Object Oriented Programming

Project Report:

Gym Management System

Name of Students	Roll no
Hafiz Muhammad Tausif	2023F-BSE-091
Syed Sarmad Arshad Hussaini	2023F-BSE-059
Anees Ahmed	2023F-BSE-080
Sufiyan Imran	2023F-BSE-094

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

The purpose of this project is to develop a comprehensive Gym Management System using MySQL, JFrame, and NetBeans IDE. This system aims to streamline the administrative and operational processes within a gym, including member registration, attendance tracking, equipment management, and staff coordination.

1.2 Project Scope

The scope of this project includes:

- Developing a user-friendly interface for gym members and staff.
- Implementing a database for storing member and equipment information.
- Creating functionalities for membership management, attendance tracking, and equipment maintenance.
- Ensuring data security and integrity.

1.3 References

- MySQL Documentation
- Java SE Documentation
- NetBeans IDE Documentation
- JFrame Tutorials

2. Overall Description

2.1 Product Perspective

The Gym Management System is designed to replace manual record-keeping processes with a digital solution that offers real-time updates and comprehensive reporting capabilities. It serves as an all-in-one platform for managing various aspects of gym operations.

2.2 Product Features & Flowchart

- **Member Registration**: Allows new members to register and existing members to update their details.
- **Attendance Tracking**: Monitors member attendance and generates reports.
- **Payment Management**: Tracks the usage and maintenance schedule of gym member payments.

2.3 Design and Implementation

The system is built using the Java programming language with JFrame for the user interface and MySQL for the database. NetBeans IDE is used for development and project management.

2.4 Assumptions and Dependencies

- Users have basic knowledge of operating a computer and using software applications.
- The system will be hosted on a local server.
- The gym has reliable internet connectivity for database operations.

3. Tools and Technologies

3.1 Tool Name & Description

- **MySQL**: Used for database management to store and retrieve gym-related data.
- **JFrame**: A GUI toolkit for Java to create the application's user interface.

- **NetBeans IDE**: An integrated development environment for writing, compiling, and debugging the project.

4. External Interface Requirements

4.1 User Interfaces

- **Login Screen**: For user authentication.
- **Dashboard**: Provides access to various functionalities of the system.
- **Member Management**: Interface for adding, updating, and viewing member details.
- **Payment Tracking**: Interface for recording and viewing member Payment.
- **Reports**: Interface for generating and viewing reports.
-

4.2 Hardware Interfaces

- A computer system with at least 4GB RAM and 500GB HDD.
- A stable internet connection for database access.
- XXAMP SERVER for MYSQL

4.3 Software Interfaces

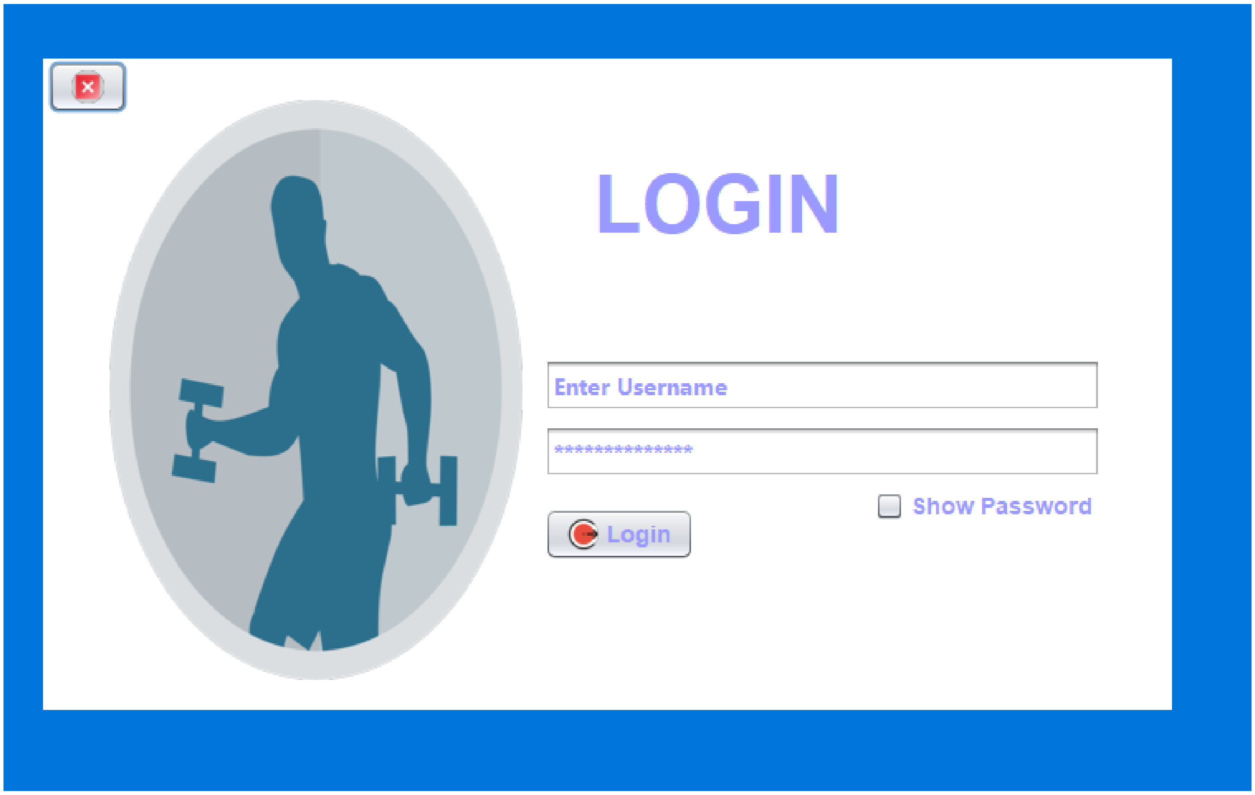
- Java Runtime Environment (JRE)
- MySQL Server
- NetBeans IDE

4.4 Communications Interfaces

- Local Area Network (LAN) for connecting to the MySQL server.

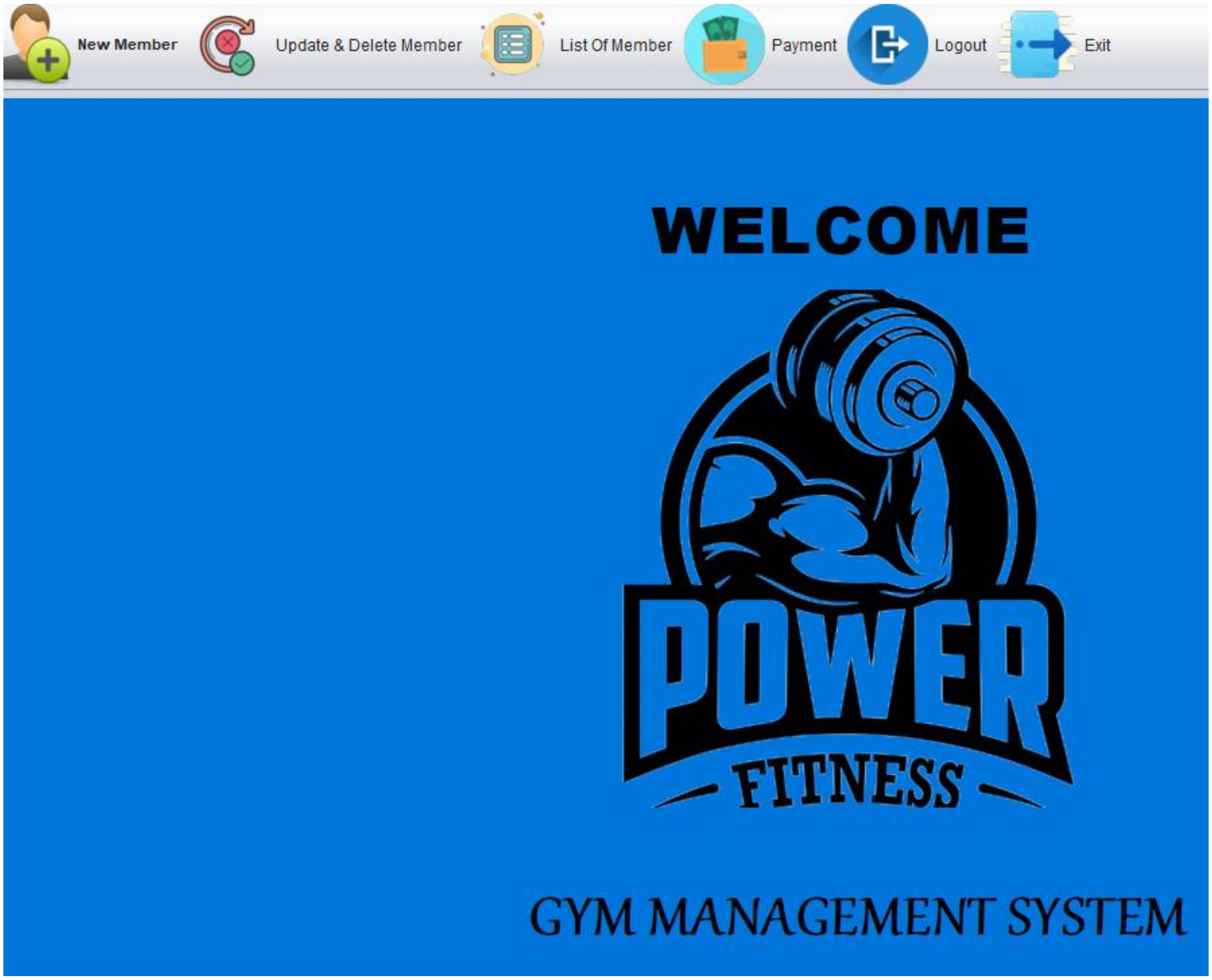
5. DESIGN

5.1 Login Page:



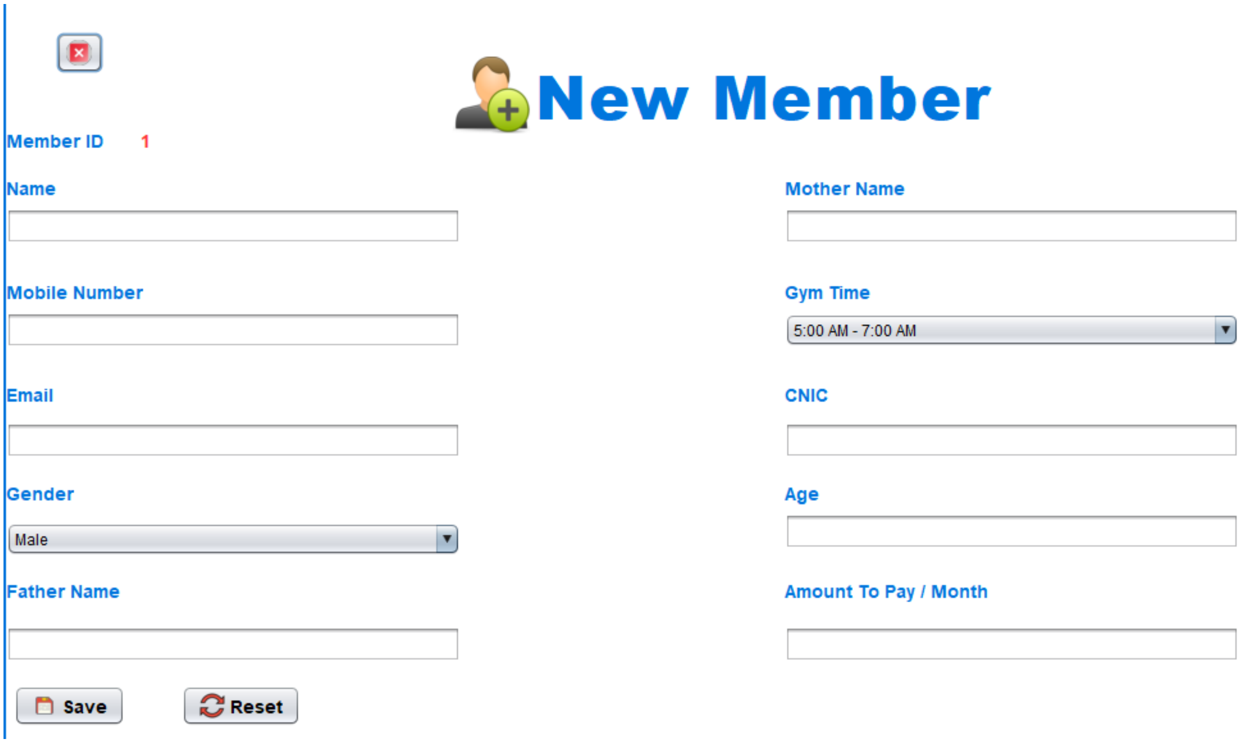
- Proper Authentication Applied
- Username & Password set to “gms” , “admin” respectively

5.2 Dashboard:



- Home Page, Navigator Of whole web-app

5.3 Create New Member:



New Member

Member ID 1

Name

Mother Name

Mobile Number

Gym Time: 5:00 AM - 7:00 AM

Email

CNIC

Gender: Male

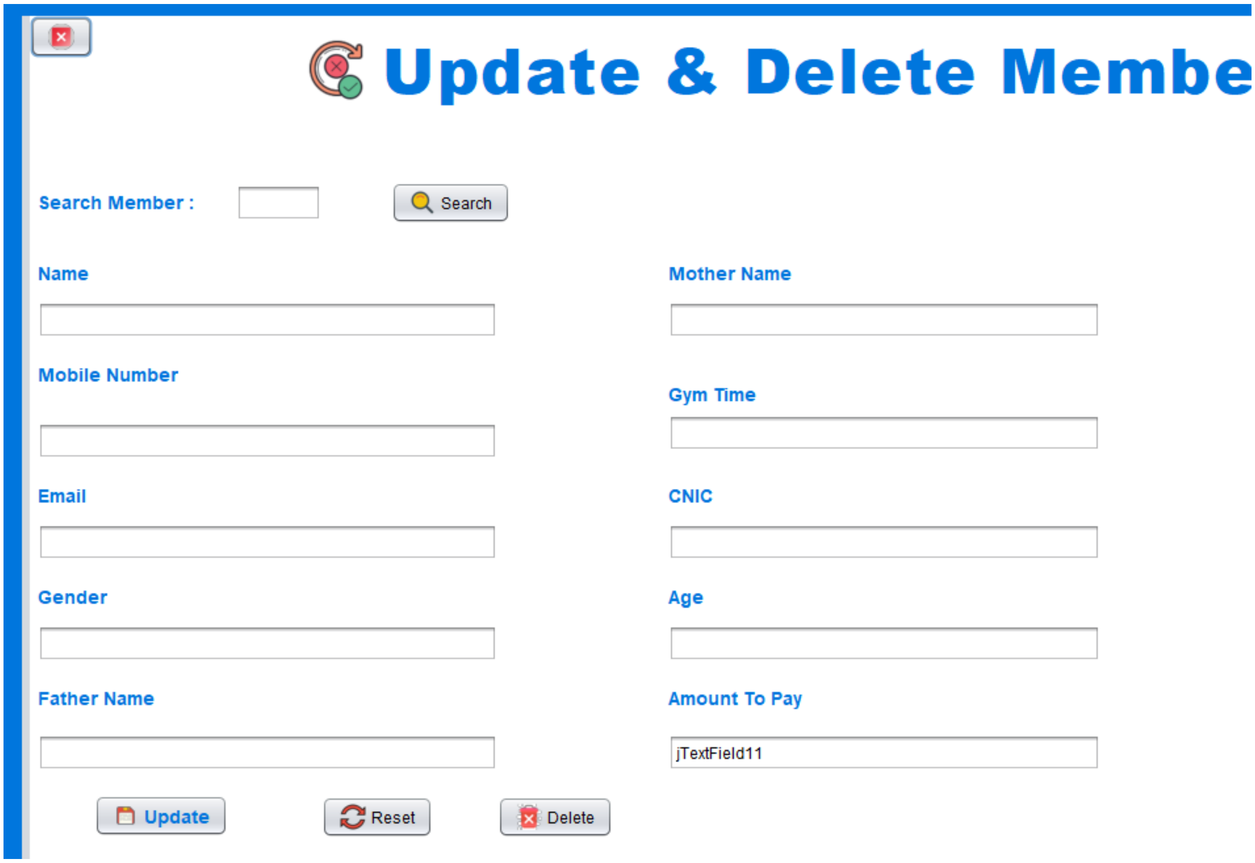
Age

Father Name

Amount To Pay / Month

- Page Aimed to create New Member Through form
- The information Directly goes into the Database

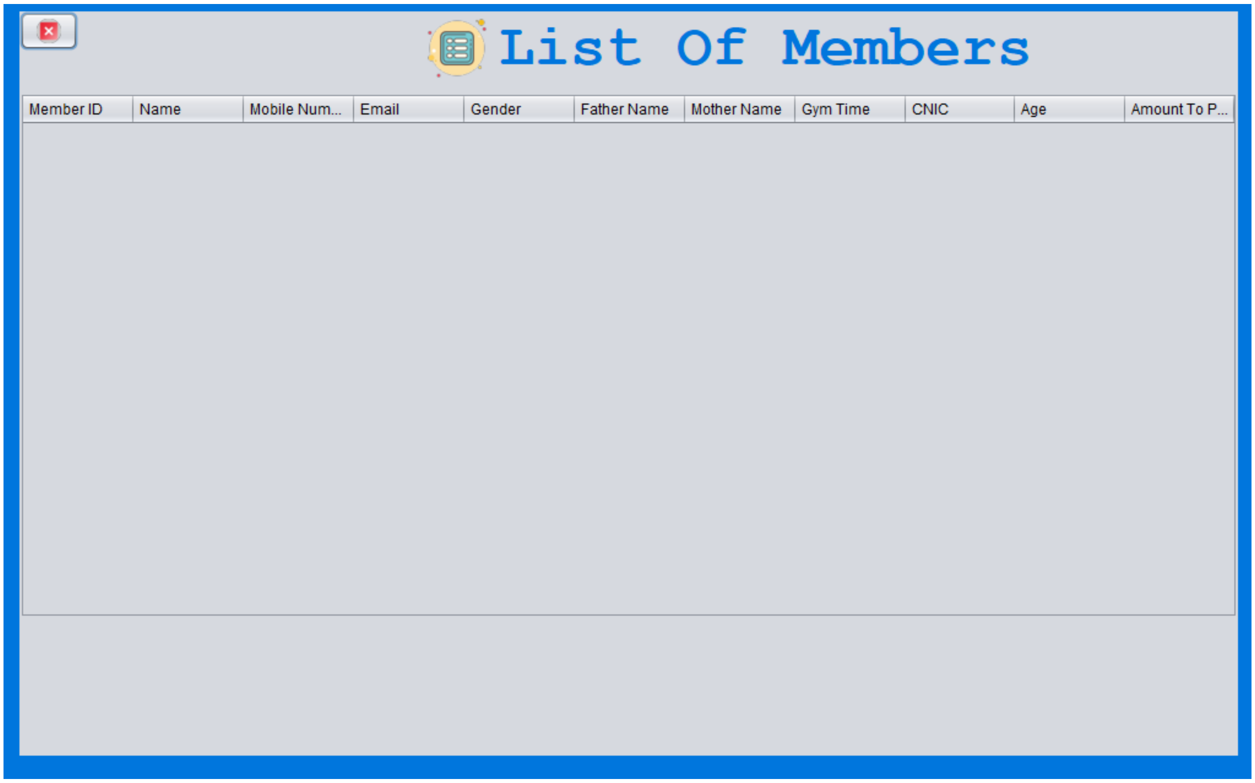
5.4 Update & Delete Page



Amount To Pay: jTextField11

- Page dedicated to Update & Delete Members

5.5 List Of Members

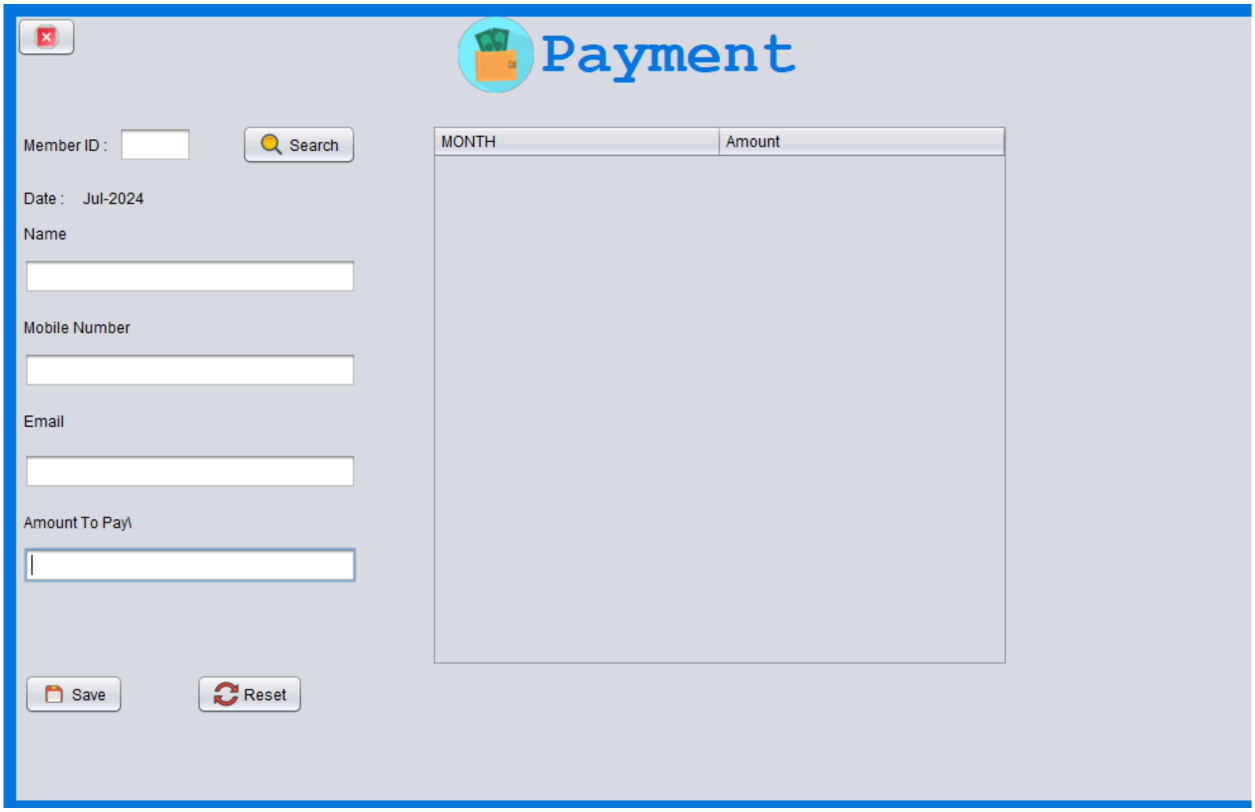


List Of Members

Member ID	Name	Mobile Num...	Email	Gender	Father Name	Mother Name	Gym Time	CNIC	Age	Amount To P...

- Page that shows Database Values Directly to the Application

5.6 Payment Management



Member ID :

Date : Jul-2024

Name

Mobile Number

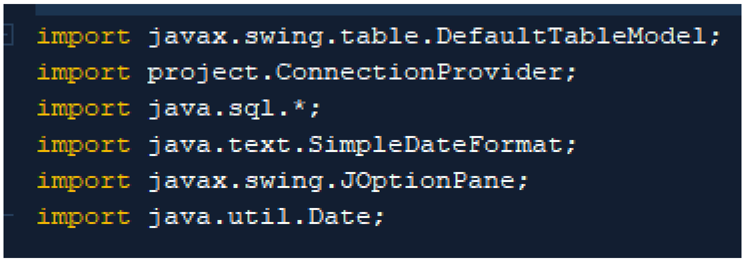
Email

Amount To Pay\

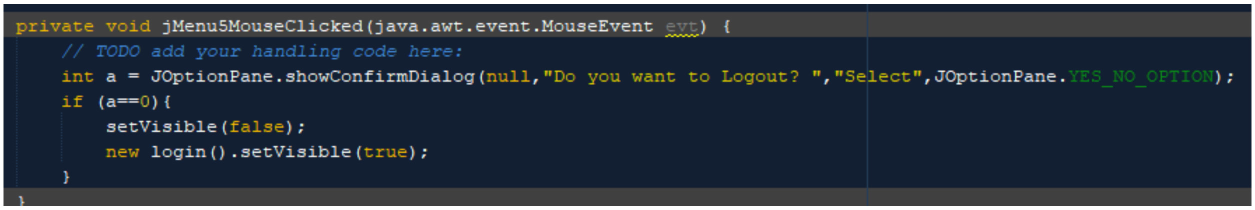
MONTH	Amount

- Manages Payments Of members of gym With Proper Date And Timing

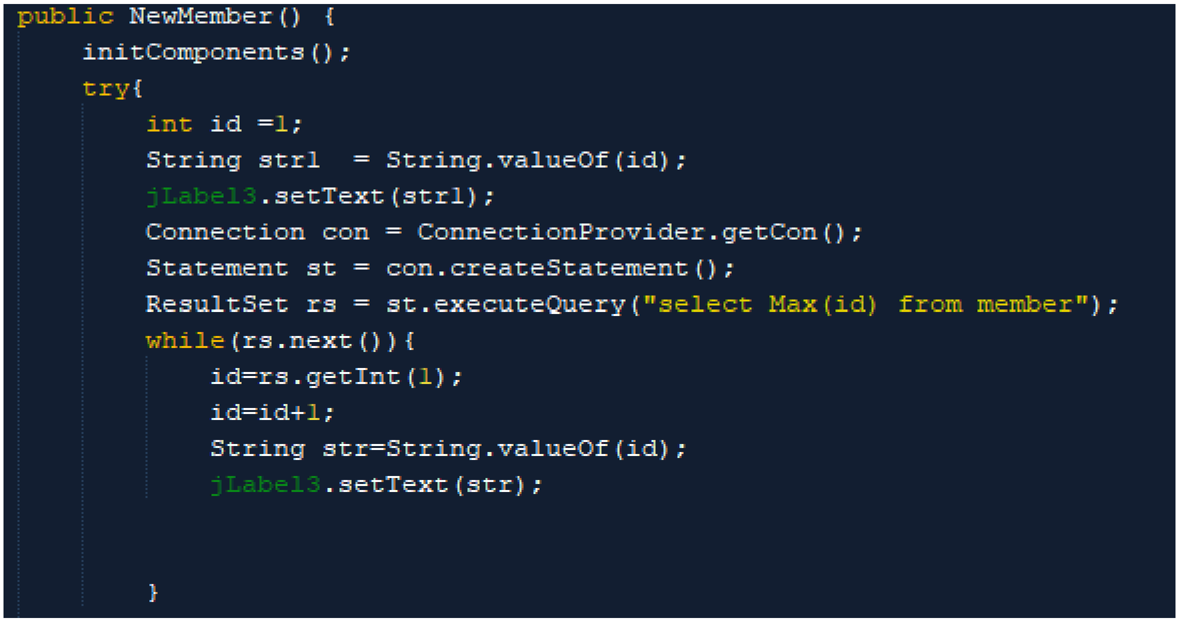
6.Code:



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

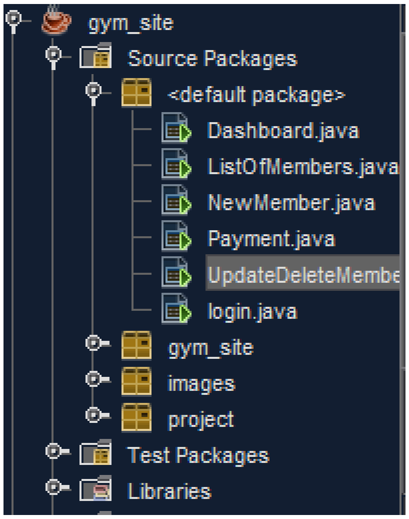


```
private void jMenu5MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int a = JOptionPane.showConfirmDialog(null,"Do you want to Logout? ","Select",JOptionPane.YES_NO_OPTION);
    if (a==0){
        setVisible(false);
        new login().setVisible(true);
    }
}
```

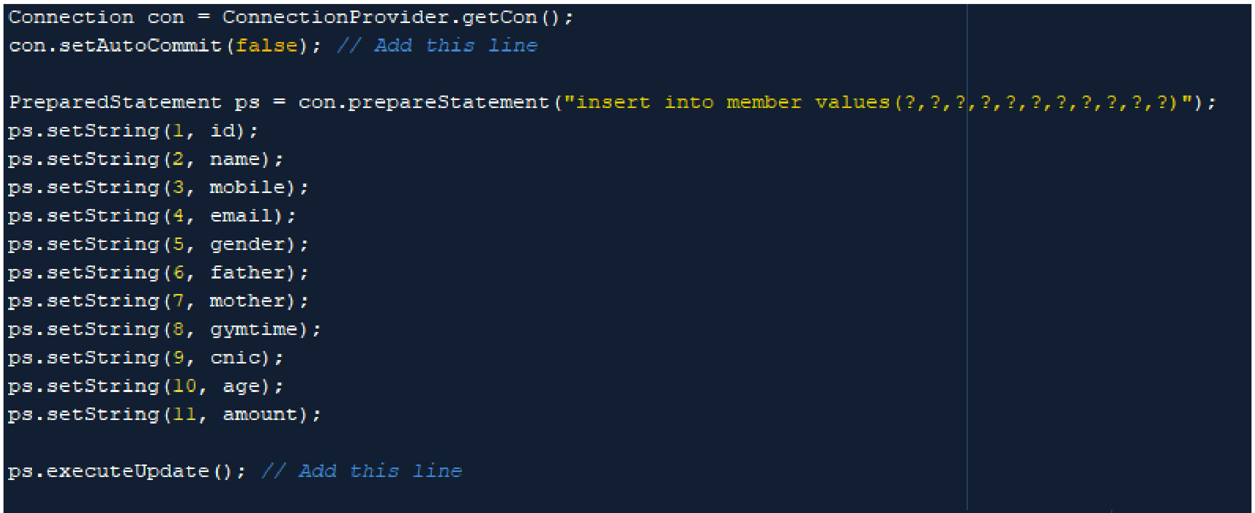


```
public NewMember() {
    initComponents();
    try{
        int id =1;
        String str1  = String.valueOf(id);
        jLabel3.setText(str1);
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery("select Max(id) from member");
        while(rs.next()){
            id=rs.getInt(1);
            id=id+1;
            String str=String.valueOf(id);
            jLabel3.setText(str);


        }
```



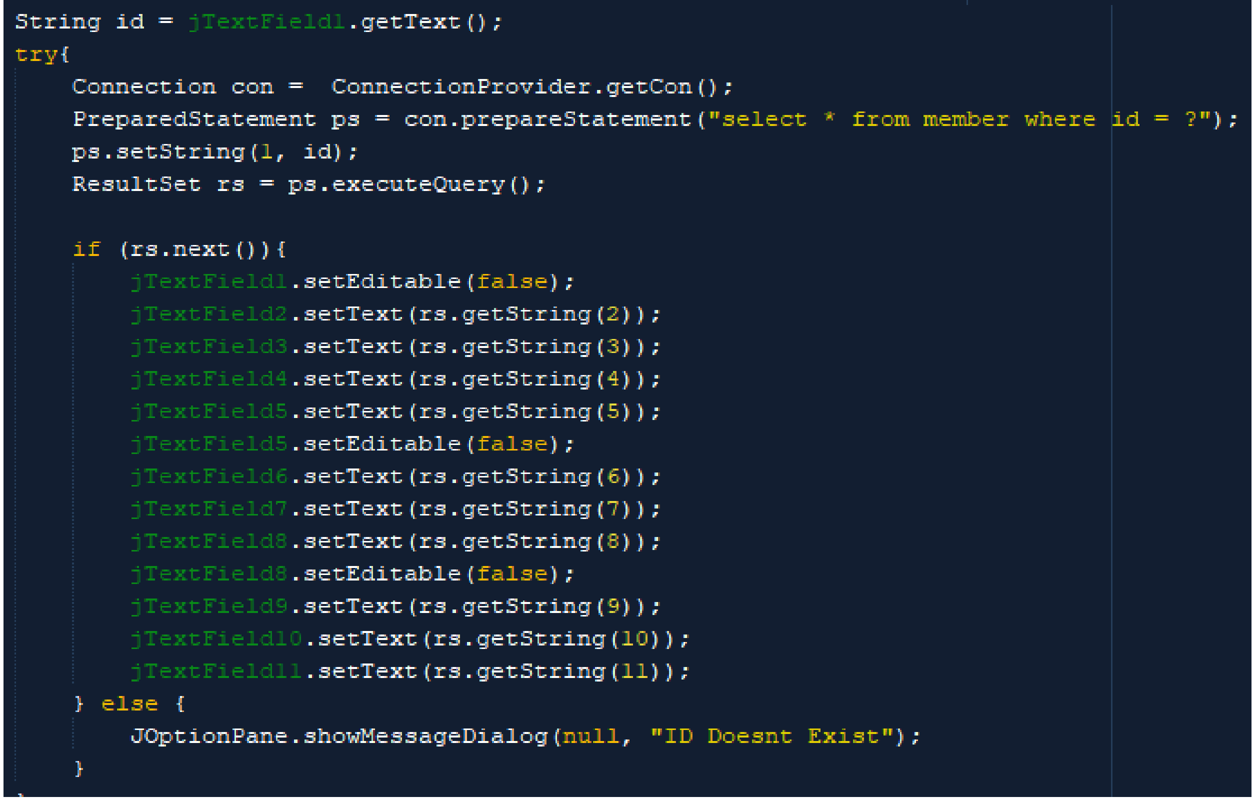
```
gym_site
    Source Packages
        <default package>
            Dashboard.java
            ListOfMembers.java
            NewMember.java
            Payment.java
            UpdateDeleteMembe
            login.java
        gym_site
        images
        project
    Test Packages
    Libraries
```



```
Connection con = ConnectionProvider.getCon();
con.setAutoCommit(false); // Add this line

PreparedStatement ps = con.prepareStatement("insert into member values(?,?,?,?,?,?,?,?,?,?,?)");
ps.setString(1, id);
ps.setString(2, name);
ps.setString(3, mobile);
ps.setString(4, email);
ps.setString(5, gender);
ps.setString(6, father);
ps.setString(7, mother);
ps.setString(8, gymtime);
ps.setString(9, cnic);
ps.setString(10, age);
ps.setString(11, amount);

ps.executeUpdate(); // Add this line
```



```
String id = jTextField1.getText();
try{
    Connection con =  ConnectionProvider.getCon();
    PreparedStatement ps = con.prepareStatement("select * from member where id = ?");
    ps.setString(1, id);
    ResultSet rs = ps.executeQuery();

    if (rs.next()){
        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));
    } else {
        JOptionPane.showMessageDialog(null, "ID Doesnt Exist");
    }
```