Employee Payroll System Project

Mohd Hozaifa
B.Tech CSE (AI) - IIMT College of Engineering

December 7, 2025

1 Introduction

This document contains the source code for the **Employee Payroll System**. The project simulates a real-world payroll management system utilizing Java OOP concepts. It handles full-time and part-time employees, dynamic salary calculations, and database management operations using ArrayLists.

2 Source Code

2.1 Employee.java (Abstract Class)

```
abstract class Employee {
    private String name;
    private int id;

    public Employee(String name, int id) {
        this.name = name;
        this.id = id;
    }
    public String getName() { return name; }
    public int getId() { return id; }

    public abstract double calculateSalary();

    @Override
    public String toString() {
        return "ID: " + id + " | Name: " + name + " | Salary: " +
   calculateSalary();
    }
}
```

Employee.java

2.2 FullTimeEmployee.java

```
class FullTimeEmployee extends Employee {
    private double monthlySalary;

```

```
    public FullTimeEmployee(String name, int id, double monthlySalary) {
        super(name, id);
        this.monthlySalary = monthlySalary;
    }
    @Override
    public double calculateSalary() {
        return monthlySalary;
    }
}
```

FullTimeEmployee.java

2.3 PartTimeEmployee.java

```
class PartTimeEmployee extends Employee {
    private int hoursWorked;
    private double hourlyRate;

    public PartTimeEmployee(String name, int id, int hoursWorked, double
    hourlyRate) {
        super(name, id);
        this.hoursWorked = hoursWorked;
        this.hourlyRate = hourlyRate;
    }
    @Override
    public double calculateSalary() {
        return hoursWorked * hourlyRate;
    }
}
```

PartTimeEmployee.java

2.4 PayrollSystem.java

```java
import java.util.ArrayList;

class PayrollSystem {
    private ArrayList<Employee> employeeList;

    public PayrollSystem() {
        employeeList = new ArrayList<>();
    }

    public void addEmployee(Employee employee) {
        employeeList.add(employee);
        System.out.println("Success: Employee " + employee.getName() + " added to system.");
    }

    public void removeEmployee(int id) {
        Employee employeeToRemove = null;
        for (Employee employee : employeeList) {
            if (employee.getId() == id) {
                employeeToRemove = employee;
                break;
            }
        }
        if (employeeToRemove != null) {
            employeeList.remove(employeeToRemove);
            System.out.println("Success: Employee ID " + id + " removed successfully.");
        } else {
            System.out.println("Error: Employee ID " + id + " not found in database.");
        }
    }

    public void displayEmployees() {
        System.out.println("\n--- Current Employee Database ---");
        for (Employee employee : employeeList) {
            System.out.println(employee);
        }
        System.out.println("---------------------------------");
    }
}
```

PayrollSystem.java

2.5 Main.java (Updated for Testing)

```java
public class Main {
    public static void main(String[] args) {
        PayrollSystem payroll = new PayrollSystem();

        System.out.println("*** SYSTEM INITIALIZATION STARTED ***\n");

        // Adding diverse employees
        payroll.addEmployee(new FullTimeEmployee("Vikas", 101, 75000.0));
```

```
        payroll.addEmployee(new PartTimeEmployee("Alexander", 102, 45, 120));
        payroll.addEmployee(new FullTimeEmployee("Roshni", 103, 82000.0));
        payroll.addEmployee(new PartTimeEmployee("Ankit", 104, 30, 150));
        payroll.addEmployee(new FullTimeEmployee("Rahul", 105, 60000.0));

        // Showing initial state
        payroll.displayEmployees();

        // Performing operations
        System.out.println("\n*** PROCESSING REMOVAL REQUESTS ***");
        payroll.removeEmployee(102); // Existing employee
        payroll.removeEmployee(104); // Existing employee
        payroll.removeEmployee(999); // Non-existing employee (Test Case)

        // Showing final state
        payroll.displayEmployees();

        System.out.println("\n*** SYSTEM SHUTDOWN COMPLETED ***");
    }
}
```

Main.java

3 Project Execution Output

The following log demonstrates the complete execution flow, covering data insertion, display, deletion, and validation logic.

```
*** SYSTEM INITIALIZATION STARTED ***

Success: Employee Vikas added to system.
Success: Employee Alexander added to system.
Success: Employee Roshni added to system.
Success: Employee Ankit added to system.
Success: Employee Rahul added to system.

--- Current Employee Database ---
ID: 101 | Name: Vikas     | Salary: 75000.0
ID: 102 | Name: Alexander | Salary: 5400.0
ID: 103 | Name: Roshni    | Salary: 82000.0
ID: 104 | Name: Ankit     | Salary: 4500.0
ID: 105 | Name: Rahul     | Salary: 60000.0
---------------------------------

*** PROCESSING REMOVAL REQUESTS ***
Success: Employee ID 102 removed successfully.
Success: Employee ID 104 removed successfully.
Error: Employee ID 999 not found in database.

--- Current Employee Database ---
ID: 101 | Name: Vikas     | Salary: 75000.0
ID: 103 | Name: Roshni    | Salary: 82000.0
ID: 105 | Name: Rahul     | Salary: 60000.0
---------------------------------

*** SYSTEM SHUTDOWN COMPLETED ***
```

Console Execution Log

Note: The output validates that the abstraction layer correctly calculates salaries for different employee types (Full-Time vs Part-Time) and the encapsulation logic protects the data integrity during removal operations.