

**PROJECT SUBMITTED TO THE
UNIVERSITY OF MUMBAI
IN PARTIAL OF FULLFILLMENT OF THE DEGREE
OF
BACHELOR OF SCIENCE**

**SMT. CHANDIBAI HIMATMAL
MANSUKHANI COLLEGE
ULHASNAGAR-421003**

**T.Y.B.Sc (COMPUTER SCIENCE)
ACADAMIC YEAR 2017-2018**

**OOAD PROJECT ON
BANK MANAGEMENT SYSTE
BY
Mr. AKSHAY GAJANAN PATIL
PROJECT INCHARGE
Mrs. KIRTI BAKADE**

ACKNOWLEDGEMENT

In successfully completing my project , I am indeed thankful to a large number of people who have encouraged and helped me in a variety of ways. It is definitely a privilege to express my gratitude to the people involved.

Firstly I would like to thank Head of the Department.

Mrs. Ritika Sachdev for allowing us to go ahead with this project and providing us their valuable guidance, and kind and helpful support.

*My profound and sincere thanks to **Mrs. Kirti Bakade** who consented to be my project guide. I express my courtesy towards her for her keen interest valuable guidance, important judgement and constant encouragement during the entire course of my study.*

Thank to her for timely and important help in completing this project report.

With limitless humanity, I thank my all teaching staff and friends who are given me enough courage to accomplish this project.

Last but not the least, I owe my sincere regards to the 'Almighty' for making me able to believe in myself and letting me utilize my potential to the fullest of my energy during the entire course of study.

(Mr. Akshay Gajanan Patil)

BANK MANAGEMENT SYSTEM

Index

SYSTEM DOCUMENTATION	5
ORGANIZATIONAL OVERVIEW	6
DESCRIPTION OF PRESENT SYSTEM	6
LIMITATION OF PRESENT SYSTEM.....	7
PROPOSED SYSTEM.....	7
ADVANTAGES OF THE PROPOSED SYSTEM.....	8
SCOPE.....	8
SOFTWARE REQUIREMENT	9
FEASIBILITY STUDY.....	10
STAKEHOLDERS	11
GANTT CHART	12
SYSTEM ANALYSIS	14
IDENTIFY THE CUSTOMER'S NEED	15
FACT FINDING TECHNIQUES.....	15
EVENT LIST.....	16
EVENT TABLE	17
ENTITY RELATIONSHIP DIAGRAM	18
USE CASE DIAGRAMS & DESCRIPTION.....	22
ACTIVITY DIAGRAM	24
CLASS DIAGRAM	26
OBJECT DIAGRAM.....	27
SEQUENCE DIAGRAM.....	28
STATE DIAGRAM	30
SYSTEM DESIGN	31
CONVERTING ERD TO TABLES.....	32
COMPONENT DIAGRAM	33
PACKAGE DIAGRAM	34
DEPLOYMENT DIAGRAM	35

BANK MANAGEMENT SYSTEM

PROGRAM FLOW CHART.....	36
STRUCTURE CHART.....	38
PROJECT DOCUMENTATION	39
MENU TREE	40
LIST OF TABLES WITH ATTRIBUTES	41
PROGRAM DESCRIPTION.....	42
PROGRAM SOURCE CODE	43
DATA ENTRY SCREENS	118
REPORT LAYOUTS.....	126
TESTING.....	129
REFERENCES & BIBLIOGRAPHY.....	134

SYSTEM DOCUMENTATION

BANK MANAGEMENT SYSTEM

ORGANIZATIONAL OVERVIEW

This project is aimed to developing an Bank Management System. This is as easy to operate user friendly, economically priced software that committed to meet the all requirements to manage small, medium and big scale.

Bank Management System is one of the leading software for complete management of Bank. It can manage the accounts information, transactions and employee management excellently. The Bank Management software provides all kinds of record keeping feature required in a Bank like Account Creation, Transaction like Withdrawal and Deposite, Update and Delete information and many more automatically.

This project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of MySQL and all the user interfaces have been designed using Java NetBeans IDE 8.2 technology. The database connectivity is planned using the SQLConnector methodology. The Standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are forwarded by the administrative staff.

DESCRIPTION OF PRESENT SYSTEM

The existing system of “Bank” is done manually which is too lengthy and tedious. Before decades the banks do not have computerized system for maintaining records and daily transaction. Client transaction and information all the entries are manually written on books. Also maintaining the record of each created account, each transaction like deposits and withdraw everything was done in books. These all entries are manually calculated by accountant. The current system is lot of time consuming and also it is very costly because it involves lot of paperwork and to manually handle such a system was very difficult task.

The system uses, low quality computer system human language communication to manage the organization this delays information transmission in the events. When a customer wants to do any transaction or to update the account information like update mobile no, link Aadhar card or address like that he just needs to come to bank and submits it all valid proof and then the employees are do the respective changes as the customer wanted. So there is no need to do any manually work and the system is also very less time consuming the any work is done in only in minutes.

BANK MANAGEMENT SYSTEM

LIMITATION OF PRESENT SYSTEM

Since the present system is manual it faces many of the drawbacks of the conventional management system. Some of the limitations of the existing system are:

➤ **Staking of files**

As the records increase the no. of files used to store.

Record also increases thus increasing the financial need.

➤ **Thorny jb**

Modification and searching of records are difficult in the existing system.

➤ **Report**

All Reports is manually after analysing 2 to 3 files.

This waste time and increase work load.

➤ **Error**

As the records are maintain and manage

By humans, it may lead to error

PROPOSED SYSTEM

To overcome the limitation of the current system and to increase the working efficiently of the system I decided to computerized the current system.

- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- There is no risk of data mismanagement at any level while the project development is under process.
- Report generation feature is provided using Crystal Reports to generate different kinds of reports in tabular format.
- It provides high level of security like hashing.
- Creating the secure transaction process.
- Maintain all information of customers like his all personal information and transaction details.

BANK MANAGEMENT SYSTEM

ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system is computerized System provides some advantages as follows.

- To improve the efficiency. Many events can be handle through this software.
- Quickly find out information of employee.
- To provide easy access. Reduce the paper work.
- To provide user friendly environment.
- Create backup of data which is stored in database.
- Only authorized person can only use, so it avoid misuse of system.
- It offers very high flexibility in defining various, dine in orders.
- That takes care of all your requirements relating to accounting and management of employees payroll.
- The reliability of data and procedure increase because of computer.

SCOPE

- The system provide large storage area, large amount of information can be stored.
- The system reduces paper work and hence information can be stored and maintained easily.
- It reduces time consumption save time.
- This system is very reliable, efficient as compared to manual system.
- Data can be checked, changes can be made wherever necessary.
- Store the required details in a database.

This software is being developed for the administrator. In this system the records of each customer request details are preserved along with their status and transaction related to them. The system is also made secured as all the updating of the items and transactions can be done by the authorized person i.e. manager, clerks, IT employee and Database Administrator.

BANK MANAGEMENT SYSTEM

SOFTWARE REQUIREMENT

Hardware Requirements:

- Hard Disk Space: Minimum 200MB Free Space
- CPU Minimum: 1000 MHz
- RAM Minimum: 1 GB

Software Requirements:

- Operating System: Windows 10 Home.
- Front End: NetBeans IDE 8.2
- Back End: MySQL Database

FEASIBILITY STUDY

Feasibility study is the most important step. It is the determination of whether the project is feasible or not. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements. The feasibility study contains all the factors i.e. Technical, Behavioral and economical feasibility.

Technical Feasibility:-

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.

Time Feasibility:-

- Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated period. If a project takes too much time it is likely to be rejected. The proposed system takes a little time period for implementation so it can be considered as Time Feasible Software.

STAKEHOLDERS

Stakeholders are those entities within or outside an organization which sponsor a project, or have an interest or a gain upon a successful completion of a project.

May have a positive or negative influence in the project completion.

Stakeholders for Bank Management System are:-

➤ Users:

The users are the who handles most of the functionalities of a System as they play an important role using the system.

➤ Employees:-

Employess are like manager, IT persons, Database administrator. Staff does the basic functions.

BANK MANAGEMENT SYSTEM

GANTT CHART

Sr. No.	Phase	Duration	Start Date	Finish Date	Actual Finish Date	Signature
1	Planning Phase	14	15-Dec-17	28-Dec-17	28-Dec-17	
1.1	Defining Problem	6	15-Dec-18	20-Dec-17	20-Dec-17	
1.2	Confirm Project Feasibility	3	21-Dec-17	23-Dec-17	23-Dec-17	
1.3	Project Schedule	2	24-Dec-17	25-Dec-17	24-Dec-17	
1.4	Staffing Project	2	26-Dec-17	27-Dec-17	27-Dec-17	
1.5	Launching project	1	28-Dec-17	28-Dec-17	28-Dec-17	
<hr/>						
2	Analysis Phase	16	29-Dec-17	13-Jan-18	13-Jan-18	
2.1	Gathering Information	6	29-Dec-17	03-Jan-18	03-Jan-18	
2.2	Defining System Requirement	6	04-Jan-18	09-Jan-18	09-Jan-18	
2.3	Prioritize Requirements	2	10-Jan-18	11-Jan-18	11-Jan-18	
2.4	Generate and evaluate alternatives	1	12-Jan-18	12-Jan-18	12-Jan-18	
2.5	Review recommendation	1	13-Jan-18	13-Jan-18	13-Jan-18	

BANK MANAGEMENT SYSTEM

3	Design Phase	27	14-Jan-18	09-Feb-18	09-Feb-18	
3.1	Design user Inteface	15	14-Jan-18	28-Jan-18	28-Jan-18	
3.2	Design Application Architecture	6	29-Jan-18	03-Feb-18	03-Feb-18	
3.3	Design and interact Database	4	04-Feb-18	04-Feb-18	04-Feb-18	
3.4	Design and Integrate system Controls	2	08-Feb-18	09-Feb-18	09-Feb-18	
4	Implementation phase	28	10-Feb-18	09-Mar-18	09-Mar-18	
4.1	Construct the software components	20	10-Feb-18	01-Mar-18	01-Mar-18	
4.2	Verify and test the compnents	5	02-Mar-18	06-Mar-18	06-Mar-18	
4.3	Training and documenting	2	07-Mar-18	08-Mar-18	08-Mar-18	
4.4	Install Software	1	09-Mar-18	09-Mar-18	09-Mar-18	
5	Support Phase	-	10-Mar-18	Till Date	Til Date	

SYSTEM ANALYSIS

In System Analysis and Design phase, the whole software development process, the overall software structure and its outlay are defined. In case of the client/server processing technology, the number of tiers required for the package architecture, the database design, the data structure design etc are all defined in this phase. After designing part a software development model is created. Analysis and Design are very important in the whole development cycle process. Any fault in the design phase could be very expensive to solve in the software development process. In this phase, the logical system of the product is developed.

System analysis is mainly conducted with the following objectives in mind:

1. Identify the customer's need.
2. Fact Finding Techniques.

IDENTIFY THE CUSTOMER'S NEED

The first step of the system analysis process involves the identification of the needs. The analyst meets the customers at the end users to define the goals are identified, the analyast moves on to an evaluation of supplementary information .Information gathered during the needs identification is the main tool for the development of system or package.

This phase some information is gathered by the customer before meeting with analyst. Customer-analyst communication results in modifications of the identification of need is to know about the products objectives and to define the goals required meeting the objective.

FACT FINDING TECHNIQUES

Fact-finding is an important activity in system analysis phase.In this stage, the functioning of the system is to be understood by the system analyst to design the proposed system.

The techinques employed to gather this data are known as fact-finding techniques.

There are various fact finding techniques like conduction personal interview, preparing a questionnaire, and etc.

- How the systems function?
- Which are the factors necessary for it to perform satisfactory?
- What are various transactions in the system?
- How these transactions affect the working of the system?
- Drawbacks of the system.

EVENT LIST

- Admin Login to the system.
- Admin manage the users.
- Admin logout from the system.
- User login to the system.
- User manage the customers account.
- User manages all the transaction of customers.
- User as well as admin will generate reports.
- User logout from the system.

BANK MANAGEMENT SYSTEM

EVENT TABLE

No.	EVENT	TRIGGER	SOURCE	Activity	RESPONSE	DESTINATION
1)	Admin login to system	Login	Admin	Login in	Logged in	Admin
2)	Admin Add user	Admin	Admin	Add user	Details of new user.	Admin
3)	Admin wants to logout.	Request for logout.	Admin	Logout Detail.	Admin logout Confirmation.	System
4)	User login to system	Request for login	user	Login in	Logged in	user
5)	User wants to manage account	Manage account	user	Manage account	Manage Details	user
6)	User wants to manage transaction	Manage transaction	user	Manager Transaction	Manage Transaction	user
7)	user wants to Update customer Record	Customer Detail.	user	update customer.	Update records.	user
8)	Admin wants to delete Employee Record	Employee Detail.	Admin	Delete employee.	Delete records	Admin
11)	Admin wants to delete customer Record	Customer Detail.	Admin	Delete customer.	Delete records.	Admin
12)	Admin wanrts to delete user record	User Details	Admin	Delete user	Delete records	Admin

ENTITY RELATIONSHIP DIAGRAM

ERD Diagram

A graphical model of the data needed by a system, including things about which information is stored & the relationships among them, produced in structured analysis & information engineering. ER Diagram represents entities or tables and their relationships with one another.

Entities

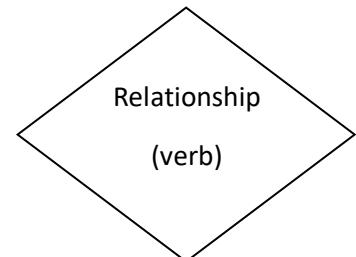
The entity is a person, object, place or event for which data is collected. . It is the "subject" of a table. The entity is represented by a rectangle and labeled with a singular noun. Each entity is represented by a box within the ERD. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity.

Relationships

ENTITY NAME

The relationships the interaction between the entities

Relationships are represented by lines between entities.



Optionality and Cardinality

Symbols at the ends of the relationship lines indicate the optionality and the cardinality of each relationship. “Optionality” expresses whether the relationship is optional or mandatory. “Cardinality” expresses the maximum number of relationships. As a relationship line is followed from an entity to another, near the related entity two symbols will appear.

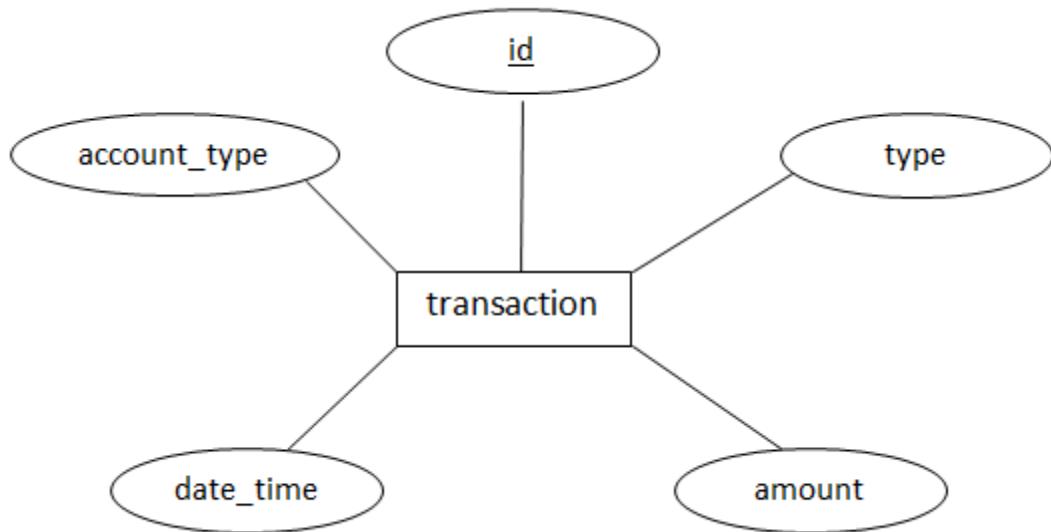
BANK MANAGEMENT SYSTEM

Attributes

Attributes are the data we collect about the entity. Attributes are drawn as ovals and are connected with a line to exactly one entity or relationship set. Attributes are often omitted as they can clutter up a diagram; other diagram techniques often list entity attributes within the rectangles drawn for entity sets.

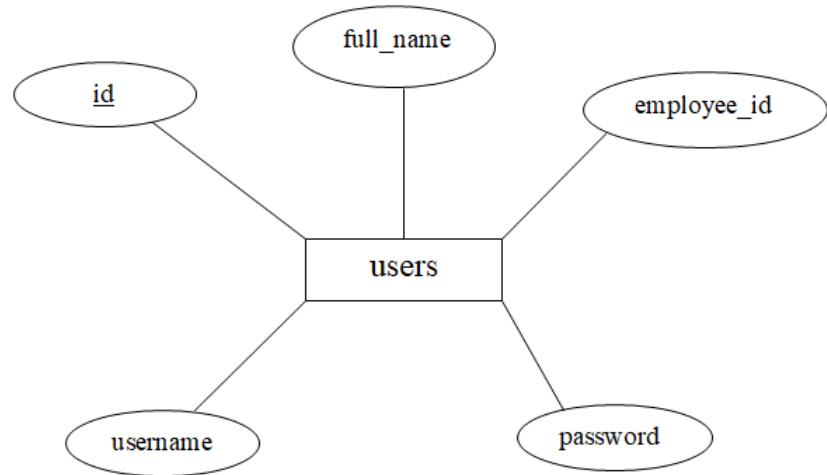


Transaction ER Diagram:

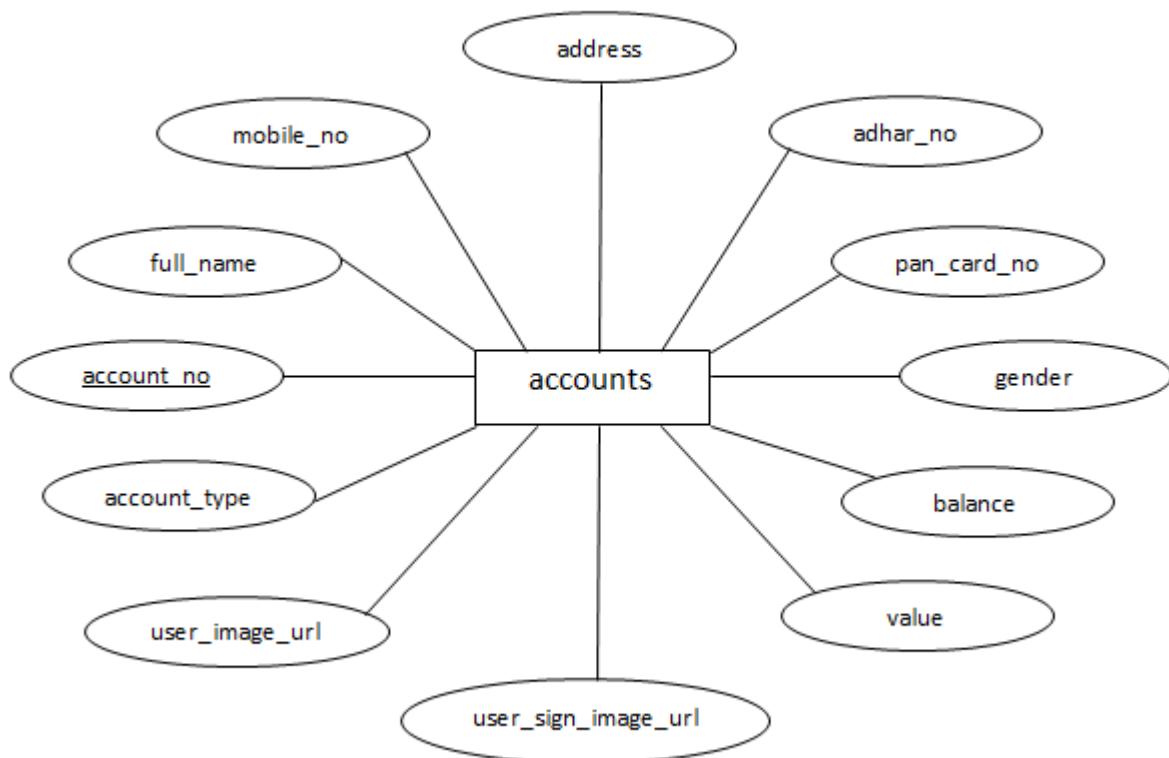


BANK MANAGEMENT SYSTEM

Users ER Diagram:

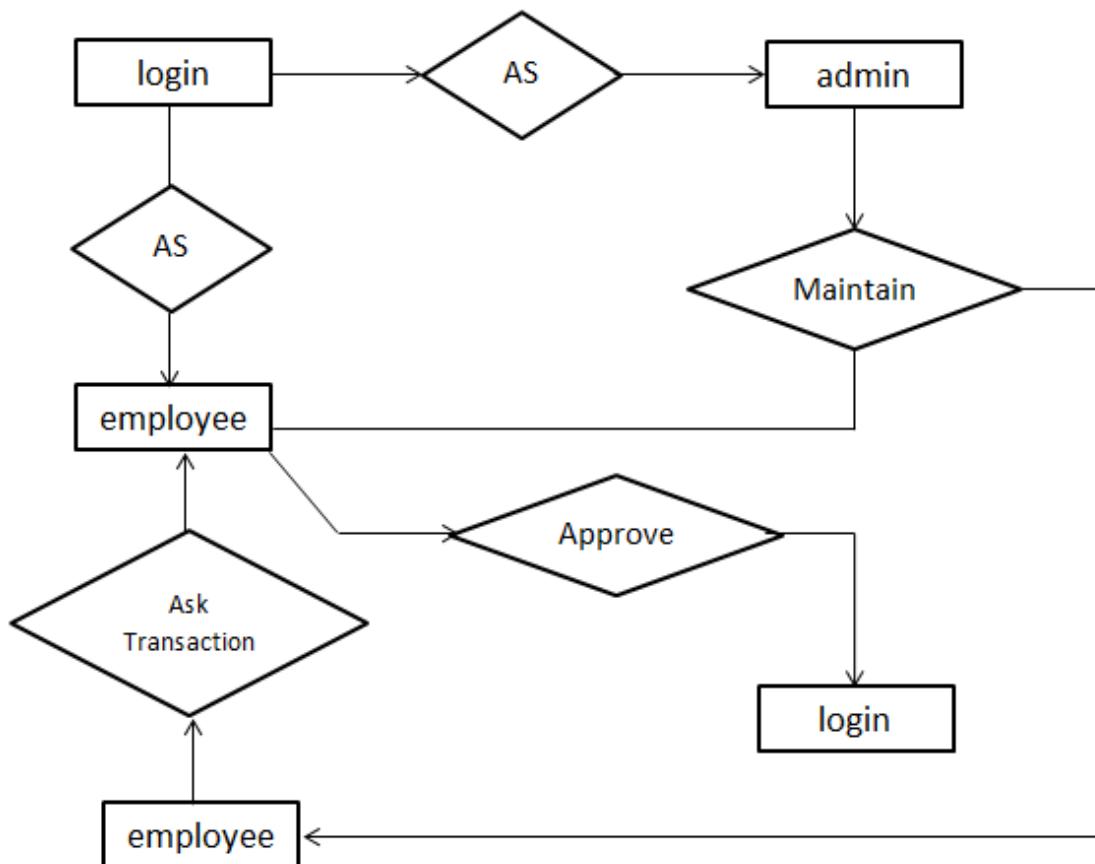


accounts ER Diagram:



BANK MANAGEMENT SYSTEM

ER Diagram:



USE CASE DIAGRAMS & DESCRIPTION

A use case diagram is the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use – case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors their goals (represented as use cases) and any dependencies between those use cases. The main purpose of use case diagram is to show what system functions are performed for which actors can be part of assumption roles of actors in the System can defected.

Interaction among actors is not shown on the use case diagram. If this interaction is essential to a coherent description of the desired behavior, perhaps the system of use case diagram boundaries should be examined. Alternatively interaction among actors can be part of the assumption used in use case.

➤ Use cases

- A use case describes a sequence of actions that provide something of measurable values to an actor and drawn as a horizontal ellipse.

➤ Actors

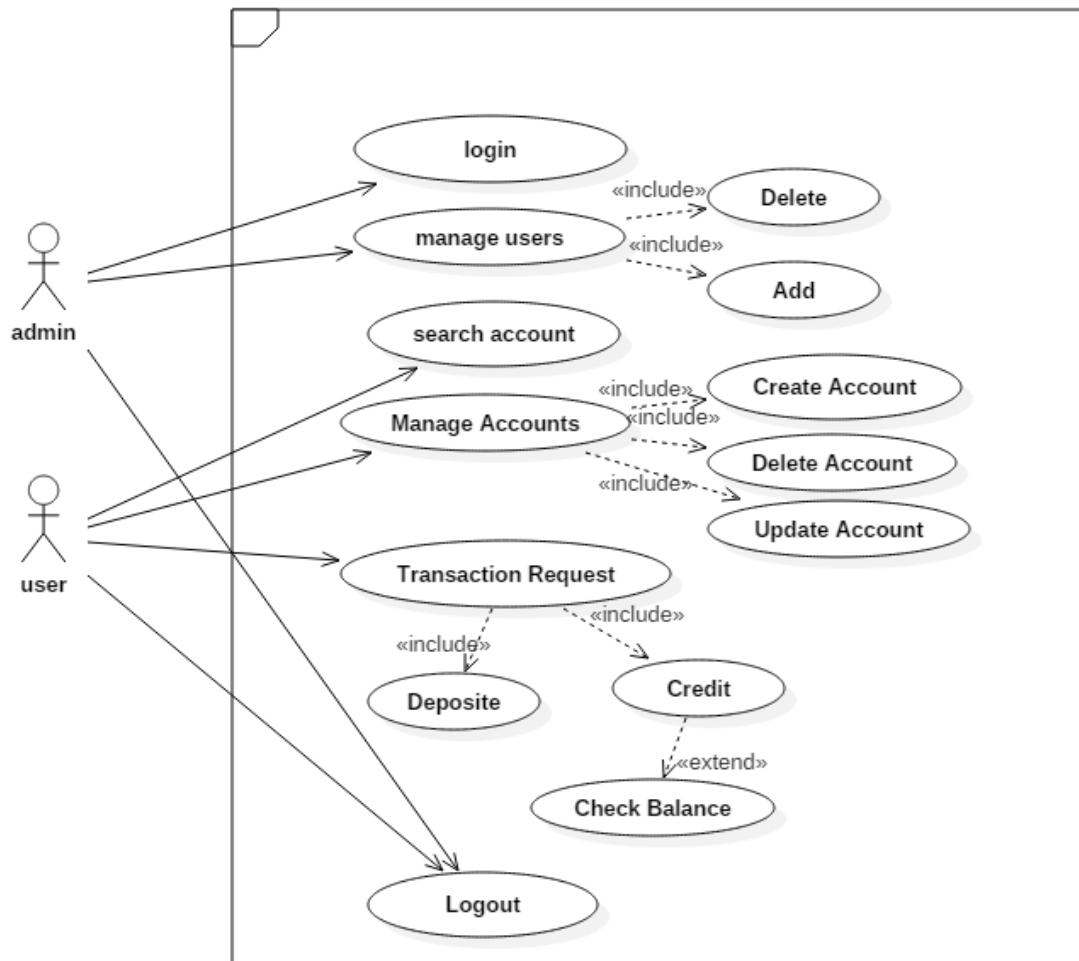
- An Actor is a person, organization external system that plays a role in one or more interaction with the system.

➤ System Boundaries boxes (Optional)

A rectangle is drawn around the use cases called the System Boundary Box to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

BANK MANAGEMENT SYSTEM

Use Case Diagram:



ACTIVITY DIAGRAM

- Activity diagrams are graphical representations of work flows of stepwise activities and action which support for choice interaction and concurrency.
- In the unified modeling language, activity diagram can be used to describe the business and the operational step by step workflows of components in the system.
- An Activity diagram shows the overall flow of control.

When to use Activity Diagram?

Activity Diagram should be used in conjunction with the other modeling techniques such as interaction diagram and state diagram.

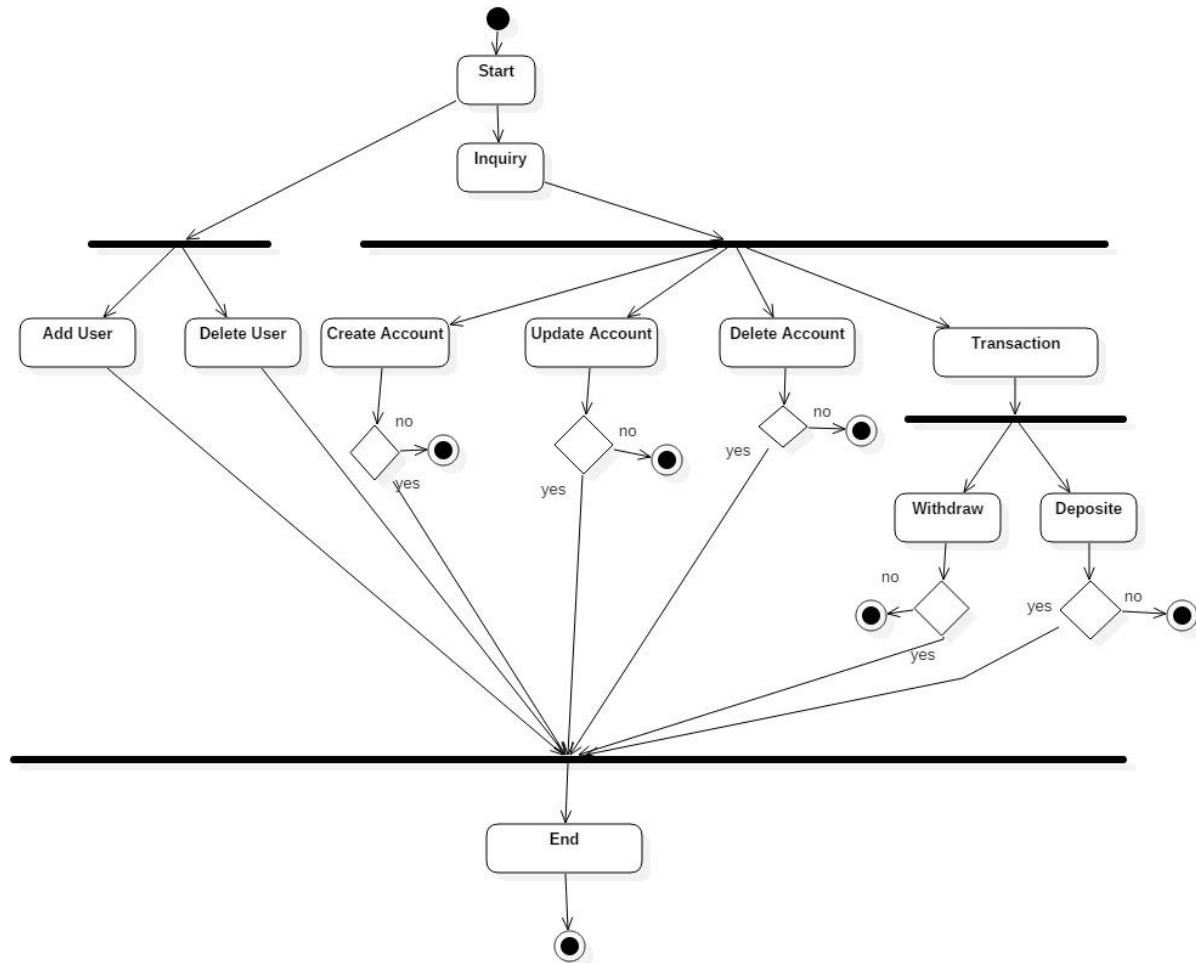
The main reason to use activity diagram is to model the workflow behind the system being designed.

Activity Diagrams are also useful for : analyzing a use case by describing what action needs to take place and when they should occurs; describing a complicated sequential algorithm; and algorithm with parallel processes.

However Activity Diagram should not take place of interaction diagram and state diagram. Activity diagrams do not give details about how object behave or how object collaborate.

BANK MANAGEMENT SYSTEM

ACTIVITY DIAGRAM :-



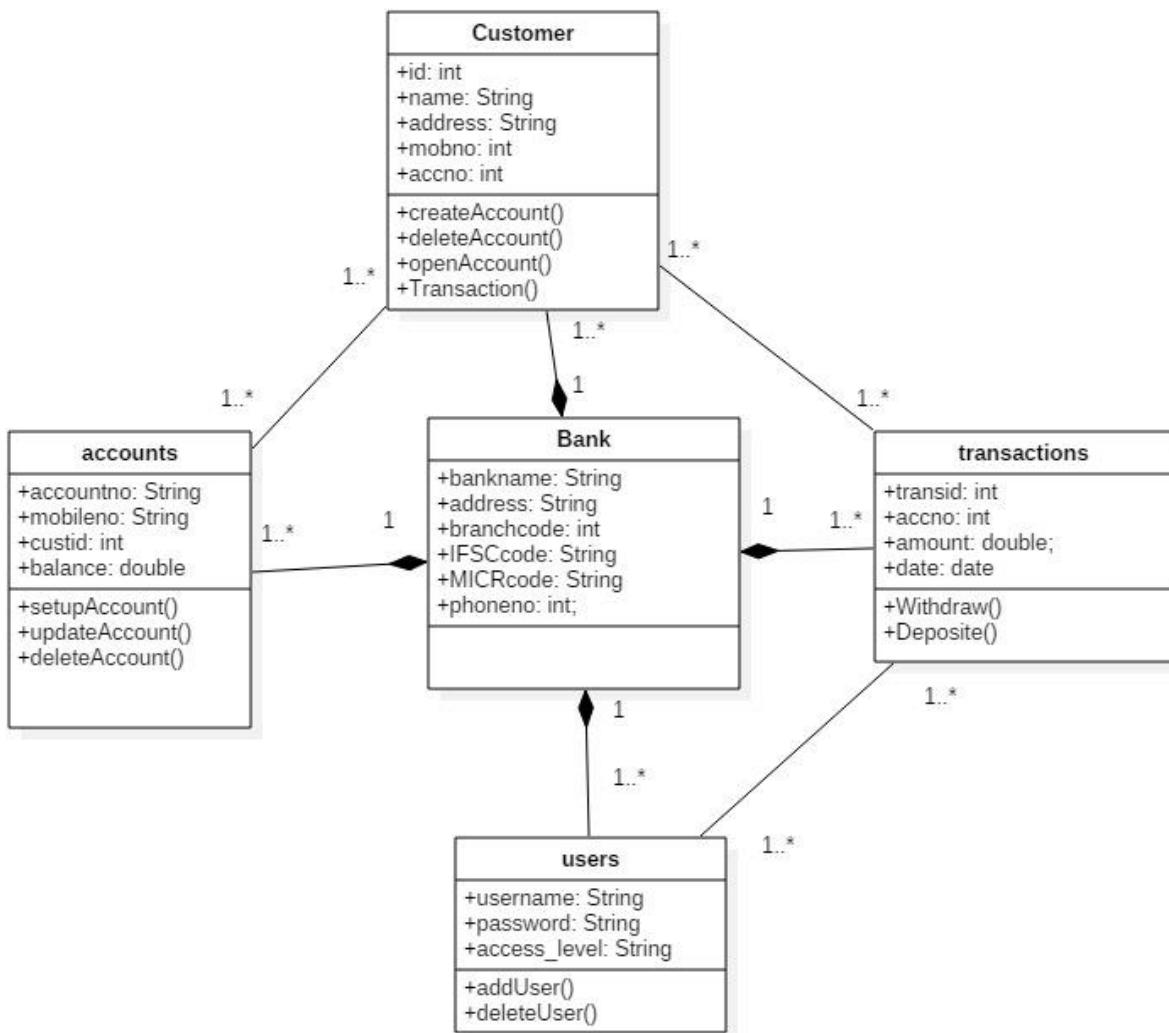
BANK MANAGEMENT SYSTEM

CLASS DIAGRAM

A Diagram is the unified modeling language (UML) is a type of static structure diagram that describes the structure of system by showing the system classes there attributes and the relationship between the classes. The purpose of class diagram is to identify and classify the object that will up the new system.

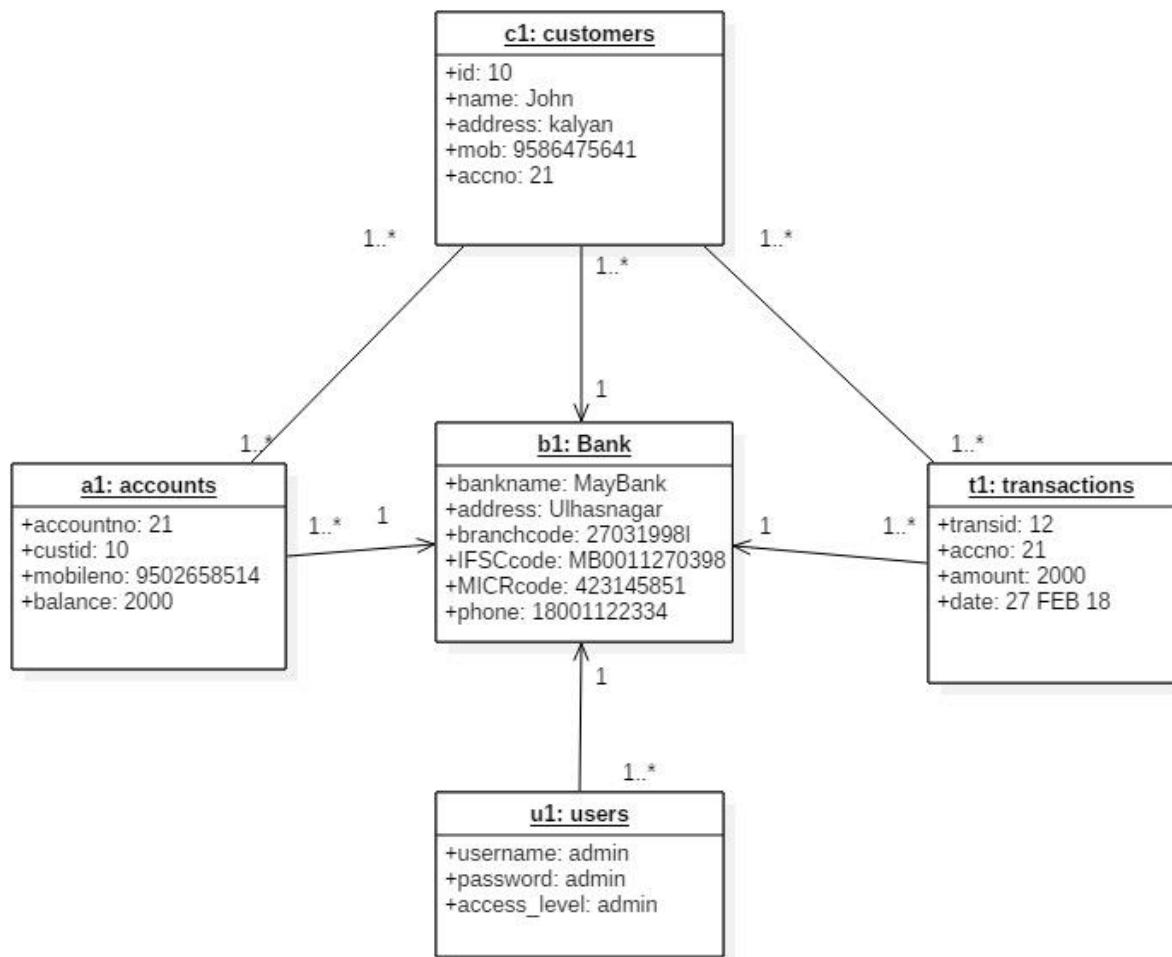
The purpose done is possible because object have method as well as attribute.

CLASS DIAGRAM :-



BANK MANAGEMENT SYSTEM

OBJECT DIAGRAM



SEQUENCE DIAGRAM

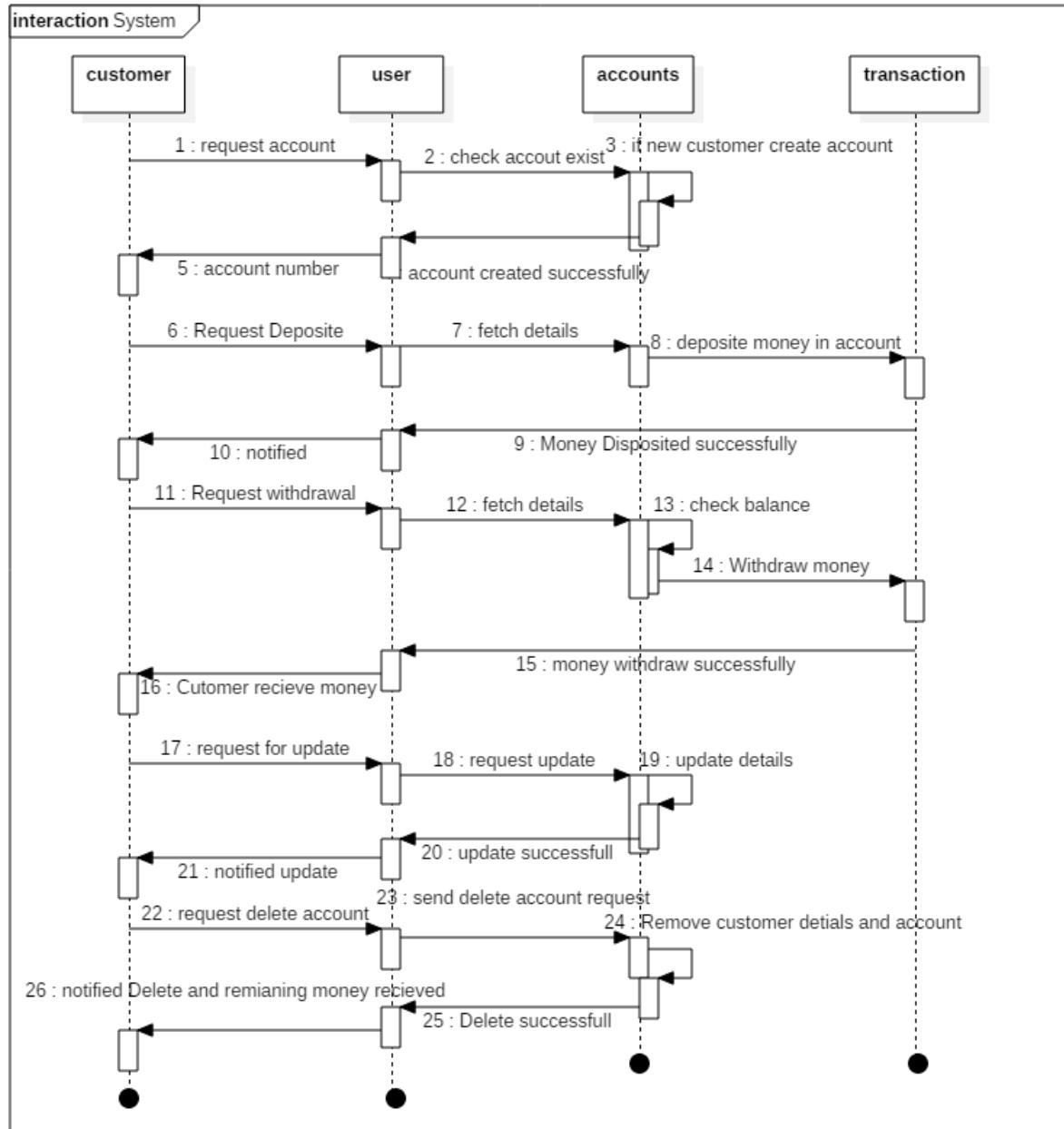
It is a kind of interaction diagram that shows how processes operate with one other and in what order sequence. Diagrams are sometimes called “Event Diagrams”, “Event Scenarios” and “timing Diagrams”

It consists of following components

- Actor:
 - An external entity that interacts with system.
- Path name:
 - Life line: They are vertical dashed lines that continue indicate to exit during the entire session of interaction indicates the objects presents overtime the life time extends to the bottom of the diagram.
- Activation:
 - It represents the time and object needs to compute task. A thin tall rectangle use to represent the time during which the object is active.

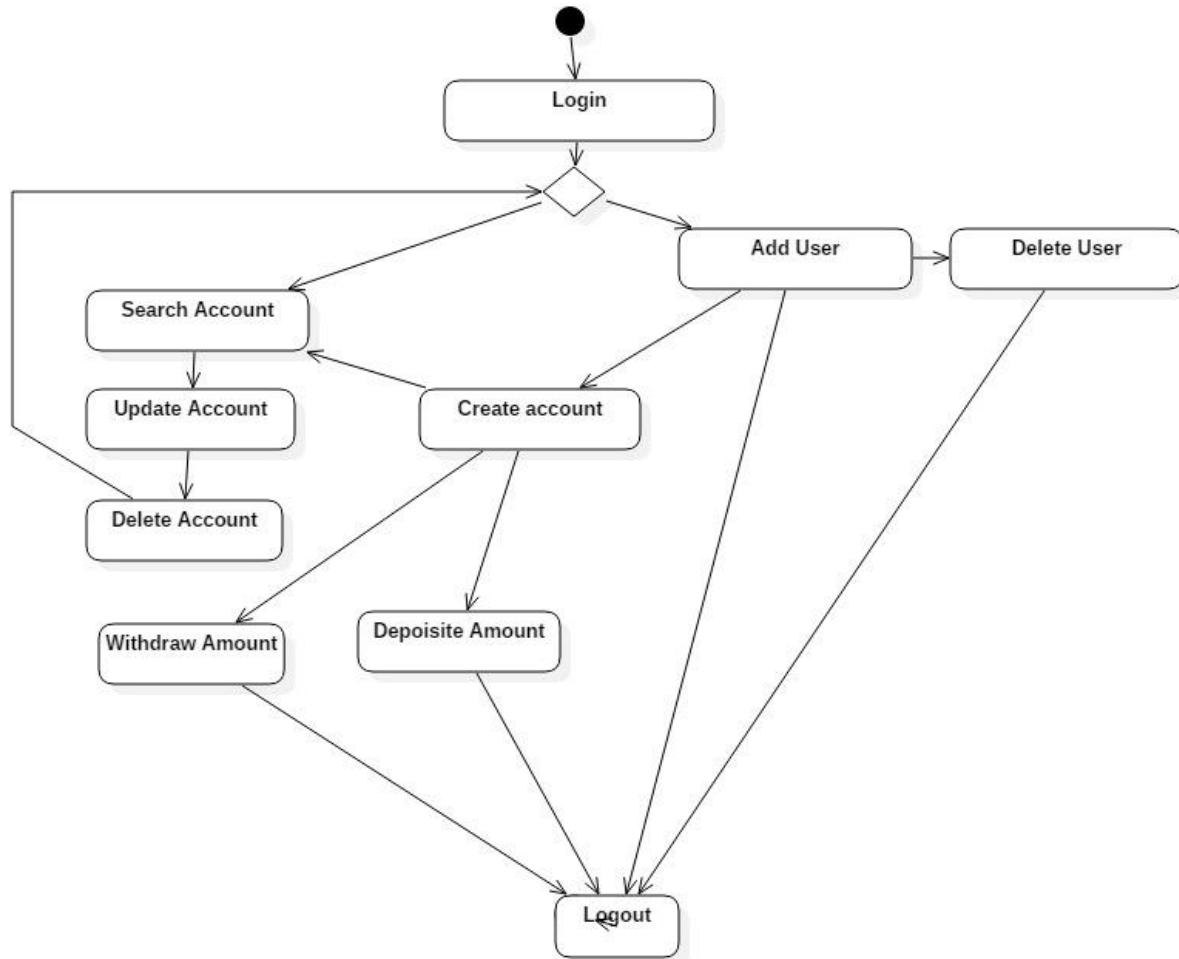
BANK MANAGEMENT SYSTEM

Sequence diagram :-



BANK MANAGEMENT SYSTEM

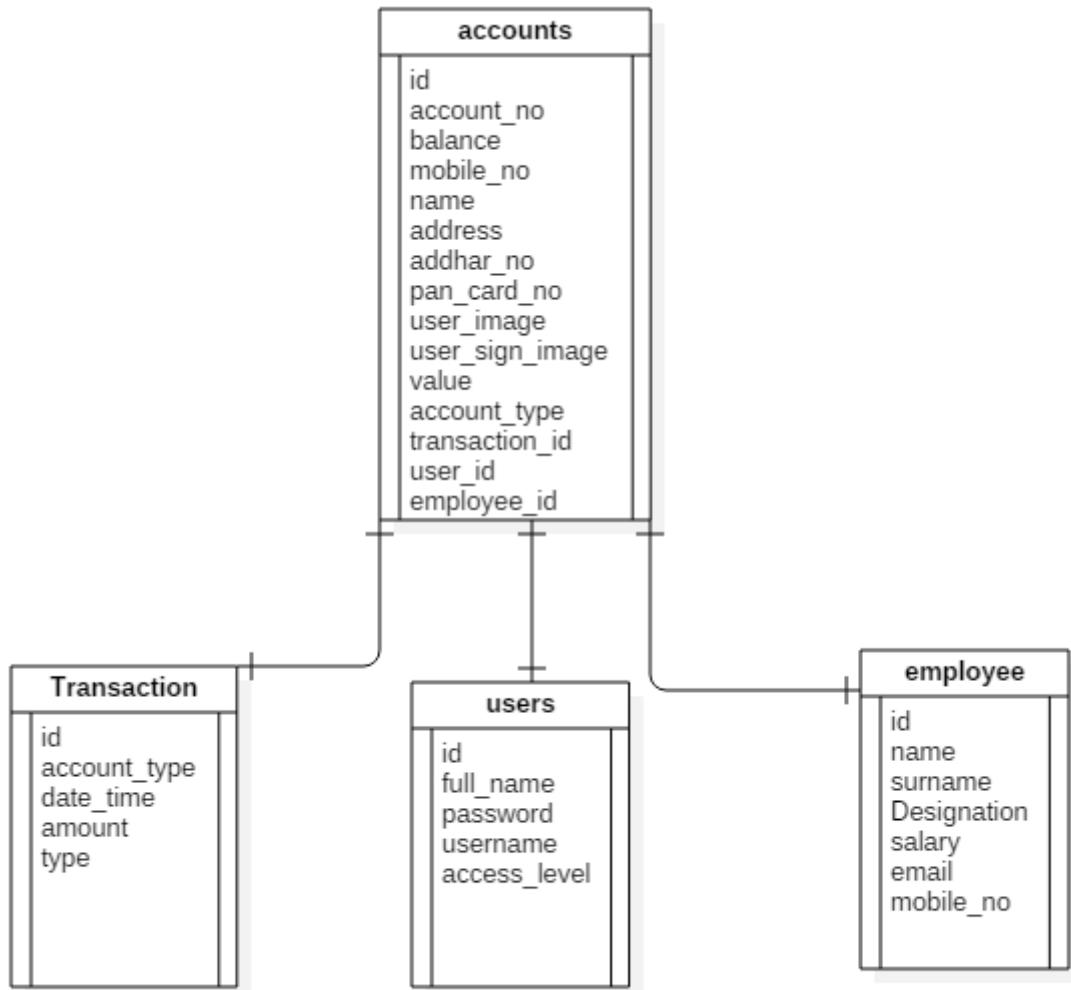
STATE DIAGRAM



SYSTEM DESIGN

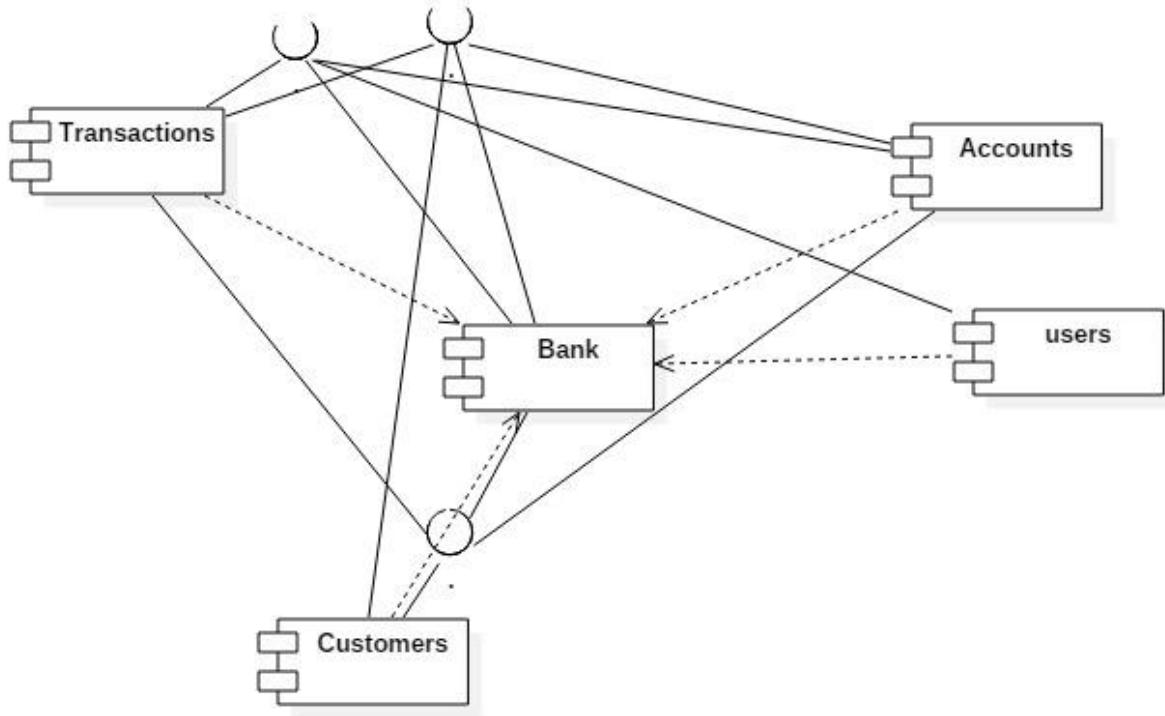
BANK MANAGEMENT SYSTEM

CONVERTING ERD TO TABLES



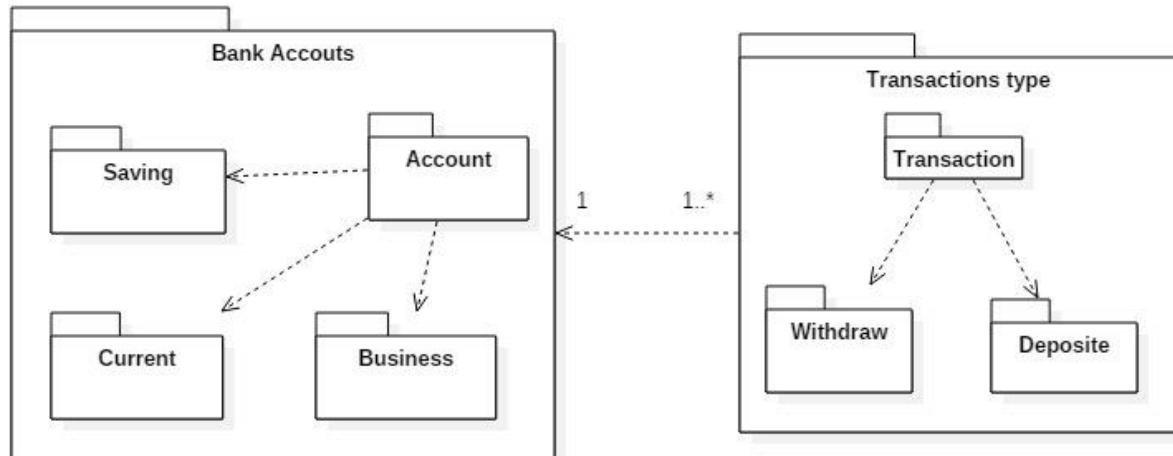
BANK MANAGEMENT SYSTEM

COMPONENT DIAGRAM



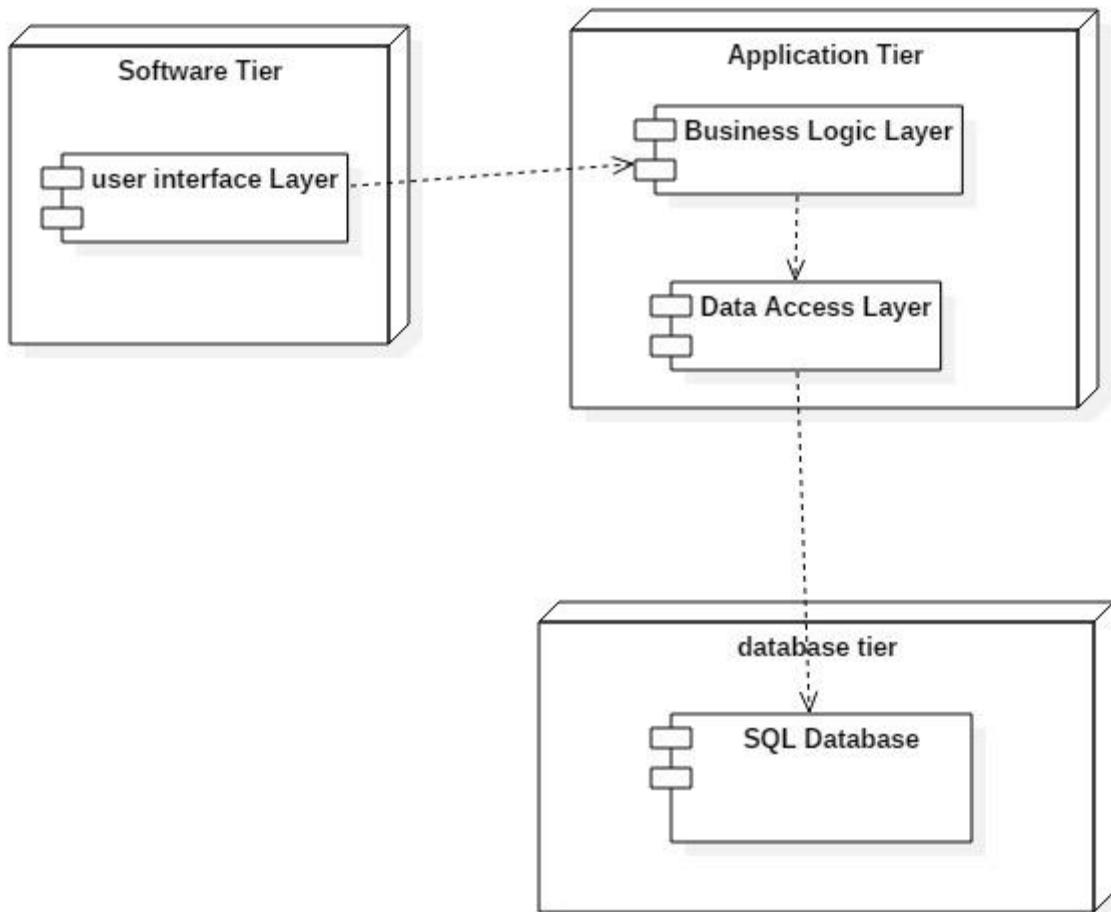
BANK MANAGEMENT SYSTEM

PACKAGE DIAGRAM



BANK MANAGEMENT SYSTEM

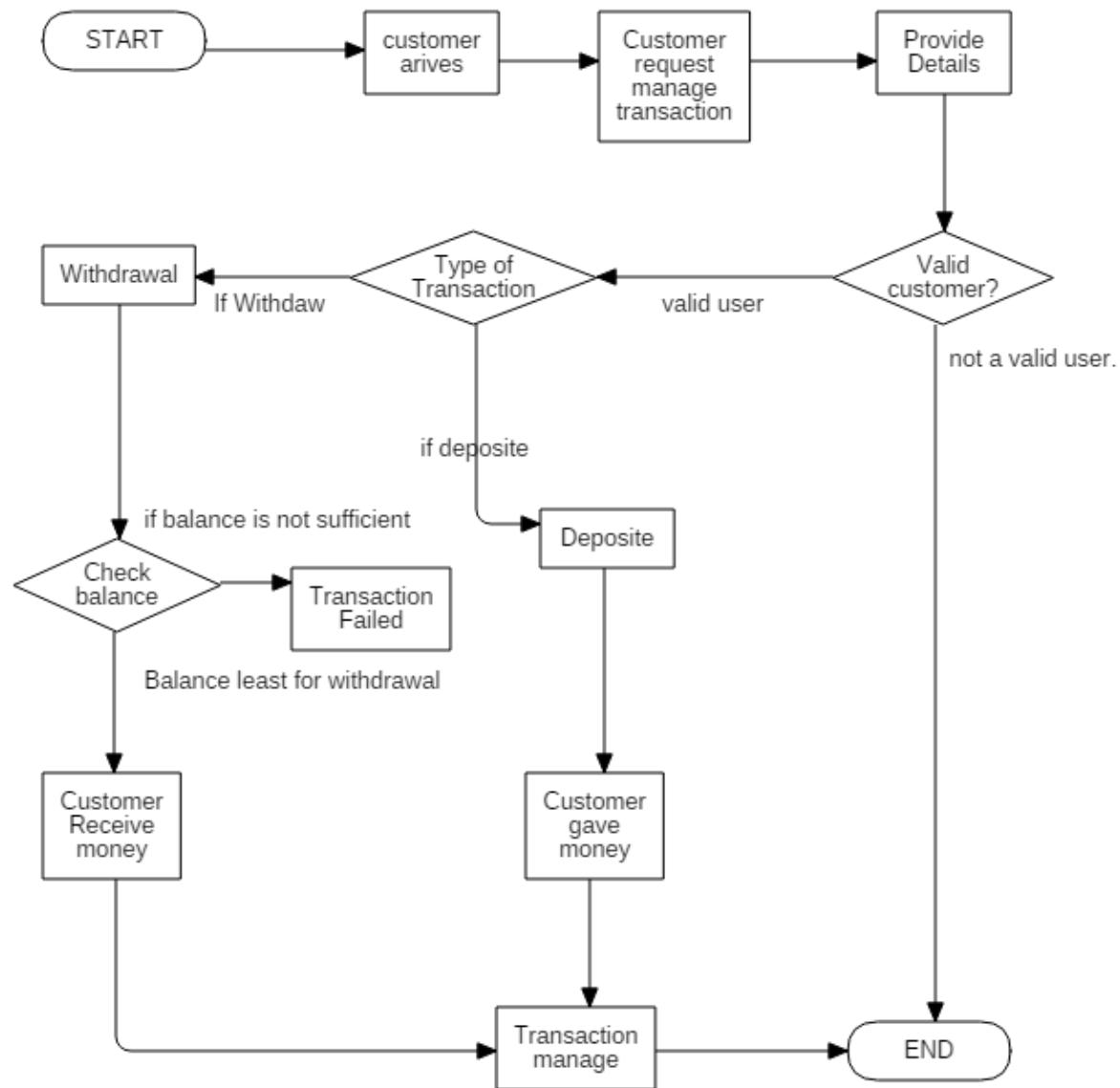
DEPLOYMENT DIAGRAM



BANK MANAGEMENT SYSTEM

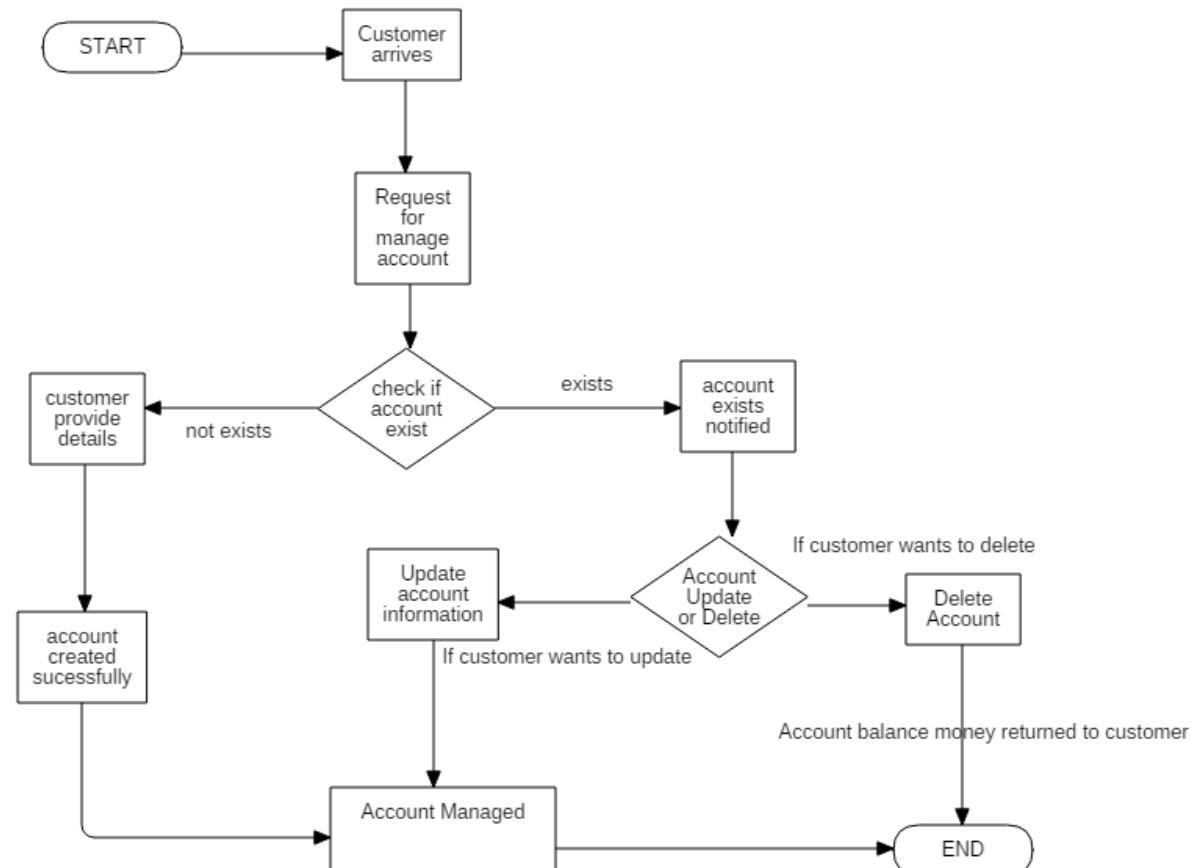
PROGRAM FLOW CHART

Flow chart for Transaction



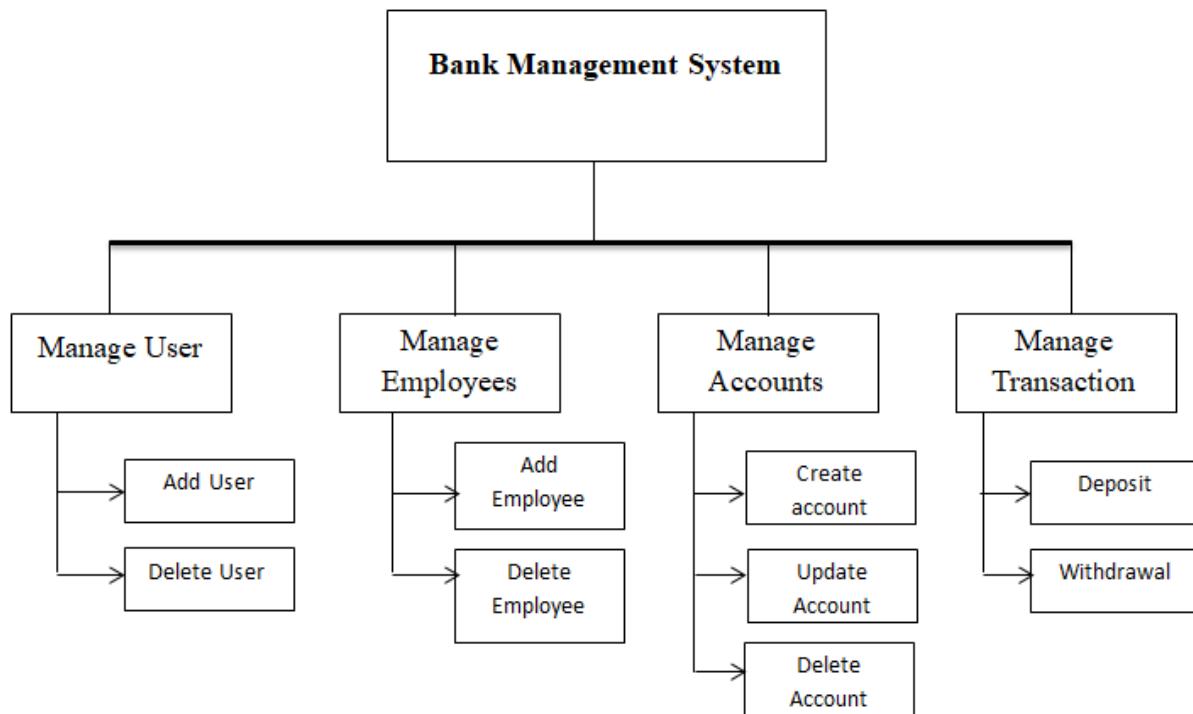
BANK MANAGEMENT SYSTEM

Flow chart for Accounts



BANK MANAGEMENT SYSTEM

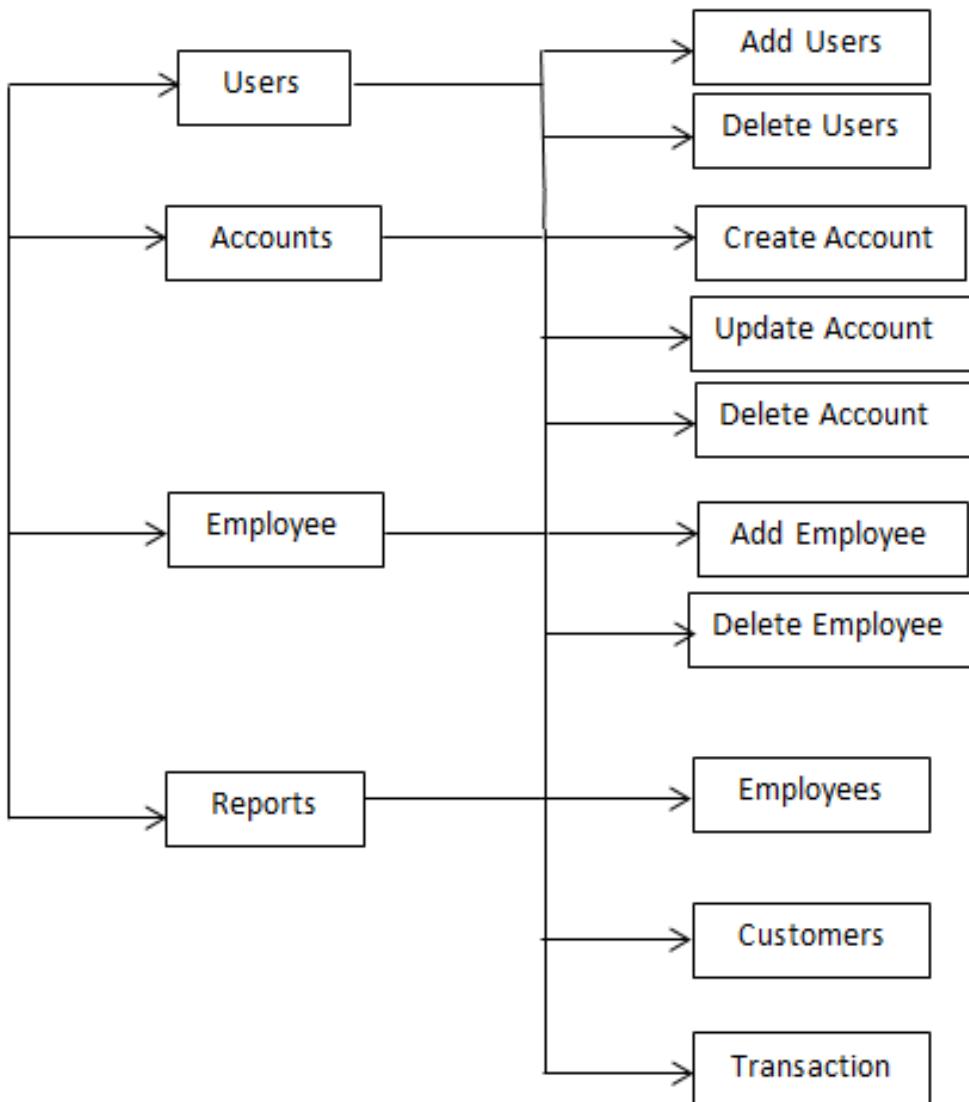
STRUCTURE CHART



PROJECT DOCUMENTATION

BANK MANAGEMENT SYSTEM

MENU TREE



BANK MANAGEMENT SYSTEM

LIST OF TABLES WITH ATTRIBUTES

Accounts : to store the accounts details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	account_no 	int(11)			No	None		AUTO_INCREMENT
2	full_name	varchar(255)	latin1_swedish_ci		No	None		
3	mobile_no 	varchar(12)	latin1_swedish_ci		No	None		
4	address	text	latin1_swedish_ci		No	None		
5	adhar_no 	varchar(12)	latin1_swedish_ci		No	None		
6	pan_card_no 	varchar(10)	latin1_swedish_ci		No	None		
7	gender	varchar(10)	latin1_swedish_ci		No	None		
8	account_type	varchar(20)	latin1_swedish_ci		No	None		
9	user_image_url	text	latin1_swedish_ci		No	None		
10	user_sign_image_url	text	latin1_swedish_ci		No	None		
11	balance	decimal(11,2)			No	0.00		
12	value	int(10)			No	None		

Employee: to store employee details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	None		AUTO_INCREMENT
2	name	varchar(255)	latin1_swedish_ci		No	None		
3	surname	varchar(255)	latin1_swedish_ci		No	None		
4	designation	varchar(255)	latin1_swedish_ci		No	None		
5	mobile_no	varchar(200)	latin1_swedish_ci		No	None		
6	email	varchar(255)	latin1_swedish_ci		No	None		
7	salary	varchar(255)	latin1_swedish_ci		No	None		

Transactions : to store transaction details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	None		AUTO_INCREMENT
2	account_no	int(11)			No	None		
3	type	varchar(255)	latin1_swedish_ci		No	None		
4	amount	varchar(255)	latin1_swedish_ci		No	None		
5	date_time	varchar(255)	latin1_swedish_ci		No	None		

Users : to store details od users

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	None		AUTO_INCREMENT
2	full_name	varchar(255)	latin1_swedish_ci		No	None		
3	employee_id	int(11)			No	None		
4	username	varchar(255)	latin1_swedish_ci		No	None		
5	password	text	latin1_swedish_ci		No	None		
6	access_level	varchar(50)	latin1_swedish_ci		No	None		

BANK MANAGEMENT SYSTEM

PROGRAM DESCRIPTION

List of Forms/reposts	Description
Authentication	This form is used to authenticate the user before entering the system.
Loading	This form is used to show the loading status graphically.
Home	This form is used to show the home page of the system.
CreateAccount	This form is used to create account for new customer.
UpdateAcount	This form is used to update the account information of account of the customer.
DeleteAccount	This form is used to delete the account of the customer permanently.
AddUser	This form is used to add the new user to access the system and delete the user from the system
Withdraw	This form is used to withdraw the money from the account of the customer.
Deposite	This form is used to deposite money into the customers account.
AddDeleteEmployee	It is used to add and delete employee from the system.
Employees	This form is used to generate the report of the employee woking on the bank.
Customers	This form is used to generate the report of the customers in the bank.
Transaction	This form is used to generate transaction report when ever user want it.
Help	This form is used to just to display the basic information of the bank.

BANK MANAGEMENT SYSTEM

PROGRAM SOURCE CODE

Authentication :-

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
public class Authentication extends javax.swing.JFrame {

    java.sql.Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    public Authentication() {
        initComponents();
        //get the database connection
        conn = DBConnect.connect();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        pnlAuthentication = new javax.swing.JPanel();
        lblUsername = new javax.swing.JLabel();
        lblPassword = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        txtUsername = new javax.swing.JTextField();
        btnLogin = new javax.swing.JButton();
        btnReset = new javax.swing.JButton();
        btnClose = new javax.swing.JButton();
        txtPassword = new javax.swing.JPasswordField();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
        lblUsername.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        lblUsername.setForeground(new java.awt.Color(153, 0, 0));
    }
}
```

BANK MANAGEMENT SYSTEM

```
lblUsername.setText("Username:");
lblPassword.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
lblPassword.setForeground(new java.awt.Color(153, 0, 0));
lblPassword.setText("Password:");
txtUsername.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtUsernameActionPerformed(evt);
    }
});

btnLogin.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnLogin.setForeground(new java.awt.Color(153, 0, 0));
btnLogin.setText("Login");
btnLogin.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLoginActionPerformed(evt);
    }
});

btnReset.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnReset.setForeground(new java.awt.Color(153, 0, 0));
btnReset.setText("Clear");
btnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResetActionPerformed(evt);
    }
});

btnClose.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnClose.setForeground(new java.awt.Color(153, 0, 0));
btnClose.setText("Close");
btnClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCloseActionPerformed(evt);
    }
});

}

}

setSize(new java.awt.Dimension(504, 335));
setLocationRelativeTo(null);
}// </editor-fold>

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

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

BANK MANAGEMENT SYSTEM

```
        System.exit(0);
    }

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    txtUsername.setText("");
    txtPassword.setText("");
}

private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        PasswordAuthentication passwordAuthentication = new PasswordAuthentication();

        String username = txtUsername.getText();
        String pass = txtPassword.getText();

        String queryString = "SELECT * FROM users WHERE username = ?";
        pst = conn.prepareStatement(queryString);
        pst.setString(1, username);

        rs = pst.executeQuery();

        //check if the password is correct
        if (rs.next()) {
            String dbPass = rs.getString("password");
            if (passwordAuthentication.authenticate(pass.toCharArray(), dbPass)) {
                //save the username and id locally
                saveUserDetails(rs.getString("username"), rs.getString("employee_id"));

                //password is correct let the user in
                dispose(); //close the current window
                Loading obj = new Loading();
                obj.setVisible(true);
                obj.setupLoading();
            } else {
                //password is not correct
                JOptionPane.showMessageDialog(this, "Incorrect password!!", "Error",
                JOptionPane.ERROR_MESSAGE);
            }
        } else {
            //username is not correct or username does not exist
            JOptionPane.showMessageDialog(this, "Incorrect username!!", "Error",
            JOptionPane.ERROR_MESSAGE);
        }
    } catch (SQLException ex) {
```

BANK MANAGEMENT SYSTEM

```
        ex.printStackTrace();
    }

}

public static void main(String args[]) throws InstantiationException {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
            UIManager.setLookAndFeel("com.jtattoo.plaf.smart.SmartLookAndFeel");
        }
    } catch (ClassNotFoundException ex) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Authentication().setVisible(true);
            }
        });
    }
}

// Variables declaration - do not modify
private javax.swing.JButton btnClose;
private javax.swing.JButton btnLogin;
private javax.swing.JButton btnReset;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel lblPassword;
private javax.swing.JLabel lblUsername;
private javax.swing.JPanel pnlAuthentication;
private javax.swing.JPasswordField txtPassword;
private javax.swing.JTextField txtUsername;
// End of variables declaration

private void saveUserDetails(String username, String userID) {

    FileWriter fw = null;
    try {
        File file = new File("data.txt");
        fw = new FileWriter(file);
        fw.write(username+"/"+userID);
        fw.flush();
    }
}
```

BANK MANAGEMENT SYSTEM

```
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            try {
                if(fw != null){
                    fw.close();
                }
            } catch (IOException ex) {
                Logger.getLogger(Authentication.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
```

Loading:-

```
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JProgressBar;

public class Loading extends javax.swing.JFrame implements Runnable {

    int progress = 0;
    Thread progressThread;

    public Loading() {
        super("Loading...");
        initComponents();
        progressThread = new Thread((Runnable)this);
    }

    public void setupLoading(){
        setVisible(true);
        progressThread.start();
    }

    public void run(){
        try {
            for(int i = 1;i <= 200;i++){
                i++;
                progress++;
            }
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
int max = progressBar.getMaximum();

int currentVal = progressBar.getValue();

if(currentVal < max){
    progressBar.setValue(progress + 1);

}

else{
    setVisible(false);
    new Home().setVisible(true);
}

Thread.sleep(50);

}

} catch (Exception e) {
}

}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JProgressBar progressBar;
// End of variables declaration

private void Processing() {
    throw new UnsupportedOperationException("Not yet implemented");
}

}
```

BANK MANAGEMENT SYSTEM

Home :-

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.sql.*;
import javax.swing.JOptionPane;

public class Home extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;

    public Home() {
        initComponents();
        conn = DBConnect.connect();
        getUserDetails();

    }

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

        setVisible(false);
        Authentication an = new Authentication();
        an.setVisible(true);
    }

    private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

    private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
        new CreateAccount().setVisible(true);
        setVisible(false);
    }

    private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
        new Search().setVisible(true);
        dispose();
    }

    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        String adm=lblUID.getText();

        try {
```

BANK MANAGEMENT SYSTEM

```
String query="Select access_level from users where id= ?";
PreparedStatement pst1=conn.prepareStatement(query);
pst1.setString(1,adm);
rs=pst1.executeQuery();

if (rs.next() && rs.getString("access_level").equals("admin")) {
    setVisible(false);
    new DeleteSearch().setVisible(true);
}
else
{
    JOptionPane.showMessageDialog(this, "Sorry, Only Manager can access this!!!", "Error",
JOptionPane.ERROR_MESSAGE);
}
} catch (Exception e) {
}

private void btnDepositeAmountActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Deposite().setVisible(true);
}

private void btnWithdrawActionPerformed(java.awt.event.ActionEvent evt) {
    new Withdraw().setVisible(true);
    setVisible(false);
}

private void btnAddUserActionPerformed(java.awt.event.ActionEvent evt) {
    String adm=lblUID.getText();

    try {
        String query="Select access_level from users where id= ?";
        PreparedStatement pst1=conn.prepareStatement(query);
        pst1.setString(1,adm);
        rs=pst1.executeQuery();

        if (rs.next() && rs.getString("access_level").equals("admin")) {
            new Users().setVisible(true);
            setVisible(false);
        }
    else
    {
        JOptionPane.showMessageDialog(this, "Sorry, Only Manager can access this!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
```

BANK MANAGEMENT SYSTEM

```
        }
    } catch (Exception e) {
    }
}

private void jmTranActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Transaction().setVisible(true);
}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void jmEmplActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Employees().setVisible(true);
}

private void jmCustomersActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Customers().setVisible(true);
}

private void jmAbtActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Help().setVisible(true);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Home().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnAddUser;
private javax.swing.JButton btnClose;
private javax.swing.JButton btnCreate;
private javax.swing.JButton btnDelete;
private javax.swing.JButton btnDepositeAmount;
private javax.swing.JButton btnLogout;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JButton btnUpdate;
private javax.swing.JButton btnWithdraw;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JMenuItem jmAbt;
private javax.swing.JMenuItem jmCustomers;
private javax.swing.JMenuItem jmEmpl;
private javax.swing.JMenuItem jmTran;
private javax.swing.JLabel lblUID;
private javax.swing.JLabel lblUN;
private javax.swing.JLabel lblUserImage;
private javax.swing.JPanel pnlHome;
// End of variables declaration

private void getUserDetails() {
    try {

        BufferedReader br = new BufferedReader(new FileReader("data.txt"));
        String data;

        data = br.readLine();

        String details[] = data.split("/");
        lblUN.setText(details[0]);
        lblUID.setText(details[1]);

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

BANK MANAGEMENT SYSTEM

Create Account:

```
import java.awt.dnd.DropTarget;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileNameExtensionFilter;

public class CreateAccount extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    Validation validator;
    File mUserImageFile;
    File mUserSignatureFile;

    DragListener dragListener1;
    DragListener dragListener2;

    public CreateAccount() {
        initComponents();
        //create connection with database
        conn = DBConnect.connect();
        generateAccountId();
        validator = new Validation();

        //init the DragListner class for user profile picture
        dragListener1 = new DragListener(lblPhoto);

        //set the target componet for dropping item
        new DropTarget(lblPhoto, dragListener1);

        //init the DragListner class for user signature picture
    }
}
```

BANK MANAGEMENT SYSTEM

```
dragListener2 = new DragListener(lblSignPhoto);
//set the target component for dropping item
new DropTarget(lblSignPhoto, dragListener2);
}

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

private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
    String name = txtName.getText();
    String mobileNo = txtMobNo.getText();
    String address = txtaAdd.getText();
    String adharNo = txtAdhar.getText();
    String panNo = txtPan.getText();
    if (validator.isEmpty(txtName, txtMobNo, txtAdhar, txtPan) > 0) {
        JOptionPane.showMessageDialog(this, "All the fields must not be empty!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    } else if (lblPhoto.getIcon() == null) {

        JOptionPane.showMessageDialog(this, "Please upload a image for user!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    } else if (address.equals("") || address == null) {
        JOptionPane.showMessageDialog(this, "All the fields must not be empty!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    } else if (!validator.isNumber(mobileNo) || mobileNo.length() < 10) {
        JOptionPane.showMessageDialog(this, "Incorrect Mobile number!!!", "Error",
JOptionPane.ERROR_MESSAGE);

    } else if (!validator.isAdharNumber(adharNo)) {
        JOptionPane.showMessageDialog(this, "Incorrect Adhar Card Number!!!", "Error",
JOptionPane.ERROR_MESSAGE);

    } else if (!validator.isPanNumber(panNo)) {
        JOptionPane.showMessageDialog(this, "Incorrect Pan Card Number!!!", "Error",
JOptionPane.ERROR_MESSAGE);

    } else if (validator.characters(name) == validator.HAS_NON_CHARACTERS) {
        JOptionPane.showMessageDialog(this, "Name Cannot Contains Number!!!", "Error",
JOptionPane.ERROR_MESSAGE);

    } else {
        setUpAccount();
    }
}
```

BANK MANAGEMENT SYSTEM

```
        }

    }

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    txtAccNo.setText("");
    txtAdhar.setText("");
    txtAmount.setText("");
    txtMobNo.setText("");
    txtName.setText("");
    txtPan.setText("");
    txtaAdd.setText("");
    lblPhoto.setText("");
    lblSignPhoto.setText("");
}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);
}

private void btnUploadUserImageActionPerformed(java.awt.event.ActionEvent evt) {
    mUserImageFile = uploadImage(lblPhoto);
}

private void btnUploadSignatureActionPerformed(java.awt.event.ActionEvent evt) {
    mUserSigantureFile = uploadImage(lblSignPhoto);
}

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

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

BANK MANAGEMENT SYSTEM

```
        new CreateAccount().setVisible(true);
    }
});

}

// Variables declaration - do not modify
private javax.swing.JButton btnClose;
private javax.swing.JButton btnCreate;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnReset;
private javax.swing.JButton btnUploadSignature;
private javax.swing.JButton btnUploadUserImage;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JList jlAccType;
private javax.swing.JLabel lblPhoto;
private javax.swing.JLabel lblSignPhoto;
private javax.swing.JRadioButton rbdFemale;
private javax.swing.JRadioButton rbdMale;
private javax.swing.JTextField txtAccNo;
private javax.swing.JTextField txtAdhar;
private javax.swing.JTextField txtAmount;
private javax.swing.JTextField txtMobNo;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtPan;
private javax.swing.JTextArea txtaAdd;
// End of variables declaration

private void setUpAccount() {
    //get the account details
    String name = txtName.getText();
```

BANK MANAGEMENT SYSTEM

```
String mobileNo = txtMobNo.getText();
String address = txtaAdd.getText();
String adharNo = txtAdhar.getText();
String panNo = txtPan.getText();
String gender = rbdMale.isSelected() ? "Male" : "Female";
String accountType = jlAccType.getSelectedValue().toString();
int balance = Integer.parseInt(txtAmount.getText());
int value = 0;

//validate all the fields
Validation Validator = new Validation();

if (balance < 1000) {
    JOptionPane.showMessageDialog(this, "If you wants to create account the minimum
1000 rupies you have to deposite!!!", "Error", JOptionPane.ERROR_MESSAGE);
} else {
    try {

        if (dragListener1.UserProfileImageFile != null) {
            mUserImageFile = dragListener1.UserProfileImageFile;
        }

        if (dragListener2.UserSignImageFile != null) {
            mUserSigantureFile = dragListener2.UserSignImageFile;
        }

        //get the image and signature filePth to store in db
        String imagePath = saveImage(mUserImageFile);
        String signatureImagePath = saveImage(mUserSigantureFile);

        //create account now
        String query = "INSERT INTO accounts(full_name, mobile_no, address, adhar_no,
pan_card_no, gender, account_type, user_image_url, user_sign_image_url,value,balance)
VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

        pst = conn.prepareStatement(query);

        //set the values
        pst.setString(1, name);
        pst.setString(2, mobileNo);
        pst.setString(3, address);
        pst.setString(4, adharNo);
        pst.setString(5, panNo);
        pst.setString(6, gender);
        pst.setString(7, accountType);
        pst.setString(8, imagePath);
        pst.setString(9, signatureImagePath);
```

BANK MANAGEMENT SYSTEM

```
        pst.setString(10, String.valueOf(value));
        pst.setString(11, String.valueOf(balance));

        if (pst.executeUpdate() == 1) {
            int accno = Integer.parseInt(txtAccNo.getText());
            JOptionPane.showMessageDialog(this, "Account created Successfully...\n Your
Account number is: " + accno, "Successfull", JOptionPane.INFORMATION_MESSAGE);

            setVisible(false);
            new Home().setVisible(true);
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(this, ex.toString(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

private void generateAccountId() {
    int currentId = 0;
    try {
        String query = "SELECT account_no FROM accounts";
        Statement st = conn.createStatement();

        rs = st.executeQuery(query);
        while (rs.next()) {
            currentId = rs.getInt("account_no");//get the last id presents in accounts
        }

        //now increament the currentId to 1 to generate new Id
        currentId++;
        txtAccNo.setText(currentId + "");

    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

private File uploadImage(JLabel label) {
    File fileToStore = null;
    JFileChooser ch = new JFileChooser();
    FileNameExtensionFilter filter = new FileNameExtensionFilter("Image File", new
String[]{"jpg", "jpeg", "png"});
    ch.setFileFilter(filter);
```

BANK MANAGEMENT SYSTEM

```
ch.addChoosableFileFilter(filter);

int returnValue = ch.showOpenDialog(null);

if (returnValue == JFileChooser.APPROVE_OPTION) {
    fileToStore = ch.getSelectedFile();

    try {
        BufferedImage userBufferedImage = ImageIO.read(fileToStore); //reads the image
        from file and buffered to imageBuffered

        //resize the image according to the size of the label
        BufferedImage resizedImage = CustomImageSize.resize(userBufferedImage,
        label.getWidth(), label.getHeight());
        //Jlabel doesn't take bufferedImage image as argument so convert it into ImageIcon
        ImageIcon imageIcon = new ImageIcon(resizedImage);

        label.setIcon(imageIcon); //set the icon

    } catch (IOException ex) {
        JOptionPane.showMessageDialog(this, "Unable to read the image file!!!", "Error",
        JOptionPane.ERROR_MESSAGE);
        ex.printStackTrace();
    }
}

return fileToStore;
}

private String saveImage(File file) {
    String filePath = null;
    //this method will store the images in /storedImages/ folder
    if (file != null) {
        try {
            //first read image
            BufferedImage bufferedImage = ImageIO.read(file);

            //save the image locally
            File imageFile = new File("./storedImages/" + "acc_no-" + txtAccNo.getText() + "-
image-" + System.currentTimeMillis() + ".jpeg");
            filePath = "./storedImages/" + imageFile.getName();
            ImageIO.write(bufferedImage, "jpg", imageFile);

        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
        }
    }
    return filePath;
}

}
```

Search:-

```
import java.awt.dnd.DropTarget;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileNameExtensionFilter;
```

```
=
public class Search extends javax.swing.JFrame {
```

```
    File mUserImageFile;
    File mUserSignFile;
```

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

```
    public Search() {
        initComponents();
        conn = DBConnect.connect();

        //init the DragListner class for user profile picture
        DragListener dragListener1 = new DragListener(lblUserSignImage);
        //set the target componet for dropping item
        new DropTarget(lblUserSignImage, dragListener1);
    }
```

BANK MANAGEMENT SYSTEM

```
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    pnlSearch = new javax.swing.JPanel();
    lblPhoto = new javax.swing.JLabel();
    lblAccNo = new javax.swing.JLabel();
    txtAccNo = new javax.swing.JTextField();
    lblName = new javax.swing.JLabel();
    txtHolderName = new javax.swing.JTextField();
    lblName1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    txtAddreass = new javax.swing.JTextArea();
    txtAccNo1 = new javax.swing.JLabel();
    txtAccNo2 = new javax.swing.JLabel();
    txtMobileNo = new javax.swing.JTextField();
    txtAdharNo = new javax.swing.JTextField();
    txtAccNo3 = new javax.swing.JLabel();
    txtPanNo = new javax.swing.JTextField();
    txtAccNo4 = new javax.swing.JLabel();
    txtAccType = new javax.swing.JTextField();
    txtAccNo5 = new javax.swing.JLabel();
    txtGender = new javax.swing.JTextField();
    btnFinish = new javax.swing.JButton();
    btnHome = new javax.swing.JButton();
    btnClose = new javax.swing.JButton();
    btnLogout = new javax.swing.JButton();
    txtAccNo6 = new javax.swing.JLabel();
    txtAccBalance = new javax.swing.JTextField();
    txtAccNo7 = new javax.swing.JLabel();
    lblUserImage = new javax.swing.JLabel();
    lblUserSignImage = new javax.swing.JLabel();
    lblSearch = new javax.swing.JButton();
    btnChooseSign = new javax.swing.JButton();
    btnReset = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();

    txtAccNo.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtAccNoActionPerformed(evt);
        }
    });

    txtHolderName.setEditable(false);

}
```

BANK MANAGEMENT SYSTEM

```
txtHolderName.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHolderNameActionPerformed(evt);
    }
});

txtAddreass.setColumns(20);
txtAddreass.setRows(5);
jScrollPane1.setViewportView(txtAddreass);

txtMobileNo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtMobileNoActionPerformed(evt);
    }
});

txtGender.setEditable(false);
txtGender.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtGenderActionPerformed(evt);
    }
});

btnFinish.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnFinishActionPerformed(evt);
    }
});

btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHomeActionPerformed(evt);
    }
});

btnClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCloseActionPerformed(evt);
    }
});

btnLogout.addActionListener(new java.awt.event.ActionListener() {
```

BANK MANAGEMENT SYSTEM

```
public void actionPerformed(java.awt.event.ActionEvent evt) {
    btnLogoutActionPerformed(evt);
}

});
```

```
txtAccBalance.setEditable(false);
txtAccBalance.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAccBalanceActionPerformed(evt);
    }
});
```

```
lblSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        lblSearchActionPerformed(evt);
    }
});
```

```
btnChooseSign.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnChooseSignActionPerformed(evt);
    }
});
```

```
btnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResetActionPerformed(evt);
    }
});
```

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

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

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

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

BANK MANAGEMENT SYSTEM

```
}

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);

}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}

private void btnFinishActionPerformed(java.awt.event.ActionEvent evt) {
    updateDetails();
}

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

private void lblSearchActionPerformed(java.awt.event.ActionEvent evt) {
    System.out.println(txtAccNo.getText());

    if (!txtAccNo.getText().equals("")) {
        try {
            String query = "SELECT * FROM accounts WHERE account_no = ? ";
            pst = conn.prepareStatement(query);
            pst.setString(1, txtAccNo.getText());

            rs = pst.executeQuery();
            int value;

            if (rs.next() && rs.getString("value").equals("0")) {
                txtHolderName.setText(rs.getString("full_name"));
                txtMobileNo.setText(rs.getString("mobile_no"));
                txtAddress.setText(rs.getString("address"));


```

BANK MANAGEMENT SYSTEM

```
txtAdharNo.setText(rs.getString("adhar_no"));
txtPanNo.setText(rs.getString("pan_card_no"));
txtGender.setText(rs.getString("gender"));
txtAccType.setText(rs.getString("account_type"));
txtAccBalance.setText(rs.getString("balance"));

mUserImageFile = new File(rs.getString("user_image_url"));
mUserSignFile = new File(rs.getString("user_sign_image_url"));

//display the images in the JLabels
displayImage(lblUserImage, mUserImageFile);
displayImage(lblUserSignImage, mUserSignFile);

} else {
    //account does not exist
    JOptionPane.showMessageDialog(this, "Account not exist or invalid account
number!!!", "Error", JOptionPane.ERROR_MESSAGE);
}

} catch (Exception e) {
    e.printStackTrace();
}
} else {
    JOptionPane.showMessageDialog(this, "Please enter account number!!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

private void btnChooseSignActionPerformed(java.awt.event.ActionEvent evt) {
    mUserSignFile = uploadImage(lblUserSignImage);
}

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    txtAccNo.setText("");
    txtAccType.setText("");
    txtGender.setText("");
    txtAdharNo.setText("");
    txtAccBalance.setText("");
    txtMobileNo.setText("");
    txtHolderName.setText("");
    txtPanNo.setText("");
    txtAddreass.setText("");
}

 ImageIcon imageIcon = new ImageIcon();
```

BANK MANAGEMENT SYSTEM

```
//set the image
lblUserImage.setIcon(imageIcon);
ImageIcon imgcon = new ImageIcon();
lblUserSignImage.setIcon(imgcon);
}

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

}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Search().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnChooseSign;
private javax.swing.JButton btnClose;
private javax.swing.JButton btnFinish;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnReset;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel lblAccNo;
private javax.swing.JLabel lblName;
private javax.swing.JLabel lblName1;
private javax.swing.JLabel lblPhoto;
private javax.swing.JButton lblSearch;
private javax.swing.JLabel lblUserImage;
private javax.swing.JLabel lblUserSignImage;
private javax.swing.JPanel pnlSearch;
private javax.swing.JTextField txtAccBalance;
private javax.swing.JTextField txtAccNo;
private javax.swing.JLabel txtAccNo1;
private javax.swing.JLabel txtAccNo2;
private javax.swing.JLabel txtAccNo3;
private javax.swing.JLabel txtAccNo4;
private javax.swing.JLabel txtAccNo5;
private javax.swing.JLabel txtAccNo6;
private javax.swing.JLabel txtAccNo7;
private javax.swing.JTextField txtAccType;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JTextArea txtAddreass;
private javax.swing.JTextField txtAdharNo;
private javax.swing.JTextField txtGender;
private javax.swing.JTextField txtHolderName;
private javax.swing.JTextField txtMobileNo;
private javax.swing.JTextField txtPanNo;
// End of variables declaration

private void displayImage(JLabel label, File imageFile) {
    try {
        BufferedImage bufferedImage = ImageIO.read(imageFile); //read the image
        //resize the the according to the size of JLabel
        BufferedImage resizedImage = CustomImageSize.resize(bufferedImage,
label.getWidth(), label.getHeight());
        //create ImageIcon
        ImageIcon imageIcon = new ImageIcon(resizedImage);
        //set the image
        label.setIcon(imageIcon);

    } catch (Exception e) {
    }
}

private File uploadImage(JLabel label) {
    File fileToStore = null;
    JFileChooser ch = new JFileChooser();
    FileNameExtensionFilter filter = new FileNameExtensionFilter("Image File", new
String[]{"jpg", "jpeg", "png"});
    ch.setFileFilter(filter);
    ch.addChoosableFileFilter(filter);

    int returnValue = ch.showOpenDialog(null);

    if (returnValue == JFileChooser.APPROVE_OPTION) {
        fileToStore = ch.getSelectedFile();

        try {
            BufferedImage userBufferedImage = ImageIO.read(fileToStore); //reads the image
from file and buffered to imageBuffered

            //resize the image acording to the size of the label
            BufferedImage resizedImage = CustomImageSize.resize(userBufferedImage,
label.getWidth(), label.getHeight());
            //Jlabel doesnt take bufferedImage image as argument so convert it into ImageIcon
            ImageIcon imageIcon = new ImageIcon(resizedImage);
            label.setIcon(imageIcon); //set the icon
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(this, "Unable to read the image file!!!", "Error",
JOptionPane.ERROR_MESSAGE);
            ex.printStackTrace();
        }
    }
    return fileToStore;
}

private void updateDetails() {
    String name = txtHolderName.getText();
    String accNo = txtAccNo.getText();
    String gender = txtGender.getText();
    String mobilenNo = txtMobileNo.getText();
    String addr = txtAddreass.getText();
    //String userImagePath = "./storedImages/" + mUserImageFile.getName();
    String userSignImagePath = saveImage(mUserSignFile);

    Validation validator = new Validation();
    //check if the fields are empty
    if (validator.isEmpty(txtAccNo, txtHolderName, txtMobileNo) == 0) {
        if (!validator.isNumber(mobilenNo) || mobilenNo.length() == 10) {

            try {
                String query = "UPDATE accounts SET full_name = ?, mobile_no = ?, address = ?,
user_sign_image_url = ? WHERE account_no = ?";
                pst = conn.prepareStatement(query);
                pst.setString(1, name);
                pst.setString(2, mobilenNo);
                pst.setString(3, addr);
                //pst.setString(4, userImagePath);
                pst.setString(4, userSignImagePath);
                pst.setString(5, accNo);

                pst.executeUpdate();
                JOptionPane.showMessageDialog(this, "Account details updated
successfully... ", "Success", JOptionPane.INFORMATION_MESSAGE);
                setVisible(false);
                new Home().setVisible(true);
            }

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

BANK MANAGEMENT SYSTEM

```
        else
        {
            JOptionPane.showMessageDialog(this, "Please Check Mobile no is not valid!!!",
"Error", JOptionPane.ERROR_MESSAGE);
        }
    }
else{
    JOptionPane.showMessageDialog(this, "Please fill all the fields!!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

}

private String saveImage(File file) {
    String filePath = null;
    //this method will store the images in /storedImages/ folder
    if (file != null) {
        try {
            //first read image
            BufferedImage bufferedImage = ImageIO.read(file);

            //save the image locally
            File imageFile = new File("./storedImages/" + "acc_no-" +txtAccNo.getText() + "-
image-" + System.currentTimeMillis() + ".jpeg");
            filePath = "./storedImages/" +imageFile.getName();
            ImageIO.write(bufferedImage, "jpg", imageFile);

        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
    return filePath;
}
}
```

BANK MANAGEMENT SYSTEM

DeleteSearch:-

```
import java.awt.dnd.DropTarget;
import java.awt.image.BufferedImage;
import java.io.File;
import java.sql.*;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

public class DeleteSearch extends javax.swing.JFrame {
    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    File mUserImageFile;
    File mUserSignFile;
    public DeleteSearch() {
        initComponents();
        conn = DBConnect.connect();

        //init the DragListner class for user profile picture
        DragListener dragListener1 = new DragListener(lblPhoto);
        //set the target componet for dropping item
        new DropTarget(lblPhoto, dragListener1);
    }

    btnSearch.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnSearchActionPerformed(evt);
        }
    });

    btnDelete.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnDeleteActionPerformed(evt);
        }
    });

    btnReset.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnResetActionPerformed(evt);
        }
    });
}
```

BANK MANAGEMENT SYSTEM

```
btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHomeActionPerformed(evt);
    }
});
btnClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCloseActionPerformed(evt);
    }
});
btnLogout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogoutActionPerformed(evt);
    }
});

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    txtAccNo.setText("");
    txtAccType.setText("");
    txtGender.setText("");
    txtAdharNo.setText("");
    txtAccBalance.setText("");
    txtMobileNo.setText("");
    txtHolderName.setText("");
    txtPanNo.setText("");
    txtAddress.setText("");
}

ImageIcon imageIcon = new ImageIcon();

//set the image
lblPhoto.setIcon(imageIcon);
lblSignPhoto.setDisabledIcon(imageIcon);

}

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    System.out.println(txtAccNo.getText());
    if (!txtAccNo.getText().equals("")) {
        try {
            String query = "SELECT * FROM accounts WHERE account_no = ?";
            pst = conn.prepareStatement(query);
            pst.setString(1, txtAccNo.getText());

            rs = pst.executeQuery();

            if (rs.next() && rs.getString("value").equals("0")) {
                txtHolderName.setText(rs.getString("full_name"));
            }
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
txtMobileNo.setText(rs.getString("mobile_no"));
txtAddress.setText(rs.getString("address"));
txtAdharNo.setText(rs.getString("adhar_no"));
txtPanNo.setText(rs.getString("pan_card_no"));
txtGender.setText(rs.getString("gender"));
txtAccType.setText(rs.getString("account_type"));
txtAccBalance.setText(rs.getString("balance"));

mUserImageFile = new File(rs.getString("user_image_url"));
mUserSignFile = new File(rs.getString("user_sign_image_url"));

//display the images in the JLabels
displayImage(lblPhoto, mUserImageFile);
displayImage(lblSignPhoto, mUserSignFile);

} else {
    //account does not exist
    JOptionPane.showMessageDialog(this, "Account does not exist!!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

} catch (Exception e) {
    e.printStackTrace();
}
} else {
    JOptionPane.showMessageDialog(this, "Please enter account number!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);
}

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

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

BANK MANAGEMENT SYSTEM

```
int value=1;
if (!txtAccNo.getText().equals(""))
{
    try
    {
        String query="UPDATE accounts SET value= ? WHERE account_no = ?";
        pst = conn.prepareStatement(query);
        pst.setString(1, String.valueOf(value));
        pst.setString(2, txtAccNo.getText());

        pst.executeUpdate();
        if (pst.executeUpdate()==1)
        {
            int accno=Integer.parseInt(txtAccNo.getText());
            JOptionPane.showMessageDialog(this,"Account number is: "+accno+" Your
Account Deleted Succssfully...\n", "Successfull", JOptionPane.INFORMATION_MESSAGE);

            setVisible(false);
            new Home().setVisible(true);
        }
        else
        {

        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, e.toString(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
else
{
    JOptionPane.showMessageDialog(this, "Please enter account number!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels())
    {
        if ("Nimbus".equals(info.getName()))
        {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(DeleteSearch.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(DeleteSearch.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(DeleteSearch.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(DeleteSearch.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new DeleteSearch().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton btnClose;
private javax.swing.JButton btnDelete;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnReset;
private javax.swing.JButton btnSearch;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel lblPhoto;
private javax.swing.JLabel lblPhoto10;
private javax.swing.JLabel lblPhoto11;
private javax.swing.JLabel lblPhoto2;
private javax.swing.JLabel lblPhoto3;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JLabel lblPhoto4;
private javax.swing.JLabel lblPhoto5;
private javax.swing.JLabel lblPhoto6;
private javax.swing.JLabel lblPhoto7;
private javax.swing.JLabel lblPhoto8;
private javax.swing.JLabel lblPhoto9;
private javax.swing.JLabel lblSignPhoto;
private javax.swing.JLabel photo;
private javax.swing.JTextField txtAccBalance;
private javax.swing.JTextField txtAccNo;
private javax.swing.JTextField txtAccType;
private javax.swing.JTextArea txtAddress;
private javax.swing.JTextField txtAdharNo;
private javax.swing.JTextField txtGender;
private javax.swing.JTextField txtHolderName;
private javax.swing.JTextField txtMobileNo;
private javax.swing.JTextField txtPanNo;
// End of variables declaration

private void displayImage(JLabel label, File imageFile) {
    try {
        BufferedImage bufferedImage = ImageIO.read(imageFile); //read the image
        //resize the the according to the size of JLabel
        BufferedImage resizedImage = CustomImageSize.resize(bufferedImage,
label.getWidth(), label.getHeight());
        //create ImageIcon
        ImageIcon imageIcon = new ImageIcon(resizedImage);
        //set the image
        label.setIcon(imageIcon);

    } catch (Exception e) {
    }
}
}
```

BANK MANAGEMENT SYSTEM

Users:-

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
public class Users extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    public Users() {
        initComponents();
        conn = DBConnect.connect();

    }
    btnAdd.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnAddActionPerformed(evt);
        }
    });

    btnHome.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnHomeActionPerformed(evt);
        }
    });

    btnLog.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnLogActionPerformed(evt);
        }
    });

    btnSearch.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnSearchActionPerformed(evt);
        }
    });

    rdoUser.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            rdoUserActionPerformed(evt);
        }
    });

}
```

BANK MANAGEMENT SYSTEM

```
rdoAdmin.setBackground(new java.awt.Color(255, 204, 0));
buttonGroup1.add(rdoAdmin);
rdoAdmin.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
rdoAdmin.setForeground(new java.awt.Color(153, 0, 0));
rdoAdmin.setText("Admin");

btnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResetActionPerformed(evt);
    }
});

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

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    setVisible(false);
}

private void btnLogActionPerformed(java.awt.event.ActionEvent evt) {
    new Authentication().setVisible(true);
    setVisible(false);
}

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    txtFullName.setText("");
    txtPassword.setText("");
    txtRetypePass.setText("");
    txtUsername.setText("");
    String id = txtEmpId.getText();
    if (!id.equals("") || id != null) {
        try {
            pst = conn.prepareStatement("SELECT * FROM employee WHERE id = ?");
            pst.setString(1, id);

            rs = pst.executeQuery();
            if (rs.next()) {
                String fullName = rs.getString("name") + " " + rs.getString("surname");
                txtFullName.setText(fullName);
            } else {

```

BANK MANAGEMENT SYSTEM

```
JOptionPane.showMessageDialog(this, "Employee with this ID does not exist!!",  
"Error", JOptionPane.ERROR_MESSAGE);  
}  
  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}  
  
}  
  
private void rdoUserActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {  
    txtEmpId.setText("");  
    txtFullName.setText("");  
    txtPassword.setText("");  
    txtRetypePass.setText("");  
    txtUsername.setText("");  
    rdoAdmin.setSelected(false);  
    rdoUser.setSelected(true);  
  
}  
public static void main(String args[]) {  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new Users().setVisible(true);  
        }  
    });  
}  
  
// Variables declaration - do not modify  
private javax.swing.JButton btnAdd;  
private javax.swing.JButton btnHome;  
private javax.swing.JButton btnLog;  
private javax.swing.JButton btnReset;  
private javax.swing.JButton btnSearch;  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JRadioButton rdoAdmin;
private javax.swing.JRadioButton rdoUser;
private javax.swing.JTextField txtEmpId;
private javax.swing.JTextField txtFullName;
private javax.swing.JTextField txtPassword;
private javax.swing.JTextField txtRetypePass;
private javax.swing.JTextField txtUsername;
// End of variables declaration

private void addUser() {
    String id = txtEmpId.getText();
    String name = txtFullName.getText();
    String username = txtUsername.getText();
    String password1 = txtPassword.getText();
    String password2 = txtRetypePass.getText();
    String accessLevel = rdoUser.isSelected() ? "user" : "admin";
    String hashedPassword = "";//to store the hashed password

    Validation validation = new Validation();
    //check if any fields are empty
    if (validation.isEmpty(txtEmpId, txtFullName, txtPassword, txtRetypePass,
        txtUsername) > 0) {
        JOptionPane.showMessageDialog(this, "All fields are required!!", "Error",
        JOptionPane.ERROR_MESSAGE);
    }

    //check if password and retype password matches
    if (password1.equals(password2)) {
        //if password matches the hash the password
        PasswordAuthentication passwordAuthentication = new PasswordAuthentication();
        hashedPassword = passwordAuthentication.hash(password1.toCharArray());

        if (checkUsername(username)) {
            JOptionPane.showMessageDialog(this, "Username already exist.Please choose another
username", "Error", JOptionPane.ERROR_MESSAGE);
        }
        else
        {
            try {
                pst = conn.prepareStatement("INSERT INTO users (full_name, employee_id, username,
password, access_level) VALUES (?, ?, ?, ?, ?)");
                pst.setString(1, name);
                pst.setString(2, id);
```

BANK MANAGEMENT SYSTEM

```
        pst.setString(3, username);
        pst.setString(4, hashedPassword);
        pst.setString(5, accessLevel);

        pst.executeUpdate();
        JOptionPane.showMessageDialog(this, "User added successfull...", "Error",
JOptionPane.INFORMATION_MESSAGE);
        setVisible(false);
        new Home().setVisible(true);

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, "Error!!!Unable to add user.", "Error",
JOptionPane.ERROR_MESSAGE);
    }

}

} else {
    JOptionPane.showMessageDialog(this, "Password did not match!!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

//check if username exists

}

private boolean checkUsername(String username) {
    boolean exist = false;
    try {
        pst = conn.prepareStatement("SELECT username FROM users WHERE username = ?");
        pst.setString(1, username);
        rs = pst.executeQuery();
        if (rs.next()) {
            exist = true;
        }
    }

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

    return exist;
}

}
```

BANK MANAGEMENT SYSTEM

Withdraw:-

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class Withdraw extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    Validation validator;

    public Withdraw() {
        initComponents();
        conn = DBConnect.connect();
        validator = new Validation();
    }

    btnWithdraw.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnWithdrawActionPerformed(evt);
        }
    });
    btnHome.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnHomeActionPerformed(evt);
        }
    });
    btnReset.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnResetActionPerformed(evt);
        }
    });

    btnLogout.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnLogoutActionPerformed(evt);
        }
    });
    btnSearch.addActionListener(new java.awt.event.ActionListener() {
```

BANK MANAGEMENT SYSTEM

```
public void actionPerformed(java.awt.event.ActionEvent evt) {
    btnSearchActionPerformed(evt);
}
});

pack();
setLocationRelativeTo(null);
}// </editor-fold>

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    setVisible(false);
}

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

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    new Authentication().setVisible(true);
    setVisible(false);
}

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    String accountNo = txtAccountNo.getText();

    if (!accountNo.equals("") || accountNo != null) {
        try {
            pst = conn.prepareStatement("SELECT * FROM accounts WHERE account_no = ?");
            pst.setString(1, accountNo);

            rs = pst.executeQuery();

            if (rs.next()&& rs.getString("value").equals("0")) {//if account exist
                txtName.setText(rs.getString("full_name"));
                txtAccType.setText(rs.getString("account_type"));

            } else {
                JOptionPane.showMessageDialog(this, "Account not exist or invalid account
number", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
}

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

    if (validator.isEmpty(txtAccountNo, txtAccType, txtAmount,txtName) > 0) {
        JOptionPane.showMessageDialog(this, "All the fields must not be empty!!!", "Error",
        JOptionPane.ERROR_MESSAGE);
    }

    else if(Double.parseDouble(txtAmount.getText())< 100)
    {
        JOptionPane.showMessageDialog(this, "The Withdraw amount must be greater than 100
        !!!", "Error", JOptionPane.ERROR_MESSAGE);
    }
    else{
        try {
            withdraw();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Withdraw().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnReset;
private javax.swing.JButton btnSearch;
private javax.swing.JButton btnWithdraw;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txtAccType;
private javax.swing.JTextField txtAccountNo;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JTextField txtAmount;
private javax.swing.JTextField txtName;
// End of variables declaration

private void withdraw() throws SQLException {
    String acc_no = txtAccountNo.getText();
    String holderName = txtName.getText();
    String accountType = txtAccType.getText();
    double amount = Double.parseDouble(txtAmount.getText());

    //validate all fields
    Validation Validator = new Validation();
    if (Validator.isEmpty(txtAccountNo, txtName, txtAccType, txtAmount) > 0) {
        JOptionPane.showMessageDialog(this, "All Fields Must Not Be Empty!!!", "Error",
        JOptionPane.ERROR_MESSAGE);
    } else {

        //first check the balance if min amount there
        double currentBalance = checkBalance(acc_no);

        if (currentBalance > 500 && (currentBalance - amount) > 500) {
            try {
                String query = "INSERT INTO transactions (account_no, type, amount, date_time)
VALUES (?, ?, ?, ?)";
                pst = conn.prepareStatement(query);
                pst.setString(1, acc_no);
                pst.setString(2, "deposit");
                pst.setString(3, String.valueOf(amount));
                pst.setString(4, String.valueOf(new java.util.Date()));

                if (pst.executeUpdate() == 1) {
                    updateBalance(acc_no, amount); //update the account balance
                    JOptionPane.showMessageDialog(this, "Amount withdrawn successfully...",
                    "Successful", JOptionPane.INFORMATION_MESSAGE);
                    clearAllFields();
                }
            } catch (SQLException ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(this, "Transcation Failed!!!", "Error",
                JOptionPane.ERROR_MESSAGE);
            }
        } else {
            //insufficient balance
    }
}
```

BANK MANAGEMENT SYSTEM

```
JOptionPane.showMessageDialog(this, "Insufficient balance!!", "Error",
JOptionPane.ERROR_MESSAGE);
}

}

private void updateBalance(String accountNo, Double withdrawAmount) {

try {
    String query = "UPDATE accounts SET balance = balance - ? WHERE account_no =
?";
    pst = conn.prepareStatement(query);
    pst.setDouble(1, withdrawAmount);
    pst.setString(2, accountNo);

    pst.executeUpdate();

} catch (SQLException ex) {
    ex.printStackTrace();
    JOptionPane.showMessageDialog(this, "Transcation Failed!!", "Error",
JOptionPane.ERROR_MESSAGE);
}
}

private void clearAllFields() {
    txtAccountNo.setText("");
    txtName.setText("");
    txtAccType.setText("");
    txtAmount.setText("");
}

private double checkBalance(String accountNo) throws SQLException {
    Statement st = conn.createStatement();
    rs = st.executeQuery("SELECT balance FROM accounts WHERE account_no = " +
accountNo);
    rs.next();
    return Double.parseDouble(rs.getString("balance"));

}
```

BANK MANAGEMENT SYSTEM

Deposite:-

```
import com.sun.media.sound.DLSModulator;
import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class Deposite extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
    ResultSet rs;
    public Deposite() {
        initComponents();
        conn = DBConnect.connect();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txtAccNo = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        txtName = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        txtAccType = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        txtDepositAmount = new javax.swing.JTextField();
        btnDeposite = new javax.swing.JButton();
        btnDeposite1 = new javax.swing.JButton();
        btnLogout = new javax.swing.JButton();
        btnHome = new javax.swing.JButton();
        btnSearch = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        jPanel1.setBackground(new java.awt.Color(255, 204, 0));
        
```

BANK MANAGEMENT SYSTEM

```
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/home.jpg")));
// NOI18N

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel2.setForeground(new java.awt.Color(153, 0, 0));
jLabel2.setText("Account Number :");

txtAccNo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAccNoActionPerformed(evt);
    }
});

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
jLabel3.setForeground(new java.awt.Color(153, 0, 0));
jLabel3.setText("Holder Name :");

txtName.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNameActionPerformed(evt);
    }
});

txtAccType.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAccTypeActionPerformed(evt);
    }
});

txtDepositAmount.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtDepositAmountActionPerformed(evt);
    }
});

btnDeposite.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnDepositeActionPerformed(evt);
    }
});

btnDeposite1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnDeposite1ActionPerformed(evt);
    }
});

btnLogout.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
```

BANK MANAGEMENT SYSTEM

```
btnLogout.setForeground(new java.awt.Color(153, 0, 0));
btnLogout.setText("Logout");
btnLogout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogoutActionPerformed(evt);
    }
});
btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHomeActionPerformed(evt);
    }
});
btnSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSearchActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

setSize(new java.awt.Dimension(629, 373));
setLocationRelativeTo(null);
}// </editor-fold>

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

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

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

BANK MANAGEMENT SYSTEM

```
}

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

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    new Authentication().setVisible(true);
    setVisible(false);
}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    new Home().setVisible(true);
    setVisible(false);
}

private void btnDepositeActionPerformed(java.awt.event.ActionEvent evt) {
    deposit();
}

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    String accountNo = txtAccNo.getText();

    if(!accountNo.equals("") || accountNo != null){
        try {
            pst = conn.prepareStatement("SELECT * FROM accounts WHERE account_no = ?");
            pst.setString(1, accountNo);

            rs = pst.executeQuery();

            if(rs.next()&& rs.getString("value").equals("0")){//if account exist
                txtName.setText(rs.getString("full_name"));
                txtAccType.setText(rs.getString("account_type"));

            }
            else{
                JOptionPane.showMessageDialog(this, "Account not exist or invalid account
number!!", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
}

private void btnDeposite1ActionPerformed(java.awt.event.ActionEvent evt) {
    clearAllFields();
}
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Deposite().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnDeposite;
private javax.swing.JButton btnDeposite1;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnSearch;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txtAccNo;
private javax.swing.JTextField txtAccType;
private javax.swing.JTextField txtDepositAmount;
private javax.swing.JTextField txtName;
// End of variables declaration

private void deposit() {
    String acc_no = txtAccNo.getText();
    String holderName = txtName.getText();
    String accountType = txtAccType.getText();
    String amount = txtDepositAmount.getText();

    //validate all fields
    Validation Validator = new Validation();
    if (Validator.isEmpty(txtAccNo, txtName, txtAccType, txtDepositAmount) > 0) {
        JOptionPane.showMessageDialog(this, "All Fields Must Not Be Empty!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    } else {
        try {
            String query = "INSERT INTO transactions (account_no, type, amount, date_time)
VALUES (?, ?, ?, ?);
```

BANK MANAGEMENT SYSTEM

```
        pst = conn.prepareStatement(query);
        pst.setString(1, acc_no);
        pst.setString(2, "deposit");
        pst.setString(3, amount);
        pst.setString(4, String.valueOf(new java.util.Date()));

        if (pst.executeUpdate() == 1) {
            updateBalance(); //update the account balance
            JOptionPane.showMessageDialog(this, "Amount deposited successfully...", "Successfull", JOptionPane.INFORMATION_MESSAGE);
            clearAllFields();
            setVisible(false);
            new Home().setVisible(true);
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(this, "Transcation Failed!!", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void updateBalance() {
    String acc_no = txtAccNo.getText();
    String amount = txtDepositAmount.getText();

    try {
        String query = "UPDATE accounts SET balance = balance + ? WHERE account_no = ?";
        pst = conn.prepareStatement(query);
        pst.setString(1, amount);
        pst.setString(2, acc_no);

        pst.executeUpdate();

    } catch (SQLException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(this, "Transcation Failed!!", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

BANK MANAGEMENT SYSTEM

```
private void clearAllFields() {  
    txtAccNo.setText("");  
    txtAccType.setText("");  
    txtDepositAmount.setText("");  
    txtName.setText("");  
}  
}
```

Employees:-

```
import java.sql.*;  
import javax.swing.JOptionPane;  
public class Employees extends javax.swing.JFrame {  
  
    Connection conn;  
    ResultSet rs;  
    Statement st;  
    public Employees() {  
        initComponents();  
        conn=DBConnect.connect();  
  
        @SuppressWarnings("unchecked")  
        // <editor-fold defaultstate="collapsed" desc="Generated Code">  
        private void initComponents() {  
  
            pnlEmp = new javax.swing.JPanel();  
            jLabel1 = new javax.swing.JLabel();  
            btnHome = new javax.swing.JButton();  
            btnSelect = new javax.swing.JButton();  
            btnClose = new javax.swing.JButton();  
            btnLogout = new javax.swing.JButton();  
            table = new javax.swing.JScrollPane();  
            jtEmp = new javax.swing.JTable();  
  
            setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);  
            setUndecorated(true);  
  
            pnlEmp.setBackground(new java.awt.Color(255, 204, 0));  
  
            jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/home.jpg")));  
            // NOI18N  
  
            btnHome.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N  
            btnHome.setForeground(new java.awt.Color(153, 0, 0));  
        }  
    }
```

BANK MANAGEMENT SYSTEM

```
btnHome.setText("Home");
btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHomeActionPerformed(evt);
    }
});

btnSelect.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnSelect.setForeground(new java.awt.Color(153, 0, 0));
btnSelect.setText("Select");
btnSelect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSelectActionPerformed(evt);
    }
});

btnClose.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnClose.setForeground(new java.awt.Color(153, 0, 0));
btnClose.setText("Close");
btnClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCloseActionPerformed(evt);
    }
});

btnLogout.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnLogout.setForeground(new java.awt.Color(153, 0, 0));
btnLogout.setText("Logout");
btnLogout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogoutActionPerformed(evt);
    }
});

jtEmp.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jtEmp.setForeground(new java.awt.Color(153, 0, 0));
jtEmp.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
))
```

BANK MANAGEMENT SYSTEM

```
));
jtEmp.setEnabled(false);
table.setViewportView(jtEmp);

javax.swing.GroupLayout pnlEmpLayout = new javax.swing.GroupLayout(pnlEmp);
pnlEmp.setLayout(pnlEmpLayout);
pnlEmpLayout.setHorizontalGroup(
    pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            pnlEmpLayout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(btnSelect, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(70, 70, 70)
                .addComponent(btnHome, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(91, 91, 91)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(71, 71, 71)
                .addComponent(btnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGapContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            pnlEmpLayout.createSequentialGroup()
                .addGapContainerGap(203, Short.MAX_VALUE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 388,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGapGap(202, 202, 202))
);
pnlEmpLayout.setVerticalGroup(
    pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmpLayout.createSequentialGroup()
            .addGapContainerGap(0, 0, Short.MAX_VALUE)
            .addComponent(table, javax.swing.GroupLayout.PREFERRED_SIZE, 81,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGapGap(57, 57, 57)
            .addComponent(table, javax.swing.GroupLayout.PREFERRED_SIZE, 157,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGapPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 230,
                Short.MAX_VALUE))
);
```

BANK MANAGEMENT SYSTEM

```
.addGroup(pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(btnSelect, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnHome, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(pnlEmp, javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(pnlEmp, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

pack();
setLocationRelativeTo(null);
}// </editor-fold>

private void btnSelectActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        st=conn.createStatement();
        rs=st.executeQuery("select * from employee");
        jtEmp.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(rootPane,"Records not fetched");
    }
}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}
```

BANK MANAGEMENT SYSTEM

```
private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);
}

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Employees().setVisible(true);
    }
});

// Variables declaration - do not modify
private javax.swing.JButton btnClose;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnSelect;
private javax.swing.JLabel jLabel1;
private javax.swing.JTable jtEmp;
private javax.swing.JPanel pnlEmp;
private javax.swing.JScrollPane table;
// End of variables declaration
}
```

Customers:-

```
import java.sql.*;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;

public class Customers extends javax.swing.JFrame {

    Connection conn;
    ResultSet rs;
    Statement st;
    public Customers() {

        initComponents();
        conn=DBConnect.connect();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        pnlEmp = new javax.swing.JPanel();
        btnHome = new javax.swing.JButton();
        btnSelect = new javax.swing.JButton();
        btnClose = new javax.swing.JButton();
        btnLogout = new javax.swing.JButton();
        table = new javax.swing.JScrollPane();
        jtEmp = new javax.swing.JTable();
        jLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        pnlEmp.setBackground(new java.awt.Color(255, 204, 0));
        pnlEmp.setOpaque(false);

        btnHome.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        btnHome.setForeground(new java.awt.Color(153, 0, 0));
        btnHome.setText("Home");
        btnHome.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnHomeActionPerformed(evt);
            }
        });

        btnSelect.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        btnSelect.setForeground(new java.awt.Color(153, 0, 0));
        btnSelect.setText("Select");
        btnSelect.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnSelectActionPerformed(evt);
            }
        });

        btnClose.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        btnClose.setForeground(new java.awt.Color(153, 0, 0));
        btnClose.setText("Close");
        btnClose.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnCloseActionPerformed(evt);
            }
        });

        btnLogout.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        btnLogout.setForeground(new java.awt.Color(153, 0, 0));
        btnLogout.setText("Logout");
        btnLogout.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLogoutActionPerformed(evt);
            }
        });

        jtEmp.setModel(DbUtils.resultSetToTableModel(rs));
        jtEmp.setRowHeight(25);
        jtEmp.setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);
        jtEmp.getTableHeader().setReorderingAllowed(false);
        jtEmp.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jtEmpMouseClicked(evt);
            }
        });

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(153, 0, 0));
        jLabel2.setText("Customer Details");

        javax.swing.GroupLayout pnlEmpLayout = new javax.swing.GroupLayout(pnlEmp);
        pnlEmp.setLayout(pnlEmpLayout);
        pnlEmpLayout.setHorizontalGroup(
            pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmpLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(btnHome)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnSelect)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnClose)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnLogout)
                .addGap(18, 18, 18)
                .addComponent(jtEmp, javax.swing.GroupLayout.PREFERRED_SIZE, 450, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jLabel2)
                .addContainerGap())
        );
        pnlEmpLayout.setVerticalGroup(
            pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlEmpLayout.createSequentialGroup()
                .addContainerGap()
                .addGroup(pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(btnHome)
                    .addComponent(btnSelect)
                    .addComponent(btnClose)
                    .addComponent(btnLogout))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jtEmp, javax.swing.GroupLayout.PREFERRED_SIZE, 150, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel2)
                .addContainerGap())
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlEmp, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlEmp, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        );
    }

    private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) { }
    private void btnSelectActionPerformed(java.awt.event.ActionEvent evt) { }
    private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) { }
    private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) { }
    private void jtEmpMouseClicked(java.awt.event.MouseEvent evt) { }
}
```

BANK MANAGEMENT SYSTEM

```
btnHome.setText("Home");
btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHomeActionPerformed(evt);
    }
});

btnSelect.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnSelect.setForeground(new java.awt.Color(153, 0, 0));
btnSelect.setText("Select");
btnSelect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSelectActionPerformed(evt);
    }
});

btnClose.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnClose.setForeground(new java.awt.Color(153, 0, 0));
btnClose.setText("Close");
btnClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCloseActionPerformed(evt);
    }
});

btnLogout.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnLogout.setForeground(new java.awt.Color(153, 0, 0));
btnLogout.setText("Logout");
btnLogout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogoutActionPerformed(evt);
    }
});

jtEmp.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jtEmp.setForeground(new java.awt.Color(153, 0, 0));
jtEmp.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
))
```

BANK MANAGEMENT SYSTEM

```
));
jtEmp.setEnabled(false);
table.setViewportView(jtEmp);

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/home.jpg")));
// NOI18N

javax.swing.GroupLayout pnlEmpLayout = new javax.swing.GroupLayout(pnlEmp);
pnlEmp.setLayout(pnlEmpLayout);
pnlEmpLayout.setHorizontalGroup(
    pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            pnlEmpLayout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(btnSelect, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(70, 70, 70)
                .addComponent(btnHome, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(91, 91, 91)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(71, 71, 71)
                .addComponent(btnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 88,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(ContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE))
                .addGroup(pnlEmpLayout.createSequentialGroup()
                    .addGap(343, 343, 343)
                    .addComponent(jLabel2)
                    .addGap(360, Short.MAX_VALUE))
                .addGroup(pnlEmpLayout.createSequentialGroup()
                    .addGap(ContainerGap())
                    .addComponent(table)
                    .addGap(ContainerGap())))
            );
        );
pnlEmpLayout.setVerticalGroup(
    pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlEmpLayout.createSequentialGroup()
            .addGap(20, 20, 20)
            .addComponent(jLabel2)
            .addGap(31, 31, 31)
            .addComponent(table, javax.swing.GroupLayout.PREFERRED_SIZE, 167,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(PreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 227,
                Short.MAX_VALUE))
        );
    );
}
```

BANK MANAGEMENT SYSTEM

```
.addGroup(pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(btnSelect, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnHome, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
        javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(pnlEmp, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(pnlEmp, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

pack();
setLocationRelativeTo(null);
}// </editor-fold>

private void btnSelectActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        st=conn.createStatement();
        rs=st.executeQuery("select
account_no,full_name,mobile_no,address,gender,account_type,balance from accounts where
value= 0");
        jtEmp.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(rootPane,"Records not fetched");
    }
}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}
```

BANK MANAGEMENT SYSTEM

```
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);
}

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
public static void main(String args[]) {

    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }

//</editor-fold>
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Customers().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JButton btnClose;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnSelect;
private javax.swing.JLabel jLabel2;
private javax.swing.JTable jtEmp;
private javax.swing.JPanel pnlEmp;
private javax.swing.JScrollPane table;
// End of variables declaration
}
```

Transaction:-

```
import java.sql.*;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
public class Transaction extends javax.swing.JFrame {

    Connection conn;
    ResultSet rs;
    Statement st;
    public Transaction() {
        initComponents();
        conn=DBConnect.connect();

    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        pnlEmp = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        btnHome = new javax.swing.JButton();
        btnSelect = new javax.swing.JButton();
        btnClose = new javax.swing.JButton();
        btnLogout = new javax.swing.JButton();
        table = new javax.swing.JScrollPane();
        jtEmp = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);

        pnlEmp.setBackground(new java.awt.Color(255, 204, 0));

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Images/home.jpg")));
```

BANK MANAGEMENT SYSTEM

```
btnHome.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnHome.setForeground(new java.awt.Color(153, 0, 0));
btnHome.setText("Home");
btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnHomeActionPerformed(evt);
    }
});

btnSelect.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnSelect.setForeground(new java.awt.Color(153, 0, 0));
btnSelect.setText("Select");
btnSelect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSelectActionPerformed(evt);
    }
});

btnClose.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnClose.setForeground(new java.awt.Color(153, 0, 0));
btnClose.setText("Close");
btnClose.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCloseActionPerformed(evt);
    }
});

btnLogout.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
btnLogout.setForeground(new java.awt.Color(153, 0, 0));
btnLogout.setText("Logout");
btnLogout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogoutActionPerformed(evt);
    }
});

jtEmp.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
jtEmp.setForeground(new java.awt.Color(153, 0, 0));
jtEmp.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
    }
));
```

BANK MANAGEMENT SYSTEM

```
"Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jtEmp.setEnabled(false);
table.setViewportView(jtEmp);

javax.swing.GroupLayout pnlEmpLayout = new javax.swing.GroupLayout(pnlEmp);
pnlEmp.setLayout(pnlEmpLayout);
pnlEmpLayout.setHorizontalGroup(
    pnlEmpLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            pnlEmpLayout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(btnSelect, javax.swing.GroupLayout.PREFERRED_SIZE, 88, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(70, 70, 70)
                .addComponent(btnHome, javax.swing.GroupLayout.PREFERRED_SIZE, 88, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(91, 91, 91)
                .addComponent(btnClose, javax.swing.GroupLayout.PREFERRED_SIZE, 88, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(71, 71, 71)
                .addComponent(btnLogout, javax.swing.GroupLayout.PREFERRED_SIZE, 88, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGapContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE))
            .addGroup(pnlEmpLayout.createSequentialGroup()
                .addGapContainerGap()
                .addComponent(table)
                .addGapContainerGap())
            .addGroup(pnlEmpLayout.createSequentialGroup()
                .addGapContainerGap(203, Short.MAX_VALUE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 388, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGapGap(202, 202, 202)))
        );
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlEmp, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponentVerticalGroup()
);
```

BANK MANAGEMENT SYSTEM

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(pnlEmp, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

pack();
setLocationRelativeTo(null);
}// </editor-fold>

private void btnSelectActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        st=conn.createStatement();
        rs=st.executeQuery("select * from transactions");
        jtEmp.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        JOptionPane.showMessageDialog(rootPane,"Records not fetched");
    }
}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}

private void btnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);      // TODO add your handling code here:
}
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Transaction.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
```

BANK MANAGEMENT SYSTEM

```
java.util.logging.Logger.getLogger(Transaction.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Transaction.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Transaction.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Transaction().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton btnClose;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnSelect;
private javax.swing.JLabel jLabel1;
private javax.swing.JTable jtEmp;
private javax.swing.JPanel pnlEmp;
private javax.swing.JScrollPane table;
// End of variables declaration
}
```

Add or Delete Employees:-

```
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class AddEmp extends javax.swing.JFrame {

    Connection conn;
    PreparedStatement pst;
```

BANK MANAGEMENT SYSTEM

```
ResultSet rs;
public AddEmp() {
    initComponents();
    conn=DBConnect.connect();
}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    txtID = new javax.swing.JTextField();
    txtName = new javax.swing.JTextField();
    txtLName = new javax.swing.JTextField();
    txtDes = new javax.swing.JTextField();
    txtMob = new javax.swing.JTextField();
    txtEmail = new javax.swing.JTextField();
    btnSearch = new javax.swing.JButton();
    btnAdd = new javax.swing.JButton();
    btnReset = new javax.swing.JButton();
    btnHome = new javax.swing.JButton();
    btnDelete = new javax.swing.JButton();
    btnLog = new javax.swing.JButton();
    jLabel8 = new javax.swing.JLabel();
    txtSal = new javax.swing.JTextField();

    txtID.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtIDActionPerformed(evt);
        }
    });
    });

    txtName.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
    txtName.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtNameActionPerformed(evt);
        }
    });
    });

}
```

BANK MANAGEMENT SYSTEM

```
txtLName.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
txtLName.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtLNameActionPerformed(evt);
    }
});

txtDes.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
txtDes.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtDesActionPerformed(evt);
    }
});

txtMob.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
txtMob.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtMobActionPerformed(evt);
    }
});

txtEmail.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
txtEmail.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEmailActionPerformed(evt);
    }
});

btnSearch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSearchActionPerformed(evt);
    }
});

btnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddActionPerformed(evt);
    }
});

btnReset.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnResetActionPerformed(evt);
    }
});

btnHome.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

BANK MANAGEMENT SYSTEM

```
        btnHomeActionPerformed(evt);
    }
});

btnDelete.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnDeleteActionPerformed(evt);
    }
});

btnLog.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogActionPerformed(evt);
    }
});

txtSal.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
txtSal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtSalActionPerformed(evt);
    }
});

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(23, 23, 23)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(54, 54, 54)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtID, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2)
        .addComponent(btnSearch))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(txtName, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel5))
```

BANK MANAGEMENT SYSTEM

```
.addComponent(txtLName, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jLabel4)
.addComponent(txtDes, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jLabel6)
.addComponent(txtMob, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jLabel7)
.addComponent(txtEmail, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addComponent(jLabel8)
.addComponent(txtSal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(42, 42, 42)

.addComponent(btnAdd)
.addComponent(btnHome)
.addComponent(btnDelete)
.addComponent(btnLog)
.addComponent(btnReset))
.addContainerGap(37, Short.MAX_VALUE))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

BANK MANAGEMENT SYSTEM

```
.addGroup(layout.createSequentialGroup()
    .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE, Short.MAX_VALUE)
);
pack();
setLocationRelativeTo(null);
}// </editor-fold>

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

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

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

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

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

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

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    int id=Integer.parseInt(txtID.getText());
    Validation validation = new Validation();
    //check if any fields are empty
    if (validation.isEmpty(txtID,txtName,txtLName,txtDes,txtEmail,txtMob,txtSal) > 0) {
        JOptionPane.showMessageDialog(this, "All fields are required!!", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
```

BANK MANAGEMENT SYSTEM

```
        }
    else
    {
        if (!txtID.equals("")){
            try {
                String query="DELETE from employee where id=?";
                PreparedStatement pst1=conn.prepareStatement(query);
                pst1.setInt(1,id);
                pst1.executeUpdate();

                JOptionPane.showMessageDialog(this, "Employee Deleted Successfully!!",
"Confirm", JOptionPane.INFORMATION_MESSAGE);
                reset();
                setVisible(false);
                new Home().setVisible(true);
            } catch (Exception e) {
                JOptionPane.showMessageDialog(this, "Employee with this ID does not exist!!",
"Error", JOptionPane.ERROR_MESSAGE);
                e.printStackTrace();
            }
        }
        else
        {
            JOptionPane.showMessageDialog(this, "Please enter employee id!!", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }
}

private void btnHomeActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Home().setVisible(true);
}

private void btnLogActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false);
    new Authentication().setVisible(true);
}

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

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    txtDes.setText("");
    txtEmail.setText("");
}
```

BANK MANAGEMENT SYSTEM

```
txtLName.setText("");
txtMob.setText("");
txtName.setText("");
txtSal.setText("");
String id=txtID.getText();
Validation validation = new Validation();
//check if any fields are empty
if (validation.isEmpty(txtID) > 0) {
    JOptionPane.showMessageDialog(this, "All fields are required!!", "Error",
JOptionPane.ERROR_MESSAGE);
}
else
{
if (txtID.getText().equals(""))
{
    JOptionPane.showMessageDialog(this, "Please enter employee id!!", "Error",
JOptionPane.ERROR_MESSAGE);
}
else
{
try {
    PreparedStatement pst1 = conn.prepareStatement("SELECT * FROM employee
WHERE id = ?");
    pst1.setString(1, id);
    rs=pst1.executeQuery();
    if(rs.next())
    {
        String name = rs.getString("name");
        String surname=rs.getString("surname");
        String des=rs.getString("designation");
        String mob=rs.getString("mobile_no");
        String sal=rs.getString("salary");
        String em=rs.getString("email");
        txtEmail.setText(em);
        txtName.setText(name);
        txtLName.setText(surname);
        txtDes.setText(des);
        txtMob.setText(mob);
        txtSal.setText(sal);
    }
}
else
{
```

BANK MANAGEMENT SYSTEM

```
JOptionPane.showMessageDialog(this, "Employee id not exist!!!", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    } catch (SQLException ex) {

        }

    }

}

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

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    String id=txtID.getText();
    Validation validation = new Validation();
    //check if any fields are empty
    if (validation.isEmpty(txtEmail,txtDes,txtLName,txtMob,txtName,txtSal) > 0) {
        JOptionPane.showMessageDialog(this, "All fields are required!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    }
    else if (!validation.isNumber(txtMob.getText()) || txtMob.getText().length() < 10) {
        JOptionPane.showMessageDialog(this, "Incorrect Mobile number!!!", "Error",
JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        if (txtID.getText().equals("")) {
            JOptionPane.showMessageDialog(this, "Please enter employee id.!!!", "Error",
JOptionPane.ERROR_MESSAGE);
        }
        else {
            try {
                PreparedStatement pst2=conn.prepareStatement("select * from employee where
id=?");
                pst2.setString(1,id);
                rs=pst2.executeQuery();
                if(rs.next())
                {
                    JOptionPane.showMessageDialog(this, "Employee id already exist!!!", "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        }
    }
}
```

BANK MANAGEMENT SYSTEM

```
else
{
    PreparedStatement pst3=conn.prepareStatement("insert into
employee(id,name,surname,designation,mobile_no,email,salary) values(?,?,?,?,?,?,?,?)");
    pst3.setString(1,id);
    pst3.setString(2,txtName.getText());
    pst3.setString(3,txtLName.getText());
    pst3.setString(4,txtDes.getText());
    pst3.setString(5,txtMob.getText());
    pst3.setString(6,txtEmail.getText());
    pst3.setString(7,txtSal.getText());



if(pst3.executeUpdate()==1)
{
    JOptionPane.showMessageDialog(this, "Employee Added Successfully",
"Error", JOptionPane.ERROR_MESSAGE);
    reset();
    setVisible(false);
    new Home().setVisible(true);
}
}

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

}

/***
 * @param args the command line arguments
 */
public static void main(String args[]) {
/* Set the Nimbus look and feel */
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
 * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 */
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
        }
    }
}
}
```

BANK MANAGEMENT SYSTEM

```
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(AddEmp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(AddEmp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(AddEmp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(AddEmp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new AddEmp().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton btnAdd;
private javax.swing.JButton btnDelete;
private javax.swing.JButton btnHome;
private javax.swing.JButton btnLog;
private javax.swing.JButton btnReset;
private javax.swing.JButton btnSearch;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
```

BANK MANAGEMENT SYSTEM

```
private javax.swing.JTextField txtDes;
private javax.swing.JTextField txtEmail;
private javax.swing.JTextField txtID;
private javax.swing.JTextField txtLName;
private javax.swing.JTextField txtMob;
private javax.swing.JTextField txtName;
private javax.swing.JTextField txtSal;
// End of variables declaration

private void reset() {
    txtDes.setText("");
    txtEmail.setText("");
    txtID.setText("");
    txtLName.setText("");
    txtMob.setText("");
    txtName.setText("");
    txtSal.setText("");
}
}
```

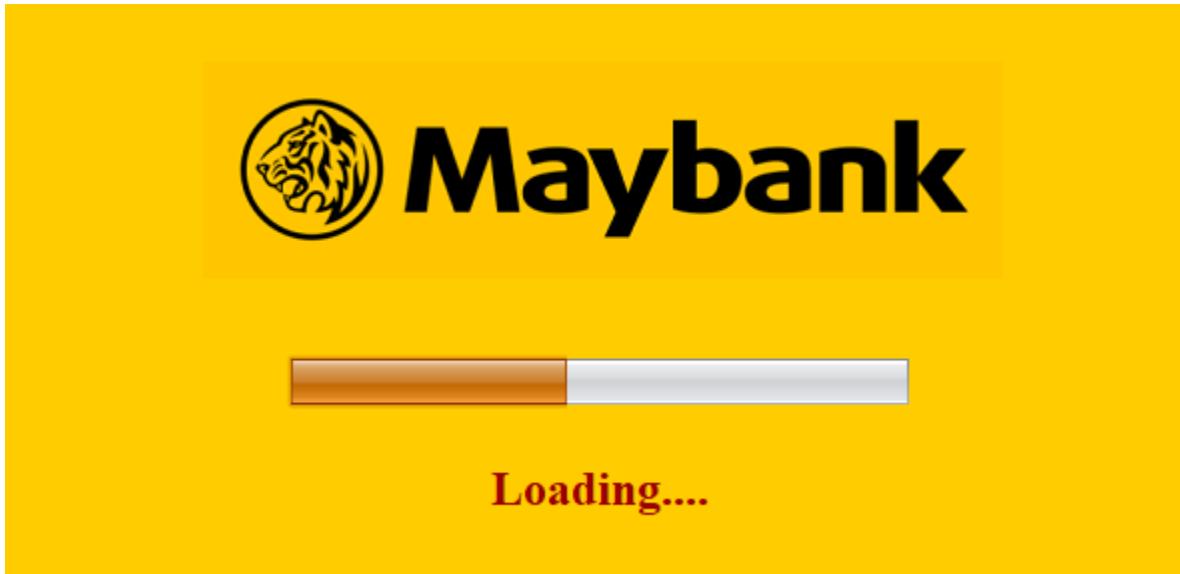
BANK MANAGEMENT SYSTEM

DATA ENTRY SCREENS

Login Form

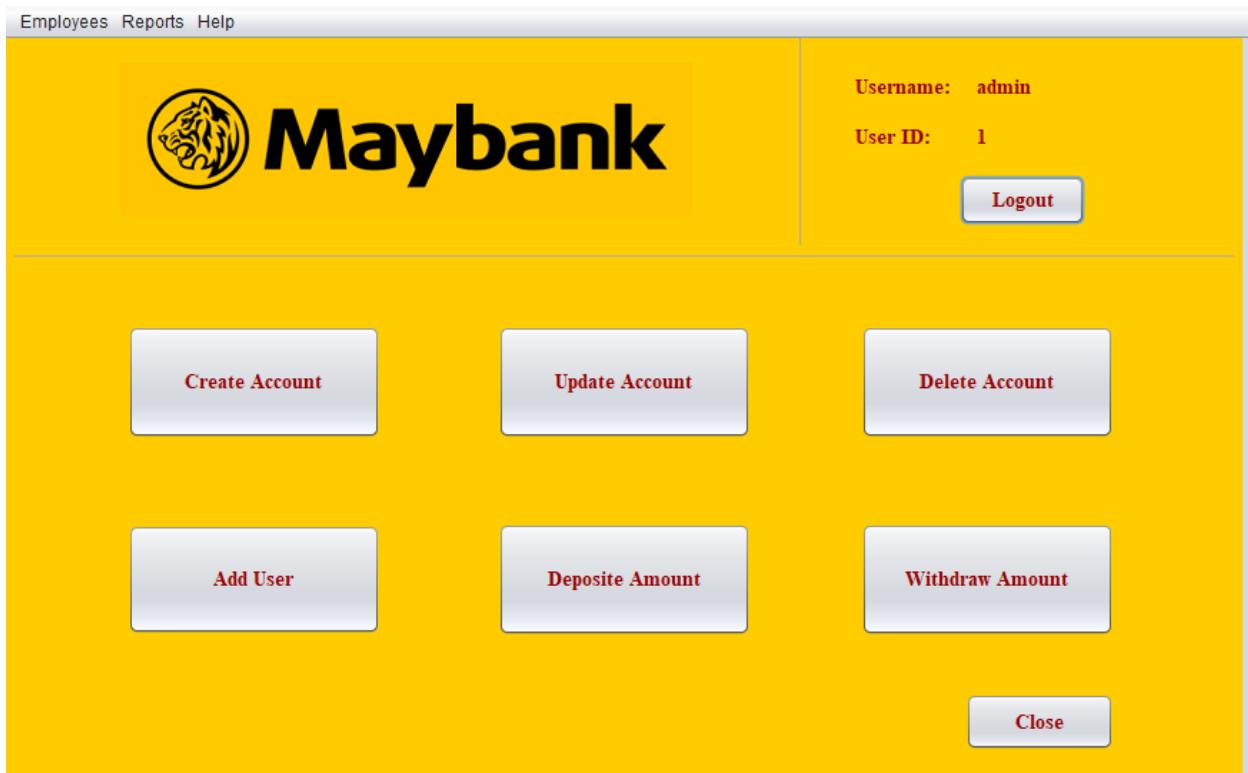


Loading



BANK MANAGEMENT SYSTEM

Home Page



BANK MANAGEMENT SYSTEM

Create Account:

 **Maybank**

Photo: 

Address:

Adhar Card No:

Pan card No:

Gender: Male Female

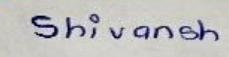
Enter Name:

Account Type:

Account No:

Mobile No:

Deposite Amount:

Signature: 

Choose

Create Account **Reset** **Close** **Home** **Logout**

BANK MANAGEMENT SYSTEM

Update Account:

 **Maybank**

Photo:		Mobile No:	1234567891		
		Adhar No:	123456789012		
		PAN No:	1234567890		
Account No:	11	Account Type:	Saving		
	Gender:	Male			
	Holder Name:	Ropali Munshi	Account Balance:	1599.00	
Address:	ulhasnagar -421003	Signature			
Finish Update		Close	Reset	Logout	Home

BANK MANAGEMENT SYSTEM

Delete Account:

Maybank

Photo:		Mobile No.:	1234567891
Account No.:	11	Adhaar No.:	123456789012
Holder Name:	Ropali Munshi	PAN No.:	1234567890
Address:	ulhasnagar -421003	Account Type:	Saving
Gender:	Male	Account Balance:	1599.00
		Signature:	
Delete		Home	Logout
		Close	Reset

BANK MANAGEMENT SYSTEM

Withdraw Money:



Maybank

Account No: **search**

Holder Name:

Account Type:

Enter Amount to Withdraw:

Withdraw **Home** **Reset** **Logout**

BANK MANAGEMENT SYSTEM

Deposit Money:



Maybank

Account Number : 11

Holder Name : Ropali Munshi

Account Type: Saving

Enter Deposite Amount : 1000

Add /Delete Employee:



Maybank

Enter User ID: 11

Enter Full Name: akshay patil

Enter Username:

Enter Password:

Retype Password:

Acess Type User Admin

BANK MANAGEMENT SYSTEM

Add/Delete Employee:



Maybank

Enter ID:

Enter First Name:

Enter Last Name:

Enter Designation:

Enter Mobile No.:

Enter Email-id:

Salary:

BANK MANAGEMENT SYSTEM

REPORT LAYOUTS

Employees:

 Maybank

id	name	surname	designation	mobile_no	email	salary
3	Rohan	Shinge	Database Admini...	8547598625	rohanshinge98@...	25000
4	Suhas	Dayre	IT Manager	8877986544	suhasdayare@gm...	80000

[Select](#) [Home](#) [Close](#) [Logout](#)

Customers:

 Maybank

account_no	full_name	mobile_no	address	gender	account_type	balance
11	Ropali Munshi	1234567891	ulhasnagar -421003	Male	Saving	1599.00
23	shivansh Mishra	8897551452	ambernath east	Male	Saving	18000.00
25	sgadsihv	2255664455	exgbdg	Male	Saving	1000.00
26	zxfgsdf	4445556644	Ulhasnagar	Male	Saving	1000.00

[Select](#) [Home](#) [Close](#) [Logout](#)

BANK MANAGEMENT SYSTEM

Transaction:

**Maybank**

id	account_no	type	amount	date_time
2	2	deposit	1000	Sun Dec 17 21:29:14 IST ...
3	2	deposit	100.0	Sun Dec 17 21:56:08 IST ...
4	2	deposit	200.0	Sun Dec 17 21:59:51 IST ...
20	23	Saving	1000.0	Fri Mar 02 15:59:53 IST 2...
21	11	deposit	1000	Sat Mar 03 11:57:56 IST ...
22	11	Saving	200.0	Sat Mar 03 11:58:23 IST ...
23	11	Withdraw	100.0	Sat Mar 03 12:13:39 IST ...

[Select](#)[Home](#)[Close](#)[Logout](#)

BANK MANAGEMENT SYSTEM

About us:



Maybank

"Your Faithful and Friendly Financial and Banking Partners"

BRANCH INFORMATION

Bank Name :-	MayBank
Branch Name :-	Manere Goan Ulhasnagar-4
Opening Date :-	27 March 1998
IFSC Code :-	MB00270398
MICR Code :-	400229221
Branch Code :-	27031998
Phone No. :-	18001122334
Working Time :-	Mon. - Fri. 09:30AM - 04:30 PM Sat. 09:30 AM - 04:30 PM (2nd and 4th Sat. Closed)
Address :-	Ground Floor, Dharmaji Palace, Near New Ramchandra Colony, Manere-goan, Ulhasnagar East, Pin code 421004

[Home](#) [Close](#) [Logout](#)

BANK MANAGEMENT SYSTEM

TESTING

Test Cases

Name of the Test Case: **Login Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Click on login button without giving username & password	It should be ask for enter username & password	Pass	It has showing error message for “enter username & password”
02	Enter username without password	It should be ask for enter password	Pass	It has displayed error message for “enter password”
03	Enter password without username	It should be ask for enter username	Pass	It has displayed error message for “enter username”
04	Enter invalid username & password	It should be show message for invalid username & password	Pass	It has displayed error message for “please enter valid username & password”
05	Enter valid username & password	It should be redirect to next form	Pass	It has redirected to next form

BANK MANAGEMENT SYSTEM

Name of the Test Case: **Create Account Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Without Entering any details directly click on create account button	All fields are required!!	Pass	All fields are required!!
02	Mobile no, addhar no and pan card no not enter properly direct click create account button	Please enter the details are carefully.	Pass	Please enter the details are carefully.
03	Without inserting customer profile and sign click create button	Customer profile and sign are mandatory.	Pass	Please enter the details are carefully.
04	Enter the balance less than 1000.	To create account at least have balance 1000	Pass	To create account at least have balance 1000
05	Enter all valid details and click on create button	Account created successfully.	Pass	Account created successfully.

Name of the Test Case: **Update Account Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Mobile no is not inserted properly	Please enter the mobile no properly	Pass	Please enter the mobile no properly
02	Without inserting customer account no and click search account button	Please enter account number.	Pass	Please enter account number.
03	Enter invalid account no and click search button	Invalid account no or account don't exist	Pass	Invalid account no or account don't exist
04	Enter all details valid then click update button.	Update details successfully.	pass	Update details successfully.

BANK MANAGEMENT SYSTEM

Name of the Test Case: **Delete Account Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Without inserting account no click on delete button	Please enter account no to delete	Pass	Please enter account no to delete
02	Enter invalid account no and click delete button	Invalid account no or account don't exist	Pass	Invalid account no or account don't exist
03	Enter all valid details and click delete button	Account Deleted successfully.	Pass	Account Deleted successfully.

Name of the Test Case: **AddDelete User Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Without entering all details click add or delete button	All fields are required!!	Pass	All fields are required!!
02	Enter existing user id and click add user button	User already exists	Pass	User already Exists
03	Enter existing username and click add user button	Username already exists.	Pass	Username already exists.
04	Enter all valid details and click Add button.	User added Successfully.	Pass	User added Successfully.
05	Enter all valid details and click Delete button	User Deleted successfully	Pass	User Deleted successfully

BANK MANAGEMENT SYSTEM

Name of the Test Case: **Deposit Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Withour entering all details click Deposit button	All fields are required!!	Pass	All fields are required!!
02	Enter invalid account no and click deposit button	Invalid account no or account doesn't exists	Pass	Invalid account no or account doesn't exists
03	Enter the deposit amount less than and click deposit button.	At least 100 rupies required to deposite.	Pass	At least 100 rupies required to deposite.
04	Enter all valid details and click deposit button	Amount Deposited Sucessfully.	Pass	Amount Deposited Sucessfully.

Name of the Test Case: **Withdraw Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Withour entering all details click withdraw button	All fields are required!!	Pass	All fields are required!!
02	Enter invalid account no and click withdraw button	Invalid account no or account doesn't exists	Pass	Invalid account no or account doesn't exists
03	Enter the withdrawal amount greater than account balance	Insufficient Balance.	Pass	Insufficient Balance.
04	Account balance gone below the minimum balance.	Insufficient Balance.		Insufficient Balance.
05	Enter all valid details and click deposit button	Amount Deposited Sucessfully.	Pass	Amount Deposited Sucessfully.

BANK MANAGEMENT SYSTEM

Name of the Test Case: **AddDelete Employee Form**

Test Case #	Test Case Description	Expected Results	Pass / Fail	Actual Results
01	Without entering all details click add or delete employee button	All fields are required!!	Pass	All fields are required!!
02	Enter existing employee id and click add employee button	Employee already exists	Pass	Employee already Exists
03	Enter invalid mobile no	Invalid mobile no.	Pass	Invalid mobile no.
04	Enter all valid details and click Add Employee button.	Employee added Successfully.	Pass	Employee added Successfully.
05	Enter all valid details and click Delete Employee button	Employee Deleted successfully	Pass	Employee Deleted successfully

REFERENCES & BIBLIOGRAPHY

Websites :-

- www.google.com
- www.youtube.com
- www.uml-diagrams.com
- www.softdaigram.com