PROFESSIONAL TRAINING REPORT

at

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

by

P DEEPAK CHOWDARY

(40110303)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

NOVEMBER 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **P DEEPAK CHOWDARY (Reg. No: 40110303)** who carried out the project entitled " **ONLINE BANKING APPLICATION**" under my supervision from August 2022 to November 2022.

Internal Guide

Dr.SANKARI M

Head of the Department

Dr. S. Vigneshwari, M.E., Ph.D.,

Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I, **P DEEPAK CHOWDARY** hereby declare that the project report entitled **" ONLINE BANKING APPLICATION"** done by me under the guidance of **Dr. SANKARI M,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

SIGNATURE OF THE CANDIDATE

DATE:

PLACE:

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board **of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph. D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the righttime during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Sankari M, for** her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion ofthe project.

ABSTRACT

Now-a-days, time is treated as money. No one is willing to spend their half a day's time at bank for transactions like balance enquiry, money transfer etc. Complete banking system which is implemented on Java provides complete solution for banking related transactions and you can perform your most important banking chores from the comfort of your home.

The Online Banking Application is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Online Banking Application. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. also to enable the user's work space to have additional functionalities which are not provided under a conventional banking project. This project has been developed to carry out the processes easily and quickly. This project is developed using JAVA language and MYSQL .

Software Prerequisite:

->Mysql
-> Eclipse

MySQL:

MySQL is a full-featured relational database management system (RDBMS). MySQL is a free, open-source relational database management system that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. MySQL is noted for its speed, reliability, and flexibility

->Eclipse

Eclipse is an IDE (interactive development environment) written to develop and debug (primarily) Java code. It contains a base workspace and an extensible plug-in system for customizing the environment.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1	4.1 BANKING SYSTEM APPLICATION	4
2	7.1.1 BANK LOGIN PAGE	6
3	7.1.2 BANK ACCOUNT CREATION PAGE	6
4	7.1.3 BANK OPTIONS PAGE	7
5	7.1.4 BALANCE ENQUIRY PAGE	7
6	7.1.5 MONEY DEPOSIT PAGE	8
7	7.1.6 MONEY WITHDRAW PAGE	8
8	7.1.7 MONEY TRANSFER PAGE	9
9	7.1.8 USER DETAILS PAGE	9

LIST OF ABBREVATIONS

1	API-Application Programming Interface
2	JRE-Java Runtime Environment
3	JDK-java Development kit
4	IDE-Internet Development Environment

TABLES OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF ABBREVATIONS	iii
1	**INTRODUCTION**	**1**
	1.1 FUNCTIONALITIES	1
2	**AIM AND OBJECTIVE**	**2**
	2.1 EXISTING SYSTEM	2
	2.2 PROPOSED SYSTEM	2
3	**EXPERIMENTAL ANALYSES**	**3**
	3.1 SYSTEM CONFIGURATION	3
	3.1.1 HARDWARE REQUIREMENTS	3
	3.1.2 HARDWARE REQUIREMENTS	3
4	**ARCHITECTURE**	**4**
	4.1 BANKING SYSTEM APPLICATION ARCHITECTURE	4
5	**CONCLUSIONS**	**5**
6	**REFERENCES**	**5**
7	**SCREENSHOTS**	**6**
	7.1 BANK SCREENSHORTS	6
8	**SOURCE CODE**	**13**
	8.1 BANKING SYSTEM APPLICATION SOURCE CODE : 1.1 BankMain.java 1.2 AccountCreation.java 1.3 Bank Account Exception.java 1.4 Account.java	

INTRODUCTION

A Bank is a commercial or state institution that provides financial services, including issuing money in form of coins, bank notes or debit cards, receiving deposits of money, lending money and processing transections. A commercial bank accepts deposits from customers and in turn makes loan based on those deposits.

This project is basically providing user friendly interface primarily specializing in Banking System A

We are using following details of clients:

1. User Account Number (Mobile Number)
2. Full Name
3. Aadhaar Number
4. Date of Birth
5. Mail ID
6. Country Name
7. Address
    - 7.1 Pin code
    - 7.2 Street name
    - 7.3 House number
    - 7.4 Area/Locality name
    - 7.5 Village/Town name
    - 7.6 District
    - 7.7 State
8. Nominee Details
    - 8.1 Nominee Full name
    - 8.2 Nominee Mobile number
    - 8.3 Nominee Aadhaar number
    - 8.4 Profession
9. Branch Name
10. Password

FUNCTIONALITIES

- Create his account
- View and modify their personal information
- View the balance and transaction history of their accounts
- Make a Transfer (Internal or external)
- Feed and account
- Viewing transections of deposit.
- View Clients personal information
- View the balance and transaction history of all client accounts

AIM

Online Banking Application is implemented in java platform. Main aim of this project is to develop a simple banking software application through which users can add account, depositing money in to account, withdraw money from the account, request for balance amount available in their account, showing transactions details.

OBJECTIVE

This Online Banking Application lets users create a new account, make cash deposits and withdrawals, access account information through software without the need of going personally to a bank.

This Online Banking Application project has been designed in Java and various actions have different panels. It keeps logs of all the transactions made whether it is withdrawal or deposit of cash. One can create a new account and even delete an existing one. This gives user the facility to work in one's own comfort and at a time convenient for them. It is secure and user-friendly and also in our banking system there is self to self transaction and also peer-to-peer transaction.

EXISTING SYSTEM

The manual Bank Application System that carry out the whole process in a bank are more prone to errors. It takes a vast amount of time to take an action be it the creation of an account or depositing money. It is especially negative when there are time constraints for the customers to get a work done. This makes it a lot inconvenient for the customer as well as it is time consuming. Banks are open only in limited hours in a day and any transaction to be made after those times in cases of emergencies become impossible. This causes a lot of inconvenience.

PROPOSED SYSTEM

This Bank Application System has been designed to overcome all the disadvantages of the traditional banking system. A user has the option of creating an account, cash transactions all at their own discretion. There is a separate module for each and every action and it is really easy to use and secure as well. This software helps save a lot of time, energy and effort that the customers have to go through in carrying out these actions personally.

EXPERIMENTAL ANALYSIS

SYSTEM CONFIGURATION

Hardware requirements:

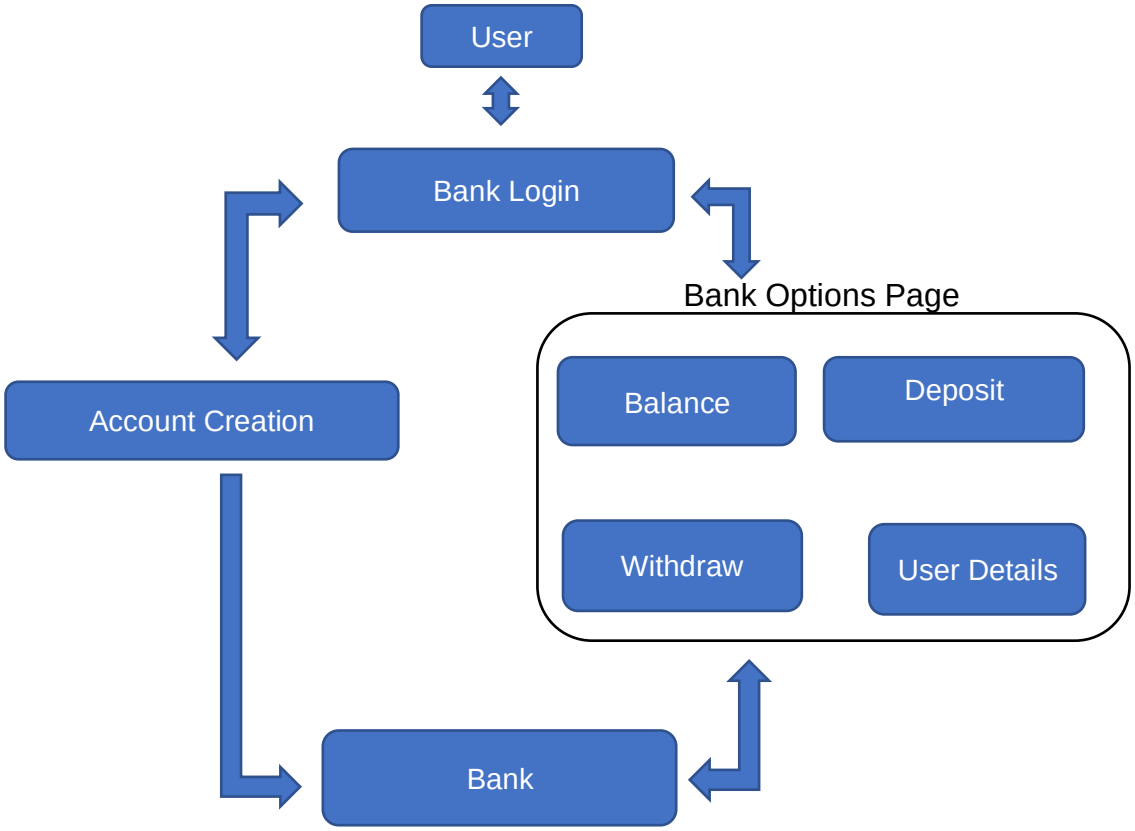
Operating System	:	Laptop Intel i3 3rd Gen clock speed 2.5 GHz & above
Hard disk	:	min. 10 GB
Ram	:	1 GB or above
Monitor resolution	:	800 X 600

Software requirements:

ANY Operating System (Windows 10 / 8, Linux, Mac)
JDK 8 (Java Development Kit)
JRE 8 (Java Runtime Environment)
Language Used : JAVA 17
Eclipse IDE
MySQL
MySQL Connecter

ARCHITECTURE

The system architecture gives an overview of the working of the system. The working of this system is described as follows: Collecting and verifying the user account number and password. If the given details are available in bank database. then they will be got access to login. Else user must create a new account. Bank Options page will be open in that page user can view Balance enquiry, Deposit money, Withdraw money, Transfer money and User Details. User can Choose any option.



Online Banking Application

This is architectural diagram of banking application ,here we can easily create account , and also deposit ,withdraw ,user Details and also balance ,there are bank options available and also there is about ,it means who created this applications and also there ia exit option in this online banking application.

CONCLUSION

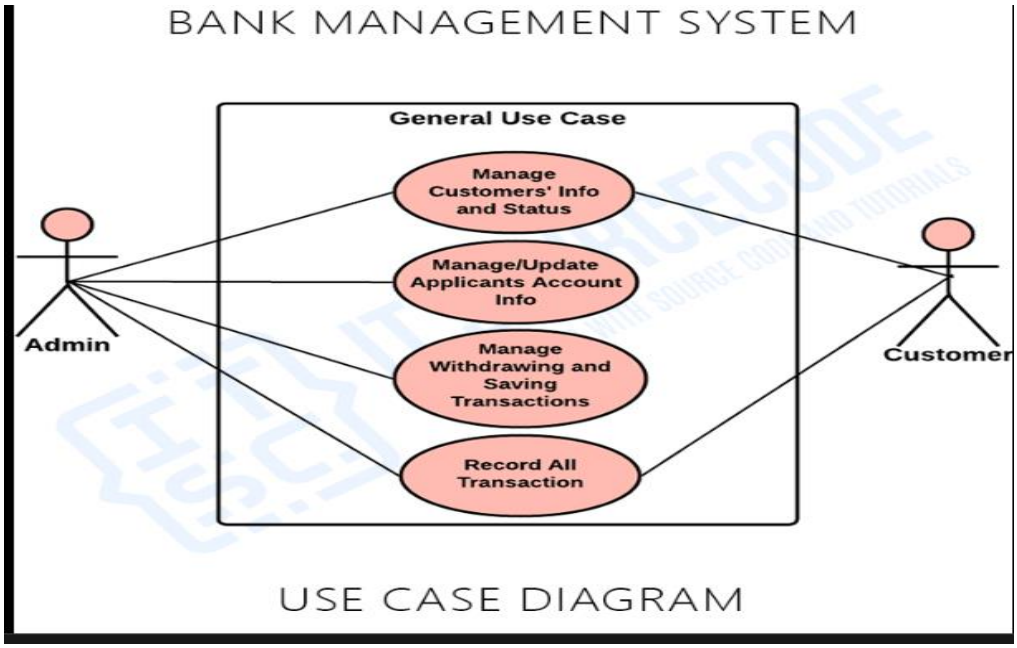
We conclude to build a sealable interface that simplifies the banking system handling.
It has the following objectives:

- ➔ Improved access to all transection and other operation.

Easy access and handling of customer account.

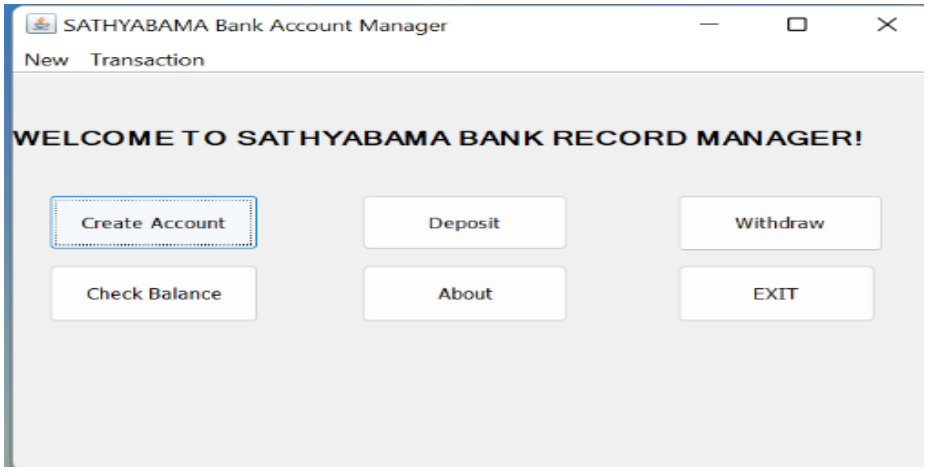
- ➔ Detailed access of customer is easy for accountant and all customer and accountant details access is simple for Data Base Manager.

USER DIAGRAM: For Traditional banking application

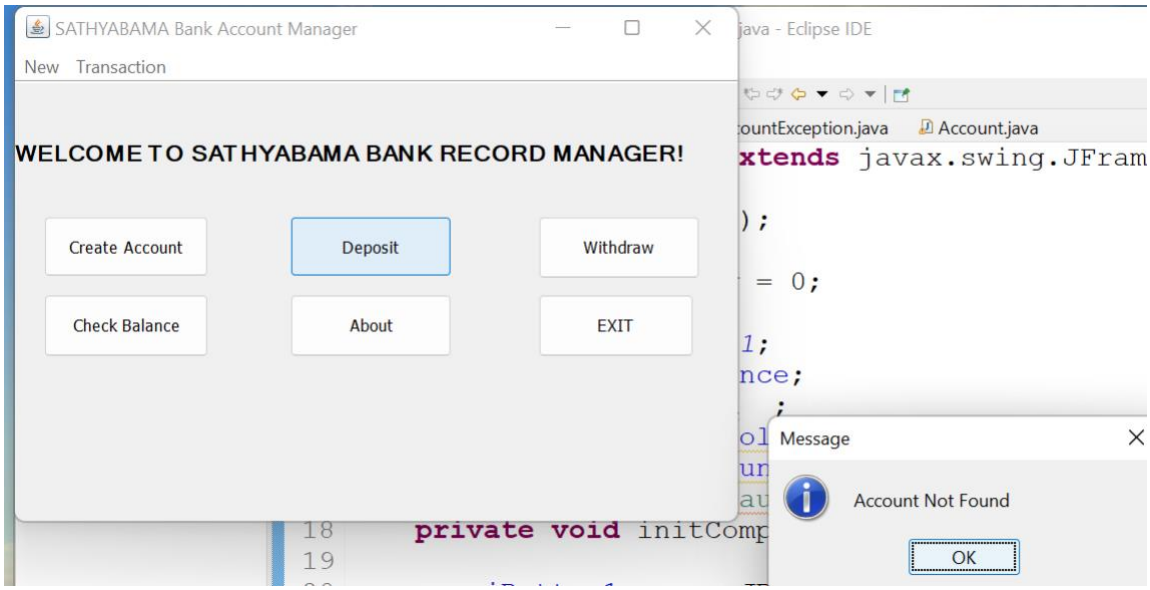


In Traditional banking application use case diagram there will be admin and a customer ,even for transaction we need to go and meet admin and also for account creation it takes av lot of time so it is not well suited,but in online banking application it is easy to create our account and also we can easly withdraw,deposit amount very easily.so we prefer online banking application as compared to traditional banking application

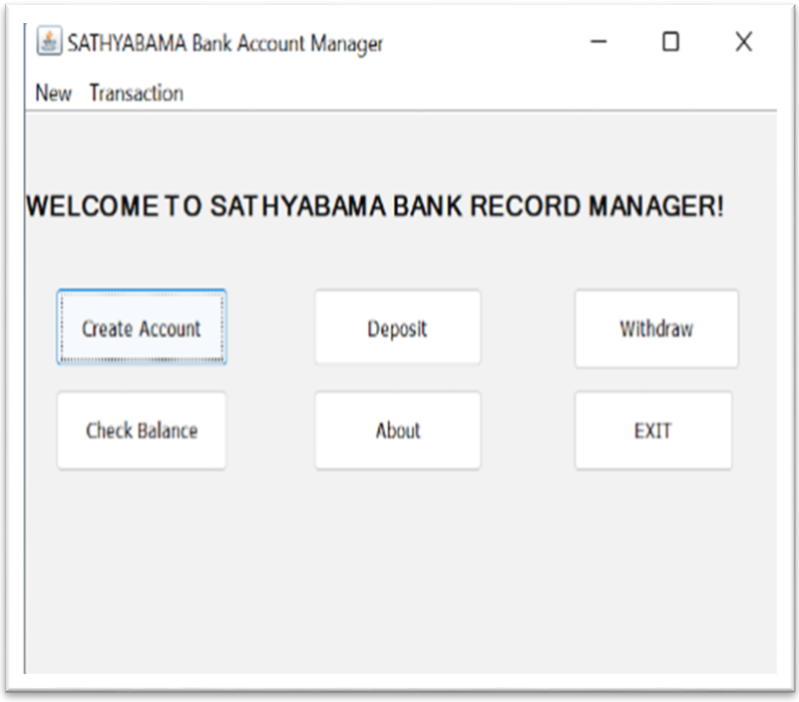
6.1->CREATE ACCOUNT

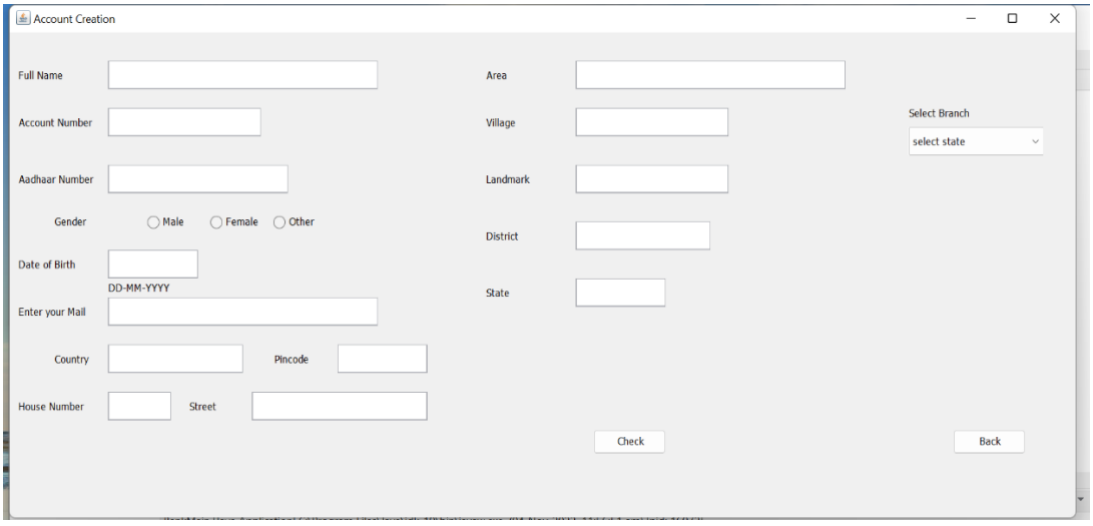


6.2->Account not found



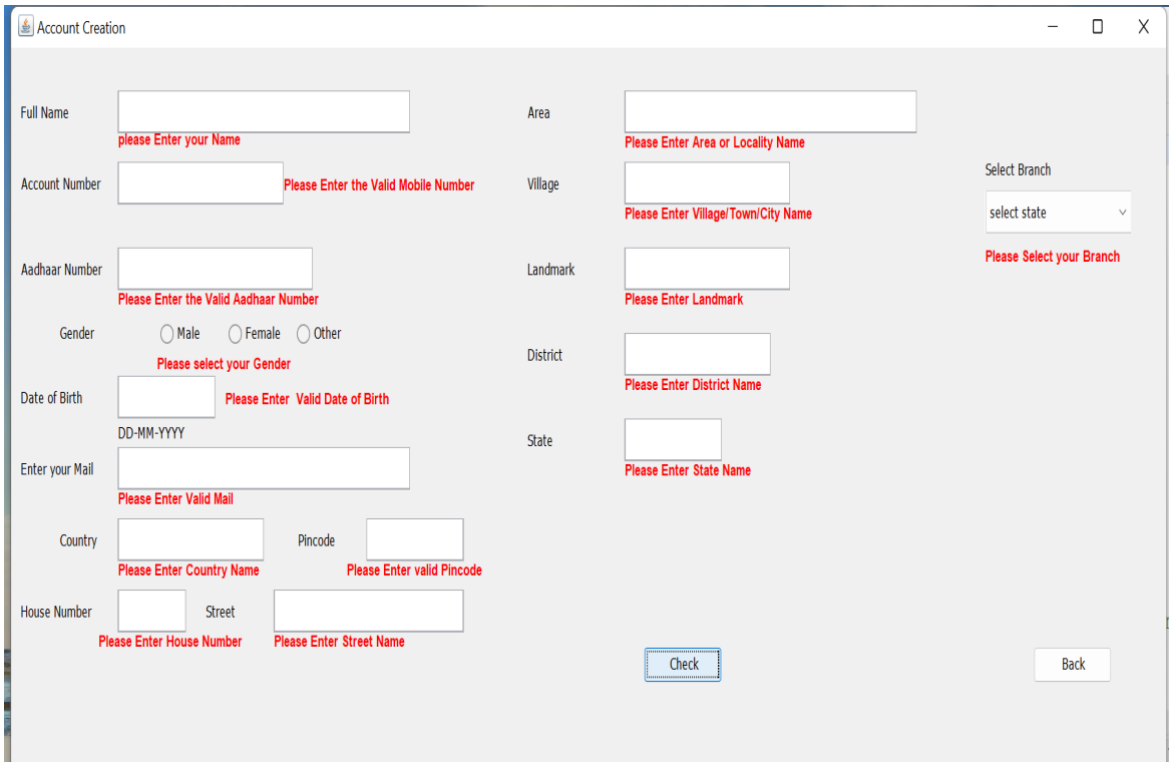
6.3->Click Create Account

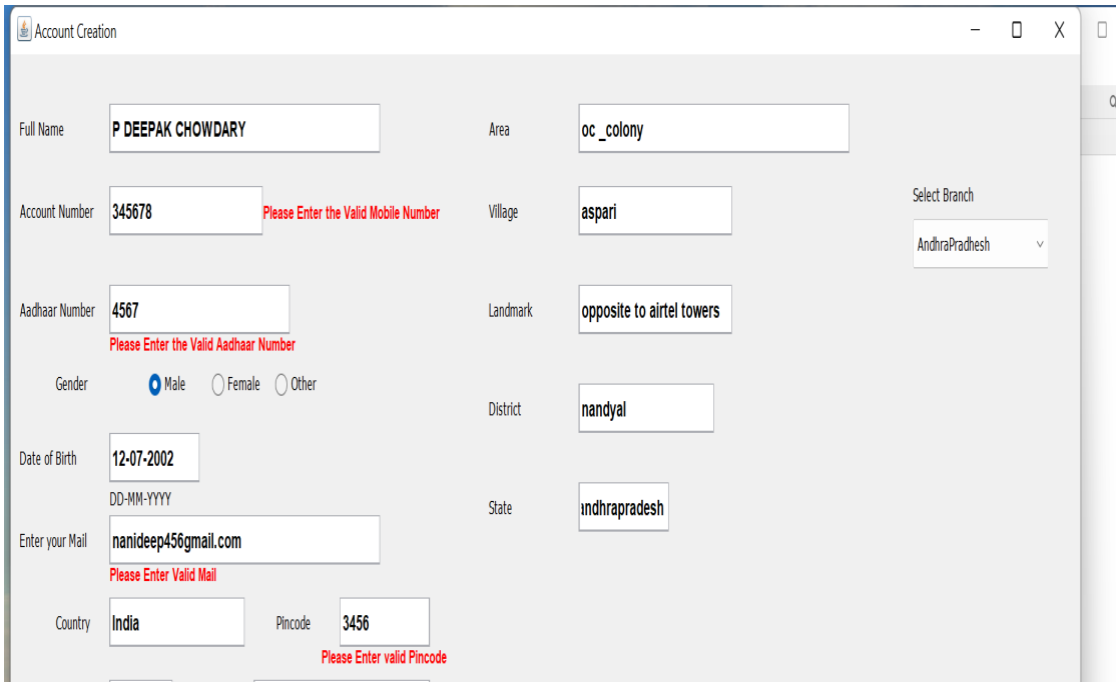


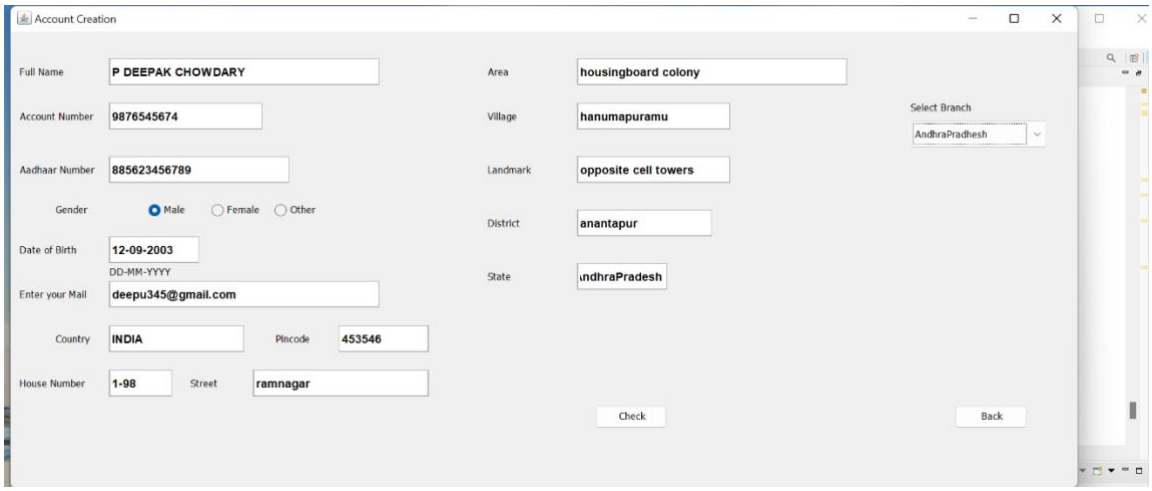


6.4->Account Creation form

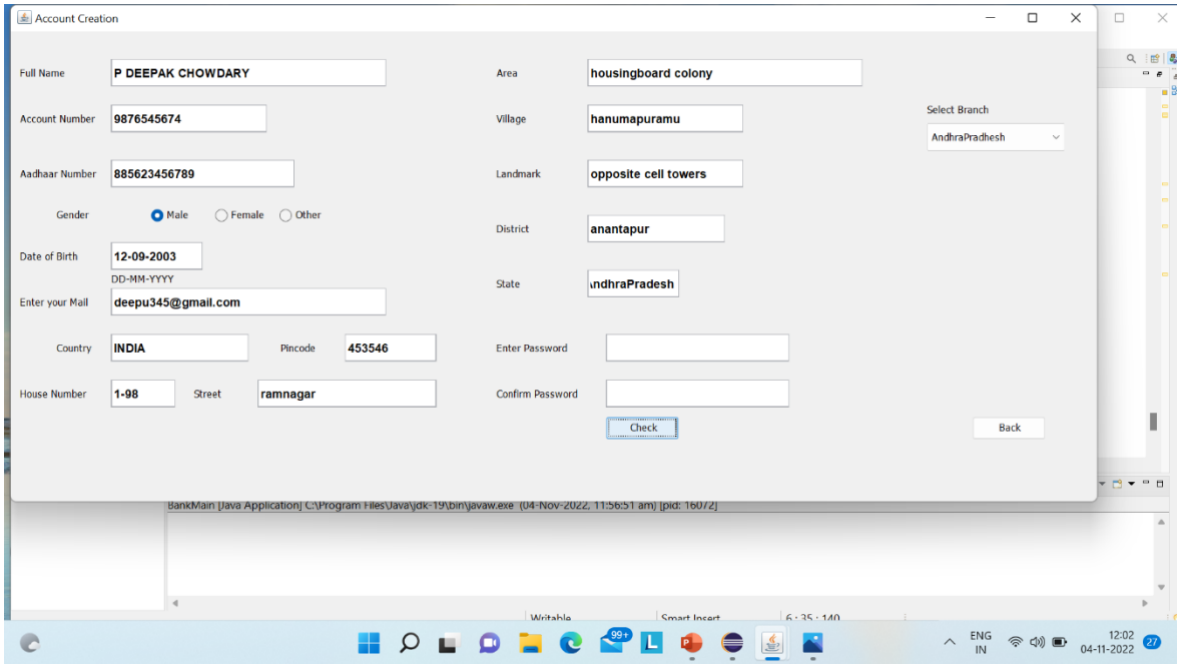
6.5->Fill All Mandatory Details

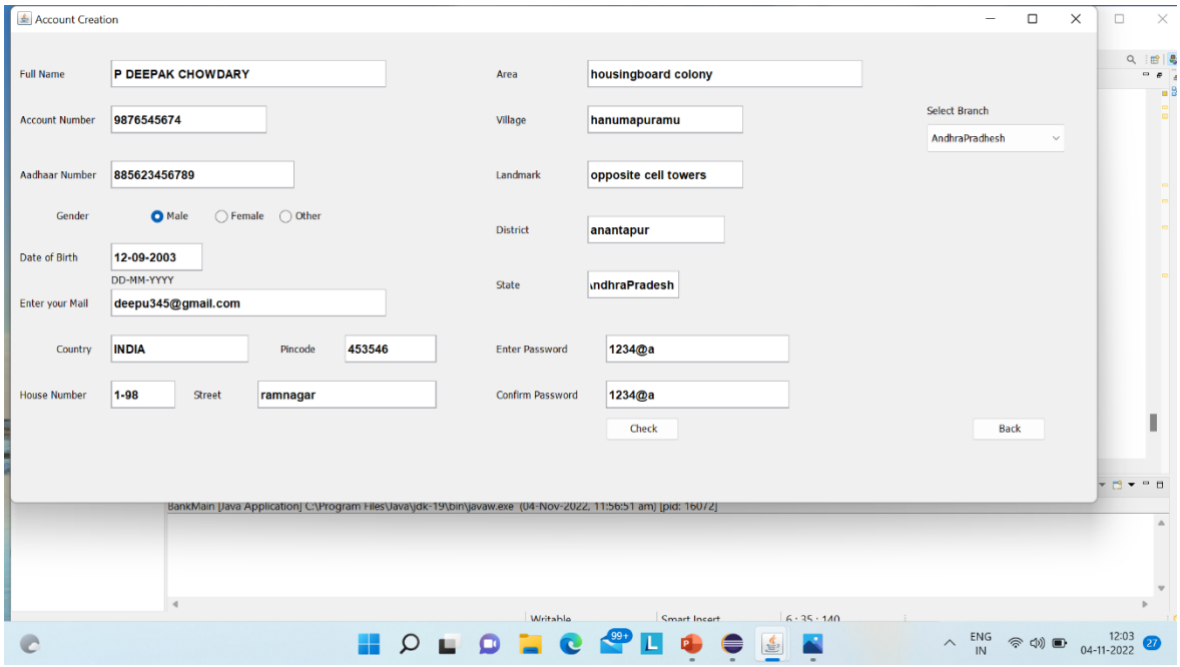




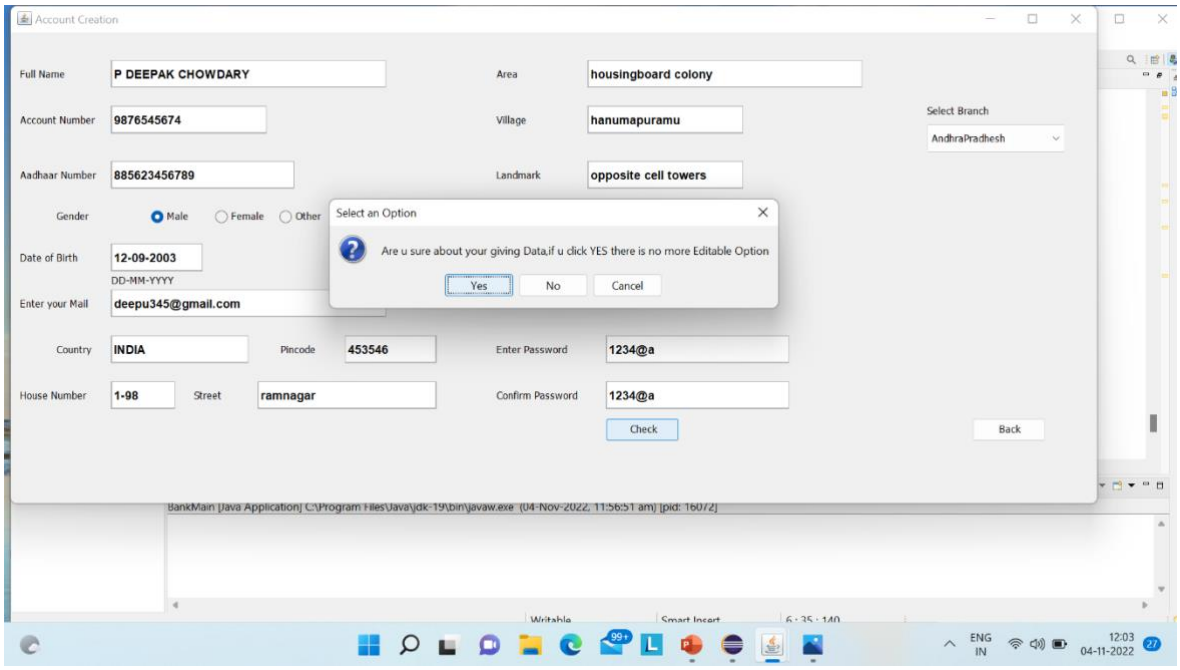


6.6->Fill all details and click and now enter user name and password

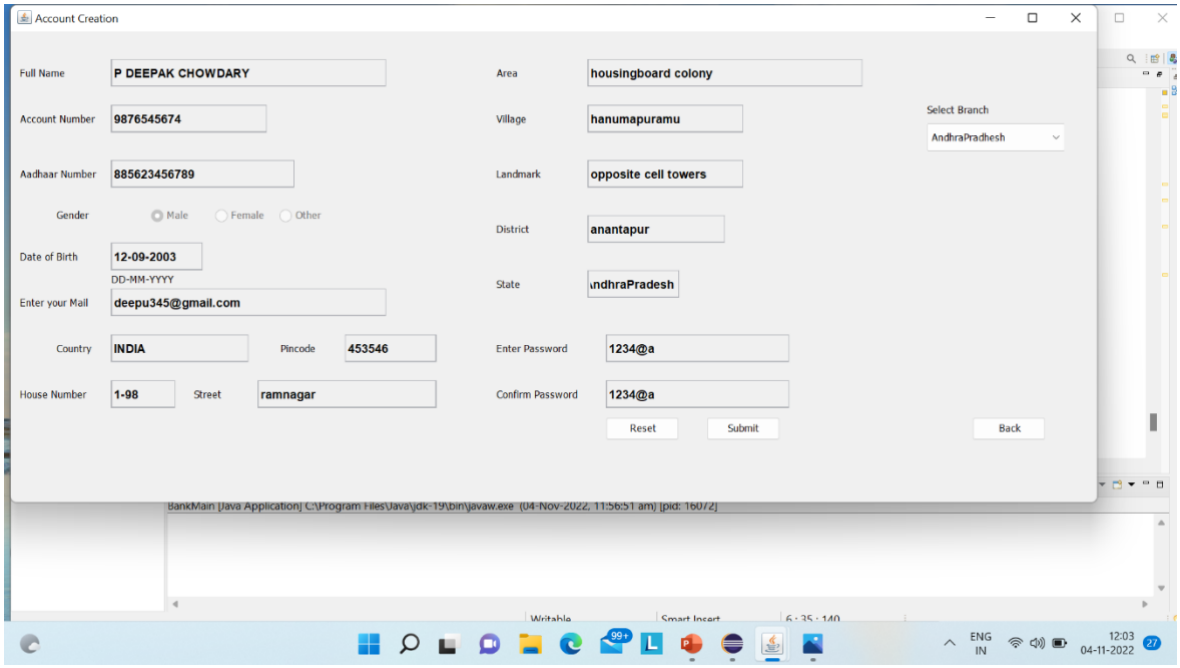




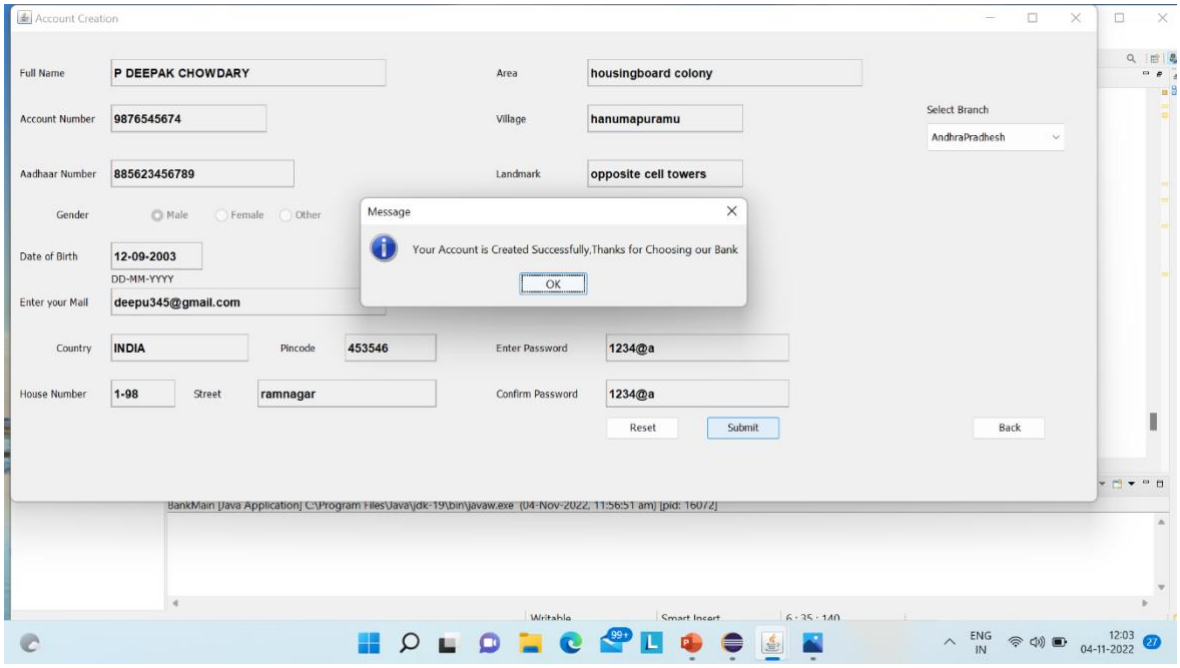
6.7->Now enter ur user name and password and click yes



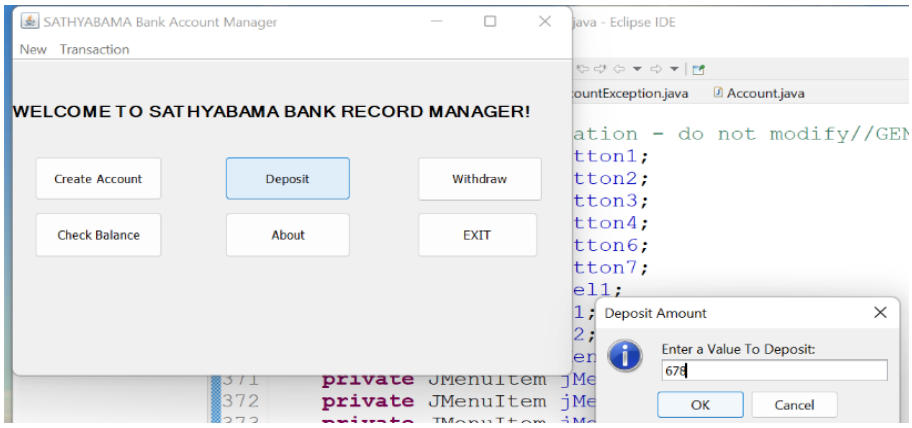
6.8-> U can reset ur form



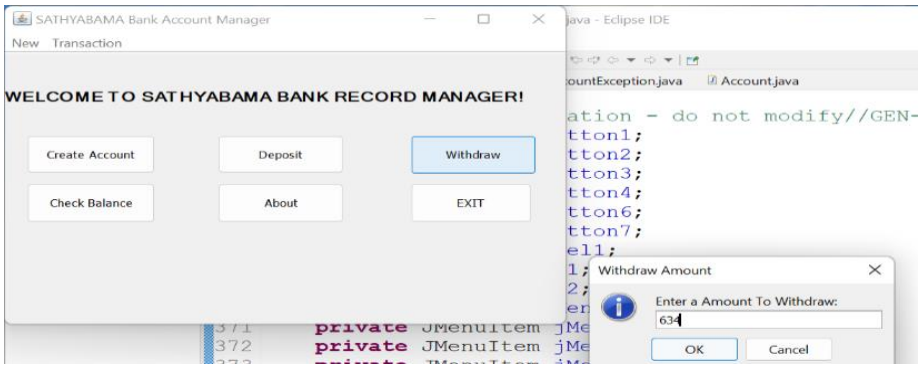
6.9->Click submit



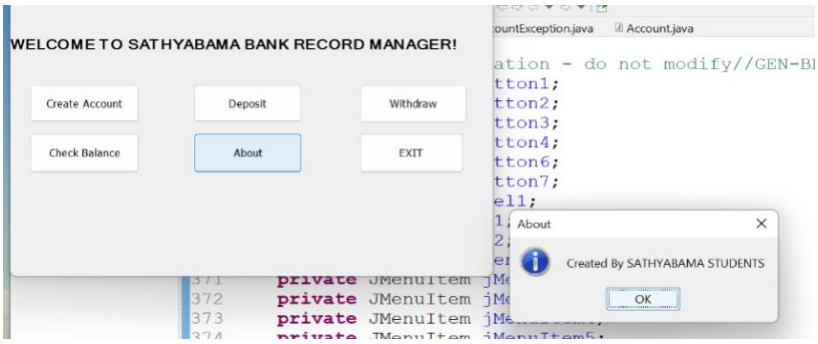
6.10->Now deposit ur money



6.11->Here u can withdraw ur money



6.12->Here u need to check ur balance



6.13->About created by Sathyabhama Students

SOURCE CODE:

1.1—BankMain.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BankMain extends javax.swing.JFrame {
    public BankMain() {
        initComponents();
    }
    static int acc_flag = 0;
    static Long num;
    static String name_1;
    private double balance;
    private long accNum  ;
    private String accHolder;
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
    private void initComponents() {


        jButton1 = new JButton();
```

```
jLabel1 = new JLabel();
jButton2 = new JButton();
jButton3 = new JButton();
jButton4 = new JButton();
jButton6 = new JButton();
jButton7 = new JButton();
jMenuBar1 = new JMenuBar();
jMenu1 = new JMenu();
jMenuItem1 = new JMenuItem();
jSeparator1 = new JPopupMenu.Separator();
jMenuItem3 = new JMenuItem();
jMenuItem4 = new JMenuItem();
jMenu2 = new JMenu();
jMenuItem5 = new JMenuItem();
jMenuItem6 = new JMenuItem();
jMenuItem7 = new JMenuItem();

setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
setTitle("ITSOURCECODE Bank Account Manager");

jButton1.setText("Create Account");
jButton1.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      jButton1ActionPerformed(evt);
            new AccountCreation();
   }
});

jLabel1.setFont(new Font("Microsoft Sans Serif", 1, 15)); // NOI18N
jLabel1.setText("WELCOME TO SATHYABAMA BANK RECORD MANAGER!");

jButton2.setText("Deposit");
```

```
jButton2.setMaximumSize(new Dimension(107, 23));
jButton2.setMinimumSize(new Dimension(107, 23));
jButton2.setPreferredSize(new Dimension(107, 23));
jButton2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setText("Withdraw");
jButton3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

jButton4.setText("Check Balance");
jButton4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

jButton6.setText("About");
jButton6.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});

jButton7.setText("EXIT");
jButton7.addActionListener(new ActionListener() {
```

```
            public void actionPerformed(ActionEvent evt) {
                jButton7ActionPerformed(evt);
            }
        });

        jMenu1.setText("New");

        jMenuItem1.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,
InputEvent.CTRL_MASK));
        jMenuItem1.setText("New Account..");
        jMenuItem1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem1);
        jMenu1.add(jSeparator1);

        jMenuItem3.setText("About");
        jMenuItem3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                jMenuItem3ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem3);

        jMenuItem4.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F4,
InputEvent.ALT_MASK));
        jMenuItem4.setText("Exit");
        jMenuItem4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                jMenuItem4ActionPerformed(evt);
            }
```

```
        });
        jMenu1.add(jMenuItem4);

        jMenuBar1.add(jMenu1);

        jMenu2.setText("Transaction");

        jMenuItem5.setText("Deposit Amount..");
        jMenuItem5.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                jMenuItem5ActionPerformed(evt);
            }
        });
        jMenu2.add(jMenuItem5);

        jMenuItem6.setText("Withdraw Amount..");
        jMenuItem6.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                jMenuItem6ActionPerformed(evt);
            }
        });
        jMenu2.add(jMenuItem6);

        jMenuItem7.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_B,
InputEvent.CTRL_MASK));
        jMenuItem7.setText("Check Account Balance..");
        jMenuItem7.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                jMenuItem7ActionPerformed(evt);
            }
        });
        jMenu2.add(jMenuItem7);
```

```
        jMenuBar1.add(jMenu2);


        setJMenuBar(jMenuBar1);


        GroupLayout layout = new GroupLayout(getContentPane());
        getContentPane().setLayout(layout);

layout.setHorizontalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEA
DING)
            .addGroup(layout.createSequentialGroup()
                .addGap(20, 20, 20)
                .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING,
false)
                    .addComponent(jButton4, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jButton1, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(55, 55, 55)
                .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
                    .addComponent(jButton2, GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton6, GroupLayout.PREFERRED_SIZE, 107,
GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                    .addComponent(jButton3, GroupLayout.PREFERRED_SIZE, 107,
GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton7, GroupLayout.PREFERRED_SIZE, 103,
GroupLayout.PREFERRED_SIZE))
                .addGap(26, 26, 26))
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel1, GroupLayout.DEFAULT_SIZE, 472,
Short.MAX_VALUE)
                .addContainerGap())
        );
```

```
layout.setVerticalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADI
NG)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jLabel1, GroupLayout.PREFERRED_SIZE, 72,
GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)

            .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)

                .addComponent(jButton3, GroupLayout.Alignment.TRAILING,
GroupLayout.PREFERRED_SIZE, 41, GroupLayout.PREFERRED_SIZE)


.addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)

                    .addComponent(jButton1, GroupLayout.PREFERRED_SIZE, 40,
GroupLayout.PREFERRED_SIZE)

                    .addComponent(jButton2, GroupLayout.PREFERRED_SIZE, 40,
GroupLayout.PREFERRED_SIZE)))

            .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)

            .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING,
false)

                .addComponent(jButton4, GroupLayout.DEFAULT_SIZE, 41,
Short.MAX_VALUE)

                .addComponent(jButton6, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jButton7, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

            .addGap(110, 110, 110))

      );


      pack();

   }// </editor-fold>//GEN-END:initComponents

   private static Account acc;


   private void jMenuItem6ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jMenuItem6ActionPerformed

        try

        {
```

```
            String num;
            num = JOptionPane.showInputDialog(null, "Enter a Amount To Withdraw:",
"Withdraw Amount", 1);
            double num1 = Double.parseDouble(num);
//              acc.withdraw(num1);
            balance = balance - num1;
         }
         catch(Exception ex)
         {
            JOptionPane.showMessageDialog(null, ex, "Error", 2);
         }
   }//GEN-LAST:event_jMenuItem6ActionPerformed


   private void jButton2ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed
      try
      {
         if(acc_flag == 1)
         {
                String num;
            num = JOptionPane.showInputDialog(null, "Enter a Value To Deposit:",
"Deposit Amount", 1);
            double num1 = Double.parseDouble(num);
            if(num1 > 0)
            {
                balance = balance + num1;
            }
            else
                JOptionPane.showMessageDialog(null,"Enter Valid Amount");
         }
         else
                JOptionPane.showMessageDialog(null,"Account Not Found");
      }
```

```
        catch(NumberFormatException | NullPointerException nfe1)
        {
            JOptionPane.showMessageDialog(null, nfe1, "Error", 2);
        }
    }//GEN-LAST:event_jButton2ActionPerformed

    private void jMenuItem1ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jMenuItem1ActionPerformed

    }//GEN-LAST:event_jMenuItem1ActionPerformed

    private void jButton6ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jButton6ActionPerformed
        JOptionPane.showMessageDialog(null,"Created By SATHYABAMA STUDENTS
", "About", 1);
    }

    private void jMenuItem3ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jMenuItem3ActionPerformed
        JOptionPane.showMessageDialog(null,"Created By SATHYABAMA
STUDENTS", "About", 1);
    }

    private void jMenuItem4ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jMenuItem4ActionPerformed
        System.exit(0);
    }

    private void jMenuItem5ActionPerformed(ActionEvent evt) {//GEN-
FIRST:event_jMenuItem5ActionPerformed
        try
        {
            if(acc_flag == 1)
            {
                    String num;
```

```
            num = JOptionPane.showInputDialog(null, "Enter a Value To Deposit:",
"Deposit Amount", 1);
            double num1 = Double.parseDouble(num);
            if(num1 > 0)
            {
                balance = balance + num1;
            }
            else
                JOptionPane.showMessageDialog(null,"Enter Valid Amount");
        }
        else
                JOptionPane.showMessageDialog(null,"Account Not Found");

     }
     catch(NumberFormatException | NullPointerException nfe1)
     {
        JOptionPane.showMessageDialog(null, nfe1, "Error", 2);
     }
   }

   private void jMenuItem7ActionPerformed(ActionEvent evt) {
     try
     {
          double num = balance;
      JOptionPane.showMessageDialog(null, "Current Balance: " + num, "Current
Balance", 1);
     }
     catch(NullPointerException npe)
     {
        JOptionPane.showMessageDialog(null, npe, "Error", 2);
     }
   }
```

```
    private void jButton1ActionPerformed(ActionEvent evt) {
    }

    private void jButton3ActionPerformed(ActionEvent evt) {
        try
          {
           if(acc_flag == 1)
           {
                    if(balance > 0)
                    {
                             String num;
                num = JOptionPane.showInputDialog(null, "Enter a Amount To
Withdraw:", "Withdraw Amount", 1);
                double num1 = Double.parseDouble(num);
                balance = balance - num1;
                 }
                 else
                          JOptionPane.showMessageDialog(null,"Insufficient Balance");
           }
           else
           {
                    JOptionPane.showMessageDialog(null,"Account Not Found");
           }
          }
          catch(Exception ex)
          {
             JOptionPane.showMessageDialog(null, ex, "Error", 2);
             ex.printStackTrace();
          }
    }

    private void jButton4ActionPerformed(ActionEvent evt) {
      try
```

```
    {
        if(acc_flag == 1)
        {
            double num = balance;
          JOptionPane.showMessageDialog(null, "Current Balance: " + num,
"Current Balance", 1);
        }
        else
        {
            JOptionPane.showMessageDialog(null,"Account Not Found");
        }
    }
    catch(NullPointerException npe)
    {
       JOptionPane.showMessageDialog(null, npe, "Error", 2);
       npe.printStackTrace();
    }
  }
  private void jButton7ActionPerformed(ActionEvent evt) {
     System.exit(0);
  }

  /**
   * @param args the command line arguments
   */
  public static void main(String args[]) {
     try {
         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Windows".equals(info.getName())) {
               javax.swing.UIManager.setLookAndFeel(info.getClassName());
               break;
            }
```

```
            }
        } catch (Exception ex) {

java.util.logging.Logger.getLogger(BankMain.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
            ex.printStackTrace();
        }
        java.awt.EventQueue.invokeLater(() -> {
            new BankMain().setVisible(true);
        });
    }


    // Variables declaration - do not modify//GEN-BEGIN:variables
    private JButton jButton1;
    private JButton jButton2;
    private JButton jButton3;
    private JButton jButton4;
    private JButton jButton6;
    private JButton jButton7;
    private JLabel jLabel1;
    private JMenu jMenu1;
    private JMenu jMenu2;
    private JMenuBar jMenuBar1;
    private JMenuItem jMenuItem1;
    private JMenuItem jMenuItem3;
    private JMenuItem jMenuItem4;
    private JMenuItem jMenuItem5;
    private JMenuItem jMenuItem6;
    private JMenuItem jMenuItem7;
    private JPopupMenu.Separator jSeparator1;
    // End of variables declaration//GEN-END:variables
}
```

1.2—Account* Creation..*java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.regex.*;
import java.sql.*;

public class AccountCreation extends BankMain implements ActionListener
{
        JFrame fbac;

        JFileChooser uphoto;

        JLabel name,account_number,aadhaar,branch;

        JLabel namecnd,account_numbercnd, gender,country,pincode,DOB,

        mail,password,confirmpassword,houseno,street,landmark,area_loc,village_t
_c,

        district,state,noname,nomobile,noaadhar,noprofession,nocountry,noarea;

        JLabel nameErr,account_numberErr,aadhaarErr,genderErr,
        countryErr,pincodeErr,DOBErr,mailErr,confirmpasswordErr,
        housenoErr,streetErr,landmarkErr,area_locErr,village_t_cErr,
        districtErr,stateErr,nonameErr,nomobileErr,noaadharErr,noprofessionErr,no
countryErr,noareaErr
        ,datedemo,branchErr;

        JTextField tname,taccount_number,taadhaar,tcountry,tpincode,tDOB,

        tmail,tpassword,tconfirmpassword,thouseno,tstreet,tlandmark,tarea_loc,tvill
age_t_c,
```

```
        tdistrict,tstate,tnoname,tnomobile,tnoaadhar,tnoprofession,tnocountry,tnoar
ea;


        JTextArea taddress;


        ButtonGroup bg;
        JRadioButton male,female,other;
        JButton check,submit,reset,uphotobtn,backbtn;
        Font fontcnd,fontArea,fontField,fontError;
        String nameUpper,spass,scpass,ssbranch;
        String
sname,saccno,saadno,sDOB,smail,scountry,spin,shouseno,sstreet,sarea_loc,svilla
ge,
                        slandmark,sdistrict,sstate;
        String sgender,smale,sfemale,sother,scpassword;
        String snoname,snomobile,snoaadno,sprofession,snocountry,snoarea_loc;
        Integer ipincode;
        Long laccno,laadno,lnomobile,lnoaadno;
        JComboBox branches;


        boolean bDOB;
        AccountCreation()
        {
                fbac = new JFrame("Account Creation");
                /*
                 * User Photo
                 */
                String[] sbranch = {"select
state","AndhraPradhesh","Telangana","Tamilnadu","Kerala","Maharashtra","Bihar"
,"Kashmir","Other"};
                branches = new
JComboBox(sbranch);branches.setBounds(1000,100,150,30);fbac.add(branches);
                /*
                 * Fonts
```

```
                 */
                fontcnd = new Font("POPPINS",Font.BOLD,10);
                fontArea = new Font("POPPINS",Font.BOLD,15);
                fontField = new Font("POPPINS",Font.BOLD,13);
                fontError = new Font("POPPINS",Font.BOLD,11);
                /*
                 * Labels
                 */
                name = new JLabel("Full
Name");name.setBounds(10,30,100,30);fbac.add(name);

                account_number = new JLabel("Account
Number");account_number.setBounds(10,80,100,30);
                fbac.add(account_number);

                aadhaar = new JLabel("Aadhaar
Number");aadhaar.setBounds(10,140,100,30);fbac.add(aadhaar);

                gender = new
JLabel("Gender");gender.setBounds(50,185,90,30);fbac.add(gender);

                country = new
JLabel("Country");country.setBounds(50,330,80,30);fbac.add(country);

                pincode = new
JLabel("Pincode");pincode.setBounds(295,330,65,30);fbac.add(pincode);

                DOB = new JLabel("Date of
Birth");DOB.setBounds(10,230,100,30);fbac.add(DOB);
                datedemo = new JLabel("DD-MM-
YYYY");datedemo.setBounds(110,260,100,20);fbac.add(datedemo);

                mail = new JLabel("Enter your
Mail");mail.setBounds(10,280,100,30);fbac.add(mail);
```

```
houseno = new JLabel("House
Number");houseno.setBounds(10,380,100,30);fbac.add(houseno);


street = new
JLabel("Street");street.setBounds(200,380,70,30);fbac.add(street);


area_loc = new
JLabel("Area");area_loc.setBounds(530,30,100,30);fbac.add(area_loc);


village_t_c = new
JLabel("Village");village_t_c.setBounds(530,80,100,30);
fbac.add(village_t_c);


landmark = new
JLabel("Landmark");landmark.setBounds(530,140,100,30);fbac.add(landmark);


district = new
JLabel("District");district.setBounds(530,200,100,30);fbac.add(district);


state = new
JLabel("State");state.setBounds(530,260,100,30);fbac.add(state);


branch = new JLabel("Select
Branch");branch.setBounds(1000,70,150,30);fbac.add(branch);


/*
 * Error Labels
 */
nameErr = new
JLabel();nameErr.setBounds(110,48,300,30);fbac.add(nameErr);
nameErr.setFont(fontError);nameErr.setForeground(Color.red);


account_numberErr = new
JLabel();account_numberErr.setBounds(280,80,250,30);
fbac.add(account_numberErr);account_numberErr.setFont(fontError);
account_numberErr.setForeground(Color.red);
```

```
                aadhaarErr = new
JLabel();aadhaarErr.setBounds(110,160,300,30);fbac.add(aadhaarErr);
                aadhaarErr.setFont(fontError);aadhaarErr.setForeground(Color.red);


                genderErr = new
JLabel();genderErr.setBounds(150,205,250,30);fbac.add(genderErr);
                genderErr.setFont(fontError);genderErr.setForeground(Color.red);


                countryErr = new
JLabel();countryErr.setBounds(110,350,250,30);fbac.add(countryErr);
                countryErr.setFont(fontError);countryErr.setForeground(Color.red);


                pincodeErr = new
JLabel();pincodeErr.setBounds(345,350,200,30);fbac.add(pincodeErr);
                pincodeErr.setFont(fontError);pincodeErr.setForeground(Color.red);


                DOBErr = new
JLabel();DOBErr.setBounds(220,230,200,30);fbac.add(DOBErr);
                DOBErr.setFont(fontError);DOBErr.setForeground(Color.red);


                mailErr = new
JLabel();mailErr.setBounds(110,300,300,30);fbac.add(mailErr);
                mailErr.setFont(fontError);mailErr.setForeground(Color.red);


                confirmpasswordErr = new
JLabel();confirmpasswordErr.setBounds(165,635,200,30);

        confirmpasswordErr.setFont(fontError);confirmpasswordErr.setForeground(
Color.red);
                fbac.add(confirmpasswordErr);


                housenoErr = new JLabel();housenoErr.setBounds(90,400,200,30);

        housenoErr.setFont(fontError);housenoErr.setForeground(Color.red);fbac.a
dd(housenoErr);
```

```
                streetErr = new JLabel();streetErr.setBounds(270,400,200,30);

        streetErr.setFont(fontError);streetErr.setForeground(Color.red);fbac.add(stre
etErr);



                area_locErr = new JLabel();area_locErr.setBounds(630,50,250,30);
                area_locErr.setFont(fontError);area_locErr.setForeground(Color.red);
                fbac.add(area_locErr);


                village_t_cErr = new
JLabel();village_t_cErr.setBounds(630,100,250,30);

        village_t_cErr.setFont(fontError);village_t_cErr.setForeground(Color.red);
                fbac.add(village_t_cErr);


                landmarkErr = new JLabel();landmarkErr.setBounds (630,160,250,30);

        landmarkErr.setFont(fontError);landmarkErr.setForeground(Color.red);fbac.
add(landmarkErr);


                districtErr = new JLabel();districtErr.setBounds(630,220,250,30);

        districtErr.setFont(fontError);districtErr.setForeground(Color.red);fbac.add(d
istrictErr);


                stateErr = new JLabel();stateErr.setBounds(630,280,250,30);

        stateErr.setFont(fontError);stateErr.setForeground(Color.red);fbac.add(state
Err);



                branchErr = new JLabel();branchErr.setBounds(1000,130,250,30);

        branchErr.setFont(fontError);branchErr.setForeground(Color.red);fbac.add(b
ranchErr);
```

```
                /*
                 * password
                 */
                password = new JLabel("Enter
Password");password.setBounds(530,330,100,30);fbac.add(password);
                password.setVisible(false);
                confirmpassword = new JLabel("Confirm
Password");confirmpassword.setBounds(530,380,120,30);
                fbac.add(confirmpassword);confirmpassword.setVisible(false);
                /*
                 * Radio Buttons Male,Female,others
                 */
                bg = new ButtonGroup();
                male = new
JRadioButton("Male");male.setBounds(150,185,60,30);bg.add(male);fbac.add(male);


                female = new
JRadioButton("Female");female.setBounds(220,185,70,30);
                bg.add(female);fbac.add(female);


                other = new
JRadioButton("Other");other.setBounds(290,185,70,30);bg.add(other);fbac.add(othe
r);


                /*
                 * Conditions
                 */
                namecnd = new
JLabel();namecnd.setBounds(110,20,300,10);fbac.add(namecnd);
//              namecnd.setText("Enter the Full name in Uppercase *");
                namecnd.setFont(fontcnd);


                account_numbercnd = new
JLabel();account_numbercnd.setBounds(80,110,450,15);
                fbac.add(account_numbercnd);account_numbercnd.setFont(fontcnd);
```

```
//                account_numbercnd.setText("(Enter ur Mobile Number in above
Field,your giving Number is your Account Number)*");


                /*

                 * TextFields

                 */

                tname = new
JTextField();tname.setBounds(110,30,300,30);fbac.add(tname);

                tname.setForeground(Color.black);tname.setFont(fontField);


                taccount_number = new
JTextField();taccount_number.setBounds(110,80,170,30);


        fbac.add(taccount_number);taccount_number.setForeground(Color.black);ta
ccount_number.setFont(fontField);


                taadhaar = new
JTextField();taadhaar.setBounds(110,140,200,30);fbac.add(taadhaar);

                taadhaar.setForeground(Color.black);taadhaar.setFont(fontField);


                tcountry = new
JTextField();tcountry.setBounds(110,330,150,30);fbac.add(tcountry);

                tcountry.setForeground(Color.black);tcountry.setFont(fontField);


                tpincode = new
JTextField();tpincode.setBounds(365,330,100,30);fbac.add(tpincode);

                tpincode.setForeground(Color.black);tpincode.setFont(fontField);


                tDOB = new
JTextField();tDOB.setBounds(110,230,100,30);fbac.add(tDOB);

                tDOB.setForeground(Color.black);tDOB.setFont(fontField);


                tmail = new
JTextField();tmail.setBounds(110,280,300,30);fbac.add(tmail);

                tmail.setForeground(Color.black);tmail.setFont(fontField);
```

```
                thouseno = new
JTextField();thouseno.setBounds(110,380,70,30);fbac.add(thouseno);
                thouseno.setForeground(Color.black);thouseno.setFont(fontField);


                tstreet = new
JTextField();tstreet.setBounds(270,380,195,30);fbac.add(tstreet);
                tstreet.setForeground(Color.black);tstreet.setFont(fontField);



                tarea_loc = new
JTextField();tarea_loc.setBounds(630,30,300,30);fbac.add(tarea_loc);
                tarea_loc.setForeground(Color.black);tarea_loc.setFont(fontField);


                tvillage_t_c = new
JTextField();tvillage_t_c.setBounds(630,80,170,30);fbac.add(tvillage_t_c);

        tvillage_t_c.setForeground(Color.black);tvillage_t_c.setFont(fontField);


                tlandmark = new
JTextField();tlandmark.setBounds(630,140,170,30);fbac.add(tlandmark);
                tlandmark.setForeground(Color.black);tlandmark.setFont(fontField);


                tdistrict = new
JTextField();tdistrict.setBounds(630,200,150,30);fbac.add(tdistrict);
                tdistrict.setForeground(Color.black);tdistrict.setFont(fontField);


                tstate = new
JTextField();tstate.setBounds(630,260,100,30);fbac.add(tstate);
                tstate.setForeground(Color.black);tstate.setFont(fontField);



                /*
                 * Password Field
                 */
                tpassword = new JTextField();tpassword.setBounds(650,330,200,30);
```

```
                fbac.add(tpassword);tpassword.setFont(fontField);
                tpassword.setVisible(false);tpassword.setForeground(Color.black);

                tconfirmpassword = new
JTextField();tconfirmpassword.setBounds(650,380,200,30);
                fbac.add(tconfirmpassword); tconfirmpassword.setVisible(false);

        tconfirmpassword.setForeground(Color.black);tconfirmpassword.setFont(fo
ntField);
                /*
                 * check Button
                 */
                check = new
JButton("Check");check.setBounds(650,420,80,25);fbac.add(check);
//              check.setBackground(Color.red);check.setForeground(Color.white);
                check.addActionListener(this);
                /*
                 * Reset Button
                 */
                reset = new
JButton("Reset");reset.setBounds(650,420,80,25);fbac.add(reset);
//              reset.setBackground(Color.red);reset.setForeground(Color.white);
                reset.setVisible(false);
                reset.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent ae) {
                                String msg = "Are you Resetting ur Data";
                                int a = JOptionPane.showConfirmDialog(fbac,msg);
                                if(a == JOptionPane.YES_OPTION)
                                {
                                        submit.setVisible(false);
                                        reset.setVisible(false);
                                        tname.setEditable(true);
                                        taccount_number.setEditable(true);
                                        taadhaar.setEditable(true);
```

```
                                male.setEnabled(true);
                                female.setEnabled(true);
                                other.setEnabled(true);
                                tcountry.setEditable(true);
                                tpincode.setEditable(true);
                                tDOB.setEditable(true);
                                tmail.setEditable(true);
                                thouseno.setEditable(true);
                                tstreet.setEditable(true);
                                tarea_loc.setEditable(true);
                                tvillage_t_c.setEditable(true);
                                tlandmark.setEditable(true);
                                tdistrict.setEditable(true);
                                tstate.setEditable(true);

                                tpassword.setEditable(true);
                                tconfirmpassword.setEditable(true);
                                check.setVisible(true);
                                branches.setEditable(true);
                        }
                    }
                });
                /*
                 * Submit Button
                 */
                submit = new
JButton("Submit");submit.setBounds(760,420,80,25);fbac.add(submit);
//                  submit.setBackground(Color.red);submit.setForeground(Color.white);
                    submit.setVisible(false);
                    submit.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent ae) {
                            try
                            {
```

```
                                        acc_flag = 1;
                                        String msg = "Your Account is Created
Successfully,Thanks for Choosing our Bank";
                                        JOptionPane.showMessageDialog(fbac,msg);
                                        fbac.setVisible(false);
                                }
                                catch(Exception e){System.out.print(e);}
                        }
                });
                /*
                 * Back Button
                 */
                backbtn = new
JButton("Back");backbtn.setBounds(1050,420,80,25);fbac.add(backbtn);

        /backbtn.setBackground(Color.red);backbtn.setForeground(Color.white);/ba
ckbtn.setVisible(true);
                backbtn.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent ae)
                        {
//                              new BankLogin();
                                new BankMain();
                                fbac.setVisible(false);
                        }
                });

                fbac.setBackground(Color.PINK);
                fbac.setSize(1200,750);
                fbac.setLayout(null);
                fbac.setVisible(true);
                fbac.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
```

```
        }
        /*
         * Implementing the Check button
         */
        public void actionPerformed(ActionEvent ae)
        {
                /*
                 * pattern Matching for Names ,Phone Number,Aadhaar Number and
Pincode
                 */
                Pattern names = Pattern.compile("[A-Za-z]+");
                sname = tname.getText();
                Matcher mnames = names.matcher(sname);

                Pattern pphn = Pattern.compile("[6-9]{1}[0-9]{9}");
                saccno = taccount_number.getText();
                Matcher maccno = pphn.matcher(saccno);

                Pattern paadno = Pattern.compile("[1-9][0-9]{11}");
                saadno = taadhaar.getText();
                Matcher maddno = paadno.matcher(saadno);

                Pattern ppin = Pattern.compile("[1-9][0-9]{5}");
                spin = tpincode.getText();
                Matcher mpin = ppin.matcher(spin);

                // checking name field
                if(tname.getText().length() !=0 && mnames.find())
                {
                        nameErr.setText("");
                        sname = nameUpper = (tname.getText()).toUpperCase();

                }
```

```
                else{   nameErr.setText("please Enter your Name");      }
                // checking Account Number field
                if(taccount_number.getText().length() !=0 && maccno.find() )
                {
                        saccno = taccount_number.getText().trim();
                        laccno = Long.valueOf(saccno);
                        account_numberErr.setText("");
                }
                else{   account_numberErr.setText("Please Enter the Valid Mobile
Number");    }
                // checking Add Number field
                if(taadhaar.getText().length() !=0 && maddno.find())
                {
                        saadno = taadhaar.getText();
                        laadno = Long.valueOf(saadno);
                        aadhaarErr.setText("");
                }
                else{   aadhaarErr.setText("Please Enter the Valid Aadhaar Number");
        }
                // checking Gender is Selected or Not
                if(male.isSelected() || female.isSelected() || other.isSelected())
                {
                        genderErr.setText("");
                        if(male.isSelected())
                        {
                                sgender = male.getText();
                        }
                        if(female.isSelected())
                        {
                                sgender = female.getText();
                        }
                        if(other.isSelected())
                        {
```

```
                sgender = other.getText();
        }
}
else{   genderErr.setText("Please select your Gender");         }

// checking Country name  field
scountry = tcountry.getText();
Matcher mcountry = names.matcher(scountry);
if(tcountry.getText().length() != 0 && mcountry.find())
{
        countryErr.setText("");
}
else { countryErr.setText("Please Enter Country Name"); }
// checking  pin
if(tpincode.getText().length() == 6 && mpin.find()  )
{
        spin = tpincode.getText();
        ipincode = Integer.valueOf(spin);
        pincodeErr.setText("");
}
else{   pincodeErr.setText("Please Enter valid Pincode");       }

// checking  Date of Birth and pattern matching
Pattern pDOB = Pattern.compile("[0-9]{2}[-][0-9]{2}[-][1-2][0-9]{3}");
sDOB = tDOB.getText();
Matcher mDOB = pDOB.matcher(sDOB);

if(tDOB.getText().length() == 10 && mDOB.find())
{
        bDOB = true;
        sDOB = tDOB.getText();
        DOBErr.setText("");
```

```
}
else{   DOBErr.setText("Please Enter  Valid Date of Birth");     }
// checking  Mail and Matching the some pattern
Pattern pmail = Pattern.compile("[a-zA-Z0-9_-]+@[a-z]+[.][a-z]{2,3}");
Matcher mmail = pmail.matcher(tmail.getText());
int flag = 0;
if(tmail.getText().length() !=0 && mmail.find())
{
        smail = tmail.getText();
        flag = 1;
        mailErr.setText("");
}
else{   mailErr.setText("Please Enter Valid Mail");         flag = 0;         }
//checking House Number
Pattern houseno = Pattern.compile("[A-Z]*[0-9]+");
shouseno = thouseno.getText();
Matcher mhouse = houseno.matcher(shouseno);
if(thouseno.getText().length() != 0 && mhouse.find())
{
        housenoErr.setText("");
}
else { housenoErr.setText("Please Enter House Number");     }
//checking Street Name
sstreet = tstreet.getText();
Matcher mstreet = names.matcher(sstreet);
if(tstreet.getText().length() != 0 && mstreet.find())
{
        streetErr.setText("");
}
else { streetErr.setText("Please Enter Street Name");    }
//checking Area/Locality
sarea_loc = tarea_loc.getText();
```

```
        Matcher marea_loc = names.matcher(sarea_loc);
        if(tarea_loc.getText().length() != 0 && marea_loc.find())
        {
                area_locErr.setText("");
        }
        else { area_locErr.setText("Please Enter Area or Locality Name");      }
        //checking Village/Town/City
        svillage = tvillage_t_c.getText();
        Matcher mvillage = names.matcher(svillage);
        if(tvillage_t_c.getText().length() != 0 && mvillage.find())
        {
                village_t_cErr.setText("");
        }
        else { village_t_cErr.setText("Please Enter Village/Town/City Name");
}
        //checking Landmark
        slandmark = tlandmark.getText();
        Matcher mlandmark = names.matcher(slandmark);
        if(tlandmark.getText().length() != 0 && mlandmark.find())
        {
                landmarkErr.setText("");
        }
        else { landmarkErr.setText("Please Enter Landmark "); }
        //checking District
        sdistrict = tdistrict.getText();
        Matcher mdistrict = names.matcher(sdistrict);
        if(tdistrict.getText().length() != 0 && mdistrict.find())
        {
                districtErr.setText("");
        }
        else { districtErr.setText("Please Enter District Name");}
        //checking State
        sstate = tstate.getText();
```

```
Matcher mstate = names.matcher(sstate);
if(tstate.getText().length() != 0 && mstate.find())
{
        stateErr.setText("");
}
else { stateErr.setText("Please Enter State Name");      }

//Branch Selection
if(branches.getSelectedIndex() != 0)
{
        branchErr.setText("");
        if(branches.getSelectedIndex() == 1)
                ssbranch = "STUN0000091" ;
        if(branches.getSelectedIndex() == 2)
                ssbranch = "STUN0000093" ;
        if(branches.getSelectedIndex() == 3)
                ssbranch = "STUN0000094" ;
        if(branches.getSelectedIndex() == 4)
                ssbranch = "STUN0000128" ;
        if(branches.getSelectedIndex() == 5)
                ssbranch = "STUN0000129" ;
        if(branches.getSelectedIndex() == 6)
                ssbranch = "STUN0000332" ;
        if(branches.getSelectedIndex() == 7)
                ssbranch = "STUN0000590" ;
        if(branches.getSelectedIndex() == 8)
                ssbranch = "STUN0123456" ;
}
else { branchErr.setText("Please Select your Branch ");         }


//invoking submit button if all conditions r passed
```

```
                if(             (tname.getText().length() !=0)
                        &&      (taccount_number.getText().length() !=0 )
                        &&      (taadhaar.getText().length() !=0 )
                        &&      (male.isSelected() || female.isSelected() ||
other.isSelected())
                        &&      (tDOB.getText().length() == 10 )
                        &&      (tmail.getText().length() > 10 && flag == 1)
                        &&      (tcountry.getText().length() !=0)
                        &&      (tpincode.getText().length() !=0 )
                        &&  (thouseno.getText().length() != 0)
                        &&      (tstreet.getText().length() !=0)
                        &&      (tarea_loc.getText().length() !=0)
                        &&  (tvillage_t_c.getText().length() != 0)
                        &&      (tlandmark.getText().length() !=0)
                        &&      (tdistrict.getText().length() !=0)
                        &&  (tstate.getText().length() != 0)

                        &&  (branches.getSelectedIndex() != 0)
                  )
                {
                        password.setVisible(true);
                        tpassword.setVisible(true);
                        confirmpassword.setVisible(true);
                        tconfirmpassword.setVisible(true);
                        spass = tpassword.getText();
                        scpass = tconfirmpassword.getText();
                        mailErr.setText("");

                        if((spass.length() != 0) && (scpass.length() != 0))
                        {
                                if(scpass.equals(spass))
                                {
                                        scpassword = scpass.trim();
```

```
                String msg = "Are u sure about your giving
Data,if u click YES there is no more Editable Option";
                int a =
JOptionPane.showConfirmDialog(fbac,msg);
                if(a == JOptionPane.YES_OPTION)
                {
                        check.setVisible(false);
                        submit.setVisible(true);
                        reset.setVisible(true);
                        tname.setEditable(false);
                        taccount_number.setEditable(false);
                        taadhaar.setEditable(false);
                        male.setEnabled(false);
                        female.setEnabled(false);
                        other.setEnabled(false);
                        tcountry.setEditable(false);
                        tpincode.setEditable(false);
                        tDOB.setEditable(false);
                        tmail.setEditable(false);
                        thouseno.setEditable(false);
                        tstreet.setEditable(false);
                        tarea_loc.setEditable(false);
                        tvillage_t_c.setEditable(false);
                        tlandmark.setEditable(false);
                        tdistrict.setEditable(false);
                        tstate.setEditable(false);

                        tpassword.setEditable(false);
                        tconfirmpassword.setEditable(false);

                        branches.setEditable(false);
                }
        }
```

```
                    else
                    {
                        submit.setVisible(false);
                        reset.setVisible(false);

        JOptionPane.showMessageDialog(fbac,"Password NOT
Matching","Alert",JOptionPane.WARNING_MESSAGE);
                    }
                }
            }
            else
            {
                password.setVisible(false);
                tpassword.setVisible(false);
                confirmpassword.setVisible(false);
                tconfirmpassword.setVisible(false);
            }
        }
        public static void main(String agrs[])
        {
            new AccountCreation();
        }
}
```

1.3--Bank Account exception.java

```
package onlinebankingapplication;

class BankAccountException extends Exception
{
        public BankAccountException()
        {
            super();
```

```
        }
        public BankAccountException(String message)
        {
                super(message);
        }
}
```

1.4--Account.java

```
//package BankAccount;

public class Account extends AccountCreation
{
//    private double balance;
//    private long accNum  ;
//    private String accHolder;
//
//     public Account(String name, long num)
//     {
//          accHolder = name;
//         accNum = laccno;
//         balance = 0.00;
//          System.out.println("Account Created");
//     }
//
//     public void deposit(double amt)
//     {
//          balance = balance + amt;
//          System.out.println("Amount Deposited.");
//     }
//
//     public void withdraw(double amt) throws BankAccountException
//     {
```

```
//          try
//          {
//            if(balance >= amt)
//            {
//                        balance = balance - amt;
//            System.out.println("Amount Withdrawn");
//            }
//            else
//            {
//                        throw new BankAccountException("Could not Withdraw:
Insufficent Amount.");
//            }
//          }
//          catch(BankAccountException bae)
//          {
//            bae.printStackTrace();
//        }
//    }
//
//    public void printBalance()
//    {
//          System.out.println("Current balance: " + balance);
//    }
//    public double getBalance()
//    {
//        return balance;
//    }
}
```

RESULT:

Hence, “Online Banking Application “ mini- project is successfully developed and executed used both front end and bank end programming.

Csdfv b

bnmjhgtrfdwewrtryujk