

Sir Syed University of Engineering & Technology
(SSUET)

Software Engineering Department

Course Name: Software Construction & Development (SWE-312)

Semester: 5th

Batch: 2022F

Section: B

PROJECT REPORT

Project Title: Payroll Management System



Submitted By:

STUDENTS NAMES

Muhammad Salman Khan (2022F-BSE-098)

Muhammad Okasha Mohsin (2022F-BSE-072)

Muhammad Kamil (2022F-BSE-086)

Submitted To:

Ms. Eman Razzaq

TABLE OF CONTENTS

1	PROJECT DESCRIPTION	4
2	CLASS DIAGRAM	5
3	THREAD SYNCHRONIZATION IN YOUR PROJECT	6
4	CONCEPT OF MUTABILITY AND IMMUTABILITY IN YOUR PROJECT	6
5	EXCEPTION HANDLING IN YOUR PROJECT	7,8,9
6	UNIT TESTING PERFORMED IN YOUR PROJECT	10
7	USER GUIDE	11,12,13,14

TEAM PROFILE

- 1. Muhammad Salman Khan (2022F-BSE-098)**
(Coding & Development)
- 2. Muhammad Okasha Mohsin (2022F-BSE-072)**
(Coding & Development)
- 3. Muhammad Kamil (2022F-BSE-086)**
(Documentation)

1. PROJECT DESCRIPTION

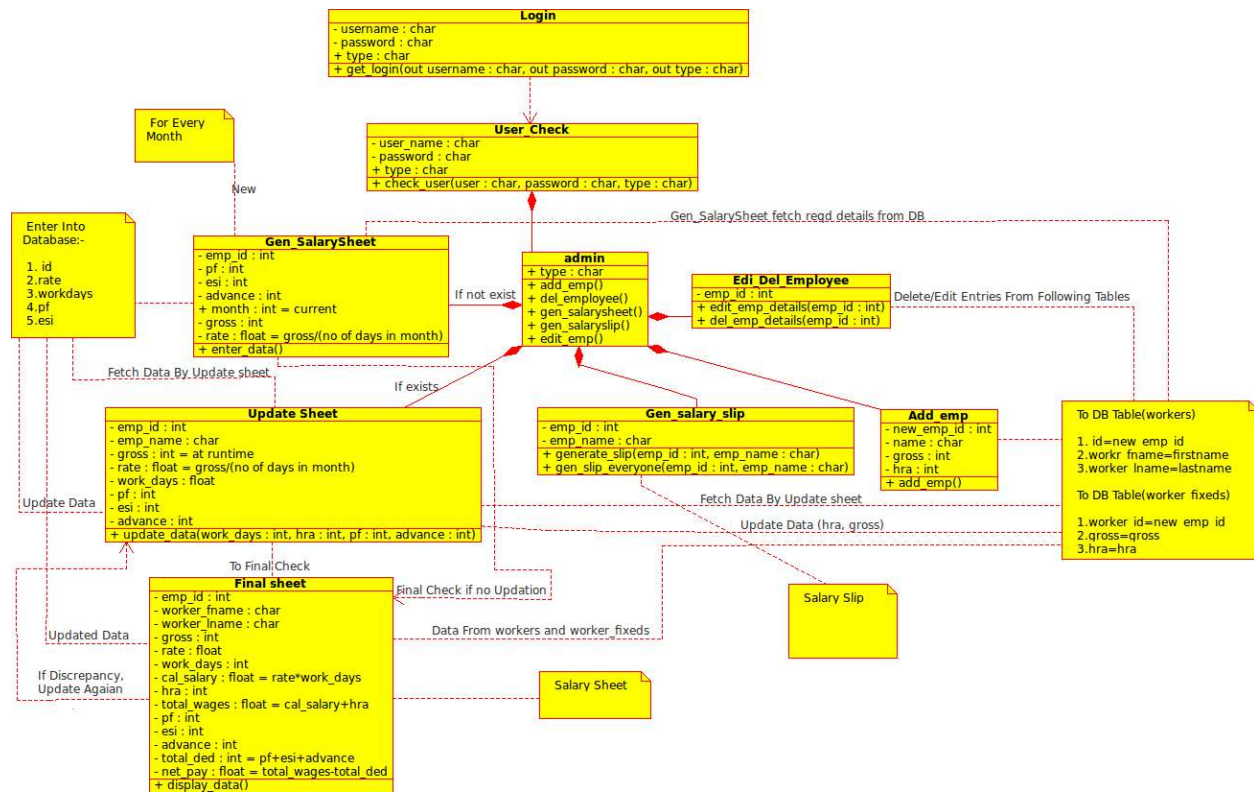
Payroll Management System Project in Java Netbeans is built fully in Java/MySQL Database. It has full-featured Graphical User Interface (GUI) with all the basic functionalities of a typical Payroll System used the company. With this **payroll management system in Java Netbeans IDE** intended to help the company manage their employee payroll efficiently.

OBJECTIVES:

The main objectives of the Payroll Management System is:

1. Add Update Delete Search
2. Employees from the System
3. Add Update Delete
4. Salary details
5. Add Update Delete
6. Leave details
7. Generate and Print Pay Slips
8. Database file is also included
9. GUI is done using Java Swing Framework

2. CLASS DIAGRAM



3. THREAD SYNCHRONIZATION IN YOUR PROJECT

```
// 1. Multithreading: A method that retrieves details in a separate thread.
public class PersonThread extends Thread {
    private Person person;

    public PersonThread(Person person) {
        this.person = person;
    }

    @Override
    public void run() {
        System.out.println("Person details: " + person.getFname() + " " + person.getLname());
    }
}
```

4. CONCEPT OF MUTABILITY AND IMMUTABILITY IN YOUR PROJECT

```
// 4. Mutability and Immutability: Immutable class for specific person details.
public final class ImmutablePerson {
    private final String name;
    private final String address;

    public ImmutablePerson(String name, String address) {
        this.name = name;
        this.address = address;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }
}
```

5. EXCEPTION HANDLING IN YOUR PROJECT

```

public boolean getEmployeeDetails(String empId) {

    try {
        String sql = "SELECT * FROM employee WHERE empId='" + empId + "'";

        ResultSet rs = objSQLRun.sqlQuery(sql);

        if (rs.next()) {

            this.empId = rs.getInt("empId");
            nic = rs.getString("nic");
            address = rs.getString("address");
            city = rs.getString("city");
            dateOfJoining = rs.getString("date_of_joining");
            department = rs.getString("department");
            designation = rs.getString("designation");
            dob = rs.getString("dob");
            fName = rs.getString("fname");
            lName = rs.getString("lname");
            gender = rs.getString("gender");
            telHome = rs.getString("tel_home");
            telMobile = rs.getString("tel_mobile");
            salType = rs.getString("salType");
            salAmount = rs.getInt("salAmount");

            return true;

        } else {

```

```

        } else {

            JOptionPane.showMessageDialog(null, "No record found for Employee ID:" + empId, "ERROR", 0);
            return false;

        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error! Failed to Retrieve Data! Please Contact Your System Administrator!\n\n" + ex.getMessage(), "ERROR", 0);
        return false;
    }
}

```

```

//automatically populate employee id field when inserting a new employee
public String setEmpIdField() {

    try {

        String sql = "SELECT MAX(empId) FROM employee";
        ResultSet rs = objSQLRun.sqlQuery(sql);

        if (rs.next()) {
            int eId = rs.getInt(1);
            eId++;
            return String.valueOf(eId);
        } else {
            JOptionPane.showMessageDialog(null, "Error! Please Contact Your System Administrator!", "ERROR", 0);
            return null;
        }

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error! Please Contact Your System Administrator!\n\n" + ex.getMessage(), "ERROR", 0);
        return null;
    }

}

public ResultSet getAllEmployeeDetails(String sql) {

    try {
        ResultSet rs = objSQLRun.sqlQuery(sql);
        return rs;
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Error! Failed to Retrieve Data! Please Contact Your System Administrator!\n\n" + ex.getMessage(), "ERROR", 0);
        return null;
    }

}

public Vector getColumnNames(ResultSet rs) {
    try {
        ResultSetMetaData rsMeta = rs.getMetaData();
        int columnCount = rsMeta.getColumnCount();
        Vector columns = new Vector();
        for (int i = 1; i <= columnCount; i++) {
            columns.addElement(rsMeta.getColumnName(i));
        }
        return columns;
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error! Failed to Retrieve Data! Please Contact Your System Administrator!\n\n" + ex.getMessage(), "ERROR", 0);
        return null;
    }
}

```



```

public Vector getColumnNames(ResultSet rs) {
    try {
        ResultSetMetaData rsMeta = rs.getMetaData();
        int columnCount = rsMeta.getColumnCount();
        Vector columns = new Vector();
        for (int i = 1; i <= columnCount; i++) {
            columns.addElement(rsMeta.getColumnName(i));
        }
        return columns;
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error! Failed to Retrieve Data! Please Contact Your System Administrator!\n\n" + ex.getMessage(), "ERROR", 0);
        return null;
    }
}

//overloaded method
public Vector getAllEmployeeDetails(ResultSet rs) {
    try {
        ResultSetMetaData rsMeta = rs.getMetaData();
        int columnCount = rsMeta.getColumnCount();

        Vector data = new Vector();

        while (rs.next()) {
            Vector row = new Vector();
            for (int i = 1; i <= columnCount; i++) {
                row.addElement(rs.getObject(i));
            }
            data.addElement(row);
        }
    }
}

```

6. UNIT TESTING PERFORMED IN YOUR PROJECT

```
package model;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class DbConnection {

    private static Connection conn = null;

    // Define JDBC driver
    private static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";

    // Database credentials
    private static final String DB_USER_NAME = "root";
    private static final String DB_PASSWORD = "root123";

    // Database details
    private static final String DB_NAME = "payroll";
    private static final String DB_URL = "jdbc:mysql://localhost:3306/" + DB_NAME + "?useSSL=false&serverTimezone=UTC";

    public static Connection getDbConnection() {
        try {
            // Load MySQL JDBC Driver
            Class.forName(JDBC_DRIVER);

            // Establish Connection
            conn = DriverManager.getConnection(DB_URL, DB_USER_NAME, DB_PASSWORD);
            System.out.println("Database Connected Successfully!");
            return conn;

        } catch (ClassNotFoundException ex1) {
            JOptionPane.showMessageDialog(null, "Error! MySQL Driver Not Found!\n\n" + ex1.getMessage(), "ERROR", JOptionPane.ERROR_MESSAGE);
            return null;

        } catch (SQLException ex2) {
            JOptionPane.showMessageDialog(null, "Error! Database Connection Failed!\n\n" + ex2.getMessage(), "ERROR", JOptionPane.ERROR_MESSAGE);
            return null;

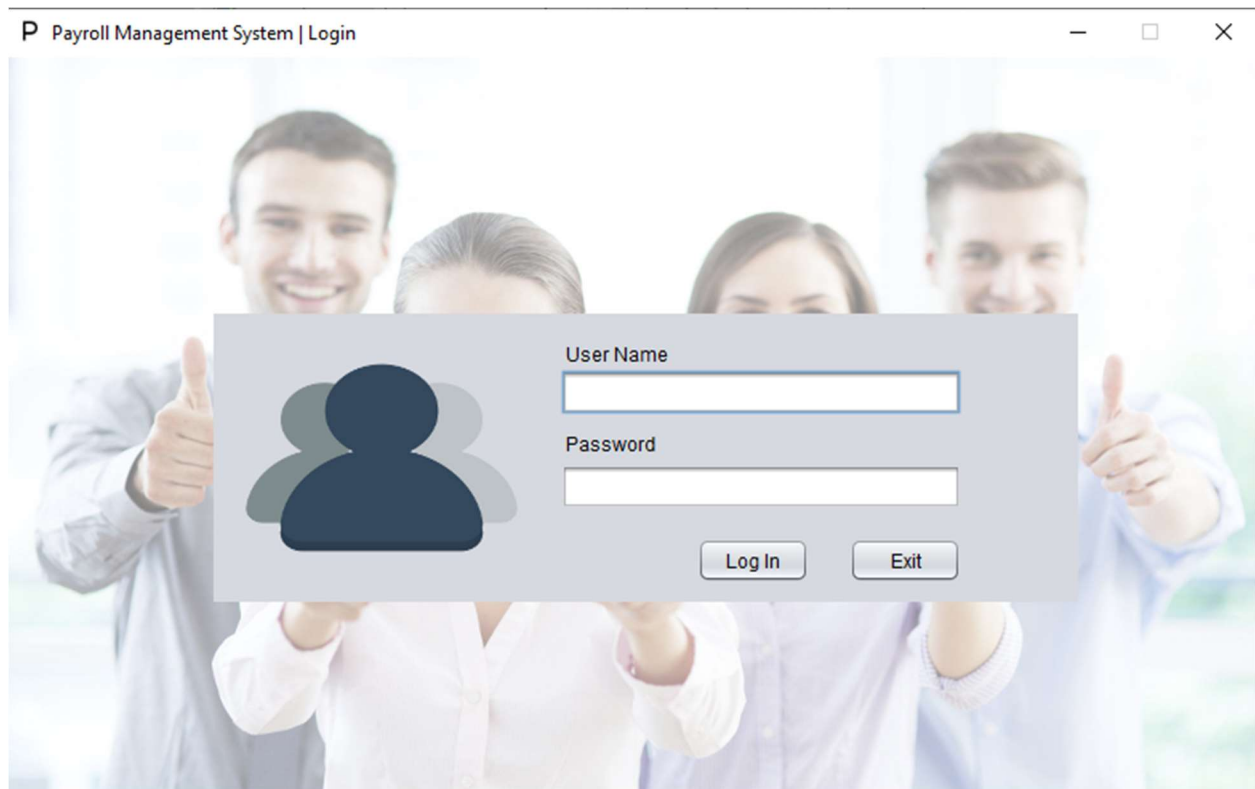
        } catch (Exception ex3) {
            JOptionPane.showMessageDialog(null, "Unexpected Error!\n\n" + ex3.getMessage(), "ERROR", JOptionPane.ERROR_MESSAGE);
            return null;
        }
    }
}
```

Output - PayrollSystem (run)

```
run:
Database Connected Successfully!
```

7. USER GUIDE

LOGIN WINDOW:



ADMIN WINDOW:



Admin can add, remove and edit information of

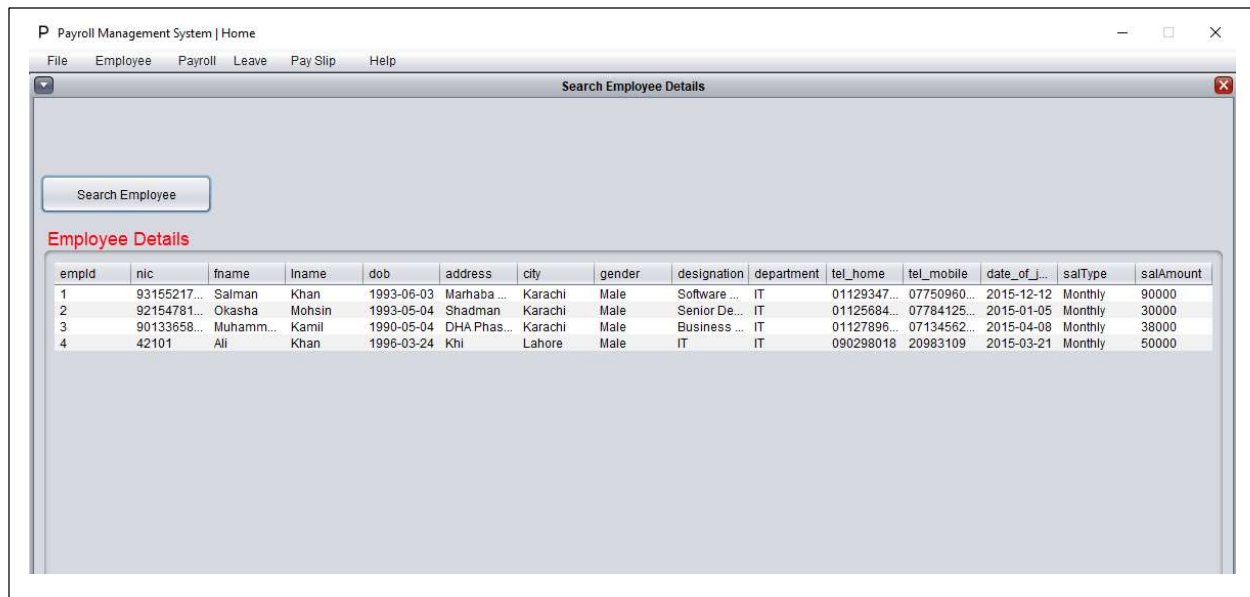
- New Employee,

The screenshot shows a window titled "Enter Employee Details" from the "Payroll Management System | Home". The window has a menu bar with "File", "Employee", "Payroll", "Leave", "Pay Slip", and "Help". The main area is divided into two sections: "New Employee Details" on the left and "Salary Details" on the right. The "New Employee Details" section contains a vertical list of input fields: "Employee ID*", "NIC*", "First Name*", "Last Name*", "Date of Birth*", "Address", "City", "Contact No" (with sub-fields for "Home" and "Mobile"), "Gender" (with radio buttons for "Male" and "Female"), "Designation*", "Department*", and "Date of Joining*". The "Salary Details" section contains a sub-section "Basic Salary" with a "Salary Type" dropdown menu (set to "Monthly") and a "Salary Amount" input field. At the bottom right, there are two buttons: "Add Employee" and "Exit".

- Update Employee Details,

The screenshot shows a window titled "Update Employee Details" from the "Payroll Management System | Home". The window has a menu bar with "File", "Employee", "Payroll", "Leave", "Pay Slip", and "Help". The main area is divided into two sections: "Update Employee Details" on the left and "Update Salary Details" on the right. The "Update Employee Details" section contains a vertical list of input fields: "Employee ID", "NIC", "First Name*", "Last Name*", "Date of Birth*", "Address", "City", "Contact No" (with sub-fields for "Home" and "Mobile"), "Designation*", "Department*", and "Date of Joining". The "Update Salary Details" section contains a sub-section "Update Basic Salary" with a "Salary Type" dropdown menu (set to "Monthly") and a "Salary Amount" input field. At the bottom, there are three buttons: "Search", "Update Employee", and "Exit".

- Search Employee Details,



Payroll Management System | Home

File Employee Payroll Leave Pay Slip Help

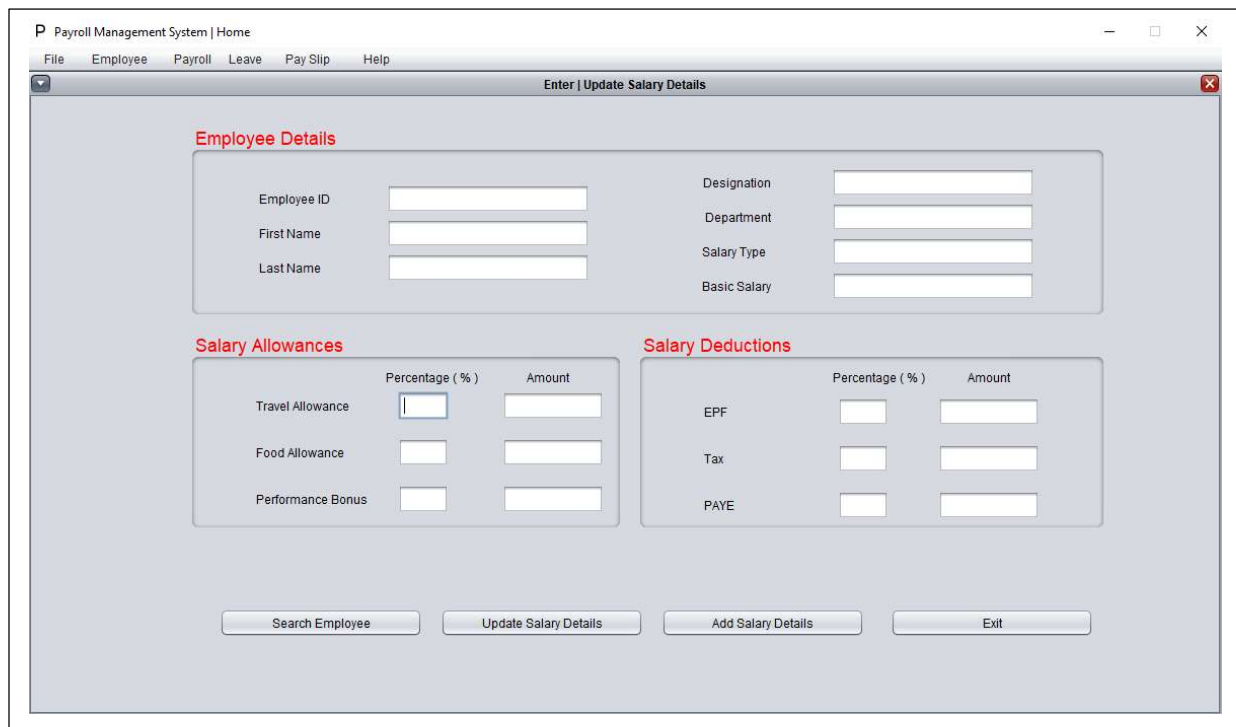
Search Employee Details

Search Employee

Employee Details

empId	nic	fname	lname	dob	address	city	gender	designation	department	tel_home	tel_mobile	date_of_j...	salType	salAmount
1	93155217...	Salman	Khan	1993-06-03	Marhaba ...	Karachi	Male	Software ...	IT	01129347...	07750960...	2015-12-12	Monthly	90000
2	92154781...	Okasha	Mohsin	1993-05-04	Shadman	Karachi	Male	Senior De...	IT	01125684...	07784125...	2015-01-05	Monthly	30000
3	90133658...	Muhamm...	Kamil	1990-05-04	DHA Phas...	Karachi	Male	Business ...	IT	01127896...	07134562...	2015-04-08	Monthly	38000
4	42101	Ali	Khan	1996-03-24	Khi	Lahore	Male	IT	IT	090298018	20983109	2015-03-21	Monthly	50000

- Payroll Details,



Payroll Management System | Home

File Employee Payroll Leave Pay Slip Help

Enter | Update Salary Details

Employee Details

Employee ID Designation

First Name Department

Last Name Salary Type

Basic Salary

Salary Allowances

	Percentage (%)	Amount
Travel Allowance	<input type="text"/>	<input type="text"/>
Food Allowance	<input type="text"/>	<input type="text"/>
Performance Bonus	<input type="text"/>	<input type="text"/>

Salary Deductions

	Percentage (%)	Amount
EPF	<input type="text"/>	<input type="text"/>
Tax	<input type="text"/>	<input type="text"/>
PAYE	<input type="text"/>	<input type="text"/>

Search Employee Update Salary Details Add Salary Details Exit

- Apply Leave,

Payroll Management System | Home

File Employee Payroll Leave Pay Slip Help

Leave Status | Apply Leave

Employee Details

Employee ID	<input type="text"/>	Designation	<input type="text"/>
First Name	<input type="text"/>	Department	<input type="text"/>
Last Name	<input type="text"/>		

Leave Details

	Available Leave Count	Apply Leave
Annual Leave	<input type="text"/>	<input type="text"/>
Casual Leave	<input type="text"/>	<input type="text"/>
Optional Leave	<input type="text"/>	<input type="text"/>

Search Employee Apply Leave Exit

- Pay Slip,

Payroll Management System | Home

File Employee Payroll Leave Pay Slip Help

Pay Slip

Employee ID :	1
Name :	Salman Khan
Designation :	Software Engineer
Department :	IT

Earnings	Amount (LKR)
Basic Pay :	90000.0
+ Travel Allowance	48.0
+ Food Allowance	40.0
+ Performance Bonus	60.0
Gross Pay :	90148.0

Deductions	Amount (LKR)
- EPF	32.0
- Employee Tax	8.0
- PAYE	8.0
Total Deductions :	48.0

Net Pay :	90100.0
------------------	----------------

Search Employee Print Pay Slip Exit