

PROJECT REPORT  
ON  
EMPLOYEE PAYROLL MANAGEMENT  
(FOR THE COURSE OF T. Y. B.Sc.  
COMPUTER SCIENCE)

DESIGNED AND DEVELOPED  
BY  
ATHASHRI AJAY GANGAN

GUIDED BY  
Ms. PRATIKSHA HARWALKAR  
DEPARTMENT OF COMPUTER SCIENCE

PARLE TILAK VIDYALAYA ASSOCIATION'S  
MULUND COLLEGE OF COMMERCE (AUTONOMOUS)  
*(Affiliated to University of Mumbai)*  
2023-24

MULUND (WEST), MUMBAI-400080  
MAHARASHTRA, INDIA

## **ACKNOWLEDGEMENT**

I have a great pleasure in representing this project report entitled “Employee Payroll Management” and I grab this opportunity to convey my immense regards towards all the distinguished people who have their valuable contribution in the hour of need.

I like to extend my gratitude to our beloved Principal Dr. Sonali Pednekar for her timely and prestigious guidance.

I take this opportunity to thank Dr. Reena Nagda, Coordinator of the Department and all the faculty members of the Department of Computer Science of Mulund College of Commerce, for giving me an opportunity to complete this project and the most needed guidance throughout the duration of the programme.

I am extremely grateful to my project guide Ms. Pratiksha Harwalkar for her valuable guidance and necessary support during each phase of the project. She was the source of continuous encouragement as each milestone was crossed.

Finally I also owe to my fellow friends who have been a constant source of help to solve the problems that cropped up during the project development process.

Athashri Ajay Gangan

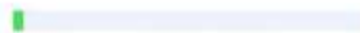
# Plagiarism Report

## Plagiarism Checker X Originality Report



### Plagiarism Checker X - Report Originality Assessment

**3%**



**Overall Similarity**

**Date:** Sep 15, 2023  
**Matches:** 125 / 4402 words  
**Sources:** 4

**Remarks:** Low similarity detected, check with your supervisor if changes are required.

**Verify Report:**  
Scan this QR Code



Plagiarism Checker X

Dashboard

Online Plagiarism

Side By Side Difference

Bulk Comparison

Settings

Activated (Enterprise) v8.0.7

Update Available

< Back

New Update is Available! v8.0.7 [Update Now](#)

### Summary

Scanned content overall summary report.

[Share Score with Friends](#)  
[Share on Facebook](#)  
[Tweet](#)

■ Plagiarized (3%) ■ Unique (97%)

Date	Sep 15, 2023
Words	125 Words Plagiarized / Total Words 4402
Source	4 Source(s) Identified
Remarks	Low similarity detected, check your supervisor if changes are required

[View Report](#)

Type here to search

10:18 AM 9/15/2023

## **INDEX**

<b>SR. NO.</b>	<b>TOPIC</b>	<b>PAGE NO.</b>
1.	TITLE	1
2.	INTRODUCTION	2
3.	REQUIREMENT SPECIFICATION	4
4.	SYSTEM DESIGN DETAILS	6
5.	SYSTEM IMPLEMENTATION	17
6.	RESULTS	73
7.	CONCLUSION AND FUTURE ENHANCEMENTS	79
8.	REFERENCES	80
9.	ANNEXURE	81

## **1. TITLE**

Title of the Project: Employee Payroll Management

Type of Project: Desktop App

Developed by: Athashri Ajay Gangan

## **2. INTRODUCTION**

### **2.1 Objective of the project:**

- Automating payroll calculations- Develop a system that automates the calculation of employee salaries, deductions, increment, minimizing manual errors and increasing efficiency.
- Employee data management- Implement a centralized database to store and manage employee information, including personal information, salary history and benefits enrollment.
- Payroll processing and scheduling- Establish a structured process for payroll processing and generating payment slip.
- Security and Confidentiality- Implement robust security measures to protect sensitive employee information, including payroll data, ensuring confidentiality and compliance with data privacy regulations.
- The company can manage employee information, create individual payroll and can generate payment slip.

### **2.2 Description of the current system:**

- End users of the current system have access to personal information of the employees.
- The user will find the login module appearing on the home page and get logged in into the system as the system administrator.

### **2.3 Limitations of current system:**

- The main drawback of this system is its huge margin of mistake.
- Furthermore, it takes time since the user must double verify the data before printing the slip in order to make certain that the payroll is precise.

**2.4 Description of proposed system:**

- Proposed system makes use of a desktop application to securely store the information and manage all reports.
- User can make reports on the system and deploys the report to employees.
- Additionally, management permits an employee's reduction and advancement.

**2.5 Advantages of proposed system:**

- Calculate the salary, deduction, increments and computes more quickly.
- Maintain data in a way that is safe and simple to use and generate appropriate payment slip.
- The system is available 24/7 for administration to produce paperless payment slips.

### **3. REQUIREMENTS SPECIFICATION**

#### **3.1 Software Requirements:**

- Application Software- NetBeans IDE
- Front end- Java Swing
- Back end- MySQL Database

#### **3.2 Hardware Requirements:**

- RAM -4GB or 8GB
- 64 Bit Operating System

#### **3.3 Data Requirements:**

- Username and Password
- Employee ID
- First Name
- Last Name
- Date of Birth
- Address
- City
- Mobile No
- Gender
- Designation/Post
- Department
- Date of Joining



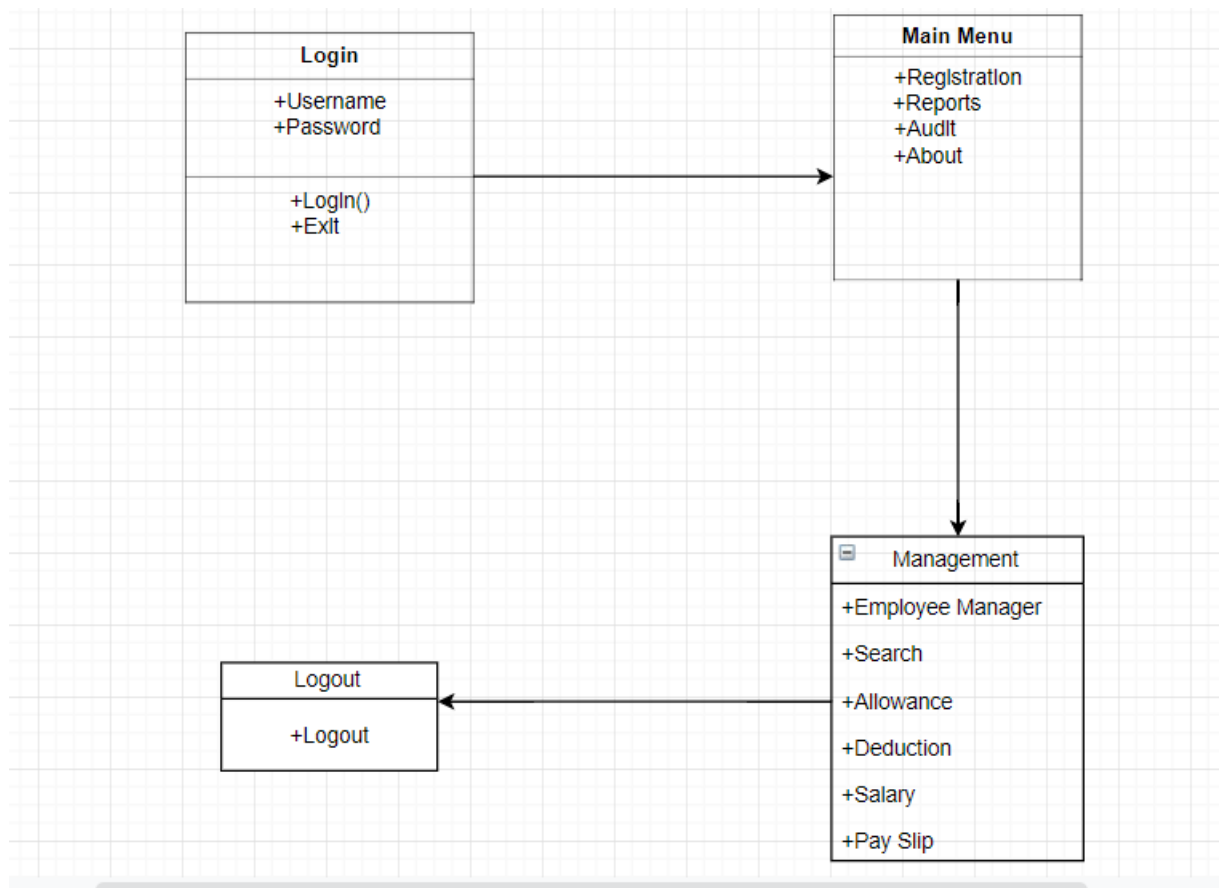
**3.4 Fact finding questions:**

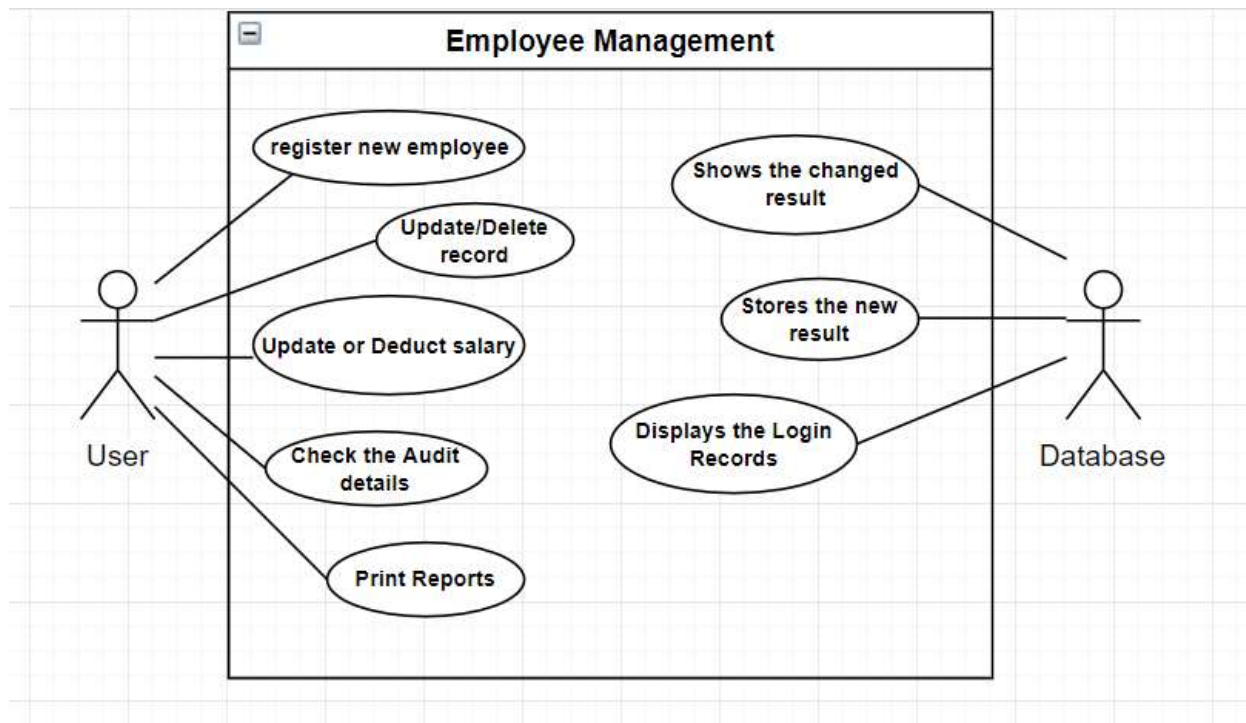
- What is the purpose of the app?
- What is the need of Employee Payroll System?
- How exactly this app works?
- What logic is used behind while creating the app?
- Can we get proper results?

## **4. SYSTEM DESIGN DETAILS**

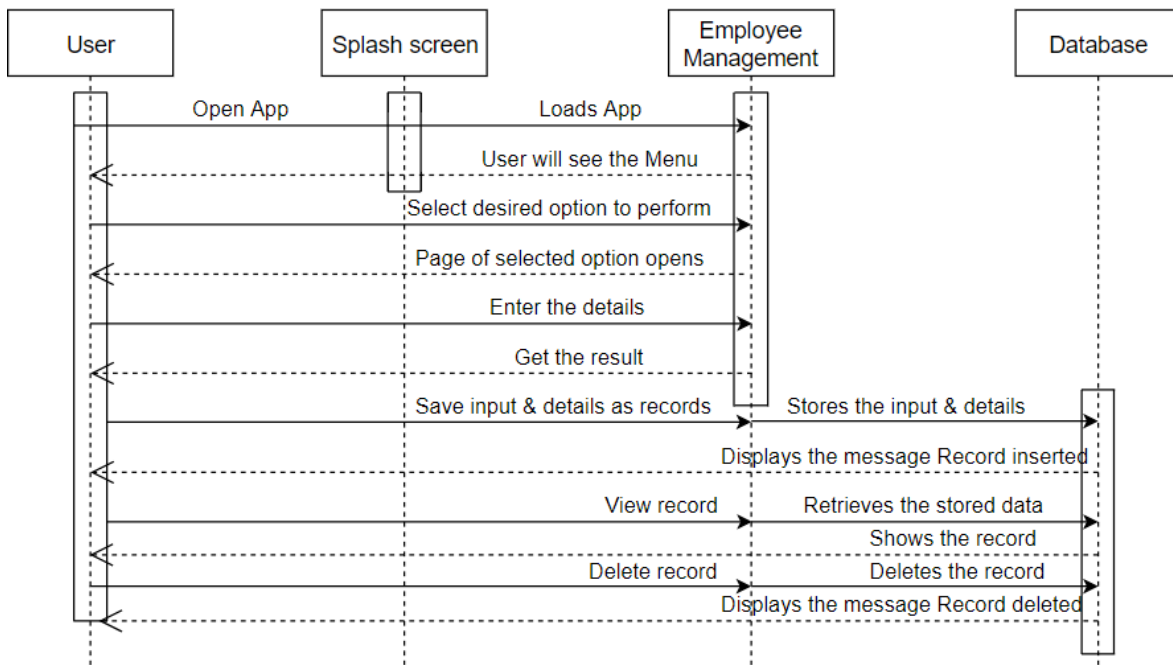
### **4.1 Event Table:**

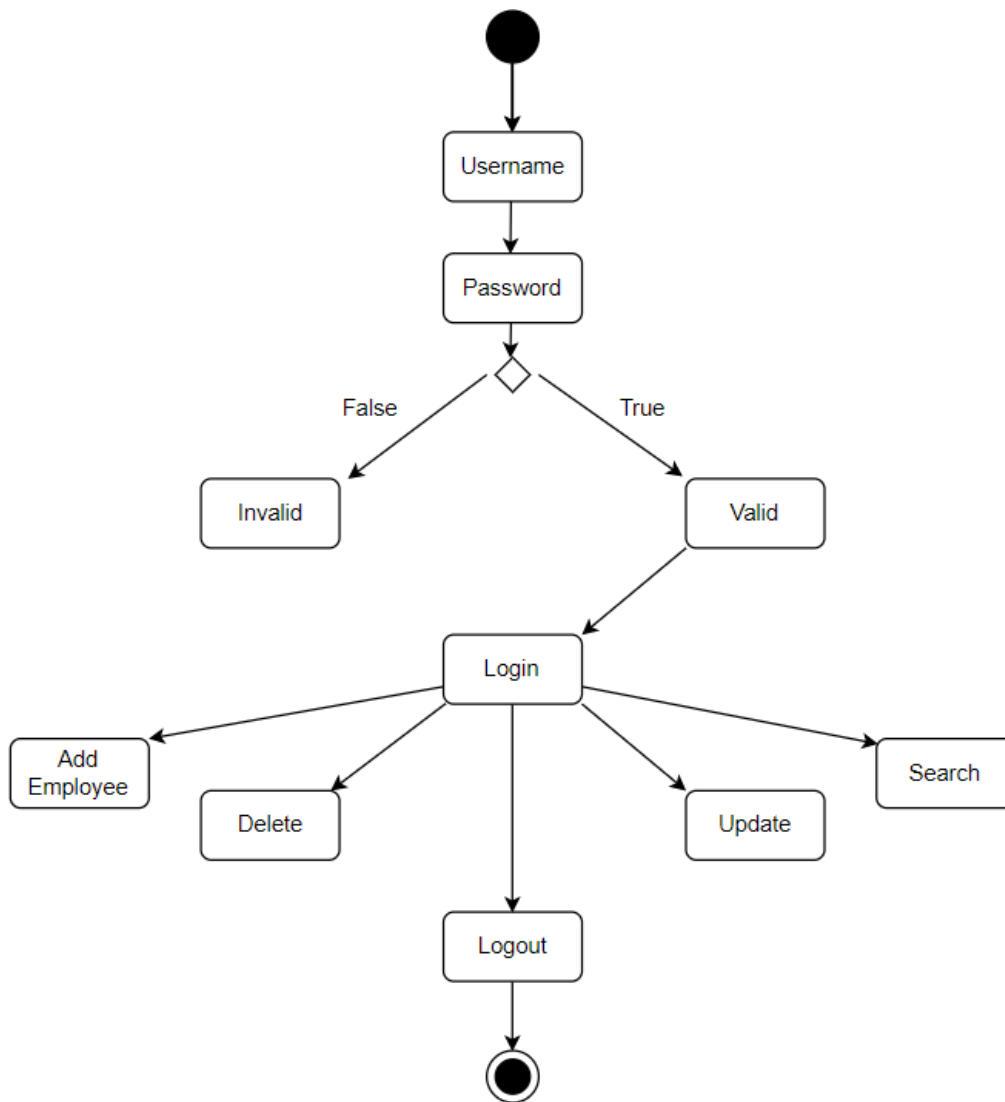
<b>Sr. No.</b>	<b>Events</b>	<b>Trigger</b>	<b>Source</b>	<b>Activity</b>	<b>Response</b>	<b>Destination</b>
1.	User Login	btn_login	User	Entering Login username and password	If username, password correct then user gets logged in else error.	login.java
2.	Employee Registration	btn_addemp	User	Adding Employee details	Record inserted & Data saved successfully	addEmployee.java
3.	Searching Employee	btn_searchemp	User	Update and Delete employee details	Record updated/deleted	searchemployee.java
4.	Updating Salary	btn_update	User	Updates the Salary of an Employee	Record Updated	updateSalary.java
5.	Allowance & Deduction	btn_allowance btn_deduction	User	Permits and Deducts the salary	Allowance added, Data saved successfully	Allowance.java & employeeDeduction.java
6.	Payment	btn_payment	User	Generates Payslip of employee	Report successfull ygenerated	searchempSalarySlip.java
7.	Logout	btn_logout	User	Logs out and returns to login page	Logged out	MainMenu.java

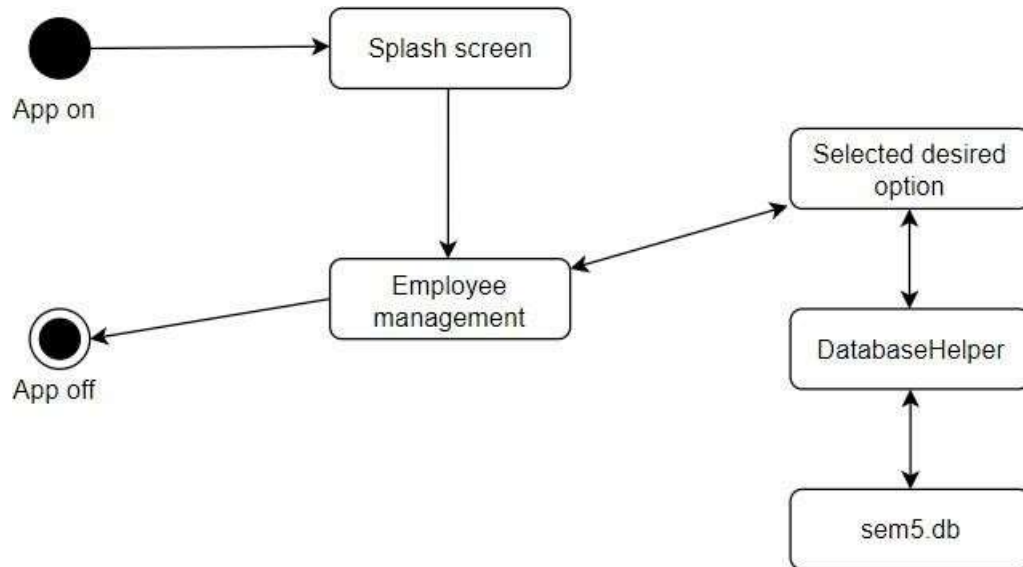
**4.2 Class Diagram:****Figure 4.2 Class Diagram for Employee Payroll Management**

**4.3 Use Case Diagram:**

**Figure 4.3 Use Case Diagram for Employee Payroll Management**

**4.4 Sequence Diagram:****Figure 4.4 Sequence Diagram for Employee Payroll Management**

**4.5 Activity Diagram:****Figure 4.5 Activity Diagram for Employee Payroll Management**

**4.6 State Diagram:**

**Figure 4.6 State Diagram for Employee Payroll Management**

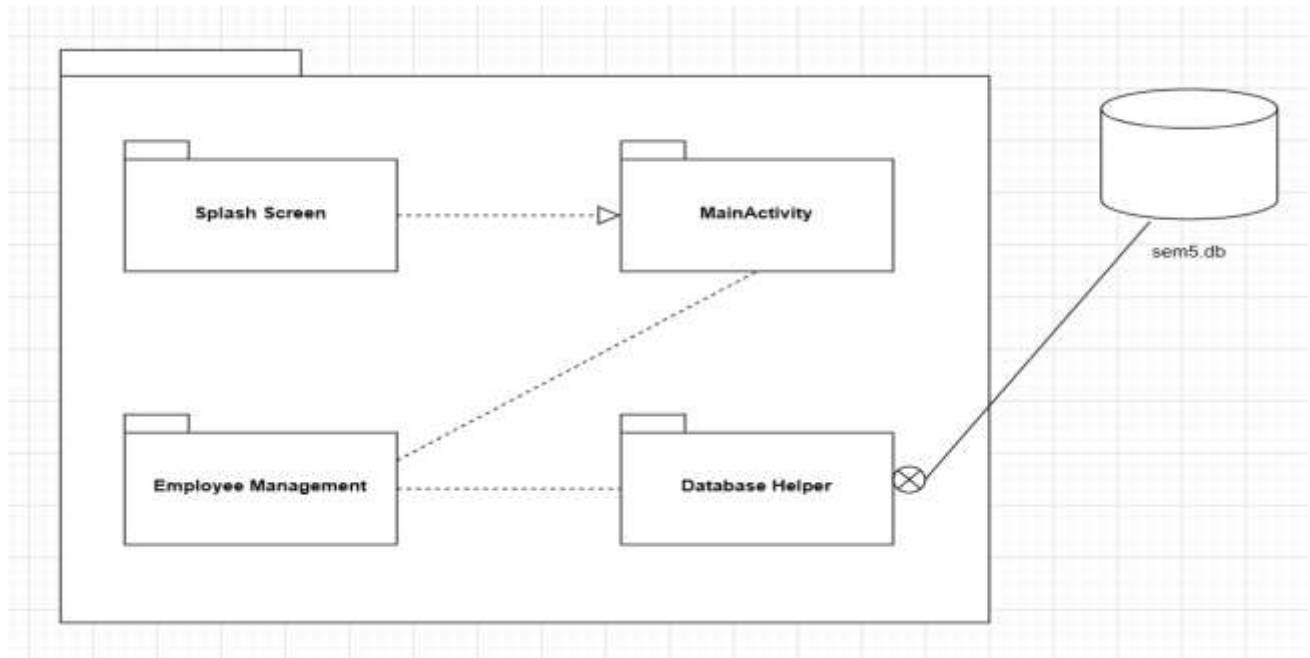
**4.7 Package Diagram:**

Figure 4.7 Package Diagram for Employee Payroll Management

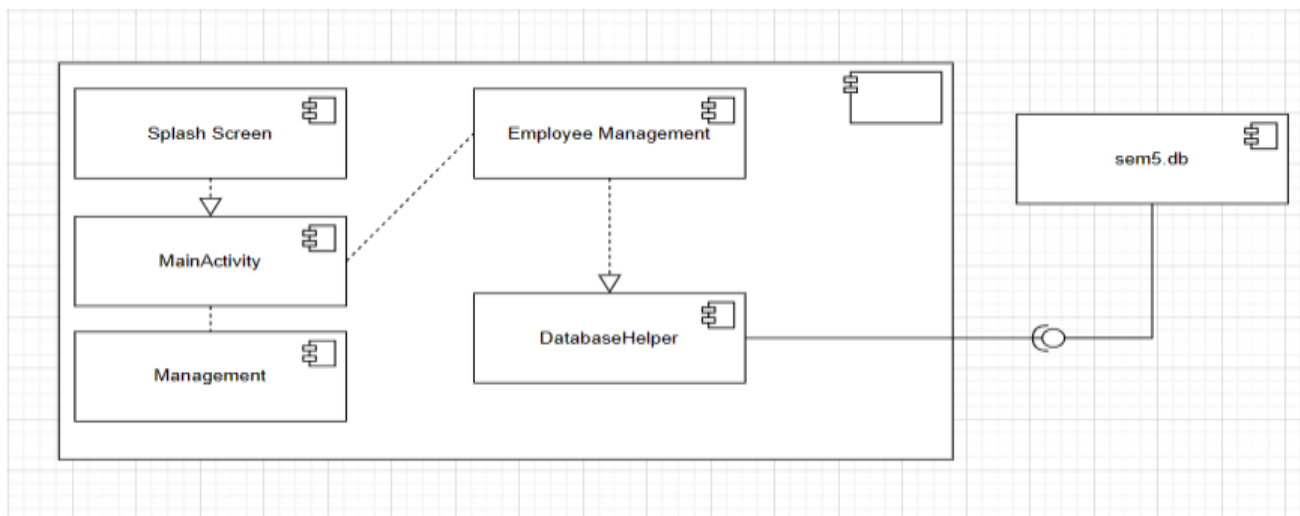
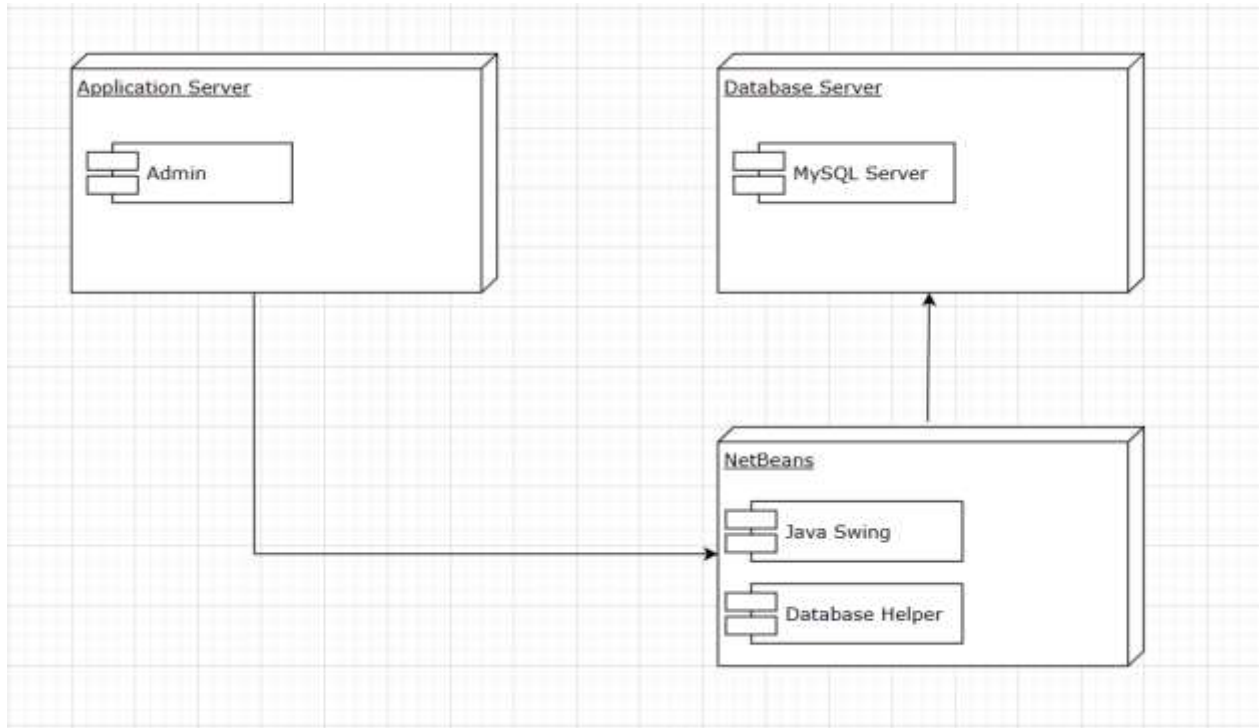
**4.8 Component Diagram:**

Figure 4.8 Component Diagram for Employee Payroll Management



**4.9 Deployment Diagram:**

**Figure 4.9 Deployment Diagram for Employee Payroll Management**

#### **4.10 Database Design:**

##### **4.10.1 Admin Table:**

```
mysql> desc admin;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
username	varchar(20)	YES		NULL	
password	varchar(20)	YES		NULL	
division	varchar(20)	YES		NULL	

4 rows in set (0.00 sec)

Table 4.10.1 Admin

##### **4.10.2 Staff information Table:**

```
mysql> desc staff_information;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
first_name	varchar(10)	YES		NULL	
surname	varchar(10)	YES		NULL	
Dob	date	YES		NULL	
Email	varchar(20)	YES		NULL	
Telephone	varchar(15)	YES		NULL	
Address	varchar(20)	YES		NULL	
Department	varchar(10)	YES		NULL	
Image	blob	YES		NULL	
Gender	varchar(10)	YES		NULL	
Salary	float(10,3)	YES		NULL	
Address2	varchar(20)	YES		NULL	
Apartment	varchar(20)	YES		NULL	
Post_code	int	YES		NULL	
Designation	varchar(15)	YES		NULL	
Status	varchar(10)	YES		NULL	
Date_hired	date	YES		NULL	
job_title	varchar(10)	YES		NULL	

18 rows in set (0.00 sec)

Table 4.10.2 Staff\_information

**4.10.3 Allowance Table:**

```
mysql> desc allowance;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
overtime	varchar(30)	YES		NULL	
medical	varchar(20)	YES		NULL	
bonus	varchar(20)	YES		NULL	
other	varchar(20)	YES		NULL	
emp_id	varchar(10)	YES		NULL	
salary	varchar(20)	YES		NULL	
rate	varchar(20)	YES		NULL	
total_allowance	varchar(20)	YES		NULL	
firstname	varchar(20)	YES		NULL	
surname	varchar(20)	YES		NULL	
created_by	varchar(30)	YES		NULL	

12 rows in set (0.00 sec)

Table 4.10.3 Allowance

**4.10.4 Deductions Table:**

```
mysql> desc deductions;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
firstname	varchar(20)	YES		NULL	
surname	varchar(20)	YES		NULL	
salary	varchar(20)	YES		NULL	
deduction_amount	varchar(20)	YES		NULL	
deduction_reason	varchar(100)	YES		NULL	
emp_id	int	YES		NULL	
made_by	varchar(20)	YES		NULL	

8 rows in set (0.00 sec)

Table 4.10.4 Deductions

**4.10.5 Audit Table:**

```
mysql> desc audit;
```

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
date	varchar(30)	YES		NULL	
status	varchar(30)	YES		NULL	

3 rows in set (0.00 sec)

Table 4.10.5 Audit

## **5. SYSTEM IMPLEMENTATION**

### **1. EmployeePayrollManagement.java (Landing page)**

```
package employee.payroll.system;

/**
 *
 * @author ATHASHRI
 */
public class EmployeePayrollSystem {

    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

### **2. db.java(Connectivity page)**

```
package employee.payroll.system;

/**
 *
 * @author ATHASHRI
```

```
*/

import java.sql.*;
import
javax.swing.*;
public class db {

    Connection conn=null;

    public static Connection java_db(){

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn
=DriverManager.getConnection("jdbc:mysql://localhost:3306/athasri_project","root","athashri");

            return conn;

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

    }
}
```

**3. EMP.java**

```
package employee.payroll.system;
```

```
/**
```

```
*
```

```
* @author ATHASHRI
```

```
*/
```

```
public class Emp {
```

```
    public static int empId;
```

**4. login.java**

```
package employee.payroll.system;
```

```
/**
```

```
*
```

```
* @author ATHASHRI
```

```
*/
```

```
import com.sun.glass.events.KeyEvent;
```

```
import java.awt.*;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.text.DateFormat;
```

```
import java.text.SimpleDateFormat;
```

```
import javax.swing.*;
```

```
import java.util.Calendar;
```

```
import java.util.Date;
```

```
import java.util.GregorianCalendar;
```

```
public class login extends javax.swing.JFrame {
```

```
    Connection conn=null;
```

```
    ResultSet rs=null;
```

```
    PreparedStatement pst=null;
```

```
    public login() {
```

```

initComponents();
conn=db.java_db();
Toolkit toolkit = getToolkit();
Dimension size = toolkit.getScreenSize();
setLocation(size.width/2 - getWidth()/2,
size.height/2 - getHeight()/2);
currentDate();
}

public void currentDate (){

    Calendar cal =new GregorianCalendar();
    int month = cal.get(Calendar.MONTH);
    int year = cal.get(Calendar.YEAR);
    int day = cal.get(Calendar.DAY_OF_MONTH);

    txt_date.setText((month+1)+"/"+day+"/"+year);

    int second = cal.get(Calendar.SECOND);
    int minute = cal.get(Calendar.MINUTE);
    int hour = cal.get(Calendar.HOUR);

    lbl_time.setText(hour+":"+minute+": "+second);
}

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

    if (txt_username.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Username Field is empty");
    } else if (txt_password.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Password Field is empty");
    }else {
        String sql = "select id,username,password from admin Where (username =? and password =?)";
        try{
            int count =0;

            pst=conn.prepareStatement(sql);

            pst.setString(1,txt_username.getText());
            pst.setString(2,txt_password.getText());

            rs=pst.executeQuery();
            {
            }
        }
    }
}

```



```

while(rs.next()){
/
    int id = rs.getInt(1);
    Emp.empId = id;
    count =count+1;
}

if(count==1){
    JOptionPane.showMessageDialog(null,"Sucess" );
    MainMenu j = new MainMenu();
    j.setVisible(true);
    this.dispose();

    Date currentDate = GregorianCalendar.getInstance().getTime();
    DateFormat df = DateFormat.getDateInstance();
    String dateString = df.format(currentDate);

    Date d = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    String timeString = sdf.format(d);

    String value0 = timeString;
    String values = dateString;

    int value = Emp.empId;
    String reg = "insert into Audit (id,date,status) values ('"+value+"','"+value0+" /
"+values+"','Logged in)";
    pst=conn.prepareStatement(reg);
    pst.execute();
    this.dispose();

}

else if(count>1){
    JOptionPane.showMessageDialog(null,"Duplicate Username or Password Access
denied");
}
else{
    JOptionPane.showMessageDialog(null,"Username and Password is not correct");
}

} catch(Exception e)

{
    JOptionPane.showMessageDialog(null, e);

}
finally {

```

```

        try{
            rs.close();
            pst.close();

        }
        catch(Exception e){

        }
    }

}
}

private void mi1ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void txt_passwordKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode()==KeyEvent.VK_ENTER)
    {
        String sql = "select id,username,password,division from User Where (username=? and
password =?)";

        try{
            int count =0;

            pst=conn.prepareStatement(sql);

            pst.setString(1,txt_username.getText());
            pst.setString(2,txt_password.getText());

            rs=pst.executeQuery();

            {
            }
            while(rs.next()){
                int id = rs.getInt(1);
                Emp.empId = id;
                count =count+1;
            }

            if(count==1){
                JOptionPane.showMessageDialog(null,"Success" );
            }
        }
    }
}

```

```

MainMenu j = new MainMenu();
j.setVisible(true);
this.dispose();

Date currentDate = GregorianCalendar.getInstance().getTime();
DateFormat df = DateFormat.getDateInstance();
String dateString = df.format(currentDate);

Date d = new Date();
SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
String timeString = sdf.format(d);

String value0 = timeString;
String values = dateString;

int value = Emp.empId;
String reg = "insert into Audit (id,date,status) values ('"+value+"','"+value0+" /
"+values+"','Logged in)";
pst=conn.prepareStatement(reg);
pst.execute();
this.dispose();

}

else if(count>1){
    JOptionPane.showMessageDialog(null,"Duplicate Username or Password Access
denied");
}

else{
    JOptionPane.showMessageDialog(null,"Username and Password is not correct");

}

} catch(Exception e)

{
    JOptionPane.showMessageDialog(null, e);

}
finally {

```

```

        try{
            rs.close();
            pst.close();

        }
        catch(Exception e){

        }
    }

}

```

### 5. MainMenu.java

```

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.pdf.PdfWriter;
import java.awt.Dimension;
import java.awt.Toolkit;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.FontFactory;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.GrayColor;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfWriter;
import com.itextpdf.text.pdf.PdfPTable;
import java.awt.Font;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.OutputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import java.util.Date;
import java.util.GregorianCalendar;

/**
 *
 * @author ATHASHRI
 */

public class MainMenu extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public MainMenu() {
        initComponents();
        conn=db.java_db();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);
        txt_emp.setText(String.valueOf(Emp.empId).toString());
    }
    private void mi5ActionPerformed(java.awt.event.ActionEvent evt) {

        Audit_details x = new Audit_details();
        x.setVisible(true);

    private void btn_updateActionPerformed(java.awt.event.ActionEvent evt) {
        updateSalary x = new updateSalary();
        x.setVisible(true);
    }
    private void btn_searchempActionPerformed(java.awt.event.ActionEvent evt) {

        searchEmployee e = new searchEmployee();
        e.setVisible(true);
    }
    private void btn_addempActionPerformed(java.awt.event.ActionEvent evt) {

        addEmployee x = new addEmployee();
        x.setVisible(true);
    }
    private void btn_allowanceActionPerformed(java.awt.event.ActionEvent evt) {

        Allowance x = new Allowance();

```

```

        x.setVisible(true);
    }

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

    login x = new login();
    x.setVisible(true);
    this.dispose();

    try{
        Date currentDate = GregorianCalendar.getInstance().getTime();
        DateFormat df = DateFormat.getDateInstance();
        String dateString = df.format(currentDate);

        Date d = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
        String timeString = sdf.format(d);

        String value0 = timeString;
        String values = dateString;

        int value = Emp.empId;
        String reg = "insert into Audit (id,date,status) values ('"+value+"','"+value0+" /
"+values+"','Logged out')";
        pst=conn.prepareStatement(reg);
        pst.execute();
        this.dispose();
    }catch (Exception e){

        JOptionPane.showMessageDialog(null, e);

    }
    finally {

        try{
            rs.close();
            pst.close();

        }
        catch(Exception e){

        }

    }

}

```

```

}

private void mi1ActionPerformed(java.awt.event.ActionEvent evt) {

    addEmployee x = new addEmployee();
    x.setVisible(true);

}

private void mi2ActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser dialog = new JFileChooser();
    dialog.setSelectedFile(new File("Employees Report.pdf"));
    int dialogResult = dialog.showSaveDialog(null);
    if (dialogResult==JFileChooser.APPROVE_OPTION){
        String filePath = dialog.getSelectedFile().getPath();

    try {

        String sql ="select * from Staff_information";

        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();

        Document myDocument = new Document();
        PdfWriter myWriter = PdfWriter.getInstance(myDocument, new FileOutputStream(filePath ));
        PdfPTable table = new PdfPTable(13);
        myDocument.open();

        float[] columnWidths = new float[] {3,8,7,5,5,9,8,9,6,6,6,8,8};
        table.setWidths(columnWidths);

        table.setWidthPercentage(100); //set table width to 100%

        myDocument.add(new Paragraph("Employees
List",FontFactory.getFont(FontFactory.TIMES_BOLD,20,Font.BOLD )))
        myDocument.add(new Paragraph(new Date().toString()));
        myDocument.add(new Paragraph(".....
....."));
    }
}

```

```

        table.addCell(new PdfPCell(new
Paragraph("ID",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new Paragraph("First
Name",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Surname",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new Paragraph("Date of
Birth",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Email",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Telephone",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Address",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Department",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Gender",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Salary",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new
Paragraph("Status",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new Paragraph("Date
Hired",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));
        table.addCell(new PdfPCell(new Paragraph("Job
Title",FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.BOLD))));

        while(rs.next())
        {

            table.addCell(new PdfPCell(new
Paragraph(rs.getString(1),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(2),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(3),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(4),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(5),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(6),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(7),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new
Paragraph(rs.getString(8),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
            table.addCell(new PdfPCell(new

```



```

Paragraph(rs.getString(10),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(11),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(16),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(17),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(18),FontFactory.getFont(FontFactory.TIMES_ROMAN,8,Font.PLAIN))));

    }

    myDocument.add(table);
    myDocument.add(new Paragraph(".....
....."));
    myDocument.close();
    JOptionPane.showMessageDialog(null,"Report was successfully generated");

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

}
finally {

    try{
        rs.close();
        pst.close();

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

    }

}
}

}

```

```

private void mi3ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    JFileChooser dialog = new JFileChooser();
    dialog.setSelectedFile(new File("Employee Allowance Report.pdf"));
    int dialogResult = dialog.showSaveDialog(null);
    if (dialogResult==JFileChooser.APPROVE_OPTION){
        String filePath = dialog.getSelectedFile().getPath();

```

```

try {

    String sql ="select * from Allowance";

    pst=conn.prepareStatement(sql);
    rs=pst.executeQuery();

    Document myDocument = new Document();
    PdfWriter myWriter = PdfWriter.getInstance(myDocument, new FileOutputStream(filePath ));
    PdfPTable table = new PdfPTable(11);
    myDocument.open();

    float[] columnWidths = new float[] { 3,7,7,5,5,9,6,5,8,8,8};
    table.setWidths(columnWidths);

    table.setWidthPercentage(100);

    myDocument.add(new Paragraph("Employees Allowance
List",FontFactory.getFont(FontFactory.TIMES_BOLD,20,Font.BOLD ));
    myDocument.add(new Paragraph(new Date().toString()));
    myDocument.add(new Paragraph(".....
....."));
    table.addCell(new PdfPCell(new
Paragraph("ID",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Overtime",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Medical",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Bonus",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Other",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new Paragraph("Employee
ID",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Salary",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Rate",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new
Paragraph("Allowance",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new Paragraph("First
Name",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
    table.addCell(new PdfPCell(new

```

```
Paragraph("Surname",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
```

```
while(rs.next())
{
```

```
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(1),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(2),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(3),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(4),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(5),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(6),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(7),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(8),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(9),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(10),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
    table.addCell(new PdfPCell(new
Paragraph(rs.getString(11),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));

}
```

```
myDocument.add(table);
```

```
myDocument.add(new Paragraph(".....
....."));
```

```
myDocument.close();
```

```
JOptionPane.showMessageDialog(null,"Report was successfully generated");
```

```
}
```

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

```
    JOptionPane.showMessageDialog(null,e);
```

```
}
```

```
finally {
```

```
    try{
```

```
        rs.close();
```

```

        pst.close();

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

}

private void btn_deductionActionPerformed(java.awt.event.ActionEvent evt) {
    employeeDeductions x = new employeeDeductions();
    x.setVisible(true);

}

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

    searchEmpSalarySlip x = new searchEmpSalarySlip();
    x.setVisible(true);

}

private void mi4ActionPerformed(java.awt.event.ActionEvent evt) {

    JFileChooser dialog = new JFileChooser();
    dialog.setSelectedFile(new File("Employee Deduction Report.pdf"));
    int dialogResult = dialog.showSaveDialog(null);
    if (dialogResult==JFileChooser.APPROVE_OPTION){
        String filePath = dialog.getSelectedFile().getPath();

    try {

        String sql ="select * from Deductions";

        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();

        Document myDocument = new Document();
        PdfWriter myWriter = PdfWriter.getInstance(myDocument, new FileOutputStream(filePath ));
        PdfPTable table = new PdfPTable(8);
        myDocument.open();
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
}

```

```

float[] columnWidths = new float[] { 3,7,7,5,5,9,6,5};
table.setWidths(columnWidths);

table.setWidthPercentage(100); //set table width to 100%

myDocument.add(new Paragraph("Employees Deduction
List",FontFactory.getFont(FontFactory.TIMES_BOLD,20,Font.BOLD)));
myDocument.add(new Paragraph(new Date().toString()));
myDocument.add(new Paragraph(".....
....."));
table.addCell(new PdfPCell(new
Paragraph("ID",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new Paragraph("First
Name",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new
Paragraph("Surname",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new
Paragraph("Salary",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new Paragraph("Deduction
Amount",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new Paragraph("Deduction
Reason",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new Paragraph("Employee
ID",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));
table.addCell(new PdfPCell(new Paragraph("Created
By",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.BOLD))));

while(rs.next())
{
table.addCell(new PdfPCell(new
Paragraph(rs.getString(1),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(2),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(3),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(4),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(5),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(6),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(7),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));
table.addCell(new PdfPCell(new
Paragraph(rs.getString(8),FontFactory.getFont(FontFactory.TIMES_ROMAN,9,Font.PLAIN))));

```

```
    }

    myDocument.add(table);
    myDocument.add(new Paragraph(".....
....."));
    myDocument.close();
    JOptionPane.showMessageDialog(null,"Report was successfully generated");

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

}
finally {

    try{
        rs.close();
        pst.close();

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

    }
}
}
```

**6. addEmployee.java**

```

import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import javax.imageio.ImageIO;
import javax.swing.*;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import java.util.regex.*;

/**
 *
 * @author ATHASHRI
 */

public class addEmployee extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public addEmployee() {
        initComponents();
        conn=db.java_db();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);
        currentDate();

        txt_emp.setText(String.valueOf(Emp.empId).toString());
    }

    public void currentDate (){

        Calendar cal =new GregorianCalendar();
        int month = cal.get(Calendar.MONTH);

```

```

int year = cal.get(Calendar.YEAR);
int day = cal.get(Calendar.DAY_OF_MONTH);

lbl_date.setText((month+1)+"/"+day+"/"+year);

//Time

int second = cal.get(Calendar.SECOND);
int minute = cal.get(Calendar.MINUTE);
int hour = cal.get(Calendar.HOUR);

lbl_time.setText(hour+": "+(minute)+": "+second);

}
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

    JFileChooser chooser = new JFileChooser();
    chooser.showOpenDialog(null);
    File f = chooser.getSelectedFile();

    filename =f.getAbsolutePath();
    ImageIcon imageIcon = new ImageIcon(new
    ImageIcon(filename).getImage().getScaledInstance(img.getWidth(), img.getHeight(),
    Image.SCALE_DEFAULT));
    img.setIcon(imageIcon);
    try {

        File image = new File(filename);
        FileInputStream fis = new FileInputStream (image);
        ByteArrayOutputStream bos= new ByteArrayOutputStream();
        byte[] buf = new byte[1024];

        for(int readNum; (readNum=fis.read(buf))!=-1; ){

            bos.write(buf,0,readNum);
        }
        person_image=bos.toByteArray();
    }

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

}
}

```



```

private void btn_clearActionPerformed(java.awt.event.ActionEvent evt) {
    txt_id.setText("");
    txt_firstname.setText("");
    txt_surname.setText("");
    txt_tel.setText("");
    txt_dob.setText("");
    txt_email.setText("");
    txt_address.setText("");
    txt_dep.setText("");
    txt_status.setText("");
    txt_salary.setText("");
    txt_add2.setText("");
    txt_pc.setText("");
    txt_desig.setText("");
    txt_job.setText("");
    txt_apr.setText("");
    txt_doj.setText("");
    img.setIcon(null);
}

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

    int p = JOptionPane.showConfirmDialog(null, "Are you sure you want to add record?", "Add
Record",JOptionPane.YES_NO_OPTION);
    if(p==0){

        try {
            String sql ="insert into Staff_information "
                + "(first_name,surname,Dob,Email,"
                + "Telephone,Address,Department,"
                + "Image,Salary,Gender,Address2,"
                + "Post_code, Designation,Status,job_title,Apartment,Date_hired,id) values
                (? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ,? ) ";

            pst=conn.prepareStatement(sql);
            pst.setString(1,txt_firstname.getText());
            pst.setString(2,txt_surname.getText());
            pst.setString(3,txt_dob.getText());
            pst.setString(4,txt_email.getText());
            pst.setString(5,txt_tel.getText());
            pst.setString(6,txt_address.getText());
            pst.setString(7,txt_dep.getText());
            pst.setString(8,filename);

            pst.setString(9,txt_salary.getText());
            pst.setString(10,gender);
            pst.setString(11,txt_add2.getText());

```

```

pst.setString(12,txt_pc.getText());
pst.setString(13,txt_desig.getText());
pst.setString(14,txt_status.getText());
pst.setString(15,txt_job.getText());
pst.setString(16,txt_apr.getText());
pst.setString(17,txt_doj.getText());

Statement stm = conn.createStatement();
rs = stm.executeQuery("select max(id) as id from staff_information");
if(rs.next()){
    int id = rs.getInt("id") + 1;
    pst.setInt(18, id);
}

pst.execute();
JOptionPane.showMessageDialog(null,"Data is saved successfully");

}
catch (Exception e)

{
    JOptionPane.showMessageDialog(null,e);
}
try {

String sq ="select * from Staff_information where (id) NOT IN (select id from User)";
pst=conn.prepareStatement(sq);
rs=pst.executeQuery();

while(rs.next()){

    String add1 =rs.getString("id");
    String add2 =rs.getString("first_name");
    String add3 =rs.getString("Dob");
    String result =add3.replace("/", "-");
    String add4 =rs.getString("Department");

    String sql ="insert into User (division,username,password,id) values
('"+add4+"','"+add2+"','"+result+"','"+add1+"') ";
    pst=conn.prepareStatement(sql);

    pst.execute();

}

}catch (Exception e)

```

```

{
    JOptionPane.showMessageDialog(null,e);
}

finally {

    try{
        rs.close();
        pst.close();

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

    Date currentDate = GregorianCalendar.getInstance().getTime();
    DateFormat df = DateFormat.getDateInstance();
    String dateString = df.format(currentDate);

    Date d = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    String timeString = sdf.format(d);

    String value0 = timeString;
    String value1 = dateString;
    String val = txt_emp.getText().toString();
    try{

        String reg= "insert into Audit (id, date, status) values ('"+val+"','"+value0+" /
"+value1+"','Added Record')";
        pst=conn.prepareStatement(reg);
        pst.execute();

    }
    catch (Exception e)

    {
        JOptionPane.showMessageDialog(null,e);
    }
    finally {

        try{
            rs.close();
            pst.close();

        }

```

```

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

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

    gender ="Female";
    r_female.setSelected(true);
    r_male.setSelected(false);

}

private void r_maleActionPerformed(java.awt.event.ActionEvent evt) {
    gender ="Male";
    r_male.setSelected(true);
    r_female.setSelected(false);
}

private void btn_attachActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser chooser = new JFileChooser();
    chooser.showOpenDialog(null);
    File f = chooser.getSelectedFile();
    filename = f.getAbsolutePath();
    ImageIcon imageIcon = new ImageIcon(new
ImageIcon(filename).getImage().getScaledInstance(img.getWidth(), img.getHeight(),
Image.SCALE_DEFAULT));
    img.setIcon(imageIcon);
}

private void txt_emailKeyReleased(java.awt.event.KeyEvent evt) {

    String PATTERN="^[a-zA-Z0-9]{0,30}[@][a-zA-Z0-9]{0,10}[.][a-zA-Z]{0,5}$";
    Pattern patt=Pattern.compile(PATTERN);
    Matcher match=patt.matcher(txt_email.getText());
    if(!match.matches()){
        elab.setText("Email is Invalid");
    }
    else{
        elab.setText(null);
    }
}
}

```

```

private void txt_telKeyReleased(java.awt.event.KeyEvent evt) {

    String PATTERN="^[0-9]{0,10}$";
    Pattern patt=Pattern.compile(PATTERN);
    Matcher match=patt.matcher(txt_tel.getText());
    if(!match.matches()){
        clab.setText("Contact No. is Invalid");
    }
    else{
        clab.setText(null);
    }
}

```

```

private void txt_pcKeyReleased(java.awt.event.KeyEvent evt) {

    String PATTERN="^[0-9]{0,6}$";
    Pattern patt=Pattern.compile(PATTERN);
    Matcher match=patt.matcher(txt_pc.getText());
    if(!match.matches()){
        pclab.setText("Postal Code. is Invalid");
    }
    else{
        pclab.setText(null);
    }
}

```

```

private ImageIcon format =null;
//strin filename
String filename = null;
byte[] person_image = null;

private String gender;

}

```

**7. searchEmployee.java**

```

import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import javax.imageio.ImageIO;
import javax.swing.*;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author ATHASHRI
 */

public class searchEmployee extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public searchEmployee() {
        initComponents();
        conn=db.java_db();
        Update_table();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);

        txt_emp.setText(String.valueOf(Emp.empId).toString());
    }
    private void Update_table() {
        try{

```

```

String sql ="select * from Staff_information";
pst=conn.prepareStatement(sql);
rs=pst.executeQuery();

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

    try{
        rs.close();
        pst.close();

    }
    catch(Exception e){

    }
}

}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {

    JFileChooser chooser = new JFileChooser();
    chooser.showOpenDialog(null);
    File f = chooser.getSelectedFile();

    filename =f.getAbsolutePath();
    ImageIcon imageIcon = new ImageIcon(new
ImageIcon(filename).getImage().getScaledInstance(img.getWidth(), img.getHeight(),
Image.SCALE_DEFAULT));
    img.setIcon(imageIcon);

}

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

    gender ="Female";
}

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

    gender ="Male";
}

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

```

```

int p = JOptionPane.showConfirmDialog(null, "Are you sure you want to delete
record?", "Delete", JOptionPane.YES_NO_OPTION);
if(p==0){
    Date currentDate = GregorianCalendar.getInstance().getTime();
    DateFormat df = DateFormat.getDateInstance();
    String dateString = df.format(currentDate);

    Date d = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    String timeString = sdf.format(d);

    String value0 = timeString;
    String value1 = dateString;
    String val = txt_emp.getText().toString();
    try{
        String reg= "insert into Audit (id, date, status) values ('"+val+"','"+value0+" /
"+value1+"','Deleted Record')";
        pst=conn.prepareStatement(reg);
        pst.execute();
    }
    catch (Exception e)

    {
        JOptionPane.showMessageDialog(null,e);
    }
    String sql ="delete from Staff_information where id=? ";
    try{
        pst=conn.prepareStatement(sql);
        pst.setString(1, txt_id.getText());
        pst.execute();

        JOptionPane.showMessageDialog(null,"Record Deleted");

        txt_id.setText("");
        txt_firstname.setText("");
        txt_surname.setText("");
        txt_tel.setText("");
        txt_dob.setText("");
        txt_email.setText("");
        txt_address.setText("");
        txt_dep.setText("");
        txt_status.setText("");
        txt_salary.setText("");
        txt_add2.setText("");
        txt_pc.setText("");
        txt_desig.setText("");
        txt_job.setText("");

```



```

        txt_apr.setText("");
        txt_doj.setText("");
        img.setIcon(null);

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);
    } finally {

        try {
            rs.close();
            pst.close();

        }
        catch (Exception e) {

        }
    }
}
try {

    String sq = "delete from User where id =?";
    pst = conn.prepareStatement(sq);
    pst.setString(1, txt_id.getText());
    pst.execute();

} catch (Exception e) {

}

}

}

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

    int p = JOptionPane.showConfirmDialog(null, "Are you sure you want to update?", "Update
Record", JOptionPane.YES_NO_OPTION);
    if (p == 0) {

        try {

            String value1 = txt_firstname.getText();
            String value2 = txt_surname.getText();
            String value3 = txt_dob.getText();
            String value4 = txt_id.getText();
            String value5 = txt_email.getText();
            String value6 = txt_tel.getText();

```

```

String value7 = txt_address.getText();
String value8 = txt_dep.getText();
String value9 = txt_add2.getText();
String value10 = txt_apartment.getText();
String value11 = txt_pc.getText();
String value12 = txt_desig.getText();
String value13 = txt_status.getText();
String value14 = txt_salary.getText();
String value15 = txt_job.getText();
String value16 = txt_doj.getText();

String sql= "update Staff_information set id='"+value4+"',first_name='"+value1+"',
surname='"+value2+"', "
    + "Dob='"+value3+"',Email='"+value5+"',Telephone='"+value6+"', "
    + "Address='"+value7+"',Department='"+value8+"', Address2 = '"+value9+"', "
    + "Apartment = '"+value10+"', Post_code = '"+value11+"', "
    + "Designation = '"+value12+"', Status = '"+value13+"', Salary = '"+value14+"', job_title
='"+value15+"', Date_Hired = '"+value16+" ' "
    + " where id='"+value4+" ' ";

pst=conn.prepareStatement(sql);
pst.execute();
JOptionPane.showMessageDialog(null, "Record Updated");

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

Date currentDate = GregorianCalendar.getInstance().getTime();
DateFormat df = DateFormat.getDateInstance();
String dateString = df.format(currentDate);

Date d = new Date();
SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
String timeString = sdf.format(d);

String value0 = timeString;
String values = dateString;
String val = txt_emp.getText().toString();
try {
    String reg= "insert into Audit (id, date, status) values ('"+val+"','"+value0+" /
"+values+"','Updated Record)";
    pst=conn.prepareStatement(reg);
    pst.execute();
}
catch (Exception e)

```

```

    {
        JOptionPane.showMessageDialog(null,e);
    }
    finally {

        try{
            rs.close();
            pst.close();
        }
        catch(Exception e){

        }
    }
}

```

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

```

    txt_id.setText("");
    txt_firstname.setText("");
    txt_surname.setText("");
    txt_tel.setText("");
    txt_dob.setText("");
    txt_email.setText("");
    txt_address.setText("");
    txt_dep.setText("");
    txt_status.setText("");
    txt_salary.setText("");
    txt_add2.setText("");
    txt_pc.setText("");
    txt_desig.setText("");
    txt_job.setText("");
    txt_apr.setText("");
    txt_doj.setText("");
    img.setIcon(null);

    r_male.setSelected(false);
    r_female.setSelected(false);
}

```

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

```

    try{
        Statement s=db.java_db().createStatement();
        ResultSet rs= s.executeQuery("SELECT * FROM Staff_information where
id='"+txt_id.getText()+"'");

```

```

if(rs.next()){
    txt_firstname.setText(rs.getString("first_name"));
    txt_surname.setText(rs.getString("surname"));
    txt_tel.setText(rs.getString("Telephone"));
    txt_dob.setText(rs.getString("Dob"));
    txt_email.setText(rs.getString("Email"));
    txt_address.setText(rs.getString("Address"));
    txt_dep.setText(rs.getString("Department"));
    txt_status.setText(rs.getString("Status"));
    txt_salary.setText(rs.getString("Salary"));
    txt_add2.setText(rs.getString("Address2"));
    txt_pc.setText(rs.getString("Post_code"));
    txt_desig.setText(rs.getString("Designation"));
    txt_job.setText(rs.getString("Job_title"));
    txt_apr.setText(rs.getString("Apartment"));
    txt_doj.setText(rs.getString("Date_hired"));
    String imgStr =rs.getString("Image");
    Image image=ImageIO.read(new File(imgStr));
    ImageIcon imageIcon = new ImageIcon(new
ImageIcon(image).getImage().getScaledInstance(img.getWidth(), img.getHeight(),
Image.SCALE_SMOOTH));
    img.setIcon(imageIcon);

    String add =rs.getString("Gender");
    if (add.equals("Male")){
        r_male.setSelected(true);
        r_female.setSelected(false);
    }
    else{
        r_female.setSelected(true);
        r_male.setSelected(false);
    }
}

    }catch (SQLException e){
        System.out.println(e);
    } catch (IOException ex) {
        Logger.getLogger(searchEmployee.class.getName()).log(Level.SEVERE, null, ex);
    }
}

String filename = null;
private String gender;

}

```

**8. Allowance.java**

```

import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import javax.imageio.ImageIO;
import javax.swing.*;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author ATHASHRI
 */

public class Allowance extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public Allowance() {
        initComponents();
        conn=db.java_db();
        Update_table();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);

        txt_emp.setText(String.valueOf(Emp.empId).toString());

```

```

    }
    private void Update_table() {
try{
    String sql ="select * from allowance";
    pst=conn.prepareStatement(sql);
    rs=pst.executeQuery();
    table_allowance.setModel(DbUtils.resultSetToTableModel(rs));
}
catch(Exception e){
JOptionPane.showMessageDialog(null, e);
}
finally {

    try{
        rs.close();
        pst.close();

    }
    catch(Exception e){

    }
}
}

private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {

    try{

        String sql ="select * from Staff_information where id=? ";

        pst=conn.prepareStatement(sql);
        pst.setString(1,txt_search.getText());
        rs=pst.executeQuery();
        if(rs.next()){
            String add1 =rs.getString("id");
            txt_empid.setText(add1);

            String add2 =rs.getString("first_name");
            txt_firstname.setText(add2);

            String add3 =rs.getString("surname");
            txt_surname.setText(add3);

            String add4 =rs.getString("Dob");
            txt_dob.setText(add4);

            String add5 =rs.getString("Salary");

```

```

        txt_salary.setText(add5);

        String add6 = rs.getString("Department");
        txt_dept.setText(add6);
    } else {
        txt_empid.setText("");
        txt_firstname.setText("");
        txt_surname.setText("");
        txt_dob.setText("");
        txt_salary.setText("");
        txt_dept.setText("");
    }

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

    try {

        rs.close();
        pst.close();

    }
    catch (Exception e) {

    }

}
}

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

    float salary = Float.parseFloat(txt_salary.getText());

    int overtime = Integer.parseInt(txt_hw.getText());

    double eight = 8;
    double days = 25;
    double dbop = 0;
    double overtimeRate = 1.5;

    double Total_Overtime = overtime * overtimeRate;
    String x = String.valueOf(Total_Overtime);
    txt_total_overtime.setText(x);
}

```

```

        dbop = salary /days/eight;
        String s = String.valueOf(dbop);
        txt_per.setText(s);

        int med = Integer.parseInt(txt_med.getText());
        int bonus = Integer.parseInt(txt_bonus.getText());
        int other = Integer.parseInt(txt_other.getText());
        int f = med+bonus+other;
        double calc = Total_Overtime * dbop+f;
        String c = String.valueOf(calc);
        lbl_total.setText(c);

    }

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

        txt_empid.setText("");
        txt_firstname.setText("");
        txt_surname.setText("");
        txt_salary.setText("");
        txt_dob.setText("");
        txt_dept.setText("");
        txt_med.setText("0");
        txt_bonus.setText("0");
        txt_other.setText("0");
        txt_hw.setText("0");
        txt_per.setText("0");
        txt_total_overtime.setText("0");
        lbl_total.setText("0.00");

    }

    private void btn_saveActionPerformed(java.awt.event.ActionEvent evt) {
        if (txt_empid.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Employee id Field is empty");
        } else if (txt_firstname.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "First name Field is empty");

        } else if (txt_surname.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Surname name Field is empty");

        }
        else if (txt_dob.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Date of Birth name Field is empty");

        }
        else if (txt_salary.getText().equals("")) {

```



```

JOptionPane.showMessageDialog(null, "Salary Field is empty");

}
else if (txt_dept.getText().equals("")) {
    JOptionPane.showMessageDialog(null, "Department Field is empty");

}

else {

    int p = JOptionPane.showConfirmDialog(null, "Are you sure you want to save record?", "Add
Record",JOptionPane.YES_NO_OPTION);
    if(p==0){
        try {

            int value = Emp.empId;
            String value1 = txt_salary.getText();
            String value2 = txt_bonus.getText();
            String value3 = txt_med.getText();
            String value4 = txt_other.getText();
            String value5 = txt_per.getText();
            String value6 = txt_hw.getText();
            String value7 = lbl_total.getText();
            String value8 = txt_empid.getText();
            String value9 = txt_firstname.getText();
            String value10 = txt_surname.getText();
            Statement stm = conn.createStatement();
            rs = stm.executeQuery("select max(id) as id from Allowance");
            int id = 0;
            if(rs.next()){
                id = rs.getInt("id") + 1;
                System.out.println(id);
            }
            String sql= "insert into Allowance
(created_by,emp_id,overtime,medical,bonus,other,salary,rate,total_allowance,firstname,surname,id)
values
('"+value+"','"+value8+"','"+value6+"','"+value3+"','"+value2+"','"+value4+"','"+value1+"','"+value5+"
','"+value7+"','"+value9+"','"+value10+"','"+ id + "')";

            pst=conn.prepareStatement(sql);
            pst.execute();
            JOptionPane.showMessageDialog(null, "Allowance Added");

        }
        catch (Exception e)

```

```

    {
        JOptionPane.showMessageDialog(null,e);
    }
    try{
        Date currentDate = GregorianCalendar.getInstance().getTime();
        DateFormat df = DateFormat.getDateInstance();
        String dateString = df.format(currentDate);

        Date d = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
        String timeString = sdf.format(d);

        String value0 = timeString;
        String values = dateString;
        String val = txt_emp.getText().toString();

        String reg= "insert into Audit (id, date, status) values ('"+val+"','"+value0+" /
"+values+"','Updated Allowance Record)";
        pst=conn.prepareStatement(reg);
        pst.execute();
    }
    catch (Exception e)

    {
        JOptionPane.showMessageDialog(null,e);
    }
    finally {

        try{
            rs.close();
            pst.close();

        }
        catch(Exception e){

        }

    }
}
}
}
Update_table();
}

```

**9. updateSalary.java**

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import javax.imageio.ImageIO;
import javax.swing.*;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author ATHASHRI
 */

public class updateSalary extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public updateSalary() {
        initComponents();
        conn=db.java_db();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);

        txt_emp.setText(String.valueOf(Emp.empId).toString());
    }

    private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {

        try{

```

```

String sql ="select * from Staff_information where id=? ";

pst=conn.prepareStatement(sql);
pst.setString(1,txt_search.getText());
rs=pst.executeQuery();

if(rs.next()){
    String add1 =rs.getString("id");
    txt_empid.setText(add1);

    String add2 =rs.getString("first_name");
    txt_firstname.setText(add2);

    String add3 =rs.getString("surname");
    txt_surname.setText(add3);

    String add4 =rs.getString("Dob");
    txt_dob.setText(add4);

    String add5 =rs.getString("Salary");
    txt_salary.setText(add5);

    String add8 =rs.getString("Department");
    txt_dept.setText(add8);
} else {
    txt_empid.setText("");
    txt_firstname.setText("");
    txt_surname.setText("");
    txt_dob.setText("");
    txt_salary.setText("");
    txt_dept.setText("");
}

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

    try{

        rs.close();
        pst.close();

    }
    catch(Exception e){

    }

}

```

```

    }
}

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

    r_percentage.setSelected(true);
    r_amount.setSelected(false);

    txt_dept2.setEnabled(false);
    txt_dept1.setEditable(true);
    txt_dept1.setEnabled(true);
    txt_dept2.setText("");

}

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

    r_amount.setSelected(true);
    r_percentage.setSelected(false);
    txt_dept1.setEnabled(false);
    txt_dept2.setEditable(true);
    txt_dept2.setEnabled(true);
    txt_dept1.setText("");

}

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

    int p = JOptionPane.showConfirmDialog(null, "Are you sure you want to update salary?", "Update
Record",JOptionPane.YES_NO_OPTION);
    if(p==0){
        try{
            float salary = Float.parseFloat(txt_salary.getText());

            if(r_percentage.isSelected()== true){
                int getPercentage = Integer.parseInt(txt_dept1.getText());
                float calcPercentage = salary /100 * getPercentage + salary;
                String xP = String.valueOf(calcPercentage);
                txt_salary.setText(xP);
            }

            else if(r_amount.isSelected()==true){
                int getAmt = Integer.parseInt(txt_dept2.getText());
                float calcAmount = salary + getAmt;
                String xA = String.valueOf(calcAmount);

```

```

        txt_salary.setText(xA);
    }
} catch(Exception e){
    JOptionPane.showMessageDialog(null,e);
}
try{
    Date currentDate = GregorianCalendar.getInstance().getTime();
    DateFormat df = DateFormat.getDateInstance();
    String dateString = df.format(currentDate);

    Date d = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    String timeString = sdf.format(d);

    String value0 = timeString;
    String values = dateString;
    String val = txt_emp.getText().toString();

    String reg= "insert into Audit (id, date, status) values ('"+val+"','"+value0+" /
"+values+"','Updated Salary Record)";
    pst=conn.prepareStatement(reg);
    pst.execute();
}
catch (Exception e)

{
    JOptionPane.showMessageDialog(null,e);
}
try{

    String value1 = txt_empid.getText();
    String value2 = txt_salary.getText();

    String sql= "update Staff_information set id='"+value1+"',Salary='"+value2+"' where
id='"+value1+"'";

    pst=conn.prepareStatement(sql);
    pst.execute();
    JOptionPane.showMessageDialog(null, "Record Updated");

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

```

```

finally {

    try{
        rs.close();
        pst.close();

    }
    catch(Exception e){

    }

}
}
}

```

### 10. employeeDeductions.java

```

import java.awt.Dimension;
import java.awt.Toolkit;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.GregorianCalendar;
import javax.swing.JOptionPane;

/**
 *
 * @author ATHASHRI
 */
public class employeeDeductions extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public employeeDeductions() {
        initComponents();
        conn=db.java_db();

        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);
    }
}

```

```

        lbl_emp.setText(String.valueOf(Emp.empId).toString());

    }

    private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {

        try{

            String sql ="select * from Staff_information where id=? ";

            pst=conn.prepareStatement(sql);
            pst.setString(1,txt_search.getText());
            rs=pst.executeQuery();
            if(rs.next()){
                String add1 =rs.getString("id");
                txt_id.setText(add1);

                String add2 =rs.getString("first_name");
                txt_firstname.setText(add2);

                String add3 =rs.getString("surname");
                txt_surname.setText(add3);

                String add4 =rs.getString("Dob");
                txt_dob.setText(add4);

                String add5 =rs.getString("Department");
                txt_dep.setText(add5);

                String add7 =rs.getString("Salary");
                txt_salary.setText(add7);

                String add8 =rs.getString("Status");
                txt_status.setText(add8);

                String add9 =rs.getString("Date_hired");
                txt_doj.setText(add9);

                String add10 =rs.getString("job_title");
                txt_job.setText(add10);

                String add17 =rs.getString("Designation");
                txt_desig.setText(add17);
            } else {
                txt_id.setText("");
                txt_firstname.setText("");
            }
        }
    }

```



```

        txt_surname.setText("");
        txt_dob.setText("");
        txt_dep.setText("");
        txt_status.setText("");
        txt_salary.setText("");
        txt_desig.setText("");
        txt_job.setText("");
        txt_doj.setText("");
    }

} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "No Data");
}
finally {

    try {

        rs.close();
        pst.close();

    }
    catch (Exception e) {

    }
}
}

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

    int p = JOptionPane.showConfirmDialog(null, "Are you sure you want to add record?", "Add
Record", JOptionPane.YES_NO_OPTION);
    if (p == 0) {

        String value3 = lbl_emp.getText();
        try {
            Statement stm = conn.createStatement();
            int id = 0;
            rs = stm.executeQuery("select max(id) as id from Deductions");
            if (rs.next()) {
                id = rs.getInt("id") + 1;
            }

            if (lbl_total.getText().equals("0.00")) {
                JOptionPane.showMessageDialog(null, "Enter some amount to deduct!");
            } else {
                String sql = "insert into Deductions
(firstname,surname,salary,deduction_amount,deduction_reason,emp_id,id,made_by) values

```

```

(?,?,?,?,""+value3+"");
    pst=conn.prepareStatement(sql);
    pst.setString(1,txt_firstname.getText());
    pst.setString(2,txt_surname.getText());
    pst.setString(3,txt_salary.getText());
    pst.setString(4,lbl_total.getText());
    pst.setString(5,txt_reason.getText());
    pst.setString(6,txt_id.getText());
    pst.setInt(7,id);

    pst.execute();
    JOptionPane.showMessageDialog(null,"Data is saved successfully");
}

}
catch (Exception e)

{
    JOptionPane.showMessageDialog(null,e);
}
try{
    Date currentDate = GregorianCalendar.getInstance().getTime();
    DateFormat df = DateFormat.getDateInstance();
    String dateString = df.format(currentDate);

    Date d = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
    String timeString = sdf.format(d);

    String value0 = timeString;
    String values = dateString;
    String val = lbl_emp.getText().toString();

    String reg= "insert into Audit (id, date, status) values (""+val+"",""+value0+" /
"+values+"','Updated Deduction Record')";
    pst.execute();
}
catch (Exception e)

{
    JOptionPane.showMessageDialog(null,e);
}
finally {

    try{
        rs.close();
        pst.close();

    }

```

```

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

```

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

```
    float salary = Float.parseFloat(txt_salary.getText());
```

```
    if(r_percentage.isSelected()== true){
        int percentage = Integer.parseInt(txt_percentage.getText());
```

```
        float total_percentage_deduction = salary /100 * percentage;
        String x = String.valueOf(total_percentage_deduction);
        float sal = salary - total_percentage_deduction;
        lbl_total.setText(x);
        lbl_sal.setText(String.valueOf(sal));
    }

```

```
    if(r_amount2.isSelected()== true){
        int deduction = Integer.parseInt(txt_deduction.getText());
```

```
        float total_amount_deduction = salary - deduction;
        String s = String.valueOf(total_amount_deduction);
```

```
        lbl_sal.setText(s);
        lbl_total.setText(String.valueOf(deduction));
```

```
    }
}

```

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

```
    txt_id.setText("");
    txt_firstname.setText("");
    txt_surname.setText("");
    txt_dob.setText("");
    txt_dep.setText("");
    txt_status.setText("");
    txt_salary.setText("");
    txt_desig.setText("");
    txt_job.setText("");

```

```
txt_doj.setText("");
lbl_total.setText("0.00");
txt_percentage.setText("");
txt_deduction.setText("");
txt_reason.setText("");
txt_search.setText("");
lbl_sal.setText("0.00");
}

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

    r_percentage.setSelected(true);
    r_amount2.setSelected(false);
    //r_amount.setEnabled(false);
    txt_deduction.setEnabled(false);
    txt_percentage.setEditable(true);
    txt_percentage.setEnabled(true);
    txt_deduction.setText("");

}

private void r_amount2ActionPerformed(java.awt.event.ActionEvent evt) {

    r_amount2.setSelected(true);
    r_percentage.setSelected(false);
    //r_percentage.setEnabled(false);
    txt_percentage.setEnabled(false);
    txt_deduction.setEditable(true);
    txt_deduction.setEnabled(true);
    txt_percentage.setText("");

}
```

**11. searchEmpSalarySlip.java**

```

import com.itextpdf.text.Document;
import com.itextpdf.text.FontFactory;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.PdfWriter;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Toolkit;
import java.io.File;
import java.io.FileOutputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Date;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

/**
 *
 * @author ATHASHRI
 */

public class searchEmpSalarySlip extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public searchEmpSalarySlip() {
        initComponents();
        conn=db.java_db();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);
    }

    private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {

        try{

            String sql ="select * from Staff_information where id=? ";

            pst=conn.prepareStatement(sql);
            pst.setString(1,txt_search.getText());

```

```

rs=pst.executeQuery();

if(rs.next()){
    String add1 =rs.getString("id");
    txt_id.setText(add1);

    String add2 =rs.getString("first_name");
    txt_firstname.setText(add2);

    String add3 =rs.getString("surname");
    txt_surname.setText(add3);

    String add4 =rs.getString("Dob");
    txt_dob.setText(add4);

    String add5 =rs.getString("Department");
    txt_dep.setText(add5);

    String add7 =rs.getString("Salary");
    txt_salary.setText(add7);

    String add8 =rs.getString("Status");
    txt_status.setText(add8);

    String add9 =rs.getString("Date_hired");
    txt_doj.setText(add9);

    String add10 =rs.getString("job_title");
    txt_job.setText(add10);

    String add17 =rs.getString("Designation");
    txt_desig.setText(add17);
} else {
    txt_id.setText("");
    txt_firstname.setText("");
    txt_surname.setText("");
    txt_dob.setText("");
    txt_dep.setText("");
    txt_status.setText("");
    txt_salary.setText("");
    txt_desig.setText("");
    txt_job.setText("");
    txt_doj.setText("");
}

} catch(Exception e){

```

```

JOptionPane.showMessageDialog(null, "No Data");
}
finally {

    try{

        rs.close();
        pst.close();

    }
    catch(Exception e){

    }

}
}

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

    String value = txt_firstname.getText();
    String value0 = txt_surname.getText();
    String value1 = txt_id.getText();
    String value2 = txt_desig.getText();
    String value3 = txt_dep.getText();


    JFileChooser dialog = new JFileChooser();
    dialog.setSelectedFile(new File(value + " " + value0+"-Salary Slip"+"pdf"));
    int dialogResult = dialog.showSaveDialog(null);
    if (dialogResult==JFileChooser.APPROVE_OPTION){
        String filePath = dialog.getSelectedFile().getPath();

    try {

        String sql="select * from Deductions where emp_id = '"+value1+"'";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        String val = "0";
        String reason = "";
        if(rs.next()){
            val = rs.getString(5);
            reason = rs.getString(6);
        }
        rs.close();
        pst.close();

        String sq ="select * from Allowance where emp_id = '"+value1+"'";

```

```

pst=conn.prepareStatement(sq);
rs=pst.executeQuery();

float calcTotal = 0.0f;
float x = 0.0f;
if(rs.next()){
    calcTotal = Float.parseFloat(txt_salary.getText());
    x = Float.valueOf(rs.getString(9));
}

float v = Float.parseFloat(val);
float total = calcTotal+ x -v;

Document myDocument = new Document();
PdfWriter myWriter = PdfWriter.getInstance(myDocument, new FileOutputStream(filePath));
myDocument.open();

myDocument.add(new Paragraph("PAY
SLIP",FontFactory.getFont(FontFactory.TIMES_BOLD,20,Font.BOLD)));
myDocument.add(new Paragraph(new Date().toString()));
myDocument.add(new Paragraph(".....
....."));
myDocument.add((new Paragraph("EMPLOYEE
DETAILS",FontFactory.getFont(FontFactory.TIMES_ROMAN,15,Font.BOLD))));
myDocument.add((new Paragraph("Name of Employee: " +value + "
"+value0,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add((new Paragraph("Designation:
"+value2,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add((new Paragraph("Department:
"+value3,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add(new Paragraph(".....
....."));
myDocument.add(new
Paragraph("SALARY",FontFactory.getFont(FontFactory.TIMES_ROMAN,15,Font.BOLD))));
myDocument.add(new Paragraph("Basic Salary:
Rs."+calcTotal,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add(new Paragraph("Overtime: "+rs.getString(2)+"
Hours",FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add(new Paragraph("Medical: Rs."
+rs.getString(3),FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add(new Paragraph("Bonus:
Rs."+rs.getString(4),FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add(new Paragraph("Other:
Rs."+rs.getString(5),FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN))));
myDocument.add(new Paragraph(".....
....."));

```



```

        myDocument.add(new
Paragraph("DEDUCTION",FontFactory.getFont(FontFactory.TIMES_ROMAN,15,Font.BOLD)));
        myDocument.add(new Paragraph("Deduction Details:
"+reason,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN)));
        myDocument.add(new Paragraph("Total Deductions : Rs."+val
,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN)));
        myDocument.add(new Paragraph(".....
....."));
        myDocument.add(new Paragraph("TOTAL
PAYMENT",FontFactory.getFont(FontFactory.TIMES_ROMAN,15,Font.BOLD)));
        myDocument.add(new Paragraph("Total Earnings:
"+rs.getString(9),FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN)));
        myDocument.add(new Paragraph("Net Pay : "
+total,FontFactory.getFont(FontFactory.TIMES_ROMAN,10,Font.PLAIN)));
        myDocument.add(new Paragraph(".....
....."));

        myDocument.newPage();
        myDocument.close();
        JOptionPane.showMessageDialog(null,"Report was successfully generated");

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

    }
    finally {

        try{
            rs.close();
            pst.close();

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

        }
    }
}
}
}

```

**12. Audit\_details.java**

```

import javax.swing.JOptionPane;
import java.awt.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Calendar;
import java.util.GregorianCalendar;
import javax.swing.*;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author ATHASHRI
 */

public class Audit_details extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs=null;
    PreparedStatement pst=null;

    public Audit_details() {
        initComponents();
        conn=db.java_db();
        Toolkit toolkit = getToolkit();
        Dimension size = toolkit.getScreenSize();
        setLocation(size.width/2 - getWidth()/2,
            size.height/2 - getHeight()/2);

        Update_table3();
    }

    private void Update_table3() {
        try{

            String sql ="select * from Audit";

            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            tbl_3.setModel(DbUtils.resultSetToTableModel(rs));

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

```

```

    }
    finally {

        try{
            rs.close();
            pst.close();

        }
        catch(Exception e){

        }
    }
}

private void txt_searchKeyReleased(java.awt.event.KeyEvent evt) {

    try{

        String sql ="select * from Audit where id=? ";

        pst=conn.prepareStatement(sql);
        pst.setString(1,txt_search.getText());
        rs=pst.executeQuery();
        tbl_3.setModel(DbUtils.resultSetToTableModel(rs));

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

        try{

            rs.close();
            pst.close();

        }
        catch(Exception e){

        }
    }
}

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

```

```
Update_table3();  
txt_search.setText("");  
}
```

## 6. RESULTS

### 6.1 Validation and Naming Conventions.

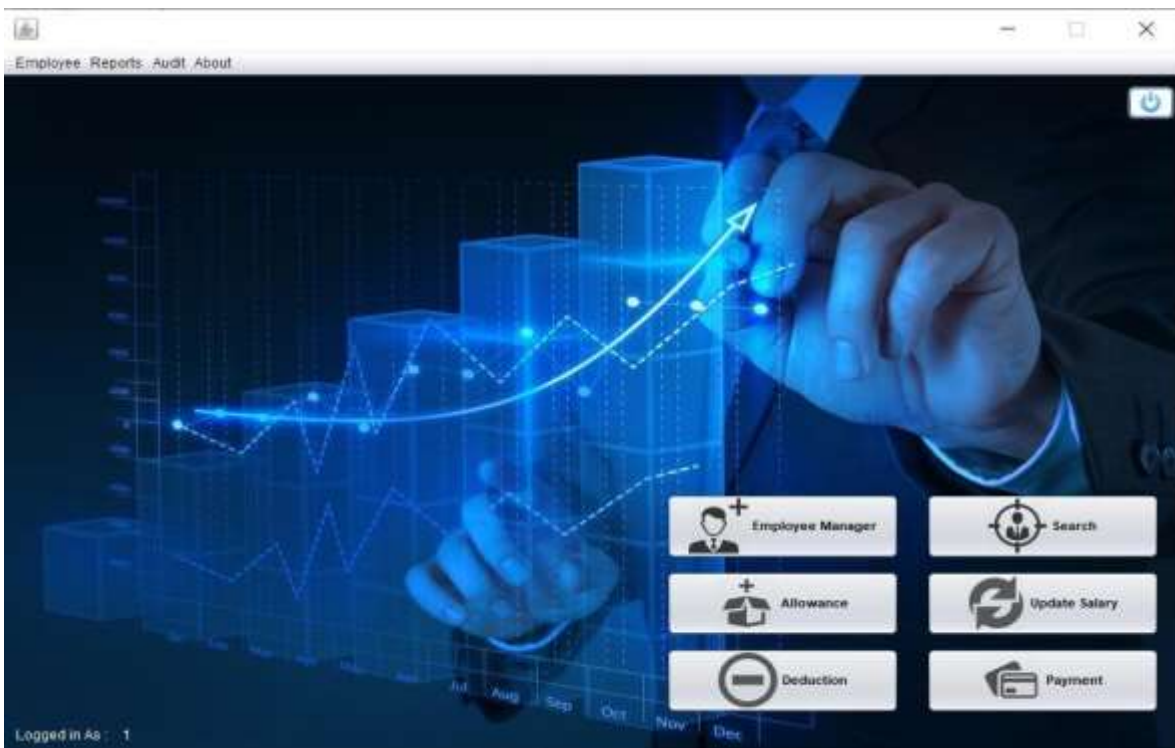
Sr. No	Control Id	Validation used	Reason
1	Username	InputVerifier	Username field is empty
2	Password	InputVerifier	Password field is empty
3	Email	RequiredFieldValidator	Email cannot be empty
4	Contact Number	RequiredFieldValidator	Phone number cannot be empty

## 6.2 Screenshots

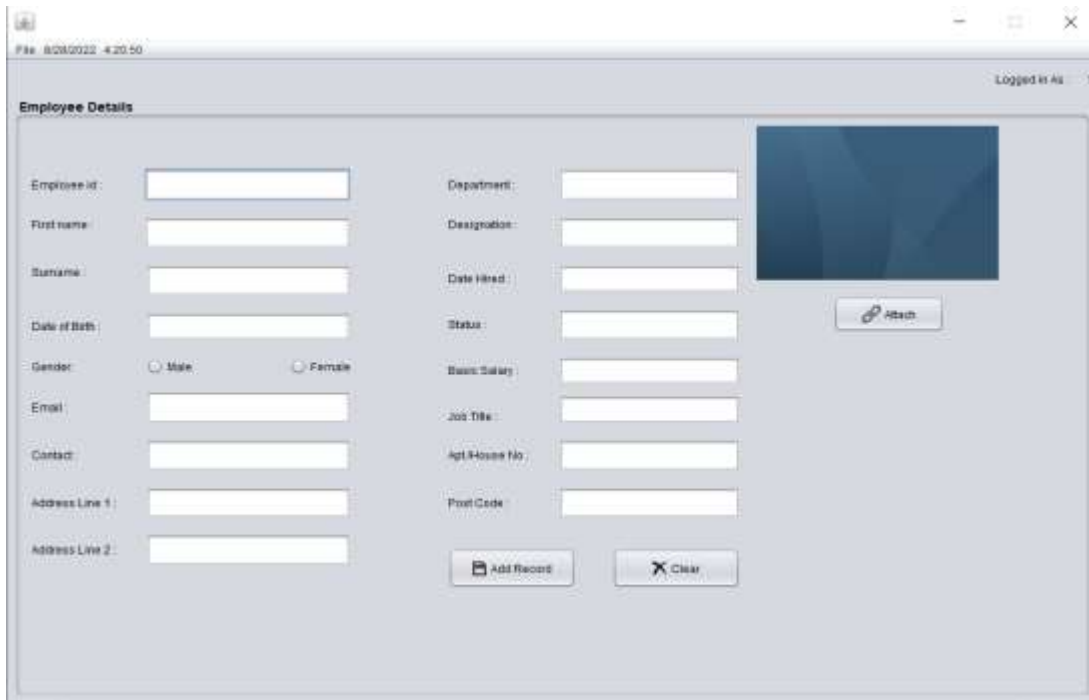
### 6.2.1 Login Page



### 6.2.2 MainMenu Page



### 6.2.3 Employee Registration Page



The screenshot shows a web application window titled "Employee Details". The window has a menu bar with "File" and a status bar with "Logged in As : 1". The main content area contains a form for employee registration. The form is organized into two columns of input fields. The left column includes fields for Employee id, First name, Surname, Date of Birth, Gender (with radio buttons for Male and Female), Email, Contact, Address Line 1, and Address Line 2. The right column includes fields for Department, Designation, Date Hired, Status, Basic Salary, Job Title, Apt./House No., and Post Code. To the right of these fields is a large blue square placeholder for a profile picture, with an "Attach" button below it. At the bottom of the form are two buttons: "Add Record" and "Clear".

Employee Details

File 8/28/2022 4:20:50 Logged in As : 1

Employee id:  Department:

First name:  Designation:

Surname:  Date Hired:

Date of Birth:  Status:

Gender: ☐ Male ☐ Female Basic Salary:

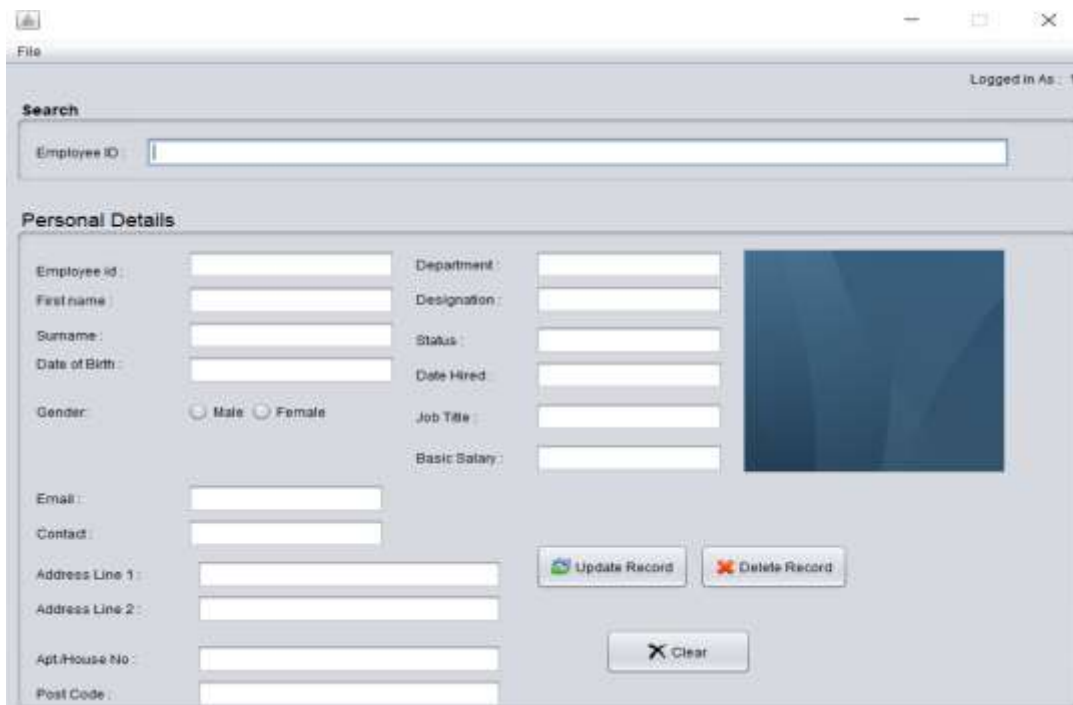
Email:  Job Title:

Contact:  Apt./House No.:

Address Line 1:  Post Code:

Address Line 2:

### 6.2.4 Search Employee Page



The screenshot shows a web application window titled "Search". The window has a menu bar with "File" and a status bar with "Logged in As : 1". The main content area contains a search form. At the top is a search bar labeled "Employee ID" with a text input field. Below this is a section titled "Personal Details" which contains a form for employee information. The form is organized into two columns of input fields. The left column includes fields for Employee id, First name, Surname, Date of Birth, Gender (with radio buttons for Male and Female), Email, Contact, Address Line 1, and Address Line 2. The right column includes fields for Department, Designation, Status, Date Hired, Job Title, and Basic Salary. To the right of these fields is a large blue square placeholder for a profile picture. At the bottom of the form are three buttons: "Update Record", "Delete Record", and "Clear".

Search

File Logged in As : 1

Employee ID

Personal Details

Employee id:  Department:

First name:  Designation:

Surname:  Status:

Date of Birth:  Date Hired:

Gender: ☐ Male ☐ Female Job Title:

Basic Salary:

Email:

Contact:

Address Line 1:

Address Line 2:

Apt./House No.:

Post Code:

**6.2.5 Allowance Page**

**Search**

Employee ID:

**Please enter the amounts**

Employee id:  Overtime:  Total Overtime:

First name:  Medical:  Rate Per Hour:

Surname:  Bonus:

Date of Birth:  Other:

Basic Salary:

Department:

id	overt	med	bonus	other	emp	salary	rate	total	first	sur	crea
1	20	50	500	0	1	108	540.0	167	Ajay	Gan	2
2	542	0	0	0	2	800	300.0	243	Ankita	Gan	2
3	54	95	0	0	3	800	400.0	324	Ana	Gan	2
4	54	95	0	0	6	800	300.0	243	Aditi	Rao	2
5	82	5	2	0	9	546	273	336	Aust	Vo	2

**Total Amount : 0.00**

Logged in As : 2

**6.2.6 Salary Update Page**

**Search**

Employee ID:

Employee id:  Date of Birth:

First name:  Basic Salary:

Surname:  Department:

Update Salary by: ☐ Percentage (%) ☐ Amount

Percentage:  Amount:

Logged in As : 2

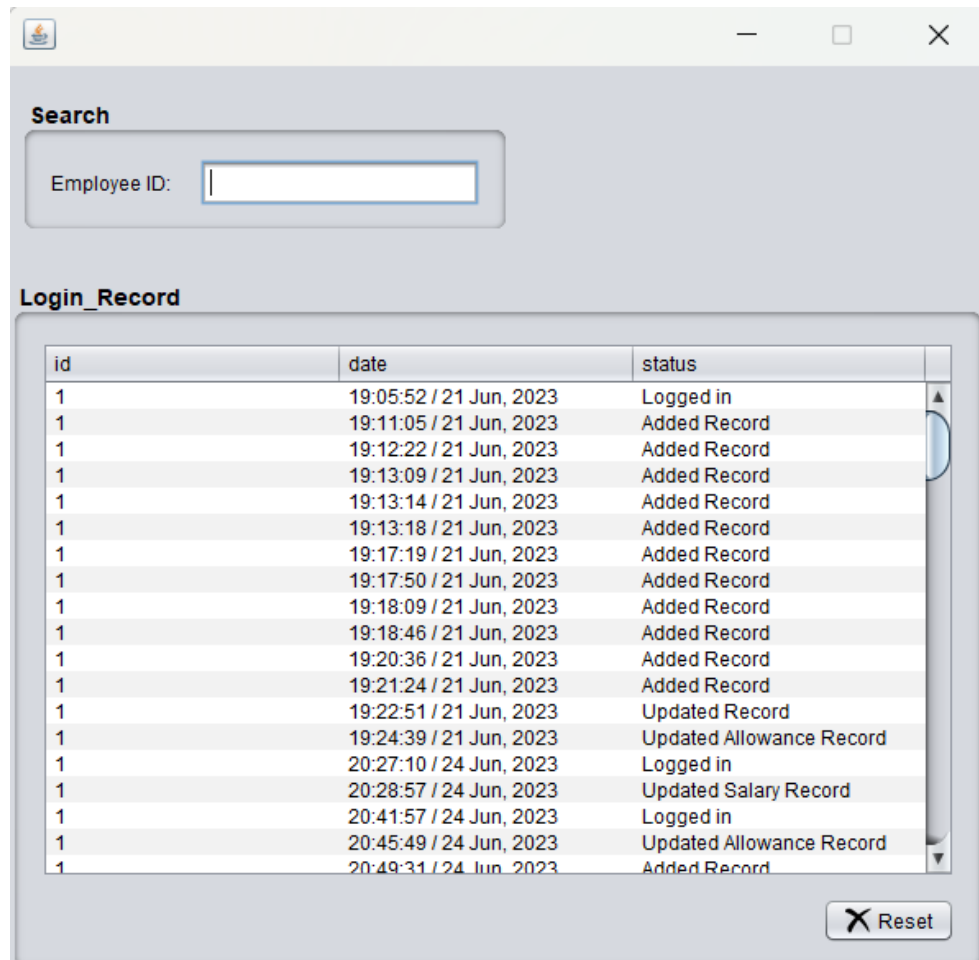


### 6.2.7 Deductions Page

The screenshot shows a web application window titled "Deductions Page". It features a "Search" section with a text input field for "Employee ID :". Below this, there are two columns of input fields: "Employee Id :", "First name :", "Surname :", "Date of Birth :", "Department :", "Designation :", "Status :", "Date Hired :", "Job Title :", and "Basic Salary :". At the bottom, there are radio buttons for "Update Salary by : Percentage (%)" and "Amount", with corresponding input fields. A "Reason:" input field is also present. On the right, it displays "Total Deduction: 0.00" and "Salary after deduction: 0.00". At the bottom, there are three buttons: "Save", "+ Calculate", and "Clear". The status "Logged in As : 1" is shown at the very bottom.

### 6.2.8 Payment Slip Page

The screenshot shows a web application window titled "Payment Slip Page". It features a "Search" section with a text input field for "Employee ID :". Below this, there are two columns of input fields: "Employee Id :", "First name :", "Surname :", "Date of Birth :", "Department :", "Designation :", "Status :", "Date Hired :", "Job Title :", and "Basic Salary :". At the bottom, there is a button labeled "Generate Slip" with a document icon.

**6.2.9** Audit Page

The screenshot displays a web application window titled "Audit Page". It features a "Search" section with a label "Employee ID:" and an adjacent text input field. Below this is a section titled "Login\_Record" containing a table with three columns: "id", "date", and "status". The table lists 20 records, showing a sequence of login and record updates for employee ID 1. A "Reset" button with a circular arrow icon is located at the bottom right of the window.

id	date	status
1	19:05:52 / 21 Jun, 2023	Logged in
1	19:11:05 / 21 Jun, 2023	Added Record
1	19:12:22 / 21 Jun, 2023	Added Record
1	19:13:09 / 21 Jun, 2023	Added Record
1	19:13:14 / 21 Jun, 2023	Added Record
1	19:13:18 / 21 Jun, 2023	Added Record
1	19:17:19 / 21 Jun, 2023	Added Record
1	19:17:50 / 21 Jun, 2023	Added Record
1	19:18:09 / 21 Jun, 2023	Added Record
1	19:18:46 / 21 Jun, 2023	Added Record
1	19:20:36 / 21 Jun, 2023	Added Record
1	19:21:24 / 21 Jun, 2023	Added Record
1	19:22:51 / 21 Jun, 2023	Updated Record
1	19:24:39 / 21 Jun, 2023	Updated Allowance Record
1	20:27:10 / 24 Jun, 2023	Logged in
1	20:28:57 / 24 Jun, 2023	Updated Salary Record
1	20:41:57 / 24 Jun, 2023	Logged in
1	20:45:49 / 24 Jun, 2023	Updated Allowance Record
1	20:49:31 / 24 Jun, 2023	Added Record

## **7. CONCLUSION AND FUTURE ENHANCEMENTS**

### **7.1 Future Enhancement**

- Adding a better GUI.
- User Verification.
- To make it more user friendly by improving the design of app.
- Attendance system.

### **7.2 Conclusion**

- No formal knowledge is needed for the user to use this system.
- This project simplifies the task of maintain records because of its user-friendly nature.
- This app helps your organization improve workforce productivity and boost overall well-being by tracking and monitoring the daily working activities of every employee.

## 8. **REFERENCES**

- <https://www.youtube.com>
- <https://www.w3schools.com>
- <https://www.stackoverflow.com>
- <https://www.github.com>

## 9. ANNEXURE

### Figures List:

Figure Number	Name of the figure	Page Number
4.2	Class Diagram	7
4.3	Use case Diagram	8
4.4	Sequence Diagram	9
4.5	Activity Diagram	10
4.6	State Diagram	11
4.7	Package Diagram	12
4.8	Component Diagram	12
4.9	Deployment Diagram	13

### Tables List:

Table Number	Name of the table	Page Number
4.1	Event Table	6
4.10.1	Database Design for Admin	14
4.10.2	Database Design for StuffInformation	14
4.10.3	Database Design for Allowance	15
4.10.4	Database Design for Deductions	15
4.10.5	Database Design for Audit	16
6.1	Validation and Naming Conventions	73