A

PROJECT REPORT

ON

BANKING SYSTEM IN JAVA

Submitted in partial fulfillment of the requirement for the award of the

"BACHELOR OF SCIENCE"

B.Sc Final Year (2021-22)

C.C.S UNIVERSITY, MEERUT

UNDER THE GUIDANCE OF:

Submitted To:

Mr. Mohit Bansal

Mr. Amit Sharma

Submitted By:

Pulkit Pal

D.A.V.(P.G.) COLLEGE,

MUZAFFARNAGAR 251001

DECLARATION

We hereby declare that the project work entitled "BANKING SYSTEM" submitted to the DAV College Muzaffarnagar is a record of an original work done by Pulkit Pal under the guidance of Mr. Mohit Bansal & Mr. Amit Sharma this project work is submitted in the partial fulfilment of the requirements for the award of the Bachelor of Science (Computer Application).

PULKIT PAL

ACKNOWLEDGEMENT

We would like to express our greatest appreciation to the all individuals who have helped and supported us throughout the project. We are thankful to our computer teacher for his ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project on “BANKING SYSTEM”.

A special acknowledgement goes to our classmates who helped us in completing the project by exchanging interesting ideas and sharing their experience.

We wish to thank our parents as well for their undivided support and interest who inspired us and encouraged us to go our own way, without whom We would be unable to complete our project.

At the end, we would like to thank our friends who displayed appreciation to our work and motivated us to continue our work.

PULKIT PAL

CONTENTS

- Data Dictionary
- List of tables
- Introduction To Java & SQL
- Source Code
- Snapshots (From Design)
- Implementation
- Maintenance
- Security Measures taken
- Conclusion & Future Scope
- Bibliography

PREFACE

We herby declare that the work which is being presented in the project entitled "BANKING SYSTEM" is an authentic work carried out by us under guidance of Mr.Mohit Bansal & Mr. Amit Sharma of D.A.V.(P.G.) College Muzaffarnagar.

This project will work as reference to use the software "BANKING SYSTEM".

We had tried our best to keep it error free still. Perfection cannot be claim by humans.

Pulkit Pal

INTRODUCTION ABOUT THE PROJECT

BANKING SYSTEM:

Simply put, banking software is a type of software that provides fitness businesses the functionality needed to manage all aspects of their business and efficiently operate their studio. Banking System software can also be referred to as transaction system software, fitness software, or Banking System scheduling software.

Regardless of the nomenclature, these platforms all share similar feature sets and are used for the same purposes. Banking System software helps fitness owners and operators manage their class and trainer scheduling, keep track of their customers, communicate with clients, and process payments.

MAIN OBJECTIVE OF THIS PROJECT:

- To provide security as only an authorized customer can interact with the system.
- To help the customer in working easy and efficiently.
- To reduce the manpower.
- To maintain the data with complete consistency & minimum redundancy.
- To help the customer in tracing the transaction quickly.

USER OF THE SOFTWARE:

From banking owners and operators to the trainers and front-desk staff, all banking customers interact with the software. However, depending on their role, how they utilize the software will be different.

Trainers and instructors will use the software to manage their availability, check their schedules, and view member information. Banking System will use the software to enroll clients in memberships, schedule classes for members, and process payments.

This means that when choosing a software, fitness studios must understand that their decision is not just about their internal operations, but also their members. Using a banking system software that provides a poor customer experience could make that member look elsewhere.

NECESSITY OF USING THE SOFTWARE:

It helps optimize studio operations:

Simply put, banking system software helps a studio operate more efficiently. While spreadsheets work when starting the business, eventually the administrative workload becomes too much. Banking system software helps automate routine tasks like data entry, member check-ins, schedule management, and membership renewals. By streamlining the day-to-day

operations at the studio, staff now have more time, focus, and energy to devote to other areas of the business.

Since the payments platform is integrated with the business software, each transaction adds more data points and gives a more clear view of the overall state of the business. It also eliminates the need to export data from various sources to put in a spreadsheet and run calculations.

SOFTWARE & HARDWARE REQUIREMENTS

- Hardware Requirements:

Processor	: Pentium 4 or above
Memory	: 512 MB RAM or above
Hard disk	: 20 MB or above
Keyboard	: 104 keys

- Software Requirements:

Operating System: Windows 7 or above
Programming Language: Java
Database Applications: JDBC
Database: MY SQL 8.0 or above
NetBeans: 12.6

SYSTEM ANALYSIS

Systems analysis refers to the process of examining a business situation with the intent of improving it through better procedure and methods. Systems development can generally be thought of as having two major components Systems Analysis and Systems Design.

Systems analysis is the process of gathering and interpreting facts, diagnosis problems and using the information to recommend improvement to the system. In brief, we can say that analysis specifies what the system should do.

System analysis is conducted with the following objectives in mind:

- Identify the Student and Staff members need.
- Evaluate the system concept for feasibility.
- Perform economic and technical analysis.
- Allocate function to hardware, software, people, database and other system elements.
- Establish cost and schedule constraints.
- Create a system definition that forms the foundation for all subsequent engineering.

FEASIBILITY STUDY:

The concept of feasibility is to determine whether or not a project is worth doing. The process followed in making this determination is called feasibility study. Once it has

been determined that a project is feasible, the system analyst can go ahead and prepare the project specification which finalizes project requirements.

Types of feasibility:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility
- Social feasibility
- Management Feasibility
- Legal Feasibility
- Time Feasibility

Here we describe only few of these in detail:

- TECHNICAL FEASIBILITY:

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. Technical needs of the system include:

- Facility to produce output in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transactions at a particular period.
- Facility to communicate data to distant location.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. Configuration should give the complete picture about the system's requirements: how many workstation are required, how these units are

interconnected so that they could operate and communicate smoothly. What speeds of input and output should be achieved at particular quality of printing. The computers are easily available in almost all the places, even in villages. The hardware needed to carry out this project include workstation with 64 MB of RAM and 2 GB HDD.
Due to all these reasons implementation of such system becomes not only feasible but reputed to the organization.

- OPERATIONAL FEASIBILITY

This is mainly related to human organization and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organizational structures are disturbed?
- What new skills will be required? Do the existing staff members have these skills?

If not, can they be trained in due course of time.

This feasibility study is carried out by a small group of people who are familiar with the information system techniques, who understand the parts of business that are relevant to the project and are skilled in the system analysis and design process. This project is not developed just for fun. They are developed on demand of the organization for which the system is being developed. Therefore the chances of resistance from the company Staff is almost nil. Any disturbance to the organization if occurs will be advantageous to the organization. Also the time required to carry out a transaction will be required to a large extent, which will

make the students and others happy and cheerful. The operators now will be able to service more students and staff members than before in same time period. There is no need to recruit new staff cooperate the system. The existing staff of the company can be trained to interact with the system, which is a GUI, based software and is easy to use. Hence the project is Operationally feasible.

- ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost-benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system.

- SOCIAL FEASIBILITY

It is the determination of whether a proposal project will be acceptable to, the staff members and students, or not. This determination typically examines the probability of the project being accepted by the group directly affected the proposed system change. To solve the actual problems in a company setting, a software or a team of engineers must incorporate a development strategy that encompasses the process, methods, and tools layers. This strategy is often referred to as a software engineering paradigm. A Process model for software engineering is chosen based on the nature of the project and application , the methods and tools to be used and the controls and deliverables that are required.

DATA FLOW DIAGRAMS

DFD's are commonly used during problem analysis. Data flow diagrams are not limited to problem analysis for software requirement specification.

A DFD shows the flow of data through the system. It views a system as a function that that transforms the inputs into desired outputs. The DFD aim to capture the transformations that take place within a system to the input data so that eventually the output data is produced. The agent that performs the transformation of data from one state to another is called a process (or bubble). The processes are shown by named circles and data flows are represented by named arrows entering or leaving the bubbles. A rectangle represents a sources or sink and is a net originator or consumer of data.

It should be pointed out that DFD is not a flowchart. A DFD represents the flow of data, while a flowchart shows the flow of control. A DFD does not represent procedural information. In drawing the DFD the designer has to specify the major transforms in the path of the data flowing from input to output.

DATA DICTIONARY

Data dictionary is a repository of various data flows defined in data flow diagram. The associated data dictionary states precisely the structure of each data flow in DFD. Although the format of dictionaries varies from tool to tool, most consists of the following information:

Name - the primary name of the data or control item, the data store or an external entity.

Alias - other names used for the first entity.

Where - used / how - used - a listing of the processes that used the data or control item and how it is used (e.g. input to the process, output from the process, as a store, as an external entity).

Content description - a notation for representing content.

Supplementary information - other information about data types, present values (if known), restrictions of limitation.

LIST OF TABLES

customer table:

accounttable											
accno	hname	acctype	age	gende r	contact	idproof	email	gaccno	gname	address	bal
1009	raj	saving	15	m	789456230	adhaar	hj@gmail.com	786	mohan	mzn	10000
51132	rajat	general	56	male	654374	adhar card	rajat@gmail.com	51134	rakesh	prempuri	50000
123	akki	saving	56	male	3467890	adhar card	akki@gmail.com	6789	akash	prempuri	140180
1233	RISHABH	GENERAL	34	MALE	74788437	ADHAR CARD	RISHU2@gmail.com	123	MINAL	SAKET	6600
1234	RIA	GENERAL	23	FEMALE	457830340	VOTER CARD	RIA2@gmail.com	1234	rishabh	krishnapuri	50000
51131	akash	saving	34	male	983765	voter id	akki@gmail.com	51132	rajat	saket	1234
51133	minal	general	67	female	768595	voter id	minal@gmail.com	51132	rajat	krishnapuri	8765

ID table:

logintable	
username	password
abc	123

INTRODUCTION TO JAVA & SQL

A Small History of Java:

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Oracle has now the steermanship for Java. In 2006 Sun started to make Java available under the GNU General Public License (GPL). Oracle continues this project called OpenJDK.
Over time new enhanced versions of Java have been released. The current version of Java is Java 1.8 which is also known as Java 8.

Java is defined by a specification and consists of a programming language, a compiler, core libraries and a runtime (Java virtual machine) The Java runtime allows software developers to write program code in other languages than the Java programming language which still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

The Java language was designed with the following properties:

- Platfoím independent: Java píogíams use the Java viítual machine as abstíaction and do not access the opeíating system diíectly. l'his makes Java píogíams highly poítable. A Java píogíam (which is standaíd-compliant and follows ceítain íules) can íun unmodified on all suppoíted platfoíms, e.g., Windows oí Linux.
- Object-oíientated píogíamming language: Except the píimitive data types, all elements in Java aíe objects.
- Strongly-typed programming language: Java is strongly- typed, e.g., the types of the used variables must be pre- defined and conversion to other objects is relatively strict, e.g., must be done in most cases by the programmer.
- Interpreted and compiled language: Java source code is transferred into the bytecode format which does not depend on the target platform. These bytecode instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical bytecode instructions into native code instructions.
- Automatic memory management: Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.

The Java syntax is similar to C++. Java is case-sensitive, e.g., variables called myValue and myvalue are treated as different variables.

Hello world Java program

```
public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello World");
```

```
    }
}
```

Java virtual machine:

The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine.

The Java virtual machine is written specifically for a specific operating system, e.g., for Linux a special implementation is required as well as for Windows

Java programs are compiled by the Java compiler into bytecode. The Java virtual machine interprets this bytecode and executes the Java program.

Features of Java

There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.

1.Simple

According to Sun, Java language is simple because:

- syntax is based on C++ (so easier for programmers to learn it after C++).
- removed many confusing and/or rarely-used features
  e.g., explicit pointers, operator overloading etc.
- No need to remove unreferenced objects because there is Automatic Garbage Collection in java.

2. Object-oriented

Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.

Object-oriented programming (OOPs) is a methodology that simplify software development and maintenance by providing some rules.

Basic concepts of OOPs are:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

3. Platform Independent

A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware- based. Java provides software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

1. Runtime Environment
2. API (Application Programming Interface)

Java code can be run on multiple platforms Example- Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms that is Write Once and Run Anywhere (WORA).

4. Secured

Java is secured because:

- No explicit pointer
- Java Programs run inside virtual machine sandbox
- Class loader: adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- Bytecode Verifier: checks the code fragments for illegal code that can violate access right to objects.
- Security Manager: determines what resources a class can access such as reading and writing to the local disk.

These Security are provided by java language. Some security can also be provided by application developer through SSL, JAAS, Cryptography etc.

5. Robust

Robust simply means strong. Java uses strong memory management. There Are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points make's java robust.

6. Architecture-neutral

There are no implementation dependent features Example- size of primitive types is fixed.

In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architecture

Portable

We may carry the java bytecode to any platform.

7. High-performance

Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

8. Distributed

We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet

9. Multi - threaded

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications etc.

The advantages of Java are as follows:

- Java is easy to learn.

Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

- Java is object-oriented.

This allows you to create modular programs and reusable code.

- Java is platform-independent.

One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

Because of Java's robustness, ease of use, cross-platform capabilities and security features, it has become a language of choice for providing worldwide Internet solutions.

SQL:

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query
Language. This tutorial will give you a quick start to SQL. It covers most of the topics required for a basic understanding of SQL and to get a feel of how it works.

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as –

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL

  (native format) etc.

As mentioned before, SQL is one of the most widely used query language over the databases. I'm going to list few of them here:

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

SOURCE CODE

```
*              To change this template, choose Tools | Templates  * and open the template in
the editor.
 */
package bankingproject;
import java.sql.*; import
javax.swing.*;

/**
 *
*              @author student
 */
public class loginform extends javax.swing.JFrame {

    /**
*              Creates new form loginform
     */
    public loginform() {
initComponents();
    }

    /**
*              This method is called from within the constructor to initialize the form.      *
WARNING: Do NOT modify this code. The content of this method is always      *
regenerated by the Form Editor.
     */
    @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();
```

```
        txtuser = new javax.swing.JTextField();
txtpassword = new javax.swing.JTextField();
loginbtn = new javax.swing.JButton();        cancelbtn =
new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBackground(new java.awt.Color(255, 0, 102));

        jLabel1.setFont(new java.awt.Font("Tw Cen MT", 3, 36)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 0, 255));        jLabel1.setText("LOGIN
FORM");

        jLabel2.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(0, 0, 153));        jLabel2.setText("USERNAME");

        jLabel3.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
jLabel3.setForeground(new java.awt.Color(0, 0, 153));        jLabel3.setText("PASSWORD");

        txtuser.setText("a");        txtuser.addActionListener(new
java.awt.event.ActionListener() {            public void
actionPerformed(java.awt.event.ActionEvent evt) {
txtuserActionPerformed(evt);
            }
        });

        loginbtn.setText("LOGIN");                loginbtn.addActionListener(new
java.awt.event.ActionListener()    {                                public    void
actionPerformed(java.awt.event.ActionEvent                evt)                {
loginbtnActionPerformed(evt);
            }
        });

        cancelbtn.setText("CANCEL");        cancelbtn.addActionListener(new
java.awt.event.ActionListener() {            public void
actionPerformed(java.awt.event.ActionEvent evt) {
cancelbtnActionPerformed(evt);
            }
        });
```

```
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);        jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(104, 104, 104)
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 216, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(29, 29, 29)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 73, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(55, 55, 55)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
.addComponent(txtuser)
                            .addComponent(txtpassword, javax.swing.GroupLayout.DEFAULT_SIZE, 152, Short.MAX_VALUE)))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(99, 99, 99)
                        .addComponent(loginbtn, javax.swing.GroupLayout.PREFERRED_SIZE, 87, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(41, 41, 41)
                        .addComponent(cancelbtn, javax.swing.GroupLayout.PREFERRED_SIZE, 86, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(35, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(32, 32, 32)
                .addComponent(jLabel1)
                .addGap(36, 36, 36)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 30, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(txtuser, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 22, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(txtpassword, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(46, 46, 46)
```

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

                .addComponent(loginbtn)

                .addComponent(cancelbtn))

            .addContainerGap(53, Short.MAX_VALUE))

        );



        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)

        );

         pack();

    }// </editor-fold>



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

// TODO add your handling code here:         String user1,user2,pass1,pass2;

user1=txtuser.getText();


        pass1=txtpassword.getText();

        try

        {

            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

            Connection con=DriverManager.getConnection("jdbc:odbc:ban");

            Statement stmt=con.createStatement();

            String query="select * from logintable";

ResultSet res=stmt.executeQuery(query);

            res.next();             user2=res.getString(1);

pass2=res.getString(2);

if(user1.equals(user2)&&pass1.equals(pass2))

JOptionPane.showMessageDialog(null,"correct");

            else

            JOptionPane.showMessageDialog(null,"incorrect");

splashform obj=new splashform();             obj.show();

        }

        catch(Exception e)

        {}
```

```
    }

    private void cancelbtnActionPerformed(java.awt.event.ActionEvent evt) {                                          //
TODO add your handling code here:
        System.exit(0);
    }

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

    /**
* @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());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(loginform.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(loginform.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(loginform.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(loginform.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
loginform().setVisible(true);
          }
      });
   }
   // Variables declaration - do not modify
private javax.swing.JButton cancelbtn;     private
javax.swing.JLabel jLabel1;     private
javax.swing.JLabel jLabel2;     private
javax.swing.JLabel jLabel3;     private
javax.swing.JPanel jPanel1;     private
javax.swing.JButton loginbtn;     private
javax.swing.JTextField txtpassword;

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

}Home.java

/* * To change this template, choose Tools | Templates * and open the template in the editor. */ package bankingproject; import java.sql.*; import javax.swing.*; /**

*

* @author student */ public class mainform extends javax.swing.JFrame {

/** * Creates new form mainform */ public mainform() { initComponents(); }

```
    /**
* This method is called from within the constructor to
  initialize the form.
* WARNING: Do NOT modify this code. The content
  of this method is always
* regenerated by the Form Editor.
     */     @SuppressWarnings("unchecked")     // <editor-
fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

        jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();         account =
new javax.swing.JMenuItem();         saving = new
javax.swing.JMenuItem();         withdrawl = new
javax.swing.JMenuItem();         exit = new
javax.swing.JMenu();         exitbtn = new
javax.swing.JMenuItem();
setDefaultCloseOperation(javax.swing.Window
Constants.EXIT_ON_CLOSE);

        jMenu1.setText("Customer Account");

        account.setText("NEW ACCOUNT FORM");
        account.addActionListener(new
java.awt.event.ActionListener()                {
public void
```

```
actionPerformed(java.awt.event.ActionEvent evt) {
accountActionPerformed(evt);
            }
        });        jMenu1.add(account);

        saving.setText("SAVING FORM");
saving.addActionListener(new
java.awt.event.ActionListener() {
public void
actionPerformed(java.awt.event.ActionEvent
evt) {                savingActionPerformed(evt);
            }
        });        jMenu1.add(saving);

        withdrawl.setText("WITHDRAWL FORM");
withdrawl.addActionListener(new
java.awt.event.ActionListener() {            public void
actionPerformed(java.awt.event.ActionEvent evt) {
withdrawlActionPerformed(evt);
            }
        });        jMenu1.add(withdrawl);

        jMenuBar1.add(jMenu1);

        exit.setText("Exit");
exit.addActionListener(new
```

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

        exitbtn.setText("YES");
exitbtn.addActionListener(new
java.awt.event.ActionListener()              {
public void
actionPerformed(java.awt.event.ActionEvent evt) {
exitbtnActionPerformed(evt);
            }
        });         exit.add(exitbtn);

        jMenuBar1.add(exit);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)
```

```
            .addGap(0, 400, Short.MAX_VALUE)        );
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)            .addGap(0, 279,
Short.MAX_VALUE)        );


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


    private void
accountActionPerformed(java.awt.event.Action Event
evt) {
        // TODO add your handling code here:
        account obj=new account();        obj.show();
    }


    private void
savingActionPerformed(java.awt.event.ActionE vent
evt) {                                      // TODO add your
handling code here:         savingform obj=new
savingform();        obj.show();
    }


    private void
withdrawlActionPerformed(java.awt.event.Actio
nEvent evt) {
        // TODO add your handling code here:
withdrawlform        obj=new        withdrawlform();
obj.show();
```

```
    }

    private void
exitActionPerformed(java.awt.event.ActionEven t evt)
{
        // TODO add your handling code here:
        System.exit(0);
    }

    private void
exitbtnActionPerformed(java.awt.event.ActionE vent
evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    /**      * @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/uiswi
ng/lookandfeel/plaf.html          */          try {              for
(javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFe els()) {
                if      ("Nimbus".equals(info.getName()))       {
javax.swing.UIManager.setLookAndFeel(info.ge
tClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(mainform.cl
ass.getName()).log(java.util.logging.Level.SEVE RE,
null, ex);          } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(mainform.cl
ass.getName()).log(java.util.logging.Level.SEVE RE,
null, ex);          } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(mainform.cl
ass.getName()).log(java.util.logging.Level.SEVE RE,
null, ex);          } catch
(javax.swing.UnsupportedLookAndFeelExcepti on ex)
{

java.util.logging.Logger.getLogger(mainform.cl
ass.getName()).log(java.util.logging.Level.SEVE RE,
null, ex);
        }
```

```
        //</editor-fold>

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

            }

        });

    }

    // Variables declaration - do not modify
private javax.swing.JMenuItem account;     private
javax.swing.JMenu exit;     private
javax.swing.JMenuItem exitbtn;     private
javax.swing.JMenu jMenu1;     private
javax.swing.JMenuBar jMenuBar1;     private
javax.swing.JMenuItem saving;     private
javax.swing.JMenuItem withdrawl;
    // End of variables declaration
}
```

```
        new NewMember().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     String id=jLabel3.getText();
     String name=jTextField1.getText();
     String mobilenumber=jTextField2.getText();
     String email=jTextField3.getText();
     String gender=(String)jComboBox1.getSelectedItem();
     String fathername=jTextField4.getText();
     String mothername=jTextField5.getText();
     String gymtime=(String)jComboBox2.getSelectedItem();
     String aadharnumber=jTextField6.getText();
     String age=jTextField7.getText();
    String amount=jTextField8.getText();
    try {
     Connection con=ConnectionProvider.getcon();
     PreparedStatement ps=con.prepareStatement("insert into member values (?,?,?,?,?,?,?,?,?,?,?)");
     ps.setString(1, id); ps.setString(2, name); ps.setString(3, mobilenumber);
     ps.setString(4, email); ps.setString(5,
     gender); ps.setString(6, fathername);
     ps.setString(7, mothername);
     ps.setString(8, gymtime); ps.setString(9,
     aadharnumber);
     ps.setString(10, age); ps.setString(11,
     amount); ps.executeUpdate();
     JOptionPane.showMessageDialog(null,"Successfully          Saved");
     setVisible(false); new NewMember().setVisible(true);
     }
    catch(Exception e)
    {
    JOptionPane.showMessageDialog(null,e);
    }
    }
    public      static      void      main(String      args[])      {
        java.awt.EventQueue.invokeLater(new    Runnable()    {
        public void run() { new NewMember().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify private
    javax.swing.JButton jButton1; private javax.swing.JButton
    jButton2; private javax.swing.JButton jButton3; private
    javax.swing.JComboBox<String> jComboBox1; private
    javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JLabel jLabel1; 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.JTextField jTextField1; private
    javax.swing.JTextField jTextField2; private
    javax.swing.JTextField jTextField3; private
    javax.swing.JTextField jTextField4; private
    javax.swing.JTextField jTextField5; private
    javax.swing.JTextField jTextField6; private
    javax.swing.JTextField jTextField7; private
    javax.swing.JTextField jTextField8;
}
```

UpdateDeleteMember.java

```
import java.sql.*;
import javax.swing.JOptionPane; import
project.ConnectionProvider;
public class UpdateDeleteMember extends javax.swing.JFrame {

    public UpdateDeleteMember() { initComponents();
    }

    @SuppressWarnings("unchecked")

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
     setVisible(false);

    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible(false); new UpdateDeleteMember().setVisible(true);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     int checkid=0;
     String id=jTextField1.getText();
     try {
     Connection con=ConnectionProvider.getcon();
     Statement st=con.createStatement();
     ResultSet rs=st.executeQuery("select *from member where id='"+id+"'");
     while(rs.next())
     {
         checkid=1;
         jTextField1.setEditable(false); jTextField2.setText(rs.getString(2));
         jTextField3.setText(rs.getString(3)); jTextField4.setText(rs.getString(4));
         jTextField5.setText(rs.getString(5)); jTextField5.setEditable(false);
         jTextField6.setText(rs.getString(6)); jTextField7.setText(rs.getString(7));
         jTextField8.setText(rs.getString(8)); jTextField8.setEditable(false);
         jTextField9.setText(rs.getString(9));
         jTextField10.setText(rs.getString(10));
         jTextField11.setText(rs.getString(11));
     }
     if(checkid==0)
         JOptionPane.showMessageDialog(null,"Member ID does not Exist");
     }
     catch(Exception e)
     {
     JOptionPane.showMessageDialog(null, e);
     }
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) { String
     id=jTextField1.getText();
     String name=jTextField2.getText();
     String mobilenumber=jTextField3.getText();
     String email=jTextField4.getText();
     String fathername=jTextField6.getText();
     String mothername=jTextField7.getText();
     String aadharnumber=jTextField9.getText();
     String age=jTextField10.getText(); String
     amount=jTextField11.getText(); try {
          Connection con=ConnectionProvider.getcon();
```

ListOfMembers.java

```
import java.sql.*;
import javax.swing.JOptionPane; import
javax.swing.table.DefaultTableModel; import
project.ConnectionProvider; public class ListOfMembers
extends javax.swing.JFrame {

    public ListOfMembers() {
        initComponents();
        DefaultTableModel model=(DefaultTableModel)jTable1.getModel(); try
        {
        Connection con=ConnectionProvider.getcon();
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("select *from member"); while(rs.next())
        {
        model.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),rs.getString(8)
,rs.getString(9),rs.getString(10),rs.getString(11)});
        }
        }
        catch(Exception e)
        {
        JOptionPane.showMessageDialog(null, e);
        }
    }

    @SuppressWarnings("unchecked")


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { setVisible(false);
    }

    public static void main(String args[]) {

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

    // Variables declaration - do not modify private
    javax.swing.JButton jButton1; private
    javax.swing.JLabel jLabel1; private
    javax.swing.JPanel jPanel1; private
    javax.swing.JScrollPane jScrollPane1; private
    javax.swing.JTable jTable1;
    // End of variables declaration
}
```

payment.java

```
import      javax.swing.table.DefaultTableModel;
import  java.sql.*;  import  java.text.DateFormat;
import      java.text.SimpleDateFormat;      import
javax.swing.JOptionPane;                     import
project.ConnectionProvider;                  import
java.util.Date;

public class payment extends javax.swing.JFrame { public void
    tableDetails()
    {
        DefaultTableModel dtm=(DefaultTableModel)jTable1.getModel();
        dtm.setRowCount(0);
        String id=jTextField1.getText();
        try {
        Connection con=ConnectionProvider.getcon();
        Statement st=con.createStatement();
        ResultSet rs=st.executeQuery("Select *from payment where id='"+id+"'");
        while(rs.next())
        {
        dtm.addRow(new Object[]{rs.getString(2),rs.getString(3)});
        }
        }
        catch(Exception e)
        {
        JOptionPane.showMessageDialog(null,e);
        }
    }
     public void date()
        {
        SimpleDateFormat dFormat=new SimpleDateFormat("MM-yyyy");
        Date date=new Date();
        String month=dFormat.format(date); jLabel4.setText(month);
        }
    public payment() { initComponents();
        date();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
     setVisible(false);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
       setVisible(false);
       new payment().setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     tableDetails(); int
     checkid=0;
     String id=jTextField1.getText();
     String month=jLabel4.getText();
     try {
     Connection con=ConnectionProvider.getcon();
     Statement st=con.createStatement();
     ResultSet rs=st.executeQuery("select *from member where id='"+id+"'");
     while(rs.next())
     {
         checkid=1;
         jTextField1.setEditable(false); jTextField2.setText(rs.getString(2));
         jTextField3.setText(rs.getString(3)); jTextField4.setText(rs.getString(4));
         jTextField5.setText(rs.getString(11));
     }
      if(checkid==0)
          JOptionPane.showMessageDialog(null,"Memeber ID does not Exist");
```

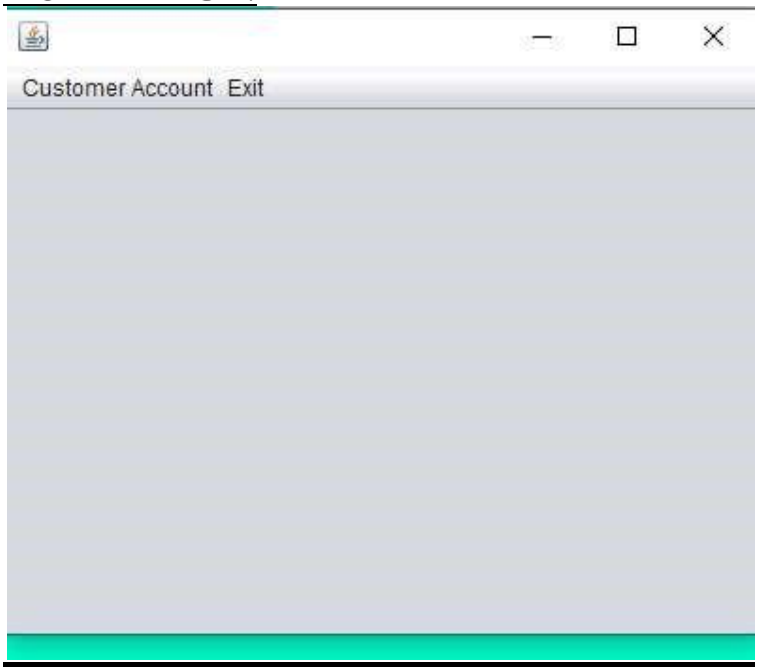
```
    ResultSet rs1=st.executeQuery("select *from payment inner join member where payment.month='"+month+"'and
payment.id='"+id+"'and member.id='"+id+"'"); while(rs1.next())
     {
     jButton3.setVisible(false);
     JOptionPane.showMessageDialog(null,"Payment is already done for this month");
     }
     }
     catch(Exception e)
     {
       JOptionPane.showMessageDialog(null, e);
     }
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
      String id=jTextField1.getText();
      String month=jLabel4.getText(); String
      amount=jTextField5.getText(); try {
      Connection con=ConnectionProvider.getcon();
      PreparedStatement ps=con.prepareStatement("insert into payment values(?,?,?)");
      ps.setString(1, id); ps.setString(2, month); ps.setString(3, amount); ps.executeUpdate();
      tableDetails();
      JOptionPane.showMessageDialog(null,"Successfully Updated"); setVisible(false);
      new payment().setVisible(true);
      }
      catch(Exception e)
            {
            JOptionPane.showMessageDialog(null, e);
            }
    }


    public static void main(String args[]) {

       java.awt.EventQueue.invokeLater(new    Runnable()    {
          public void run() { new payment().setVisible(true);
          }
       });
    }
    // Variables declaration - do not modify private
    javax.swing.JButton         jButton1;         private
    javax.swing.JButton         jButton2;         private
    javax.swing.JButton         jButton3;         private
    javax.swing.JButton         jButton4;         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;          private
    javax.swing.JScrollPane     jScrollPane1;     private
    javax.swing.JTable          jTable1;          private
    javax.swing.JTextField      jTextField1;      private
    javax.swing.JTextField      jTextField2;      private
    javax.swing.JTextField      jTextField3;      private
    javax.swing.JTextField      jTextField4;      private
    javax.swing.JTextField jTextField5;
    // End of variables declaration
}
```

SNAPSHOTS (FROM DESIGN)

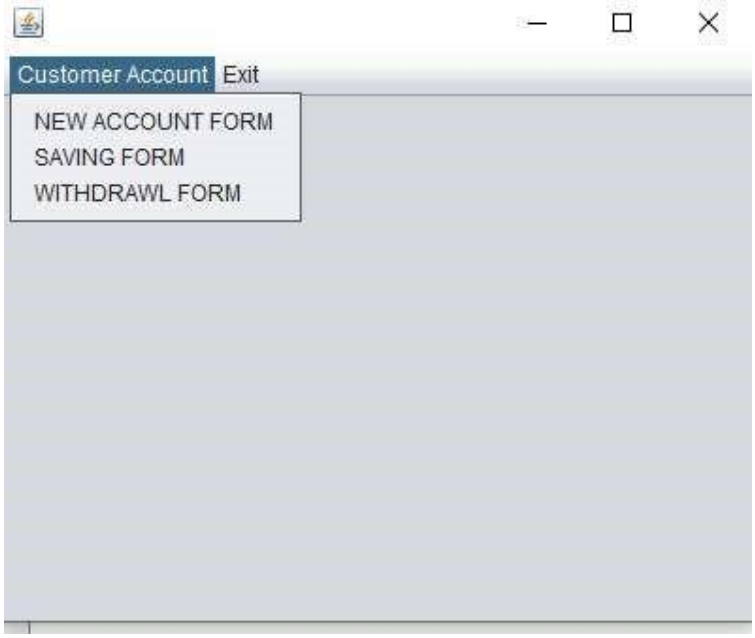
LOGIN PAGE:

SPLASH PAGE:

HOME PAGE:



ADD NEW MEMBER PAGE:



ADD NEW MEMBER PAGE:

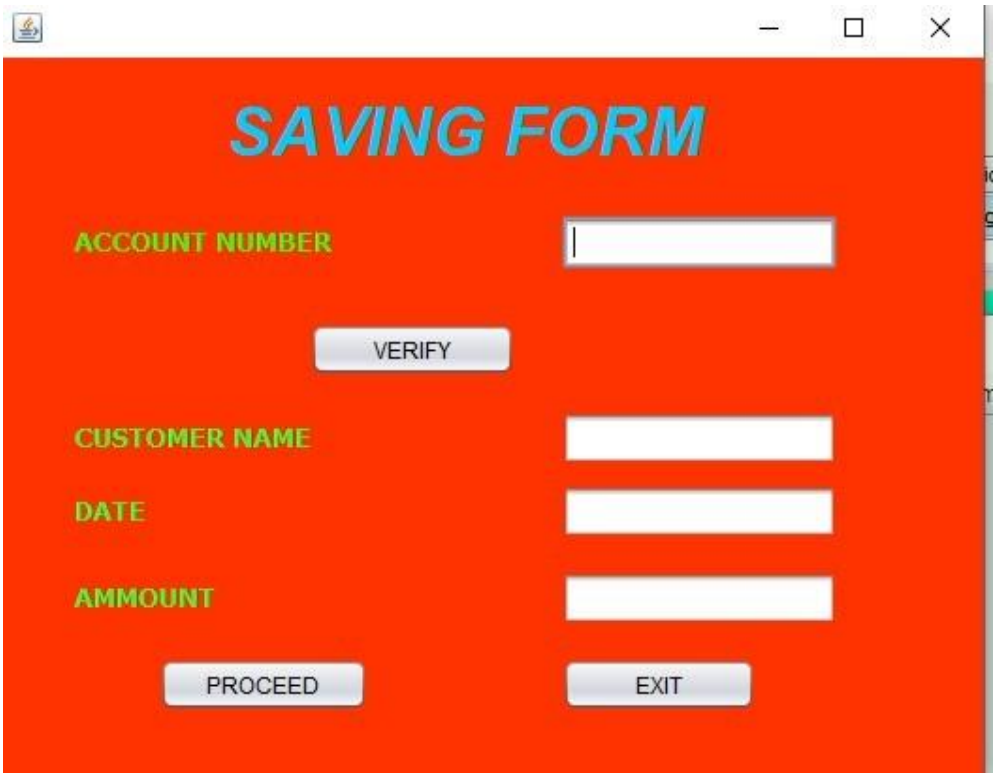
UPDATE & DELETE MEMBER PAGE:

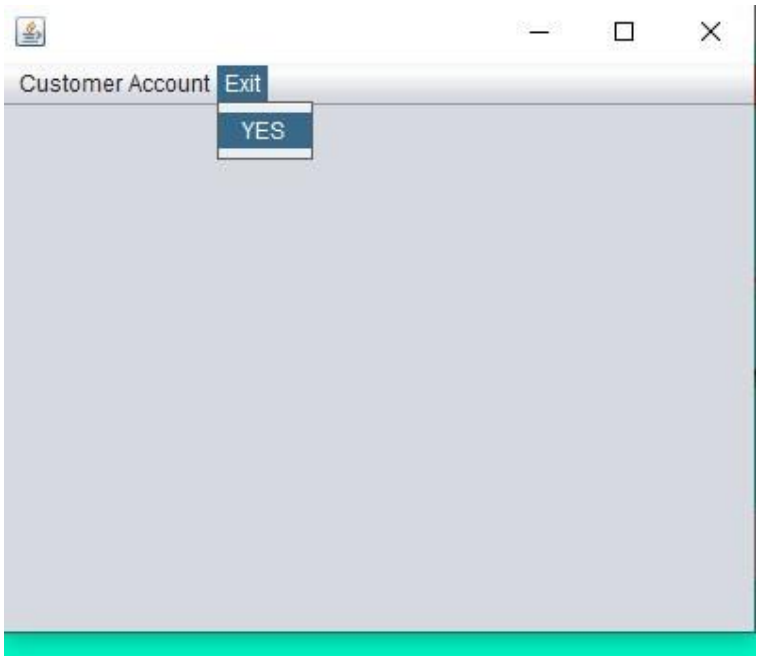

ADD NEW ACCOUNT
ACCOUNT NUMBER
2021
HOLDER NAME
RAJ
ACCOUNT TYPE
SAVING
AGE
28
GENDER
M
CONTACT
7894561230
ID PROOF
ADHAAR
EMAIL ID
MANAB@GMAIL.COM
GUARANTER ACCOUNT NUMBER
786
GUARANTER NAME
RAGKUMAR
BALANCE
10000
ADDRESS
MZN
SAVE
NEXT
EXIT

LIST OF MEMBERS PAGE:

accounttable											
accno	hname	acctype	age	gende r	contact	idproof	email	gaccno	gname	address	bal
1009	raj	saving	15	m	7894562 30	adhaar	hj@gmail.com	786	mohan	mzn	10000
5113 2	rajat	general	56	male	654374	adhar card	rajat@gmail.com	51134	rakesh	prempuri	50000
123	akki	saving	56	male	3467890	adhar card	akki@gmail.com	6789	akash	prempuri	140180
1233	RISHABH	GENERAL	34	MALE	7478843 7	ADHAR CARD	RISHU2@gmail.com	123	MINAL	SAKET	6600
1234	RIA	GENERAL	23	FEMALE	4578303 40	VOTER CARD	RIA2@gmail.com	1234	rishabh	krishnapuri	50000
5113 1	akash	saving	34	male	983765	voter id	akki@gmail.com	51132	rajat	saket	1234
5113 3	minal	general	67	female	768595	voter id	minal@gmail.com	51132	rajat	krishnapuri	8765

WITHDRAWL PAGE:




Customer Account
Exit
YES

IMPLEMENTATION

INTRODUCTION:

A crucial phase in the system life cycle is the successful implementation of the new system design.
Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be completely new, replacing an existing system. In either case, proper implementation becomes necessary so that a reliable system based on the requirements of the organization can be provided. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it. It has been observed that even the best system can't show good result if the analysts managing the implementation do not attend to every important details. This is an area where the systems analysts need to work with utmost care.

For the successful implementation of a system, the training of the users of the system plays an important role. Because even well designed system can succeed or fail because of the way they are operated and used. Therefore, the quality of the training received by the personnel involved with the system in various capacities helps or hinders and may even prevent the successful implementation of management information system.

Those who are directly or indirectly related with the system development work must know in detail what their roles will be, how they can take efficient use of the system and what the system will or will not do for them, therefore both system operators and users need training.

SYSTEMS OPERATORS TRAINING:

Running of the system successfully depend on the personnel working in the computer center. They are responsible for providing the necessary support. Their training must ensure that they are able to handle all possible operations, both routine and extra-ordinary in nature. Operators training should include such fundamentals as how to turn the equipment on and use it, how to power off and a knowledge of what constitutes normal operation. The operators should also be trained on different type of malfunctioning, how what steps should be taken whenever they arise. As part of their to recognize them and what steps should be taken whenever a troubleshooting list that identifies possible training, operators should be given both a troubleshooting list that same problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problem arise. Training also involves familiarization with run procedures, which involves working through the sequence of activities needed to use a new system on an ongoing basis.

USER TRAINING:

User may be trained on use of equipment, particularly in the case where, for example, a
Microcomputer is in use and the individual involved is both operator and user. In such cases, user may be trivial to the analyst, such as how to turn off the equipment without danger of data loss, are significant problems to new user who are not familiar with computers. This is also the case with our system. In most of the cases, user training deals with the operation of the system itself, with properly trained of entering transactions, editing data, formulating inquiries, deleting and inserting of records. No training is complete without familiarizing user with simple systems maintenance activities. Weakness in any aspect of training may lead to awkward situations that cause user frustration and errors.
In our project also it is necessary to train users on how to commit the transactions. Training of operators and users can be organized in several different ways.

Most important ways to cutomers are:

- Vendor and in-service training
- In-house training

Often the best source of training on equipment is the vendor who supplies the equipment. Most vendors offer extensive educational programs as part of their services. In the case of our project too, we'll provide training to the

users of the project. Conversions is also important in the implementation phase. Conversion is the process of changing from the old system to the new one. It must be properly planned and executed. Four methods of conversions are common in use.

They are:

1. Parallel systems
2. Direct conversion
3. Pilot systems
4. Systems phase-in

Each method should be considered in the light of the opportunities that it offers and problems that it may create. However, it may be possible that sometimes, we be forced to apply one method over others, even though other methods over others, even though other methods may be more beneficial. In general systems conversions should be accomplished in shortest possible time. Long conversion periods create problems for all persons for all persons involved including both analysts and users.

PARALLEL SYSTEMS:

The most secure method of converting from an old to new system is to run both systems parallel. Under this approach, users continue to operate the old system in the usual manner but they also start using the new system. This method is the safest one.

DIRECT CONVERSION:

This method converts from the old systems to the new system abruptly, sometimes over a weekend or even midnight. The old system is used until a planned conversion day, when it is replaced by the new system. There are not parallel activities.

PILOT SYSTEM:

Pilot approach is often preferred in the case of the new system which involves new system which involves new techniques or some drastic changes in organization performance. In this method, a working version of the system is implemented in one part of the organization, such as a single work area or department.

PHASE-IN METHOD:

This method is used when it is not possible to install a new system throughout an organization all at once. It allows some users to take advantage of the new system early. Also it allows training and installation without unnecessary use of resources. The project developed by us should be implemented using the parallel systems conversion method.

MAINTENANCE

The maintenance phase is the last phase of the system development life cycle. Yet a life cycle is circular in that last activity leads back to the first. This means that the process of maintaining an information system is the process of returning to the beginning of the system development life cycle and repeating development steps until the change is implemented.

Four major activities occur within maintenance:

- Obtaining maintenance requests
- Transforming requests into changes
- Designing changes
- Implementing changes

Obtaining maintenance request that a formal process be established whereby user can submit change requests. Most companies have some sort of document to request new development, to report problem, or to request new system features with an existing system.

Once a request is received, analysis must be conducted to gain an understanding of the scope of the request. It must be determined how the request will affect the current system and duration of such project. Next, a

change request can be transformed into a formal design change, which can be fed into the maintenance implementation phase. There are several types of maintenance that you can perform on an information system.

These are:

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

CORRECTIVE MAINTENANCE:

Corrective maintenance refers to the changes made to repair defects in the design, coding, or implementation of the system. Most corrective maintenance problems surface soon after installation. When the corrective maintenance problem surface, they are typically urgent and need to be resolved to curtail possible interruptions in normal business activities.

ADAPTIVE MAINTENANCE:

Adaptive maintenance involves making changes to an information system to evolve its functionality to changing

business needs or to migrate it to a different operating environment. Contrary to Corrective maintenance, adaptive maintenance is generally a small part of an organization's maintenance effort but does add value to the organization.

PERFECTIVE MAINTENANCE:

Perfective maintenance involves making enhancement to improve processing performance, interface usability or to add desired, but not necessarily required, system features.

PREVENTIVE MAINTENANCE:

Preventive maintenance involves changes made to a system to reduce the chance of future system failure. Preventive maintenance might be used to increase the number of records that a system can process far beyond what is currently needed.

SECURITY MEASURES TAKEN

The dictionary definition of security encompasses a set of measures taken to guard against theft, attack, crime and sabotage. Here, security measures include control and monitoring physical access to the software.

Authentication is one of the physical security measures taken to deny access to unauthorized users and to allow legitimate users. Since unauthorized access or destruction of data can violet individual's privacy. Corruption or theft of business data can result in significant and potential catastrophic loss or mills. So, in the software under this project, the most common authentication mechanism namely "Password" is applied to prevent from any unauthorized access.

In the software, when logging in, it is a request to the user for a secret and user specific password.

The software matches the user supplied password. If it matches with software inherent coded password, then the main menu of the software is available to the user to access otherwise it denies and asks to enter right password. Similarly, an authentication known as Access Groups is applied to protect the table created in oracle environment from unauthorized access.

CONCLUSION & FUTURE SCOPE

CONCLUSION:

Banking System project objective of this project was to build a program for maintaining banking system project details of all banking system project customers and inventory banking system project system developed is able to meet all banking system project basic requirements. Banking system project management of banking system project records will be also benefited by banking system project proposed system, as it will automate banking system project whole procedure, which will reduce banking system project workload. Banking system project security of banking system project system is also one of banking system project prime concerns.

Banking system project is always a room for improvement in any software, however efficient banking system project system may be. Banking system project important thing is that banking system project system should be flexible enough for future modifications. banking system project system has been factored into different modules to make system adapt to banking

system project further system project changes. Every effort has been made to cover all user requirements and make it user friendly.

- Goal achieved: banking system project System is able provide banking system project interface to banking system project owner so that he can replicate his desired data.

- User friendliness: Though banking system project most part of banking system project system is supposed to act in banking system project background, efforts have been made to make banking system project foreground interaction with user(owner) as smooth as possible.

FUTURE SCOPE:

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

BIBLIOGRAPHY

BOOKS:

- Effective Java by Joshua Bloch
- Head First Java by Kathy Sierra & Bert Bates
- Java: A Beginner's Guide by Herbert Schildt
- Java for Dummies by Barry A. Burt
- Clean Code by Robert C. Martin
- Thinking in Java by Bruce Eckel
- Java: The Complete Reference by Herbert Schildt
- Murach's MySQL 3rd Edition by Joel Murach
- MySQL Explained: Your Step By Step Guide To Database Design by Andrew Comeau
  MySQL 8 Administrator's Guide

ONLINE REFERENCE:

- www.youtube.com
- www.w3schools.com
- www.wilipedia.com
- www.geeksforgeeks.com
- www.beginnersbook.com