCurrentAmount.getText();
        float value = Float.parseFloat(CAmount);

        try {
            String sql = "UPDATE tblSavingAccount SET Amount='"+value+"'
WHERE Account_No='"+id+"';

            pst=conn.prepareStatement(sql);

            pst.execute();

            JOptionPane.showMessageDialog(null, "Deposit Canceled!");
            ProgressBar.setValue(0);
            btnCancel.setEnabled(false);
            btnCancel.setVisible(false);

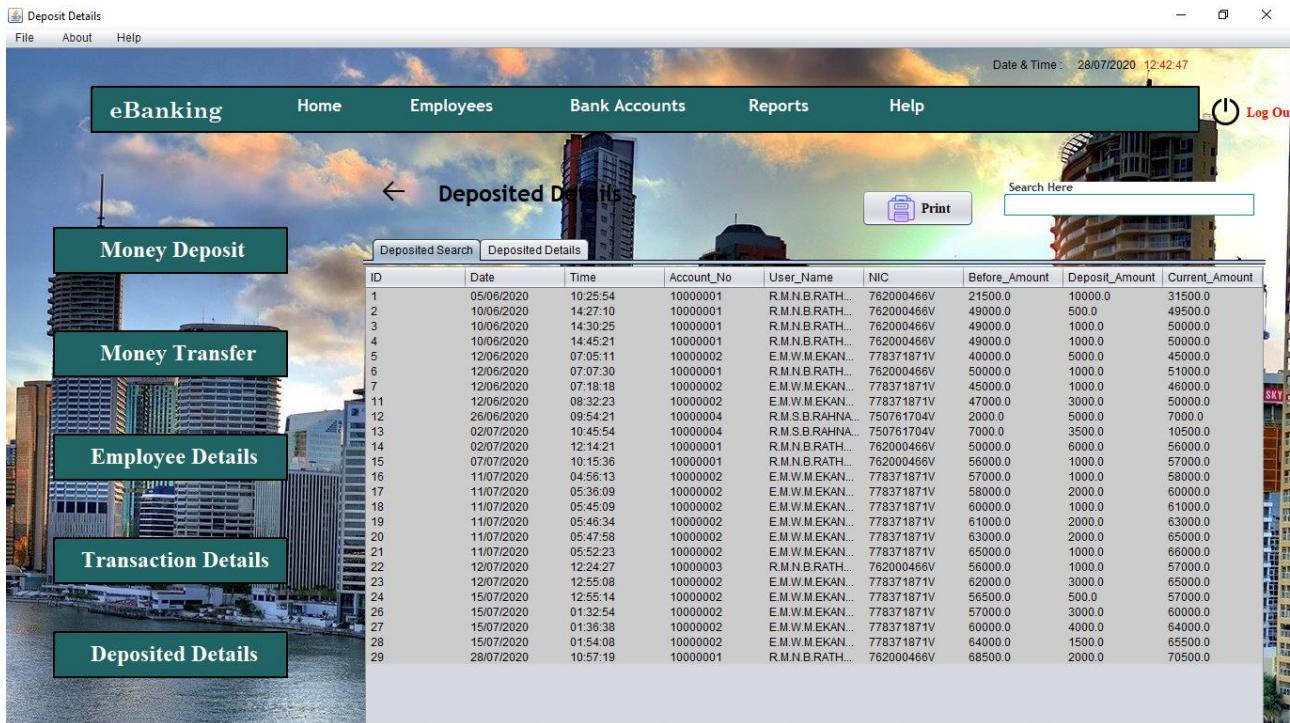
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }

        tbAccNo.setText("");
        tbUserName.setText("");
        tbNIC.setText("");
        tbCurrentAmount.setText("");
        tbDepositAmount.setText("");
        ProgressBar.setValue(0);
        lblProgress.setText("");
    }

}

```

➤ Deposited Details Search



//Deposited details search

```
public class DepositDetails extends javax.swing.JFrame {
```

```
    Connection conn;
```

```
    PreparedStatement pst;
```

```
    ResultSet rs;
```

```
    public DepositDetails() {
```

```
        initComponents();
```

```
        Login log = new Login();
```

```
        log.setIcon();
```

```
        setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
        lblUser.setText(Home.lblUser.getText());
```

```
        conn = DBconnect.connect();
```

```
        date();
```

```
        time();
```

```
        tableLoad();
```

```
}
```

```

//Table Load

public void tableLoad(){

    try {

        String sql = "SELECT * FROM tblDeposit";

        pst = conn.prepareStatement(sql);

        rs = pst.executeQuery();

        tblDeposit.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

//Search Details

public void search(){

    String srch = tbSearch.getText();

    try {

        String sql = "SELECT * FROM tblDeposit WHERE ID LIKE '%" + srch + "%' || Date LIKE '%" + srch + "%' || Account_No Like '%" + srch + "%' || User_Name Like '%" + srch + "%' || NIC Like '%" + srch + "%'";

        pst = conn.prepareStatement(sql);

        rs = pst.executeQuery();

        tblDeposit.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

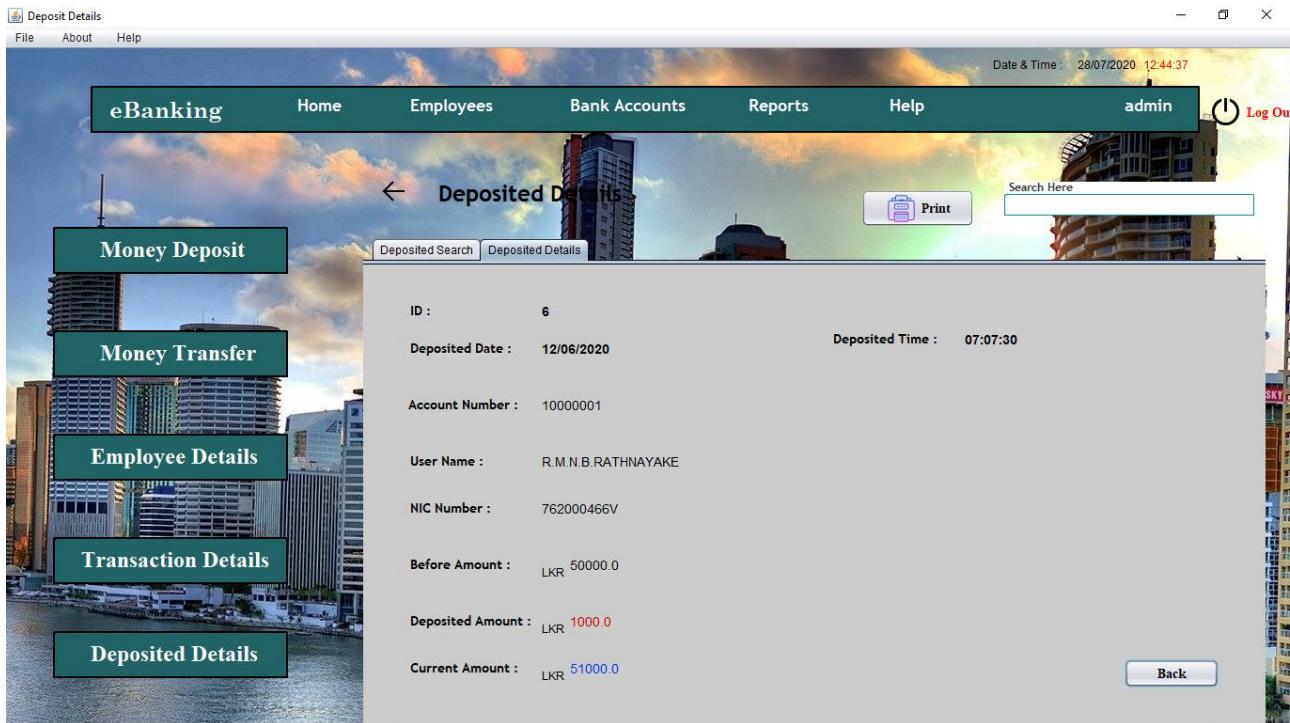
private void tbSearchKeyReleased(java.awt.event.KeyEvent evt) {

    search();

}

```

➤ Deposited Details



//Deposited Details

```
public void tableData(){

    int r = tblDeposit.getSelectedRow();

    String id = tblDeposit.getValueAt(r, 0).toString();
    String date = tblDeposit.getValueAt(r, 1).toString();
    String time = tblDeposit.getValueAt(r, 2).toString();
    String accno=tblDeposit.getValueAt(r, 3).toString();
    String uname =tblDeposit.getValueAt(r, 4).toString();
    String nic=tblDeposit.getValueAt(r, 5).toString();
    String BAmount =tblDeposit.getValueAt(r, 6).toString();
    String DAmount =tblDeposit.getValueAt(r, 7).toString();
    String CAmount =tblDeposit.getValueAt(r, 8).toString();
```

```
lblID.setText(id);
```

```
lblIDDate.setText(date);
```

```

lblDTime.setText(time);
lblAccountNo.setText(accno);
lblUName.setText(uname);
lblNIC.setText(nic);
lblBamount.setText(BAmount);
lblDamount.setText(DAmount);
lblCamount.setText(CAmount);

}

//Clear Data

public void clear(){
    lblID.setText("");
    lblDDate.setText("");
    lblAccountNo.setText("");
    lblUName.setText("");
    lblNIC.setText("");
    lblBamount.setText("");
    lblDamount.setText("");
    lblCamount.setText("");
}

private void tblDepositMouseClicked(java.awt.event.MouseEvent evt) {
    JTpane.setSelectedIndex(1);
    tableData();
}

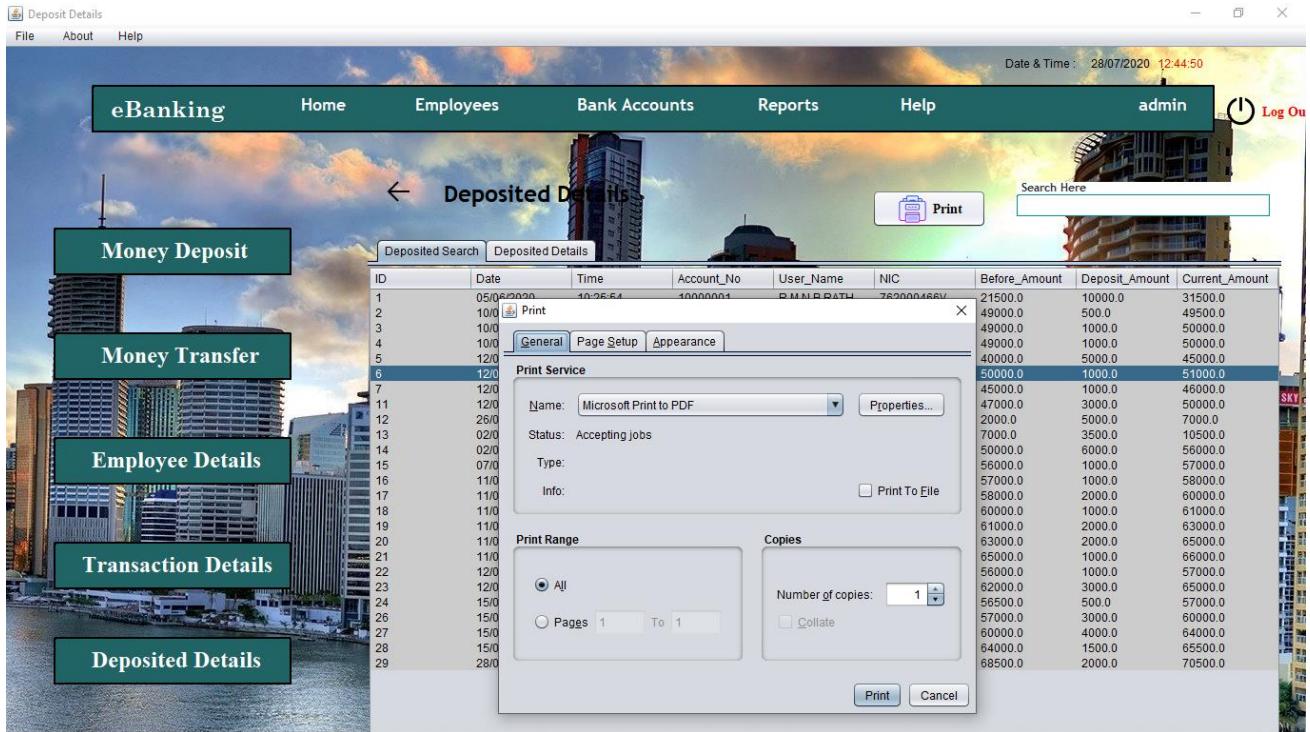
private void tblDepositMousePressed(java.awt.event.MouseEvent evt) {
    tableData();
}

```

```
//Back Button
```

```
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {  
    clear();  
    tableLoad();  
}
```

➤ Deposited Details Print



//Deposited details Print

public void print(){

 MessageFormat header = new MessageFormat("Deposit Report");

 MessageFormat footer = new MessageFormat("-eBankONLINE-");

 try {

 tblDeposit.print(JTable.PrintMode.NORMAL, header, footer);

 } catch (java.awt.print.PrinterException e) {

 System.err.format("Cannot Print", e.getMessage());

 }

}

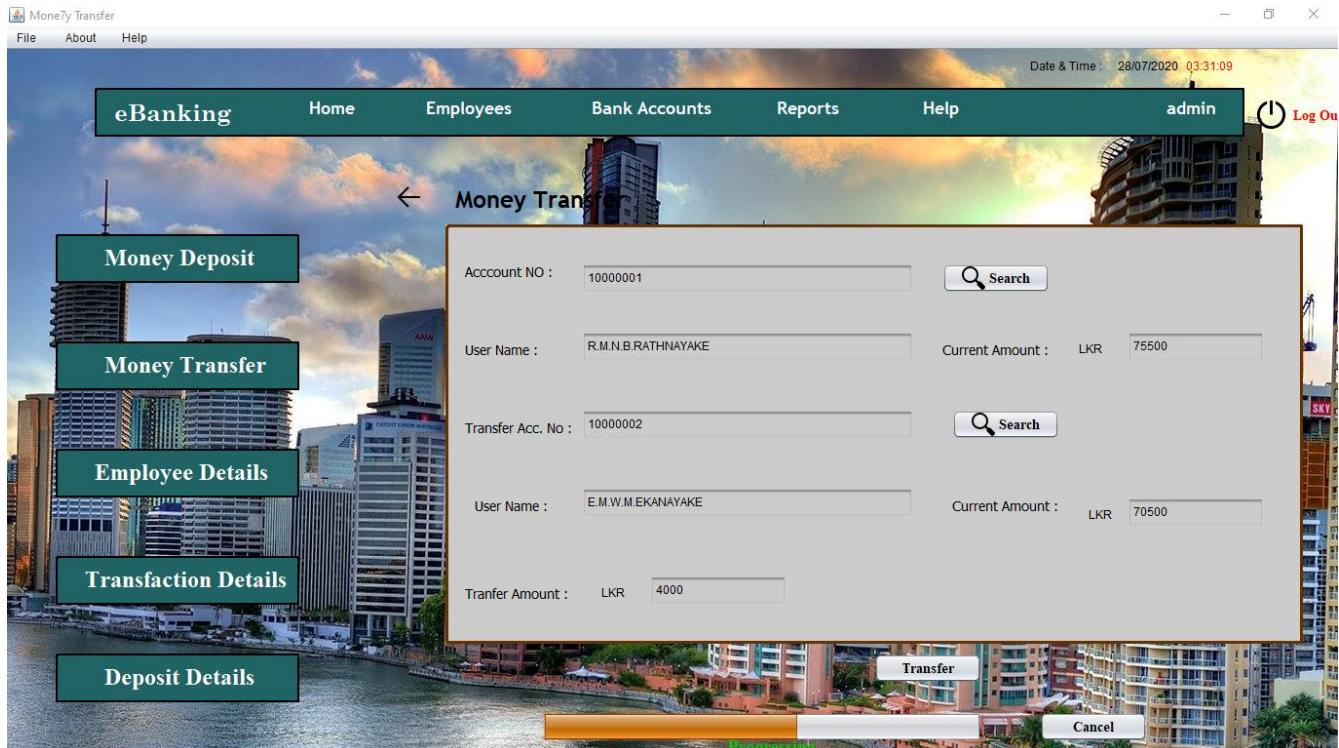
//Print Button

private void btnPrint3ActionPerformed(java.awt.event.ActionEvent evt) {

 print();

}

➤ Money Transfer



```
//Money Transfer
public class MoneyTransfer extends javax.swing.JFrame {
```

```
    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    PreparedStatement pst2;
    PreparedStatement pst3;
    PreparedStatement pst4;
    PreparedStatement pst5;
    progressThread th;
```

```
public MoneyTransfer() {
    initComponents();
    Login log = new Login();
    log.setIcon();
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    lblUser.setText(Home.lblUser.getText());
    conn = DBconnect.connect();
    btnCancel.enable(false);
    btnCancel.hide();
    Clear();
    date();
    time();
```

```

    }
//Search First Account

public void searchAcc1(){
    String srch = tbAccNo.getText();
    try {
        String sql = "SELECT * FROM tblSavingAccount WHERE
        Account_No='"+srch+"';

        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if (rs.next()){

            String name = rs.getString("User_Name");
            tbAccUserName.setText(name);
            String CAmount = rs.getString("Amount");
            tbCurrentAmount.setText(CAmount);

            rs.close();
            pst.close();

        }
        else{
            JOptionPane.showMessageDialog(null, "Invalid Input!");
        }

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

//Search Second Account

public void searchAcc2(){
    String srch = tbTransferAccNo.getText();
    String check = tbAccNo.getText();

    if(srch.equals(check)){

        JOptionPane.showMessageDialog(null, "Same Accounts.Pleas check
        Account Numbers");
        tbTransferAccNo.setText("");
    }
    else{
        try {

```

```

        String sql = "SELECT * FROM tblSavingAccount WHERE
Account_No='"+srch+"';

pst = conn.prepareStatement(sql);
rs = pst.executeQuery();

if (rs.next()){
    String name = rs.getString("User_Name");
    tbTransferUserName.setText(name);
    String CAmount = rs.getString("Amount");
    tbCurrentAmount1.setText(CAmount);

    rs.close();
    pst.close();

}
else{
    JOptionPane.showMessageDialog(null, "Invalid Input!");
}
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
}
}

}

//Progress Bar

```

```

class progressThread extends Thread{

JProgressBar proBar;

progressThread(JProgressBar proBar){

    proBar = ProgressBar;
}

public void run(){

    String Tamount = tbTransferAmount.getText();

    if(Float.parseFloat(Tamount) <= 20000){
        int min = 0;
        int max =50;
        ProgressBar.setMaximum(min);
        ProgressBar.setMaximum(max);
    }
}

```

```

    ProgressBar.setValue(0);

    for(int i=min; i<=max; i++){
        ProgressBar.setValue(i);

        try {
            sleep(500);

        } catch (InterruptedException ex) {
            Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
        }

        if (i==max){
            JOptionPane.showMessageDialog(null, "Transfaction Completed!");
            ProgressBar.setValue(0);
            lblProgress.setText("");
            btnCancel.enable(false);
            btnCancel.hide();
            clear();
        }
    }

    else{
        int min = 0;
        int max =100;
        ProgressBar.setMaximum(min);
        ProgressBar.setMaximum(max);
        ProgressBar.setValue(0);

        for(int i=min; i<=max; i++){
            ProgressBar.setValue(i);

            try {
                sleep(500);

            } catch (InterruptedException ex) {
                Logger.getLogger(Deposit.class.getName()).log(Level.SEVERE, null, ex);
            }

            if (i==max){

```

```

JOptionPane.showMessageDialog(null, "Transfaction Successfully!");
ProgressBar.setValue(0);
lblProgress.setText("");
btnCancel.enable(false);
btnCancel.hide();
clear();
}

}

}

}

}

//Clear All the Boxes

public void clear(){
    tbAccNo.setText("");
    tbAccUserName.setText("");
    tbCurrentAmount.setText("");
    tbTransferAccNo.setText("");
    tbTransferUserName.setText("");
    tbCurrentAmount1.setText("");
    tbTransferAmount.setText("");
    ProgressBar.hide();
}

//Search Button 1

private void btnAccSearchActionPerformed(java.awt.event.ActionEvent evt) {
    searchAcc1();
}

//Search Button 2
private void btnTransSearchActionPerformed(java.awt.event.ActionEvent evt) {
    searchAcc2();
}

//Transfer Button

private void btnTransfer1ActionPerformed(java.awt.event.ActionEvent evt) {

    String fill=tbTransferAmount.getText();
    if(fill.equals("")){

        JOptionPane.showMessageDialog(null, "Please Enter Amount you want
        to Transfer!");
    }
}

```

```

    }
    else{
        int check = JOptionPane.showConfirmDialog(null, "Are you sure?");

        if(check == 0){
            String date = lblDate.getText();
            String time = lblTime.getText();
            String Aid = tbAccNo.getText();
            String Auname = tbAccUserName.getText();
            String Aamount = tbCurrentAmount.getText();
            String Bid = tbTransferAccNo.getText();
            String Buname = tbTransferUserName.getText();
            String Bamount = tbCurrentAmount1.getText();
            String Tamamount = tbTransferAmount.getText();
            float MinValue = Float.parseFloat(Aamount)-Float.parseFloat(Tamamount);
            float AddValue = Float.parseFloat(Bamount)+Float.parseFloat(Tamamount);

            if(Float.parseFloat(Aamount)<=      Float.parseFloat(Tamamount)      &&
               Float.parseFloat(Aamount) <= 1500 && Float.parseFloat(Tamamount) -
               Float.parseFloat(Aamount) <=1500 ){

                JOptionPane.showMessageDialog(null, "Insufficient Balance!");

            }
            else{

                try {

                    String     sql      =      "UPDATE      tblSavingAccount      SET
                    Amount='"+MinValue+"' WHERE Account_No='"+Aid+"' ";

                    String     SQL2      =      "UPDATE      tblSavingAccount      SET
                    Amount='"+AddValue+"' WHERE Account_No='"+Bid+"'";

                    String SQL3 = "UPDATE tblClient SET Amount='"+MinValue+"'
                    WHERE Account_No='"+Aid+"' ";

                    String SQL4 = "UPDATE tblClient SET Amount='"+AddValue+"'
                    WHERE Account_No='"+Bid+"'";

                    String     SQL5      ="INSERT      INTO      tblMoneyTransfer
                    (Date,Time,From_Acc,From_User_Name,FBefore_Amount,To_Acc
                    ,To_User_Name,TBefore_Amount,Transferred_Amount)  VALUES
                    ('"+date+"','"+time+"','"+Aid+"','"+Auname+"','"+Aamount+"','"+Bid
                    +"','"+Buname+"','"+Bamount+"','"+Tamount+"')";

                }
            }
        }
    }
}

```

```

pst = conn.prepareStatement(sql);
pst2 = conn.prepareStatement(SQL2);
pst3 = conn.prepareStatement(SQL3);
pst4 = conn.prepareStatement(SQL4);
pst5 = conn.prepareStatement(SQL5);
pst.execute();
pst2.execute();
pst3.execute();
pst4.execute();
pst5.execute();
ProgressBar.show();
lblProgress.setText("Progressing....");
th = new progressThread(ProgressBar);
th.start();
btnCancel.show();
btnCancel.enable();
tbAccNo.setEditable(false);
tbTransferAccNo.setEditable(false);

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

//Cancel Button
private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {

    th.stop();
    int check = JOptionPane.showConfirmDialog(null, "Do you want to cancel?");

    if(check == 0){

        String Aid = tbAccNo.getText();
        String Aamount = tbCurrentAmount.getText();
        String Bid = tbTransferAccNo.getText();
        String Bamount = tbCurrentAmount1.getText();

        float MinValue = Float.parseFloat(Aamount);
        float AddValue = Float.parseFloat(Bamount);
    }
}

```

```
try {

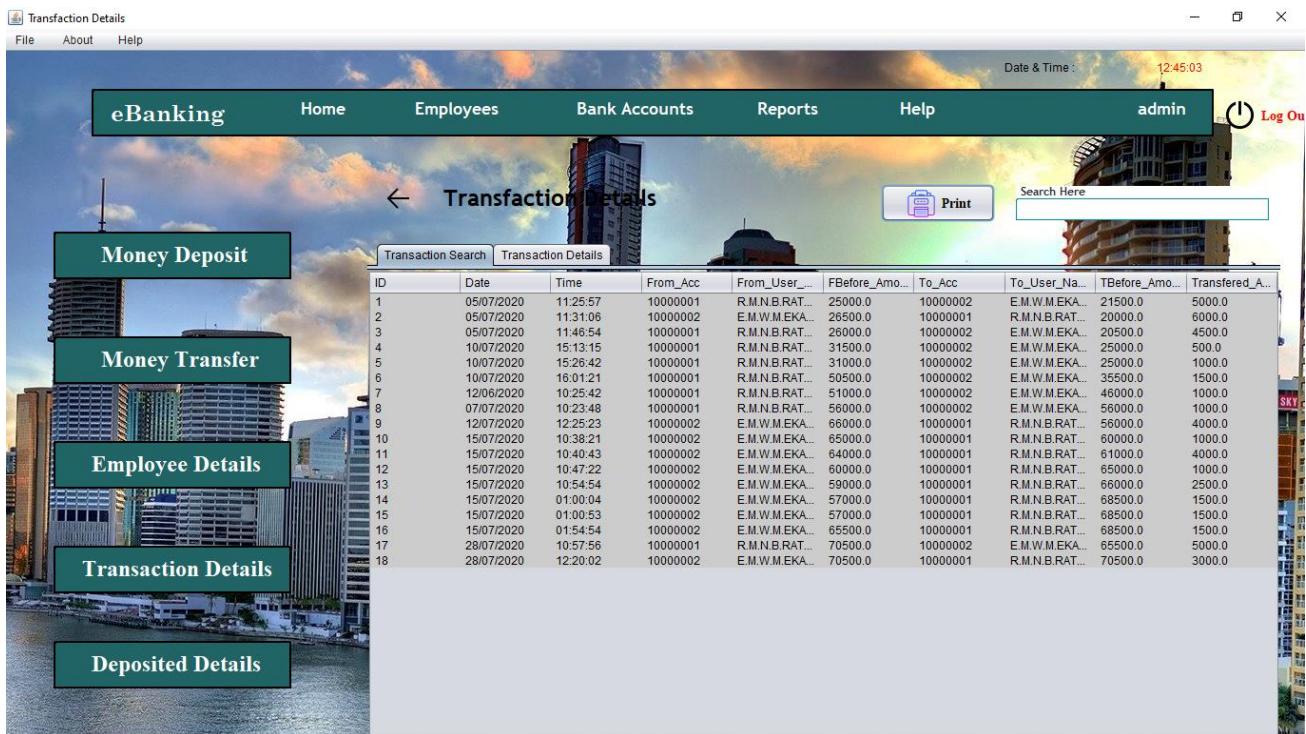
    String     sql      =      "UPDATE      tblSavingAccount      SET
    Amount='"+MinValue+"' WHERE Account_No='"+Aid+"' ";

    String     SQL2     =      "UPDATE      tblSavingAccount      SET
    Amount='"+AddValue+"' WHERE Account_No='"+Bid+"' ";

    pst = conn.prepareStatement(sql);
    pst2 =conn.prepareStatement(SQL2);
    pst.execute();
    pst2.execute();
    JOptionPane.showMessageDialog(null, "Tranfaction Canceled!");
    ProgressBar.setValue(0);

} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
clear();
lblProgress.setText("");
btnCancel.enable(false);
btnCancel.hide();
}
```

➤ Transaction Details Search



//Transaction Details Search

```
public class Transactiondetails extends javax.swing.JFrame {
```

```
    Connection conn;
```

```
    PreparedStatement pst;
```

```
    ResultSet rs;
```

```
    public Transactiondetails() {
```

```
        initComponents();
```

```
        Login log = new Login();
```

```
        log.setIcon();
```

```
        setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
        lblUser.setText(Home.lblUser.getText());
```

```
        conn = DBconnect.connect();
```

```
        date();
```

```
        time();
```

```
        tableLoad();
```

```
}
```

```

//Table Load

public void tableLoad(){

    try {

        String sql = "SELECT * FROM tblMoneyTransfer";
        pst=conn.prepareStatement(sql);
        rs = pst.executeQuery();
        tblTransfer.setModel(DbUtils.resultSetToTableModel(rs));

    }

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

//Search

public void search(){

    String srch = tbSearch.getText();

    try {

        String sql = "SELECT * FROM tblMoneyTransfer WHERE ID LIKE '%" +
        srch + "%' || From_User_Name LIKE '%" + srch + "%' || From_Acc Like
        '%" + srch + "%'";

        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        tblTransfer.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {
}
}

```

```
        JOptionPane.showMessageDialog(null, e);

    }

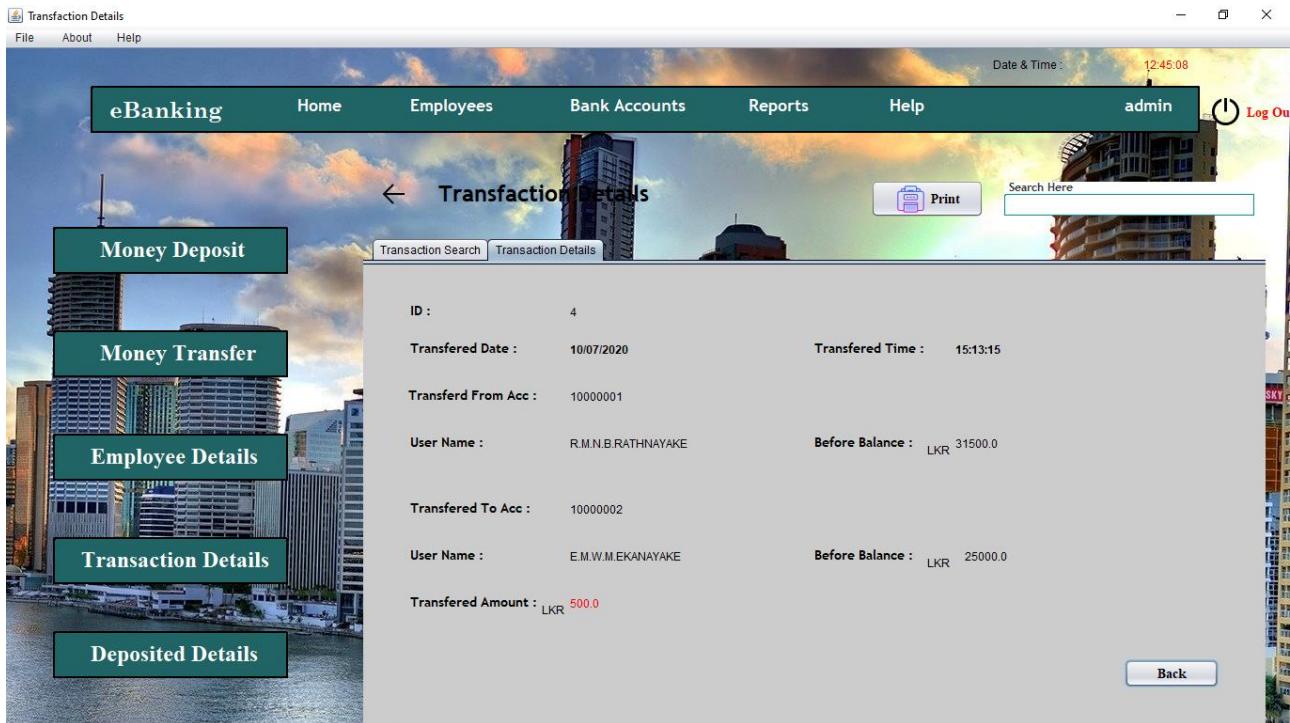
}

private void tbSearchKeyReleased(java.awt.event.KeyEvent evt) {

    search();

}
```

➤ Transaction Details



//Transaction Details

```
public void tableData(){  
    int r = tblTransfer.getSelectedRow();  
    String id =tblTransfer.getValueAt(r, 0).toString();  
    String date =tblTransfer.getValueAt(r, 1).toString();  
    String time =tblTransfer.getValueAt(r, 2).toString();  
    String facc =tblTransfer.getValueAt(r, 3).toString();  
    String funame =tblTransfer.getValueAt(r, 4).toString();  
    String fcamount=tblTransfer.getValueAt(r, 5).toString();  
    String tacc =tblTransfer.getValueAt(r, 6).toString();  
    String tuname =tblTransfer.getValueAt(r, 7).toString();  
    String tcamount =tblTransfer.getValueAt(r, 8).toString();  
    String transamount =tblTransfer.getValueAt(r, 9).toString();  
}
```

```
lblID.setText(id);
```

```
lblTDate.setText(date);
```

```

lblTTime.setText(time);
lblFAcc.setText(facc);
lblFuname.setText(funame);
lblFbalance.setText(fcamount);
lblTAcc.setText(tacc);
lblTuname.setText(tuname);
lblTbalance.setText(tcamount);
lblTransferredA.setText(transamount);

}

//Clear data

public void clear(){

    lblID.setText("");
    lblTDate.setText("");
    lblFAcc.setText("");
    lblFuname.setText("");
    lblFbalance.setText("");
    lblTAcc.setText("");
    lblTuname.setText("");
    lblTbalance.setText("");
    lblTransferredA.setText("");

}

private void tblTransferMouseClicked(java.awt.event.MouseEvent evt) {

    JTpane.setSelectedIndex(1);
    tableData();
}

private void tblTransferMousePressed(java.awt.event.MouseEvent evt) {

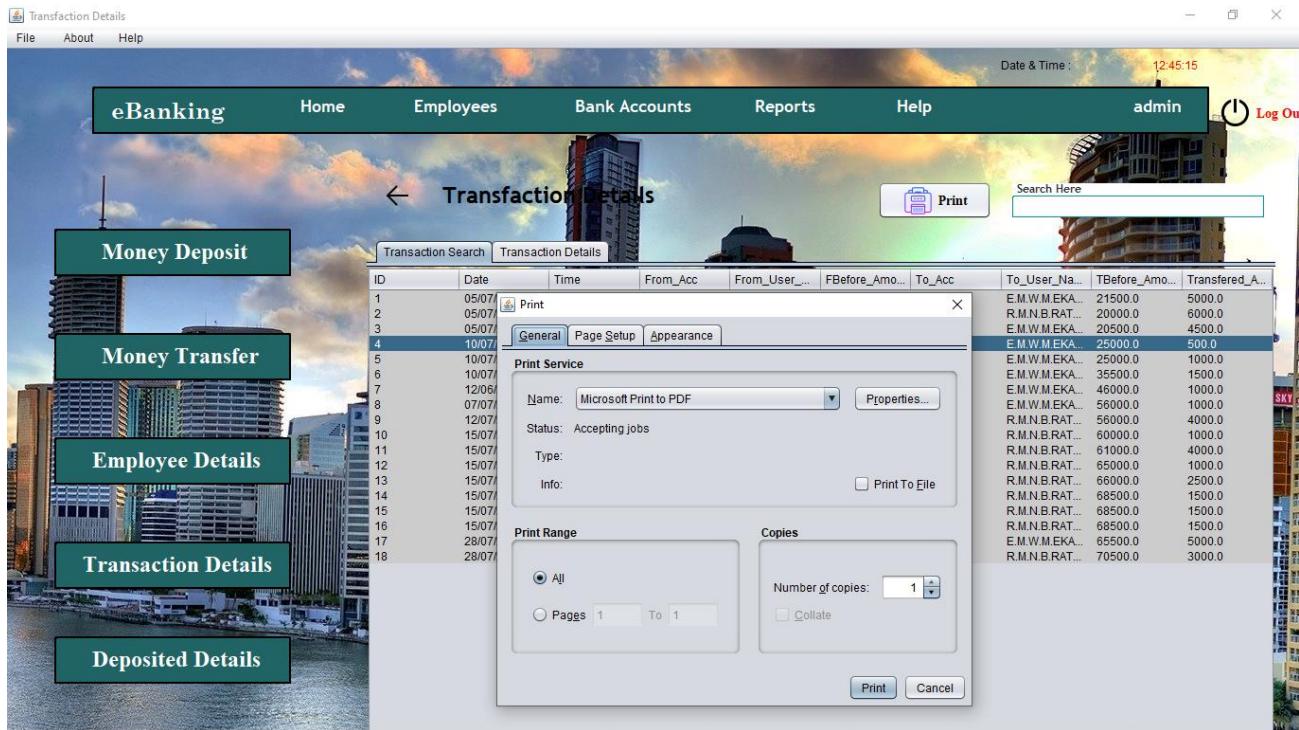
    tableData();
}

}

```

```
//Back Button  
private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {  
    clear();  
    tableLoad();  
}  
}
```

➤ Transaction Details Report



//Transaction Details Print

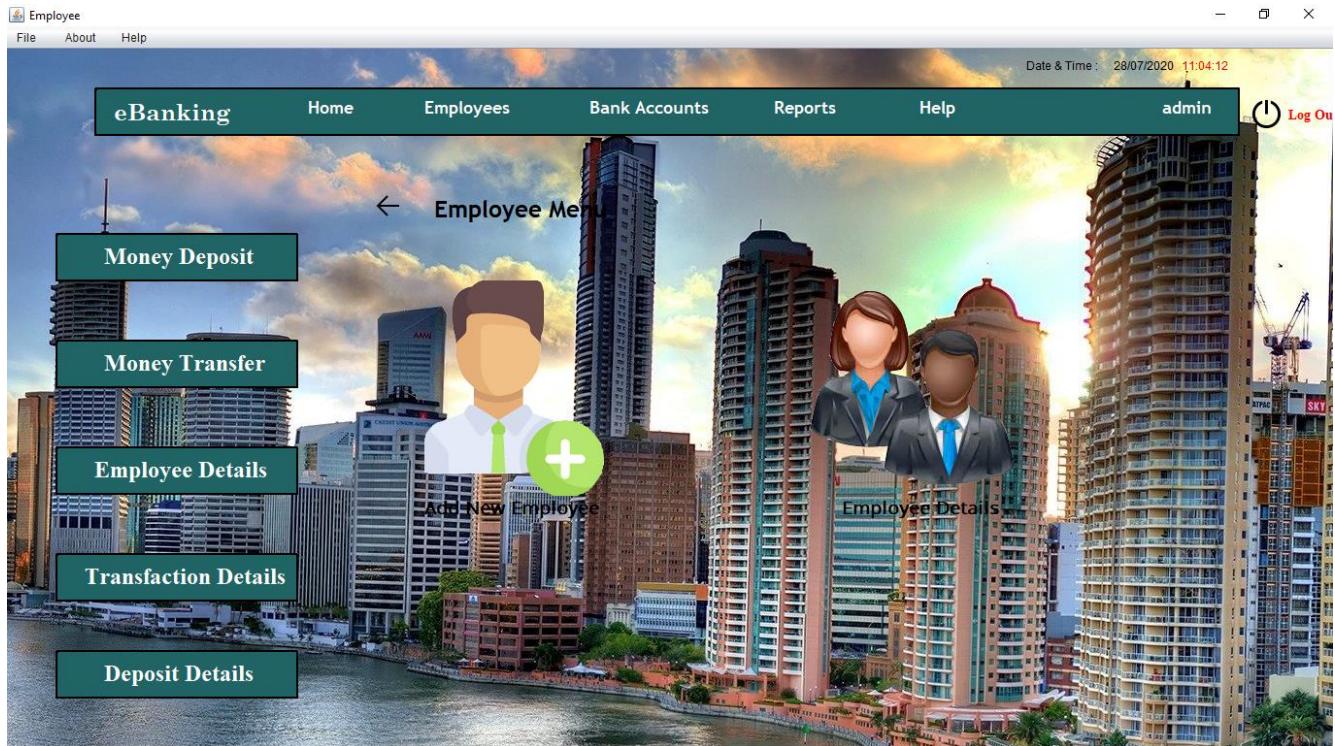
```
public void print(){
    MessageFormat header = new MessageFormat("Deposit Report");
    MessageFormat footer = new MessageFormat("-eBankONLINE-");
```

```
    try {
        tblTransfer.print(JTable.PrintMode.NORMAL, header, footer);
    } catch (java.awt.print.PrinterException e) {
        System.err.format("Cannot Print", e.getMessage());
    }
}
```

//Print Button

```
private void btnPrint3ActionPerformed(java.awt.event.ActionEvent evt) {
    print();
}
```

➤ Employee



//Employee Menu

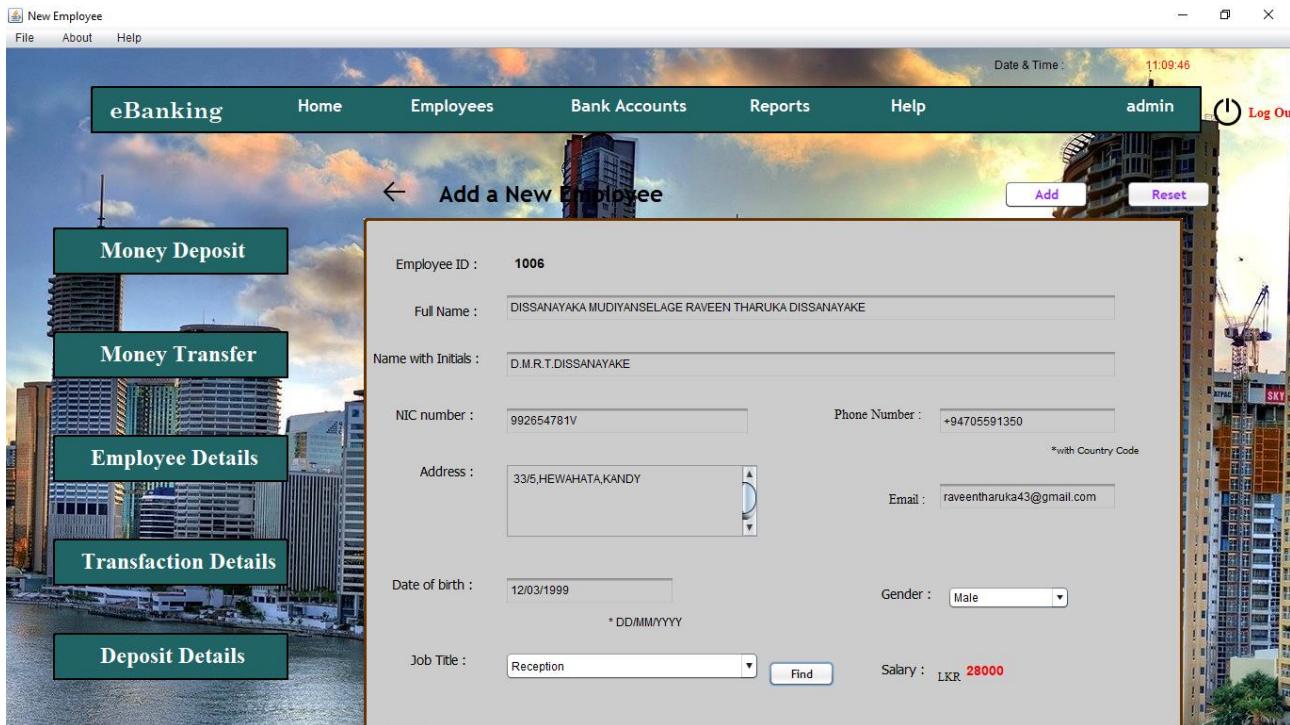
//New Employee

```
private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {  
    newEmployee nemp = new newEmployee();  
    nemp.setVisible(true);  
    this.dispose();  
}
```

//Employee Details

```
private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {  
    EmployeeAccountSearch EAS = new EmployeeAccountSearch();  
    EAS.setVisible(true);  
    this.dispose();  
}
```

➤ New Employee



//New Employee

```
public class newEmployee extends javax.swing.JFrame {
```

```
    Connection conn;
```

```
    PreparedStatement pst;
```

```
    ResultSet rs;
```

```
    public newEmployee() {
```

```
        initComponents();
```

```
        Login log = new Login();
```

```
        log.setIcon();
```

```
        setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
        lblUser.setText(Home.lblUser.getText());
```

```
        date();
```

```
        time();
```

```
        conn = DBconnect.connect();
```

```
        empID();
```

```
        clear();
```

```

}

//Automatically detect Employee ID

public void empID(){

    try {

        String sql = "SELECT * FROM tblEmployee ORDER BY Emp_ID DESC";

        pst = conn.prepareStatement(sql);

        rs = pst.executeQuery();

        if (rs.next()){

            int id = rs.getInt("Emp_ID");

            int relID = id +1;

            lblEmplID.setText(Integer.toString(relID));

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

//Clear all data

public void clear(){

    tbFName.setText("");

    tbNameWithInitials.setText("");

    tbNIC.setText("");

    tbAddress.setText("");

    tbBirth.setText("");

    tbPhone.setText("");

    tbEmail.setText("");

    cbJobTitle.setSelectedIndex(0);

    cbGender.setSelectedIndex(0);

    lblSalary.setText("");
}

```

```

}

//Salary find button

private void btnSalActionPerformed(java.awt.event.ActionEvent evt) {

    String job = cbJobTitle.getSelectedItem().toString();

    try {

        String sql = "SELECT * FROM tblSalary WHERE Job_Title='"+job+"'";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if(rs.next()){

            String sal = rs.getString("Salary");
            lblSalary.setText(sal);

        }

        else{
            JOptionPane.showMessageDialog(null, "Select a job!");
        }

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

}

//Add Button

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {

    String jdate = lblDate.getText();
    String fname = tbFName.getText();
    String ename = tbNameWithInitials.getText();
    String nic = tbNIC.getText();
    String address = tbAddress.getText();
    String bd = tbBirth.getText();
    String phone = tbPhone.getText();
    String email = tbEmail.getText();
}

```

```

String job = cbJobTitle.getSelectedItem().toString();
String gen = cbGender.getSelectedItem().toString();
float sal = Float.parseFloat(lblSalary.getText());

if(ename.equals("") || nic.equals("") || address.equals("") || bd.equals("") ||
phone.equals("") || job.equals(0) || gen.equals(0)){
    JOptionPane.showMessageDialog(null, "Please fill all the details!");
}

else{
    try {
        String sql = "INSERT INTO tblEmployee
(Emp_Name,FullName,NIC,Address,BD,Gender,Email,Phone,Job_Title,
Salary,Joined_Date)
VALUES
(" +ename+ "," +fname+ "," +nic+ "," +address+ "," +bd+ "," +gen+ "," +ema
il+ "," +phone+ "," +job+ "," +sal+ "," +jdate+ ")";
        pst = conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Employee Added
Successfully!");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
    empID();
    clear();
}

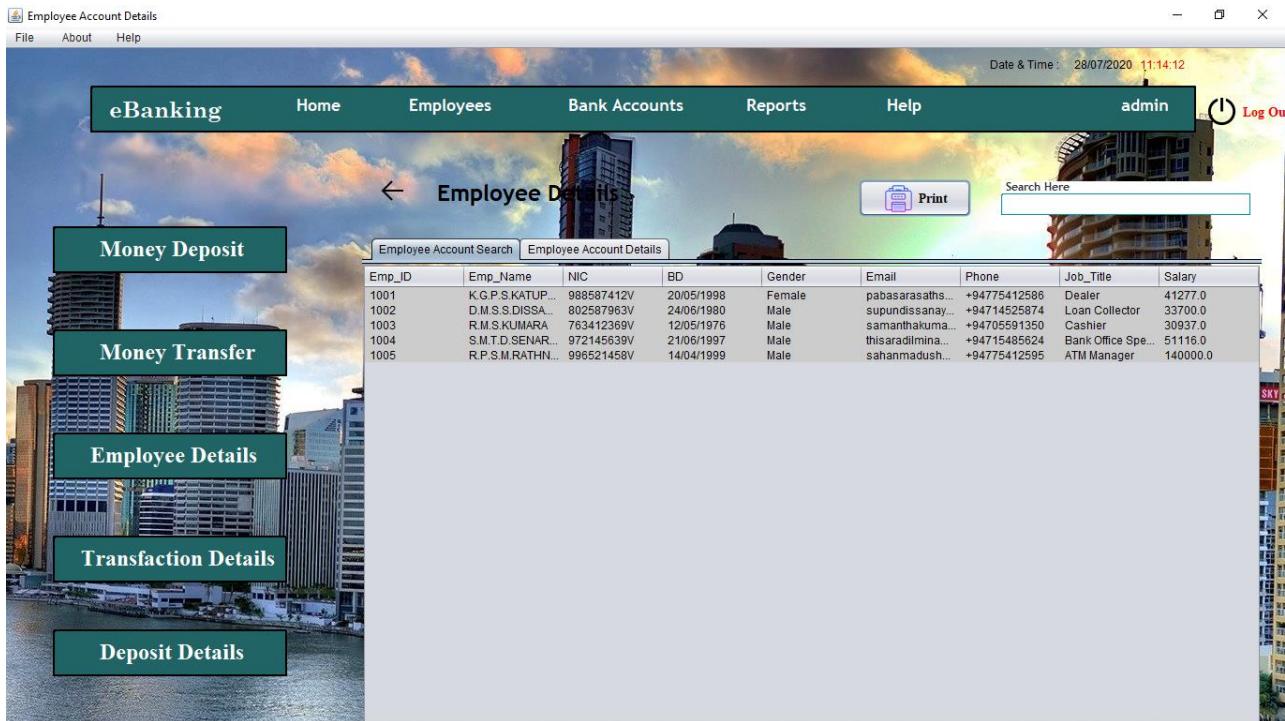
}

//btnReset

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    clear();
}

```

➤ Employee Details Search



//Employee Details Search

```
public class EmployeeAccountSearch extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;

    public EmployeeAccountSearch() {
        initComponents();
        Login log = new Login();
        log.setIcon();
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        lblUser.setText(Home.lblUser.getText());
        conn = DBconnect.connect();
        date();
        time();
        tableLoad();
        editBoxdisable();
    }
}
```

```

}

//Table Load

public void tableLoad() {

    try {

        String sql = "SELECT
        Emp_ID,Emp_Name,NIC,BD,Gender,Email,Phone,Job_Title,Sala
        ry FROM tblEmployee";

        pst = conn.prepareStatement(sql);

        rs = pst.executeQuery();

        tblEmp.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

//Search

public void search(){

    String srch = tbSearch.getText();

    try {

        String sql = "SELECT * FROM tblEmployee WHERE Emp_Name LIKE
        '%" + srch + "%' || Account_No Like '%" + srch + "%' || NIC LIKE '%" +
        srch + "%'";

        pst = conn.prepareStatement(sql);

        rs = pst.executeQuery();

        tblEmp.setModel(DbUtils.resultSetToTableModel(rs));

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

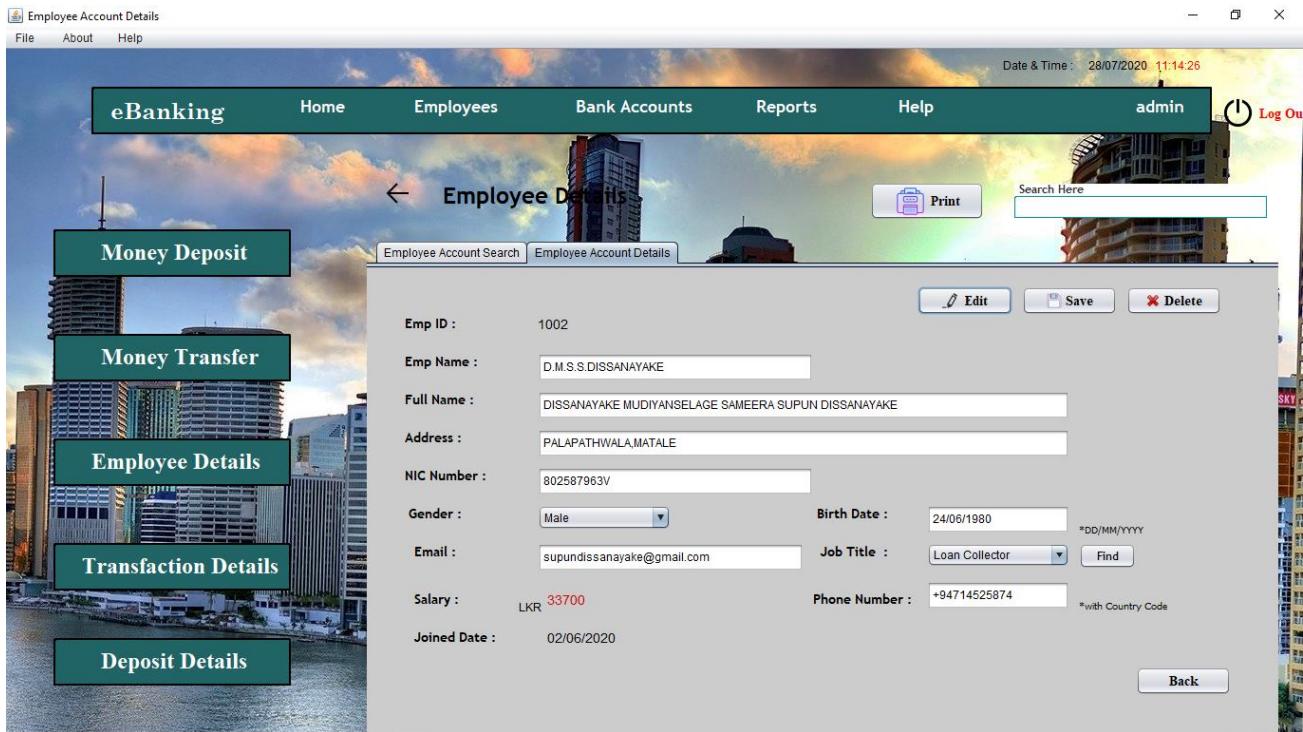
}

private void tbSearchKeyReleased(java.awt.event.KeyEvent evt) {

    search();    }

```

➤ Employee Details



//Employee Details

```

public void tableData(){

    int r = tblEmp.getSelectedRow();

    String EmpID = tblEmp.getValueAt(r, 0).toString();

    try {

        String sql = "SELECT * FROM tblEmployee WHERE Emp_ID ="+EmpID +"";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();

        if (rs.next()) {

            String id = rs.getString("Emp_ID");
            lblEmpID.setText(id);

            String UName = rs.getString("Emp_Name");
            tbEname.setText(UName);

            String FName = rs.getString("FullName");
            tbFName.setText(FName);

            String nic = rs.getString("NIC");

```

```

tbNIC.setText(nic);

String Address = rs.getString("Address");

tbAddress.setText(Address);

String bd = rs.getString("BD");

tbBD.setText(bd);

String gen = rs.getString("Gender");

cbGender.setSelectedItem(gen);

String jobTitle = rs.getString("Job_Title");

cbJTitle.setSelectedItem(jobTitle);

String email = rs.getString("Email");

tbEmail.setText(email);

String phone = rs.getString("Phone");

tbPhone.setText(phone);

String salary = rs.getString("Salary");

lblSalary.setText(salary);

String jDate= rs.getString("Joined_Date");

lblJDate.setText(jDate);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

//Make textboxes as editable

public void editBoxEnable(){

tbEname.setEditable(true);

tbFName.setEditable(true);

tbNIC.setEditable(true);

tbAddress.setEditable(true);

tbBD.setEditable(true);

```

```
cbJTitle.setEditable(true);
cbGender.setEditable(true);
tbEmail.setEditable(true);
tbPhone.setEditable(true);
}

//Disable Text boxes

public void editBoxdisable(){
    tbEname.setEditable(false);
    tbFName.setEditable(false);
    tbNIC.setEditable(false);
    tbAddress.setEditable(false);
    tbBD.setEditable(false);
    cbJTitle.setEditable(false);
    cbGender.setEditable(false);
    tbEmail.setEditable(false);
    tbPhone.setEditable(false);
    lblJDate.setText("");
}

//Clear all textboxes

public void tbClear(){
    lblEmpID.setText("");
    tbEname.setText("");
    tbFName.setText("");
    tbNIC.setText("");
    tbAddress.setText("");
    tbBD.setText("");
    cbJTitle.setSelectedIndex(0);
    cbGender.setSelectedIndex(0);
    tbEmail.setText("");
}
```

```

tbPhone.setText("");
lblSalary.setText("");
}

private void tblEmpMouseClicked(java.awt.event.MouseEvent evt) {
    JTpane.setSelectedIndex(1);
    tableData();
}

private void tblEmpMousePressed(java.awt.event.MouseEvent evt) {
    tableData();
}

//Find Salary

private void btnSalActionPerformed(java.awt.event.ActionEvent evt) {
    String job = cbJTitle.getSelectedItem().toString();
    try {
        String sql = "SELECT * FROM tblSalary WHERE Job_Title='"+job+"'";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        if(rs.next()){
            String sal = rs.getString("Salary");
            lblSalary.setText(sal);
        }
        else{
            JOptionPane.showMessageDialog(null, "Select a job!");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

```

//Edit Button

private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    editBoxEnable();
}

//Save Button

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    int id = Integer.parseInt(lblEmpID.getText());
    String ename = tbEname.getText();
    String fname = tbFName.getText();
    String address = tbAddress.getText();
    String nic = tbNIC.getText();
    String gen = cbGender.getSelectedItem().toString();
    String email = tbEmail.getText();
    String bd = tbBD.getText();
    String jobTitle = cbJTitle.getSelectedItem().toString();
    String phone = tbPhone.getText();
    float sal = Float.parseFloat(lblSalary.getText());

    try {
        String sql = "UPDATE tblEmployee SET Emp_Name='"+ename+"',FullName='"+fname+"',NIC='"+nic+"',Address='"+address+"',BD='"+bd+"',Gender='"+gen+"',Email='"+email+"',Phone='"+phone+"',Job_Title='"+jobTitle+"',Salary = '"+sal+"' WHERE Emp_ID='"+id+"'";
        pst = conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Updated Successfully!");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
    tableLoad();
}

```

```

tbClear();

editBoxdisable();

}

//Delete Button

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {

    String id = lblEmpID.getText();

    int check = JOptionPane.showConfirmDialog(null, "Do you want to delete?");

    if(check == 0){

        try {

            String sql = "DELETE FROM tblEmployee WHERE Account_No='"+id+"';

            pst = conn.prepareStatement(sql);

            pst.execute();

            JOptionPane.showMessageDialog(null, "Account Deleted!");



        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }

        tableLoad();

    }

}

//Back Button

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

    tbClear();

    editBoxdisable();

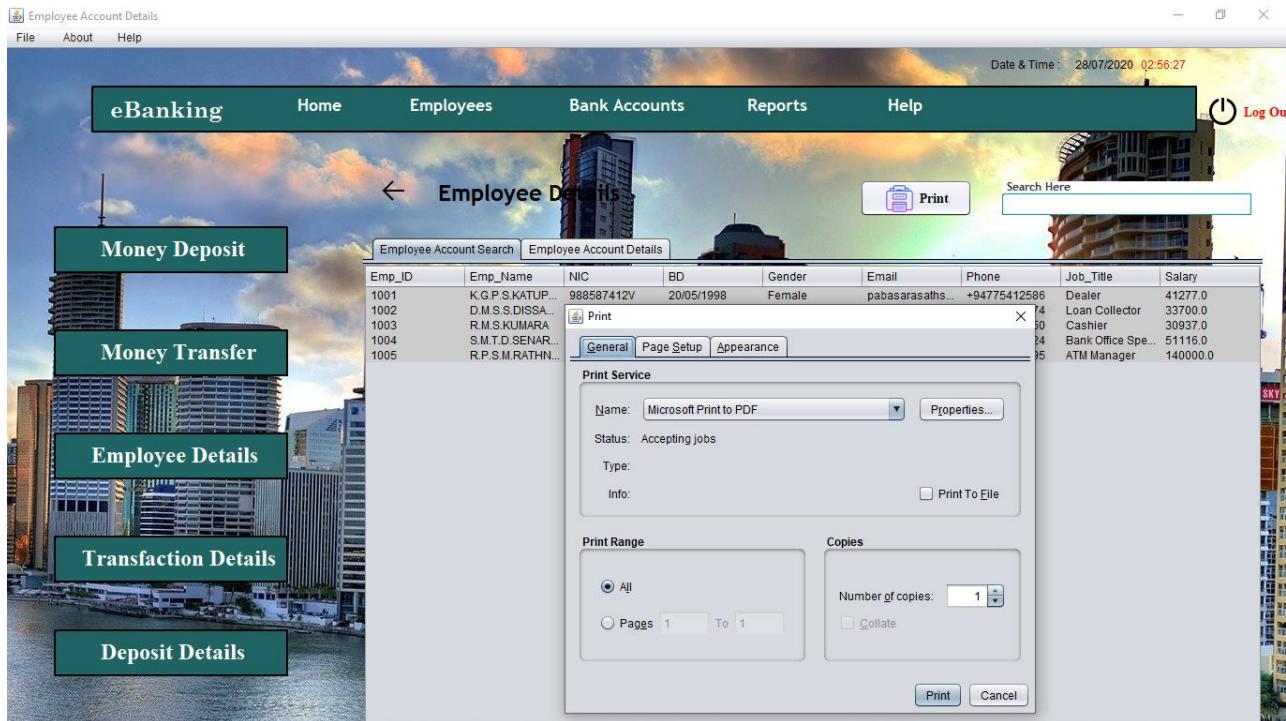
    tableLoad();

}

}

```

➤ Employee Details Print



//Employee Details Print

```
public void print(){
```

```
    MessageFormat header = new MessageFormat("Deposit Report");
```

```
    MessageFormat footer = new MessageFormat("-eBankONLINE-");
```

```
    try {
```

```
        tblEmp.print(JTable.PrintMode.NORMAL, header, footer);
```

```
    } catch (java.awt.print.PrinterException e) {
```

```
        System.err.format("Cannot Print", e.getMessage());
```

```
    }
```

```
}
```

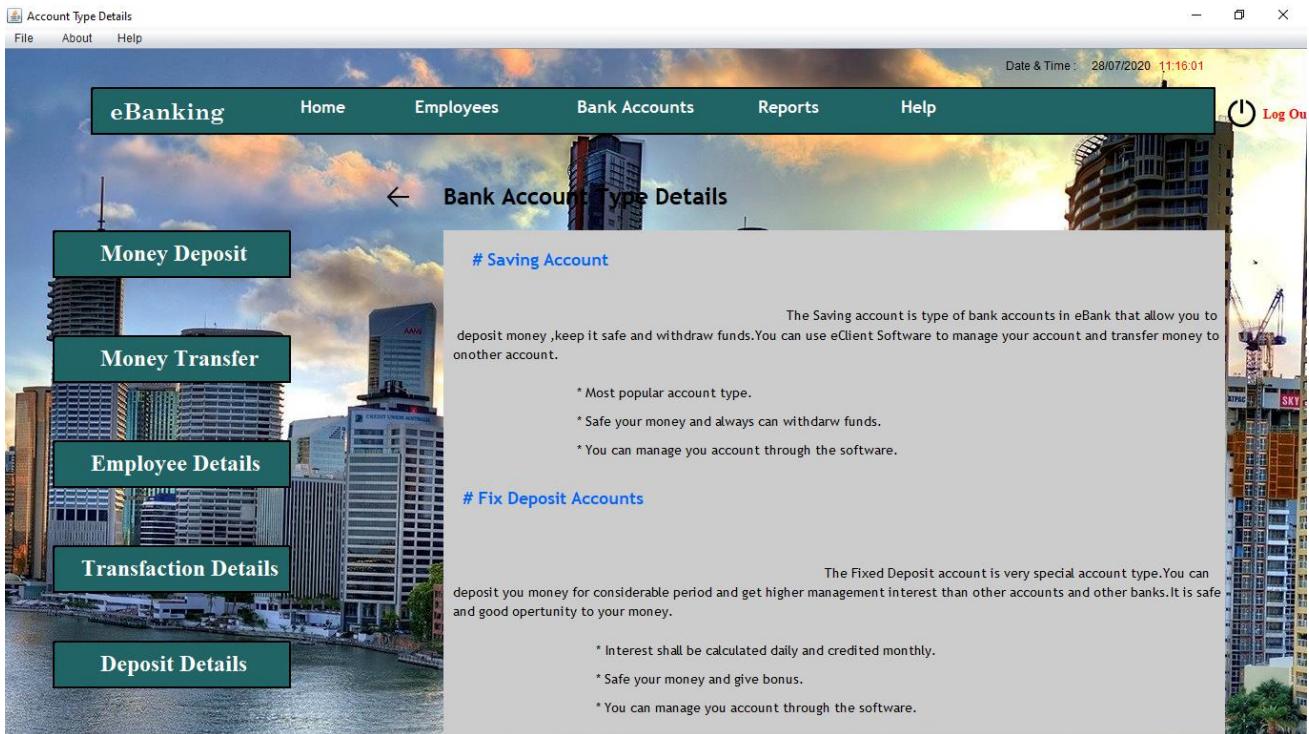
//Print Button

```
private void btnPrint3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    print();
```

```
}
```

➤ Bank Account Types



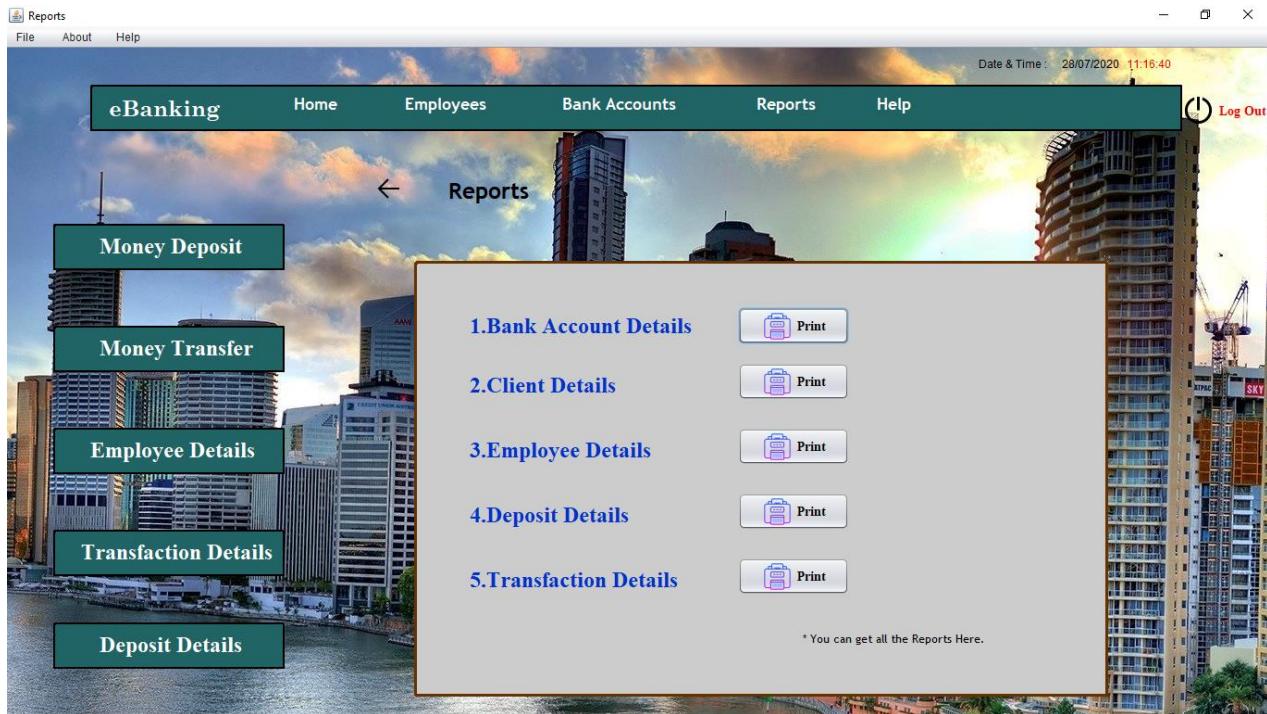
//Bank Account Types

/* There are two bank accounts in this bank.This form displays all the details of these accounts named Saving Account and Fix Deposit Account.

*/

```
public AccountTypeDetails() {  
    initComponents();  
  
    Login log = new Login();  
  
    log.setIcon();  
  
    lblUser.setText(Home.lblUser.getText());  
  
    setExtendedState(JFrame.MAXIMIZED_BOTH);  
  
    date();  
  
    time();  
}
```

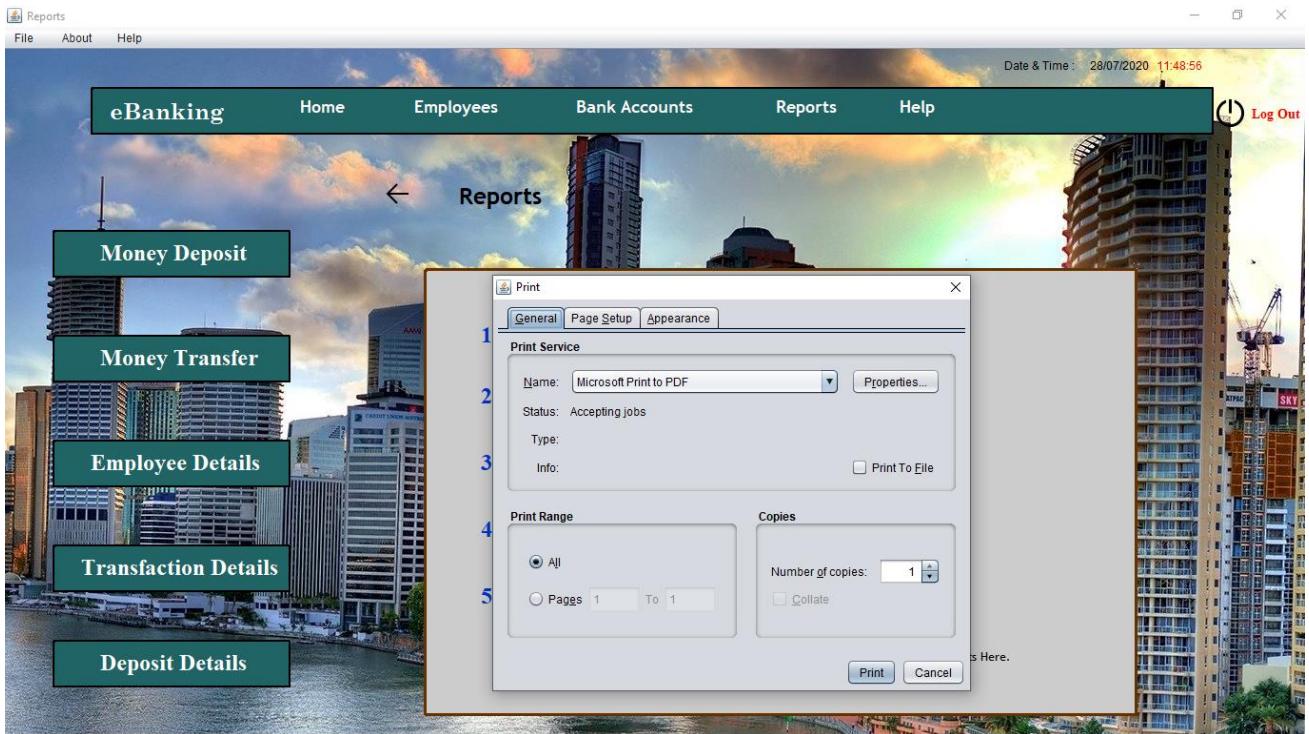
➤ Reports



//All the reports can get here

```
public Reports() {  
    initComponents();  
    Login log = new Login();  
    log.setIcon();  
    setExtendedState(JFrame.MAXIMIZED_BOTH);  
    lblUser.setText(Home.lblUser.getText());  
    date();  
    time();  
}
```

➤ Reports Print



//Reports Print

//Bank Account Details

```
private void btnPrint1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    BankAccountSearch BAS = new BankAccountSearch();
```

```
    BAS.print();
```

```
}
```

//Client Details

```
private void btnPrint2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    ClientAccountSearch CAS = new ClientAccountSearch();
```

```
    CAS.print();
```

```
}
```

//Employee Details

```
private void btnPrint3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    EmployeeAccountSearch EAS = new EmployeeAccountSearch();
```

```
    EAS.print();
```

```
}
```

```
//Deposited Details

private void btnPrint4ActionPerformed(java.awt.event.ActionEvent evt) {

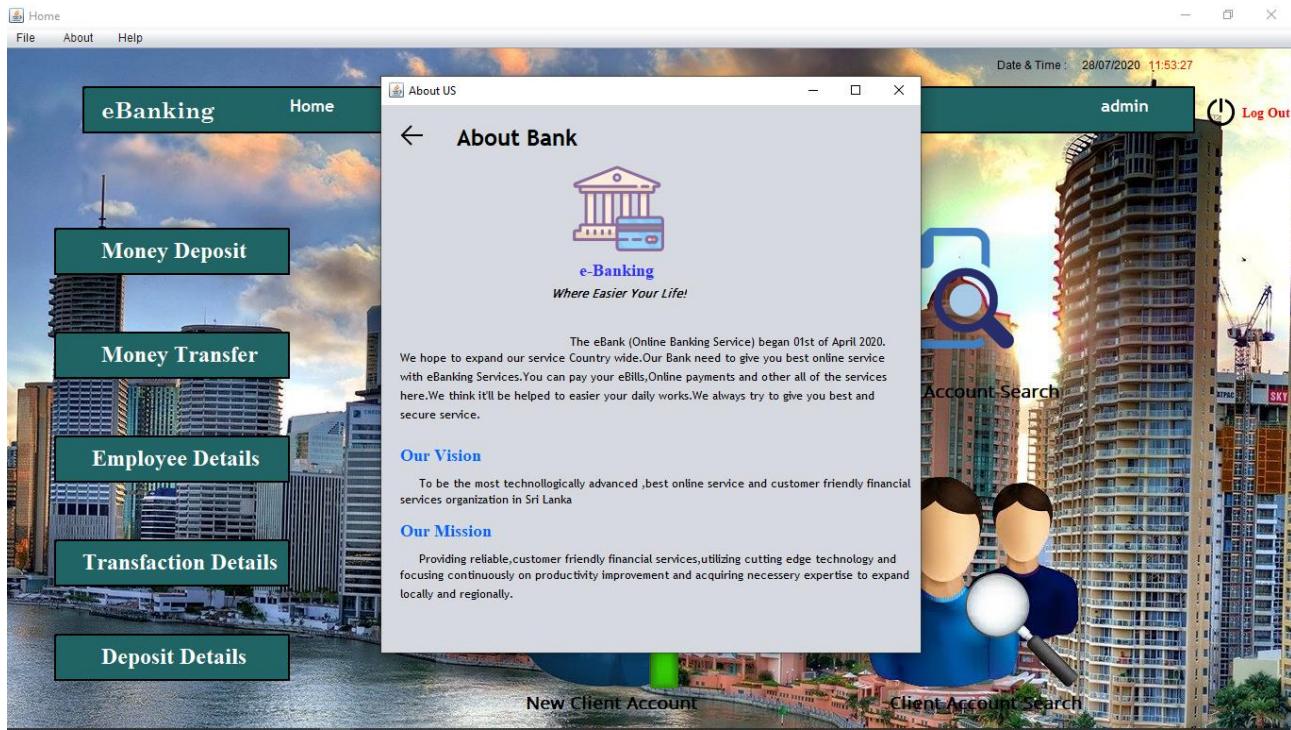
    DepositDetails DAD = new DepositDetails();
    DAD.print();
}
```

```
//Transaction Details

private void btnPrintActionPerformed(java.awt.event.ActionEvent evt) {

    Transfactiondetails TD = new Transfactiondetails();
    TD.print();
}
```

➤ About Bank

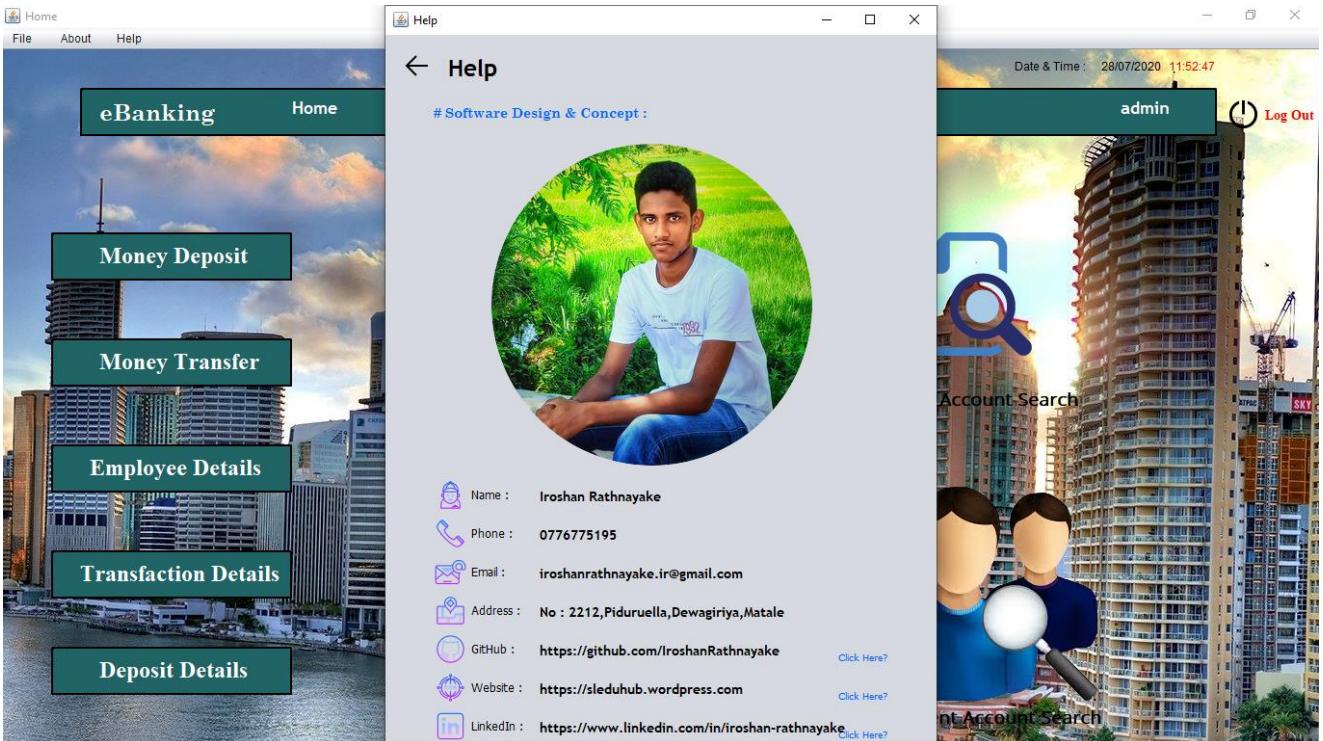


//About Bank

/* You can see here about bank details and also you can see Vision and Mission of the bank. There is when bank was started and the service of the bank.

*/

➤ Help



//Help

/*Here is the developer contacts and all the information.If system has any problem.bank can contact the developer.*/

//gitHub - <https://github.com/IroshanRathnayake>

```
public void gitHub(){  
  
    try {  
  
        String github = "https://github.com/IroshanRathnayake";  
  
        java.awt.Desktop.getDesktop().browse(java.net.URI.create(github));  
  
    } catch (IOException ex) {  
  
        Logger.getLogger(Help.class.getName()).log(Level.SEVERE, null, ex);  
  
    }  
}
```

//website - <https://sleduhub.wordpress.com/>

```
public void website(){  
  
    try {
```

```

        String web = "https://sleduhub.wordpress.com/";
        java.awt.Desktop.getDesktop().browse(java.net.URI.create(web));
    } catch (IOException ex) {
        Logger.getLogger(Help.class.getName()).log(Level.SEVERE, null, ex);
    }
}

//LinkedIn Profile - https://www.linkedin.com/in/iroshan-rathnayake-59773a1b0/
public void linkedIn(){
    try {
        String linkedin = "https://www.linkedin.com/in/iroshan-rathnayake-59773a1b0/";
        java.awt.Desktop.getDesktop().browse(java.net.URI.create(linkedin));
    } catch (IOException ex) {
        Logger.getLogger(Help.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jLabel17MouseClicked(java.awt.event.MouseEvent evt) {
    gitHub();
}

private void jLabel19MouseClicked(java.awt.event.MouseEvent evt) {
    website();
}

private void jLabel5MouseClicked(java.awt.event.MouseEvent evt) {
    linkedIn();
}

}

```

➤ Menu Bar



File

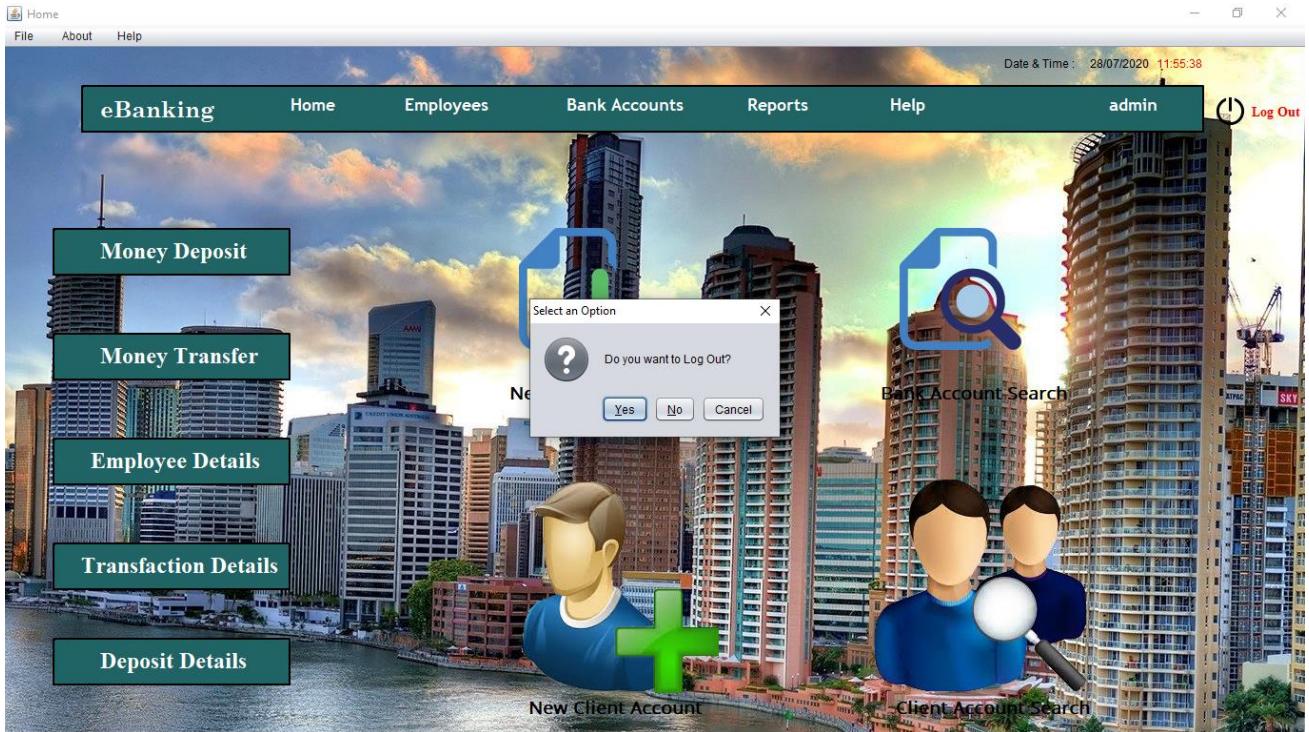


About



Help

➤ Log Out

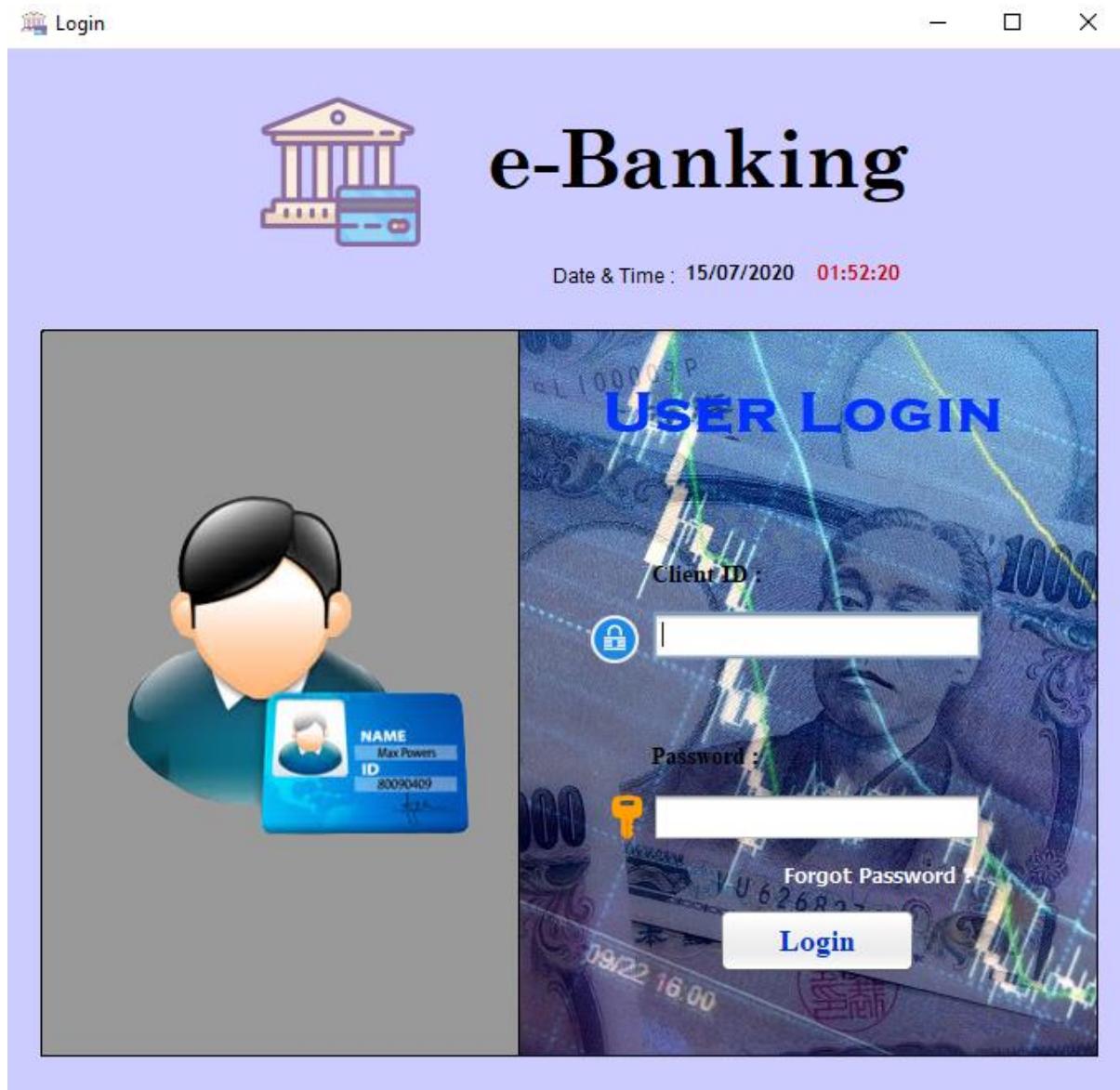


//Log Out

```
private void jLabel10MouseClicked(java.awt.event.MouseEvent evt) {  
    int check = JOptionPane.showConfirmDialog(null, "Do you want to Log  
    Out?");  
  
    if (check == 0){  
        Login log = new Login();  
        log.setVisible(true);  
        this.dispose();  
    }  
}
```

eClient Software

➤ Login



```
//Login

public class Login extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    public Login() {
        initComponents();
    }
```

```

conn =DBconnect.connect();

timeDate();

setIcon();

}

public void login(){

String id =tbClientID.getText();

String pass = tbPass.getText();


try {

String sql = "SELECT * FROM tblClient WHERE Client_ID =? &&
Password=?";

pst = conn.prepareStatement(sql);

pst.setString(1, id);

pst.setString(2, pass);

rs =pst.executeQuery();




if(id.equals("") || pass.equals("")){

JOptionPane.showMessageDialog(null, "Please Enter Your Details!");

}

else if(rs.next()){

Home hm = new Home();

hm.setVisible(true);

this.dispose();

}

else{

JOptionPane.showMessageDialog(null, "Access Denied!");

tbClientID.setText("");

tbPass.setText("");

}

}

```

```

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

//Login Button

private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    login();
}

private void tbClientIDActionPerformed(java.awt.event.ActionEvent evt) {
    login();
}

//Forgot Password

private void lblForgotPassMouseClicked(java.awt.event.MouseEvent evt) {
    String Fid = tbClientID.getText();
    if(tbClientID.getText().equals ""){
        JOptionPane.showMessageDialog(null, "Please Enter your Client ID!");
    }
    else{
        try {
            String sql = "SELECT * FROM tblClient WHERE Client_ID =?";
            pst = conn.prepareStatement(sql);
            pst.setString(1, Fid);
            rs = pst.executeQuery();

            if(rs.next()){
                ForgotPassword fp = new ForgotPassword();
                fp.setVisible(true);
            }
        }
    }
}

```

```
    }

    else{
        JOptionPane.showMessageDialog(null, "Invalid Client ID!");

    }

} catch (Exception e) {

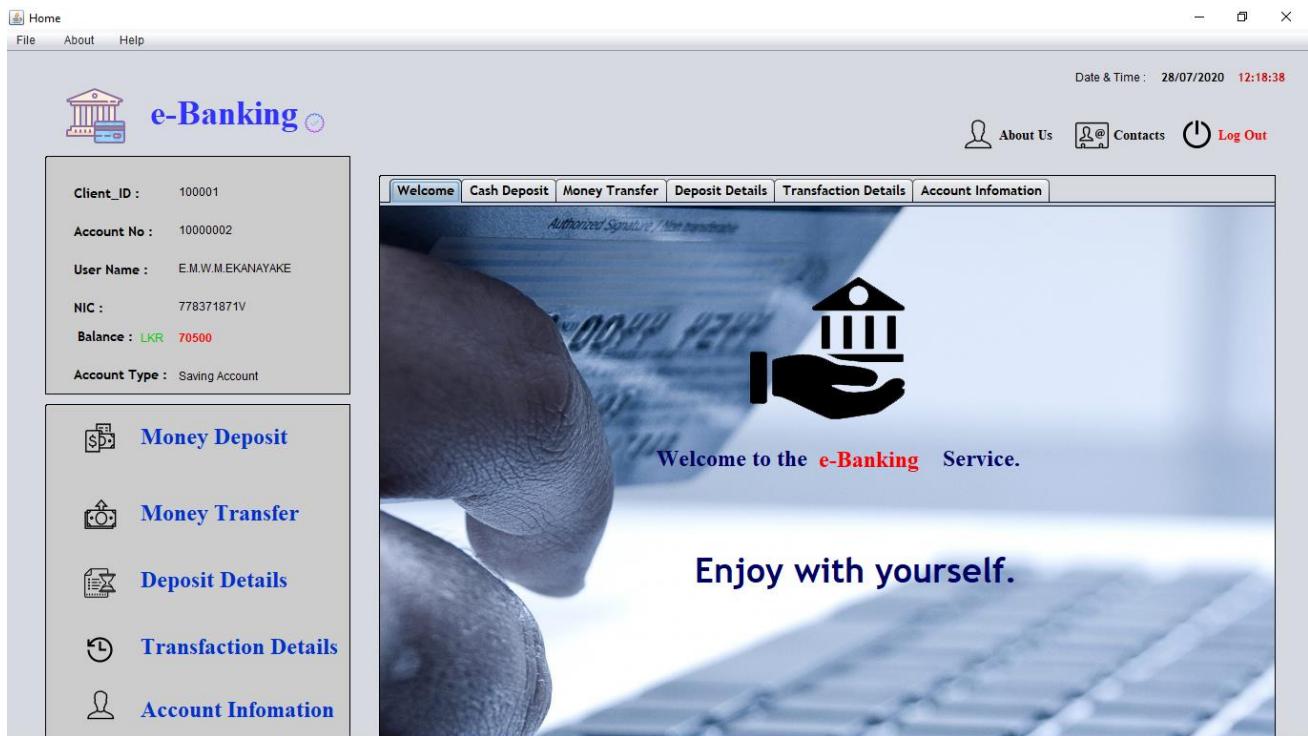
    JOptionPane.showMessageDialog(null, e);

}

}

}
```

➤ Welcome (Home Screen)



/ /Home Screen

```
public class Home extends javax.swing.JFrame{
```

```
    int OTP;  
    String status;  
    String fileName=null;  
    public static final String ACCOUNT_SID =  
        "ACe55609f8867185957887cee60eb9e083";  
    public static final String AUTH_TOKEN =  
        "fe93469492948a5fa1851e63f19febee";
```

```
    Connection conn;  
    PreparedStatement pst;  
    PreparedStatement pst2;  
    PreparedStatement pst3;  
    PreparedStatement pst4;
```

```
PreparedStatement pst5;
ResultSet rs;
progressThread th;
progressThread2 th2;

public Home() {
    initComponents();
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    conn = DBconnect.connect();
    accDetail();
    depositTableLoad();
    transferTableLoad();
    timeDate();
    accInfo();
    btnPrint.hide();
    clear();
}

//Time and Date

public void timeDate(){
    Date d = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    lblDate.setText(sdf.format(d));

    new Timer(0, new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {
```

```

        Date d = new Date();

        SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss");

        lblTime.setText(sdf.format(d));

    }

}{

}.start();

}

//Account Details

public void accDetail(){

    int id = Integer.parseInt(Login.tbClientID.getText());

    try {

        String sql ="SELECT * FROM tblClient WHERE Client_ID = '"+id+"'";

        pst =conn.prepareStatement(sql);

        rs =pst.executeQuery();

        if(rs.next()){

            String ID = rs.getString("Client_ID");

            lblClientID.setText(ID);

            String AccNo = rs.getString("Account_No");

            lblAccNo.setText(AccNo);

            String name = rs.getString("Client_Name");

            lblUname.setText(name);

            String nic = rs.getString("NIC");

            lblNIC.setText(nic);

            String balance = rs.getString("Amount");

            lblBalance.setText(balance);

            tbDCurrentAmount.setText(balance);

            lblAvailabaleBal.setText(balance);

            String acctype = rs.getString("Account_Type");

```

```
lblAccType.setText(acctype);

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

private void jLabel6MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(5);

btnPrint.setEnabled(false);

}

private void jLabel23MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(5);

btnPrint.setEnabled(false);

}

private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(3);

btnPrint.show();

btnPrint.setEnabled(true);

status = "Deposit";

}

private void jLabel22MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(3);

btnPrint.show();

btnPrint.setEnabled(true);

status = "Deposit";

}

private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {
```

```
JTpane.setSelectedIndex(4);

btnPrint.show();
btnPrint.setEnabled(true);
status = "Transfer";
}

private void jLabel24MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(4);

btnPrint.show();
btnPrint.setEnabled(true);
status = "Transfer";
}

private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(1);

btnPrint.setEnabled(false);
}

private void jLabel21MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(2);

btnPrint.setEnabled(false);
}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(1);

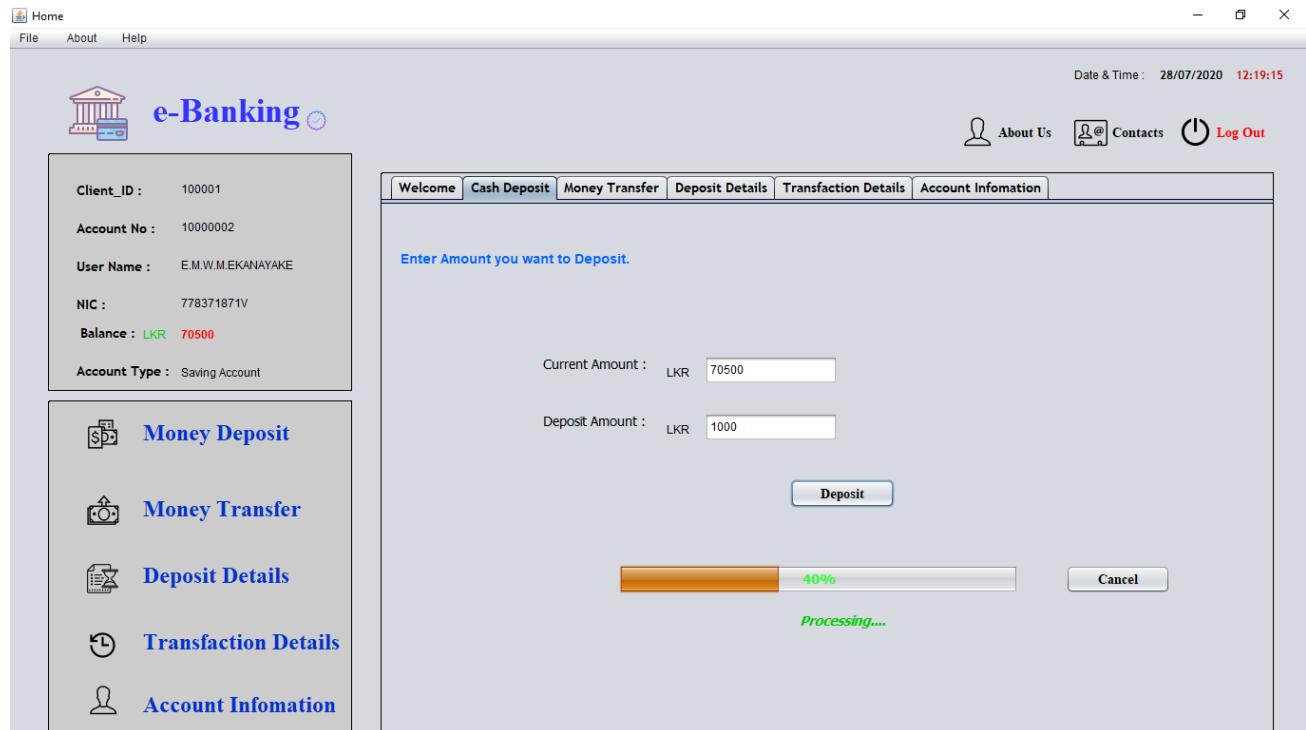
btnPrint.setEnabled(false);
}

private void lblDepositMouseClicked(java.awt.event.MouseEvent evt) {

JTpane.setSelectedIndex(1);

btnPrint.setEnabled(false);
}
```

➤ Deposit Money



//Deposit Money

```
class progressThread extends Thread{  
    JProgressBar proBar;  
    progressThread(JProgressBar proBar){  
        proBar = ProgressBar;  
    }  
    public void run(){  
        int min = 0;  
        int max =10;  
  
        ProgressBar.setMaximum(min);  
        ProgressBar.setMaximum(max);  
        ProgressBar.setValue(0);  
        for(int i=min; i<=max; i++){  
            ProgressBar.setValue(i);  
    }  
}
```

```

try {
    sleep(500);

} catch (InterruptedException ex) {
    Logger.getLogger(Home.class.getName()).log(Level.SEVERE, null, ex);
}

if (i== max){
    JOptionPane.showMessageDialog(null, "Deposited Successfully!");
    ProgressBar.setValue(0);

    accDetail();
    depositTableLoad();
}

try {
    Twilio.init(ACCOUNT_SID, AUTH_TOKEN);

    Message message = Message
        .creator(new PhoneNumber(tbPhone1.getText()), // to
        new PhoneNumber("+12058962320"), // from
        "eBankONLINE \n "+ "Deposit for
        "+tbDepositAmount.getText()+" to " +lblAccNo.getText()+" at
        "+lblTime.getText()+" .Thank You"
        .create();

    System.out.println(message.getSid());
    tbDepositAmount.enable(true);
    tbDepositAmount.setEditable(true);
    clear();
}

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

```

```
        }

    }

}

public void clear(){

    //Account Info

    btnSave.setEnabled(false);

    //Money Deposit

    tbDepositAmount.setText("");

    ProgressBar.setValue(0);

    ProgressBar.hide();

    lblDProgress.hide();

    btnDCancel.hide();

    //Money Transfer

    lblResend.hide();

    lblTimer.hide();

    tbTransferTo.setText("");

    tbTransferUserName.setText("");

    tbTCurrentAmount.setText("");

    tbTransferAmount.setEnabled(true);

    tbTransferAmount.setText("");

    lblOTP.hide();

    tbOTP.setText("");

    tbOTP.hide();

    btnVerify.hide();

    TProgressBar.setValue(0);
```

```

TProgressBar.hide();
lblProgress1.hide();
btnTCancel.hide();
}

//Deposit Button

private void btnDeposit1ActionPerformed(java.awt.event.ActionEvent evt) {
    int check = JOptionPane.showConfirmDialog(null, "Are Details correct?");
    if(check == 0){
        String date = lblDate.getText();
        String time = lblTime.getText();
        String id = lblAccNo.getText();
        String uname = lblUname.getText();
        String nic = lblNIC.getText();
        String CAmount = tbDCurrentAmount.getText();
        String DAmount = tbDepositAmount.getText();
        float value = Float.parseFloat(CAmount) + Float.parseFloat(DAmount);
        try {
            String sql = "UPDATE tblSavingAccount SET Amount='"+value+"'
WHERE Account_No='"+id+"'";
            String SQL2 ="UPDATE tblClient SET Amount='"+value+"' WHERE
Account_No='"+id+"'";
            String SQL3="INSERT INTO tblDeposit
(Date,Time,Account_No,User_Name,NIC,Before_Amount,Deposit_Amou
nt,Current_Amount) VALUES
('"+date+"','"+time+"','"+id+"','"+uname+"','"+nic+"','"+CAmount+"','"+DA
mount+"','"+value+"')";
            pst=conn.prepareStatement(sql);
            pst2 = conn.prepareStatement(SQL2);
            pst3 = conn.prepareStatement(SQL3);
            pst.execute();
            pst2.execute();
        }
    }
}

```

```

        pst3.execute();

        ProgressBar.show();

        lblDProgress.show();

        lblDProgress.setText("Processing....");

        th = new progressThread(ProgressBar);

        th.start();

        btnDCancel.show();

        btnDCancel.enable();

        tbDepositAmount.enable(false);

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

}

//Cancel Button

private void btnDCancelActionPerformed(java.awt.event.ActionEvent evt) {

    th.stop();

    int check = JOptionPane.showConfirmDialog(null, "Do you want to cancel
deposit?");

    if(check == 0){

        String id = lblAccNo.getText();

        String CAmount = tbDCurrentAmount.getText();

        float value = Float.parseFloat(CAmount);

        try {

            String sql = "UPDATE tblSavingAccount SET Amount='"+value+"'
WHERE Account_No='"+id+"';

            String SQL2 = "UPDATE tblClient SET Amount='"+value+"'
WHERE
Account_No='"+id+"';


```

```
String SQL3= "DELETE FROM tblDeposit ORDER BY ID DESC LIMIT 1";

pst=conn.prepareStatement(sql);

pst2 = conn.prepareStatement(SQL2);

pst3 = conn.prepareStatement(SQL3);

pst.execute();

pst2.execute();

pst3.execute();

JOptionPane.showMessageDialog(null, "Deposit Canceled!");

ProgressBar.setValue(0);

ProgressBar.hide();

btnDCancel.enable(false);

btnDCancel.hide();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

tbDepositAmount.setText("");

ProgressBar.setValue(0);

lblDProgress.setText("");

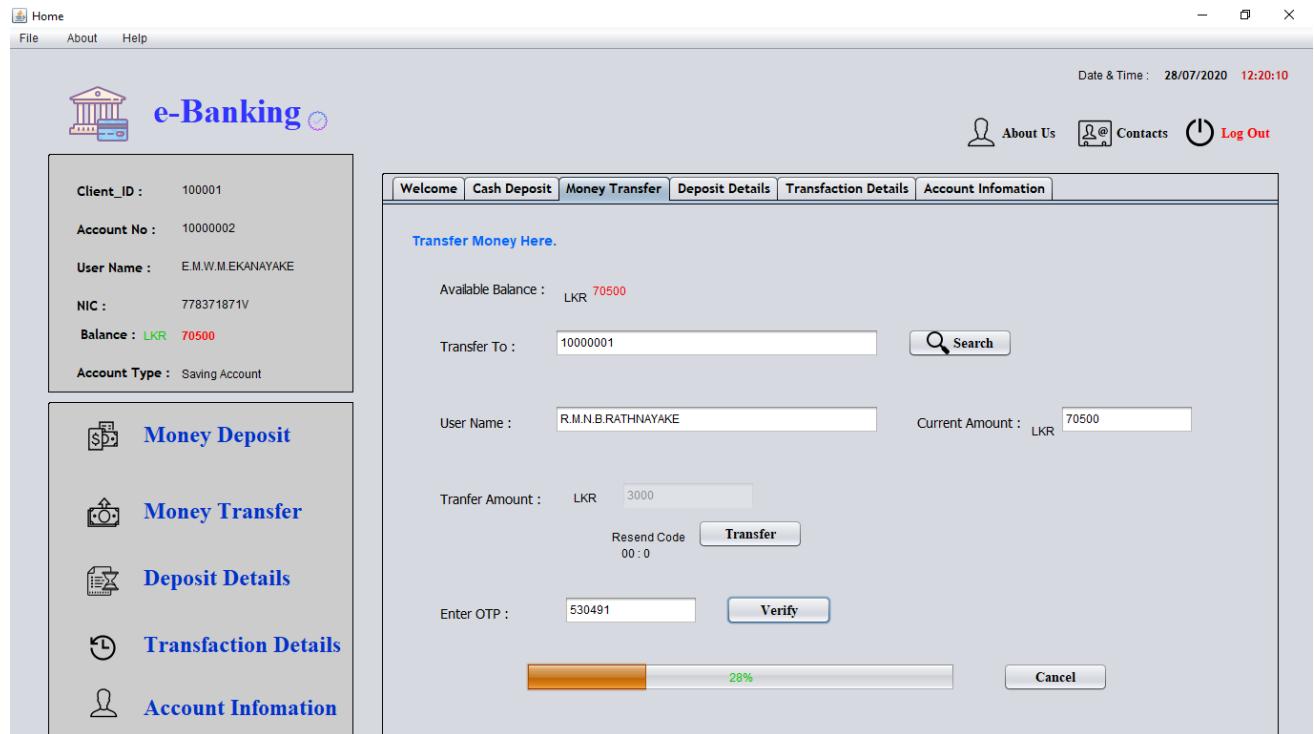
}

accDetail();

tbDepositAmount.enable(true);

}
```

➤ Money Transfer



//Money Transfer

```
class progressThread2 extends Thread{  
    JProgressBar proBar;  
    progressThread2(JProgressBar proBar){  
        proBar = TProgressBar;  
    }  
    public void run(){  
        String Tamount = tbTransferAmount.getText();  
        int min = 0;  
        int max =50;  
        TProgressBar.setMaximum(min);  
        TProgressBar.setMaximum(max);  
        TProgressBar.setValue(0);  
        for(int i=min; i<=max; i++){  
            TProgressBar.setValue(i);  
    }  
}
```

```

try {
    sleep(500);

} catch (InterruptedException ex) {
    Logger.getLogger(Home.class.getName()).log(Level.SEVERE, null, ex);
}

if (i==max){
    JOptionPane.showMessageDialog(null, "Transfaction Completed!");
    ProgressBar.setValue(0);
    lblProgress1.setText("");
}

try {
    Twilio.init(ACCOUNT_SID, AUTH_TOKEN);

    Message message = Message
        .creator(new PhoneNumber(tbPhone1.getText()), // to
new PhoneNumber("+12058962320"), // from
"eBankONLINE \nTransfer to "+tbTransferTo.getText()+" on
for LKR " +tbTransferAmount.getText()+" on
"+lblDate.getText()+" "+lblTime.getText()+" .Thank You")
        .create();

    System.out.println(message.getSid());
    tbDepositAmount.enable(true);
    tbDepositAmount.setEditable(true);
    clear();
}

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

accDetail();
transferTableLoad();

```

```

        clear();

    }

}

}

}

//Search Account

public void searchAcc(){

    String srch = tbTransferTo.getText();

    String check = lblAccNo.getText();

    if(srch.equals(check)){

        JOptionPane.showMessageDialog(null, "Same Accounts.Plese check
Account Numbers");

        tbTransferTo.setText("");

    }

    else{

        try {

            String sql = "SELECT * FROM tblSavingAccount WHERE
Account_No='"+srch+"'";

            pst = conn.prepareStatement(sql);

            rs = pst.executeQuery();

            if (rs.next()){

                String name = rs.getString("User_Name");

                tbTransferUserName.setText(name);

                String CAmount = rs.getString("Amount");

                tbTCurrentAmount.setText(CAmount);

                rs.close();

                pst.close();

            }

        }

    }

}

```

```

else{
    JOptionPane.showMessageDialog(null, "Invalid Input!");
}

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

}

}

}

}

//Timer

@SuppressWarnings("unchecked")

int i = 30;

Timer timer = new Timer(1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        i--;
        if(i>=0){
            lblTimer.setText("00 : "+i);
            if(i==0){
                btnTransfer.setEnabled(true);
            }
        }
    }
});

//Search Button

private void btnTransfer3ActionPerformed(java.awt.event.ActionEvent evt) {
    searchAcc();
}

```

```

//Transfer Button

private void btnTransferActionPerformed(java.awt.event.ActionEvent evt) {

    String TT = tbTransferTo.getText();
    String TA = tbTransferAmount.getText();
    String phone = tbPhone1.getText();

    if(TT.equals("") || TA.equals("") || phone.equals("")){

        JOptionPane.showMessageDialog(null, "Please fill all the details!");

    }

    else{

        lblResend.show();
        lblTimer.show();
        timer.start();
        tbTransferAmount.setEnabled(false);
        tbPhone1.setEnabled(false);
        lblOTP.show();
        tbOTP.show();
        btnVerify.show();
        btnTransfer.setEnabled(false);

    }

    try {

        Twilio.init(ACCOUNT_SID, AUTH_TOKEN);
        Random rand = new Random();
        OTP =rand.nextInt(999999);
        Message message = Message
            .creator(new PhoneNumber(phone), // to
                    new PhoneNumber("+12058962320"), // from
                    "eBankONLINE \nPlease Enter OTP : "+OTP+" to complete your online
                    transfection.Thank You")
    }
}

```

```

    .create();

    JOptionPane.showMessageDialog(null, "OTP sent successfully!");

    System.out.println(message.getSid());

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);

}

}

}

//Verify Button

private void btnVerifyActionPerformed(java.awt.event.ActionEvent evt) {

    String TT = tbTransferTo.getText();

    String TA = tbTransferAmount.getText();

    String phone = tbPhone1.getText();

    if(TT.equals("") || TA.equals("") || phone.equals("")){

        JOptionPane.showMessageDialog(null, "Please fill all the details!");

    }

    else{

        int check = JOptionPane.showConfirmDialog(null, "Are you sure?");

        if(check == 0){

            String date = lblDate.getText();

            String time = lblTime.getText();

            String Aid = lblAccNo.getText();

            String Auname = lblUname.getText();

            String Aamount = lblAvailabaleBal.getText();

            String Bid = tbTransferTo.getText();

            String Buname = tbTransferUserName.getText();

            String Bamount = tbTCurrentAmount.getText();

```

```

String Tamount = tbTransferAmount.getText();

float MinValue = Float.parseFloat(Aamount)-Float.parseFloat(Tamount);

float AddValue = Float.parseFloat(Bamount)+Float.parseFloat(Tamount);

if(Float.parseFloat(Aamount)<= Float.parseFloat(Tamount) &&
Float.parseFloat(Aamount) <= 1500 && Float.parseFloat(Tamount) -
Float.parseFloat(Aamount) <=1500 ){

    JOptionPane.showMessageDialog(null, "Insufficient Balance!");

}

else{

    try {

        String sql = "UPDATE tblSavingAccount SET
        Amount='"+MinValue+"' WHERE Account_No='"+Aid+"' ";

        String SQL2 = "UPDATE tblSavingAccount SET
        Amount='"+AddValue+"' WHERE Account_No='"+Bid+"'";

        String SQL3 = "UPDATE tblClient SET Amount='"+MinValue+"'
        WHERE Account_No='"+Aid+"' ";

        String SQL4 = "UPDATE tblClient SET Amount='"+AddValue+"'
        WHERE Account_No='"+Bid+"'";

        String SQL5 ="INSERT INTO tblMoneyTransfer
        (Date,Time,From_Acc,From_User_Name,FFBefore_Amount,To_Acc
        ,To_User_Name,TBefore_Amount,Transferred_Amount) VALUES
        ('"+date+"','"+time+"','"+Aid+"','"+AUname+"','"+Aamount+"','"+Bid
        +"','"+BUname+"','"+Bamount+"','"+Tamount+"')";

        pst = conn.prepareStatement(sql);

        pst2 =conn.prepareStatement(SQL2);

        pst3 = conn.prepareStatement(SQL3);

        pst4 = conn.prepareStatement(SQL4);

        pst5 = conn.prepareStatement(SQL5);

        pst.execute();

        pst2.execute();
    }
}

```



```
try {

    String sql = "UPDATE tblSavingAccount SET
    Amount='"+MinValue+"' WHERE Account_No='"+Aid+"' ";
    String SQL2 = "UPDATE tblSavingAccount SET
    Amount='"+AddValue+"' WHERE Account_No='"+Bid+"' ";

    pst = conn.prepareStatement(sql);
    pst2 =conn.prepareStatement(SQL2);
    pst.execute();
    pst2.execute();
    JOptionPane.showMessageDialog(null, "Tranfaction Canceled!");
    ProgressBar.setValue(0);

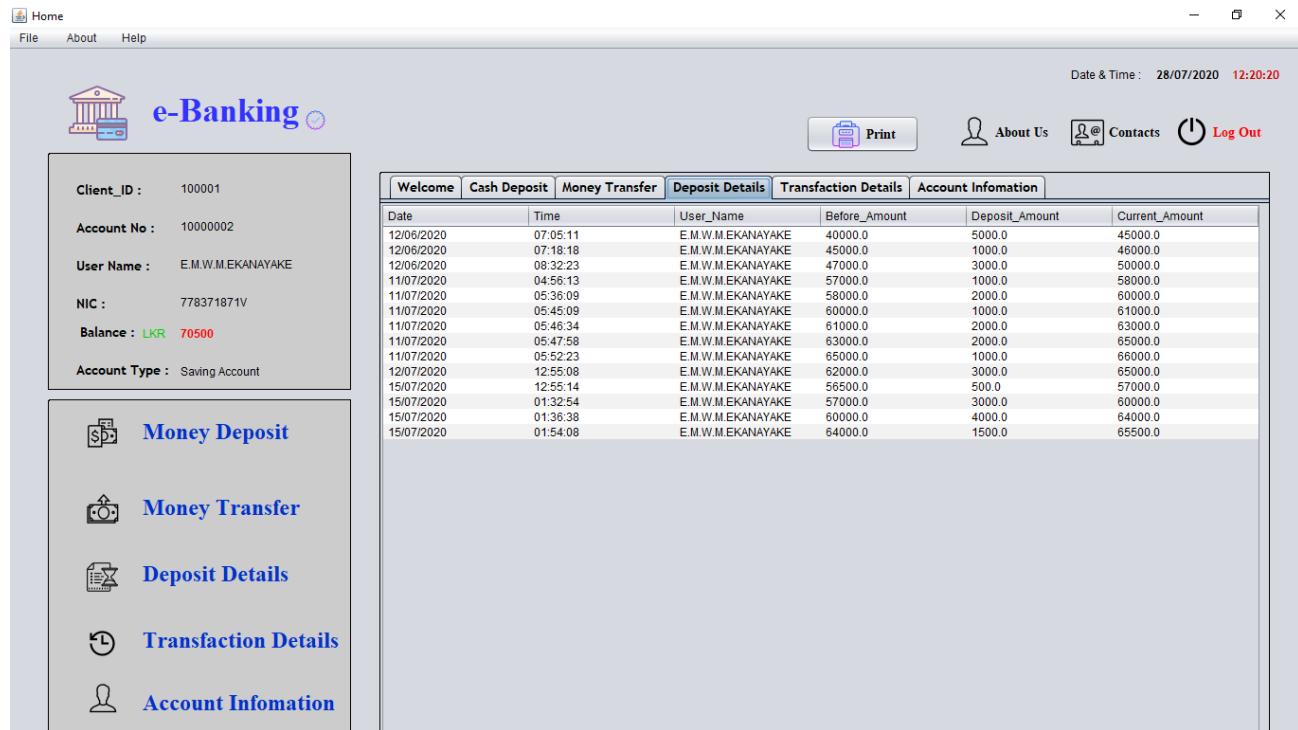
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}

clear();
lblProgress1.setText("");
btnTCancel.enable(false);
btnTCancel.hide();

clear();
}

}
```

➤ Deposited Details



The screenshot shows the e-Banking application interface. The main content area displays a table of transaction details. The table has columns: Date, Time, User_Name, Before_Amount, Deposit_Amount, and Current_Amount. The data in the table is as follows:

Date	Time	User_Name	Before_Amount	Deposit_Amount	Current_Amount
12/06/2020	07:05:11	E.M.W.M.EKANAYAKE	40000.0	5000.0	45000.0
12/06/2020	07:18:18	E.M.W.M.EKANAYAKE	45000.0	1000.0	46000.0
12/06/2020	08:32:23	E.M.W.M.EKANAYAKE	47000.0	3000.0	50000.0
11/07/2020	04:56:13	E.M.W.M.EKANAYAKE	57000.0	1000.0	58000.0
11/07/2020	05:36:09	E.M.W.M.EKANAYAKE	58000.0	2000.0	60000.0
11/07/2020	05:45:09	E.M.W.M.EKANAYAKE	60000.0	1000.0	61000.0
11/07/2020	05:46:34	E.M.W.M.EKANAYAKE	61000.0	2000.0	63000.0
11/07/2020	05:47:58	E.M.W.M.EKANAYAKE	63000.0	2000.0	65000.0
11/07/2020	05:52:23	E.M.W.M.EKANAYAKE	65000.0	1000.0	66000.0
12/07/2020	12:55:08	E.M.W.M.EKANAYAKE	62000.0	3000.0	65000.0
15/07/2020	12:55:14	E.M.W.M.EKANAYAKE	56500.0	500.0	57000.0
15/07/2020	01:32:54	E.M.W.M.EKANAYAKE	57000.0	3000.0	60000.0
15/07/2020	01:36:38	E.M.W.M.EKANAYAKE	60000.0	4000.0	64000.0
15/07/2020	01:54:08	E.M.W.M.EKANAYAKE	64000.0	1500.0	65500.0

The sidebar on the left contains links for Money Deposit, Money Transfer, Deposit Details, Transaction Details, and Account Information.

//Deposited Details

```
public void depositTableLoad(){

    int id = Integer.parseInt(lblAccNo.getText());

    try {

        String sql = "SELECT
Date,Time,User_Name,Before_Amount,Deposit_Amount,Current_Amount FROM
tblDeposit WHERE Account_No='"+id+"';

        pst = conn.prepareStatement(sql);

        rs = pst.executeQuery();

        tblDeposit.setModel(DbUtils.resultSetToTableModel(rs));

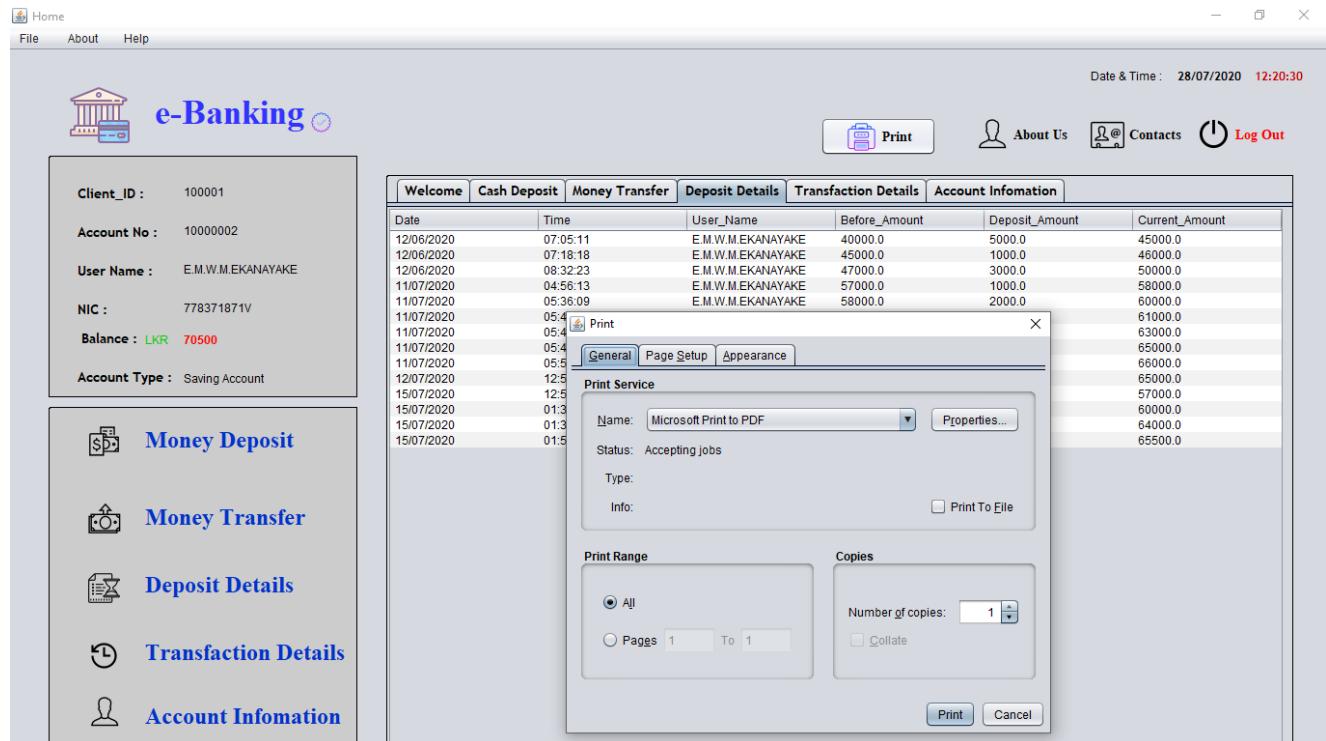
    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}
```

➤ Deposited Details Print

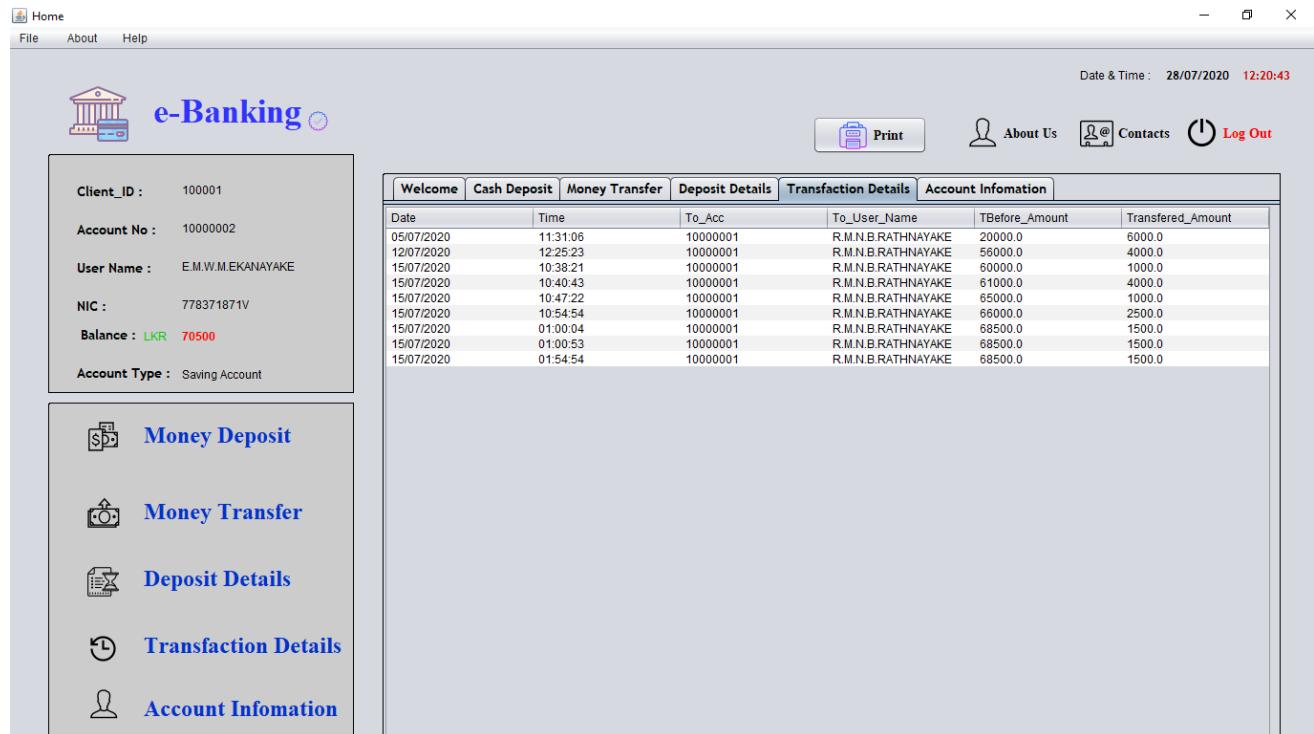


//Deposited Details Print

```
private void btnPrintActionPerformed(java.awt.event.ActionEvent evt) {
    if(status.equals("Deposit")){
        MessageFormat header = new MessageFormat("Deposit Report");
        MessageFormat footer = new MessageFormat("-eBankONLINE-");
        try {
            tblDeposit.print(JTable.PrintMode.NORMAL, header, footer);
        } catch (java.awt.print.PrinterException e) {
            System.err.format("Cannot Print", e.getMessage());
        }
    }
    else if (status.equals("Transfer")){
        MessageFormat header = new MessageFormat("Money Transfaction Report");
        MessageFormat footer = new MessageFormat("-eBankONLINE-");
    }
}
```

```
try {  
    tblTransfer.print(JTable.PrintMode.NORMAL, header, footer);  
}  
catch (java.awt.print.PrinterException e) {  
    System.err.format("Cannot Print", e.getMessage());  
}  
}  
}  
}
```

➤ Transaction Details



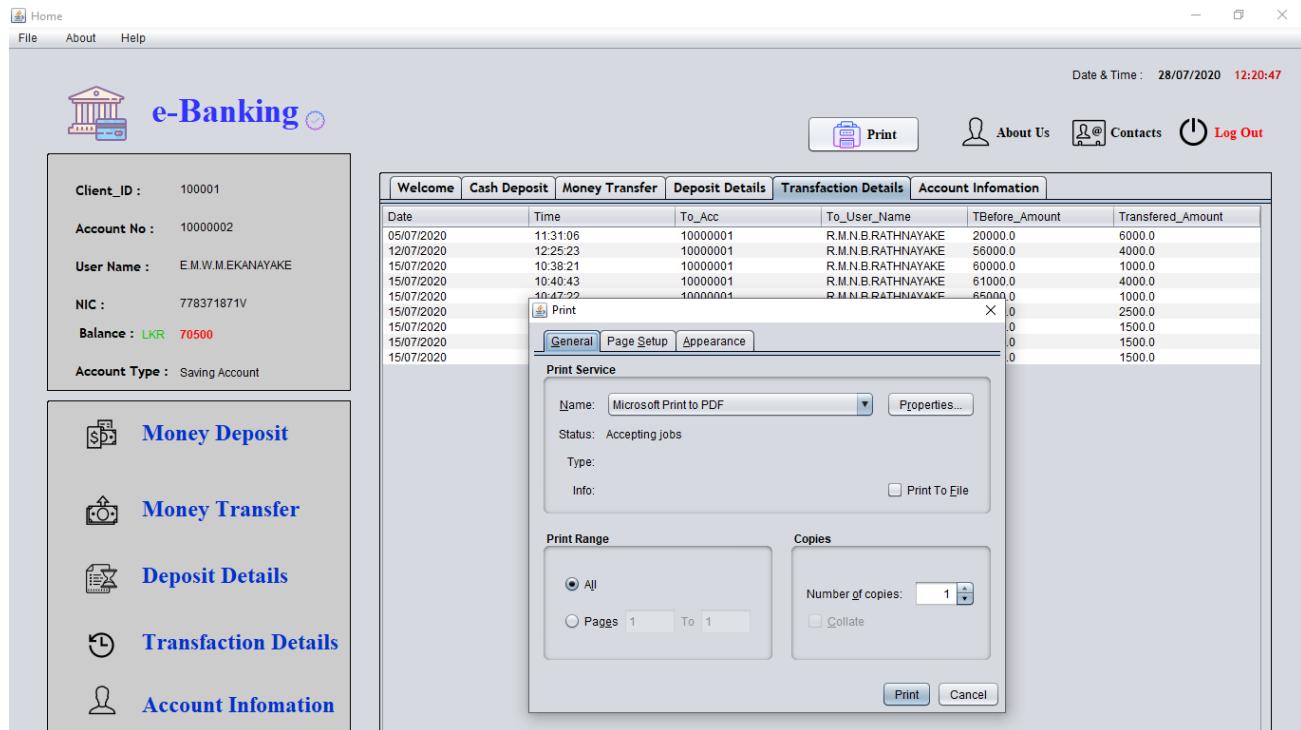
The screenshot shows the e-Banking application interface. The top navigation bar includes 'Home', 'File', 'About', and 'Help' on the left, and 'Date & Time : 28/07/2020 12:20:43' on the right. The main content area features a sidebar on the left with icons for 'Money Deposit', 'Money Transfer', 'Deposit Details', 'Transaction Details', and 'Account Information'. The main panel displays a table of transaction history with columns: Date, Time, To_Acc, To_User_Name, TBefore_Amount, and Transferred_Amount. The table data is as follows:

Date	Time	To_Acc	To_User_Name	TBefore_Amount	Transferred_Amount
05/07/2020	11:31:06	10000001	R.M.N.B.RATHNAYAKE	20000.0	6000.0
12/07/2020	12:25:23	10000001	R.M.N.B.RATHNAYAKE	56000.0	4000.0
15/07/2020	10:38:21	10000001	R.M.N.B.RATHNAYAKE	60000.0	1000.0
15/07/2020	10:40:43	10000001	R.M.N.B.RATHNAYAKE	61000.0	4000.0
15/07/2020	10:47:22	10000001	R.M.N.B.RATHNAYAKE	65000.0	1000.0
15/07/2020	10:54:54	10000001	R.M.N.B.RATHNAYAKE	66000.0	2500.0
15/07/2020	01:00:04	10000001	R.M.N.B.RATHNAYAKE	68500.0	1500.0
15/07/2020	01:00:53	10000001	R.M.N.B.RATHNAYAKE	68500.0	1500.0
15/07/2020	01:54:54	10000001	R.M.N.B.RATHNAYAKE	68500.0	1500.0

//Transaction Details

```
public void transferTableLoad(){
    int id = Integer.parseInt(lblAccNo.getText());
    try {
        String sql = "SELECT
Date,Time,To_Acc,To_User_Name,TBefore_Amount,Transferred_Amount FROM
tblMoneyTransfer WHERE From_Acc='"+id+"'";
        pst = conn.prepareStatement(sql);
        rs = pst.executeQuery();
        tblTransfer.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

➤ Transaction Details Print



//Transaction Details Print

```

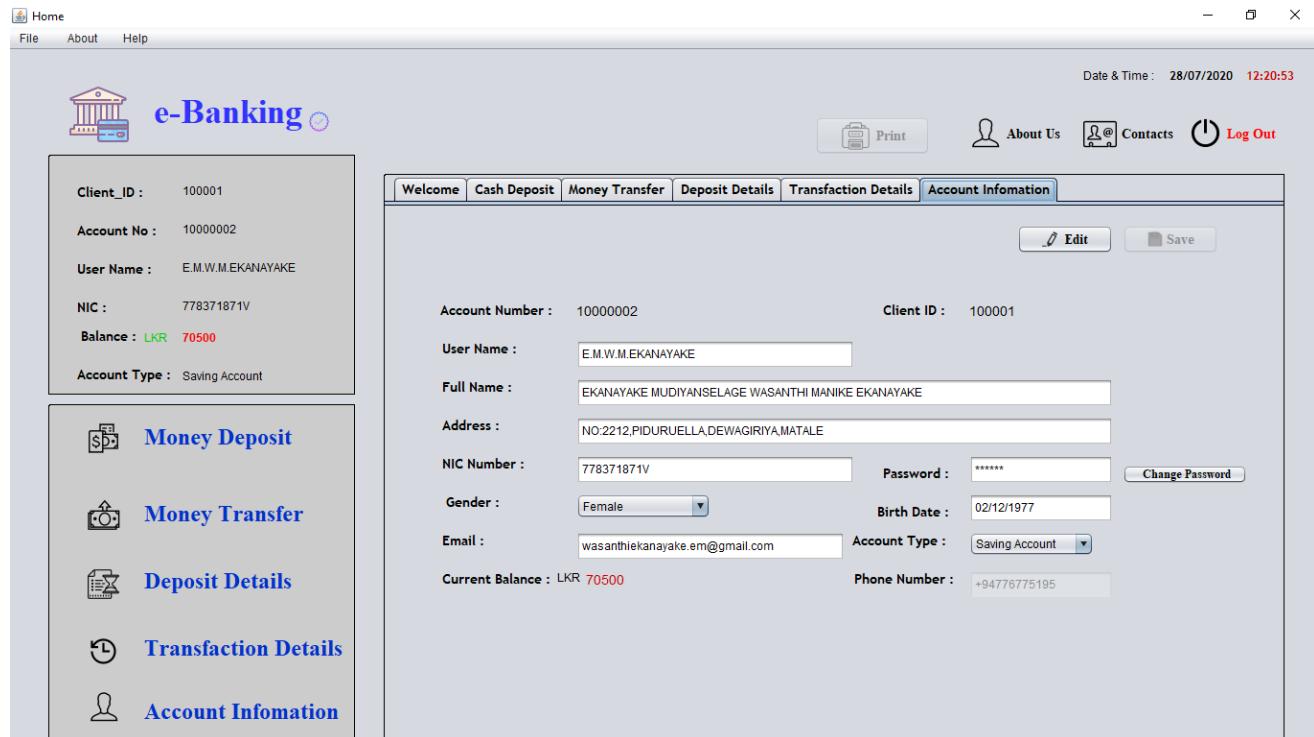
private void btnPrintActionPerformed(java.awt.event.ActionEvent evt) {
    if(status.equals("Deposit")){
        MessageFormat header = new MessageFormat("Deposit Report");
        MessageFormat footer = new MessageFormat("-eBankONLINE-");
        try {
            tblDeposit.print(JTable.PrintMode.NORMAL, header, footer);
        } catch (java.awt.print.PrinterException e) {
            System.err.format("Cannot Print", e.getMessage());
        }
    }
    else if (status.equals("Transfer")){
        MessageFormat header = new MessageFormat("Money Transfaction
Report");
    }
}

```

```
MessageFormat footer = new MessageFormat("-eBankONLINE-");

try {
    tblTransfer.print(JTable.PrintMode.NORMAL, header, footer);
} catch (java.awt.print.PrinterException e) {
    System.err.format("Cannot Print", e.getMessage());
}
}
```

➤ Account Information



//Account Information

```
public void accInfo(){
    int id = Integer.parseInt(Login.tbClientID.getText());

    try {
        String sql ="SELECT * FROM tblClient WHERE Client_ID = '"+id+"'";
        pst =conn.prepareStatement(sql);
        rs =pst.executeQuery();

        if(rs.next()){
            String ID = rs.getString("Client_ID");
            lblCliID.setText(ID);

            String AccNo = rs.getString("Account_No");
            lblAccountNo.setText(AccNo);

            String name = rs.getString("Client_Name");
        }
    }
}
```

```

        tbUname.setText(name);

        String fname = rs.getString("FullName");

        tbFName.setText(fname);

        String address = rs.getString("Address");

        tbAddress.setText(address);

        String nic = rs.getString("NIC");

        tbNIC.setText(nic);

        String gen = rs.getString("Gender");

        cbGender.setSelectedItem(gen);

        String bd = rs.getString("BD");

        tbBD.setText(bd);

        String email = rs.getString("Email");

        tbEmail.setText(email);

        String acctype = rs.getString("Account_Type");

        cbAccType.setSelectedItem(acctype);

        String phone = rs.getString("Phone");

        tbPhone1.setText(phone);

        String balance = rs.getString("Amount");

        lblBalance1.setText(balance);

    }

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);

}

//Edit Button

private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {

    tbUname.setEditable(true);

    tbFName.setEditable(true);

    tbAddress.setEditable(true);

```

```

tbBD.setEditable(true);
tbEmail.setEditable(true);
tbBD.setEditable(true);
btnSave.setEnabled(true);

}

//Save Button
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {

    int id = Integer.parseInt(lblClientID.getText());

    String uname = tbUname.getText();
    String Fname = tbFName.getText();
    String address = tbAddress.getText();
    String nic = tbNIC.getText();
    String bd = tbBD.getText();
    String gen = cbGender.getSelectedItem().toString();
    String acctype = cbAccType.getSelectedItem().toString();
    String email = tbEmail.getText();
    String phone = tbPhone1.getText();

    try {

        String sql = "UPDATE tblClient SET
Client_Name='"+uname+"',FullName='"+Fname+"',NIC='"+nic+"',Addres
s='"+address+"',BD='"+bd+"',Gender='"+gen+"',Account_Type='"+acctyp
e+"',Email='"+email+"',Phone='"+phone+"' WHERE Client_ID = '"+id+"';

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

        JOptionPane.showMessageDialog(null, "Account Updated Successfully!");
    }
}

```

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

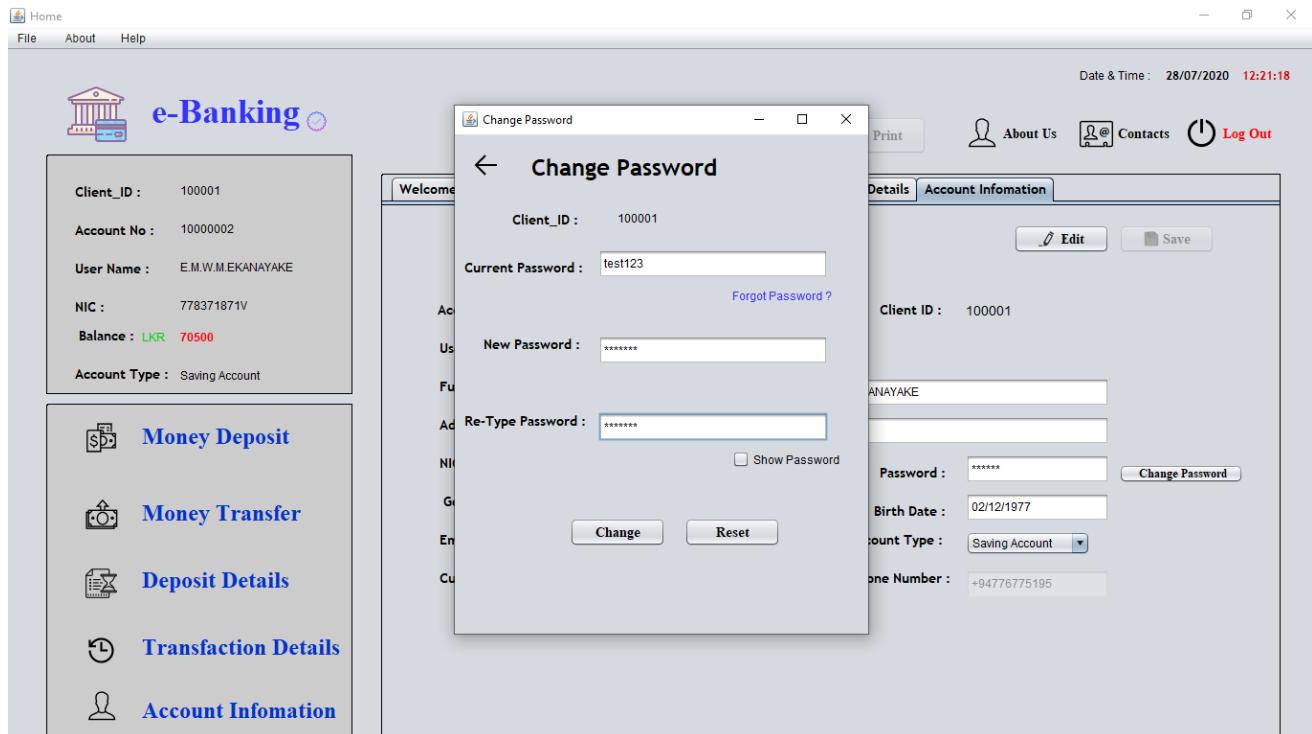
        tbUname.setEditable(false);
        tbFName.setEditable(false);
        tbAddress.setEditable(false);
        tbBD.setEditable(false);
        tbEmail.setEditable(false);
        tbBD.setEditable(false);
        btnSave.setEnabled(false);

    }

//Change Passwod

private void btnChangePassActionPerformed(java.awt.event.ActionEvent evt) {
    ChangePassword cp = new ChangePassword();
    cp.setVisible(true);
}
```

➤ Change Password



//Change Password

```
public class ChangePassword extends javax.swing.JFrame {
```

```
    Connection conn;
```

```
    PreparedStatement pst;
```

```
    PreparedStatement pst2;
```

```
    ResultSet rs;
```

```
    public ChangePassword() {
```

```
        initComponents();
```

```
        conn = DBconnect.connect();
```

```
        lblClientID.setText(Login.tbClientID.getText());
```

```
    }
```

```
    public void reset(){
```

```
        tbCurrentPass.setText("");
```

```
        tbNewPass.setText("");
```

```
        tbReNewPass.setText("");
```

```
        lblWrong.hide();
```

```

    }

//Show Password

private void cbShowActionPerformed(java.awt.event.ActionEvent evt) {

    if(cbShow.isSelected()){

        tbNewPass.setEchoChar((char)0);

        tbReNewPass.setEchoChar((char)0);

    }

    else{

        tbNewPass.setEchoChar('*');

        tbReNewPass.setEchoChar('*');

    }

}

//Change Button

private void btnChangeActionPerformed(java.awt.event.ActionEvent evt) {

    String id = lblClientID.getText();

    String cupass = tbCurrentPass.getText();

    String newpass = tbNewPass.getText();

    String retype = tbReNewPass.getText();




    try {

        String sql = "SELECT * FROM tblClient WHERE Client_ID=? &&
        Password = ?";

        pst = conn.prepareStatement(sql);

        pst.setString(1, id);

        pst.setString(2, cupass);

        rs = pst.executeQuery();




        if(rs.next()){

            if(newpass.equals(retype)){


```

```

String SQL2 ="UPDATE tblClient SET Password="""+newpass+"""
WHERE Client_ID = """+id+"""";

pst = conn.prepareStatement(SQL2);

pst.execute();

JOptionPane.showMessageDialog(null, "Password Changed
Successfully!");

reset();

}

else{

lblWrong.show();

lblWrong.setText("Password does not match,Please Retry!");

}

}

else{

JOptionPane.showMessageDialog(null, "Current Password incorrect!");

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

}

//Forgot Password

private void lblForgotPassMouseClicked(java.awt.event.MouseEvent evt) {

ForgotPassword fp = new ForgotPassword();

fp.setVisible(true);

this.dispose();

}

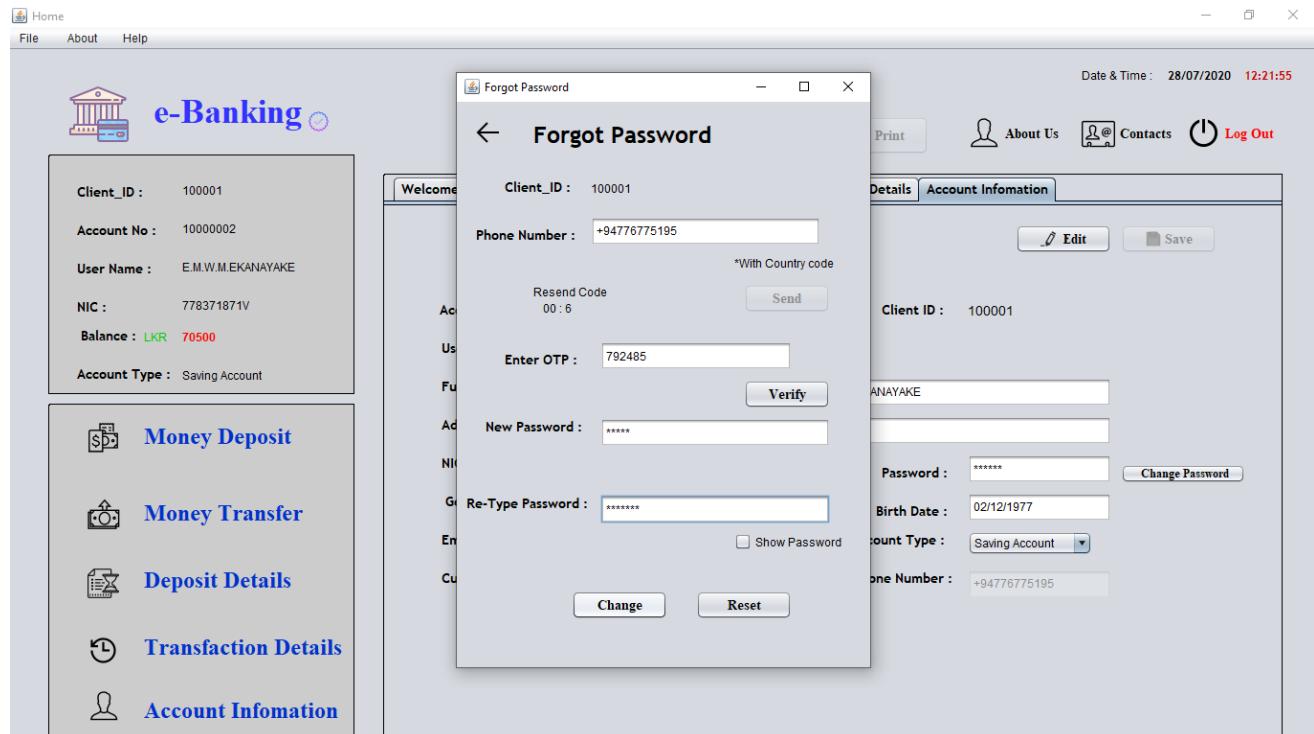
private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {

reset();

}

```

➤ Forgot Password



//Forgot Password

```
public class ForgotPassword extends javax.swing.JFrame {
```

```
    Connection conn;
```

```
    PreparedStatement pst;
```

```
    ResultSet rs;
```

```
    int OTP;
```

```
    public static final String ACCOUNT_SID =  
        "ACe55609f8867185957887cee60eb9e083";
```

```
    public static final String AUTH_TOKEN =  
        "fe93469492948a5fa1851e63f19febee";
```

```
    public ForgotPassword() {
```

```
        initComponents();
```

```
conn = DBconnect.connect();
reset();
lblClientID.setText(Login.tbClientID.getText());

}

public void reset (){
    lblTimer.hide();
    lblResend.hide();

    lblOTP.hide();
    tbVerifyOTP.hide();
    btnVerify.hide();

    lblnewpass.hide();
    lblrenewpass.hide();
    tbNewPass.hide();
    tbReNewPass.hide();
    lblWrong.hide();
    cbShow.hide();
    btnChange.hide();
    btnReset.hide();
}

public void OTPSend(){
    try {
        Twilio.init(ACCOUNT_SID, AUTH_TOKEN);
        Random rand = new Random();
        OTP =rand.nextInt(999999);
    }
}
```

```

Message message = Message
    .creator(new PhoneNumber(tbPhone.getText()), // to
    new PhoneNumber("+12058962320"), // from
    "eBankONLINE \n " + " Please Enter OTP : "+OTP+ " to complete you
    online request.Thank You")
    .create();

JOptionPane.showMessageDialog(null, "OTP sent successfully!");
System.out.println(message.getId());
} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);
}

}

//Timer
@SuppressWarnings("unchecked")
int i = 30;
Timer timer = new Timer(1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        i--;
        if(i>=0){
            lblTimer.setText("00 : "+i);
            if(i==0){
                btnSend.setEnabled(true);
            }
        }
    }
});

```

```

//Send Button

private void btnSendActionPerformed(java.awt.event.ActionEvent evt) {

    String id = lblClientID.getText();
    String cupass = tbPhone.getText();

    if(tbPhone.equals("+94")){
        JOptionPane.showMessageDialog(null, "Please enter your phone
Number!");
    }
    else{
        OTPSend();
        btnSend.setEnabled(false);

        lblTimer.show();
        lblResend.show();
        timer.start();
        lblOTP.show();
        tbVerifyOTP.show();
        btnVerify.show();
    }
}

//Verify Button

private void btnVerifyActionPerformed(java.awt.event.ActionEvent evt) {

    int verify = Integer.parseInt(tbVerifyOTP.getText());

    if(tbVerifyOTP.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Please enter the OTP!");
    }
}

```

```

else if (verify == OTP){

    JOptionPane.showMessageDialog(null, "Change Password here!");

    lblnewpass.show();

    lblrenewpass.show();

    tbNewPass.show();

    tbReNewPass.show();

    cbShow.show();

    btnChange.show();

    btnReset.show();

}

else{

    JOptionPane.showMessageDialog(null, "Incorrect OTP!");

    tbVerifyOTP.setText("");

}

//Password Show

private void cbShowActionPerformed(java.awt.event.ActionEvent evt) {

    if(cbShow.isSelected()){

        tbNewPass.setEchoChar((char)0);

        tbReNewPass.setEchoChar((char)0);

    }

    else{

        tbNewPass.setEchoChar('*');

        tbReNewPass.setEchoChar('*');

    }

}

```

```

//Change Button

private void btnChangeActionPerformed(java.awt.event.ActionEvent evt) {

    String id = lblClientID.getText();
    String newpass = tbNewPass.getText();
    String retype = tbReNewPass.getText();

    try {

        if(newpass.equals(retype)){

            String SQL2 ="UPDATE tblClient SET Password='"+newpass+"'
WHERE Client_ID = '"+id+"'";

            pst = conn.prepareStatement(SQL2);
            pst.execute();

            JOptionPane.showMessageDialog(null, "Password Changed
Successfully!");

            reset();
            this.dispose();

        }

        else{

            lblWrong.show();
            lblWrong.setText("Password does not match,Please Retry!");

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

}

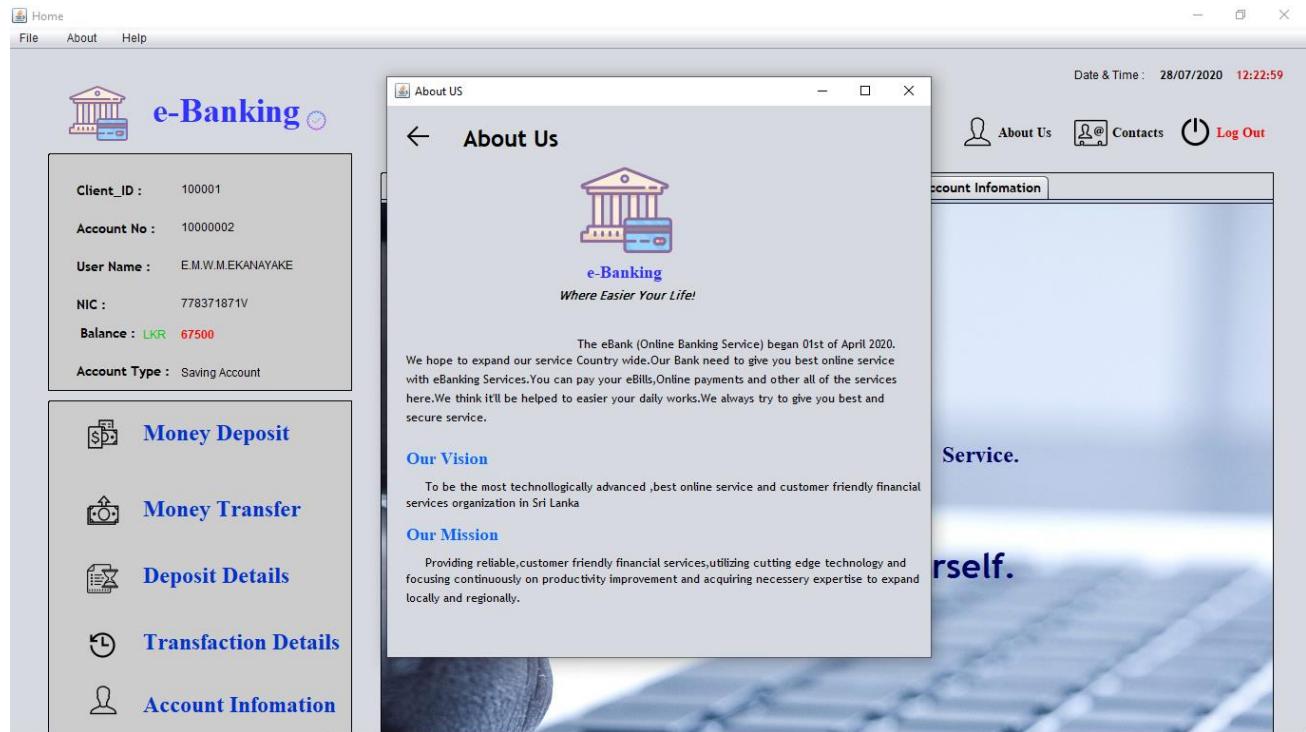
private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {

    tbNewPass.setText("");
    tbReNewPass.setText("");
    lblWrong.hide();

}

```

➤ About Us

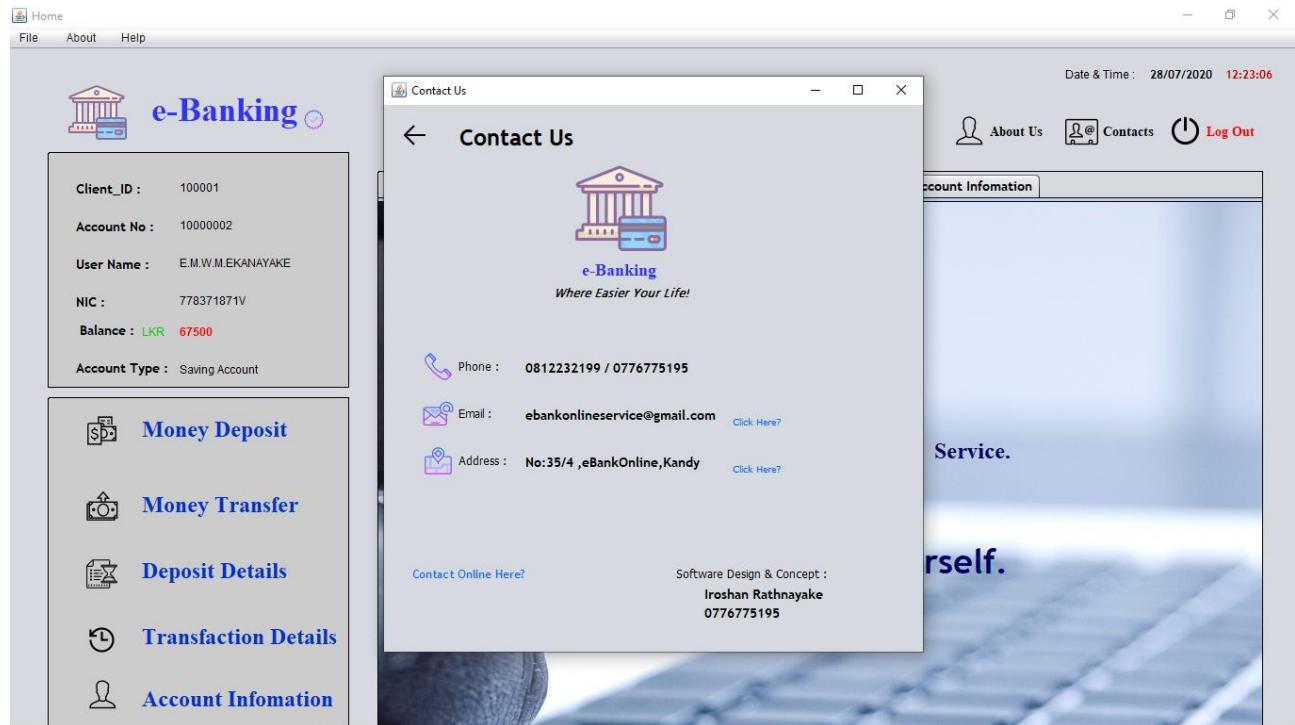


//AboutUs

/* You can see here about bank details and also you can see Vision and Mission of the bank. There is when bank was started and the service of the bank.

*/

➤ Contact Us



```
//Contact Us

//Email - https://mail.google.com/mail/u/0/#inbox?compose=new

private void jLabel12MouseClicked(java.awt.event.MouseEvent evt) {

    try {

        String url = "https://mail.google.com/mail/u/0/#inbox?compose=new";
        java.awt.Desktop.getDesktop().browse(java.net.URI.create(url));
    } catch (IOException ex) {
        Logger.getLogger(Contact.class.getName()).log(Level.SEVERE, null, ex);
    }
}

//Address

private void jLabel5MouseClicked(java.awt.event.MouseEvent evt) {

    try {

        String url =
"https://www.google.lk/maps/place/Kandy/@7.2946291,80.5907618
,13z/data=!3m1!4b1!4m5!3m4!1s0x3ae366266498acd3:0x411a3818a
1e03c35!8m2!3d7.2905715!4d80.6337262";
    }
}
```

```
        java.awt.Desktop.getDesktop().browse(java.net.URI.create(url));

    } catch (IOException ex) {

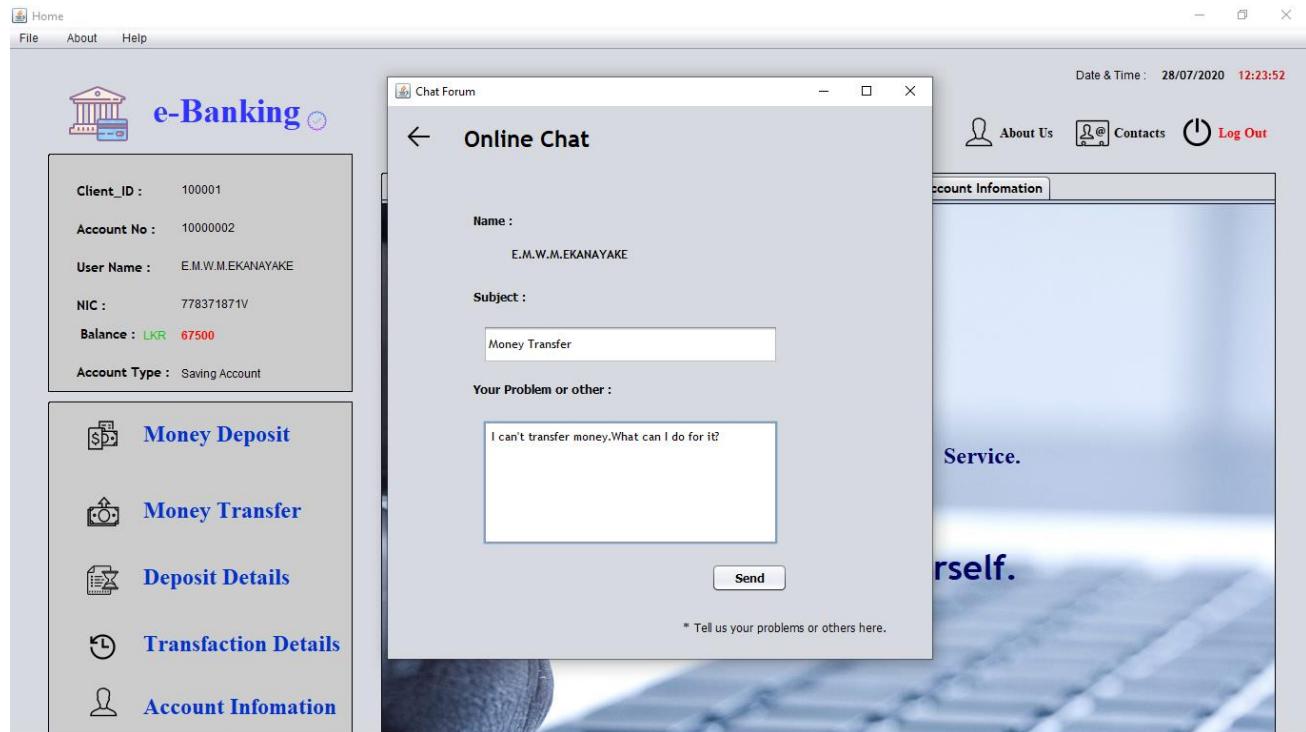
        Logger.getLogger(Contact.class.getName()).log(Level.SEVERE, null, ex);
    }
}

//Online Chat

private void lblOnlineChatMouseClicked(java.awt.event.MouseEvent evt) {

    OnlineMessage msg = new OnlineMessage();
    msg.setVisible(true);
    this.dispose();
}
```

➤ Online Chat



//Online Chat

```
public class OnlineMessage extends javax.swing.JFrame {
```

```
    String phone = null;
```

```
    String ClientID = null;
```

```
    Connection conn;
```

```
    PreparedStatement pst;
```

```
    ResultSet rs;
```

```
    public static final String ACCOUNT_SID =
```

```
        "ACe55609f8867185957887cee60eb9e083";
```

```
    public static final String AUTH_TOKEN =
```

```
        "fe93469492948a5fa1851e63f19febee";
```

```
    public OnlineMessage() {
```

```
        initComponents();
```

```
        conn = DBconnect.connect();
```

```
        info();
```

```

    }

public void info(){

    String id = Login.tbClientID.getText();

    try {

        String sql ="SELECT * FROM tblClient WHERE Client_ID = '"+id+"'";
        pst =conn.prepareStatement(sql);
        rs =pst.executeQuery();

        if(rs.next()){

            ClientID =rs.getString("Client_ID");
            String name = rs.getString("Client_Name");
            lblName.setText(name);
            phone = rs.getString("Phone");
        }

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

//Send Button

private void btnSendActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        Twilio.init(ACCOUNT_SID, AUTH_TOKEN);
        com.twilio.rest.api.v2010.account.Message message =
        com.twilio.rest.api.v2010.account.Message
            .creator(new PhoneNumber("+94776775195"), // to
new PhoneNumber("+12058962320"), // from

```

```
    "eBank \nClient ID - "+ClientID+
    "\n"+lblName.getText()+"\n"+tbSubject.getText()+"\n"+tbB
    ody.getText()+"\n"+phone)

    .create();

    System.out.println(message.getSid());

    JOptionPane.showMessageDialog(null, "Thanks for messaging
    us.We try to be as \nresponsive as possible.We'll get back
    \nyou soon.");

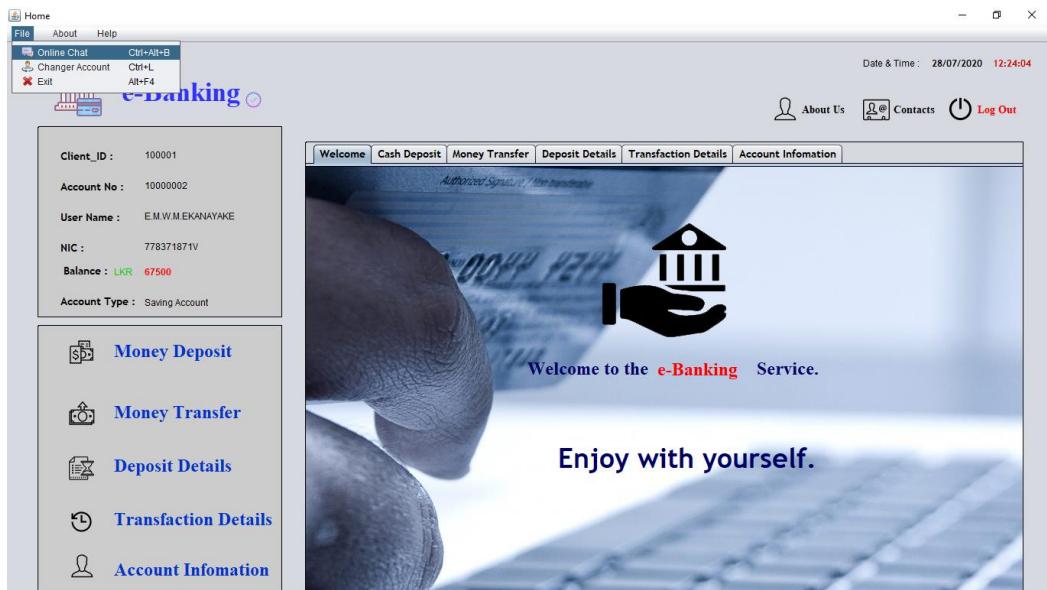
    tbSubject.setText("");
    tbBody.setText("");

} catch (Exception e) {

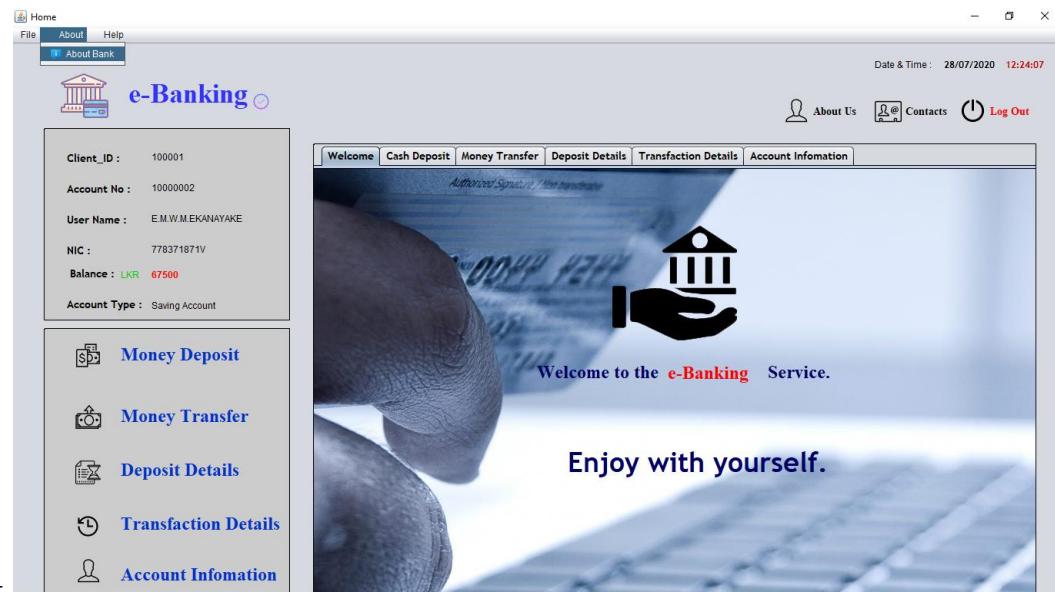
    JOptionPane.showMessageDialog(null, e);

}
```

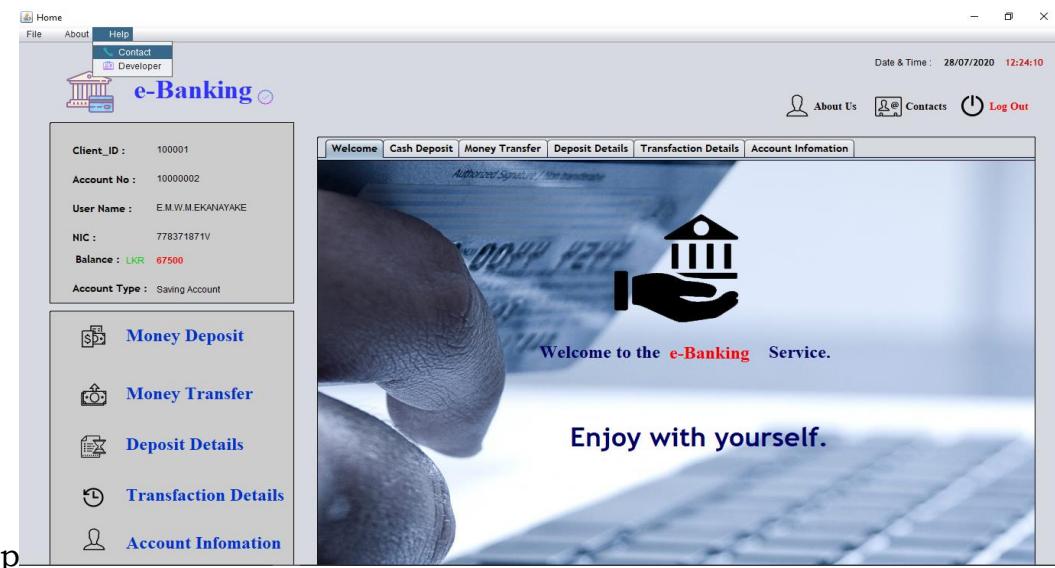
➤ Menu Bar



File

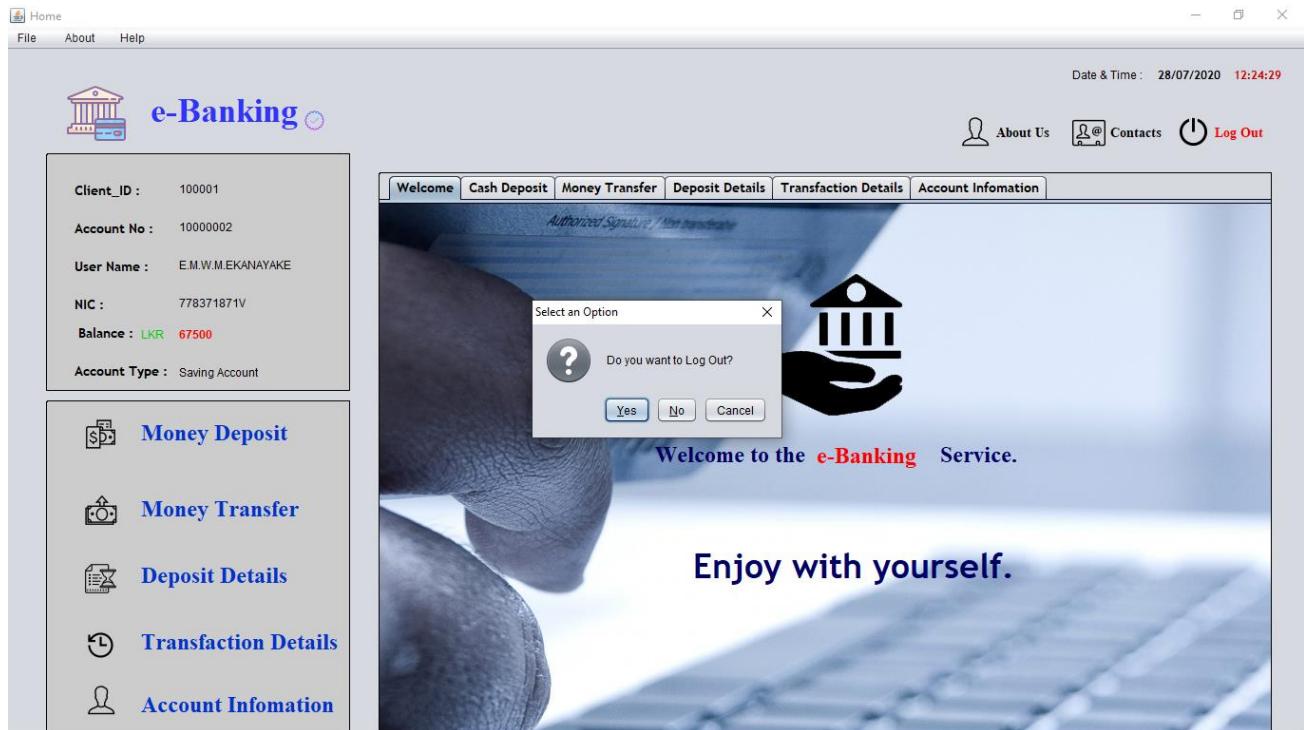


About



Help

➤ Log Out



//Log Out

```
private void jLabel7MouseClicked(java.awt.event.MouseEvent evt) {  
    int check = JOptionPane.showConfirmDialog(null, "Do you want to Log Out?");  
  
    if (check == 0){  
        Login log = new Login();  
        log.setVisible(true);  
        this.dispose();  
    }  
}
```

Development

❖ Future Scope

In the future, We hope to implement online bill payments system with our eClient software. We also offer the opportunity to make local and International transactions with a bank card and this software and we hope to create a mobile application to continue your banking experience.

❖ Testing

➤ In the software testing we got below mention problems.

- Some function may not work correctly.
- Database problems.
- Coding errors.
- Some background and font color using problems.

❖ Implementation

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project. They are

- End user training
- End user Education
- Training on the application software

Evolution & Conlution

❖ Evolution

No	Test Case	Input	Expected Output	Output	Result
01	Login Authentication	Entering incorrect Username and Password	Error message.Access Denied	Error message.Access Denied	Pass
02	Login Authentication	Entering incorrect Username and Password	Access Granted	Access Granted	Pass
03	Adding an Account	Leaving Account No text box blank	Error message.Adding Failed	Error message.Adding Failed	Pass
04	Adding an Account	Adding an Account no which already exist in the system	Error message.Adding Failed	Error message.Adding Failed	Pass
05	Deleting an Account	Pressing delete button without selecting an Account	Error message.Process Failed	Error message.Process Failed	Pass

❖ Conclusion

It has been a matter of immense pleasure, honor and challenge to have this opportunity from some Bank to take up this project and complete it successfully. While developing this project we have learnt a lot about projects & software development. I have also learnt how to make it user friendly (easy to use and handle) by hiding the complicated parts of it from the users. During the development process I studied carefully and understood the criteria for making a software more demanding, I also realized the importance of maintaining a minimal margin for error

References

A.Anon (n.d) - <https://stackoverflow.com/>

Notes