INTRODUCTION

Our project "Self Help Group Management System" is an attempt to automate the working of overall organization. It forms an integral component in every one's life as the need of money any person can make small investment daily and can get the loan with the lowest interest. It offers wide range of programmes and services to enhance personal growth and self-development. To overcome the limitation of existing system, I am trying to create software, which give facility for reporting or access any type of information about the members and the all investments, transactions and loans. This software will be great relief to the admin for reporting, tracking record and searching the information about members.

This software gives an overview of the entire transactions in a sort interval time and will be great help to grant the loan procedure by their paper work and reducing their time.

The goal of our project is to provide assistance and guidance to admin for making their plans and grow their investment to the self-help group. This Self-Help Group management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also provides security check to reduce fraud. The system will check the user's existence in the database and provide the set of services with respect to the role of the user. The main purpose of developing Self Help Group management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

OBJECTIVE

This Self-Help Group management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also, provide security check to reduce fraud. The system will check the user's existence in the database and provide the set of services with respect to the role of the user. The main purpose

of developing Self Help Group management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

This software gives an overview of the entire transactions in a sort interval time and will be great help to grant the loan procedure by their paper work and reducing their time.

- To allow only authorized user to access various function and processed available in the system.
- Locate any A/C wanted by the user.
- Provide loan and determine loan instalment automatically.
- To transactions of money either debited or credited to their accounts
- Reduced clerical work as most of the work done by computer.
- Provide greater speed & reduced time consumption

In this project we have automate the bank process like Account Opening, Daily Transactions, Loan Sanctions, and Account Maintenance. In this group, management system use can also search record of a particular Account Holder.

PROJECT CATEGORY

1. RDBMS

A relational database management system (RDBMS) is a database engine/system based on the relational model specified by Edgar F. Codd--the father of modern relational database design--in 1970. A relational database management system (RDBMS) is a program that lets you create, update, and administer a relational database. Most commercial RDBMS's use the Structured Query Language (SQL) to access the database, although SQL was invented after the development of the relational model and is not necessary for its use. Virtually all relational database systems use SQL (Structured query Language) for querying and maintaining the database.

Most modern commercial and open-source database applications are relational in nature. The most important relational database features include an ability to use tables for data storage while maintaining and enforcing certain data relationships.

The leading RDBMS products are Oracle, IBM's DB2 and Microsoft's SQL Server. Despite repeated challenges by competing technologies, as well as the claim by some experts that no current RDBMS has fully implemented relational principles, the majority of new corporate databases are still being created and managed with an RDBMS.

2. OOPS

Object-oriented programming (OOP) is a programming language model organized around objects rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.

The programming challenge was seen as how to write the logic, not how to define the data. Object-oriented programming takes the view that what we really care about are the objects we want to manipulate rather than the logic required to manipulate them. Examples of objects range from human beings (described by name, address, and so forth) to buildings and floors (whose properties can be described and managed) down to the little widgets on a computer desktop (such as buttons and scroll bars).

The first step in OOP is to identify all the objects the programmer wants to manipulate and how they relate to each other, an exercise often known as data modelling . Once an object has been identified, it is generalized as a class of objects (think of Plato's concept of the "ideal" chair that stands for all chairs) which defines the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. Objects communicate with well-defined interfaces called messages.

SCOPE

Self Help Group activities are considered to be the life blood of the national economy . Without banking services, trading and business activities cannot be carried on smoothly. Self Help Groups are the distributors and protectors of liquid capital which is of vital significance to a developing country.

Efficient administration of the banking system helps in the economic Growth of the nation. Self Help Group is useful to trade and commerce.

Manage large number of customer details with ease. Particular A/c information can be modified A particular customer record can be modified for one or more field’s customer name, address by providing A/c number.

Create a statistical report to facilitate the finance department work. Activities like updating, modification, deletion of records should be easier. A customer record can be easily deleted by authorize user by providing A/c number.

The proposed system provides faster data access, data entry and retrieval. Using this system user can create new customer account and maintain its data efficiently and effectively. All records of account and customer are stored in separate files. Which are maintained constantly update by system.

The proposed system is more efficient, fast, reliable, user friendly. Over and above the proposed system does not have any possibility of data loss during processing.

Modules

I am working on a real-life project, which is a complex task. To deal with complex project and its functions we have to classify these modules into its further sub-modules.

1. Login
2. Members
3. Loan
4. Loan Payment
5. Transactions
6. Admin

Maintaining the Member.

Add Member Details

Update Member Details

Delete Member Details

Track Member Status

Maintaining the Loan details.

Loan Sanctions

Settle Loan

Loan Status

Maintaining the Transaction.

Add Transactions (Debits/Credits)

Track Transactions

Maintaining the Loan Payments.

Pay Loan

Track Loan

Maintaining the User login

New user

Update user

Delete user

TOOLS AND PLATFORMS USED

Platform:

It is the system infrastructure, which is required for deployment of project. It, includes hardware and software requirements.

Hardware:

1. Dual core processor or above

2. 512 MB RAM or above

3. 10 GB HDD or above

4. Graphics Monitor

5. Keyboard

6. Laser Mouse

7. Printer(any type)

SOFTWARE:

1. Windows O.S (windows 7 or above)

2. JDK

TOOLS:-

1. Java (front end)

2. MySQL (back end)

REASON FOR SELECTING TOOLS

Java (Front-end tool):-

Due to following features in java, I have selected it as my projects front-end tool:-

- ✓ Platform Independent : can be deployed on any machine.
- ✓ UNICODE: It will work on more character set than ASCII and can be compatible with several languages.
- ✓ Security : true object-oriented programming will create a secured architecture
- ✓ Multithreading: will be able to process simultaneous request when used over a network.

+++

MySQL (back-end tool):-

MySQL supports 10 CODD rules and therefore we can say that it is nearest to being a complete RDBMS.

Following features of MySQL has convinced me to choose MySQL as a backend tool of my project:-

- ✓ Easy secured database
- ✓ Multiuser support
- ✓ Fast query execution
- ✓ Compatibility with java
- ✓ Most suitable for implementation PL/SQL
- ✓ Support rich library functions
- ✓ Goodwill

LIMITATION

- The failure of a larger institution could have serious ramifications for the entire system in that if one universal bank were to collapse, it could lead to a systemic financial crisis.
- Universal bankers may be tempted to take excessive risks.
- Self Help Group may deploy their own assets in securities with consequent risk to commercial and saving deposits.
- Vulnerable to high risks due to investment Self Help Group activities coupled with focus on commercial banking activities.
- Unsound loans may be made in order to shore up the price of securities or the financial position of companies in which a Self-Help Group had invested its own assets.

PROPOSED WORK

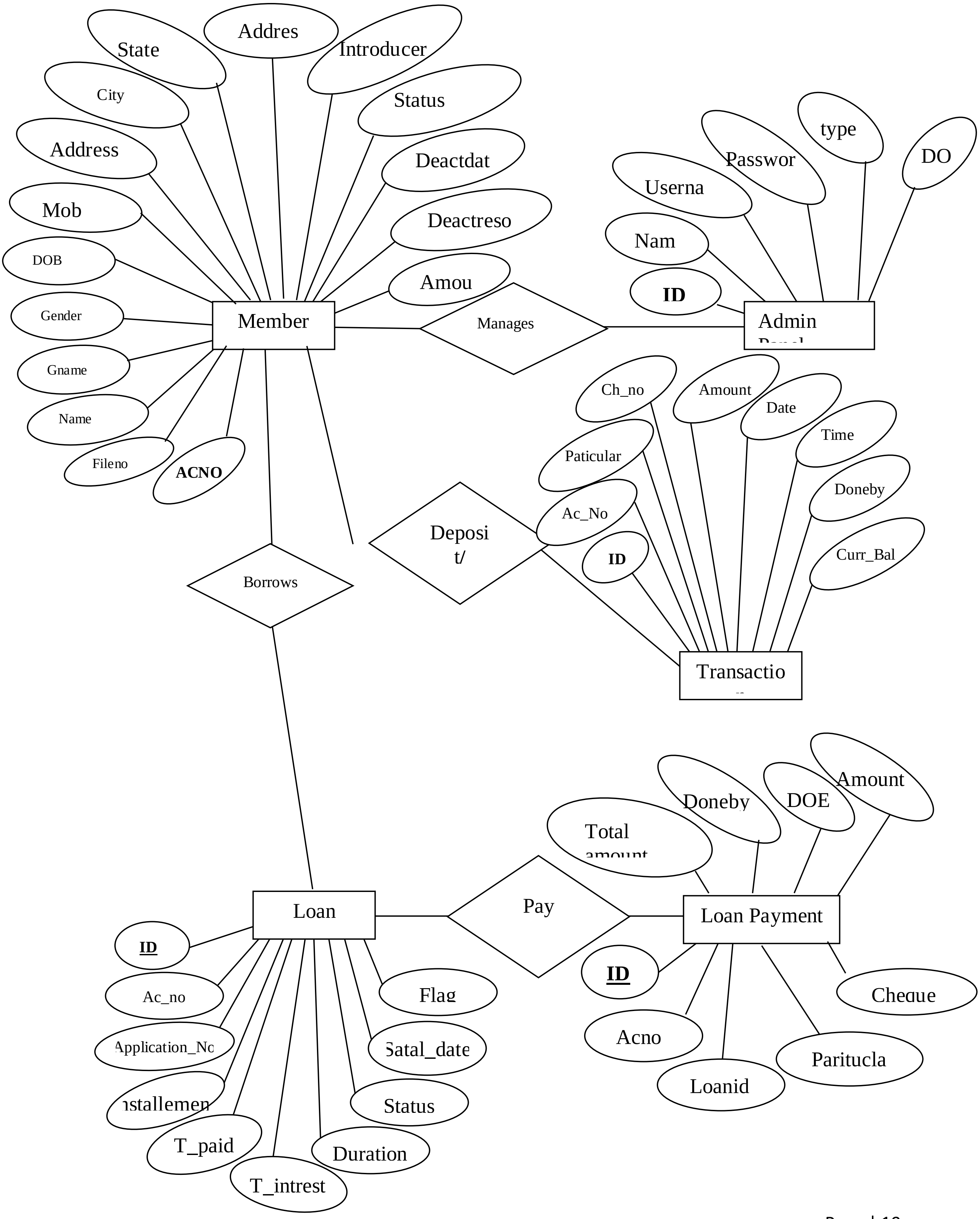
To find more about present system’s working mechanism such as the ways of getting inputs and providing outputs, I interviewed the current users of the system by asking following question:

- Are they comfortable with the present system?
- What flaws exists in the current system?
- Do they feel the necessity of new system?
- What will be their requirements from new system?
- Are they satisfied with their role in new system?

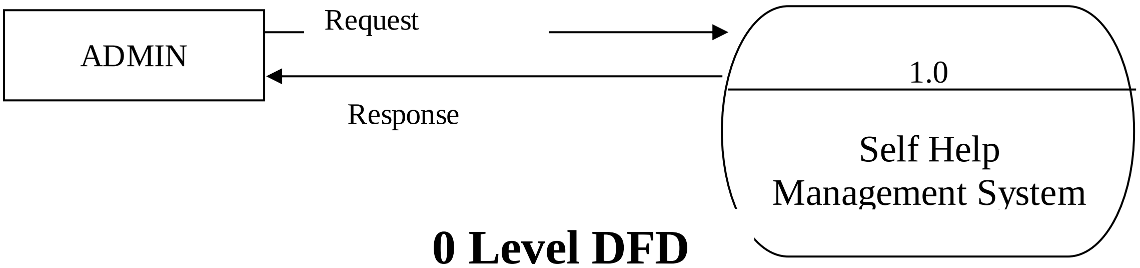
After carrying out these interviews, I drew conclusion about the Top Management’s requirements and whether users are in support of the new system.

This billing software is technically feasible. The Primary technical includes the availability of windows 7 or higher version of operating system installed in the network. MS Access is also required which was already installed. To develop program java8 was required which was also available. Reliability, access power and data security were also available. The system can also be developed if the new technology is acquired. Thus, through all the ends technical feasibility was met.

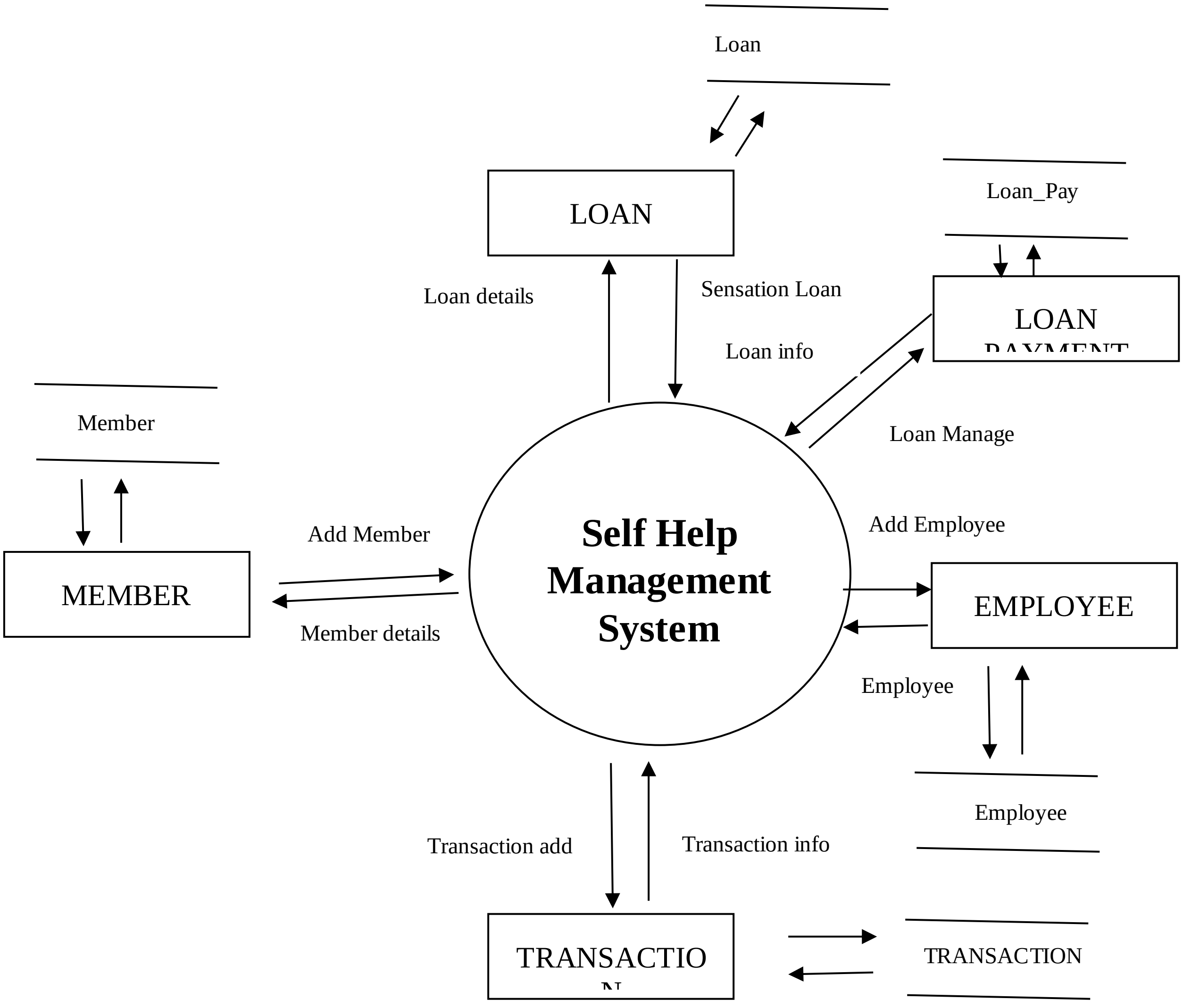
ENTITY RELATIONSHIP DIAGRAM(E.R DIAGRAM)



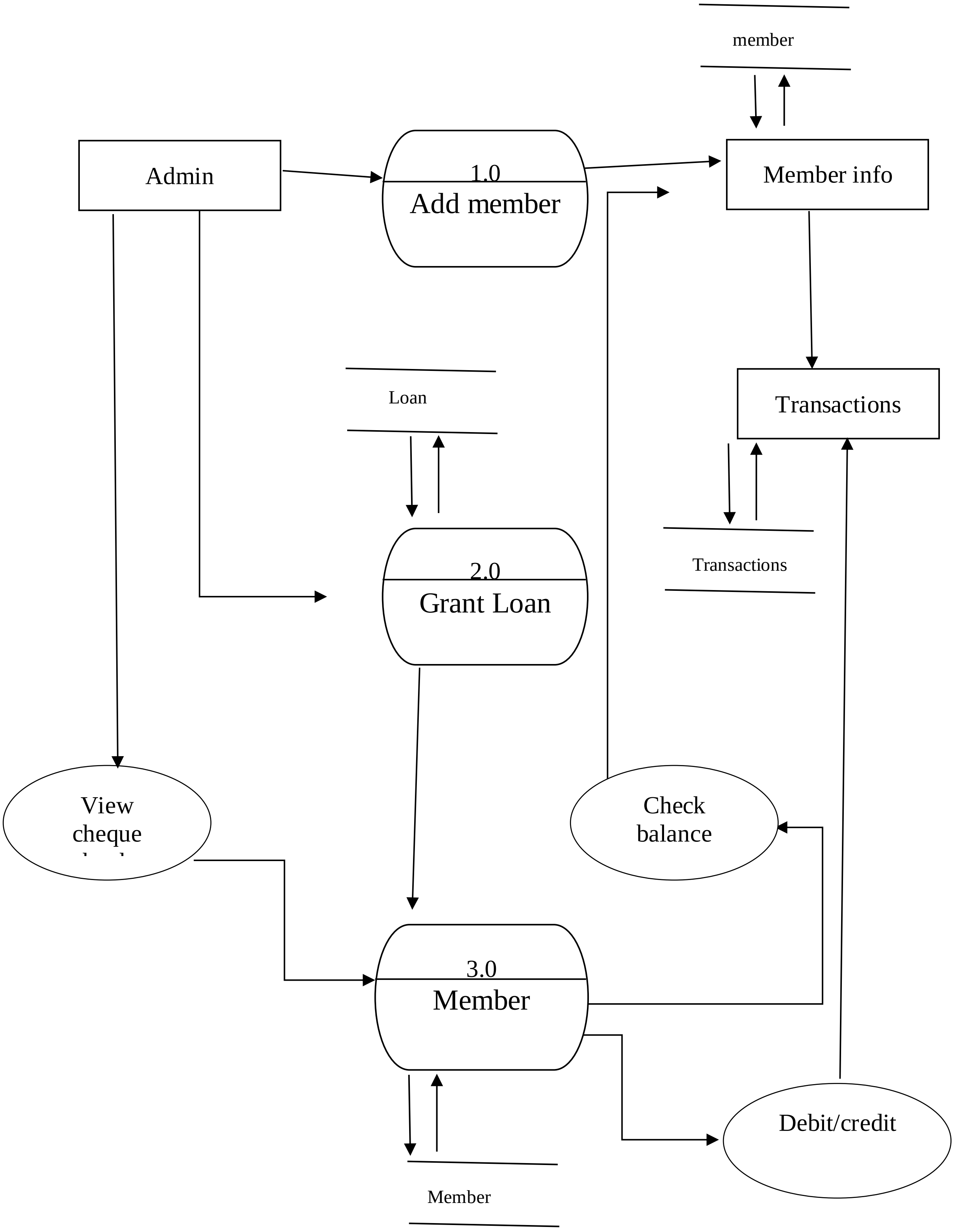
DATAFLOW DIAGRAM(D.F.D)



0 Level DFD



1st LEVEL DFD


member
Admin
1.0
Add member
Member info
Loan
Transactions
2.0
Grant Loan
Transactions
View
cheque
Check
balance
3.0
Member
Debit/credit
Member

2nd LEVEL

TABLE

TABLE : LOGIN

FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
ID	INT	5	PRIMARY KEY
NAME	VARCHAR	30	
USERNAME	VARCHAR	15	UNIQUE
PASSWORD	VARCHAR	12	ENCRYPTED
TYPE	VARCHAR	10	ADMIN/ EMLOYEE
ADDRESS	VARCHAR	35	
MOBILENO	INT	10	
DATE	DATETIME		

TABLE : MEMBERDETAIL

FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
ACNO	INT	6	PRIMARY KEY
FILENO	INT	5	
FNAME	VARCHAR	15	
MNAME	VARCHAR	15	
LNAME	VARCHAR	15	
GTYPE	VARCHAR	15	
GNAME	VARCHAR	25	
MARITALSTATUS	VARCHAR	10	
GENDER	VARCHAR	6	
DOB	DATE		DD/MM/YYYY
CATEGORY	VARCHAR		
MOBILE_NO	INT	10	
EMAIL	VARCHAR	30	
ADDRESS	VARCHAR	50	
CITY	VARCHAR	20	
STATE	VARCHAR	20	

ADDRESSP	VARCHAR	45	
INTROAC1	INT	6	
INTROAC2	INT	6	
NAMEN	VARCHAR	30	
DOBN	DATE		DD/MM/YYYY
ADDRESSN	VARCHAR	45	
GENDERN	VARCHAR	10	
OPENEDBY	VARCHAR	15	
OPENEDDATE	DATETIME		DD/MM/YYYY
STATUS	VARCHAR	10	
DEACTIVATEDATE	DATE		DD/MM/YYYY
DEACTIVATEREASON	VARCHAR	50	
AMOUNT	FLOAT		

TABLE : TRANSACTION

FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
ID	VARCHAR	6	PRIMARY KEY
ACNO	INT		
PATICULAR	VARCHAR	6	
CHEQUENO	VARCHAR	20	
AMOUNT	INT		
DATE	DATE		YYYY-MM-DD
TIME	TIME		HH:MM:SS
DONEBY	VARCHAR	30	
CURRENTBALANCE	INT		

TABLE : LOAN

FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
ID	INT		PRIMARY KEY
ACNO	INT		
DATE	DATE		
TIME	TIME		
LOANAMOUNT	INT	8	
DURATION	INT	3	
INTEREST	FLOAT		
TOTALINTEREST	FLOAT		
INSTALLMENT	FLOAT		
TOTALAMOUNT	FLOAT		
PAID TILL	DATE		DD/MM/YYYY
PURPOSEOFLOAN	VARCHAR	60	
GRANTERACNO	INT	6	
TOTALINCOME	INT		
DONEBY	VARCHAR	15	
STATUS	VARCHAR	10	
TOTALPAIDAMOUNT	INT		
SETTLEMENTDATE	DATE		
REMARK	VARCHAR	60	
FLAG	INT		

TABLE : LOAN_PAYMENY

FIELD NAME	DATA TYPE	FIELD SIZE	DESCRIPTION
ID	INT		PRIMARY KEY
ACNO	INT		
LOANID	INT		
PARTICULAR	VARCHAR	10	
CHEQUENO	VARCHAR	10	
AMOUNT	INT		
DOE	DATE		DD/MM/YYYY
TOTALAMOUNTPAID	INT		
DONEBY	VARCHAR	15	

ANALYSIS & FEASIBILITY STUDY

REQUIREMENT ANALYSIS

System Analysis: Analysis of system is a much important task to lookup with Without sufficient analysis of requirements the firm could not be converted into a system. The job of system analysis is done on several factors which I found would be necessary to solve every, complainer's issues are resolved on due time. The maximum approach behind analysis is for the sake of convince and service requirements of the customer. So, the analysis is done from customer's point of view and applied to design of system architecture.

Several tools and design approach has been applied to the working aspect for the solution centre working efficiency as per desired specification and service categories of demand.

While the detailed analysis of complaint management system's workflow and possible inputs I have selected these analysis approaches:-

System Study

1. Existing System
2. Proposed System
3. Outcomes from the proposed System

1.1 Existing System:-

There was an existing system, which was totally comprised on paper work. The system used to record complaints through manual mode of working methodologies. The database was used to be updated and manipulated every year into a file-based record. In file-based record data redundancy and lack of security was a common phenomenon.

This was needed to be automated and computerized as per the change in time and demand of technology.

1.2 Proposed System:-

There is a lot of tasks in existing system that needs to be modified as per system requirements deployed on computer system. For this the traditional file-based system needs to be designed as per relational database requirements. The work of creating a platform in the form of user friendly and meaningful GUI (Graphical User Interface) where the user will be aware from the tasks and services more easily.

1.3 Outcome from the proposed system:-

✓ Easily usable user interface that will make task for complaint resolution more easily and accurately.

✓ Necessary integration checks for performing reliable data handling approach.

✓ User authentication feature enables the system secure and enhance durability of the system.

✓ System will become robust and automated.

✓ Maximum complaint registration and its effective resolution will be provided on due time.

✓ Since MySQL is back-end for this system data can be linked over several platforms which will verify the front-end of the system (Java) i:e platform independency.

Output Analysis

The phase results in described model of the complaint management system. The prototype describes system data, methods and system flow of data. This phase consists of reports that will prescribe the cost and relevant benefit analysis respectively.

Hence, I went through the analysis of several prototype data models for Complaint Management System(Simplex Solutions Pvt Ltd) .These are as following:-

Identification of need

Whenever there is an opportunity to develop a system, the organizations usually face problems or have opportunity due to the following:

- Rapid complaint registration and creation of Invoice/receipt
- To record the customer profile, spares and components information
- Inefficiency of an existing system
- Structural error in the existing system, etc.

Thus, a thorough analysis of the situation is required. So, to short-out the problems, I have scanned the following:

- The information being supplied and its form
- The control of the information processing
- Data security.

Preliminary Investigation

When the request is made, the first system activity "***Preliminary Investigation***" begins. Its purpose is to evaluate project request. It is the first step in Software Development Life Cycle (SDLC) to determine the feasibility of the system.

For this, I have scanned the request made by the organisation. I have gone through all the preliminary investigation stage.

- Clarity and understand the project request
- Determine the size of the project

- Determine the technical and operational feasibility of alternative approaches

System Development Life Cycle

System development Life cycle consist of two major steps of System analysis and design. It includes set of activities that analysts, designers and users carry out to develop and implement an information system. So, here we followed the different stages of system development life cycle to develop Billing Software efficiently.
The system development life cycle consists of the following activities:

1. Preliminary Investigation
2. Determination Requirements
3. Design of System
4. Development of System
5. System Testing
6. Implementation

Preliminary Investigation

For this, the need arises to understand the viewpoint of two important entities…
Top management and users
In order to gather pertinent information, I interviewed the Top Management and asked the following questions:

- How the present system works?
- What all drawbacks are in the present system?
- What is their vision about the new system?
- What specific facilities they want from new system, those are currently not in scope of existing system?

- How will data flow in the system?
- Who will be authenticated to access data and his/her access rights?

To find more about present system's working mechanism such as the ways of getting inputs and providing outputs, I interviewed the current users of the system by asking following question:

- Are they comfortable with the present system?
- What flaws exists in the current system?
- Do they feel the necessity of new system?
- What will be their requirements from new system?
- Are they satisfied with their role in new system?

After carrying out these interviews, I drew conclusion about the Top Management's requirements and whether users are in support of the new system.

This activity is consisting of three parts:-

Feasibility Study

Feasibility study tried to determine whether a given solution would work or not. Its main objective is not to solve the problem, but to acquire its scope. It focuses on following:

- Meet user requirements
- Best utilization of available resources
- Develop a cost-effective system
- Develop a technically feasible system

It is further classifying into three aspects: -

Technical Feasibility

Issues to be studies are, whether the work for the project will be done with current equipment, existing S/W technology and available personnel? If the new technology is required, then what is the likelihood that is can be developed?

This billing software is technically feasible. The Primary technical includes the availability of windows 7 or higher version of operating system installed in the network. MS Access is also required which was already installed. To develop program java8 was required which was also available. Reliability, access power and data security were also available. The system can also be developed if the new technology is acquired. Thus, through all the ends technical feasibility was met.

Economic Feasibility

Issues to be studied are, whether the new system is cost effective or not? The benefits in the form of reduced cost?

This billing software is economically feasible. As the hardware was installed form quite beginning, the cost on project of hardware is low. Similarly, the software loaded for this project was used even after this project was developed for many other applications. The software cost was under budget. Moreover, the technical requirements were already available so there was no further expenditure for buying software packages.

Operational Feasibility

Issues to be studied are, is there sufficient support for management and users? Is the current method acceptable to users? Will the proposed system cause any harm?

This billing software is operationally feasible. This application provides the necessary information to the user such as how to enter the information regarding different operations performed on the database. The application was planned in such a way that no prior knowledge was required to go through the various operations. The user just needed to have the basic knowledge of computers.

This software also possesses behavioral feasibility. It’s because the users of the system are computer professionals and know the advantages of computer system. Moreover, this application is going to enhance their skills and reduce their workloads.

Request Approval

Not all the requested projects are desirable or feasible. However, those that are both feasible and desirable should be put into scheduled. After the project is approved, its cost, priority, completion time and personnel requirements are estimated and used to determine where to add it to any project list.

The top management reviewed the above factors and gave their ascent. After the consent to request approval, the design was taken up.

RESULTS & DISCUSSION

The application was planned in such a way that no prior knowledge was required to go through the various operations. The user just needed to have the basic knowledge of computers.

This software also possesses behavioral feasibility. It’s because the users of the system are computer professionals and know the advantages of computer system. Moreover, this application is going to enhance their skills and reduce their workloads.

The phase results in described model of the complaint management system. The prototype describes system data, methods and system flow of data. This phase consists of reports that will prescribe the cost and relevant benefit analysis respectively.

There is a lot of tasks in existing system that needs to be modified as per system requirements deployed on computer system. For this the traditional file-based system needs to be designed as per relational database requirements.

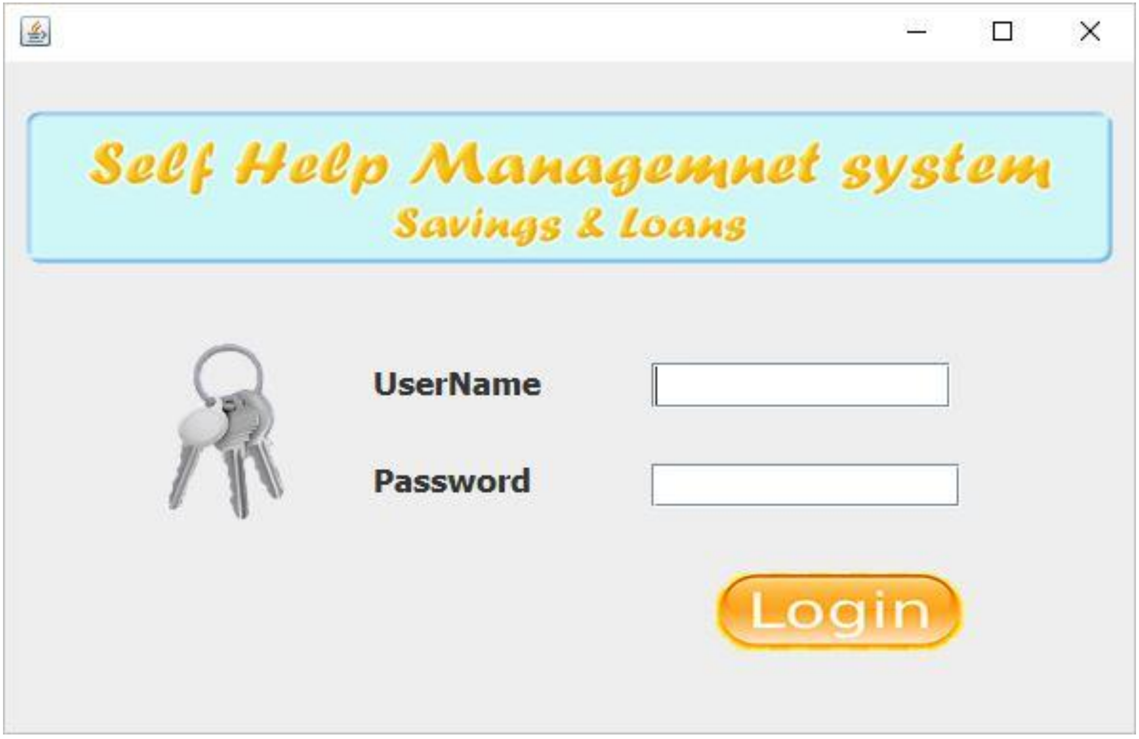
FORMS AND CODING FOR FRONT-END

DbConnection

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

public class dbconnection {
    public Statement conmethod(){
        Statement stmt=null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
    //register ur driver
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/sss?charact
erEncoding=UTF-8","root","nknknk");
            stmt= con.createStatement();


        } catch (Exception e) {
            e.printStackTrace();

            // TODO: handle exception
        }
        return stmt;

    }


}
```

LOGIN FORM

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JLabel;

import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

import javax.swing.JPasswordField;
import java.awt.Color;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.Cursor;


public class login extends JFrame {

    private JPanel contentPane;
    private JTextField username;
    dbconnection dbc = new dbconnection();
    private JTextField password;

    /**
     * Launch the application.
     */

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    login frame = new login();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public login() {
        setBackground(Color.LIGHT_GRAY);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 608, 391);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        username = new JTextField();
        username.setFont(new Font("Tahoma", Font.PLAIN, 17));
        username.setBounds(339, 158, 157, 24);
        contentPane.add(username);
        username.setColumns(10);

        JLabel lblUsername = new JLabel("UserName");
        lblUsername.setFont(new Font("Tahoma", Font.BOLD, 17));
        lblUsername.setBounds(193, 158, 96, 20);
        contentPane.add(lblUsername);

        JLabel lblPassword = new JLabel("Password");
        lblPassword.setFont(new Font("Tahoma", Font.BOLD, 17));
        lblPassword.setBounds(193, 209, 96, 20);
        contentPane.add(lblPassword);

        password = new JPasswordField();
        password.setFont(new Font("Tahoma", Font.PLAIN, 17));
        password.setBounds(339, 211, 162, 23);
        contentPane.add(password);

        JLabel lblNewLabel = new JLabel(new
ImageIcon("image/logo1.png"));
        lblNewLabel.setFont(new Font("Source Sans Pro Semibold",
Font.PLAIN, 34));
        lblNewLabel.setBounds(10, 26, 572, 80);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_2 = new JLabel(new
ImageIcon("image/log.png"));
        lblNewLabel_2.setBounds(60, 124, 112, 138);
        contentPane.add(lblNewLabel_2);

        JLabel lblNewLabel_1 = new JLabel(new
ImageIcon("image/login.png"));

    lblNewLabel_1.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_C
URSOR));
        lblNewLabel_1.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
               logincode();
            }
        });
        lblNewLabel_1.setBounds(371, 266, 135, 45);
        contentPane.add(lblNewLabel_1);
    }
    private void logincode() {
        String uname=username.getText();
        String upass=password.getText();
        String sql=("select * from login");
        if (uname.equals("") || upass.equals("")) {
            reset();
            JOptionPane.showMessageDialog(null, "Username or Password
must required");
        }else{
            try {
               Statement stmt=dbc.conmethod();
               ResultSet rs=stmt.executeQuery(sql);
               int count =0;
               while(rs.next()){
                  String user=rs.getString("username");
                  String pass=rs.getString("password");
                  String username=rs.getString("name");
                  if((uname.equals(user))&&(upass.equals(pass))){
                           count+=1;
                           new mainpage(username).setVisible(true);
                           dispose();
                  }
               }
               if (count==0){
                  reset();
                  JOptionPane.showMessageDialog(null, "Invalid user or
password");
               }
            } catch (Exception e){
JOptionPane.showMessageDialog(null, e);}
        }
    }
    public void reset(){
        username.setText("");
        password.setText("");
        username.requestFocus(true);
    }
}
```



MAIN PAGE

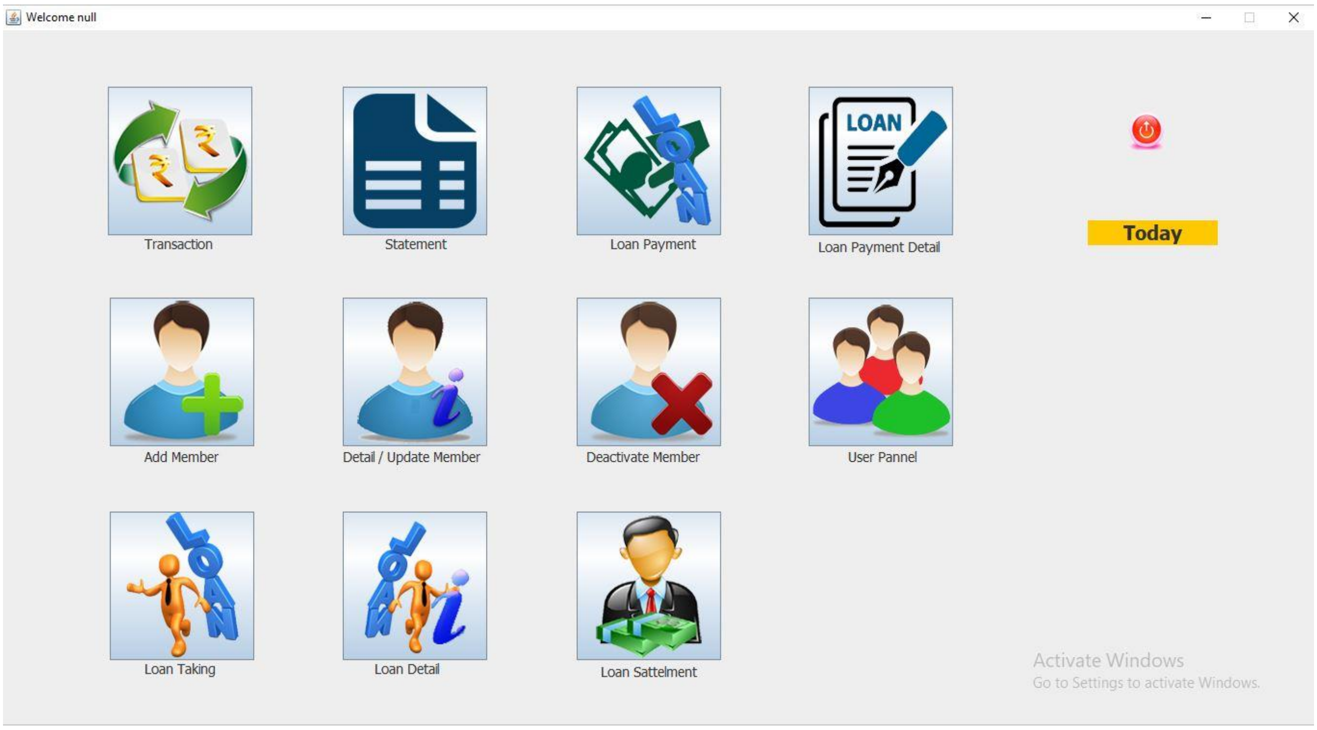
```java
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.EventQueue;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import java.awt.Frame;

import javax.swing.JButton;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;

import javax.swing.JLabel;
import java.awt.Color;
import java.awt.Cursor;
import javax.swing.SwingConstants;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;


public class mainpage extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JLabel lblTxtTCrAmount;
    private JLabel lblTxtTDrAmount;
    private JLabel lblTxtTLPAmount;
    private JLabel lblTxtLoanSatteled;
    private JLabel lblTxtLoanSaction;
    private JLabel lblTxtTransaction;

    java.util.Date datet= Calendar.getInstance().getTime();
    DateFormat formatter= new SimpleDateFormat("yyyy-MM-dd");
    String today = formatter.format(datet);


    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    mainpage frame = new mainpage(null);
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public mainpage(String username) {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        setResizable(false);
        setTitle("Welcome "+username);
        setSize(new Dimension(1370,730));
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JButton btnAddMember = new JButton(new
ImageIcon("image/member.png"));

    btnAddMember.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CU
RSOR));
        btnAddMember.setFont(new Font("Tahoma", Font.PLAIN, 16));
        btnAddMember.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new addmember(username).setVisible(true);
            }
        });
        btnAddMember.setBounds(111, 270, 150, 150);
        contentPane.add(btnAddMember);

        JButton btnMemberDetail = new JButton(new
ImageIcon("image/memberinfo.png"));

    btnMemberDetail.setCursor(Cursor.getPredefinedCursor(Cursor.HAND
_CURSOR));
        btnMemberDetail.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new memberdetail().setVisible(true);
            }
        });
        btnMemberDetail.setFont(new Font("Tahoma", Font.PLAIN, 16));
        btnMemberDetail.setBounds(353, 270, 150, 150);
        contentPane.add(btnMemberDetail);

        JButton btnDeleteMember = new JButton(new
ImageIcon("image/memberminus.png"));
```

```
    btnDeleteMember.setCursor(Cursor.getPredefinedCursor(Cursor.HAND
_CURSOR));
       btnDeleteMember.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new deactivatemember().setVisible(true);
           }
       });
       btnDeleteMember.setFont(new Font("Tahoma", Font.PLAIN, 16));
       btnDeleteMember.setBounds(596, 270, 150, 150);
       contentPane.add(btnDeleteMember);

       JButton btnLoanTaking = new JButton(new
ImageIcon("image/loan.png"));

    btnLoanTaking.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_C
URSOR));
       btnLoanTaking.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new loantaking(username).setVisible(true);
           }
       });
       btnLoanTaking.setFont(new Font("Tahoma", Font.PLAIN, 16));
       btnLoanTaking.setBounds(111, 486, 150, 150);
       contentPane.add(btnLoanTaking);

       JButton btnLoanPayment = new JButton(new
ImageIcon("image/loanpay.png"));

    btnLoanPayment.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_
CURSOR));
       btnLoanPayment.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new loanpayment(username).setVisible(true);
           }
       });
       btnLoanPayment.setFont(new Font("Tahoma", Font.PLAIN, 16));
       btnLoanPayment.setBounds(596, 57, 150, 150);
       contentPane.add(btnLoanPayment);

       JButton btnLoanDetail = new JButton(new
ImageIcon("image/loaninfo.png"));

    btnLoanDetail.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_C
URSOR));
       btnLoanDetail.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new loandetail().setVisible(true);
           }
       });
       btnLoanDetail.setFont(new Font("Tahoma", Font.PLAIN, 16));
       btnLoanDetail.setBounds(353, 486, 150, 150);
       contentPane.add(btnLoanDetail);

       JButton btnTransaction = new JButton(new
ImageIcon("image/transaction.png"));
```

```
    btnTransaction.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_
CURSOR));
        btnTransaction.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new transaction(username).setVisible(true);
            }
        });
        btnTransaction.setFont(new Font("Tahoma", Font.PLAIN, 16));
        btnTransaction.setBounds(109, 57, 150, 150);
        contentPane.add(btnTransaction);

        JButton btnMiniStatement = new JButton(new
ImageIcon("image/ministatement.png"));

    btnMiniStatement.setCursor(Cursor.getPredefinedCursor(Cursor.HAN
D_CURSOR));
        btnMiniStatement.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new ministatement().setVisible(true);
            }
        });
        btnMiniStatement.setFont(new Font("Tahoma", Font.PLAIN,
16));
        btnMiniStatement.setBounds(353, 57, 150, 150);
        contentPane.add(btnMiniStatement);

        JButton btnLoanSettlemnt = new JButton(new
ImageIcon("image/loansat.png"));

    btnLoanSettlemnt.setCursor(Cursor.getPredefinedCursor(Cursor.HAN
D_CURSOR));
        btnLoanSettlemnt.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new loansattle().setVisible(true);
            }
        });
        btnLoanSettlemnt.setFont(new Font("Tahoma", Font.PLAIN,
16));
        btnLoanSettlemnt.setBounds(596, 486, 150, 150);
        contentPane.add(btnLoanSettlemnt);

        JPanel panel = new JPanel();
        panel.setBounds(1093, 240, 203, 321);
        contentPane.add(panel);
        panel.setLayout(null);
        panel.setVisible(false);

        JLabel lblNewLabel = new JLabel("Total Cr. Amount");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblNewLabel.setBounds(10, 11, 123, 17);
        panel.add(lblNewLabel);

        JLabel lblTotalDrAmount = new JLabel("Total Dr. Amount");
        lblTotalDrAmount.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblTotalDrAmount.setBounds(10, 71, 123, 17);
        panel.add(lblTotalDrAmount);

        JLabel lblTotalLoanPaid = new JLabel("Total Loan Paid
Amount");
        lblTotalLoanPaid.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblTotalLoanPaid.setBounds(10, 137, 166, 17);
        panel.add(lblTotalLoanPaid);

        JLabel lblNoOfTransaction = new JLabel("No. of
Transaction");
        lblNoOfTransaction.setFont(new Font("Tahoma", Font.BOLD,
14));
        lblNoOfTransaction.setBounds(10, 202, 136, 17);
        panel.add(lblNoOfTransaction);

        JLabel lblNoOfLoan = new JLabel("No. of Loan Saction");
        lblNoOfLoan.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblNoOfLoan.setBounds(10, 234, 136, 17);
        panel.add(lblNoOfLoan);

        JLabel lblNoOfLoan_1 = new JLabel("No. of Loan Satteled");
        lblNoOfLoan_1.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblNoOfLoan_1.setBounds(10, 266, 146, 17);
        panel.add(lblNoOfLoan_1);

        JLabel lblRs = new JLabel("Rs");
        lblRs.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblRs.setBounds(20, 32, 27, 17);
        panel.add(lblRs);

        JLabel label = new JLabel("Rs");
        label.setFont(new Font("Tahoma", Font.PLAIN, 14));
        label.setBounds(20, 91, 27, 17);
        panel.add(label);

        JLabel label_1 = new JLabel("Rs");
        label_1.setFont(new Font("Tahoma", Font.PLAIN, 14));
        label_1.setBounds(20, 158, 27, 17);
        panel.add(label_1);

        lblTxtTCrAmount = new JLabel("0");
        lblTxtTCrAmount.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblTxtTCrAmount.setBounds(57, 32, 99, 17);
        panel.add(lblTxtTCrAmount);

        lblTxtTDrAmount = new JLabel("0");
        lblTxtTDrAmount.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblTxtTDrAmount.setBounds(55, 91, 99, 17);
        panel.add(lblTxtTDrAmount);

        lblTxtTLPAmount = new JLabel("0");
        lblTxtTLPAmount.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblTxtTLPAmount.setBounds(51, 157, 99, 17);
        panel.add(lblTxtTLPAmount);

        lblTxtLoanSatteled = new JLabel("0");
        lblTxtLoanSatteled.setFont(new Font("Tahoma",
Font.PLAIN,14));
        lblTxtLoanSatteled.setBounds(160, 264, 34, 17);
        panel.add(lblTxtLoanSatteled);

        lblTxtLoanSaction = new JLabel("0");
        lblTxtLoanSaction.setFont(new Font("Tahoma", Font.PLAIN,
14));
        lblTxtLoanSaction.setBounds(159, 235, 35, 17);
        panel.add(lblTxtLoanSaction);

        lblTxtTransaction = new JLabel("0");
        lblTxtTransaction.setFont(new Font("Tahoma", Font.PLAIN,
14));
        lblTxtTransaction.setBounds(160, 204, 32, 17);
        panel.add(lblTxtTransaction);

        JButton btnPaymentDetail = new JButton(new
ImageIcon("image/loanpayinfo.png"));

    btnPaymentDetail.setCursor(Cursor.getPredefinedCursor(Cursor.HAN
D_CURSOR));
        btnPaymentDetail.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new loanpaymentdetail().setVisible(true);
           }
        });
        btnPaymentDetail.setFont(new Font("Tahoma", Font.PLAIN,
16));
        btnPaymentDetail.setBounds(837, 57, 150, 150);
        contentPane.add(btnPaymentDetail);

        JButton btnAdminPannel = new JButton(new
ImageIcon("image/userpannel.png"));

    btnAdminPannel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_
CURSOR));
        btnAdminPannel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new adminpannel().setVisible(true);
           }
        });
        btnAdminPannel.setFont(new Font("Tahoma", Font.PLAIN, 16));
        btnAdminPannel.setBounds(837, 270, 150, 150);
        contentPane.add(btnAdminPannel);

        JLabel lblNewLabel_1 = new JLabel("Today");
        lblNewLabel_1.addMouseListener(new MouseAdapter() {
           @Override
           public void mouseEntered(MouseEvent e) {
              panel.setVisible(true);
              today();
           }
           @Override
           public void mouseExited(MouseEvent e) {
              panel.setVisible(false);
           }
        });
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 20));
        lblNewLabel_1.setOpaque(true);
        lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);

    lblNewLabel_1.setCursor(Cursor.getPredefinedCursor(Cursor.WAIT_C
URSOR));
        lblNewLabel_1.setBackground(Color.ORANGE);
        lblNewLabel_1.setBounds(1127, 192, 135, 25);
        contentPane.add(lblNewLabel_1);

        JLabel lblNewLabel_2 = new JLabel(new
ImageIcon("image/logout.png"));

    lblNewLabel_2.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_C
URSOR));
        lblNewLabel_2.addMouseListener(new MouseAdapter() {
           @Override
           public void mouseClicked(MouseEvent e) {
              new login().setVisible(true);
              dispose();
           }
        });
        lblNewLabel_2.setBounds(1165, 79, 46, 42);
        contentPane.add(lblNewLabel_2);

        JLabel lblNewLabel_3 = new JLabel("Transaction");
        lblNewLabel_3.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblNewLabel_3.setBounds(147, 207, 76, 17);
        contentPane.add(lblNewLabel_3);

        JLabel lblStatement = new JLabel("Statement");
        lblStatement.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblStatement.setBounds(397, 207, 76, 17);
        contentPane.add(lblStatement);

        JLabel lblLoanPayment = new JLabel("Loan Payment");
        lblLoanPayment.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblLoanPayment.setBounds(631, 207, 89, 17);
        contentPane.add(lblLoanPayment);

        JLabel lblLoanPaymentDetail = new JLabel("Loan Payment
Detail");
        lblLoanPaymentDetail.setFont(new Font("Tahoma", Font.PLAIN,
14));
        lblLoanPaymentDetail.setBounds(847, 210, 140, 17);
        contentPane.add(lblLoanPaymentDetail);

        JLabel lblAddMember = new JLabel("Add Member");
        lblAddMember.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblAddMember.setBounds(147, 422, 89, 17);
        contentPane.add(lblAddMember);

        JLabel lblDetailUpdate = new JLabel("Detail / Update
Member");
```

```
        lblDetailUpdate.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblDetailUpdate.setBounds(353, 422, 150, 17);
        contentPane.add(lblDetailUpdate);

        JLabel lblDeactivateMember = new JLabel("Deactivate 
Member");
        lblDeactivateMember.setFont(new Font("Tahoma", Font.PLAIN, 
14));
        lblDeactivateMember.setBounds(606, 422, 126, 17);
        contentPane.add(lblDeactivateMember);

        JLabel lblUserPannel = new JLabel("User Pannel");
        lblUserPannel.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblUserPannel.setBounds(878, 422, 76, 17);
        contentPane.add(lblUserPannel);

        JLabel lblLoanTaking = new JLabel("Loan Taking");
        lblLoanTaking.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblLoanTaking.setBounds(147, 636, 76, 17);
        contentPane.add(lblLoanTaking);

        JLabel lblLoanDetail = new JLabel("Loan Detail");
        lblLoanDetail.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblLoanDetail.setBounds(386, 636, 76, 17);
        contentPane.add(lblLoanDetail);

        JLabel lblLoanSattelment = new JLabel("Loan Sattelment");
        lblLoanSattelment.setFont(new Font("Tahoma", Font.PLAIN, 
14));
        lblLoanSattelment.setBounds(621, 639, 111, 17);
        contentPane.add(lblLoanSattelment);
    }
    private void today(){
        String sql1,sql2,sql3,sql4,Particular;
        Integer count =0,cr=0,dr=0,loan=0;

        sql1="select * from sss.transaction where date='"+today+"'";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql1);
           while(rs.next()){
              count++;
              Particular=rs.getString("particular");
              if(Particular.equals("To Cash") 
||Particular.equals("To Cheque")){
                cr=cr+Integer.parseInt(rs.getString("amount"));
              }else if(Particular.equals("By Cash") 
||Particular.equals("By Cheque")){
                dr=dr+Integer.parseInt(rs.getString("amount"));
              }
           }
           lblTxtTransaction.setText(count.toString());
           lblTxtTCrAmount.setText(cr.toString());
           lblTxtTDrAmount.setText(dr.toString());
        } catch (Exception e) { JOptionPane.showMessageDialog(null, 
e);}
```

```
        sql2="select * from sss.loanpayment where date='"+today+"'";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql2);
           while(rs.next()){
              loan=loan+Integer.parseInt(rs.getString("amount"));
           }
           lblTxtTLPAmount.setText(String.valueOf(loan));
        } catch (Exception e) { JOptionPane.showMessageDialog(null,
e);}
        sql3="select * from sss.loantaking where date='"+today+"'";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql3);
           Integer x=0;
           while(rs.next()){
              Integer z=Integer.parseInt(rs.getString("flag"));
              if(z==0){
                 x++;
              }
           }
           lblTxtLoanSaction.setText(x.toString());
        } catch (Exception e) { JOptionPane.showMessageDialog(null,
e);}
        sql4="select * from sss.loantaking where
satelmentdate='"+today+"'";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql4);
           Integer y=0;
           while(rs.next()){
              Integer z=Integer.parseInt(rs.getString("flag"));
              if(z==1){
                 y++;
              }
           }
           lblTxtLoanSatteled.setText(y.toString());
        } catch (Exception e) { JOptionPane.showMessageDialog(null,
e);}
     }
}
```



ADD MEMBER

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import java.awt.Frame;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JOptionPane;
import javax.swing.JTabbedPane;
import javax.swing.JLabel;

import java.awt.Font;

import javax.swing.SwingConstants;

import java.awt.Component;
import java.awt.Color;
import java.awt.TextArea;

import javax.swing.JTextField;
import javax.swing.JRadioButton;
import javax.swing.JComboBox;
import javax.swing.JCheckBox;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

import org.eclipse.wb.swing.FocusTraversalOnArray;


public class addmember extends JFrame {

    Integer accountno,count1=0,count2=0;
    String uname;
    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtFileno;
    private JTextField txtFname;
    private JTextField txtMname;
    private JTextField txtLname;
    private JTextField txtGname;
    private JTextField txtEmail;
    private JTextField txtLine1;
    private JTextField txtLine2;
    private JTextField txtMobileno;
    private JTextField txtLandmark;
    private JTextField txtCity;
    private JTextField txtState;
    private JTextField txtDistrict;
    private JTextField txtPincode;
    private JTextField txtPincodeP;
    private JTextField txtStateP;
    private JTextField txtCityP;
    private JTextField txtDistrictP;
    private JTextField txtLandmarkP;
    private JTextField txtMobilenoP;
    private JTextField txtLine2P;
    private JTextField txtLine1P;
    private JTextField txtIntroac1;
    private JTextField txtIntroac2;
    private JTextField txtFnameN;
    private JTextField txtMnameN;
    private JTextField txtLnameN;
    private JTextField txtRelation;
    private TextArea txtArea;
    private JComboBox comboDD;
    private JComboBox comboMM;
    private JComboBox comboYYYY;
    private JComboBox comboGender;
    private JComboBox comboMarital;
    private JComboBox comboCategory;
    private JComboBox comboDDN;
    private JComboBox comboMMN;
    private JComboBox comboYYYYN;
    private JComboBox comboGenderN;
    private JLabel lblIntro1name;
    private JLabel lblIntro2name;
    private JRadioButton rdbtnFather;
    private JRadioButton rdbtnHusband;
    private ButtonGroup btngrp;
    private JButton btnSaveBasic;
    private JButton btnResetBasic;
    private JButton btnCancelBasic;
    private JButton btnSaveAddress;
    private JButton btnResetAddress;
    private JButton btnCancelAddress;
    private JTabbedPane tabbedPane;
    private JPanel panel_1;

    java.util.Date datet= Calendar.getInstance().getTime();
    DateFormat format= new SimpleDateFormat("dd/MM/yyyy");
    DateFormat formattime= new SimpleDateFormat("hh:mm:ssa");
    String today = format.format(datet);
    String time = formattime.format(datet);

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
       EventQueue.invokeLater(new Runnable() {
           public void run() {
              try {
                 addmember frame = new addmember(null);
                 frame.setVisible(true);
              } catch (Exception e) {
                 e.printStackTrace();
              }
           }
       });
    }

    /**
     * Create the frame.
     */
    public addmember(String username) {
       uname=username;
       setResizable(false);
       setExtendedState(Frame.MAXIMIZED_BOTH);
       setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
       setSize(new Dimension(1364,720));
       contentPane = new JPanel();
       contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
       setContentPane(contentPane);
       contentPane.setLayout(null);
       tabbedPane = new JTabbedPane(JTabbedPane.TOP);
       tabbedPane.setBounds(91, 69, 1153, 557);
       contentPane.add(tabbedPane);

       JPanel panel = new JPanel();
       tabbedPane.addTab("Basic Detail", null, panel, null);
       panel.setLayout(null);

       JLabel lblBasicDetail = new JLabel("BASIC DETAIL");

    lblBasicDetail.setHorizontalAlignment(SwingConstants.CENTER);
       lblBasicDetail.setFont(new Font("Tahoma", Font.BOLD, 18));
       lblBasicDetail.setBounds(108, 41, 918, 24);
       lblBasicDetail.setBackground(Color.ORANGE);
       lblBasicDetail.setOpaque(true);
       panel.add(lblBasicDetail);

       JLabel lblAccountNo = new JLabel("A/C NO");
       lblAccountNo.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblAccountNo.setBounds(213, 101, 74, 24);
       panel.add(lblAccountNo);

       JLabel lblDate = new JLabel("Date");
       lblDate.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblDate.setBounds(461, 101, 56, 24);
       panel.add(lblDate);

       JLabel lblFileNo = new JLabel("File No");
       lblFileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblFileNo.setBounds(693, 101, 63, 24);
       panel.add(lblFileNo);

       txtFileno = new JTextField();
       txtFileno.setBounds(785, 101, 134, 22);
       panel.add(txtFileno);
       txtFileno.setColumns(10);

       JLabel lblName = new JLabel("Name");
       lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblName.setBounds(213, 149, 74, 24);
       panel.add(lblName);

       JLabel lblFirstName = new JLabel("First Name");
       lblFirstName.setFont(new Font("Tahoma", Font.BOLD, 13));
       lblFirstName.setBounds(264, 178, 74, 24);
       panel.add(lblFirstName);

       JLabel lblMiddleName = new JLabel("Middle Name");
       lblMiddleName.setFont(new Font("Tahoma", Font.BOLD, 13));
       lblMiddleName.setBounds(514, 178, 93, 24);
       panel.add(lblMiddleName);

       JLabel lblLastName = new JLabel("Last Name");
       lblLastName.setFont(new Font("Tahoma", Font.BOLD, 13));
       lblLastName.setBounds(732, 178, 74, 24);
       panel.add(lblLastName);
```

```
        txtFname = new JTextField();
        txtFname.setBounds(264, 209, 187, 20);
        panel.add(txtFname);
        txtFname.setColumns(10);

        txtMname = new JTextField();
        txtMname.setColumns(10);
        txtMname.setBounds(514, 209, 169, 20);
        panel.add(txtMname);

        txtLname = new JTextField();
        txtLname.setColumns(10);
        txtLname.setBounds(732, 209, 187, 20);
        panel.add(txtLname);

        //Auto increment account Number
        int count =1;
        String sql=("select acno from memberdetail");
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while(rs.next()){
              count+=1;
           }
        } catch (Exception e) { JOptionPane.showMessageDialog(null,
e);}
        Integer acc=null;
        if (count<=100000){
           acc=100000+count;
        }
        else{
           JOptionPane.showMessageDialog(null, "Acount Exceed to
100000. Meet to Software developer");
        }
        accountno=acc;
        String accstr = accountno.toString();

        JLabel lblacno = new JLabel(accstr);
        lblacno.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblacno.setBounds(317, 101, 134, 21);
        panel.add(lblacno);

        JLabel label = new JLabel(today);
        label.setFont(new Font("Tahoma", Font.PLAIN, 14));
        label.setBounds(516, 101, 93, 21);
        panel.add(label);

        rdbtnFather = new JRadioButton("Father Name");
        rdbtnFather.setFont(new Font("Tahoma", Font.BOLD, 15));
        rdbtnFather.setBounds(213, 260, 141, 27);
        panel.add(rdbtnFather);

        rdbtnHusband = new JRadioButton("Husband Name");
        rdbtnHusband.setFont(new Font("Tahoma", Font.BOLD, 15));
        rdbtnHusband.setBounds(370, 260, 141, 27);
        panel.add(rdbtnHusband);

        btngrp = new ButtonGroup();
        btngrp.add(rdbtnFather);
        btngrp.add(rdbtnHusband);

        txtGname = new JTextField();
        txtGname.setColumns(10);
        txtGname.setBounds(557, 264, 362, 22);
        panel.add(txtGname);

        JLabel lblEmail  = new JLabel("Email");
        lblEmail.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblEmail.setBounds(213, 318, 63, 24);
        panel.add(lblEmail);

        txtEmail = new JTextField();
        txtEmail.setColumns(10);
        txtEmail.setBounds(290, 321, 344, 22);
        panel.add(txtEmail);

        JLabel lblDOB = new JLabel("D O B");
        lblDOB.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblDOB.setBounds(671, 318, 44, 24);
        panel.add(lblDOB);

        String
dd[]={"01","02","03","04","05","06","07","08","09","10","11","12","1
3","14","15","16","17","18","19","20","21","22","23","24","25","26",
"27","28","29","30","31"};
        comboDD = new JComboBox(dd);
        comboDD.setBounds(726, 315, 49, 27);
        panel.add(comboDD);

        String
mm[]={"01","02","03","04","05","06","07","08","09","10","11","12"};
        comboMM = new JComboBox(mm);
        comboMM.setBounds(785, 315, 50, 28);
        panel.add(comboMM);

        comboYYYY = new JComboBox(new Object[]{});
        for(int i=2019; i>=(2019-100); i--)
        {
           comboYYYY.addItem(i);
        }
        comboYYYY.setBounds(845, 316, 74, 27);
        panel.add(comboYYYY);

        JLabel lblGender = new JLabel("Gender");
        lblGender.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGender.setBounds(213, 372, 68, 24);
        panel.add(lblGender);

        JLabel lblMa = new JLabel("Marital Status");
        lblMa.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblMa.setBounds(435, 372, 106, 24);
```

```
panel.add(lblMa);

JLabel lblCategory = new JLabel("Category");
lblCategory.setFont(new Font("Tahoma", Font.BOLD, 15));
lblCategory.setBounds(732, 372, 68, 24);
panel.add(lblCategory);

comboGender = new JComboBox();
comboGender.addItem("Male");
comboGender.addItem("Female");
comboGender.setBounds(286, 376, 87, 20);
panel.add(comboGender);

comboMarital = new JComboBox();
comboMarital.addItem("Married");
comboMarital.addItem("Unmarried");
comboMarital.setBounds(561, 376, 106, 20);
panel.add(comboMarital);

comboCategory = new JComboBox();
comboCategory.addItem("GEN");
comboCategory.addItem("OBC");
comboCategory.addItem("SC/ST");
comboCategory.setBounds(832, 376, 87, 20);
panel.add(comboCategory);

btnCancelBasic = new JButton("Cancel");
btnCancelBasic.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
    }
});
btnCancelBasic.setBounds(836, 439, 89, 23);
panel.add(btnCancelBasic);

btnResetBasic = new JButton("Reset");
btnResetBasic.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        resetBasicDetail();
    }
});
btnResetBasic.setBounds(693, 439, 89, 23);
panel.add(btnResetBasic);

btnSaveBasic = new JButton("Save & Next");
btnSaveBasic.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        saveBasicDetail();
    }
});
btnSaveBasic.setBounds(531, 439, 106, 23);
panel.add(btnSaveBasic);

panel_1 = new JPanel();
tabbedPane.addTab("Address Detail", null, panel_1, null);
tabbedPane.setEnabledAt(1, false);
```

```
        panel_1.setLayout(null);

        JLabel lblAddress = new JLabel("ADDRESS");
        lblAddress.setOpaque(true);
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblAddress.setHorizontalAlignment(SwingConstants.CENTER);
        lblAddress.setBackground(Color.ORANGE);
        lblAddress.setBounds(96, 42, 954, 23);
        panel_1.add(lblAddress);

        JLabel lblCurrentAddress = new JLabel("Current Address");
        lblCurrentAddress.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblCurrentAddress.setBounds(129, 85, 127, 23);
        panel_1.add(lblCurrentAddress);

        JLabel lblLine = new JLabel("Line 1");
        lblLine.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblLine.setBounds(129, 137, 60, 23);
        panel_1.add(lblLine);

        JLabel lblLine_1 = new JLabel("Line 2");
        lblLine_1.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblLine_1.setBounds(129, 181, 60, 23);
        panel_1.add(lblLine_1);

        JLabel lblMobileNo = new JLabel("Mobile No.");
        lblMobileNo.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblMobileNo.setBounds(129, 225, 72, 23);
        panel_1.add(lblMobileNo);

        JLabel lblLandmark = new JLabel("LandMark");
        lblLandmark.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblLandmark.setBounds(129, 273, 72, 23);
        panel_1.add(lblLandmark);

        JLabel lblCity = new JLabel("City");
        lblCity.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblCity.setBounds(129, 323, 44, 23);
        panel_1.add(lblCity);

        JLabel lblState = new JLabel("State");
        lblState.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblState.setBounds(129, 376, 44, 23);
        panel_1.add(lblState);

        JLabel label_1 = new JLabel("");
        label_1.setBackground(Color.GRAY);
        label_1.setOpaque(true);
        label_1.setBounds(567, 76, 10, 330);
        panel_1.add(label_1);

        txtLine1 = new JTextField();
        txtLine1.setBounds(231, 139, 298, 23);
        panel_1.add(txtLine1);
        txtLine1.setColumns(10);
        txtLine2 = new JTextField();
        txtLine2.setColumns(10);
        txtLine2.setBounds(231, 181, 298, 23);
        panel_1.add(txtLine2);

        txtMobileno = new JTextField();
        txtMobileno.addKeyListener(new KeyAdapter() {
            @Override
            public void keyReleased(KeyEvent e) {
                if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                    txtMobileno.setEditable(true);
                } else {
                    txtMobileno.setEditable(false);
                    Toolkit.getDefaultToolkit().beep();
                    JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                    txtMobileno.setEditable(true);
                }
            }
        });
        txtMobileno.setColumns(10);
        txtMobileno.setBounds(231, 225, 298, 23);
        panel_1.add(txtMobileno);

        txtLandmark = new JTextField();
        txtLandmark.setColumns(10);
        txtLandmark.setBounds(231, 273, 298, 23);
        panel_1.add(txtLandmark);

        txtCity = new JTextField();
        txtCity.setColumns(10);
        txtCity.setBounds(183, 325, 121, 23);
        panel_1.add(txtCity);

        JLabel lblDistrict = new JLabel("District");
        lblDistrict.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblDistrict.setBounds(340, 323, 48, 23);
        panel_1.add(lblDistrict);

        JLabel lblPincode = new JLabel("Pincode");
        lblPincode.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblPincode.setBounds(340, 376, 60, 23);
        panel_1.add(lblPincode);

        txtState = new JTextField();
        txtState.setColumns(10);
        txtState.setBounds(183, 378, 121, 23);
        panel_1.add(txtState);

        txtDistrict = new JTextField();
        txtDistrict.setColumns(10);
        txtDistrict.setBounds(408, 325, 121, 23);
        panel_1.add(txtDistrict);
```

```
txtPincode = new JTextField();
txtPincode.setColumns(10);
txtPincode.setBounds(410, 376, 121, 23);
panel_1.add(txtPincode);

txtPincodeP = new JTextField();
txtPincodeP.setColumns(10);
txtPincodeP.setBounds(899, 376, 121, 23);
panel_1.add(txtPincodeP);

JLabel label_2 = new JLabel("Pincode");
label_2.setFont(new Font("Tahoma", Font.BOLD, 14));
label_2.setBounds(829, 376, 60, 23);
panel_1.add(label_2);

txtStateP = new JTextField();
txtStateP.setColumns(10);
txtStateP.setBounds(672, 378, 121, 23);
panel_1.add(txtStateP);

JLabel label_3 = new JLabel("State");
label_3.setFont(new Font("Tahoma", Font.BOLD, 14));
label_3.setBounds(618, 376, 44, 23);
panel_1.add(label_3);

JLabel label_4 = new JLabel("City");
label_4.setFont(new Font("Tahoma", Font.BOLD, 14));
label_4.setBounds(618, 323, 44, 23);
panel_1.add(label_4);

txtCityP = new JTextField();
txtCityP.setColumns(10);
txtCityP.setBounds(672, 325, 121, 23);
panel_1.add(txtCityP);

JLabel lblDistrict_1 = new JLabel("District");
lblDistrict_1.setFont(new Font("Tahoma", Font.BOLD, 14));
lblDistrict_1.setBounds(829, 323, 60, 23);
panel_1.add(lblDistrict_1);

txtDistrictP = new JTextField();
txtDistrictP.setColumns(10);
txtDistrictP.setBounds(897, 325, 121, 23);
panel_1.add(txtDistrictP);

txtLandmarkP = new JTextField();
txtLandmarkP.setColumns(10);
txtLandmarkP.setBounds(720, 273, 298, 23);
panel_1.add(txtLandmarkP);

JLabel label_6 = new JLabel("LandMark");
label_6.setFont(new Font("Tahoma", Font.BOLD, 14));
label_6.setBounds(618, 273, 72, 23);
panel_1.add(label_6);

JLabel label_7 = new JLabel("Mobile No.");
```

```
        label_7.setFont(new Font("Tahoma", Font.BOLD, 14));
        label_7.setBounds(618, 225, 72, 23);
        panel_1.add(label_7);

        txtMobilenoP = new JTextField();
        txtMobilenoP.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtMobilenoP.setEditable(true);
              } else {
                 txtMobilenoP.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtMobilenoP.setEditable(true);
              }
           }
        });
        txtMobilenoP.setColumns(10);
        txtMobilenoP.setBounds(720, 225, 298, 23);
        panel_1.add(txtMobilenoP);

        txtLine2P = new JTextField();
        txtLine2P.setColumns(10);
        txtLine2P.setBounds(720, 181, 298, 23);
        panel_1.add(txtLine2P);

        JLabel label_8 = new JLabel("Line 2");
        label_8.setFont(new Font("Tahoma", Font.BOLD, 14));
        label_8.setBounds(618, 181, 60, 23);
        panel_1.add(label_8);

        JLabel label_9 = new JLabel("Line 1");
        label_9.setFont(new Font("Tahoma", Font.BOLD, 14));
        label_9.setBounds(618, 137, 60, 23);
        panel_1.add(label_9);

        txtLine1P = new JTextField();
        txtLine1P.setColumns(10);
        txtLine1P.setBounds(720, 139, 298, 23);
        panel_1.add(txtLine1P);

        JCheckBox chckbxAddressP = new JCheckBox("Permanent
Address");
        chckbxAddressP.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              if(chckbxAddressP.isSelected()){
                 txtLine1P.setText(txtLine1.getText());
                 txtLine2P.setText(txtLine2.getText());
                 txtMobilenoP.setText(txtMobileno.getText());
                 txtLandmarkP.setText(txtLandmark.getText());
                 txtCityP.setText(txtCity.getText());
                 txtDistrictP.setText(txtDistrict.getText());
                 txtStateP.setText(txtState.getText());
```

```
                txtPincodeP.setText(txtPincode.getText());
            }
            else{
                txtLine1P.setText("");
                txtLine2P.setText("");
                txtMobilenoP.setText("");
                txtLandmarkP.setText("");
                txtCityP.setText("");
                txtDistrictP.setText("");
                txtStateP.setText("");
                txtPincodeP.setText("");
            }
        }
    });
    chckbxAddressP.setFont(new Font("Tahoma", Font.BOLD, 15));
    chckbxAddressP.setBounds(619, 87, 184, 23);
    panel_1.add(chckbxAddressP);

    btnCancelAddress = new JButton("Cancel");
    btnCancelAddress.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            cancel();
        }
    });
    btnCancelAddress.setBounds(927, 449, 89, 23);
    panel_1.add(btnCancelAddress);

    btnResetAddress = new JButton("Reset");
    btnResetAddress.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            resetAddressDetail();
        }
    });
    btnResetAddress.setBounds(796, 449, 89, 23);
    panel_1.add(btnResetAddress);

    btnSaveAddress = new JButton("Save & Next");
    btnSaveAddress.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            saveAddressDetail();
        }
    });
    btnSaveAddress.setBounds(640, 449, 112, 23);
    panel_1.add(btnSaveAddress);

    JPanel panel_2 = new JPanel();
    tabbedPane.addTab("Other Detail", null, panel_2, null);
    tabbedPane.setEnabledAt(2, false);
    panel_2.setLayout(null);

    JLabel lblAcIntroducer = new JLabel("A/C INTRODUCER");
    lblAcIntroducer.setOpaque(true);

lblAcIntroducer.setHorizontalAlignment(SwingConstants.CENTER);
    lblAcIntroducer.setFont(new Font("Tahoma", Font.BOLD, 16));
    lblAcIntroducer.setBackground(Color.ORANGE);
```

```
        lblAcIntroducer.setBounds(98, 47, 954, 23);
        panel_2.add(lblAcIntroducer);

        JLabel lblNewLabel_1 = new JLabel("Introducer 1st A/C No.");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblNewLabel_1.setBounds(167, 103, 187, 23);
        panel_2.add(lblNewLabel_1);

        JLabel lblIntroducerndAc = new JLabel("Introducer 2nd A/C
No.");
        lblIntroducerndAc.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblIntroducerndAc.setBounds(167, 142, 187, 23);
        panel_2.add(lblIntroducerndAc);

        JLabel lblNomineeDetail = new JLabel("NOMINEE DETAIL");
        lblNomineeDetail.setOpaque(true);

     lblNomineeDetail.setHorizontalAlignment(SwingConstants.CENTER);
        lblNomineeDetail.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblNomineeDetail.setBackground(Color.ORANGE);
        lblNomineeDetail.setBounds(98, 199, 954, 23);
        panel_2.add(lblNomineeDetail);

        txtIntroac1 = new JTextField();
        txtIntroac1.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtIntroac1.setEditable(true);
              } else {
                 txtIntroac1.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtIntroac1.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtIntroac1.getText());
              if(acno>200000){
                 txtIntroac1.setEditable(false);
                 JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                 txtIntroac1.setText("");
                 txtIntroac1.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                 introducer1();
              }


              String sql=("select * from memberdetail");
              try {
                 Statement stmt=dbc.conmethod();
                 ResultSet rs=stmt.executeQuery(sql);
                 while(rs.next()){
```

```
    if(txtIntroac1.getText().equals(rs.getString("acno"))){
                           count1=1;
                      }
                 }
                 if (count1==0 && acno >= 100000 && acno<=199999){
                      JOptionPane.showMessageDialog(null, "A/C Not 
Found, Enter Valid A/C No.");
                      txtIntroac1.requestFocus(true);
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        txtIntroac1.setBounds(377, 106, 202, 20);
        panel_2.add(txtIntroac1);
        txtIntroac1.setColumns(10);

        txtIntroac2 = new JTextField();
        txtIntroac2.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {

              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' || 
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtIntroac2.setEditable(true);
              } else {
                 txtIntroac2.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only 
numeric digits(0-9) ");
                 txtIntroac2.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtIntroac2.getText());
              if(acno>200000){
                 txtIntroac2.setEditable(false);
                 JOptionPane.showMessageDialog(null, "Please Enter 
valid Account No");
                 txtIntroac2.setText("");
                 txtIntroac2.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                 introducer2();
              }


              String sql=("select * from memberdetail");
              try {
                 Statement stmt=dbc.conmethod();
                 ResultSet rs=stmt.executeQuery(sql);
                 while(rs.next()){

    if(txtIntroac2.getText().equals(rs.getString("acno"))){
                           count2=1;
                      }
                 }
```

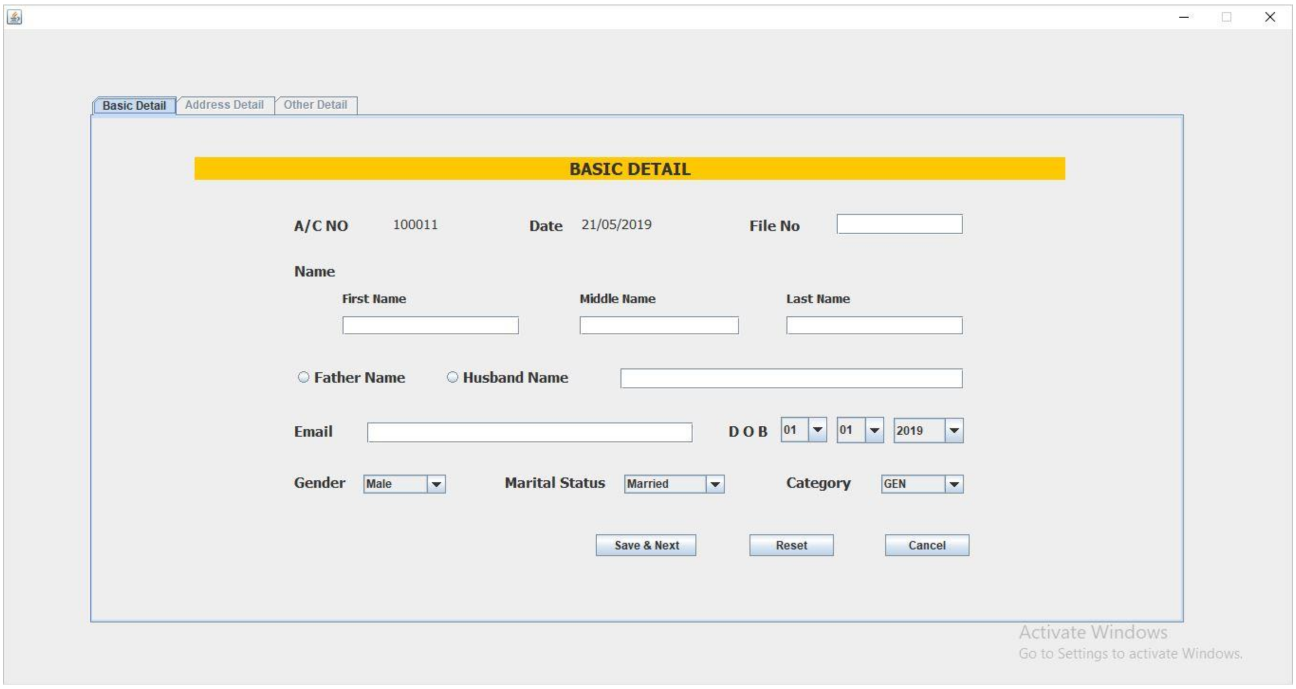
```
                if (count2==0 && acno >= 100000 && acno<=199999){
                     JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                     txtIntroac2.requestFocus(true);
                }
            } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
          }
       });
       txtIntroac2.setColumns(10);
       txtIntroac2.setBounds(377, 145, 202, 20);
       panel_2.add(txtIntroac2);

       JLabel lblName_1 = new JLabel("Name");
       lblName_1.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblName_1.setBounds(636, 103, 73, 23);
       panel_2.add(lblName_1);

       JLabel label_11 = new JLabel("Name");
       label_11.setFont(new Font("Tahoma", Font.BOLD, 15));
       label_11.setBounds(636, 142, 73, 23);
       panel_2.add(label_11);

       lblIntro1name = new JLabel();
       lblIntro1name.setFont(new Font("Tahoma", Font.PLAIN, 15));
       lblIntro1name.setBounds(739, 103, 181, 20);
       panel_2.add(lblIntro1name);

       lblIntro2name= new JLabel();
       lblIntro2name.setFont(new Font("Tahoma", Font.PLAIN, 15));
       lblIntro2name.setBounds(739, 143, 181, 20);
       panel_2.add(lblIntro2name);

       JLabel lblFirstName_1 = new JLabel("First Name");
       lblFirstName_1.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblFirstName_1.setBounds(167, 255, 100, 23);
       panel_2.add(lblFirstName_1);

       JLabel lblMiddleName_1 = new JLabel("Middle Name");
       lblMiddleName_1.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblMiddleName_1.setBounds(458, 255, 100, 23);
       panel_2.add(lblMiddleName_1);

       JLabel lblLastName_1 = new JLabel("Last Name");
       lblLastName_1.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblLastName_1.setBounds(750, 255, 100, 23);
       panel_2.add(lblLastName_1);

       txtFnameN = new JTextField();
       txtFnameN.setColumns(10);
       txtFnameN.setBounds(167, 289, 222, 20);
       panel_2.add(txtFnameN);

       txtMnameN = new JTextField();
       txtMnameN.setColumns(10);
       txtMnameN.setBounds(458, 289, 222, 20);
```

```
        panel_2.add(txtMnameN);

        txtLnameN = new JTextField();
        txtLnameN.setColumns(10);
        txtLnameN.setBounds(750, 289, 231, 20);
        panel_2.add(txtLnameN);

        JLabel lblRelationWithAccount = new JLabel("Relation with
Account Holder");
        lblRelationWithAccount.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblRelationWithAccount.setBounds(167, 336, 222, 23);
        panel_2.add(lblRelationWithAccount);

        txtRelation = new JTextField();
        txtRelation.setColumns(10);
        txtRelation.setBounds(422, 339, 222, 20);
        panel_2.add(txtRelation);

        JLabel lblDOB_1 = new JLabel("D O B");
        lblDOB_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblDOB_1.setBounds(750, 336, 52, 23);
        panel_2.add(lblDOB_1);

        comboDDN = new JComboBox(dd);
        comboDDN.setBounds(812, 339, 40, 20);
        panel_2.add(comboDDN);

        comboMMN = new JComboBox(mm);
        comboMMN.setBounds(862, 339, 40, 20);
        panel_2.add(comboMMN);

        comboYYYYN = new JComboBox();
        for(int j=2019; j>=(2019-100); j--)
         {
            comboYYYYN.addItem(j);
         }
        comboYYYYN.setBounds(912, 339, 69, 20);
        panel_2.add(comboYYYYN);

        JLabel lblAddress_1 = new JLabel("Address");
        lblAddress_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress_1.setBounds(167, 392, 73, 23);
        panel_2.add(lblAddress_1);

        txtArea = new TextArea();
        txtArea.setBounds(250, 392, 246, 76);
        panel_2.add(txtArea);

        JLabel lblGender_1 = new JLabel("Gender");
        lblGender_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGender_1.setBounds(545, 398, 69, 23);
        panel_2.add(lblGender_1);

        comboGenderN = new JComboBox();
        comboGenderN.addItem("Male");
        comboGenderN.addItem("Female");
        comboGenderN.setBounds(657, 401, 69, 20);
        panel_2.add(comboGenderN);

        JButton btnCancel_2 = new JButton("Cancel");
        btnCancel_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                cancel();
            }
        });
        btnCancel_2.setBounds(892, 469, 89, 23);
        panel_2.add(btnCancel_2);

        JButton btnReset_2 = new JButton("Reset");
        btnReset_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                resetOtherDetails();
            }
        });
        btnReset_2.setBounds(775, 469, 89, 23);
        panel_2.add(btnReset_2);

        JButton btnCreateAc = new JButton("Create A/C");
        btnCreateAc.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                saveOtherDetails();
            }
        });
        btnCreateAc.setBounds(636, 469, 108, 23);
        panel_2.add(btnCreateAc);
    }
    private void saveBasicDetail()
    {
        String
FileNumber,FirstName,MiddleName,LastName,GType="",GName,Email,Marita
lstatus,DateOfBirth,Gen,Cat,sql1;
        FileNumber =txtFileno.getText();
        FirstName =txtFname.getText();
        MiddleName = txtMname.getText();
        LastName =txtLname.getText();
        Email= txtEmail.getText();
        Maritalstatus=comboMarital.getSelectedItem().toString();
        if(rdbtnFather.isSelected())
            GType = "Father";
        else if(rdbtnHusband.isSelected())
            GType = "Husband";
        else JOptionPane.showMessageDialog(null, "Select Gardian
Name");
        GName =txtGname.getText();
        DateOfBirth = comboYYYY.getSelectedItem().toString() +"/"+
comboMM.getSelectedItem().toString() +"/"+
comboDD.getSelectedItem().toString();
        Gen = comboGender.getSelectedItem().toString();
        Cat = comboCategory.getSelectedItem().toString();
        sql1="INSERT INTO
sss.memberdetail(acno,fileno,fname,mname,lname,gtype,gname,email,mar
italstatus,dob,gender,category,openedby)
values('"+accountno+"','"+FileNumber+"','"+FirstName+"','"+MiddleNam
e+"','"+LastName+"','"+GType+"','"+GName+"','"+Email+"','"+Maritalst
atus+"','"+DateOfBirth+"','"+Gen+"','"+Cat+"','"+uname+"')";
        if (accountno.equals("") || FileNumber.equals("") ||
FirstName.equals("") || GType.equals("") || DateOfBirth.equals("")
|| Gen.equals("") || Cat.equals("")  ){
           JOptionPane.showMessageDialog(null, "Fill Requried
Data");
        }else
        {
           try {
              Statement stmt=dbc.conmethod();
              stmt.execute(sql1);
              resetBasicDetail();
              tabbedPane.setSelectedIndex(1);
              tabbedPane.setEnabledAt(0, false);
              tabbedPane.setEnabledAt(1, true);
              JOptionPane.showMessageDialog(null, "Basic Details
Stored");
              stmt.close();
           }

    catch(com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintVi
olationException e){
              resetBasicDetail();
              JOptionPane.showInternalMessageDialog(null, "Duplicate
Data");
           }
           catch (SQLException
e){JOptionPane.showMessageDialog(null, e); }
           catch(java.lang.NullPointerException ex){
JOptionPane.showMessageDialog(null, ex); }}
    }

    private void resetBasicDetail()
    {
        txtFileno.setText("");
        txtFname.setText("");
        txtMname.setText("");
        txtLname.setText("");
        txtGname.setText("");
        txtEmail.setText("");
        comboDD.setSelectedIndex(0);
        comboMM.setSelectedIndex(0);
        comboYYYY.setSelectedIndex(0);
        comboCategory.setSelectedIndex(0);
        comboGender.setSelectedIndex(0);
        btngrp.clearSelection();
        txtFileno.requestFocus();
    }

    private void saveAddressDetail(){
        String
Line1,Line2,Mobile,Lmark,City,Dist,State,Pin,Line1p,Line2p,Mobilep,L
markp,Cityp,Distp,Statep,Pinp,sql2;
        Line1=txtLine1.getText();
        Line2=txtLine2.getText();
        Mobile=txtMobileno.getText();
        Lmark=txtLandmark.getText();
        City= txtCity.getText();
        Dist=txtDistrict.getText();
        State=txtState.getText();
        Pin=txtPincode.getText();
        Line1p=txtLine1P.getText();
        Line2p=txtLine2P.getText();
        Mobilep=txtMobilenoP.getText();
        Lmarkp=txtLandmarkP.getText();
        Cityp=txtCityP.getText();
        Distp=txtDistrictP.getText();
        Statep=txtStateP.getText();
        Pinp=txtPincodeP.getText();

        sql2="UPDATE sss.memberdetail SET line1='"+ Line1 +"',
line2='"+ Line2 +"', mobileno='" +Mobile+ "', landmark='"+Lmark+"',
city='"+City+"', distric='"+Dist+"', state='"+State+"',
pincode='"+Pin+"',line1p='"+ Line1p +"', line2p='"+ Line2p +"',
mobilenop='" +Mobilep+ "', landmarkp='"+Lmarkp+"',
cityp='"+Cityp+"', districp='"+Distp+"', statep='"+Statep+"',
pincodep='"+Pinp+"' WHERE acno='"+accountno+"'";
        if (Line1.equals("") ||  Mobile.equals("") ||
Lmark.equals("") || City.equals("") || Dist.equals("") ||
State.equals("") || Pin.equals("") || Line1p.equals("") ||
Mobilep.equals("") || Lmarkp.equals("") || Cityp.equals("") ||
Distp.equals("") || Statep.equals("") || Pinp.equals("") ){
           JOptionPane.showMessageDialog(null, "Fill Required
Data");
        }else
        {
           try {
           Statement stmt=dbc.conmethod();
           String mobno=txtMobileno.getText();
           String mobnop=txtMobilenoP.getText();
           if(mobno.length() != 10 || mobnop.length() != 10){
              JOptionPane.showMessageDialog(null, "Please Enter the
10 digit Mobile no");
           }else{
              stmt.execute(sql2);
              tabbedPane.setEnabledAt(1, false);
              tabbedPane.setEnabledAt(2, true);
              resetAddressDetail();
              tabbedPane.setSelectedIndex(2);
              JOptionPane.showMessageDialog(null, "Address Detais
Stored");
           }
        }
    catch(com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintVi
olationException e){
```

```
                JOptionPane.showInternalMessageDialog(null, "Duplicate 
Data");
            }
            catch (SQLException e){JOptionPane.showMessageDialog(null, 
e);}
            }
        }

        private void resetAddressDetail() {
            txtLine1.setText("");
            txtLine2.setText("");
            txtMobileno.setText("");
            txtLandmark.setText("");
            txtCity.setText("");
            txtDistrict.setText("");
            txtState.setText("");
            txtPincode.setText("");
            txtLine1P.setText("");
            txtLine2P.setText("");
            txtMobilenoP.setText("");
            txtLandmarkP.setText("");
            txtCityP.setText("");
            txtDistrictP.setText("");
            txtStateP.setText("");
            txtPincodeP.setText("");
            txtLine1.requestFocus(true);
        }

        private void saveOtherDetails(){
            String Intro1, Intro2, FirstNameN, MiddleNameN, LastNameN, 
Relation, DateOfBirthN, AddressN, GenderN, sql3;
            Intro1=txtIntroac1.getText();
            Intro2=txtIntroac2.getText();
            FirstNameN=txtFnameN.getText();
            MiddleNameN=txtMnameN.getText();
            LastNameN=txtLnameN.getText();
            Relation=txtRelation.getText();
            DateOfBirthN=
comboYYYYN.getSelectedItem().toString()+"/"+comboMMN.getSelectedItem
().toString()+"/"+comboDDN.getSelectedItem().toString();
            AddressN=txtArea.getText();
            GenderN=comboGenderN.getSelectedItem().toString();

            sql3 = "UPDATE `sss`.`memberdetail` SET 
`introac1`='"+Intro1+"', `introac2`='"+Intro2+"', 
`fnamen`='"+FirstNameN+"', `mnamen`='"+MiddleNameN+"', 
`lnamen`='"+LastNameN+"', `relation`='"+Relation+"', 
`dobn`='"+DateOfBirthN+"', `addressn`='"+AddressN+"', 
`gendern`='"+GenderN+"', `openeddate`=now(), `status`='Active' WHERE 
`acno`='"+accountno+"'";
            if (Intro1.equals("") ||  Intro2.equals("") || 
FirstNameN.equals("") || Relation.equals("") || 
DateOfBirthN.equals("") || AddressN.equals("") || GenderN.equals("")
){
                JOptionPane.showMessageDialog(null, "Fill Required 
Detail");
```

```
        }else if (count1==0 ){
           JOptionPane.showMessageDialog(null, "Please Enter Correct 
First Introducer");
           txtIntroac1.requestFocus(true);
        }else if (count2==0 ){
           JOptionPane.showMessageDialog(null, "Please Enter Correct 
Second Introducer");
           txtIntroac2.requestFocus(true);
        }
        else
        {
           try {
           Statement stmt=dbc.conmethod();
           stmt.execute(sql3);
           resetOtherDetails();
           JOptionPane.showMessageDialog(null, "Account Created. 
Account No is : "+accountno);
           dispose();
        }

    catch(com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintVi
olationException e){
           JOptionPane.showInternalMessageDialog(null, "Duplicate 
Data");
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null, 
e);}
        }
    }
    private void resetOtherDetails() {
        txtIntroac1.setText("");
        txtIntroac2.setText("");
        txtFname.setText("");
        txtMname.setText("");
        txtLname.setText("");
        txtRelation.setText("");
        txtArea.setText("");
        comboGenderN.setSelectedIndex(0);
        comboDDN.setSelectedIndex(0);
        comboMMN.setSelectedIndex(0);
        comboYYYYN.setSelectedIndex(0);
        lblIntro1name.setText("");
        lblIntro2name.setText("");
    }
    private void cancel(){
        String sql4;
        sql4="DELETE FROM memberdetail WHERE acno='"+accountno+"'";
        try {
           Statement stmt=dbc.conmethod();
           stmt.execute(sql4);
           dispose();
        } catch (Exception e) {JOptionPane.showMessageDialog(null, 
e);}
    }
    private void introducer1(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtIntroac1.getText()+"";
        String status="", name=null;
        String intro1=txtIntroac1.getText();
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              status=rs.getString("status");
              name=rs.getString("fname") +"
"+rs.getString("mname")+" "+rs.getString("lname");
           }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
        if(status.equals("Active")){
           lblIntro1name.setText(name);
        }else if(Integer.valueOf(intro1)>100000 &&
Integer.valueOf(intro1)<999999  && status.equals("Deactivate")){
           JOptionPane.showMessageDialog(null,   "A/c is
Deactivated. Please Insert Valid A/C no.");
           txtIntroac1.setText("");
           lblIntro1name.setText("");
        }
    }
    private void introducer2(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtIntroac2.getText()+"";
        String status="", name="";
        String intro2=txtIntroac2.getText();
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              status=rs.getString("status");
              name=rs.getString("fname") +"
"+rs.getString("mname")+" "+rs.getString("lname");
           }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
        if(status.equals("Active")){
           lblIntro2name.setText(name);
        }else if(Integer.valueOf(intro2)>100000 &&
Integer.valueOf(intro2)<999999 && status.equals("Deactivate")){
           JOptionPane.showMessageDialog(null,   "A/c is
Deactivated. Please Insert Valid A/C no.");
           txtIntroac2.setText("");
           lblIntro2name.setText("");
        }
        if(txtIntroac1.getText().equals(txtIntroac2.getText())){
           JOptionPane.showMessageDialog(null,   "Similar to A/C 1,
Please Enter the Other A/C no.");
           txtIntroac2.setText("");
           lblIntro2name.setText("");
        }
    }
}
```

Add member first tab



BASIC DETAIL

A/C NO 100011 Date 21/05/2019 File No

Name

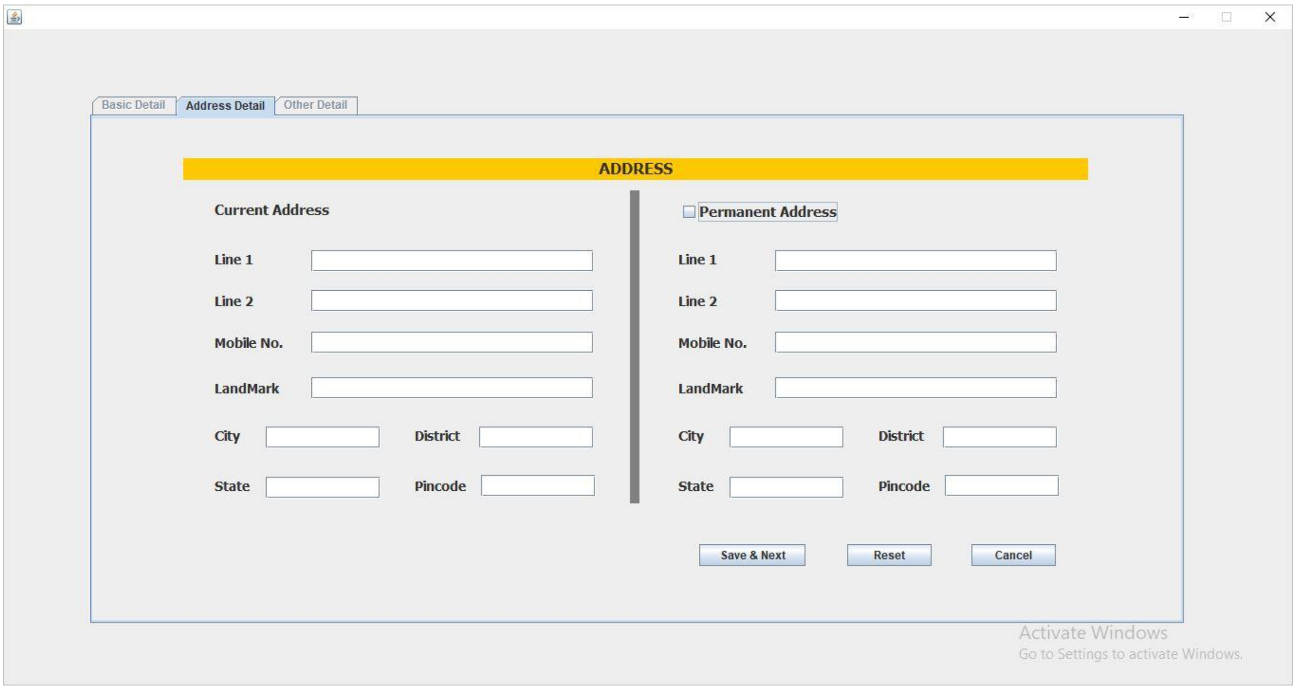
First Name Middle Name Last Name

○ Father Name ○ Husband Name

Email D O B 01 01 2019

Gender Male Marital Status Married Category GEN

Add member second tab



ADDRESS

Current Address

Line 1

Line 2

Mobile No.

LandMark

City District

State Pincode

□ Permanent Address

Line 1

Line 2

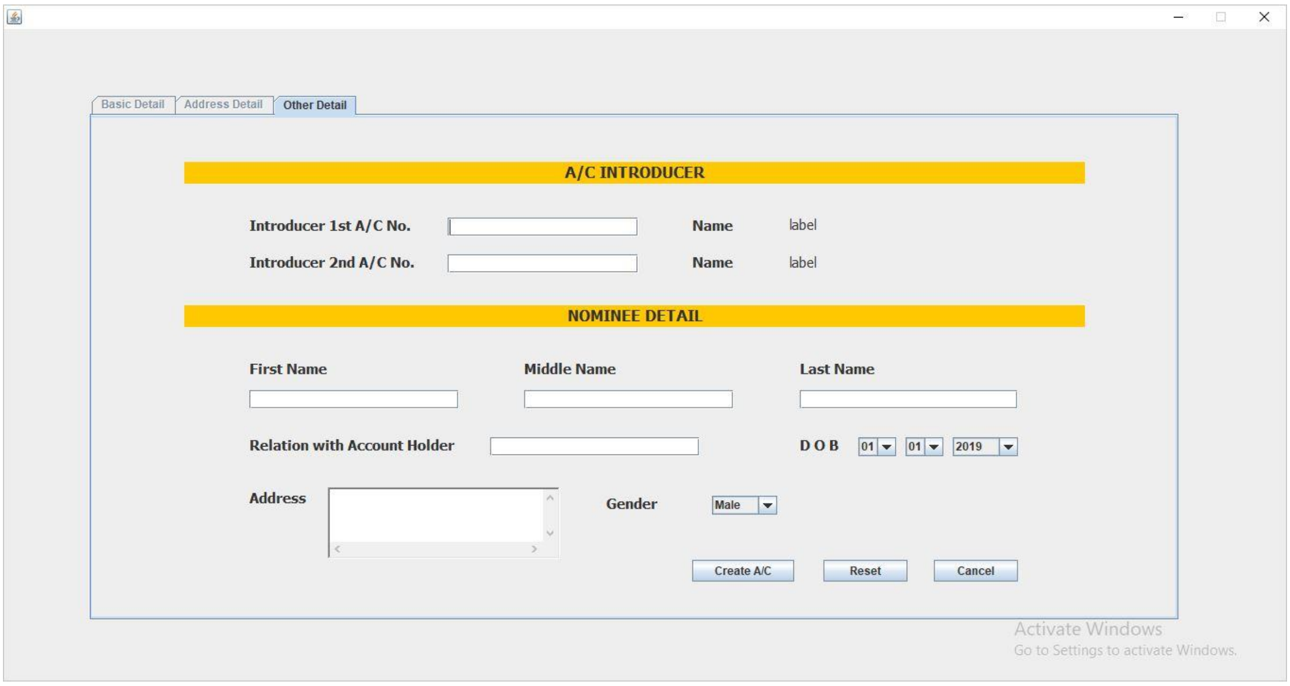
Mobile No.

LandMark

City District

State Pincode

Add member Third tab



Name label

Name label

D O B 01 01 2019

Gender Male

MEMBER DETAIL

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import java.awt.Frame;

import javax.swing.JLabel;

import java.awt.Color;
import java.awt.Font;

import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.Statement;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;


public class memberdetail extends JFrame {

	dbconnection dbc = new dbconnection();
	private JPanel contentPane;
	private JTextField txtAcno;
	private JTextField txtFName;
	private JTextField txtMName;
	private JTextField txtLName;
	private JTextField txtEmail;
	private JTextField txtGName;
	private JTextField txtLine1;
	private JTextField txtLine2;
	private JTextField txtMobile;
	private JTextField txtLandmark;
	private JTextField txtCity;
	private JTextField txtState;
	private JTextField txtDistrict;
	private JTextField txtPin;
	private JTextField txtLine1P;
	private JTextField txtMobileP;
	private JTextField txtLine2P;
	private JTextField txtLandmarkP;
	private JTextField txtDistrictP;
	private JTextField txtPinP;
	private JTextField txtCityP;
	private JTextField txtStateP;
	private JLabel lblTxtStatus;
	private JLabel lblTxtDate;
	private JLabel lblTxtOpenedBy;
	private JLabel lblTxtFileNo;
	private JLabel lblTxtDOB;
	private JLabel lblTxtIntroac1;
	private JLabel lblTxtIntroac2;
	private JLabel lblTxtNameN;
	private JLabel lblTxtGenderN;
	private JLabel lblTxtDOBN;
	private JLabel lblTxtAddressN;
	private JLabel lblTxtRelation;
	private JLabel lblTxtLoan;
	private JLabel lblTxtTotalAmount;
	private JLabel lblTxtPaidAmount;
	private JLabel lblTxtRestAmount;
	private JComboBox comboGType;
	private JComboBox comboGender;
	private JComboBox comboMaritalStatus;
	private JComboBox comboCategory;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
```

```
              try {
                memberdetail frame = new memberdetail();
                frame.setVisible(true);
              } catch (Exception e) {
                e.printStackTrace();
              }
           }
        });
    }

    /**
     * Create the frame.
     */
    public memberdetail() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        setSize(new Dimension(1364,720));
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("BASIC DETAIL");
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setOpaque(true);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setBounds(62, 24, 1222, 19);
        contentPane.add(lblNewLabel);

        JLabel lblAcNo = new JLabel("A/C NO");
        lblAcNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAcNo.setBounds(89, 64, 62, 19);
        contentPane.add(lblAcNo);

        txtAcno = new JTextField();
        txtAcno.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                txtAcno.setEditable(true);
              } else {
                txtAcno.setEditable(false);
                Toolkit.getDefaultToolkit().beep();
                JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                txtAcno.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcno.getText());
              if(acno>200000){
                txtAcno.setEditable(false);
                JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                txtAcno.setText("");
                txtAcno.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                alldetail();
                loandetail();
              }


              String sql=("select * from memberdetail");
              try {
                Statement stmt=dbc.conmethod();
                ResultSet rs=stmt.executeQuery(sql);
                int count =0;
                while(rs.next()){

    if(txtAcno.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                }
                if (count==0 && acno >= 100000 && acno<=199999){
                      JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                      txtAcno.requestFocus(true);
                }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}

           }
        });
        txtAcno.setBounds(166, 65, 199, 21);
        contentPane.add(txtAcno);
        txtAcno.setColumns(10);

        lblTxtStatus = new JLabel("Status");
        lblTxtStatus.setBackground(Color.LIGHT_GRAY);
        lblTxtStatus.setOpaque(true);
        lblTxtStatus.setHorizontalAlignment(SwingConstants.CENTER);
        lblTxtStatus.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTxtStatus.setBounds(425, 64, 96, 19);
        contentPane.add(lblTxtStatus);

        lblTxtDate = new JLabel("Date");
        lblTxtDate.setHorizontalAlignment(SwingConstants.CENTER);
        lblTxtDate.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTxtDate.setBounds(545, 64, 183, 19);
        contentPane.add(lblTxtDate);

        JLabel lblAccountOpenBy = new JLabel("Account Open By");

    lblAccountOpenBy.setHorizontalAlignment(SwingConstants.CENTER);
        lblAccountOpenBy.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAccountOpenBy.setBounds(750, 64, 132, 19);
        contentPane.add(lblAccountOpenBy);

        lblTxtOpenedBy = new JLabel();
        lblTxtOpenedBy.setHorizontalAlignment(SwingConstants.LEFT);
        lblTxtOpenedBy.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtOpenedBy.setBounds(905, 64, 125, 19);
        contentPane.add(lblTxtOpenedBy);

        JLabel lblFile = new JLabel("File No");
        lblFile.setHorizontalAlignment(SwingConstants.CENTER);
        lblFile.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblFile.setBounds(1016, 64, 96, 19);
        contentPane.add(lblFile);

        lblTxtFileNo = new JLabel();
        lblTxtFileNo.setHorizontalAlignment(SwingConstants.LEFT);
        lblTxtFileNo.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtFileNo.setBounds(1122, 64, 96, 19);
        contentPane.add(lblTxtFileNo);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(89, 106, 51, 19);
        contentPane.add(lblName);

        JLabel lblFirstName = new JLabel("First Name");
        lblFirstName.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblFirstName.setBounds(111, 131, 96, 19);
        contentPane.add(lblFirstName);

        JLabel lblMiddleName = new JLabel("Middle Name");
        lblMiddleName.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblMiddleName.setBounds(306, 135, 96, 19);
        contentPane.add(lblMiddleName);

        JLabel lblLastName = new JLabel("Last Name");
        lblLastName.setFont(new Font("Tahoma", Font.BOLD, 14));
        lblLastName.setBounds(508, 135, 96, 19);
        contentPane.add(lblLastName);

        txtFName = new JTextField();
        txtFName.setEditable(false);
        txtFName.setBounds(111, 159, 159, 20);
        contentPane.add(txtFName);
        txtFName.setColumns(10);

        txtMName = new JTextField();
        txtMName.setEditable(false);
        txtMName.setColumns(10);
        txtMName.setBounds(306, 159, 159, 20);
        contentPane.add(txtMName);

        txtLName = new JTextField();
        txtLName.setEditable(false);
        txtLName.setColumns(10);
        txtLName.setBounds(508, 159, 159, 20);
        contentPane.add(txtLName);

        JLabel lblEmail = new JLabel("Email");
        lblEmail.setFont(new Font("Tahoma", Font.BOLD, 15));
```

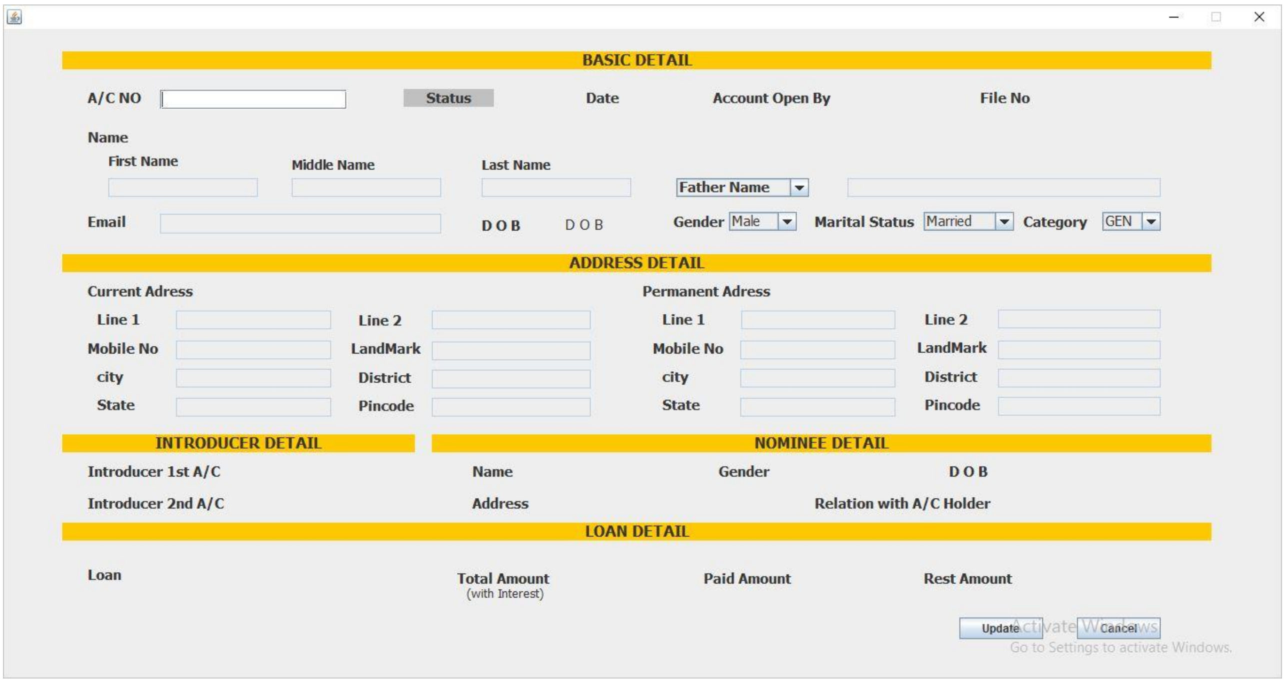
```
        lblEmail.setBounds(89, 196, 51, 19);
        contentPane.add(lblEmail);

        txtEmail = new JTextField();
        txtEmail.setEditable(false);
        txtEmail.setColumns(10);
        txtEmail.setBounds(166, 197, 299, 21);
        contentPane.add(txtEmail);

        comboGType = new JComboBox();
        comboGType.setFont(new Font("Tahoma", Font.BOLD, 15));
        comboGType.addItem("Father Name");
        comboGType.addItem("Husband Name");
        comboGType.setBounds(715, 159, 141, 20);
        contentPane.add(comboGType);

        txtGName = new JTextField();
        txtGName.setEditable(false);
        txtGName.setDragEnabled(true);
        txtGName.setBounds(897, 159, 333, 20);
        contentPane.add(txtGName);
        txtGName.setColumns(10);

        JLabel lblD = new JLabel("D O B");
        lblD.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblD.setBounds(508, 200, 51, 19);
        contentPane.add(lblD);

        lblTxtDOB = new JLabel("D O B");
        lblTxtDOB.setHorizontalAlignment(SwingConstants.CENTER);
        lblTxtDOB.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtDOB.setBounds(568, 199, 99, 19);
        contentPane.add(lblTxtDOB);

        JLabel lblGender = new JLabel("Gender");
        lblGender.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGender.setBounds(712, 196, 62, 19);
        contentPane.add(lblGender);

        JLabel lblMatrialStatus = new JLabel("Marital Status");
        lblMatrialStatus.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblMatrialStatus.setBounds(862, 196, 106, 19);
        contentPane.add(lblMatrialStatus);

        comboGender = new JComboBox();
        comboGender.addItem("Male");
        comboGender.addItem("Female");
        comboGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
        comboGender.setBounds(771, 195, 72, 20);
        contentPane.add(comboGender);

        comboMaritalStatus = new JComboBox();
        comboMaritalStatus.addItem("Married");
        comboMaritalStatus.addItem("Unmarried");
        comboMaritalStatus.setFont(new Font("Tahoma", Font.PLAIN,
15));
```

```
        comboMaritalStatus.setBounds(978, 195, 96, 20);
        contentPane.add(comboMaritalStatus);

        JLabel lblCategory = new JLabel("Category");
        lblCategory.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCategory.setBounds(1084, 196, 74, 19);
        contentPane.add(lblCategory);

        comboCategory = new JComboBox();
        comboCategory.addItem("GEN");
        comboCategory.addItem("OBC");
        comboCategory.addItem("ST/SC");
        comboCategory.setFont(new Font("Tahoma", Font.PLAIN, 15));
        comboCategory.setBounds(1168, 195, 62, 20);
        contentPane.add(comboCategory);

        JLabel lblAddressDetail = new JLabel("ADDRESS DETAIL");
        lblAddressDetail.setOpaque(true);

     lblAddressDetail.setHorizontalAlignment(SwingConstants.CENTER);
        lblAddressDetail.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblAddressDetail.setBackground(Color.ORANGE);
        lblAddressDetail.setBounds(62, 240, 1222, 19);
        contentPane.add(lblAddressDetail);

        JLabel lblCurrentAdress = new JLabel("Current Adress");
        lblCurrentAdress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCurrentAdress.setBounds(89, 270, 118, 19);
        contentPane.add(lblCurrentAdress);

        JLabel lblLine = new JLabel("Line 1");
        lblLine.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLine.setBounds(99, 300, 51, 19);
        contentPane.add(lblLine);

        JLabel lblLine_1 = new JLabel("Line 2");
        lblLine_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLine_1.setBounds(377, 301, 51, 19);
        contentPane.add(lblLine_1);

        JLabel lblMobileNo = new JLabel("Mobile No");
        lblMobileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblMobileNo.setBounds(89, 331, 84, 19);
        contentPane.add(lblMobileNo);

        JLabel lblLandmark = new JLabel("LandMark");
        lblLandmark.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLandmark.setBounds(369, 331, 84, 19);
        contentPane.add(lblLandmark);

        JLabel lblCity = new JLabel("city");
        lblCity.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCity.setBounds(99, 361, 51, 19);
        contentPane.add(lblCity);

        txtLine1 = new JTextField();
```

```
txtLine1.setEditable(false);
txtLine1.setColumns(10);
txtLine1.setBounds(183, 300, 165, 20);
contentPane.add(txtLine1);

txtLine2 = new JTextField();
txtLine2.setEditable(false);
txtLine2.setColumns(10);
txtLine2.setBounds(455, 300, 169, 20);
contentPane.add(txtLine2);

txtMobile = new JTextField();
txtMobile.setEditable(false);
txtMobile.setColumns(10);
txtMobile.setBounds(183, 332, 165, 20);
contentPane.add(txtMobile);

txtLandmark = new JTextField();
txtLandmark.setEditable(false);
txtLandmark.setColumns(10);
txtLandmark.setBounds(455, 333, 169, 20);
contentPane.add(txtLandmark);

txtCity = new JTextField();
txtCity.setEditable(false);
txtCity.setColumns(10);
txtCity.setBounds(183, 362, 165, 20);
contentPane.add(txtCity);

JLabel lblDistric = new JLabel("District");
lblDistric.setFont(new Font("Tahoma", Font.BOLD, 15));
lblDistric.setBounds(377, 362, 62, 19);
contentPane.add(lblDistric);

JLabel lblState = new JLabel("State");
lblState.setFont(new Font("Tahoma", Font.BOLD, 15));
lblState.setBounds(99, 391, 51, 19);
contentPane.add(lblState);

JLabel lblPincode = new JLabel("Pincode");
lblPincode.setFont(new Font("Tahoma", Font.BOLD, 15));
lblPincode.setBounds(377, 392, 62, 19);
contentPane.add(lblPincode);

txtState = new JTextField();
txtState.setEditable(false);
txtState.setColumns(10);
txtState.setBounds(183, 393, 165, 20);
contentPane.add(txtState);

txtDistrict = new JTextField();
txtDistrict.setEditable(false);
txtDistrict.setColumns(10);
txtDistrict.setBounds(455, 363, 169, 20);
contentPane.add(txtDistrict);
```

```
        txtPin = new JTextField();
        txtPin.setEditable(false);
        txtPin.setColumns(10);
        txtPin.setBounds(455, 393, 169, 20);
        contentPane.add(txtPin);

        txtLine1P = new JTextField();
        txtLine1P.setEditable(false);
        txtLine1P.setColumns(10);
        txtLine1P.setBounds(784, 300, 164, 20);
        contentPane.add(txtLine1P);

        JLabel label_2 = new JLabel("Line 1");
        label_2.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_2.setBounds(700, 300, 51, 19);
        contentPane.add(label_2);

        JLabel lblPermanentAdress = new JLabel("Permanent Adress");
        lblPermanentAdress.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblPermanentAdress.setBounds(679, 270, 141, 19);
        contentPane.add(lblPermanentAdress);

        JLabel label_4 = new JLabel("Mobile No");
        label_4.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_4.setBounds(690, 331, 84, 19);
        contentPane.add(label_4);

        txtMobileP = new JTextField();
        txtMobileP.setEditable(false);
        txtMobileP.setColumns(10);
        txtMobileP.setBounds(783, 332, 165, 20);
        contentPane.add(txtMobileP);

        JLabel label_5 = new JLabel("Line 2");
        label_5.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_5.setBounds(979, 300, 51, 19);
        contentPane.add(label_5);

        txtLine2P = new JTextField();
        txtLine2P.setEditable(false);
        txtLine2P.setColumns(10);
        txtLine2P.setBounds(1057, 299, 173, 20);
        contentPane.add(txtLine2P);

        txtLandmarkP = new JTextField();
        txtLandmarkP.setEditable(false);
        txtLandmarkP.setColumns(10);
        txtLandmarkP.setBounds(1057, 332, 173, 20);
        contentPane.add(txtLandmarkP);

        JLabel label_6 = new JLabel("LandMark");
        label_6.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_6.setBounds(971, 330, 84, 19);
        contentPane.add(label_6);
        txtDistrictP = new JTextField();
        txtDistrictP.setEditable(false);
        txtDistrictP.setColumns(10);
        txtDistrictP.setBounds(1057, 362, 173, 20);
        contentPane.add(txtDistrictP);

        txtPinP = new JTextField();
        txtPinP.setEditable(false);
        txtPinP.setColumns(10);
        txtPinP.setBounds(1057, 392, 173, 20);
        contentPane.add(txtPinP);

        JLabel label_7 = new JLabel("Pincode");
        label_7.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_7.setBounds(979, 391, 62, 19);
        contentPane.add(label_7);

        JLabel lblDistrict = new JLabel("District");
        lblDistrict.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblDistrict.setBounds(979, 361, 62, 19);
        contentPane.add(lblDistrict);

        txtCityP = new JTextField();
        txtCityP.setEditable(false);
        txtCityP.setColumns(10);
        txtCityP.setBounds(783, 362, 165, 20);
        contentPane.add(txtCityP);

        JLabel label_9 = new JLabel("city");
        label_9.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_9.setBounds(700, 361, 51, 19);
        contentPane.add(label_9);

        JLabel label_10 = new JLabel("State");
        label_10.setFont(new Font("Tahoma", Font.BOLD, 15));
        label_10.setBounds(700, 391, 51, 19);
        contentPane.add(label_10);

        txtStateP = new JTextField();
        txtStateP.setEditable(false);
        txtStateP.setColumns(10);
        txtStateP.setBounds(783, 393, 165, 20);
        contentPane.add(txtStateP);

        JLabel lblIntroducerDetail = new JLabel("INTRODUCER
DETAIL");
        lblIntroducerDetail.setOpaque(true);

     lblIntroducerDetail.setHorizontalAlignment(SwingConstants.CENTER
);
        lblIntroducerDetail.setFont(new Font("Tahoma", Font.BOLD,
16));
        lblIntroducerDetail.setBackground(Color.ORANGE);
        lblIntroducerDetail.setBounds(62, 432, 375, 19);
        contentPane.add(lblIntroducerDetail);
        JLabel lblIntroducerstAc = new JLabel("Introducer 1st A/C");
        lblIntroducerstAc.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblIntroducerstAc.setBounds(89, 462, 148, 19);
        contentPane.add(lblIntroducerstAc);

        JLabel lblIntroducerndAc = new JLabel("Introducer 2nd A/C");
        lblIntroducerndAc.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblIntroducerndAc.setBounds(89, 496, 148, 19);
        contentPane.add(lblIntroducerndAc);

        JLabel lblNomineeDetail = new JLabel("NOMINEE DETAIL");
        lblNomineeDetail.setOpaque(true);

    lblNomineeDetail.setHorizontalAlignment(SwingConstants.CENTER);
        lblNomineeDetail.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblNomineeDetail.setBackground(Color.ORANGE);
        lblNomineeDetail.setBounds(455, 432, 829, 19);
        contentPane.add(lblNomineeDetail);

        lblTxtIntroac1 = new JLabel();
        lblTxtIntroac1.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtIntroac1.setBounds(254, 462, 148, 19);
        contentPane.add(lblTxtIntroac1);

        lblTxtIntroac2 = new JLabel();
        lblTxtIntroac2.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtIntroac2.setBounds(254, 496, 148, 19);
        contentPane.add(lblTxtIntroac2);

        JLabel lblName_1 = new JLabel("Name");
        lblName_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName_1.setBounds(498, 462, 62, 19);
        contentPane.add(lblName_1);

        JLabel lblGender_1 = new JLabel("Gender");
        lblGender_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGender_1.setBounds(760, 462, 62, 19);
        contentPane.add(lblGender_1);

        JLabel lblDOB_1 = new JLabel("D O B");
        lblDOB_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblDOB_1.setBounds(1005, 462, 62, 19);
        contentPane.add(lblDOB_1);

        lblTxtNameN = new JLabel();
        lblTxtNameN.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtNameN.setBounds(568, 462, 148, 19);
        contentPane.add(lblTxtNameN);

        lblTxtGenderN = new JLabel();
        lblTxtGenderN.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGenderN.setBounds(839, 462, 141, 19);
        contentPane.add(lblTxtGenderN);
        lblTxtDOBN = new JLabel();
        lblTxtDOBN.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtDOBN.setBounds(1077, 462, 141, 19);
        contentPane.add(lblTxtDOBN);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(498, 496, 62, 19);
        contentPane.add(lblAddress);

        lblTxtAddressN = new JLabel();
        lblTxtAddressN.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddressN.setBounds(578, 496, 266, 19);
        contentPane.add(lblTxtAddressN);

        JLabel lblRelationWithA = new JLabel("Relation with A/C
Holder");
        lblRelationWithA.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRelationWithA.setBounds(862, 496, 193, 19);
        contentPane.add(lblRelationWithA);

        lblTxtRelation = new JLabel();
        lblTxtRelation.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtRelation.setBounds(1077, 496, 153, 19);
        contentPane.add(lblTxtRelation);

        JLabel lblLoanDetail = new JLabel("LOAN DETAIL");
        lblLoanDetail.setOpaque(true);
        lblLoanDetail.setHorizontalAlignment(SwingConstants.CENTER);
        lblLoanDetail.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblLoanDetail.setBackground(Color.ORANGE);
        lblLoanDetail.setBounds(62, 526, 1222, 19);
        contentPane.add(lblLoanDetail);

        JButton btnCancel = new JButton("Cancel");
        btnCancel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnCancel.setBounds(1141, 627, 89, 23);
        contentPane.add(btnCancel);

        JButton btnEdit = new JButton("Update");
        btnEdit.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new updatemember().setVisible(true);
              dispose();
           }
        });
        btnEdit.setBounds(1016, 627, 89, 23);
        contentPane.add(btnEdit);

        JLabel lblLoan = new JLabel("Loan");
        lblLoan.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoan.setBounds(89, 572, 51, 19);
        contentPane.add(lblLoan);

        JLabel lblTotalAmount = new JLabel("Total Amount");
        lblTotalAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalAmount.setBounds(482, 576, 112, 19);
        contentPane.add(lblTotalAmount);

        JLabel lblPaidAmount = new JLabel("Paid Amount");
        lblPaidAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPaidAmount.setBounds(744, 576, 112, 19);
        contentPane.add(lblPaidAmount);

        JLabel lblRestAmount = new JLabel("Rest Amount");
        lblRestAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRestAmount.setBounds(978, 576, 112, 19);
        contentPane.add(lblRestAmount);

        lblTxtLoan = new JLabel();
        lblTxtLoan.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtLoan.setBounds(166, 572, 262, 19);
        contentPane.add(lblTxtLoan);

        lblTxtTotalAmount = new JLabel();
        lblTxtTotalAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalAmount.setBounds(596, 576, 132, 19);
        contentPane.add(lblTxtTotalAmount);

        lblTxtPaidAmount = new JLabel();
        lblTxtPaidAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtPaidAmount.setBounds(850, 576, 118, 19);
        contentPane.add(lblTxtPaidAmount);

        lblTxtRestAmount = new JLabel();
        lblTxtRestAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtRestAmount.setBounds(1084, 576, 146, 19);
        contentPane.add(lblTxtRestAmount);

        JLabel lblwithInterest = new JLabel();
        lblwithInterest.setText("(with Interest)");
        lblwithInterest.setFont(new Font("Tahoma", Font.PLAIN, 13));
        lblwithInterest.setBounds(492, 592, 84, 19);
        contentPane.add(lblwithInterest);
    }


    private void alldetail(){
        String sql1="SELECT * FROM sss.memberdetail where acno="+
txtAcno.getText() +"";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql1);
           while (rs.next()){
              lblTxtStatus.setText(rs.getString("status"));
              if(rs.getString("status").equals("Active")){
                 lblTxtStatus.setBackground(Color.GREEN);
                 lblTxtDate.setText(rs.getString("openeddate"));
              }
              else{
                 lblTxtStatus.setBackground(Color.RED);
                 lblTxtDate.setText(rs.getString("deactivateddate"));
              }
              lblTxtOpenedBy.setText(rs.getString("openedby"));
              lblTxtFileNo.setText(rs.getString("fileno"));
              txtFName.setText(rs.getString("fname"));
              txtLName.setText(rs.getString("lname"));
              txtMName.setText(rs.getString("mname"));

    if(rs.getString("gtype").equals("Father")){comboGType.setSelecte
dIndex(0);}
              else
if(rs.getString("gtype").equals("Husband")){comboGType.setSelectedIn
dex(1);}
              txtGName.setText(rs.getString("gname"));
              txtEmail.setText(rs.getString("email"));
              lblTxtDOB.setText(rs.getString("dob"));

    if(rs.getString("gender").equals("Male")){comboGender.setSelecte
dIndex(0);}
              else
if(rs.getString("gender").equals("Female")){comboGender.setSelectedI
ndex(1);}

    if(rs.getString("maritalstatus").equals("Married")){comboMarital
Status.setSelectedIndex(0);}
              else
if(rs.getString("maritalstatus").equals("Unmarried")){comboMaritalSt
atus.setSelectedIndex(1);}

    if(rs.getString("category").equals("GEN")){comboCategory.setSele
ctedIndex(0);}
              else
if(rs.getString("category").equals("OBC")){comboCategory.setSelected
Index(1);}
              else
if(rs.getString("category").equals("SC/ST")){comboCategory.setSelect
edIndex(2);}
              txtLine1.setText(rs.getString("line1"));
              txtLine2.setText(rs.getString("line2"));
              txtMobile.setText(rs.getString("mobileno"));
              txtLandmark.setText(rs.getString("landmark"));
              txtCity.setText(rs.getString("city"));
              txtDistrict.setText(rs.getString("distric"));
              txtState.setText(rs.getString("state"));
              txtPin.setText(rs.getString("pincode"));
              txtLine1P.setText(rs.getString("line1p"));
              txtLine2P.setText(rs.getString("line2p"));
              txtMobileP.setText(rs.getString("mobilenop"));
              txtLandmarkP.setText(rs.getString("landmarkp"));
              txtCityP.setText(rs.getString("cityp"));
```

```
                txtDistrictP.setText(rs.getString("districp"));
                txtStateP.setText(rs.getString("statep"));
                txtPinP.setText(rs.getString("pincodep"));

    lblTxtIntroac1.setText(rs.getString("introac1"));lblTxtIntroac2.
setText(rs.getString("introac2"));
                lblTxtNameN.setText(rs.getString("fnamen")+"
"+rs.getString("mnamen")+" "+rs.getString("lnamen"));
                lblTxtAddressN.setText(rs.getString("addressn"));
                lblTxtGenderN.setText(rs.getString("gendern"));
                lblTxtDOBN.setText(rs.getString("dobn"));
                lblTxtRelation.setText(rs.getString("relation"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }


    private void loandetail(){
        String Accountno=txtAcno.getText(),opendate="", lastdate="";
        Integer totalamount=0,totalamountpaid=0;
        String sql=("SELECT acno, DATE_FORMAT(date, '%d/%m/%Y') As
opendate, totalamount, paidtill, totalamountpaid FROM loantaking
WHERE acno='"+Accountno+"' ");
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            int count =0;
            while(rs.next()){
                opendate=rs.getString("opendate");
                lastdate=rs.getString("paidtill");
                totalamount=rs.getInt("totalamount");
                totalamountpaid=rs.getInt("totalamountpaid");
                if(txtAcno.getText().equals(rs.getString("acno"))){
                    count=1;
                }
            }
            if (count==1){
                lblTxtLoan.setText(opendate+" to "+lastdate);

    lblTxtTotalAmount.setText(String.valueOf(totalamount));

    lblTxtPaidAmount.setText(String.valueOf(totalamountpaid));
                lblTxtRestAmount.setText(String.valueOf(totalamount-
totalamountpaid));
            }else{
                lblTxtLoan.setText("You havn't take any Loan");
                lblTxtTotalAmount.setText("");
                lblTxtPaidAmount.setText("");
                lblTxtRestAmount.setText("");
            }
        } catch (Exception ex){ JOptionPane.showMessageDialog(null,
ex);}
    }
}
```



Father Name

Gender Male

Marital Status Married

Category GEN

MEMBER UPDATE

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Frame;
import java.awt.Toolkit;

import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;
import javax.swing.JCheckBox;
import javax.swing.JButton;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;


public class updatemember extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtAcno;
    private JTextField txtFName;
    private JTextField txtMName;
    private JTextField txtLName;
    private JTextField txtEmail;
    private JTextField txtGName;
    private JTextField txtLine1;
    private JTextField txtLine2;
    private JTextField txtMobile;
    private JTextField txtLandmark;
    private JTextField txtCity;
    private JTextField txtState;
    private JTextField txtDistrict;
    private JTextField txtPin;
    private JTextField txtLine1P;
    private JTextField txtMobileP;
    private JTextField txtLine2P;
    private JTextField txtLandmarkP;
    private JTextField txtDistrictP;
    private JTextField txtPinP;
    private JTextField txtCityP;
    private JTextField txtStateP;
    private JLabel lblTxtDate;
    private JLabel lblTxtStatus;
    private JLabel lblTxtFileNo;
    private JComboBox comboGType;
    private JComboBox comboGender;
    private JComboBox comboMaritalStatus;
    private JComboBox comboCategory;
    private JTextField txtDOB;
    private JCheckBox chckbxAddressP;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
           public void run() {
               try {
                   updatemember frame = new updatemember();
                   frame.setVisible(true);
               } catch (Exception e) {
                   e.printStackTrace();
               }
           }
        });
    }

    /**
     * Create the frame.
     */
    public updatemember() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        setSize(new Dimension(1364,720));
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("BASIC DETAIL");
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setOpaque(true);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setBounds(62, 24, 1222, 19);
        contentPane.add(lblNewLabel);

        JLabel lblAcNo = new JLabel("A/C NO");
        lblAcNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAcNo.setBounds(89, 64, 62, 19);
        contentPane.add(lblAcNo);

        txtAcno = new JTextField();
        txtAcno.addKeyListener(new KeyAdapter() {
           @Override
```

```
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                txtAcno.setEditable(true);
              } else {
                txtAcno.setEditable(false);
                Toolkit.getDefaultToolkit().beep();
                JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                txtAcno.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcno.getText());
              if(acno>200000){
                txtAcno.setEditable(false);
                JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                txtAcno.setText("");
                txtAcno.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                detail();
              }


              String sql=("select * from memberdetail");
              try {
                Statement stmt=dbc.conmethod();
                ResultSet rs=stmt.executeQuery(sql);
                int count =0;
                while(rs.next()){

    if(txtAcno.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                }
                if (count==0 && acno >= 100000 && acno<=199999){
                      txtAcno.setEditable(false);
                      JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                      txtAcno.setEditable(true);
                      txtAcno.requestFocus(true);
                }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        txtAcno.setBounds(166, 65, 199, 21);
        contentPane.add(txtAcno);
        txtAcno.setColumns(10);

        lblTxtStatus = new JLabel("Status");
        lblTxtStatus.setBackground(Color.LIGHT_GRAY);
        lblTxtStatus.setOpaque(true);
        lblTxtStatus.setHorizontalAlignment(SwingConstants.CENTER);
        lblTxtStatus.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTxtStatus.setBounds(425, 64, 96, 19);
```

```
   contentPane.add(lblTxtStatus);

   lblTxtDate = new JLabel("Date");
   lblTxtDate.setHorizontalAlignment(SwingConstants.CENTER);
   lblTxtDate.setFont(new Font("Tahoma", Font.BOLD, 15));
   lblTxtDate.setBounds(555, 64, 199, 19);
   contentPane.add(lblTxtDate);

   JLabel lblAccountOpenBy = new JLabel("Account Open By");

lblAccountOpenBy.setHorizontalAlignment(SwingConstants.CENTER);
   lblAccountOpenBy.setFont(new Font("Tahoma", Font.BOLD, 15));
   lblAccountOpenBy.setBounds(764, 64, 132, 19);
   contentPane.add(lblAccountOpenBy);

   JLabel lblFileNo = new JLabel("File No");
   lblFileNo.setHorizontalAlignment(SwingConstants.CENTER);
   lblFileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
   lblFileNo.setBounds(1016, 64, 96, 19);
   contentPane.add(lblFileNo);

   lblTxtFileNo = new JLabel();
   lblTxtFileNo.setHorizontalAlignment(SwingConstants.CENTER);
   lblTxtFileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
   lblTxtFileNo.setBounds(1122, 64, 96, 19);
   contentPane.add(lblTxtFileNo);

   JLabel lblName = new JLabel("Name");
   lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
   lblName.setBounds(89, 106, 51, 19);
   contentPane.add(lblName);

   JLabel lblFirstName = new JLabel("First Name");
   lblFirstName.setFont(new Font("Tahoma", Font.BOLD, 14));
   lblFirstName.setBounds(111, 131, 96, 19);
   contentPane.add(lblFirstName);

   JLabel lblMiddleName = new JLabel("Middle Name");
   lblMiddleName.setFont(new Font("Tahoma", Font.BOLD, 14));
   lblMiddleName.setBounds(306, 135, 96, 19);
   contentPane.add(lblMiddleName);

   JLabel lblLastName = new JLabel("Last Name");
   lblLastName.setFont(new Font("Tahoma", Font.BOLD, 14));
   lblLastName.setBounds(508, 135, 96, 19);
   contentPane.add(lblLastName);

   txtFName = new JTextField();
   txtFName.setBounds(111, 159, 159, 20);
   contentPane.add(txtFName);
   txtFName.setColumns(10);

   txtMName = new JTextField();
   txtMName.setColumns(10);
   txtMName.setBounds(306, 159, 159, 20);
   contentPane.add(txtMName);
```

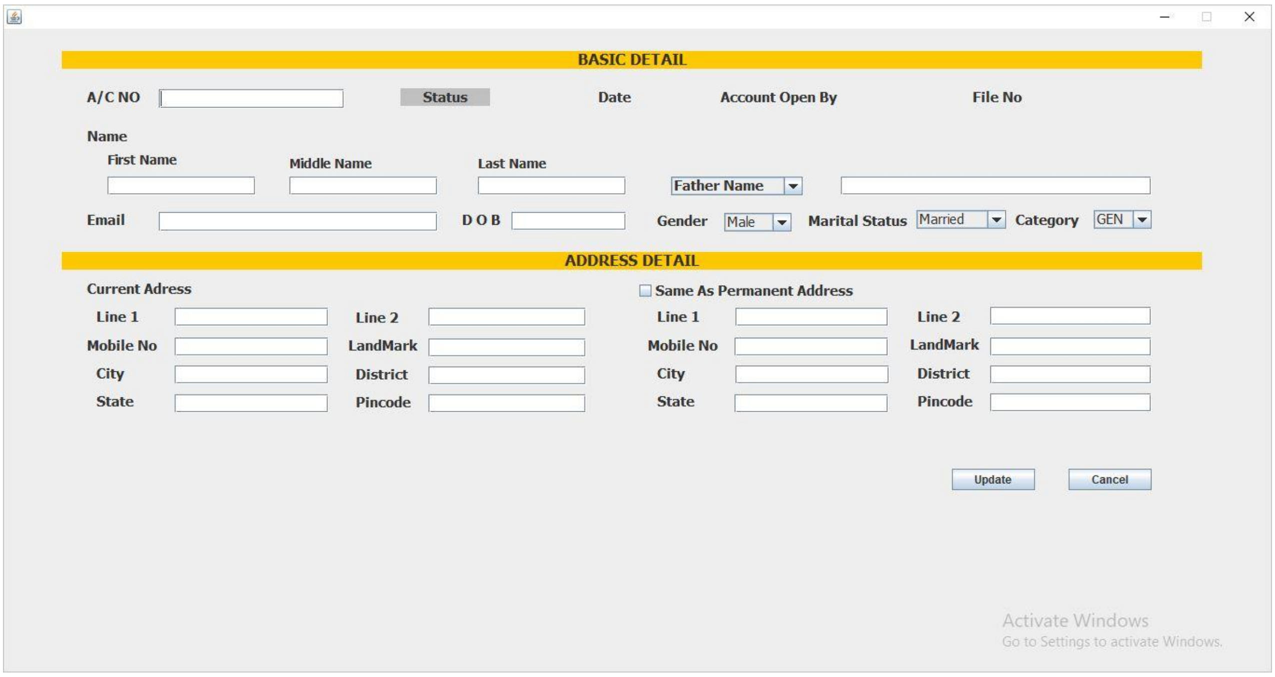
```
txtLName = new JTextField();
txtLName.setColumns(10);
txtLName.setBounds(508, 159, 159, 20);
contentPane.add(txtLName);

JLabel lblEmail = new JLabel("Email");
lblEmail.setFont(new Font("Tahoma", Font.BOLD, 15));
lblEmail.setBounds(89, 196, 51, 19);
contentPane.add(lblEmail);

txtEmail = new JTextField();
txtEmail.setColumns(10);
txtEmail.setBounds(166, 197, 299, 21);
contentPane.add(txtEmail);

comboGType = new JComboBox();
comboGType.setFont(new Font("Tahoma", Font.BOLD, 15));
comboGType.addItem("Father Name");
comboGType.addItem("Husband Name");
comboGType.setBounds(715, 159, 141, 20);
contentPane.add(comboGType);

txtGName = new JTextField();
txtGName.setDragEnabled(true);
txtGName.setBounds(897, 159, 333, 20);
contentPane.add(txtGName);
txtGName.setColumns(10);

JLabel lblDOB = new JLabel("D O B");
lblDOB.setFont(new Font("Tahoma", Font.BOLD, 15));
lblDOB.setBounds(491, 196, 43, 19);
contentPane.add(lblDOB);


JLabel lblGender = new JLabel("Gender");
lblGender.setFont(new Font("Tahoma", Font.BOLD, 15));
lblGender.setBounds(700, 197, 62, 19);
contentPane.add(lblGender);

JLabel lblMatrialStatus = new JLabel("Marital Status");
lblMatrialStatus.setFont(new Font("Tahoma", Font.BOLD, 15));
lblMatrialStatus.setBounds(862, 197, 106, 19);
contentPane.add(lblMatrialStatus);

comboGender = new JComboBox();
comboGender.addItem("Male");
comboGender.addItem("Female");
comboGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
comboGender.setBounds(772, 198, 72, 20);
contentPane.add(comboGender);

comboMaritalStatus = new JComboBox();
comboMaritalStatus.addItem("Married");
comboMaritalStatus.addItem("Unmarried");
```

```
        comboMaritalStatus.setFont(new Font("Tahoma", Font.PLAIN,
15));
        comboMaritalStatus.setBounds(978, 195, 96, 20);
        contentPane.add(comboMaritalStatus);

        JLabel lblCategory = new JLabel("Category");
        lblCategory.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCategory.setBounds(1084, 196, 74, 19);
        contentPane.add(lblCategory);

        comboCategory = new JComboBox();
        comboCategory.addItem("GEN");
        comboCategory.addItem("OBC");
        comboCategory.addItem("ST/SC");
        comboCategory.setFont(new Font("Tahoma", Font.PLAIN, 15));
        comboCategory.setBounds(1168, 195, 62, 20);
        contentPane.add(comboCategory);

        JLabel lblAddressDetail = new JLabel("ADDRESS DETAIL");
        lblAddressDetail.setOpaque(true);

    lblAddressDetail.setHorizontalAlignment(SwingConstants.CENTER);
        lblAddressDetail.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblAddressDetail.setBackground(Color.ORANGE);
        lblAddressDetail.setBounds(62, 240, 1222, 19);
        contentPane.add(lblAddressDetail);

        JLabel lblCurrentAdress = new JLabel("Current Adress");
        lblCurrentAdress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCurrentAdress.setBounds(89, 270, 118, 19);
        contentPane.add(lblCurrentAdress);

        JLabel lblLine = new JLabel("Line 1");
        lblLine.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLine.setBounds(99, 300, 51, 19);
        contentPane.add(lblLine);

        JLabel lblLine_1 = new JLabel("Line 2");
        lblLine_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLine_1.setBounds(377, 301, 51, 19);
        contentPane.add(lblLine_1);

        JLabel lblMobileNo = new JLabel("Mobile No");
        lblMobileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblMobileNo.setBounds(89, 331, 84, 19);
        contentPane.add(lblMobileNo);

        JLabel lblLandmark = new JLabel("LandMark");
        lblLandmark.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLandmark.setBounds(369, 331, 84, 19);
        contentPane.add(lblLandmark);

        JLabel lblCity = new JLabel("City");
        lblCity.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblCity.setBounds(99, 361, 51, 19);
        contentPane.add(lblCity);
```

```
txtLine1 = new JTextField();
txtLine1.setColumns(10);
txtLine1.setBounds(183, 300, 165, 20);
contentPane.add(txtLine1);

txtLine2 = new JTextField();
txtLine2.setColumns(10);
txtLine2.setBounds(455, 300, 169, 20);
contentPane.add(txtLine2);

txtMobile = new JTextField();
txtMobile.setColumns(10);
txtMobile.setBounds(183, 332, 165, 20);
contentPane.add(txtMobile);

txtLandmark = new JTextField();
txtLandmark.setColumns(10);
txtLandmark.setBounds(455, 333, 169, 20);
contentPane.add(txtLandmark);

txtCity = new JTextField();
txtCity.setColumns(10);
txtCity.setBounds(183, 362, 165, 20);
contentPane.add(txtCity);

JLabel lblDistric = new JLabel("District");
lblDistric.setFont(new Font("Tahoma", Font.BOLD, 15));
lblDistric.setBounds(377, 362, 62, 19);
contentPane.add(lblDistric);

JLabel lblState = new JLabel("State");
lblState.setFont(new Font("Tahoma", Font.BOLD, 15));
lblState.setBounds(99, 391, 51, 19);
contentPane.add(lblState);

JLabel lblPincode = new JLabel("Pincode");
lblPincode.setFont(new Font("Tahoma", Font.BOLD, 15));
lblPincode.setBounds(377, 392, 62, 19);
contentPane.add(lblPincode);

txtState = new JTextField();
txtState.setColumns(10);
txtState.setBounds(183, 393, 165, 20);
contentPane.add(txtState);

txtDistrict = new JTextField();
txtDistrict.setColumns(10);
txtDistrict.setBounds(455, 363, 169, 20);
contentPane.add(txtDistrict);

txtPin = new JTextField();
txtPin.setColumns(10);
txtPin.setBounds(455, 393, 169, 20);
contentPane.add(txtPin);
```

```
txtLine1P = new JTextField();
txtLine1P.setColumns(10);
txtLine1P.setBounds(784, 300, 164, 20);
contentPane.add(txtLine1P);

JLabel label_2 = new JLabel("Line 1");
label_2.setFont(new Font("Tahoma", Font.BOLD, 15));
label_2.setBounds(700, 300, 51, 19);
contentPane.add(label_2);

JLabel label_4 = new JLabel("Mobile No");
label_4.setFont(new Font("Tahoma", Font.BOLD, 15));
label_4.setBounds(690, 331, 84, 19);
contentPane.add(label_4);

txtMobileP = new JTextField();
txtMobileP.setColumns(10);
txtMobileP.setBounds(783, 332, 165, 20);
contentPane.add(txtMobileP);

JLabel label_5 = new JLabel("Line 2");
label_5.setFont(new Font("Tahoma", Font.BOLD, 15));
label_5.setBounds(979, 300, 51, 19);
contentPane.add(label_5);

txtLine2P = new JTextField();
txtLine2P.setColumns(10);
txtLine2P.setBounds(1057, 299, 173, 20);
contentPane.add(txtLine2P);

txtLandmarkP = new JTextField();
txtLandmarkP.setColumns(10);
txtLandmarkP.setBounds(1057, 332, 173, 20);
contentPane.add(txtLandmarkP);

JLabel label_6 = new JLabel("LandMark");
label_6.setFont(new Font("Tahoma", Font.BOLD, 15));
label_6.setBounds(971, 330, 84, 19);
contentPane.add(label_6);

txtDistrictP = new JTextField();
txtDistrictP.setColumns(10);
txtDistrictP.setBounds(1057, 362, 173, 20);
contentPane.add(txtDistrictP);

txtPinP = new JTextField();
txtPinP.setColumns(10);
txtPinP.setBounds(1057, 392, 173, 20);
contentPane.add(txtPinP);

JLabel label_7 = new JLabel("Pincode");
label_7.setFont(new Font("Tahoma", Font.BOLD, 15));
label_7.setBounds(979, 391, 62, 19);
contentPane.add(label_7);

JLabel lblDistrict = new JLabel("District");
```

```java
    lblDistrict.setFont(new Font("Tahoma", Font.BOLD, 15));
    lblDistrict.setBounds(979, 361, 68, 19);
    contentPane.add(lblDistrict);

    txtCityP = new JTextField();
    txtCityP.setColumns(10);
    txtCityP.setBounds(784, 362, 165, 20);
    contentPane.add(txtCityP);

    JLabel lblCity_1 = new JLabel("City");
    lblCity_1.setFont(new Font("Tahoma", Font.BOLD, 15));
    lblCity_1.setBounds(700, 361, 51, 19);
    contentPane.add(lblCity_1);

    JLabel label_10 = new JLabel("State");
    label_10.setFont(new Font("Tahoma", Font.BOLD, 15));
    label_10.setBounds(700, 391, 51, 19);
    contentPane.add(label_10);

    txtStateP = new JTextField();
    txtStateP.setColumns(10);
    txtStateP.setBounds(783, 393, 165, 20);
    contentPane.add(txtStateP);

    chckbxAddressP = new JCheckBox("Same As Permanent Address");
    chckbxAddressP.setFont(new Font("Tahoma", Font.BOLD, 15));
    chckbxAddressP.setBounds(677, 270, 237, 23);
    contentPane.add(chckbxAddressP);

    JButton btnCancel = new JButton("Cancel");
    btnCancel.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dispose();
        }
    });
    btnCancel.setBounds(1141, 473, 89, 23);
    contentPane.add(btnCancel);

    JButton btnUpdate = new JButton("Update");
    btnUpdate.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            update();
            dispose();
        }
    });
    btnUpdate.setBounds(1016, 473, 89, 23);
    contentPane.add(btnUpdate);

    txtDOB = new JTextField();
    txtDOB.setColumns(10);
    txtDOB.setBounds(544, 197, 123, 20);
    contentPane.add(txtDOB);
}

private void detail(){
        String sql1="SELECT * FROM sss.memberdetail where acno="+
txtAcno.getText() +"";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql1);
           while (rs.next()){
              lblTxtStatus.setText(rs.getString("status"));
              if(rs.getString("status").equals("Active")){
                 lblTxtStatus.setBackground(Color.GREEN);
                 lblTxtDate.setText(rs.getString("openeddate"));
              }
              else{
                 lblTxtStatus.setBackground(Color.RED);
                 lblTxtDate.setText(rs.getString("deactivateddate"));
              }
              lblTxtFileNo.setText(rs.getString("fileno"));
              txtFName.setText(rs.getString("fname"));
              txtLName.setText(rs.getString("lname"));
              txtMName.setText(rs.getString("mname"));

    if(rs.getString("gtype").equals("Father")){comboGType.setSelecte
dIndex(0);}
              else
if(rs.getString("gtype").equals("Husband")){comboGType.setSelectedIn
dex(1);}
              txtGName.setText(rs.getString("gname"));
              txtEmail.setText(rs.getString("email"));
              txtDOB.setText(rs.getString("dob"));

    if(rs.getString("gender").equals("Male")){comboGender.setSelecte
dIndex(0);}
              else
if(rs.getString("gender").equals("Female")){comboGender.setSelectedI
ndex(1);}

    if(rs.getString("maritalstatus").equals("Married")){comboMarital
Status.setSelectedIndex(0);}
              else
if(rs.getString("maritalstatus").equals("Unmarried")){comboMaritalSt
atus.setSelectedIndex(1);}

    if(rs.getString("category").equals("GEN")){comboCategory.setSele
ctedIndex(0);}
              else
if(rs.getString("category").equals("OBC")){comboCategory.setSelected
Index(1);}
              else
if(rs.getString("category").equals("SC/ST")){comboCategory.setSelect
edIndex(2);}
              txtLine1.setText(rs.getString("line1"));
              txtLine2.setText(rs.getString("line2"));
              txtMobile.setText(rs.getString("mobileno"));
              txtLandmark.setText(rs.getString("landmark"));
              txtCity.setText(rs.getString("city"));
              txtDistrict.setText(rs.getString("distric"));
              txtState.setText(rs.getString("state"));
              txtPin.setText(rs.getString("pincode"));
              txtLine1P.setText(rs.getString("line1p"));
              txtLine2P.setText(rs.getString("line2p"));
              txtMobileP.setText(rs.getString("mobilenop"));
              txtLandmarkP.setText(rs.getString("landmarkp"));
              txtCityP.setText(rs.getString("cityp"));
              txtDistrictP.setText(rs.getString("districp"));
              txtStateP.setText(rs.getString("statep"));
              txtPinP.setText(rs.getString("pincodep"));

    if(rs.getString("line1").equals(rs.getString("line1P")) &&
rs.getString("city").equals(rs.getString("cityP")))
              {
                chckbxAddressP.setSelected(true);
              }
           }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }

    private void update(){
        String
Accountno,FirstName,MiddleName,LastName,GType="",GName,DOB,Email,Mar
italstatus,Gen,Cat,Line1,Line2,Mobile,Lmark,City,Dist,State,Pin,Line
1p,Line2p,Mobilep,Lmarkp,Cityp,Distp,Statep,Pinp,sql;
        Accountno=txtAcno.getText();
        FirstName =txtFName.getText();
        MiddleName = txtMName.getText();
        LastName =txtLName.getText();
        Email= txtEmail.getText();

    Maritalstatus=comboMaritalStatus.getSelectedItem().toString();
        GType=comboGType.getSelectedItem().toString();
        GName =txtGName.getText();
        DOB = txtDOB.getText();
        Gen = comboGender.getSelectedItem().toString();
        Cat = comboCategory.getSelectedItem().toString();
        Line1=txtLine1.getText();
        Line2=txtLine2.getText();
        Mobile=txtMobile.getText();
        Lmark=txtLandmark.getText();
        City= txtCity.getText();
        Dist=txtDistrict.getText();
        State=txtState.getText();
        Pin=txtPin.getText();
        Line1p=txtLine1P.getText();
        Line2p=txtLine2P.getText();
        Mobilep=txtMobileP.getText();
        Lmarkp=txtLandmarkP.getText();
        Cityp=txtCityP.getText();
        Distp=txtDistrictP.getText();
        Statep=txtStateP.getText();
        Pinp=txtPinP.getText();

        sql="UPDATE sss.memberdetail SET fname='"+FirstName+"',
mname='"+MiddleName+"', lname='"+LastName+"', gtype='"+ GType +"',
gname='"+ GName +"', email='"+Email+"',
maritalstatus='"+Maritalstatus+"', dob='"+DOB+"',gender='"+Gen+"',
category='"+Cat+"', line1='"+Line1+"', line2='"+Line2+"',
mobileno='" +Mobile+ "', landmark='"+Lmark+"', city='"+City+"',
distric='"+Dist+"', state='"+State+"', pincode='"+Pin+"',line1p='"+
Line1p +"', line2p='"+ Line2p +"', mobilenop='" +Mobilep+ "',
landmarkp='"+Lmarkp+"', cityp='"+Cityp+"', districp='"+Distp+"',
statep='"+Statep+"', pincodep='"+Pinp+"' WHERE
acno='"+Accountno+"'";
        if (FirstName.equals("") || MiddleName.equals("") ||
LastName.equals("") || Email.equals("") || DOB.equals("") ||
GName.equals("") || Line1.equals("") ||  Mobile.equals("") ||
Lmark.equals("") || City.equals("") || Dist.equals("") ||
State.equals("") || Pin.equals("") || Line1p.equals("") ||
Mobilep.equals("") || Lmarkp.equals("") || Cityp.equals("") ||
Distp.equals("") || Statep.equals("") || Pinp.equals("") ){
            JOptionPane.showMessageDialog(null, "Fill Required
Data");
        }else
        {
            try {
            Statement stmt=dbc.conmethod();
            stmt.execute(sql);
            JOptionPane.showMessageDialog(null, "Detail is Updated");
        }

    catch(com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintVi
olationException e){
            JOptionPane.showInternalMessageDialog(null, "Duplicate
Data");
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
        }
    }
}
```



Father Name

Gender Male

Marital Status Married

Category GEN

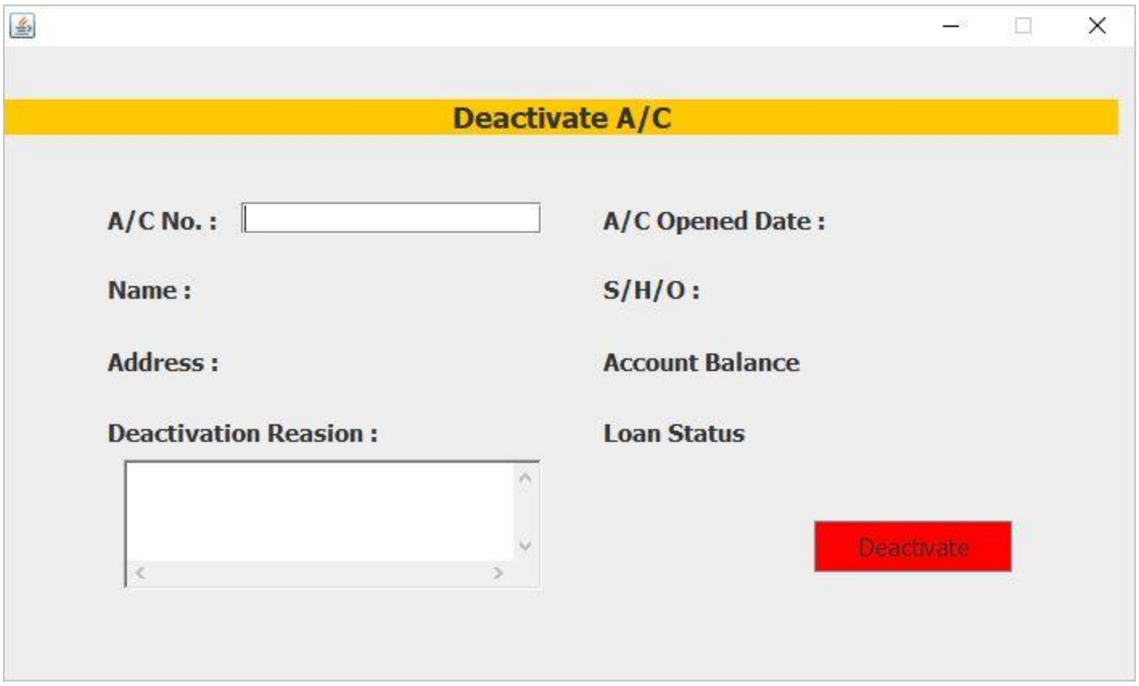
DEACTIVATE MEMBER

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.TextArea;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;

import java.awt.Color;

import javax.swing.SwingConstants;

import java.awt.Font;

import javax.swing.JTextField;
import javax.swing.JButton;

import java.awt.Frame;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```
public class deactivatemember extends JFrame {

    dbconnection dbc = new dbconnection();
    Integer flag=0;
    private JPanel contentPane;
    private JTextField txtAcno;
    private TextArea txtArea;
    private JLabel lblTxtAddress;
    private JLabel lblTxtName;
    private JLabel lblTxtOpenDate;
    private JLabel lblTxtAcBalance;
    private JLabel lblTxtLoanStatus;
    private JLabel lblTxtGName;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    deactivatemember frame = new deactivatemember();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public deactivatemember() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 708, 423);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel  = new JLabel("Deactivate A/C");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setOpaque(true);
        lblNewLabel.setBounds(0, 33, 692, 22);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("A/C No. :");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblNewLabel_1.setBounds(64, 97, 77, 22);
        contentPane.add(lblNewLabel_1);
        JLabel lblName = new JLabel("Name :");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(64, 140, 77, 22);
        contentPane.add(lblName);

        JLabel lblAddress = new JLabel("Address :");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(64, 185, 77, 22);
        contentPane.add(lblAddress);

        JLabel lblDeactivationReasion = new JLabel("Deactivation
Reasion :");
        lblDeactivationReasion.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblDeactivationReasion.setBounds(64, 229, 181, 22);
        contentPane.add(lblDeactivationReasion);

        JLabel lblLoanStatus = new JLabel("Loan Status");
        lblLoanStatus.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanStatus.setBounds(372, 229, 118, 22);
        contentPane.add(lblLoanStatus);

        JLabel lblAccountBalance = new JLabel("Account Balance");
        lblAccountBalance.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblAccountBalance.setBounds(372, 185, 138, 22);
        contentPane.add(lblAccountBalance);

        JLabel lblSho = new JLabel("S/H/O :");
        lblSho.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblSho.setBounds(372, 140, 77, 22);
        contentPane.add(lblSho);

        JLabel lblAcOpenedDate = new JLabel("A/C Opened Date :");
        lblAcOpenedDate.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAcOpenedDate.setBounds(372, 97, 138, 22);
        contentPane.add(lblAcOpenedDate);

        txtAcno = new JTextField();
        txtAcno.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtAcno.setEditable(true);
              } else {
                 txtAcno.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtAcno.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcno.getText());
              if(acno>200000){
                 txtAcno.setEditable(false);
```

```
                    JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                    txtAcno.setText("");
                    txtAcno.setEditable(true);
                }
                if(acno >= 100000 && acno<=199999){
                    detail();
                    status();
                }

                String Accountno=txtAcno.getText(),status="";
                String sql=("SELECT * FROM memberdetail WHERE
acno='"+Accountno+"' ");
                try {
                    Statement stmt=dbc.conmethod();
                    ResultSet rs=stmt.executeQuery(sql);
                    int count =0;
                    while(rs.next()){
                        status=rs.getString("status");

    if(txtAcno.getText().equals(rs.getString("acno"))){
                            count=1;
                        }
                    }
                    if (count==0 && acno >= 100000 && acno<=199999){
                        txtAcno.setEditable(false);
                        JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                        txtAcno.setEditable(true);
                        txtAcno.requestFocus(true);
                    }
                    if(status.equals("Deactivate")){
                        JOptionPane.showMessageDialog(null, "A/C
Already Deactivated");
                        txtAcno.setText("");
                    }
                } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
            }
        });
        txtAcno.setBounds(147, 97, 187, 20);
        contentPane.add(txtAcno);
        txtAcno.setColumns(10);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(151, 142, 183, 22);
        contentPane.add(lblTxtName);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(151, 185, 222, 22);
        contentPane.add(lblTxtAddress);

        lblTxtOpenDate = new JLabel();
        lblTxtOpenDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtOpenDate.setBounds(520, 97, 123, 22);
        contentPane.add(lblTxtOpenDate);

        lblTxtAcBalance = new JLabel();
        lblTxtAcBalance.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAcBalance.setBounds(520, 185, 123, 22);
        contentPane.add(lblTxtAcBalance);

        lblTxtLoanStatus = new JLabel();
        lblTxtLoanStatus.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLoanStatus.setBounds(484, 229, 181, 22);
        contentPane.add(lblTxtLoanStatus);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(460, 140, 183, 22);
        contentPane.add(lblTxtGName);

        txtArea = new TextArea();
        txtArea.setBounds(74, 257, 260, 81);
        contentPane.add(txtArea);

        JButton btnNewButton = new JButton("Deactivate");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                deactivate();
            }
        });
        btnNewButton.setBackground(Color.RED);
        btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 15));
        btnNewButton.setBounds(503, 295, 123, 32);
        contentPane.add(btnNewButton);
    }
    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcno.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
                lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
                lblTxtOpenDate.setText(rs.getString("openeddate"));
                lblTxtGName.setText(rs.getString("gname"));
                lblTxtOpenDate.setText(rs.getString("openeddate"));
                lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("line2")+", "+rs.getString("city"));
                lblTxtAcBalance.setText(rs.getString("amount"));
                }
            }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void status(){
        String status="",sql=("select * from loantaking where
acno="+ txtAcno.getText() +"");
```

```
        int count=0;
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while(rs.next()){
              flag=rs.getInt("flag");
              status=rs.getString("status");
              if(txtAcno.getText().equals(rs.getString("acno"))){
                 count=1;
              }
           }
           if (count==1){
              if(flag==1){
                 lblTxtLoanStatus.setText(status);
              }
              if(flag==0){
                 lblTxtLoanStatus.setText("You have take Loan");
              }
           }else{
              lblTxtLoanStatus.setText("haven't take Loan");
           }
        } catch (Exception ex){ JOptionPane.showMessageDialog(null,
ex);}
    }
    private void deactivate(){
        String Accountno,reason,sql;
        Accountno=txtAcno.getText();
        reason =txtArea.getText();
        sql="UPDATE sss.memberdetail SET
deactivatedreason='"+reason+"', deactivateddate=now(),
status='"+"Deactivate"+"' WHERE acno='"+Accountno+"'";
        if (txtArea.getText().equals("") || reason.equals("")){
           JOptionPane.showMessageDialog(null, "Fill Required
Data");
        }else
        {
           try {
           Statement stmt=dbc.conmethod();
           if(flag==0){
              JOptionPane.showMessageDialog(null, "You haven't paid
the Loan, please paid first");
           }else{
              stmt.execute(sql);
              JOptionPane.showMessageDialog(null, "Account is
Deactivated");
              dispose();
           }
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
        }
    }
}
```



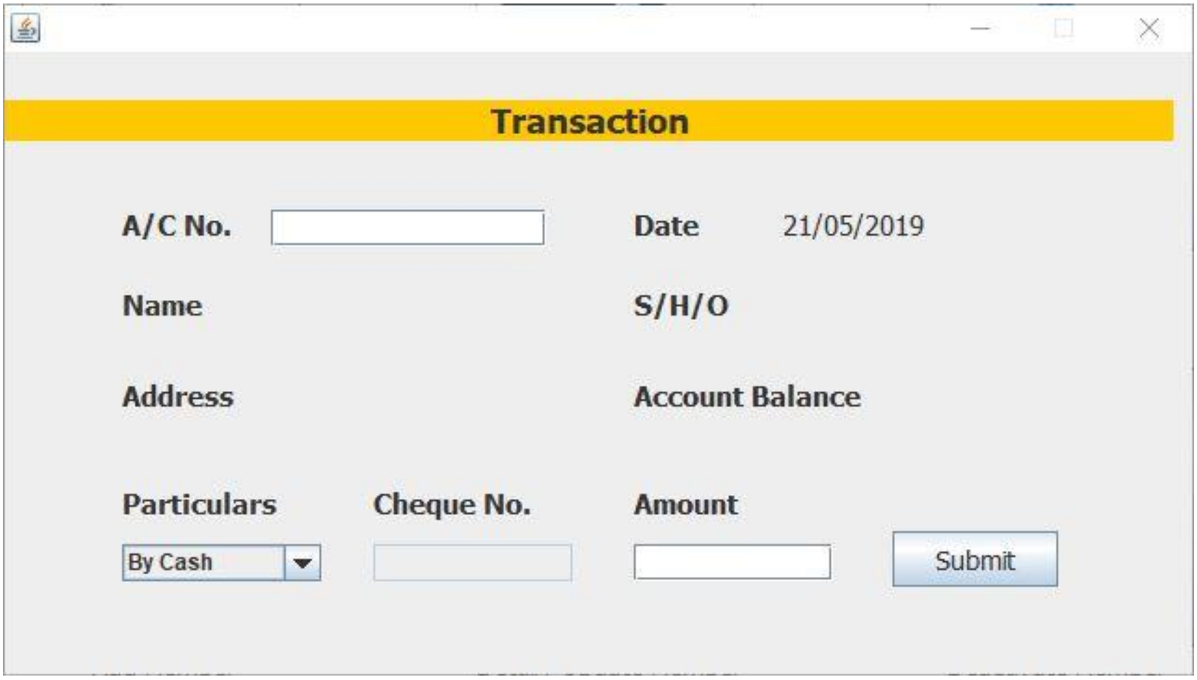
TRANSACTION

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

import java.awt.Color;
import java.awt.Font;

import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;

import java.awt.Frame;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionListener;
```

```java
import java.awt.event.ActionEvent;
import java.lang.*;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;


public class transaction extends JFrame {

    dbconnection dbc = new dbconnection();
    String uname;
    private JPanel contentPane;
    private JTextField txtAcno;
    private JTextField txtAmount;
    private JTextField txtChequeNo;
    private JLabel lblTxtName;
    private JLabel lblTxtAddress;
    private JLabel lblTxtDate;
    private JLabel lblTxtGName;
    private JLabel lblTxtAcBalance;
    private JComboBox comboParticular;
    private Integer currentbalance;

    java.util.Date datet= Calendar.getInstance().getTime();
    DateFormat format= new SimpleDateFormat("dd/MM/yyyy");
    String today = format.format(datet);

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    transaction frame = new transaction(null);
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public transaction(String username) {
        uname=username;
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 644, 364);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        JLabel lblNewLabel = new JLabel("Transaction");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setOpaque(true);
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setBounds(0, 26, 628, 22);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("A/C No.");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblNewLabel_1.setBounds(63, 82, 70, 22);
        contentPane.add(lblNewLabel_1);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(63, 125, 70, 22);
        contentPane.add(lblName);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(63, 174, 70, 22);
        contentPane.add(lblAddress);

        JLabel lblAccountBalance = new JLabel("Account Balance");
        lblAccountBalance.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblAccountBalance.setBounds(338, 174, 122, 22);
        contentPane.add(lblAccountBalance);

        JLabel lblSho = new JLabel("S/H/O");
        lblSho.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblSho.setBounds(338, 125, 70, 22);
        contentPane.add(lblSho);

        JLabel lblDate = new JLabel("Date");
        lblDate.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblDate.setBounds(338, 82, 70, 22);
        contentPane.add(lblDate);

        txtAcno = new JTextField();
        txtAcno.addKeyListener(new KeyAdapter() {
            @Override
            public void keyReleased(KeyEvent e) {if (e.getKeyChar()
>= '0' && e.getKeyChar() <= '9' || e.getKeyChar()==8 ||
e.getKeyChar()==10) {
                txtAcno.setEditable(true);
            } else {
                txtAcno.setEditable(false);
                Toolkit.getDefaultToolkit().beep();
                JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                txtAcno.setEditable(true);
            }
            Integer acno=Integer.valueOf(txtAcno.getText());
            if(acno>200000){
                txtAcno.setEditable(false);
```

```
              JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
              txtAcno.setText("");
              txtAcno.setEditable(true);
           }
           if(acno >= 100000 && acno<=199999){
              detail();
           }

           String Accountno=txtAcno.getText(),status="";
           String sql=("SELECT * FROM memberdetail WHERE
acno='"+Accountno+"' ");
           try {
              Statement stmt=dbc.conmethod();
              ResultSet rs=stmt.executeQuery(sql);
              int count =0;
              while(rs.next()){
                 status=rs.getString("status");
                 if(txtAcno.getText().equals(rs.getString("acno"))){
                      count=1;
                 }
              }
              if (count==0 && acno >= 100000 && acno<=199999){
                 txtAcno.setEditable(false);
                 JOptionPane.showMessageDialog(null, "A/C Not Found,
Enter Valid A/C No.");
                 txtAcno.setText("");
                 txtAcno.setEditable(true);
                 txtAcno.requestFocus(true);
              }
              if(status.equals("Deactivate")){
                 JOptionPane.showMessageDialog(null, "A/C
Deactivated, please Enter Active A/c.");
                 txtAcno.setText("");
              }
           } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        txtAcno.setBounds(143, 85, 148, 20);
        contentPane.add(txtAcno);
        txtAcno.setColumns(10);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(143, 125, 148, 22);
        contentPane.add(lblTxtName);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(143, 174, 148, 22);
        contentPane.add(lblTxtAddress);

        lblTxtDate = new JLabel(today);
        lblTxtDate.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtDate.setBounds(418, 82, 148, 22);
        contentPane.add(lblTxtDate);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(418, 125, 163, 22);
        contentPane.add(lblTxtGName);

        lblTxtAcBalance = new JLabel();
        lblTxtAcBalance.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAcBalance.setBounds(470, 174, 122, 22);
        contentPane.add(lblTxtAcBalance);

        JLabel lblParticulars = new JLabel("Particulars");
        lblParticulars.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblParticulars.setBounds(63, 232, 91, 22);
        contentPane.add(lblParticulars);

        JLabel lblChequeNo = new JLabel("Cheque No.");
        lblChequeNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblChequeNo.setBounds(198, 232, 91, 22);
        contentPane.add(lblChequeNo);

        JLabel lblAmount = new JLabel("Amount");
        lblAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAmount.setBounds(338, 232, 91, 22);
        contentPane.add(lblAmount);

        txtAmount = new JTextField();
        txtAmount.setColumns(10);
        txtAmount.setBounds(338, 265, 107, 20);
        contentPane.add(txtAmount);

        txtChequeNo = new JTextField();
        txtChequeNo.setEditable(false);
        txtChequeNo.setColumns(10);
        txtChequeNo.setBounds(198, 265, 107, 20);
        contentPane.add(txtChequeNo);

        comboParticular = new JComboBox();
        comboParticular.addItemListener(new ItemListener() {
           public void itemStateChanged(ItemEvent e) {
              if(comboParticular.getSelectedItem().equals("By
Cheque")){
                 txtChequeNo.setEditable(true);
              }
              else{
                 txtChequeNo.setEditable(false);
              }
           }
        });
        comboParticular.addItem("By Cash");
        comboParticular.addItem("By Cheque");
        comboParticular.addItem("To Cash");
        comboParticular.setBounds(63, 265, 107, 20);
        contentPane.add(comboParticular);
        JButton btnNewButton = new JButton("Submit");
        btnNewButton.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              transaction1();
           }
        });
        btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 14));
        btnNewButton.setBounds(477, 258, 89, 30);
        contentPane.add(btnNewButton);
    }
    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcno.getText() +"";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              currentbalance=rs.getInt("amount");
              lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
              lblTxtGName.setText(rs.getString("gname"));
              lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("line2")+", "+rs.getString("city"));

    lblTxtAcBalance.setText(String.valueOf(currentbalance));
              }
           }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }



private void transaction1(){
        String accountno,particular,cheque,sql1,sql2;
        Integer totalamount=0,amount=0;
        accountno=txtAcno.getText();
        particular=comboParticular.getSelectedItem().toString();
        cheque=txtChequeNo.getText();
        amount=Integer.parseInt(txtAmount.getText());
        if (particular.equals("By Cash")|| particular.equals("By
Cheque")){
           totalamount=currentbalance+amount;
        } else if(particular.equals("To Cash")){
           totalamount=currentbalance-amount;
        }

        sql1="INSERT INTO
sss.transaction(acno,particular,chequeno,amount,date,time,curentbala
nce,doneby)
values('"+accountno+"','"+particular+"','"+cheque+"','"+amount+"',no
w(),now(),'"+totalamount+"','"+uname+"')";
        sql2="UPDATE sss.memberdetail SET amount='"+totalamount+"'
WHERE acno='"+accountno+"' ";
        if (txtAmount.getText().equals("") || accountno.equals("")
|| (comboParticular.getSelectedItem().equals("To
Cheque")&&txtChequeNo.getText()=="")){
```

```
           JOptionPane.showMessageDialog(null, "Please Fill the all 
field");
        }else{
           try {
           Statement stmt=dbc.connethod();
        
    if(currentbalance<amount&&comboParticular.getSelectedItem().equa
ls("To Cash")){
              JOptionPane.showMessageDialog(null, "Enter Amount is 
more than Account Balance");
              txtAmount.setText(String.valueOf(currentbalance));
           }
           else{
              stmt.execute(sql1);
              stmt.execute(sql2);
              JOptionPane.showMessageDialog(null, "Transaction 
Successsfull");
              dispose();
           }
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null, 
e);}
        }
    }
}
```



Transaction

A/C No.

Date 21/05/2019

Name

S/H/O

Address

Account Balance

Particulars

Cheque No.

Amount

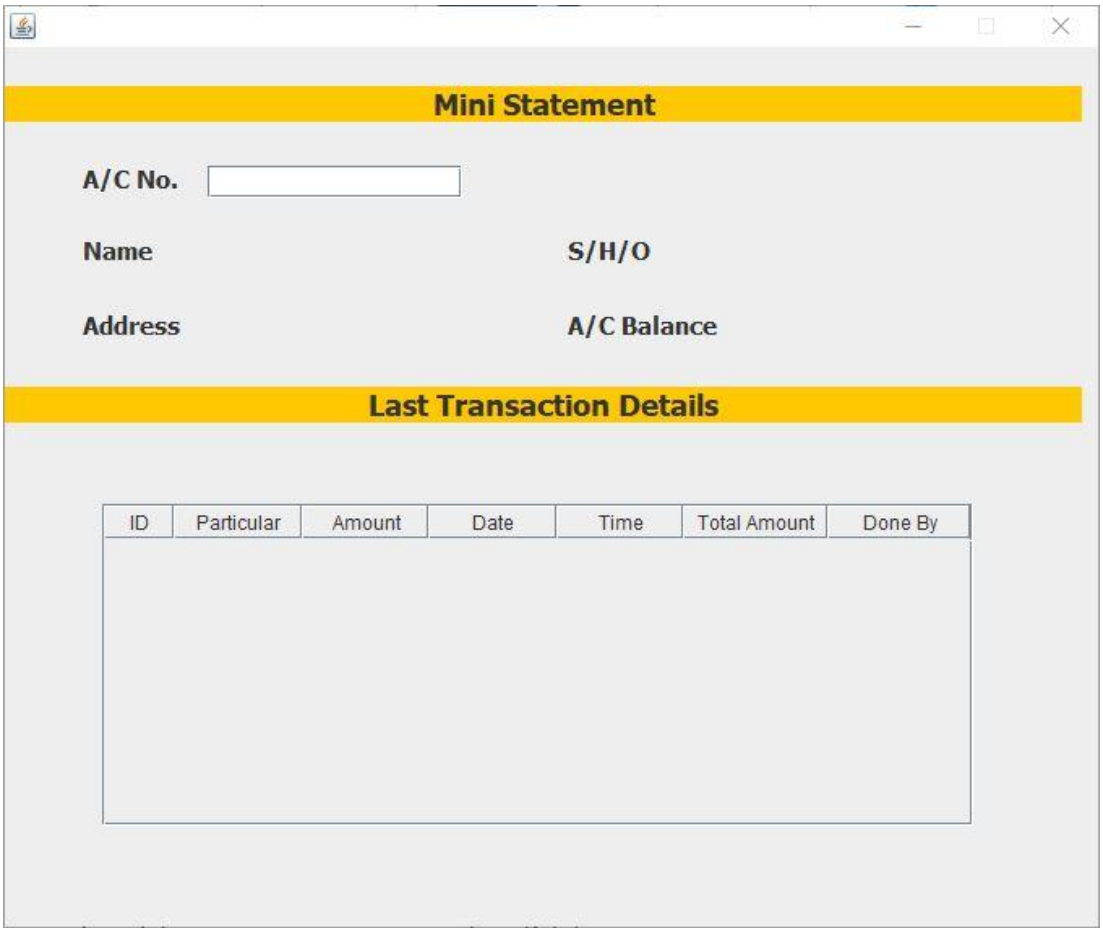
By Cash

TRANSACTION STATEMENT

```java
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

import java.awt.Font;
import java.awt.Color;

import javax.swing.JTextField;

import java.awt.Frame;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;


public class ministatement extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtAcNo;
    private JLabel lblTxtName;
    private JLabel lblTxtAddress;
    private JLabel lblTxtGName;
    private JLabel lblTxtAcBalance;
    private JTable table;
    DefaultTableModel model = new DefaultTableModel();
    Object[] row = new Object[7];

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    ministatement frame = new ministatement();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public ministatement() {
        setResizable(false);
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 680, 570);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Mini Statement");
        lblNewLabel.setOpaque(true);
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setBounds(0, 24, 664, 22);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("A/C No.");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblNewLabel_1.setBounds(48, 70, 76, 22);
        contentPane.add(lblNewLabel_1);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(48, 114, 54, 22);
        contentPane.add(lblName);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(48, 160, 76, 22);
        contentPane.add(lblAddress);

        JLabel lblAcBalance = new JLabel("A/C Balance");
        lblAcBalance.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAcBalance.setBounds(347, 160, 92, 22);
        contentPane.add(lblAcBalance);

        JLabel lblSho = new JLabel("S/H/O");
        lblSho.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblSho.setBounds(347, 114, 76, 22);
        contentPane.add(lblSho);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(124, 114, 157, 22);
        contentPane.add(lblTxtName);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(124, 160, 190, 22);
        contentPane.add(lblTxtAddress);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(418, 114, 190, 22);
        contentPane.add(lblTxtGName);

        lblTxtAcBalance = new JLabel();
        lblTxtAcBalance.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAcBalance.setBounds(449, 160, 159, 22);
        contentPane.add(lblTxtAcBalance);

        txtAcNo = new JTextField();
        txtAcNo.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtAcNo.setEditable(true);
              } else {
                 txtAcNo.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtAcNo.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcNo.getText());
              if(acno>200000){
                 txtAcNo.setEditable(false);
                 reset();
                 JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                 txtAcNo.setText("");
                 txtAcNo.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                 detail();
                 model.getDataVector().removeAllElements();
                 transactiondetail();
              }

              String sql=("SELECT * FROM memberdetail");
              try {
                 Statement stmt=dbc.connethod();
                 ResultSet rs=stmt.executeQuery(sql);
                 int count =0;
                 while(rs.next()){

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                 }
                 if (count==0 && acno >= 100000 && acno<=199999){
                      txtAcNo.setEditable(false);
```

```
                    model.getDataVector().removeAllElements();
                    reset();
                    JOptionPane.showMessageDialog(null, "A/C Not 
Found, Enter Valid A/C No.");
                    txtAcNo.setEditable(true);
                    txtAcNo.requestFocus(true);
                }
            } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
            }
        });
        txtAcNo.setBounds(125, 73, 157, 20);
        contentPane.add(txtAcNo);
        txtAcNo.setColumns(10);
        
        JLabel lblLastFiveDetails = new JLabel("Last Transaction 
Details");
        lblLastFiveDetails.setOpaque(true);
        
    lblLastFiveDetails.setHorizontalAlignment(SwingConstants.CENTER)
;
        lblLastFiveDetails.setFont(new Font("Tahoma", Font.BOLD, 
18));
        lblLastFiveDetails.setBackground(Color.ORANGE);
        lblLastFiveDetails.setBounds(0, 209, 664, 22);
        contentPane.add(lblLastFiveDetails);
        
        table = new JTable();
        table.setBounds(60, 281, 516, 198);
        contentPane.add(table);
        Object[] columns = 
{"ID","Particular","Amount","Date","Time","Total Amount","Done By"};
         table.setModel(model);
         model.setColumnIdentifiers(columns);
         table.setRowHeight(30);
         JScrollPane pane = new JScrollPane(table);
         pane.setBounds(60, 281, 537, 198);
         contentPane.setLayout(null);
         contentPane.add(pane);
         table.getColumnModel().getColumn(0).setPreferredWidth(15);
         table.getColumnModel().getColumn(1).setPreferredWidth(50);
         table.getColumnModel().getColumn(2).setPreferredWidth(50);
         table.getColumnModel().getColumn(3).setPreferredWidth(50);
         table.getColumnModel().getColumn(4).setPreferredWidth(50);
         table.getColumnModel().getColumn(5).setPreferredWidth(60);
         table.getColumnModel().getColumn(6).setPreferredWidth(60);
         table.setAutoResizeMode(JTable.AUTO_RESIZE_LAST_COLUMN);
        
    }
    private void transactiondetail(){
        String id=null, particular=null, amount=null, 
date=null,time=null,curramount=null,doneby=null;
        String sql1="SELECT * FROM sss.transaction where acno='"+ 
txtAcNo.getText() +"' ORDER BY date DESC";
        try {
            Statement stmt=dbc.conmethod();
```

```
           ResultSet rs=stmt.executeQuery(sql1);
           while (rs.next()){
              id = rs.getString("id");
              particular = rs.getString("particular");
              amount = rs.getString("amount");
              date = rs.getString("date");
              time = rs.getString("time");
              curramount = rs.getString("curentbalance");
              doneby = rs.getString("doneby");

              row[0]= id;
              row[1]= particular;
              row[2]= amount;
              row[3]= date;
              row[4]= time;
              row[5]= curramount;
              row[6]= doneby;

              model.addRow(row);
           }

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

    }
    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcNo.getText() +"";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
              lblTxtGName.setText(rs.getString("gname"));
              lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("line2")+", "+rs.getString("city"));
              lblTxtAcBalance.setText("amount");
              }
           }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void reset(){
        lblTxtName.setText("");
        lblTxtGName.setText("");
        lblTxtAddress.setText("");
        lblTxtAcBalance.setText("");
    }
}
```



Mini Statement

A/C No.

Name

S/H/O

Address

A/C Balance

Last Transaction Details

ID	Particular	Amount	Date	Time	Total Amount	Done By

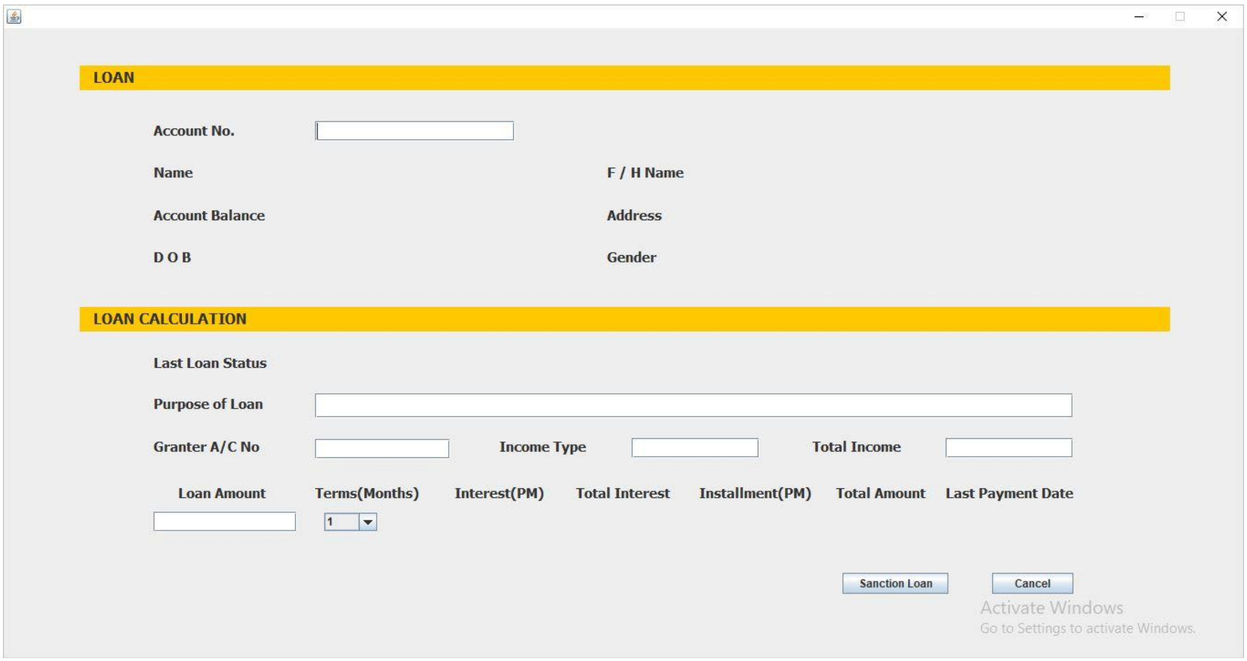
LOAN SACTION

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Frame;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;

import java.awt.Font;
import java.awt.Color;

import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JComboBox;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;


public class loantaking extends JFrame {

    dbconnection dbc = new dbconnection();
    Integer loanacno;
    String uname;
    private JPanel contentPane;
    private JTextField txtAcNo;
    private JTextField txtPurpose;
    private JTextField txtGranterAc;
    private JTextField txtIncomeType;
    private JTextField txtTotalIncome;
    private JTextField txtLoanAmount;
    private JLabel lblTxtName;
    private JLabel lblTxtAcBalance;
    private JLabel lblTxtDOB;
    private JLabel lblTxtGender;
    private JLabel lblTxtAddress;
    private JLabel lblTxtGName;
    private JLabel lblTxtLoanStatus;
    private JLabel lblTxtInterest;
    private JLabel lblTxtTotalInterest;
    private JLabel lblTxtInstallment;
    private JLabel lblTxtTotalAmount;
    private JLabel lblTxtLastPaymentDate;
    private JComboBox comboTerms;

    java.util.Date datet= Calendar.getInstance().getTime();
    DateFormat formatter= new SimpleDateFormat("dd/MM/yyyy");
    DateFormat formattertime= new SimpleDateFormat("hh:mm:ssa");
    DateFormat formatteryear= new SimpleDateFormat("yyyy");
    DateFormat formattermonth= new SimpleDateFormat("MM");
    DateFormat formatterday= new SimpleDateFormat("dd");
    String today = formatter.format(datet);
    String time = formattertime.format(datet);
    String year = formatteryear.format(datet);
    String month = formattermonth.format(datet);
    String day = formatterday.format(datet);
    String d=null;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    loantaking frame = new loantaking(null);
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public loantaking(String username) {
        uname=username;
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        setResizable(false);
        setSize(new Dimension(1364,720));
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        //auto increment loan account no.
        int count =1;
        String sql=("select acno from loantaking");
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while(rs.next()){
                count+=1;
            }
        } catch (Exception e) { JOptionPane.showMessageDialog(null,
e);}
        loanacno=100+count;

        JLabel lblLo = new JLabel("   LOAN");
        lblLo.setBackground(Color.ORANGE);
        lblLo.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblLo.setOpaque(true);
        lblLo.setBounds(83, 41, 1195, 27);
        contentPane.add(lblLo);

        JLabel lblAccountNo = new JLabel("Account No.");
        lblAccountNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAccountNo.setBounds(164, 102, 108, 20);
        contentPane.add(lblAccountNo);

        txtAcNo = new JTextField();
        txtAcNo.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtAcNo.addKeyListener(new KeyAdapter() {
            @Override
            public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                txtAcNo.setEditable(true);
              } else {
                txtAcNo.setEditable(false);
                Toolkit.getDefaultToolkit().beep();
                JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                txtAcNo.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcNo.getText());
              if(acno>200000){
                txtAcNo.setEditable(false);
                JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                txtAcNo.setText("");
                txtAcNo.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                detail();
                status();
              }

              String sql=("SELECT * FROM memberdetail");
              try {
                Statement stmt=dbc.conmethod();
                ResultSet rs=stmt.executeQuery(sql);
                int count =0;
                while(rs.next()){

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                }
                if (count==0 && acno >= 100000 && acno<=199999){
                      txtAcNo.setEditable(false);
                      JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                      txtAcNo.setText("");
                      txtAcNo.setEditable(true);
                      txtAcNo.requestFocus(true);
                }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        txtAcNo.setBounds(341, 102, 219, 22);
        contentPane.add(txtAcNo);
        txtAcNo.setColumns(10);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(164, 148, 80, 20);
        contentPane.add(lblName);

        JLabel lblAccountBalance = new JLabel("Account Balance");
        lblAccountBalance.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblAccountBalance.setBounds(164, 195, 122, 20);
        contentPane.add(lblAccountBalance);

        JLabel lblDOB = new JLabel("D O B");
        lblDOB.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblDOB.setBounds(164, 241, 92, 20);
        contentPane.add(lblDOB);

        JLabel lblGender = new JLabel("Gender");
        lblGender.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGender.setBounds(661, 241, 92, 20);
        contentPane.add(lblGender);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(661, 195, 122, 20);
        contentPane.add(lblAddress);

        JLabel lblFH = new JLabel("F / H Name");
        lblFH.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblFH.setBounds(661, 148, 92, 20);
        contentPane.add(lblFH);

        lblTxtAcBalance = new JLabel();
        lblTxtAcBalance.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAcBalance.setBounds(341, 195, 190, 20);
        contentPane.add(lblTxtAcBalance);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(341, 148, 190, 20);
        contentPane.add(lblTxtName);

        lblTxtDOB = new JLabel();
        lblTxtDOB.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtDOB.setBounds(341, 241, 190, 20);
        contentPane.add(lblTxtDOB);

        lblTxtGender = new JLabel();
        lblTxtGender.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGender.setBounds(812, 241, 214, 20);
        contentPane.add(lblTxtGender);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(812, 195, 413, 20);
        contentPane.add(lblTxtAddress);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(812, 148, 214, 20);
        contentPane.add(lblTxtGName);
        JLabel lblLoanCalculation = new JLabel("   LOAN
CALCULATION");
        lblLoanCalculation.setOpaque(true);
        lblLoanCalculation.setFont(new Font("Tahoma", Font.BOLD,
16));
        lblLoanCalculation.setBackground(Color.ORANGE);
        lblLoanCalculation.setBounds(83, 306, 1195, 27);
        contentPane.add(lblLoanCalculation);

        JLabel lblLastLoanStatus = new JLabel("Last Loan Status");
        lblLastLoanStatus.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblLastLoanStatus.setBounds(164, 357, 140, 20);
        contentPane.add(lblLastLoanStatus);

        JLabel lblPurposeOfLoan = new JLabel("Purpose of Loan");
        lblPurposeOfLoan.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPurposeOfLoan.setBounds(164, 402, 140, 20);
        contentPane.add(lblPurposeOfLoan);

        lblTxtLoanStatus = new JLabel();
        lblTxtLoanStatus.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLoanStatus.setBounds(341, 357, 412, 20);
        contentPane.add(lblTxtLoanStatus);

        txtPurpose = new JTextField();
        txtPurpose.setFont(new Font("Tahoma", Font.PLAIN, 17));
        txtPurpose.setColumns(10);
        txtPurpose.setBounds(341, 401, 831, 27);
        contentPane.add(txtPurpose);

        JLabel lblGranterAcNo = new JLabel("Granter A/C No");
        lblGranterAcNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGranterAcNo.setBounds(164, 449, 140, 20);
        contentPane.add(lblGranterAcNo);

        txtGranterAc = new JTextField();
        txtGranterAc.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                txtGranterAc.setEditable(true);
              } else {
                txtGranterAc.setEditable(false);
                Toolkit.getDefaultToolkit().beep();
                JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                txtGranterAc.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtGranterAc.getText());
              if(acno>200000){
                txtGranterAc.setEditable(false);
                JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
```

```
                txtGranterAc.setText("");
                txtGranterAc.setEditable(true);
              }
              String sql=("select * from memberdetail");
              int count=0;
              try {
                Statement stmt=dbc.conmethod();
                ResultSet rs=stmt.executeQuery(sql);
                while(rs.next()){

    if(txtGranterAc.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                }
                if (count==0 && acno >= 100000 && acno<=199999){
                      JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                      txtGranterAc.requestFocus(true);
                }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        txtGranterAc.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtGranterAc.setColumns(10);
        txtGranterAc.setBounds(341, 451, 148, 22);
        contentPane.add(txtGranterAc);

        JLabel lblIncomeType = new JLabel("Income Type");
        lblIncomeType.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblIncomeType.setBounds(543, 449, 116, 20);
        contentPane.add(lblIncomeType);

        txtIncomeType = new JTextField();
        txtIncomeType.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtIncomeType.setColumns(10);
        txtIncomeType.setBounds(688, 450, 140, 22);
        contentPane.add(txtIncomeType);

        JLabel lblTotalIncome = new JLabel("Total Income");
        lblTotalIncome.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalIncome.setBounds(886, 449, 116, 20);
        contentPane.add(lblTotalIncome);

        txtTotalIncome = new JTextField();
        txtTotalIncome.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtTotalIncome.setColumns(10);
        txtTotalIncome.setBounds(1032, 450, 140, 22);
        contentPane.add(txtTotalIncome);

        JLabel lblLoanAmount = new JLabel("Loan Amount");
        lblLoanAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanAmount.setBounds(191, 500, 108, 20);
        contentPane.add(lblLoanAmount);

        txtLoanAmount = new JTextField();
        txtLoanAmount.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              interest();
           }
        });
        txtLoanAmount.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtLoanAmount.setColumns(10);
        txtLoanAmount.setBounds(164, 531, 157, 22);
        contentPane.add(txtLoanAmount);

        JLabel lblTermmonths = new JLabel("Terms(Months)");
        lblTermmonths.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTermmonths.setBounds(341, 500, 116, 20);
        contentPane.add(lblTermmonths);

        JLabel lblInterestpm = new JLabel("Interest(PM)");
        lblInterestpm.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInterestpm.setBounds(494, 500, 108, 20);
        contentPane.add(lblInterestpm);

        JLabel lblTotalInterest = new JLabel("Total Interest");
        lblTotalInterest.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalInterest.setBounds(627, 500, 108, 20);
        contentPane.add(lblTotalInterest);

        JLabel lblInstallmentpm = new JLabel("Installment(PM)");
        lblInstallmentpm.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInstallmentpm.setBounds(762, 500, 140, 20);
        contentPane.add(lblInstallmentpm);

        JLabel lblTotalPayment = new JLabel("Total Amount");
        lblTotalPayment.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalPayment.setBounds(912, 500, 108, 20);
        contentPane.add(lblTotalPayment);

        JLabel lblLastPaymentDate = new JLabel("Last Payment Date");
        lblLastPaymentDate.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblLastPaymentDate.setBounds(1032, 500, 140, 20);
        contentPane.add(lblLastPaymentDate);

        lblTxtInterest = new JLabel();
        lblTxtInterest.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtInterest.setBounds(494, 533, 108, 20);
        contentPane.add(lblTxtInterest);

        lblTxtTotalInterest = new JLabel();
        lblTxtTotalInterest.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalInterest.setBounds(627, 533, 108, 20);
        contentPane.add(lblTxtTotalInterest);

        lblTxtInstallment = new JLabel();
        lblTxtInstallment.setFont(new Font("Tahoma", Font.PLAIN,
15));
```

```
        lblTxtInstallment.setBounds(762, 533, 140, 20);
        contentPane.add(lblTxtInstallment);

        lblTxtTotalAmount = new JLabel();
        lblTxtTotalAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalAmount.setBounds(912, 533, 108, 20);
        contentPane.add(lblTxtTotalAmount);

        lblTxtLastPaymentDate = new JLabel();
        lblTxtLastPaymentDate.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLastPaymentDate.setBounds(1032, 533, 140, 20);
        contentPane.add(lblTxtLastPaymentDate);

        comboTerms = new JComboBox();
        for(Integer i=1; i<=60; i++){
           comboTerms.addItem(i);
        }
        comboTerms.addItemListener(new ItemListener() {
           public void itemStateChanged(ItemEvent e) {
              interest();
           }
        });
        comboTerms.setBounds(351, 532, 58, 20);
        contentPane.add(comboTerms);

        JButton btnSanctionLoan = new JButton("Sanction Loan");
        btnSanctionLoan.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              loanSaction();
           }
        });
        btnSanctionLoan.setBounds(919, 598, 116, 23);
        contentPane.add(btnSanctionLoan);

        JButton btnCancel = new JButton("Cancel");
        btnCancel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnCancel.setBounds(1083, 598, 89, 23);
        contentPane.add(btnCancel);
    }

    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcNo.getText() +"";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
              lblTxtGName.setText(rs.getString("gname"));
              lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("line2")+", "+rs.getString("city")+",
"+rs.getString("distric")+", "+rs.getString("state"));
              lblTxtAcBalance.setText(rs.getString("amount"));
              lblTxtGender.setText(rs.getString("gender"));
              lblTxtDOB.setText(rs.getString("dob"));
              }
           }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void status(){
        String status="",sql=("select * from loantaking where
acno="+ txtAcNo.getText() +"");
        int count=0,flag=0;
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while(rs.next()){
              flag=rs.getInt("flag");
              status=rs.getString("status");
              if(txtAcNo.getText().equals(rs.getString("acno"))){
                 count=1;
              }
           }
           if (count==1){
              if(flag==1){
                 lblTxtLoanStatus.setText(status);
              }
              if(flag==0){
                 lblTxtLoanStatus.setText("You have already taken the
Loan");
              }
           }else{
              lblTxtLoanStatus.setText("You haven't take any Loan
Yet");
           }
        } catch (Exception ex){ JOptionPane.showMessageDialog(null,
ex);}
    }
    private void interest() {
        int Terms, LoanAmount=0, TotalInterest,TotalAmount,
Installment, InterestPM;
        Terms=
Integer.parseInt(comboTerms.getSelectedItem().toString());
        if(txtLoanAmount.getText().isEmpty()){}else{
        LoanAmount=Integer.parseInt(txtLoanAmount.getText());}
        TotalInterest=(LoanAmount * Terms*6)/100;
        TotalAmount=LoanAmount+TotalInterest;
        Installment=TotalAmount/Terms;
        InterestPM= TotalInterest/Terms;
        lblTxtTotalAmount.setText(String.valueOf(TotalAmount));
        lblTxtInstallment.setText(String.valueOf(Installment));
        lblTxtTotalInterest.setText(String.valueOf(TotalInterest));
        lblTxtInterest.setText(String.valueOf(InterestPM));

        //Last Payment Date
        Integer a,c,m,y;
        c=Integer.parseInt(comboTerms.getSelectedItem().toString());
        c=c+Integer.parseInt(month);
        a=c/12;
        m=c%12;
        y=Integer.parseInt(year)+a;
        d= day+"/"+m.toString()+"/"+y.toString();
        lblTxtLastPaymentDate.setText(d);
    }
    private void loanSaction(){
        String
acno,purposeofloan,granteracno,incometype,totalincome,loanamount="",
terms,interest,totalinterest,installment,totalamount,paymenttill=nul
l,sql1;
        acno = txtAcNo.getText();
        purposeofloan = txtPurpose.getText();
        granteracno = txtGranterAc.getText();
        incometype = txtIncomeType.getText();
        totalincome = txtTotalIncome.getText();
        loanamount = txtLoanAmount.getText();
        terms= comboTerms.getSelectedItem().toString();
        interest = lblTxtInterest.getText();
        totalinterest = lblTxtTotalInterest.getText();
        installment = lblTxtInstallment.getText();
        totalamount = lblTxtTotalAmount.getText();
        paymenttill = lblTxtLastPaymentDate.getText();


        sql1="INSERT INTO
sss.loantaking(id,acno,date,time,loanamount,duration,totalamount,tot
alinterest,interest,installment,paidtill,purposeofloan,granteracno,i
ncometype,totalincome,doneby)
values('"+loanacno+"','"+acno+"',now(),now(),'"+loanamount+"','"+ter
ms+"','"+totalamount+"','"+totalinterest+"','"+interest+"','"+instal
lment+"','"+paymenttill+"','"+purposeofloan+"','"+granteracno+"','"+
incometype+"','"+totalincome+"','"+uname+"')";
        if (purposeofloan.equals("") || granteracno.equals("") ||
loanamount.equals("") ){
            JOptionPane.showMessageDialog(null, "Fill Requried
Data");
        }else
        {
            try {
               Statement stmt=dbc.conmethod();
               stmt.execute(sql1);
               JOptionPane.showMessageDialog(null, "Loan Saction");
               dispose();
               stmt.close();
            }

    catch(com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintVi
olationException e){
               JOptionPane.showInternalMessageDialog(null, "Duplicate
Data");
            }
```

```
              catch (SQLException
e){JOptionPane.showMessageDialog(null, e); }
              catch(java.lang.NullPointerException ex){
JOptionPane.showMessageDialog(null, ex); }}
    }
}
```



LOAN

Account No.

Name

F / H Name

Account Balance

Address

D O B

Gender

LOAN CALCULATION

Last Loan Status

Purpose of Loan

Granter A/C No

Income Type

Total Income

Loan Amount

Terms(Months) 1

Interest(PM)

Total Interest

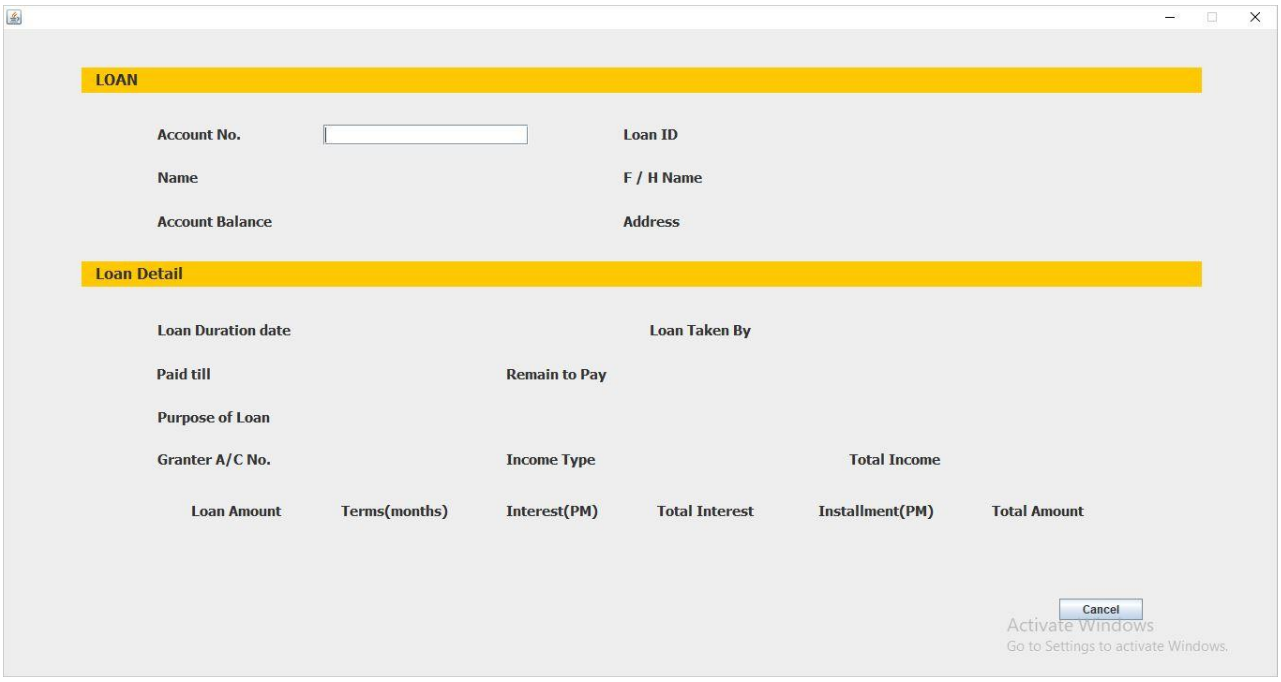
Installment(PM)

Total Amount

Last Payment Date

LOAN DETAIL

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Frame;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.sql.ResultSet;
import java.sql.Statement;

public class loandetail extends JFrame {
    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtAcNo;
    private JLabel lblTxtName;
    private JLabel lblTxtAcBalance;
    private JLabel lblTxtAddress;
    private JLabel lblTxtGName;
    private JLabel lblTxtPurposeOfLoan;
    private JLabel lblTxtGranterAcNo;
    private JLabel lblTxtIncomeType;
    private JLabel lblTxtTotalIncome;
    private JLabel lblTxtLoanAmount;
    private JLabel lblTxtTerms;
    private JLabel lblTxtInterest;
    private JLabel lblTxtTotalAmount;
    private JLabel lblTxtInstallment;
    private JLabel lblTxtTotalInterest;
    private JLabel lblTxtRemainToPay;
    private JLabel lblTxtPaidTill;
    private JLabel lblTxtDuration;
    private JLabel lblTxtLoanId;
    private JLabel lblTxtLoanBy;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    loandetail frame = new loandetail();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public loandetail() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        setResizable(false);
        setSize(new Dimension(1364,720));
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblLo = new JLabel("   LOAN");
        lblLo.setBackground(Color.ORANGE);
        lblLo.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblLo.setOpaque(true);
        lblLo.setBounds(83, 41, 1195, 27);
        contentPane.add(lblLo);

        JLabel lblAccountNo = new JLabel("Account No.");
        lblAccountNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAccountNo.setBounds(164, 102, 108, 20);
        contentPane.add(lblAccountNo);

        txtAcNo = new JTextField();
        txtAcNo.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {

              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtAcNo.setEditable(true);
              } else {
                 txtAcNo.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtAcNo.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcNo.getText());
              if(acno>200000){
                 txtAcNo.setEditable(false);
                 JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                 txtAcNo.setText("");
                 txtAcNo.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                 detail();
                 loandetail();
              }
              String sql=("SELECT * FROM
memberdetail"),sql1=("SELECT * FROM loantaking");
              int count=0,count1=0;
              try {
                 Statement stmt=dbc.conmethod();
                 ResultSet rs=stmt.executeQuery(sql);
                 while(rs.next()){

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                 }
                 if (count==0 && acno >= 100000 && acno<=199999){
                      txtAcNo.setEditable(false);
                      JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                      txtAcNo.setText("");
                      txtAcNo.setEditable(true);
                      txtAcNo.requestFocus(true);
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
              try {
                 Statement stmt=dbc.conmethod();
                 ResultSet rs=stmt.executeQuery(sql1);
                 while(rs.next()){

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                             count1=1;
                       }
                 }
                 if (count==1 && count1==0 && acno >= 100000 &&
acno<=199999){
                       txtAcNo.setEditable(false);
                       JOptionPane.showMessageDialog(null, "You
havn't take any Loan Yet, Enter Valid A/C No.");
                       txtAcNo.setText("");
                       txtAcNo.setEditable(true);
                       txtAcNo.requestFocus(true);
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
            }
        });
        txtAcNo.setBounds(341, 102, 219, 22);
        contentPane.add(txtAcNo);
        txtAcNo.setColumns(10);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(164, 148, 80, 20);
        contentPane.add(lblName);

        JLabel lblAccountBalance = new JLabel("Account Balance");
        lblAccountBalance.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblAccountBalance.setBounds(164, 195, 122, 20);
        contentPane.add(lblAccountBalance);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(661, 195, 122, 20);
        contentPane.add(lblAddress);

        JLabel lblFH = new JLabel("F / H Name");
        lblFH.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblFH.setBounds(661, 148, 92, 20);
        contentPane.add(lblFH);

        lblTxtAcBalance = new JLabel();
        lblTxtAcBalance.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAcBalance.setBounds(341, 195, 190, 20);
        contentPane.add(lblTxtAcBalance);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(341, 148, 190, 20);
        contentPane.add(lblTxtName);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(812, 195, 379, 20);
        contentPane.add(lblTxtAddress);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(812, 148, 314, 20);
        contentPane.add(lblTxtGName);

        JLabel lblLoanDetail = new JLabel("    Loan Detail");
        lblLoanDetail.setOpaque(true);
        lblLoanDetail.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblLoanDetail.setBackground(Color.ORANGE);
        lblLoanDetail.setBounds(83, 248, 1195, 27);
        contentPane.add(lblLoanDetail);

        JLabel lblLoanDuratinDate = new JLabel("Loan Duration
date");
        lblLoanDuratinDate.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblLoanDuratinDate.setBounds(164, 311, 152, 20);
        contentPane.add(lblLoanDuratinDate);

        JLabel lblPaidTill = new JLabel("Paid till");
        lblPaidTill.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPaidTill.setBounds(163, 358, 71, 20);
        contentPane.add(lblPaidTill);

        JLabel lblRemainToPay = new JLabel("Remain to Pay");
        lblRemainToPay.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRemainToPay.setBounds(536, 358, 108, 20);
        contentPane.add(lblRemainToPay);

        JLabel lblPurposeOfLoan = new JLabel("Purpose of Loan");
        lblPurposeOfLoan.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPurposeOfLoan.setBounds(164, 404, 122, 20);
        contentPane.add(lblPurposeOfLoan);

        JLabel lblGranterAcNo = new JLabel("Granter A/C No.");
        lblGranterAcNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblGranterAcNo.setBounds(164, 449, 122, 20);
        contentPane.add(lblGranterAcNo);

        JLabel lblIncomeType = new JLabel("Income Type");
        lblIncomeType.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblIncomeType.setBounds(536, 449, 108, 20);
        contentPane.add(lblIncomeType);

        JLabel lblTotalIncome = new JLabel("Total Income");
        lblTotalIncome.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalIncome.setBounds(902, 449, 108, 20);
        contentPane.add(lblTotalIncome);

        JLabel lblLoanAmount = new JLabel("Loan Amount");
        lblLoanAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanAmount.setBounds(200, 504, 108, 20);
        contentPane.add(lblLoanAmount);
        JLabel lblTermsmonths = new JLabel("Terms(months)");
        lblTermsmonths.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTermsmonths.setBounds(360, 504, 122, 20);
        contentPane.add(lblTermsmonths);

        JLabel lblInterest = new JLabel("Interest(PM)");
        lblInterest.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInterest.setBounds(536, 504, 108, 20);
        contentPane.add(lblInterest);

        JLabel lblTotalInterest = new JLabel("Total Interest");
        lblTotalInterest.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalInterest.setBounds(697, 504, 108, 20);
        contentPane.add(lblTotalInterest);

        JLabel lblInstallmentpm = new JLabel("Installment(PM)");
        lblInstallmentpm.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInstallmentpm.setBounds(869, 504, 132, 20);
        contentPane.add(lblInstallmentpm);

        JLabel lblTotalAmount = new JLabel("Total Amount");
        lblTotalAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalAmount.setBounds(1054, 504, 108, 20);
        contentPane.add(lblTotalAmount);

        JButton btnNewButton = new JButton("Cancel");
        btnNewButton.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnNewButton.setBounds(1126, 608, 89, 23);
        contentPane.add(btnNewButton);

        lblTxtDuration = new JLabel();
        lblTxtDuration.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtDuration.setBounds(341, 311, 325, 20);
        contentPane.add(lblTxtDuration);

        lblTxtPaidTill = new JLabel();
        lblTxtPaidTill.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtPaidTill.setBounds(263, 358, 190, 20);
        contentPane.add(lblTxtPaidTill);

        lblTxtRemainToPay = new JLabel();
        lblTxtRemainToPay.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtRemainToPay.setBounds(691, 358, 204, 20);
        contentPane.add(lblTxtRemainToPay);

        lblTxtPurposeOfLoan = new JLabel();
        lblTxtPurposeOfLoan.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtPurposeOfLoan.setBounds(326, 401, 889, 27);
        contentPane.add(lblTxtPurposeOfLoan);

        lblTxtGranterAcNo = new JLabel();
        lblTxtGranterAcNo.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtGranterAcNo.setBounds(308, 449, 175, 20);
        contentPane.add(lblTxtGranterAcNo);

        lblTxtIncomeType = new JLabel();
        lblTxtIncomeType.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtIncomeType.setBounds(663, 449, 152, 20);
        contentPane.add(lblTxtIncomeType);

        lblTxtTotalIncome = new JLabel();
        lblTxtTotalIncome.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalIncome.setBounds(1040, 449, 175, 20);
        contentPane.add(lblTxtTotalIncome);

        lblTxtLoanAmount = new JLabel();
        lblTxtLoanAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLoanAmount.setBounds(200, 535, 108, 20);
        contentPane.add(lblTxtLoanAmount);

        lblTxtTerms = new JLabel();
        lblTxtTerms.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtTerms.setBounds(360, 535, 122, 20);
        contentPane.add(lblTxtTerms);

        lblTxtInterest = new JLabel();
        lblTxtInterest.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtInterest.setBounds(536, 535, 108, 20);
        contentPane.add(lblTxtInterest);

        lblTxtTotalAmount = new JLabel();
        lblTxtTotalAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalAmount.setBounds(1054, 535, 108, 20);
        contentPane.add(lblTxtTotalAmount);

        lblTxtInstallment = new JLabel();
        lblTxtInstallment.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtInstallment.setBounds(869, 535, 132, 20);
        contentPane.add(lblTxtInstallment);

        lblTxtTotalInterest = new JLabel();
        lblTxtTotalInterest.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalInterest.setBounds(697, 535, 108, 20);
        contentPane.add(lblTxtTotalInterest);

        JLabel lblLoanId = new JLabel("Loan ID");
        lblLoanId.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanId.setBounds(661, 102, 92, 20);
        contentPane.add(lblLoanId);

        lblTxtLoanId = new JLabel();
        lblTxtLoanId.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtLoanId.setBounds(812, 102, 314, 20);
        contentPane.add(lblTxtLoanId);

        lblTxtLoanBy = new JLabel();
        lblTxtLoanBy.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtLoanBy.setBounds(844, 311, 204, 20);
        contentPane.add(lblTxtLoanBy);

        JLabel lblLoanAssignedBy = new JLabel("Loan Taken By");
        lblLoanAssignedBy.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblLoanAssignedBy.setBounds(689, 311, 126, 20);
        contentPane.add(lblLoanAssignedBy);
    }
    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
                lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
                lblTxtGName.setText(rs.getString("gname"));
                lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("line2")+", "+rs.getString("city")+",
"+rs.getString("distric")+", "+rs.getString("state"));
                lblTxtAcBalance.setText(rs.getString("amount"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void loandetail(){
        String sql="SELECT * FROM sss.loantaking where acno="+
txtAcNo.getText() +"";
        Integer restamount;
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
                restamount=rs.getInt("totalamount")-
rs.getInt("totalamountpaid");
                lblTxtLoanId.setText(rs.getString("id"));

    lblTxtPurposeOfLoan.setText(rs.getString("purposeofloan"));

    lblTxtGranterAcNo.setText(rs.getString("granteracno"));
                lblTxtIncomeType.setText(rs.getString("incometype"));

    lblTxtTotalIncome.setText(rs.getString("totalincome"));
                lblTxtLoanAmount.setText(rs.getString("loanamount"));
                lblTxtTerms.setText(rs.getString("duration"));
                lblTxtInterest.setText(rs.getString("interest"));

    lblTxtTotalInterest.setText(rs.getString("totalinterest"));

    lblTxtInstallment.setText(rs.getString("installment"));

    lblTxtTotalAmount.setText(rs.getString("totalamount"));
                lblTxtRemainToPay.setText(restamount.toString());
                lblTxtPaidTill.setText(rs.getString("paidtill"));
                lblTxtLoanBy.setText(rs.getString("doneby"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
        String sql1="SELECT paidtill,DATE_FORMAT(date, '%d/%m/%y')
As mydate FROM sss.loantaking where acno="+ txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql1);
            while (rs.next()){
                lblTxtDuration.setText(rs.getString("mydate")+" to
"+rs.getString("paidtill"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
}
```



LOAN

Account No.
Loan ID
Name
F / H Name
Account Balance
Address

Loan Detail

Loan Duration date
Loan Taken By
Paid till
Remain to Pay
Purpose of Loan
Granter A/C No.
Income Type
Total Income

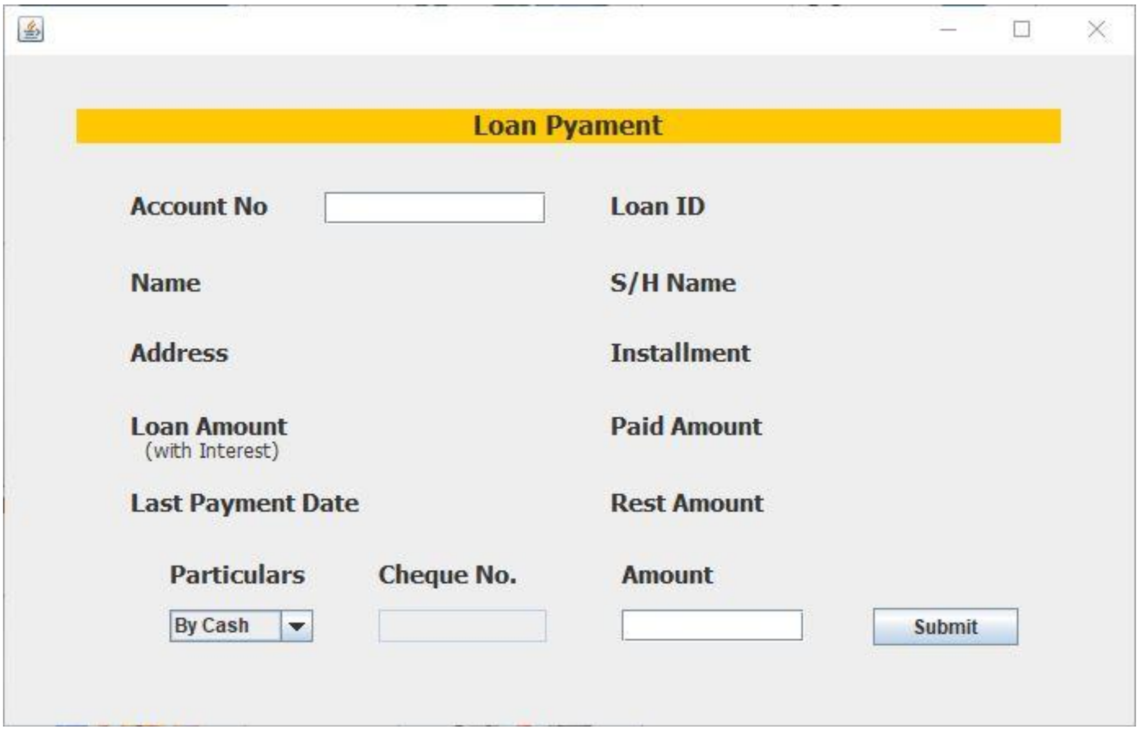
Loan Amount
Terms(months)
Interest(PM)
Total Interest
Installment(PM)
Total Amount

LOAN PAYMENT

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;

import java.awt.Color;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;

import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;


public class loanpayment extends JFrame {

    dbconnection dbc = new dbconnection();
    Integer id,loanacno;
    String uname;
    private JPanel contentPane;
    private JTextField txtAcNo;
    private JTextField txtAmount;
    private JLabel lblTxtName;
    private JLabel lblTxtAddress;
    private JLabel lblTxtLoanAmount;
    private JLabel lblTxtRestAmount;
    private JLabel lblTxtInstallment;
    private JLabel lblTxtGName;
    private JLabel lblTxtLoanId;
    private JLabel lblTxtLastPaymentDate;
    JLabel lblTxtPaidAmount;
    private JComboBox comboParticular;
    int currentbalance,totalamountpaid,loanrestamount=0;
    private JTextField txtCheque;

    public static void main(String[] args) {
      EventQueue.invokeLater(new Runnable() {
        public void run() {
          try {
            loanpayment frame = new loanpayment(null);
            frame.setVisible(true);
          } catch (Exception e) {
            e.printStackTrace();
```

```
              }
           }
        });
    }
    public loanpayment(String username) {
        uname=username;
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(350, 150, 708, 450);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        //auto increment id
        int count =1;
        String sql=("select acno from loanpayment");
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while(rs.next()){
              count+=1;
           }
        } catch (Exception e) { JOptionPane.showMessageDialog(null,
e);}
        id=10+count;

        JLabel lblLoanPyament = new JLabel("Loan Pyament");

    lblLoanPyament.setHorizontalAlignment(SwingConstants.CENTER);
        lblLoanPyament.setOpaque(true);
        lblLoanPyament.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblLoanPyament.setBackground(Color.ORANGE);
        lblLoanPyament.setBounds(44, 33, 603, 21);
        contentPane.add(lblLoanPyament);

        JLabel lblAccountNo = new JLabel("Account No");
        lblAccountNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAccountNo.setBounds(77, 82, 97, 21);
        contentPane.add(lblAccountNo);

        txtAcNo = new JTextField();
        txtAcNo.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtAcNo.setEditable(true);
              } else {
                 txtAcNo.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtAcNo.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcNo.getText());
              if(acno>200000){
                txtAcNo.setEditable(false);
                JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                txtAcNo.setText("");
                txtAcNo.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                detail();
                loandetail();
              }

              String Accountno=txtAcNo.getText(),status="";
              String sql=("SELECT * FROM memberdetail WHERE
acno='"+Accountno+"' "),sql1=("SELECT * FROM loantaking");
              int count=0,count1=0;
              try {
                Statement stmt=dbc.conmethod();
                ResultSet rs=stmt.executeQuery(sql);
                while(rs.next()){
                      status=rs.getString("status");

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                           count=1;
                      }
                 }
                if (count==0 && acno >= 100000 && acno<=199999){
                      txtAcNo.setEditable(false);
                      JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                      txtAcNo.setEditable(true);
                      txtAcNo.requestFocus(true);
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
              try {
                Statement stmt=dbc.conmethod();
                ResultSet rs=stmt.executeQuery(sql1);
                while(rs.next()){

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                           count1=1;
                      }
                 }
                if (count==1 && count1==0 &&  acno >= 100000 &&
acno<=199999){
                      txtAcNo.setEditable(false);
                      JOptionPane.showMessageDialog(null, "You
havn't take any Loan Yet, Enter Valid A/C No.");
                      txtAcNo.setText("");
                      txtAcNo.setEditable(true);
                      txtAcNo.requestFocus(true);
                 }else{
                      if(status.equals("Deactivate")){
                           JOptionPane.showMessageDialog(null, "A/C
Already Deactivated");
                           txtAcNo.setText("");
```

```
                        }
                  }
               } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
            }
        });
        txtAcNo.setBounds(196, 84, 136, 20);
        contentPane.add(txtAcNo);
        txtAcNo.setColumns(10);

        JLabel lblLoanId = new JLabel("Loan ID");
        lblLoanId.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanId.setBounds(371, 82, 76, 21);
        contentPane.add(lblLoanId);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(77, 129, 67, 21);
        contentPane.add(lblName);

        JLabel lblShName = new JLabel("S/H Name");
        lblShName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblShName.setBounds(371, 129, 88, 21);
        contentPane.add(lblShName);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(77, 172, 76, 21);
        contentPane.add(lblAddress);

        JLabel lblInstallment = new JLabel("Installment");
        lblInstallment.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInstallment.setBounds(371, 172, 97, 21);
        contentPane.add(lblInstallment);

        JLabel lblLoanAmount = new JLabel("Loan Amount");
        lblLoanAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanAmount.setBounds(77, 217, 97, 21);
        contentPane.add(lblLoanAmount);

        JLabel lblWithInterest = new JLabel("(with Interest)");
        lblWithInterest.setFont(new Font("Tahoma", Font.PLAIN, 12));
        lblWithInterest.setBounds(86, 232, 88, 21);
        contentPane.add(lblWithInterest);

        JLabel lblRestAmount = new JLabel("Rest Amount");
        lblRestAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRestAmount.setBounds(371, 264, 97, 21);
        contentPane.add(lblRestAmount);

        JLabel lblLastPaymentDate = new JLabel("Last Payment Date");
        lblLastPaymentDate.setFont(new Font("Tahoma", Font.BOLD,
15));
        lblLastPaymentDate.setBounds(77, 264, 140, 21);
        contentPane.add(lblLastPaymentDate);
        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(196, 129, 152, 21);
        contentPane.add(lblTxtName);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(196, 172, 152, 21);
        contentPane.add(lblTxtAddress);

        lblTxtLoanAmount = new JLabel();
        lblTxtLoanAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLoanAmount.setBounds(196, 217, 152, 21);
        contentPane.add(lblTxtLoanAmount);

        lblTxtRestAmount = new JLabel();
        lblTxtRestAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtRestAmount.setBounds(493, 264, 162, 21);
        contentPane.add(lblTxtRestAmount);

        lblTxtInstallment = new JLabel();
        lblTxtInstallment.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtInstallment.setBounds(493, 172, 162, 21);
        contentPane.add(lblTxtInstallment);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(493, 129, 162, 21);
        contentPane.add(lblTxtGName);

        lblTxtLoanId = new JLabel();
        lblTxtLoanId.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtLoanId.setBounds(493, 82, 162, 21);
        contentPane.add(lblTxtLoanId);

        lblTxtLastPaymentDate = new JLabel();
        lblTxtLastPaymentDate.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLastPaymentDate.setBounds(247, 264, 112, 21);
        contentPane.add(lblTxtLastPaymentDate);

        JLabel lblParticulars = new JLabel("Particulars");
        lblParticulars.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblParticulars.setBounds(101, 308, 97, 21);
        contentPane.add(lblParticulars);

        JLabel lblAmount = new JLabel("Amount");
        lblAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAmount.setBounds(378, 308, 76, 21);
        contentPane.add(lblAmount);

        txtAmount = new JTextField();
        txtAmount.setColumns(10);
        txtAmount.setBounds(378, 340, 112, 20);
        contentPane.add(txtAmount);

        comboParticular = new JComboBox();
        comboParticular.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                if(comboParticular.getSelectedItem().equals("By
Cheque")){
                    txtCheque.setEditable(true);
                }
            }
        });
        comboParticular.addItem("By Cash");
        comboParticular.addItem("By Cheque");
        comboParticular.setBounds(101, 340, 88, 20);
        contentPane.add(comboParticular);

        JButton btnSubmit = new JButton("Submit");
        btnSubmit.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                loanpaidamount();
            }
        });
        btnSubmit.setBounds(532, 339, 89, 23);
        contentPane.add(btnSubmit);

        txtCheque = new JTextField();
        txtCheque.setEditable(false);
        txtCheque.setColumns(10);
        txtCheque.setBounds(229, 340, 103, 20);
        contentPane.add(txtCheque);

        JLabel lblChequeNo = new JLabel("Cheque No.");
        lblChequeNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblChequeNo.setBounds(229, 308, 103, 21);
        contentPane.add(lblChequeNo);

        JLabel lblPaidAmount = new JLabel("Paid Amount");
        lblPaidAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPaidAmount.setBounds(371, 217, 97, 21);
        contentPane.add(lblPaidAmount);

        lblTxtPaidAmount = new JLabel();
        lblTxtPaidAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtPaidAmount.setBounds(493, 217, 162, 21);
        contentPane.add(lblTxtPaidAmount);
    }

    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
             lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
             lblTxtGName.setText(rs.getString("gname"));
             lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("city"));

    currentbalance=Integer.parseInt(rs.getString("amount"));
          }
       }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void loandetail(){
       String sql="SELECT * FROM sss.loantaking where acno="+
txtAcNo.getText() +"";
       try {
          Statement stmt=dbc.conmethod();
          ResultSet rs=stmt.executeQuery(sql);
          while (rs.next()){
             Integer totalamount=0,paidamount=0;
             totalamountpaid=rs.getInt("totalamountpaid");
             lblTxtLoanId.setText(rs.getString("id"));
             lblTxtLoanAmount.setText(rs.getString("totalamount"));

    lblTxtInstallment.setText(rs.getString("installment"));
             lblTxtLoanId.setText(rs.getString("id"));

    lblTxtLastPaymentDate.setText(rs.getString("paidtill"));
             loanacno=rs.getInt("id");
             totalamount=rs.getInt("totalamount");
             paidamount=rs.getInt("totalamountpaid");
             loanrestamount=totalamount-paidamount;
             lblTxtPaidAmount.setText(String.valueOf(paidamount));

    lblTxtRestAmount.setText(String.valueOf(loanrestamount));
          }
       }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }

    private void loanpaidamount(){
       String accountno,particular,cheque,sql1,sql2;
       int amount=0;
       accountno=txtAcNo.getText();
       particular=comboParticular.getSelectedItem().toString();
       cheque=txtCheque.getText();
       amount=Integer.parseInt(txtAmount.getText());
       totalamountpaid=totalamountpaid+amount;
       sql1="UPDATE sss.loantaking SET
totalamountpaid='"+totalamountpaid+"' WHERE acno='"+accountno+"' ";
       sql2="INSERT INTO
sss.loanpayment(id,acno,loanacno,particular,cheque,amount,date,time,
amountpaid,doneby)
values('"+id+"','"+accountno+"','"+loanacno+"','"+particular+"','"+c
heque+"','"+amount+"',now(),now(),'"+totalamountpaid+"','"+uname+"')
";
       if (txtAmount.equals("") && accountno.equals("")){
          JOptionPane.showMessageDialog(null, "Invalid Data");
       }else
       {
          try {
          Statement stmt=dbc.conmethod();
          if(Integer.valueOf(txtAmount.getText())<=
loanrestamount){
             stmt.execute(sql1);
             stmt.execute(sql2);
             JOptionPane.showMessageDialog(null, "Amount paid");
             dispose();
          }else{
             JOptionPane.showMessageDialog(null, "Amount Exceed to
Loan Remaining Amount");
             txtAmount.setText(String.valueOf(loanrestamount));
          }
       }
       catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
       }
    }
}
```



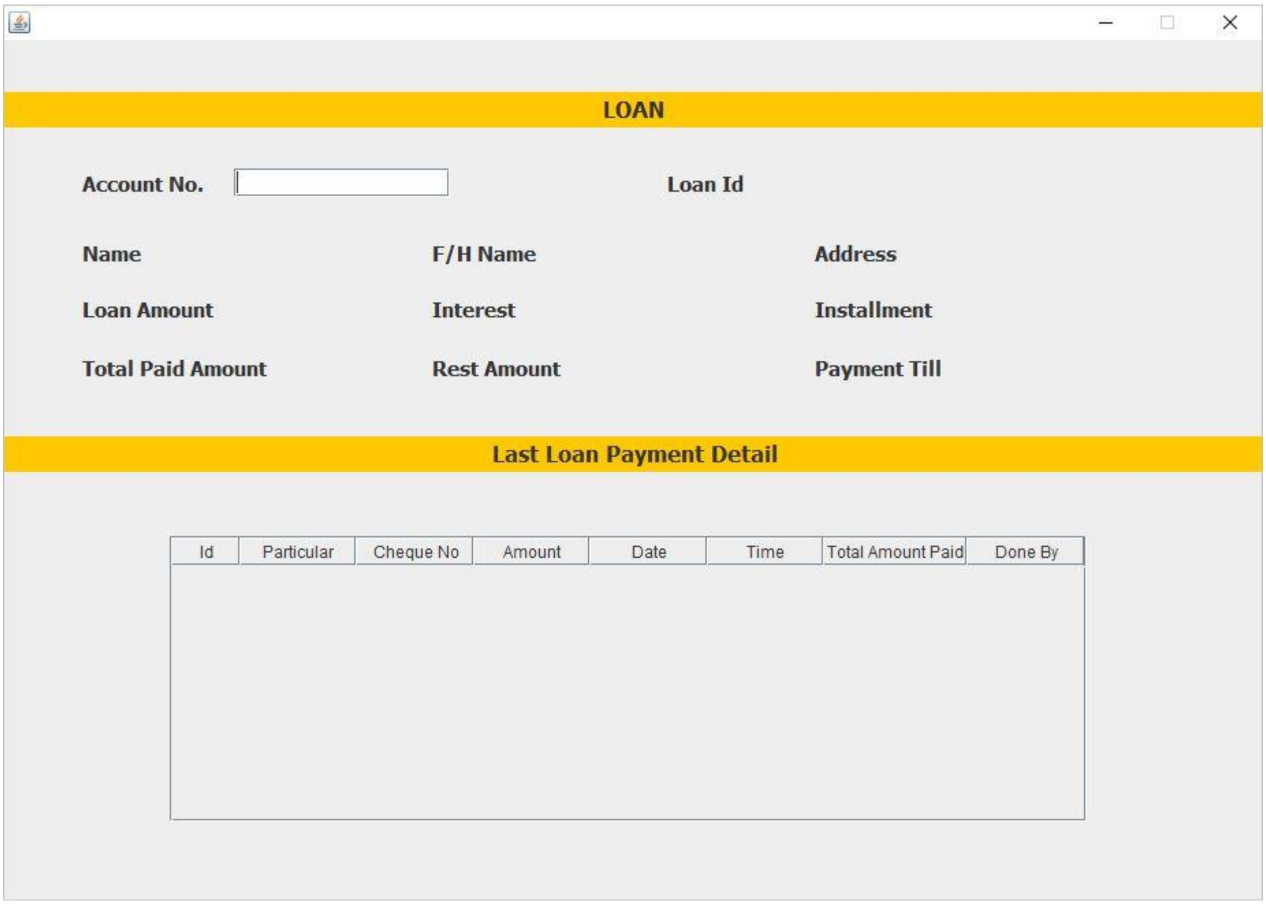
LOAN PAYMENT STATEMENT

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;

import java.awt.Color;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.JTable;

import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.Frame;


public class loanpaymentdetail extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtAcNo;
    private JLabel lblTxtName;
    private JLabel lblTxtLoanAmount;
    private JLabel lblTxtGName;
    private JLabel lblTxtInstallment;
    private JLabel lblTxtPaymentTill;
    private JLabel lblTxtAddress;
    private JLabel lblTxtTotalPaidAmount;
    private JLabel lblTxtRestAmount;
    private JLabel lblTxtLoanId;
    private JLabel lblLastLoanPayment;
    private JLabel lblTxtInterest;
    private JTable table;
    DefaultTableModel model = new DefaultTableModel();
    Object[] row = new Object[8];

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    loanpaymentdetail frame = new loanpaymentdetail();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public loanpaymentdetail() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 30, 926, 658);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("LOAN");
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setOpaque(true);
        lblNewLabel.setBounds(0, 38, 920, 26);
        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Account No.");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblNewLabel_1.setBounds(57, 96, 89, 19);
        contentPane.add(lblNewLabel_1);

        JLabel lblName = new JLabel("Name");
        lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblName.setBounds(57, 147, 65, 19);
        contentPane.add(lblName);

        JLabel lblLoanAmount = new JLabel("Loan Amount");
        lblLoanAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanAmount.setBounds(57, 188, 102, 19);
        contentPane.add(lblLoanAmount);

        JLabel lblRestAmount = new JLabel("Rest Amount");
        lblRestAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRestAmount.setBounds(313, 231, 102, 19);
        contentPane.add(lblRestAmount);

        JLabel lblInstallment = new JLabel("Installment");
        lblInstallment.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInstallment.setBounds(593, 188, 89, 19);
        contentPane.add(lblInstallment);

        JLabel lblFhName = new JLabel("F/H Name");
        lblFhName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblFhName.setBounds(313, 147, 89, 19);
        contentPane.add(lblFhName);

        JLabel lblLoanId = new JLabel("Loan Id");
        lblLoanId.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanId.setBounds(485, 96, 89, 19);
        contentPane.add(lblLoanId);

        JLabel lblTotalPaidAmount = new JLabel("Total Paid Amount");
```

```
        lblTotalPaidAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalPaidAmount.setBounds(57, 231, 147, 19);
        contentPane.add(lblTotalPaidAmount);

        JLabel lblPaymentTill = new JLabel("Payment Till");
        lblPaymentTill.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPaymentTill.setBounds(593, 231, 102, 19);
        contentPane.add(lblPaymentTill);

        JLabel lblAddress = new JLabel("Address");
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(593, 147, 73, 19);
        contentPane.add(lblAddress);

        txtAcNo = new JTextField();
        txtAcNo.addKeyListener(new KeyAdapter() {
            @Override
            public void keyReleased(KeyEvent e) {
               if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                  txtAcNo.setEditable(true);
               } else {
                  txtAcNo.setEditable(false);
                  Toolkit.getDefaultToolkit().beep();
                  JOptionPane.showMessageDialog(null, "* Enter only numeric
digits(0-9) ");
                  txtAcNo.setEditable(true);
               }
               Integer acno=Integer.valueOf(txtAcNo.getText());
               if(acno>200000){
                  txtAcNo.setEditable(false);
                  reset();
                  JOptionPane.showMessageDialog(null, "Please Enter valid
Account No");
                  txtAcNo.setText("");
                  txtAcNo.setEditable(true);
               }
               if(acno >= 100000 && acno<=199999){
                  detail();
                  loandetail();
                  model.getDataVector().removeAllElements();
                  paymentdetail();
               }
               String sql=("SELECT * FROM memberdetail"),sql1=("SELECT *
FROM loantaking");
               int count=0,count1=0;
               try {
                  Statement stmt=dbc.conmethod();
                  ResultSet rs=stmt.executeQuery(sql);
                  while(rs.next()){
                        if(txtAcNo.getText().equals(rs.getString("acno"))){
                              count=1;
                        }
                  }
                  if (count==0 && acno >= 100000 && acno<=199999){
                        txtAcNo.setEditable(false);
                        reset();
                        model.getDataVector().removeAllElements();
                        JOptionPane.showMessageDialog(null, "A/C Not Found,
Enter Valid A/C No.");
                        txtAcNo.setText("");
                        txtAcNo.setEditable(true);
                        txtAcNo.requestFocus(true);
                  }
               } catch (Exception ex){ JOptionPane.showMessageDialog(null,
ex);}
               try {
                  Statement stmt=dbc.conmethod();
                  ResultSet rs=stmt.executeQuery(sql1);
                  while(rs.next()){
                        if(txtAcNo.getText().equals(rs.getString("acno"))){
                              count1=1;
                        }
                  }
                  if (count==1 && count1==0 && acno >= 100000 &&
acno<=199999){
                        txtAcNo.setEditable(false);
                        reset();
                        model.getDataVector().removeAllElements();
                        JOptionPane.showMessageDialog(null, "You havn't
take any Loan Yet, Enter Valid A/C No.");
                        txtAcNo.setText("");
                        txtAcNo.setEditable(true);
                        txtAcNo.requestFocus(true);
                  }
               } catch (Exception ex){ JOptionPane.showMessageDialog(null,
ex);}
            }
        });
        txtAcNo.setBounds(168, 94, 158, 21);
        contentPane.add(txtAcNo);
        txtAcNo.setColumns(10);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtName.setBounds(121, 147, 182, 19);
        contentPane.add(lblTxtName);

        lblTxtLoanAmount = new JLabel();
        lblTxtLoanAmount.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtLoanAmount.setBounds(169, 188, 102, 19);
        contentPane.add(lblTxtLoanAmount);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtGName.setBounds(399, 147, 188, 19);
        contentPane.add(lblTxtGName);

        lblTxtInstallment = new JLabel();
        lblTxtInstallment.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtInstallment.setBounds(705, 188, 89, 19);
        contentPane.add(lblTxtInstallment);

        lblTxtPaymentTill = new JLabel();
        lblTxtPaymentTill.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtPaymentTill.setBounds(705, 231, 102, 19);
        contentPane.add(lblTxtPaymentTill);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtAddress.setBounds(663, 147, 259, 19);
        contentPane.add(lblTxtAddress);
        lblTxtTotalPaidAmount = new JLabel();
        lblTxtTotalPaidAmount.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtTotalPaidAmount.setBounds(200, 231, 109, 19);
        contentPane.add(lblTxtTotalPaidAmount);

        lblTxtRestAmount = new JLabel();
        lblTxtRestAmount.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtRestAmount.setBounds(425, 231, 102, 19);
        contentPane.add(lblTxtRestAmount);

        lblTxtLoanId = new JLabel();
        lblTxtLoanId.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtLoanId.setBounds(584, 96, 89, 19);
        contentPane.add(lblTxtLoanId);

        lblLastLoanPayment = new JLabel();
        lblLastLoanPayment.setText("Last Loan Payment Detail");
        lblLastLoanPayment.setOpaque(true);
        lblLastLoanPayment.setHorizontalAlignment(SwingConstants.CENTER);
        lblLastLoanPayment.setFont(new Font("Tahoma", Font.BOLD, 16));
        lblLastLoanPayment.setBackground(Color.ORANGE);
        lblLastLoanPayment.setBounds(0, 290, 922, 26);
        contentPane.add(lblLastLoanPayment);

        table = new JTable();
        table.setBounds(143, 363, 649, 248);
        contentPane.add(table);
        Object[] columns = {"Id","Particular","Cheque
No","Amount","Date","Time","Total Amount Paid","Done By"};
        table.setModel(model);
        model.setColumnIdentifiers(columns);
        table.setRowHeight(30);
        JScrollPane pane = new JScrollPane(table);
        pane.setBounds(121, 363, 671, 209);
        contentPane.setLayout(null);
        contentPane.add(pane);
        table.getColumnModel().getColumn(0).setPreferredWidth(15);
        table.getColumnModel().getColumn(1).setPreferredWidth(50);
        table.getColumnModel().getColumn(2).setPreferredWidth(50);
        table.getColumnModel().getColumn(3).setPreferredWidth(50);
        table.getColumnModel().getColumn(4).setPreferredWidth(50);
        table.getColumnModel().getColumn(5).setPreferredWidth(50);
        table.getColumnModel().getColumn(6).setPreferredWidth(70);
        table.getColumnModel().getColumn(7).setPreferredWidth(50);
        table.setAutoResizeMode(JTable.AUTO_RESIZE_LAST_COLUMN);

        JLabel lblInterest = new JLabel("Interest");
        lblInterest.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInterest.setBounds(313, 188, 89, 19);
        contentPane.add(lblInterest);

        lblTxtInterest = new JLabel();
        lblTxtInterest.setFont(new Font("Dialog", Font.PLAIN, 15));
        lblTxtInterest.setBounds(412, 188, 102, 19);
        contentPane.add(lblTxtInterest);
    }

    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcNo.getText() +"";
```

```
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
               lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
               lblTxtGName.setText(rs.getString("gname"));
               lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("city"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null, ex);}
    }
    private void loandetail(){
        String sql="SELECT * FROM sss.loantaking where acno="+
txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
               Integer totalamount,paidamount,restamount;
               lblTxtLoanId.setText(rs.getString("id"));
               lblTxtLoanAmount.setText(rs.getString("loanamount"));
               lblTxtInstallment.setText(rs.getString("installment"));
               lblTxtInterest.setText(rs.getString("totalinterest"));
               lblTxtPaymentTill.setText(rs.getString("paidtill"));

    lblTxtTotalPaidAmount.setText(rs.getString("totalamountpaid"));
               totalamount=rs.getInt("totalamount");
               paidamount=rs.getInt("totalamountpaid");
               restamount=totalamount-paidamount;
               lblTxtRestAmount.setText(String.valueOf(restamount));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null, ex);}
    }

    private void paymentdetail(){
        String id=null, particular=null, cheque=null, amount=null,
date=null, time=null, totalamountpaid=null, doneby=null;
        String sql1="SELECT * FROM sss.loanpayment where acno='"+
txtAcNo.getText() +"' ORDER BY date DESC";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql1);
            while (rs.next()){
               id = rs.getString("id");
               particular = rs.getString("particular");
               cheque = rs.getString("cheque");
               amount = rs.getString("amount");
               date = rs.getString("date");
               time = rs.getString("time");
               totalamountpaid = rs.getString("amountpaid");
               doneby = rs.getString("doneby");

               row[0]= id;
               row[1]= particular;
               row[2]= cheque;
               row[3]= amount;
               row[4]= date;
               row[5]= time;
               row[6]= totalamountpaid;
               row[7]= doneby;
```

```
                model.addRow(row);
            }

        }catch(Exception ex){JOptionPane.showMessageDialog(null, ex);}
    }
    private void reset(){
        lblTxtLoanId.setText("");
        lblTxtName.setText("");
        lblTxtGName.setText("");
        lblTxtAddress.setText("");
        lblTxtLoanAmount.setText("");
        lblTxtInstallment.setText("");
        lblTxtInterest.setText("");
        lblTxtPaymentTill.setText("");
        lblTxtTotalPaidAmount.setText("");
        lblTxtRestAmount.setText("");
    }
}
```



LOAN

Account No. Loan Id

Name F/H Name Address

Loan Amount Interest Installment

Total Paid Amount Rest Amount Payment Till

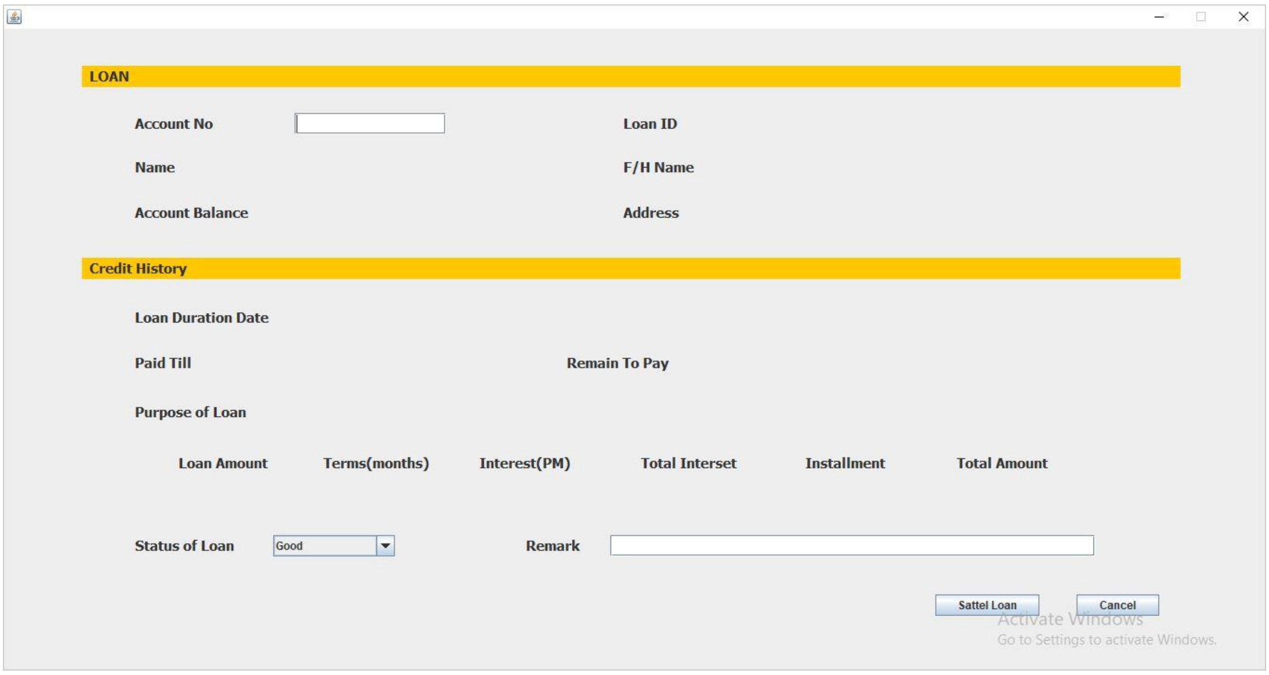
Last Loan Payment Detail

Id	Particular	Cheque No	Amount	Date	Time	Total Amount Paid	Done By

LOAN SETTLEMENT

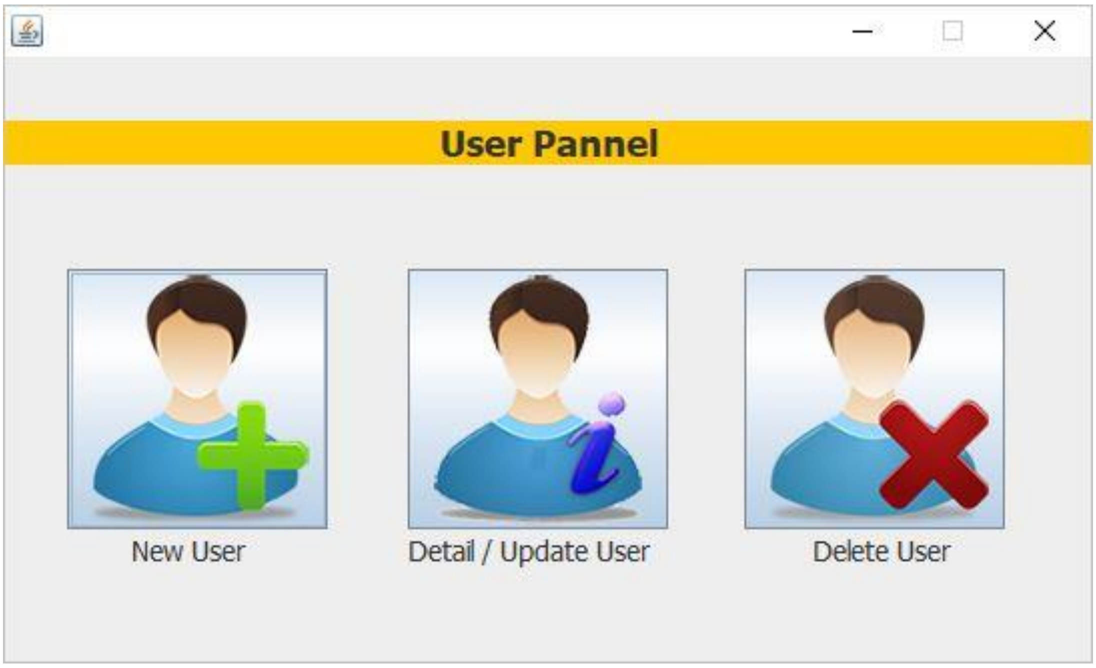
```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Frame;
import java.awt.Toolkit;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;

import java.awt.Font;
import java.awt.Color;

import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;


public class loansattle extends JFrame {

    dbconnection dbc = new dbconnection();
    Integer flag=0,restamount;
    private JPanel contentPane;
    private JTextField txtAcNo;
    private JTextField txtRemark;
    private JLabel lblTxtName;
    private JLabel lblTxtAcBalance;
    private JLabel lblTxtGName;
    private JLabel lblTxtAddress;
    private JLabel lblTxtLoanId;
    private JLabel lblTxtPaidTill;
    private JLabel lblTxtPurposeOfLoan;
    private JLabel lblTxtRemainToPay;
    private JLabel lblTxtTotalAmount;
    private JLabel lblTxtInstallment;
    private JLabel lblTxtTotalInterest;
    private JLabel lblTxtInterest;
    private JLabel lblTxtTerms;
    private JLabel lblTxtLoanAmount;
    private JLabel lblTxtLoanDuration;
    private JComboBox comboStatus;
    public static void main(String[] args) {
       EventQueue.invokeLater(new Runnable() {
          public void run() {
             try {
                loansattle frame = new loansattle();
                frame.setVisible(true);
             } catch (Exception e) {
                e.printStackTrace();
             }
          }
       });
    }

    // Create the frame.
    public loansattle() {
       setExtendedState(Frame.MAXIMIZED_BOTH);
       setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
       setResizable(false);
       setSize(new Dimension(1364,720));
       contentPane = new JPanel();
       contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
       setContentPane(contentPane);
       contentPane.setLayout(null);

       JLabel lblNewLabel = new JLabel("  LOAN");
       lblNewLabel.setBackground(Color.ORANGE);
       lblNewLabel.setOpaque(true);
       lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblNewLabel.setBounds(84, 40, 1183, 23);
       contentPane.add(lblNewLabel);

       JLabel lblCreditHistory = new JLabel("  Credit History");
       lblCreditHistory.setOpaque(true);
       lblCreditHistory.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblCreditHistory.setBackground(Color.ORANGE);
       lblCreditHistory.setBounds(84, 247, 1183, 23);
       contentPane.add(lblCreditHistory);

       JLabel lblNewLabel_1 = new JLabel("Account No");
       lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblNewLabel_1.setBounds(141, 91, 102, 23);
       contentPane.add(lblNewLabel_1);

       JLabel lblName = new JLabel("Name");
       lblName.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblName.setBounds(141, 138, 102, 23);
       contentPane.add(lblName);

       JLabel lblAccountBalance = new JLabel("Account Balance");
       lblAccountBalance.setFont(new Font("Tahoma", Font.BOLD,
15));
       lblAccountBalance.setBounds(141, 187, 131, 23);
       contentPane.add(lblAccountBalance);

       JLabel lblAddress = new JLabel("Address");
```

```
        lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAddress.setBounds(667, 187, 86, 23);
        contentPane.add(lblAddress);

        JLabel lblFhName = new JLabel("F/H Name");
        lblFhName.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblFhName.setBounds(667, 138, 86, 23);
        contentPane.add(lblFhName);

        txtAcNo = new JTextField();
        txtAcNo.addKeyListener(new KeyAdapter() {
           @Override
           public void keyReleased(KeyEvent e) {
              if (e.getKeyChar() >= '0' && e.getKeyChar() <= '9' ||
e.getKeyChar()==8 || e.getKeyChar()==10) {
                 txtAcNo.setEditable(true);
              } else {
                 txtAcNo.setEditable(false);
                 Toolkit.getDefaultToolkit().beep();
                 JOptionPane.showMessageDialog(null, "* Enter only
numeric digits(0-9) ");
                 txtAcNo.setEditable(true);
              }
              Integer acno=Integer.valueOf(txtAcNo.getText());
              if(acno>200000){
                 txtAcNo.setEditable(false);
                 JOptionPane.showMessageDialog(null, "Please Enter
valid Account No");
                 txtAcNo.setText("");
                 txtAcNo.setEditable(true);
              }
              if(acno >= 100000 && acno<=199999){
                 detail();
                 loandetail();
              }

              String Accountno=txtAcNo.getText(),status="";
              String sql=("SELECT * FROM memberdetail WHERE
acno='"+Accountno+"' "),sql1=("SELECT * FROM loantaking");
              int count=0,count1=0;
              try {
                 Statement stmt=dbc.conmethod();
                 ResultSet rs=stmt.executeQuery(sql);
                 while(rs.next()){
                       status=rs.getString("status");

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                           count=1;
                       }
                 }
                 if (count==0 && acno >= 100000 && acno<=199999){
                       txtAcNo.setEditable(false);
                       JOptionPane.showMessageDialog(null, "A/C Not
Found, Enter Valid A/C No.");
                       txtAcNo.setEditable(true);
                       txtAcNo.requestFocus(true);
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
              try {
                 Statement stmt=dbc.conmethod();
                 ResultSet rs=stmt.executeQuery(sql1);
                 while(rs.next()){

    if(txtAcNo.getText().equals(rs.getString("acno"))){
                            count1=1;
                       }
                 }
                 if (count==1 && count1==0 &&  acno >= 100000 &&
acno<=199999){
                      txtAcNo.setEditable(false);
                      JOptionPane.showMessageDialog(null, "You
havn't take any Loan Yet, Enter Valid A/C No.");
                      txtAcNo.setText("");
                      txtAcNo.setEditable(true);
                      txtAcNo.requestFocus(true);
                 }else{
                      if(status.equals("Deactivate")){
                            JOptionPane.showMessageDialog(null, "A/C
Already Deactivated");
                            txtAcNo.setText("");
                       }
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
            }
        });
        txtAcNo.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtAcNo.setBounds(313, 91, 163, 23);
        contentPane.add(txtAcNo);
        txtAcNo.setColumns(10);

        lblTxtName = new JLabel();
        lblTxtName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtName.setBounds(313, 138, 262, 23);
        contentPane.add(lblTxtName);

        lblTxtAcBalance = new JLabel();
        lblTxtAcBalance.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAcBalance.setBounds(313, 187, 131, 23);
        contentPane.add(lblTxtAcBalance);

        lblTxtGName = new JLabel();
        lblTxtGName.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtGName.setBounds(798, 138, 312, 23);
        contentPane.add(lblTxtGName);

        lblTxtAddress = new JLabel();
        lblTxtAddress.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtAddress.setBounds(798, 187, 377, 23);
        contentPane.add(lblTxtAddress);
        JLabel lblLoanDurationDate = new JLabel("Loan Duration 
Date");
        lblLoanDurationDate.setFont(new Font("Tahoma", Font.BOLD, 
15));
        lblLoanDurationDate.setBounds(141, 300, 154, 23);
        contentPane.add(lblLoanDurationDate);

        JLabel lblPaidTill = new JLabel("Paid Till");
        lblPaidTill.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPaidTill.setBounds(141, 349, 86, 23);
        contentPane.add(lblPaidTill);

        JLabel lblPurposeOfLoan = new JLabel("Purpose of Loan");
        lblPurposeOfLoan.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblPurposeOfLoan.setBounds(141, 402, 140, 23);
        contentPane.add(lblPurposeOfLoan);

        lblTxtLoanDuration = new JLabel();
        lblTxtLoanDuration.setFont(new Font("Tahoma", Font.PLAIN, 
15));
        lblTxtLoanDuration.setBounds(313, 300, 411, 23);
        contentPane.add(lblTxtLoanDuration);

        lblTxtPaidTill = new JLabel();
        lblTxtPaidTill.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtPaidTill.setBounds(313, 349, 219, 23);
        contentPane.add(lblTxtPaidTill);

        lblTxtPurposeOfLoan = new JLabel();
        lblTxtPurposeOfLoan.setFont(new Font("Tahoma", Font.PLAIN, 
15));
        lblTxtPurposeOfLoan.setBounds(313, 402, 797, 23);
        contentPane.add(lblTxtPurposeOfLoan);

        JLabel lblRemainToPay = new JLabel("Remain To Pay");
        lblRemainToPay.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRemainToPay.setBounds(606, 349, 140, 23);
        contentPane.add(lblRemainToPay);

        lblTxtRemainToPay = new JLabel();
        lblTxtRemainToPay.setFont(new Font("Tahoma", Font.PLAIN, 
15));
        lblTxtRemainToPay.setBounds(756, 349, 219, 23);
        contentPane.add(lblTxtRemainToPay);

        JLabel lblLoanAmount = new JLabel("Loan Amount");
        lblLoanAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanAmount.setBounds(188, 457, 112, 23);
        contentPane.add(lblLoanAmount);

        JLabel lblTermsmonths = new JLabel("Terms(months)");
        lblTermsmonths.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTermsmonths.setBounds(344, 457, 123, 23);
        contentPane.add(lblTermsmonths);

        JLabel lblInterestpm = new JLabel("Interest(PM)");
        lblInterestpm.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblInterestpm.setBounds(512, 457, 112, 23);
        contentPane.add(lblInterestpm);

        JLabel lblTotalInterest = new JLabel("Installment");
        lblTotalInterest.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalInterest.setBounds(863, 457, 112, 23);
        contentPane.add(lblTotalInterest);

        JLabel lblTotalAmount = new JLabel("Total Amount");
        lblTotalAmount.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalAmount.setBounds(1026, 457, 112, 23);
        contentPane.add(lblTotalAmount);

        JLabel lblTotalInterset = new JLabel("Total Interset");
        lblTotalInterset.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblTotalInterset.setBounds(686, 457, 112, 23);
        contentPane.add(lblTotalInterset);

        lblTxtTotalAmount = new JLabel();
        lblTxtTotalAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalAmount.setBounds(1026, 486, 112, 23);
        contentPane.add(lblTxtTotalAmount);

        lblTxtInstallment = new JLabel();
        lblTxtInstallment.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtInstallment.setBounds(863, 486, 112, 23);
        contentPane.add(lblTxtInstallment);

        lblTxtTotalInterest = new JLabel();
        lblTxtTotalInterest.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtTotalInterest.setBounds(686, 486, 112, 23);
        contentPane.add(lblTxtTotalInterest);

        lblTxtInterest = new JLabel();
        lblTxtInterest.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtInterest.setBounds(512, 486, 112, 23);
        contentPane.add(lblTxtInterest);

        lblTxtTerms = new JLabel();
        lblTxtTerms.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtTerms.setBounds(344, 486, 123, 23);
        contentPane.add(lblTxtTerms);

        lblTxtLoanAmount = new JLabel();
        lblTxtLoanAmount.setFont(new Font("Tahoma", Font.PLAIN,
15));
        lblTxtLoanAmount.setBounds(188, 486, 112, 23);
        contentPane.add(lblTxtLoanAmount);

        JLabel lblStatusOfLoan = new JLabel("Status of Loan");
        lblStatusOfLoan.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblStatusOfLoan.setBounds(141, 546, 112, 23);
        contentPane.add(lblStatusOfLoan);

        comboStatus = new JComboBox();
        comboStatus.addItem("Good");
        comboStatus.addItem("On Timed");
        comboStatus.addItem("Satisfied");
        comboStatus.addItem("Bad");
        comboStatus.setBounds(290, 546, 131, 23);
        contentPane.add(comboStatus);

        JLabel lblRemark = new JLabel("Remark");
        lblRemark.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblRemark.setBounds(562, 546, 80, 23);
        contentPane.add(lblRemark);

        txtRemark = new JTextField();
        txtRemark.setFont(new Font("Tahoma", Font.PLAIN, 15));
        txtRemark.setColumns(10);
        txtRemark.setBounds(653, 546, 522, 23);
        contentPane.add(txtRemark);

        JButton btnCancel = new JButton("Cancel");
        btnCancel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnCancel.setBounds(1155, 610, 89, 23);
        contentPane.add(btnCancel);

        JButton btnSattelLoan = new JButton("Sattel Loan");
        btnSattelLoan.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              if(restamount==0){
                 loansattled();
              }else{
                 JOptionPane.showMessageDialog(null, "Please first
pay All Amount");
              }
           }
        });
        btnSattelLoan.setBounds(1003, 610, 112, 23);
        contentPane.add(btnSattelLoan);

        JLabel lblLoanId = new JLabel("Loan ID");
        lblLoanId.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblLoanId.setBounds(667, 91, 86, 23);
        contentPane.add(lblLoanId);

        lblTxtLoanId = new JLabel();
        lblTxtLoanId.setFont(new Font("Tahoma", Font.PLAIN, 15));
        lblTxtLoanId.setBounds(798, 91, 312, 23);
        contentPane.add(lblTxtLoanId);
    }
    private void detail(){
        String sql="SELECT * FROM sss.memberdetail where acno="+
txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
               lblTxtName.setText(rs.getString("fname")+"
"+rs.getString("mname")+" "+rs.getString("lname"));
               lblTxtGName.setText(rs.getString("gname"));
               lblTxtAddress.setText(rs.getString("line1")+",
"+rs.getString("line2")+", "+rs.getString("city")+",
"+rs.getString("distric")+", "+rs.getString("state"));
               lblTxtAcBalance.setText(rs.getString("amount"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void loandetail(){
        String sql1="SELECT * FROM sss.loantaking where acno="+
txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql1);
            while (rs.next()){
               restamount=rs.getInt("totalamount")-
rs.getInt("totalamountpaid");
               lblTxtLoanId.setText(rs.getString("id"));

    lblTxtPurposeOfLoan.setText(rs.getString("purposeofloan"));
               lblTxtLoanAmount.setText(rs.getString("loanamount"));
               lblTxtTerms.setText(rs.getString("duration"));
               lblTxtInterest.setText(rs.getString("interest"));

    lblTxtTotalInterest.setText(rs.getString("totalinterest"));

    lblTxtInstallment.setText(rs.getString("installment"));

    lblTxtTotalAmount.setText(rs.getString("totalamount"));
               lblTxtRemainToPay.setText(restamount.toString());
               lblTxtPaidTill.setText(rs.getString("paidtill"));
               flag=rs.getInt("flag");
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
        String sql2="SELECT paidtill,DATE_FORMAT(date, '%d/%m/%y')
As mydate FROM sss.loantaking where acno="+ txtAcNo.getText() +"";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql2);
            while (rs.next()){
               lblTxtLoanDuration.setText(rs.getString("mydate")+" to
"+rs.getString("paidtill"));
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void loansattled(){
        String accountno,status,remark,sql;
        accountno=txtAcNo.getText();
        status=comboStatus.getSelectedItem().toString();
        remark=txtRemark.getText();
        sql="UPDATE sss.loantaking SET status='"+status+"',
remark='"+remark+"' , satelmentdate=now()  , satelmenttime=now(),
flag='1' WHERE acno='"+accountno+"' ";
        if (remark.equals("") || accountno.equals("")){
            JOptionPane.showMessageDialog(null, "Please fill the
required data");
        }else
        {
            try {
            Statement stmt=dbc.conmethod();
            if(flag==1){
                JOptionPane.showMessageDialog(null, "You have Already
Satteled the Loan");
            }else{
                stmt.execute(sql);
                JOptionPane.showMessageDialog(null, "Loan Satteled");
                dispose();
            }
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
        }
    }
}
```



ADMIN PANNEL

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;

import java.awt.Font;

import javax.swing.SwingConstants;

import java.awt.Color;

import javax.swing.JButton;

import java.awt.Frame;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Cursor;


public class adminpannel extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    adminpannel frame = new adminpannel();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public adminpannel() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 548, 331);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("User Pannel");
        lblNewLabel.setBackground(Color.ORANGE);
        lblNewLabel.setOpaque(true);
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
        lblNewLabel.setBounds(0, 32, 542, 22);
        contentPane.add(lblNewLabel);

        JButton btnNewButton = new JButton(new
ImageIcon("image/newuser.png"));

    btnNewButton.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CU
RSOR));
        btnNewButton.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new signup().setVisible(true);
           }
        });
        btnNewButton.setBounds(31, 106, 130, 130);
        contentPane.add(btnNewButton);

        JButton btnU = new JButton(new
ImageIcon("image/userinfo.png"));

    btnU.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        btnU.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new userdetail().setVisible(true);
           }
        });
        btnU.setBounds(201, 106, 130, 130);
        contentPane.add(btnU);

        JButton btnUserDelete = new JButton(new
ImageIcon("image/userdelete.png"));

    btnUserDelete.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_C
URSOR));
        btnUserDelete.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new userdelete().setVisible(true);
           }
        });
        btnUserDelete.setBounds(369, 106, 130, 130);
        contentPane.add(btnUserDelete);

        JLabel lblNewLabel_1 = new JLabel("New User");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblNewLabel_1.setBounds(63, 238, 63, 17);
        contentPane.add(lblNewLabel_1);

        JLabel lblDetailUpdate = new JLabel("Detail / Update User");
        lblDetailUpdate.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblDetailUpdate.setBounds(201, 238, 130, 17);
        contentPane.add(lblDetailUpdate);

        JLabel lblDeleteUser = new JLabel("Delete User");
        lblDeleteUser.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblDeleteUser.setBounds(403, 238, 77, 17);
        contentPane.add(lblDeleteUser);
    }
}
```



NEW USER

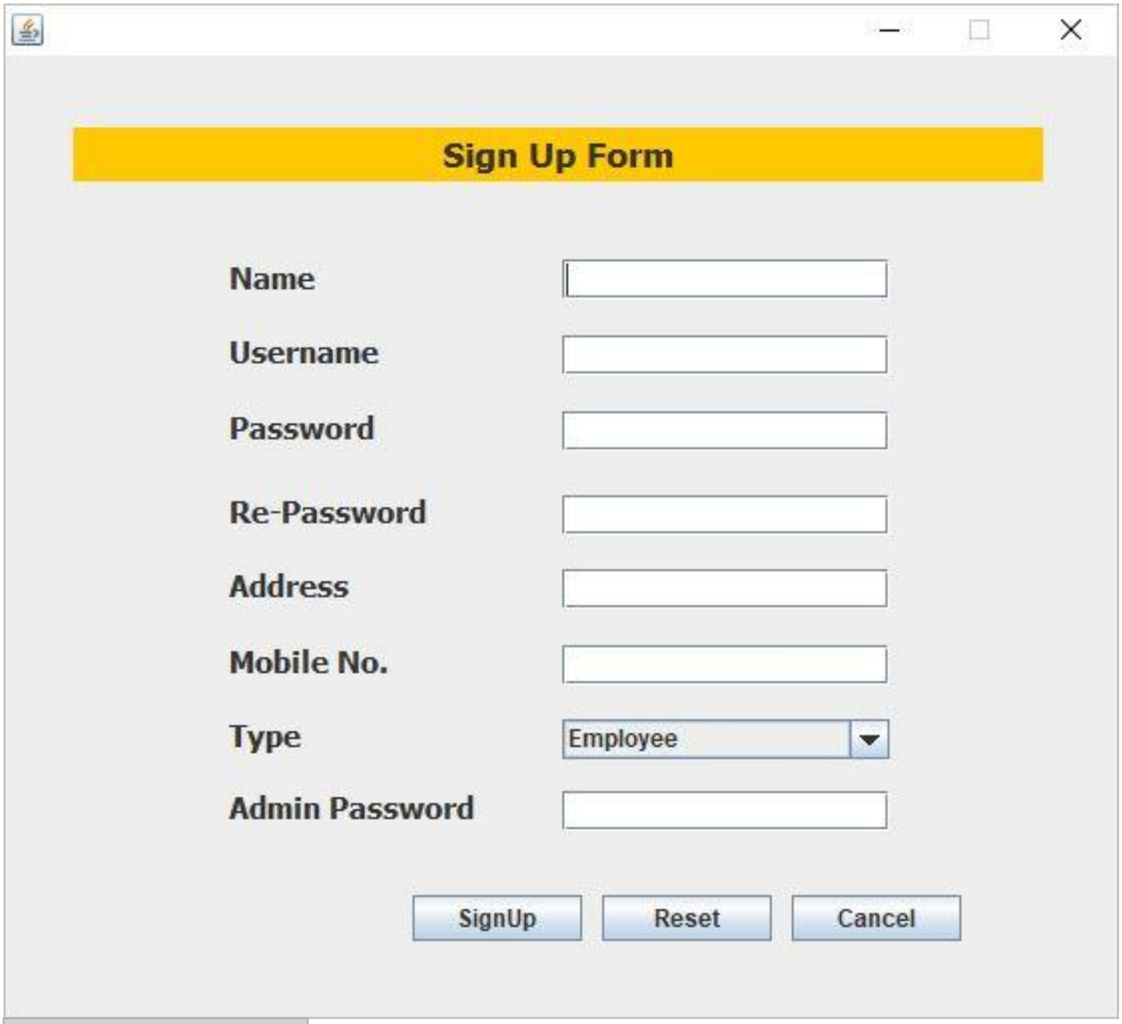
```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import java.awt.Frame;

import javax.swing.JLabel;
import javax.swing.JPasswordField;

import java.awt.Font;
import java.awt.Color;

import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;


public class signup extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtName;
    private JTextField txtUserName;
    private JTextField txtPassword;
    private JTextField txtRePassword;
    private JTextField txtAdress;
    private JTextField txtAdminPassword;
    private JComboBox comboType;
    private JTextField txtMobileNo;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    signup frame = new signup();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public signup() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 563, 510);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblSignUpForm = new JLabel("Sign Up Form");
        lblSignUpForm.setHorizontalAlignment(SwingConstants.CENTER);
        lblSignUpForm.setBackground(Color.ORANGE);
        lblSignUpForm.setFont(new Font("Tahoma", Font.BOLD, 17));
        lblSignUpForm.setOpaque(true);
        lblSignUpForm.setBounds(34, 36, 486, 27);
```

```
contentPane.add(lblSignUpForm);

JLabel lblNewLabel = new JLabel("Name");
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 15));
lblNewLabel.setBounds(112, 102, 54, 19);
contentPane.add(lblNewLabel);

JLabel lblUsername = new JLabel("Username");
lblUsername.setFont(new Font("Tahoma", Font.BOLD, 15));
lblUsername.setBounds(112, 139, 75, 19);
contentPane.add(lblUsername);

JLabel lblPassword = new JLabel("Password");
lblPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
lblPassword.setBounds(112, 177, 75, 19);
contentPane.add(lblPassword);

JLabel lblRepassword = new JLabel("Re-Password");
lblRepassword.setFont(new Font("Tahoma", Font.BOLD, 15));
lblRepassword.setBounds(112, 219, 99, 19);
contentPane.add(lblRepassword);

JLabel lblAddress = new JLabel("Address");
lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
lblAddress.setBounds(112, 256, 85, 19);
contentPane.add(lblAddress);

JLabel lblType = new JLabel("Type");
lblType.setFont(new Font("Tahoma", Font.BOLD, 15));
lblType.setBounds(112, 331, 54, 19);
contentPane.add(lblType);

JLabel lblAdminPassword = new JLabel("Admin Password");
lblAdminPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
lblAdminPassword.setBounds(112, 367, 123, 19);
contentPane.add(lblAdminPassword);

txtName = new JTextField();
txtName.setBounds(279, 102, 164, 20);
contentPane.add(txtName);
txtName.setColumns(10);

txtUserName = new JTextField();
txtUserName.setColumns(10);
txtUserName.setBounds(279, 140, 164, 20);
contentPane.add(txtUserName);

txtPassword = new JPasswordField();
txtPassword.setColumns(10);
txtPassword.setBounds(279, 178, 164, 20);
contentPane.add(txtPassword);

txtRePassword = new JPasswordField();
txtRePassword.setColumns(10);
txtRePassword.setBounds(279, 220, 164, 20);
contentPane.add(txtRePassword);
```

```
    txtAdress = new JTextField();
    txtAdress.setColumns(10);
    txtAdress.setBounds(279, 257, 164, 20);
    contentPane.add(txtAdress);

    txtAdminPassword = new JPasswordField();
    txtAdminPassword.setColumns(10);
    txtAdminPassword.setBounds(279, 368, 164, 20);
    contentPane.add(txtAdminPassword);

    comboType = new JComboBox();
    comboType.setBounds(279, 332, 164, 20);
    comboType.addItem("Employee");
    comboType.addItem("Admin");
    contentPane.add(comboType);

    JButton btnCancel = new JButton("Cancel");
    btnCancel.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          dispose();
       }
    });
    btnCancel.setBounds(394, 420, 85, 23);
    contentPane.add(btnCancel);

    JButton btnReset = new JButton("Reset");
    btnReset.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          reset();
       }
    });
    btnReset.setBounds(299, 420, 85, 23);
    contentPane.add(btnReset);

    JButton btnSignup = new JButton("SignUp");
    btnSignup.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          String uname=txtUserName.getText();
          String pass=txtPassword.getText();
          String repass=txtRePassword.getText();
          String apass=txtAdminPassword.getText();
          String sql=("select * from login");
          if (pass.equals(repass)) {
             try {
             Statement stmt=dbc.conmethod();
             ResultSet rs=stmt.executeQuery(sql);
             int count =0;
             while(rs.next()){
                   String username=rs.getString("username");
                   String adminpassword=rs.getString("password");
                   String type=rs.getString("type");

 if((adminpassword.equals(apass))&&(type.equals("Admin"))){
                             signup1();
                             break;
```

```
                      }
                      else if(uname.equals(username)){
                           count=1;
                      }else{
                           JOptionPane.showMessageDialog(null,
"Invalid Admin Password");
                           break;
                      }
                 }
                 if (count==1){
                      JOptionPane.showMessageDialog(null, "This
UserName is Already Taken");
                      txtUserName.requestFocus(true);
                 }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
              }else{
                 JOptionPane.showMessageDialog(null, "Re-Password is
not Matched");
                 txtRePassword.requestFocus(true);
              }
           }
       });
       btnSignup.setBounds(204, 420, 85, 23);
       contentPane.add(btnSignup);

       JLabel lblMobileNo = new JLabel("Mobile No.");
       lblMobileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
       lblMobileNo.setBounds(112, 294, 85, 19);
       contentPane.add(lblMobileNo);

       txtMobileNo = new JTextField();
       txtMobileNo.setColumns(10);
       txtMobileNo.setBounds(279, 295, 164, 20);
       contentPane.add(txtMobileNo);
    }
    private void signup1(){
       String name,username,password,type,address,mobileno,sql;
       name=txtName.getText();
       username=txtUserName.getText();
       password=txtPassword.getText();
       type=comboType.getSelectedItem().toString();
       address=txtAdress.getText();
       mobileno=txtMobileNo.getText();
       sql="INSERT INTO
sss.login(name,username,password,type,address,mobileno,date)
values('"+name+"','"+username+"','"+password+"','"+type+"','"+addres
s+"','"+mobileno+"',now())";
       if (name.equals("") || username.equals("") ||
password.equals("") || address.equals("") || mobileno.equals("")){
           JOptionPane.showMessageDialog(null, "Please fill the all
field");
       }else
       {
           try {
           Statement stmt=dbc.conmethod();
```

```
                stmt.execute(sql);
                JOptionPane.showMessageDialog(null, "Data Entered");
                dispose();
            }
            catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
            }
        }
        private void reset(){
            txtName.setText("");
            txtUserName.setText("");
            txtPassword.setText("");
            txtRePassword.setText("");
            txtAdress.setText("");
            txtAdminPassword.setText("");
            txtMobileNo.setText("");
            comboType.setSelectedIndex(0);
            txtName.requestFocus(true);
        }
}
```



Sign Up Form

Name	
Username	
Password	
Re-Password	
Address	
Mobile No.	
Type	Employee
Admin Password	

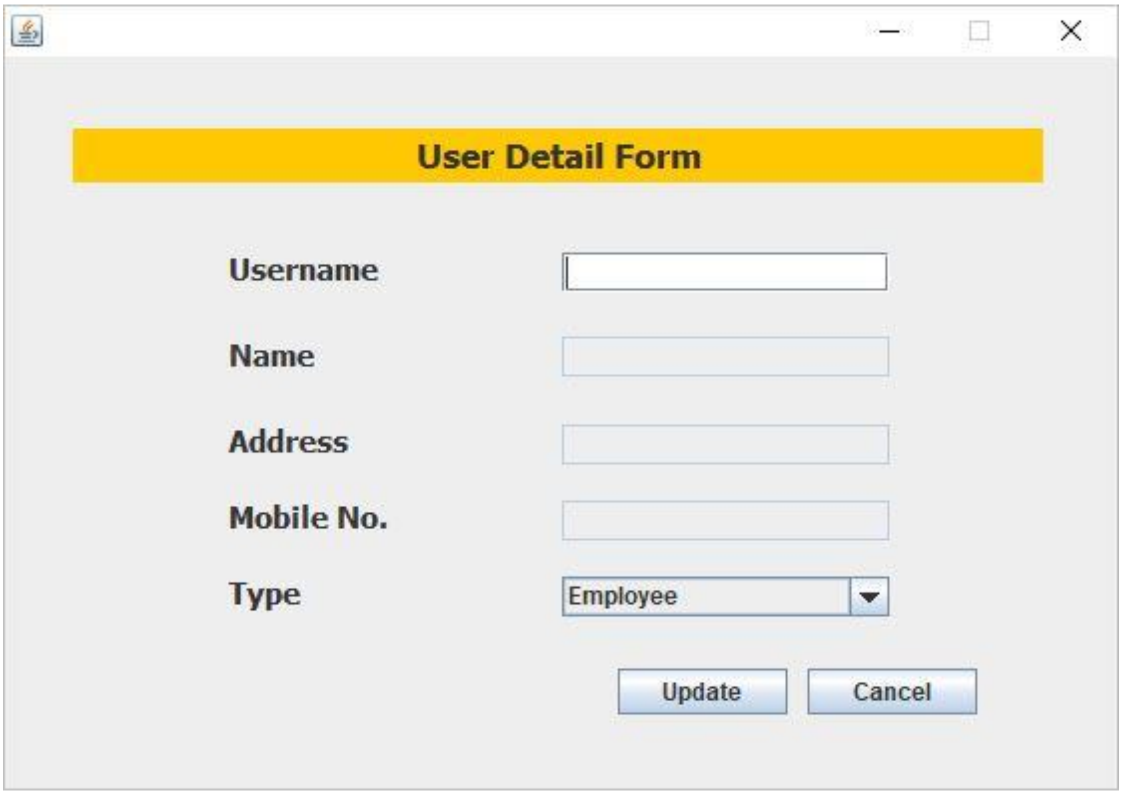
USER DETAIL

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
```

```java
import java.awt.Font;
import java.awt.Frame;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.ResultSet;
import java.sql.Statement;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;


public class userdetail extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtName;
    private JTextField txtUserName;
    private JTextField txtAdress;
    private JComboBox comboType;
    private JTextField txtMobileNo;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    userdetail frame = new userdetail();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public userdetail() {
        setResizable(false);
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 563, 395);
        contentPane = new JPanel();
```

```
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblSignUpForm = new JLabel("User Detail Form");
lblSignUpForm.setHorizontalAlignment(SwingConstants.CENTER);
lblSignUpForm.setBackground(Color.ORANGE);
lblSignUpForm.setFont(new Font("Tahoma", Font.BOLD, 17));
lblSignUpForm.setOpaque(true);
lblSignUpForm.setBounds(34, 36, 486, 27);
contentPane.add(lblSignUpForm);

JLabel lblNewLabel = new JLabel("Name");
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 15));
lblNewLabel.setBounds(112, 140, 54, 19);
contentPane.add(lblNewLabel);

JLabel lblUsername = new JLabel("Username");
lblUsername.setFont(new Font("Tahoma", Font.BOLD, 15));
lblUsername.setBounds(112, 97, 75, 19);
contentPane.add(lblUsername);

JLabel lblAddress = new JLabel("Address");
lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
lblAddress.setBounds(112, 183, 85, 19);
contentPane.add(lblAddress);

JLabel lblType = new JLabel("Type");
lblType.setFont(new Font("Tahoma", Font.BOLD, 15));
lblType.setBounds(112, 259, 54, 19);
contentPane.add(lblType);

txtName = new JTextField();
txtName.setEditable(false);
txtName.setBounds(279, 140, 164, 20);
contentPane.add(txtName);
txtName.setColumns(10);

txtUserName = new JTextField();
txtUserName.addKeyListener(new KeyAdapter() {
   @Override
   public void keyReleased(KeyEvent e) {
      userdetail1();
   }
});
txtUserName.setColumns(10);
txtUserName.setBounds(279, 98, 164, 20);
contentPane.add(txtUserName);

txtAdress = new JTextField();
txtAdress.setEditable(false);
txtAdress.setColumns(10);
txtAdress.setBounds(279, 184, 164, 20);
contentPane.add(txtAdress);

comboType = new JComboBox();
```

```
        comboType.setBounds(279, 260, 164, 20);
        comboType.addItem("Employee");
        comboType.addItem("Admin");
        contentPane.add(comboType);

        JButton btnCancel = new JButton("Cancel");
        btnCancel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnCancel.setBounds(402, 306, 85, 23);
        contentPane.add(btnCancel);

        JButton btnSignup = new JButton("Update");
        btnSignup.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              new userupdate().setVisible(true);
           }
        });
        btnSignup.setBounds(307, 306, 85, 23);
        contentPane.add(btnSignup);

        JLabel lblMobileNo = new JLabel("Mobile No.");
        lblMobileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblMobileNo.setBounds(112, 221, 85, 19);
        contentPane.add(lblMobileNo);

        txtMobileNo = new JTextField();
        txtMobileNo.setEditable(false);
        txtMobileNo.setColumns(10);
        txtMobileNo.setBounds(279, 222, 164, 20);
        contentPane.add(txtMobileNo);
     }
     private void userdetail1(){
        String sql="SELECT * FROM sss.login where username='"+
txtUserName.getText() +"'";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              txtName.setText(rs.getString("name"));
              txtAdress.setText(rs.getString("address"));
              txtMobileNo.setText(rs.getString("mobileno"));

     if(rs.getString("type").equals("Employee")){comboType.setSelecte
dIndex(0);}
              else
if(rs.getString("type").equals("Admin")){comboType.setSelectedIndex(
1);}
           }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
     }

}
```



User Detail Form

Username

Name

Address

Mobile No.

Type Employee

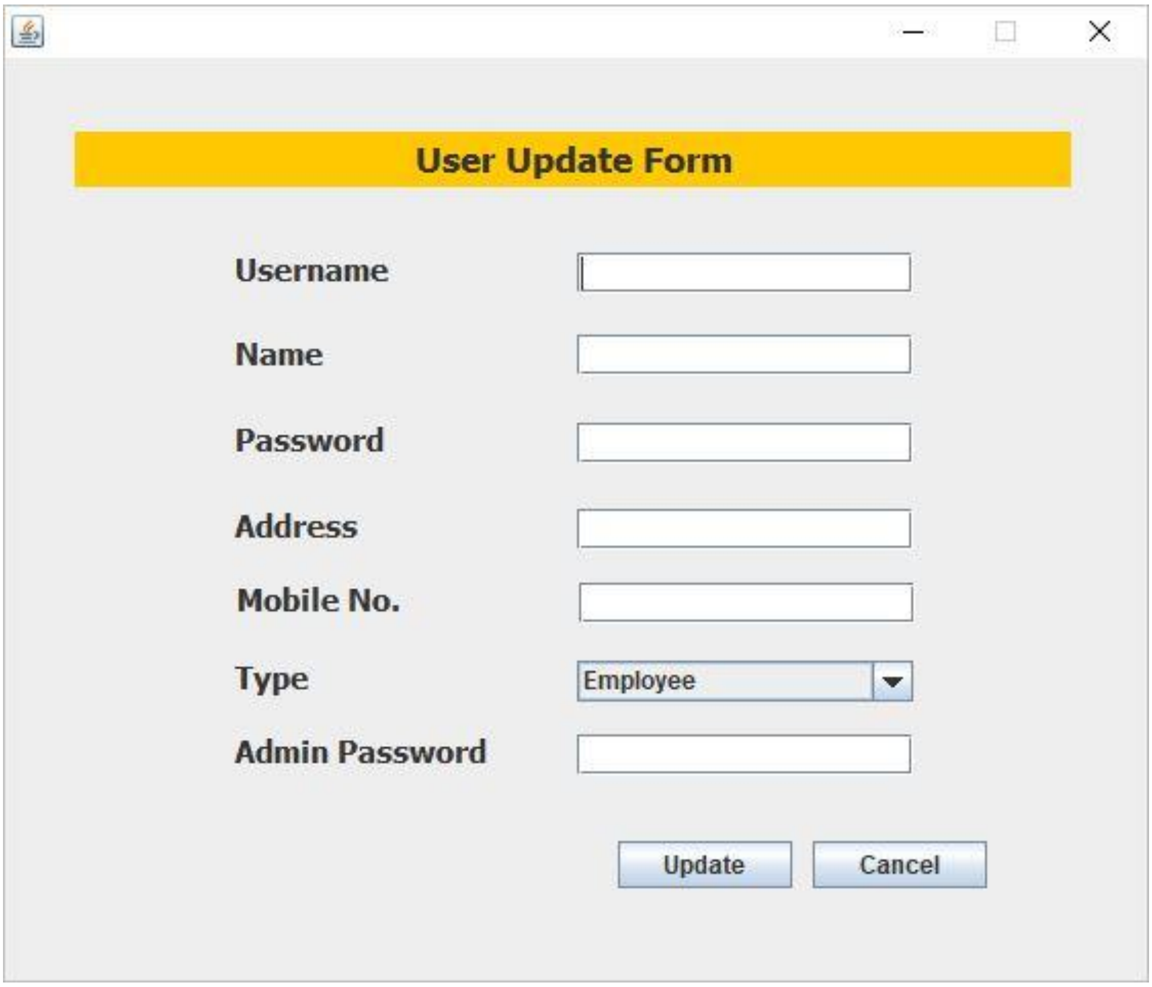
USER UPDATE

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Frame;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
```

```
public class userupdate extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtName;
    private JTextField txtUserName;
    private JTextField txtPassword;
    private JTextField txtAdress;
    private JTextField txtAdminPassword;
    private JComboBox comboType;
    private JTextField txtMobileNo;


    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    userupdate frame = new userupdate();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public userupdate() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 563, 479);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblSignUpForm = new JLabel("User Update Form");
        lblSignUpForm.setHorizontalAlignment(SwingConstants.CENTER);
        lblSignUpForm.setBackground(Color.ORANGE);
        lblSignUpForm.setFont(new Font("Tahoma", Font.BOLD, 17));
        lblSignUpForm.setOpaque(true);
        lblSignUpForm.setBounds(34, 36, 486, 27);
        contentPane.add(lblSignUpForm);

        JLabel lblNewLabel = new JLabel("Name");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblNewLabel.setBounds(112, 135, 54, 19);
        contentPane.add(lblNewLabel);
```

```
JLabel lblUsername = new JLabel("Username");
lblUsername.setFont(new Font("Tahoma", Font.BOLD, 15));
lblUsername.setBounds(112, 94, 75, 19);
contentPane.add(lblUsername);

JLabel lblPassword = new JLabel("Password");
lblPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
lblPassword.setBounds(112, 177, 75, 19);
contentPane.add(lblPassword);

JLabel lblAddress = new JLabel("Address");
lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
lblAddress.setBounds(112, 219, 85, 19);
contentPane.add(lblAddress);

JLabel lblType = new JLabel("Type");
lblType.setFont(new Font("Tahoma", Font.BOLD, 15));
lblType.setBounds(112, 293, 54, 19);
contentPane.add(lblType);

JLabel lblAdminPassword = new JLabel("Admin Password");
lblAdminPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
lblAdminPassword.setBounds(112, 329, 123, 19);
contentPane.add(lblAdminPassword);

txtName = new JTextField();
txtName.setBounds(279, 135, 164, 20);
contentPane.add(txtName);
txtName.setColumns(10);

txtUserName = new JTextField();
txtUserName.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent e) {
        userdetail();
    }
});
txtUserName.setColumns(10);
txtUserName.requestFocus(true);
txtUserName.setBounds(279, 95, 164, 20);
contentPane.add(txtUserName);

txtPassword = new JPasswordField();
txtPassword.setColumns(10);
txtPassword.setBounds(279, 178, 164, 20);
contentPane.add(txtPassword);

txtAdress = new JTextField();
txtAdress.setColumns(10);
txtAdress.setBounds(279, 220, 164, 20);
contentPane.add(txtAdress);

txtAdminPassword = new JPasswordField();
txtAdminPassword.setColumns(10);
txtAdminPassword.setBounds(279, 330, 164, 20);
```

```
        contentPane.add(txtAdminPassword);

        comboType = new JComboBox();
        comboType.setBounds(279, 294, 164, 20);
        comboType.addItem("Employee");
        comboType.addItem("Admin");
        contentPane.add(comboType);

        JButton btnCancel = new JButton("Cancel");
        btnCancel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnCancel.setBounds(394, 382, 85, 23);
        contentPane.add(btnCancel);

        JButton btnSignup = new JButton("Update");
        btnSignup.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              String uname=txtUserName.getText();
              String apass=txtAdminPassword.getText();
              String sql=("select * from login");
              int count=0;
              String adminpassword="",type="";

              try {
              Statement stmt=dbc.conmethod();
              ResultSet rs=stmt.executeQuery(sql);
              while(rs.next()){
                 String username=rs.getString("username");
                 adminpassword=rs.getString("password");
                 type=rs.getString("type");
                 if((username.equals(uname))){
                      count=1;
                 }
              }
              if(count==1){
                 userupdate1();
              }else
if((adminpassword.equals(apass))&&(type.equals("Admin"))){
                 userupdate1();
              }
              else{
                 JOptionPane.showMessageDialog(null, "Invalid
Username or Admin Password");
              }
              } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        btnSignup.setBounds(299, 382, 85, 23);
        contentPane.add(btnSignup);

        JLabel lblMobileNo = new JLabel("Mobile No.");
        lblMobileNo.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblMobileNo.setBounds(113, 255, 85, 19);
        contentPane.add(lblMobileNo);

        txtMobileNo = new JTextField();
        txtMobileNo.setColumns(10);
        txtMobileNo.setBounds(280, 256, 164, 20);
        contentPane.add(txtMobileNo);
    }
    private void userdetail(){
        String sql="SELECT * FROM sss.login where username='"+
txtUserName.getText() +"'";
        try {
           Statement stmt=dbc.conmethod();
           ResultSet rs=stmt.executeQuery(sql);
           while (rs.next()){
              txtName.setText(rs.getString("name"));
              txtAdress.setText(rs.getString("address"));
              txtPassword.setText(rs.getString("password"));
              txtMobileNo.setText(rs.getString("mobileno"));

    if(rs.getString("type").equals("Employee")){comboType.setSelecte
dIndex(0);}
              else
if(rs.getString("type").equals("Admin")){comboType.setSelectedIndex(
1);}
           }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }

    private void userupdate1(){
        String name,username,password,type,address,mobileno,sql;
        name=txtName.getText();
        username=txtUserName.getText();
        password=txtPassword.getText();
        type=comboType.getSelectedItem().toString();
        address=txtAdress.getText();
        mobileno=txtMobileNo.getText();
        sql="UPDATE sss.login SET
name='"+name+"',password='"+password+"',type='"+type+"',address='"+a
ddress+"',mobileno='"+mobileno+"' WHERE username='"+username+"' ";
        if (name.equals("") || username.equals("") ||
password.equals("") || address.equals("") || mobileno.equals("")){
           JOptionPane.showMessageDialog(null, "Please fill the all
field");
        }else
        {
           try {
           Statement stmt=dbc.conmethod();
           stmt.execute(sql);
           JOptionPane.showMessageDialog(null, "Data Entered");
           dispose();
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
        }
        }
    }
```



User Update Form

Username

Name

Password

Address

Mobile No.

Type: Employee

Admin Password

USER DELETE

```java
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Frame;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;

import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;


public class userdelete extends JFrame {

    dbconnection dbc = new dbconnection();
    private JPanel contentPane;
    private JTextField txtName;
    private JTextField txtUserName;
    private JTextField txtAdress;
    private JTextField txtAdminPassword;
    private JComboBox comboType;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    userdelete frame = new userdelete();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public userdelete() {
        setExtendedState(Frame.MAXIMIZED_BOTH);
        setResizable(false);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 563, 400);
```

```
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

JLabel lblSignUpForm = new JLabel("User Delete Form");
lblSignUpForm.setHorizontalAlignment(SwingConstants.CENTER);
lblSignUpForm.setBackground(Color.ORANGE);
lblSignUpForm.setFont(new Font("Tahoma", Font.BOLD, 17));
lblSignUpForm.setOpaque(true);
lblSignUpForm.setBounds(34, 36, 486, 27);
contentPane.add(lblSignUpForm);

JLabel lblNewLabel = new JLabel("Name");
lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 15));
lblNewLabel.setBounds(112, 141, 54, 19);
contentPane.add(lblNewLabel);

JLabel lblUsername = new JLabel("Username");
lblUsername.setFont(new Font("Tahoma", Font.BOLD, 15));
lblUsername.setBounds(112, 100, 75, 19);
contentPane.add(lblUsername);

JLabel lblAddress = new JLabel("Address");
lblAddress.setFont(new Font("Tahoma", Font.BOLD, 15));
lblAddress.setBounds(112, 183, 85, 19);
contentPane.add(lblAddress);

JLabel lblType = new JLabel("Type");
lblType.setFont(new Font("Tahoma", Font.BOLD, 15));
lblType.setBounds(112, 221, 54, 19);
contentPane.add(lblType);

txtName = new JTextField();
txtName.setEditable(false);
txtName.setBounds(279, 141, 164, 20);
contentPane.add(txtName);
txtName.setColumns(10);

txtUserName = new JTextField();
txtUserName.addKeyListener(new KeyAdapter() {
   @Override
   public void keyReleased(KeyEvent e) {
      userdetail();
   }
});
txtUserName.setColumns(10);
txtUserName.setBounds(279, 101, 164, 20);
contentPane.add(txtUserName);

txtAdress = new JTextField();
txtAdress.setEditable(false);
txtAdress.setColumns(10);
txtAdress.setBounds(279, 184, 164, 20);
contentPane.add(txtAdress);
```

```
        comboType = new JComboBox();
        comboType.setBounds(279, 222, 164, 20);
        comboType.addItem("Employee");
        comboType.addItem("Admin");
        contentPane.add(comboType);

        JButton btnCancel = new JButton("Cancel");
        btnCancel.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {
              dispose();
           }
        });
        btnCancel.setBounds(405, 308, 85, 23);
        contentPane.add(btnCancel);

        JButton btnSignup = new JButton("Delete");
        btnSignup.addActionListener(new ActionListener() {
           public void actionPerformed(ActionEvent e) {

              String apass=txtAdminPassword.getText();
              String sql=("select * from login");
              try {
              Statement stmt=dbc.conmethod();
              ResultSet rs=stmt.executeQuery(sql);
              while(rs.next()){
                 String adminpassword=rs.getString("password");
                 String type=rs.getString("type");

    if((adminpassword.equals(apass))&&(type.equals("Admin"))){
                           userremove();
                           break;
                 }else{
                      JOptionPane.showMessageDialog(null, "Invalid
Admin Password");
                      break;
                 }
              }
           } catch (Exception ex){
JOptionPane.showMessageDialog(null, ex);}
           }
        });
        btnSignup.setBounds(310, 308, 85, 23);
        contentPane.add(btnSignup);

        JLabel lblAdminPassword = new JLabel("Admin Password");
        lblAdminPassword.setFont(new Font("Tahoma", Font.BOLD, 15));
        lblAdminPassword.setBounds(112, 263, 123, 19);
        contentPane.add(lblAdminPassword);

        txtAdminPassword = new JPasswordField();
        txtAdminPassword.setColumns(10);
        txtAdminPassword.setBounds(279, 264, 164, 20);
        contentPane.add(txtAdminPassword);
    }

    private void userdetail(){
        String sql="SELECT * FROM sss.login where username='"+
txtUserName.getText() +"'";
        try {
            Statement stmt=dbc.conmethod();
            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
               txtName.setText(rs.getString("name"));
               txtAdress.setText(rs.getString("address"));

    if(rs.getString("type").equals("Employee")){comboType.setSelecte
dIndex(0);}
               else
if(rs.getString("type").equals("Admin")){comboType.setSelectedIndex(
1);}
            }
        }catch(Exception ex){JOptionPane.showMessageDialog(null,
ex);}
    }
    private void userremove(){
        String username,sql;
        username=txtUserName.getText();
        sql="DELETE FROM sss.login WHERE username='"+username+"' ";
        try {
            Statement stmt=dbc.conmethod();
            stmt.execute(sql);
            JOptionPane.showMessageDialog(null, "Account Deleted");
            dispose();
        }
        catch (SQLException e){JOptionPane.showMessageDialog(null,
e);}
    }
}
```

Testing

Software testing is a process of examine if the software meets those stated requirements or not. Software testing is the process of evaluating software for its quality in terms of meeting in the requirements accuracy and efficiency. Software testing is also required to test the performance of the system. Software of a good quality not only works for correctly but efficiently as well.

- **Unit Testing**: The individual components such as procedures, function written for connection establishment, clearing the forms after certain operations and testing certain thing like user name, password, length of the text etc. are tested in my project.

- **Module Testing**: A module is collection of dependents components; various forms designed for various purposes such as updating main stock or transferring from maim stock to sale stock, issue of medicine to various department. All these dependent modules are tested individually

- **System Testing**: At the end all the individual and dependent modules are integrated to make up the entire systems. This process was concerned with finding errors. So, I tested the software as a whole and see the workflow of it with the purpose of finding error.

For testing this project, I checked on dummy data, whether the software is operating according to the given requirements or not.

- White Box Testing
- Black Box Testing
- Alpha Testing
- Beta Testing

White box testing

White box testing is also known as structural testing or glass box testing. Its goal is to test the internal code of the software it tests the program at the level of source code. Hair the tester has the knowledge of actual source code of the software at what is tested is the inner structure of the program. Test cases are written with the knowledge of the logic of program. We are only concerned with the testing of accuracy of the logic of the program. We don't focus upon the requirements of the software

Black box testing

Black box testing is also known as functional testing. The sole purpose of black box testing is to test the application or software from its functionality point of view. In this type of testing, the software is tested to check whether the software fulfils, all the specified requirements. In black box testing, a tester is not concerned about testing the logic of the program. The internal details of the program are not known to the tester. In this type of testing, the software is like a black box to the tester where internal details are undisclosed. The tester only Test the functionality of the program by supplying an input observing the output. Black box testing is a detailed inspection of the software functionality against the already specified requirements for which it is developed. We may agree that black box testing verify the software to ensure that it does exactly that, which it is required to do.

Alpha testing

Alpha testing, a proposed customer/end user who is ready to use the software is identified. End user perform the testing at the developer site itself. The software is used by the end user, why the developer observes the process. The end user uses the software in its natural way, and the developer focus on the noticing the way the user takes and uses the software. Developers focus on noticing the usability problems faced by the end user as well as the errors countered.

Beta testing

In beta testing, an end user is identified who is ready to perform the testing at his site. Beta testing is performed at the site of inducer and the developer is not present. The software product is treated as life application by the end user and used in its natural way. Note that the developer is not present at the moment. Hence, in beta testing the environment in which testing is performed is actually the simulation of the environment where the product will be actually used. The end user testing the software and records all the errors encountered. The recorded error and usability problem mark conveyed to the developers. Developers takes their time to understand the error and make necessary change in the software.

CONCLUSION

Finally at last, this project is based on banking system.
Where any user can deposit money, takes loans, withdraw money and many more things.

This project made for mainly use in those area which are very far from cities and peoples of there were not more educated.

The project has been designed to use all the resources and manual data into further processing and to make system robust. Certain enhancement can isolate drawbacks of this system. The following features can make some improvement in the project:-

- Payment gateway will provide payment facilities to customers form which they can make payment online.
- SMS gateway will allow user-friendly automation of software for notification related services.
- Bug reporting feature will allow further upgradation of software into an error free project.

REFRENCES

To complete this project, I have taken reference from the following sources.

1. *CORE JAVA BY C.KOMALAVALI.*
2. *INTRODUCTION TO DBMS CIMAGE PUBLICATION.*
3. *DATABASE MANAGEMENT BY JOHANNES GEHRKE.*
4. *INTRODUCTION TO JAVA BY SANJIB K. SAHU.*
5. *INTRODUCTION TO MYSQL:* *https://www.tutorialspoint.com/mysql/mysql-introduction.htm*
6. *JAVA SWING:* *https://www.javatpoint.com/java-swing*

7. *Umar, Mubarak Albarka (2020): Comprehensive study of software testing: Categories, levels, techniques, and types. TechRxiv. Preprint.* *https://doi.org/10.36227/techrxiv.12578714.v2*

8. *Slade, Anthony Lynn. Brigham Young University ProQuest Dissertations Publishing, 2003. 28108736.*

   https://www.proquest.com/docview/2440906054