2021

# Payroll Management System

WAJAHAT QAZI

# Table of Contents

**Chapter 1** ..... 3
Abstract: ..... 3
Introduction ..... 3
1.2. Project Outline ..... 3
1.2. Motivation ..... 4
1.3. Problem Identification: ..... 4
1.5 Project Goals ..... 5
1.6. Goals Criteria ..... 5
**Chapter 2** ..... 6
Comparison ..... 6
2.1 Why Customized Solution: ..... 6
2.1.2 Analysis of Different Salary Calculators: ..... 6
2.2. Employee Salary Management System: ..... 7
**Chapter 3** ..... 8
Requirement Analysis ..... 8
3.1. System Requirements: ..... 8
3.1.1 Functional Requirements: ..... 8
3.1.2 Non-functional Requirements: ..... 9
**Chapter 4** ..... 10
Case Study for Payroll Management System: ..... 10
**Business Rules:** ..... 10
Entity Relationship Diagram: ..... 11
Enhanced Entity Relationship Diagram: ..... 12
Relational Mapping from EERD: ..... 13
Payroll Database Creation ..... 14
**Employee Table:** ..... 14
DDL: ..... 14
DML: ..... 14
**Attendance Table:** ..... 15
DDL: ..... 15
DML: ..... 15
**Salary Table:** ..... 16

DDL: 16
DML: 16
**Designation Table** 17
DDL: 17
DML: 17
DQL**:** 18
Data Normalization: 19
Connecting Mysql with Java: 20
Graphical User Interfaces: 21
Login: 21
Front Page: 22
Employee: 23
Attendance: 24
Salary: 25
Designation: 26
Database: 27

# Chapter 1


## Abstract:

"Payroll Management System" is designed to make the existing manual system automatic with the help of computerized equipment and full-edged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This application can maintain and view computerized records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client.


## Introduction

This chapter provides an introduction to the project, motivation and criteria to achieve the objectives.

### 1.2. Project Outline

The task is to build a salary management system. Current salary system is manual therefore the organization wants to switch to an automated computerized salary management system. After building this system we have to integrate it with the existing computerized system. The existing system is dealing with the client registrations, keeping records of clients, client billing etc. Hence we can say, employee salary system will be a subpart of the existing computerized system. Employee salary management system is enabling the organization to handle salaries of employees. The managers or team leaders of this organization are able to fill out all necessary information of an employee, i.e., residential address, contact information, designation details, salary and other relevant information. This system should be capable enough to calculate the salaries of employees.By the end of each month. Upon request, the employees can receive their salary slips through email. Moreover, the system should be able to calculate tax deductions of every employee.

## 1.2. Motivation

All calculations such as employee salary, employee tax, organization tax calculations etc. are being done manually at the moment which is a time-consuming task. Hence, a system is required that can perform all above said operations automatically. Moreover, the system should be user friendly, flexible, fast and highly secure.

## 1.3. Problem Identification:

The organization is using a manual system to calculate salaries and tax of employees. They also have to calculate and keep record of tax of whole organization and maintain tax files manually.All this work requires a lot of paper work, is extremely time-consuming job, and accordingly costly as well, as they have to hire more man power. Since there is always a risk of human errors present in a manual system so the chances of errors are very high and to figure out such errors is also a very lengthy procedure. Therefore, the organization decided to switch from a manual system to an automated computerized salary management system. The requirement of the organization is to develop an application that is able to deal with salary and tax calculation of employees within the organization and maintain its data base. Furthermore, this system should be able to generate automatic files such as, salary slips, bank files and tax files in response of queries from the data stored in the database.

## 1.5 Project Goals

The system should be capable of performing following functions:

- Store basic information regarding employees of the organization.
- Store salary information of employees (entered by the Human Resource Manager) such as, Basic Salary, Fuel, salary before tax, tax percentage, total amount of tax paid, salary after tax, social security fee, on monthly basis.
- System should be able to generate the following salary information:

    1. Tax Calculation
    2. Social-Security Calculation
    3. Insurance Calculation
    4. Net Salary Calculation

- Social security fee depends upon employee basic salary. Hence, percentage on the salaries would be variable.
- Salary slips can be sent to the employees upon request.

## 1.6. Goals Criteria

The criteria to achieve the goals of our project is as following:

- To generate bank file/invoice System should be able to generate a bank file and salaries should be delivered to employee accounts.
- Performance and efficiency the system should perform according to the requirements and provide appropriate and accurate results.

# Chapter 2

## Comparison:

In this chapter we will compare a system of our own choice with different salary calculation software available in the market. On the basis of this comparison, we will describe the motivation of developing customized solution

### 2.1 Why Customized Solution:

There are a variety of software tools available in the market that are able to calculate the salaries of employees. Now the question arises, why is it required to build a customized software?

#### 2.1.2 Analysis of Different Salary Calculators:

**Salary Calculation Software:**

This software is able to do calculations of employee salaries on the basis of data provided by the work tracking system. It is basically a combination of two systems, i.e., “work management software” and “Salary calculations”. Second one is totally depending on the first one by taking data as input from work management software and calculating the salary on the basis of this data. The software has the following features:

- “Planning Project”
- “Keeping track of working results and helping managers to define work results”
- “Helps to Calculate salary from work results”

**Analysis of Salary Calculation Software:**

- This software is facilitating work management and task scheduling more than the salary calculation. Salary calculation is one of its secondary feature, whereas work management is its primary feature.
- This software is not providing any feature to calculate taxes and social securities.
- This software is unable to generate pay/salary slips and bank/invoice slips.
- The company has to purchase the license per user to use this software.
- This software is not providing any interfaces or plug-ins that can be used to merge it with the existing system working with-in the organization.
- As with the growth of company, it may require to enhance the system or if company needs any amendments in the system, it would be very hard to achieve such goals as the purchased softwares are delivered always in. Exe form.

## 2.2. Employee Salary Management System:

The System we are going to develop is according to the user requirements which will perform salary calculation as well as tax calculation. Moreover, the system will be user friendly and flexible enough to be enhanced according to the needs of the users in future.

# Chapter 3

## Requirement Analysis

In this chapter, we are going to discuss the requirements, design of the system and analyze them using some appropriate software models.

### 3.1. System Requirements:

Requirement analysis is an important phase of the system development cycle which provides us all the specification of system in detail which are very essential to build the system and also provides us knowledge about the behavior of the system. Collection of system requirements is a very critical point because the whole system is based on this knowledge and it also provides input to the next following stages of the system development life cycle. We can classify the requirements of the system in two categories as following:

- Functional Requirements
- Non Functional Requirements

#### 3.1.1 Functional Requirements:

Functional Requirements specifically defines functionalities of the system, behavior of the system and the goals to achieve it. Functions that describe the behavior of the system are considered as behavior requirements and will be shown in the form of use cases. The importance and description of functional requirements are explained in the following table:

| ID NO. | FR | Importance | Description |
|---|---|---|---|
| 1 | Security | Essential | User authentication is required. |
| 2 | Web Interface | Essential | Provides interaction between user and database. |
| 3 | Database | Essential | Stores the calculated taxes and salaries. |
| 4 | File System | Essential | Stores bank files and tax files. |
| 5 | Search | Required | To search out tax files, salary slip and bank files. |

### 3.1.2 Non-functional Requirements:

Non-functional requirements are also known as quality attributes of a system. Hence, it provides us knowledge regarding the operations instead of behavior or functionalities, contradicting with the FR in this manner. Non-FR are described in the system architecture helping us to achieve the quality goals and improves the functionalities of the system. Non-functional requirements are as following:

| ID NO. | Non FR | Importance | Description |
|---|---|---|---|
| 1 | Performance | Essential | Performance of system should be adequate and fast. |
| 2 | User Friendly Environment | Essential | System interface should be simple and user friendly. |
| 3 | Application Maintenance | Essential | Maintenance and documentation should be done throughly. |
| 4 | Paging | Essential | System should be capable of splitting large amount of data using data paging to present information in a user friendly way. |
| 5 | Application Scalability | Required | System should be flexible and expendable for future use. |
| 6 | Platform Independence | Required | System should be capable to work in any environment. |

# Chapter 4

## Case Study for Payroll Management System:

Web Matrix, a software company is having employees and calculation of salaries and record of attendance is getting complicated for the company so company wants an automated system which can perform in this domain. Company's Human Resource Manager is the only person to use this system.

There are many employees working in the company who can be identified by their names, emails, addresses and contact information but mainly are identified by their employee id which is unique for every employee. Two major types of employees are working in the company. Contractual employees have signed a contract valid for different time periods and are not specified in any single domain while Permanent employees are specialized in single domains.

All the employees are majorly working in three designations. Programmers charge according to their lines of code, Designers charge according to the design and Testers charge per validation.

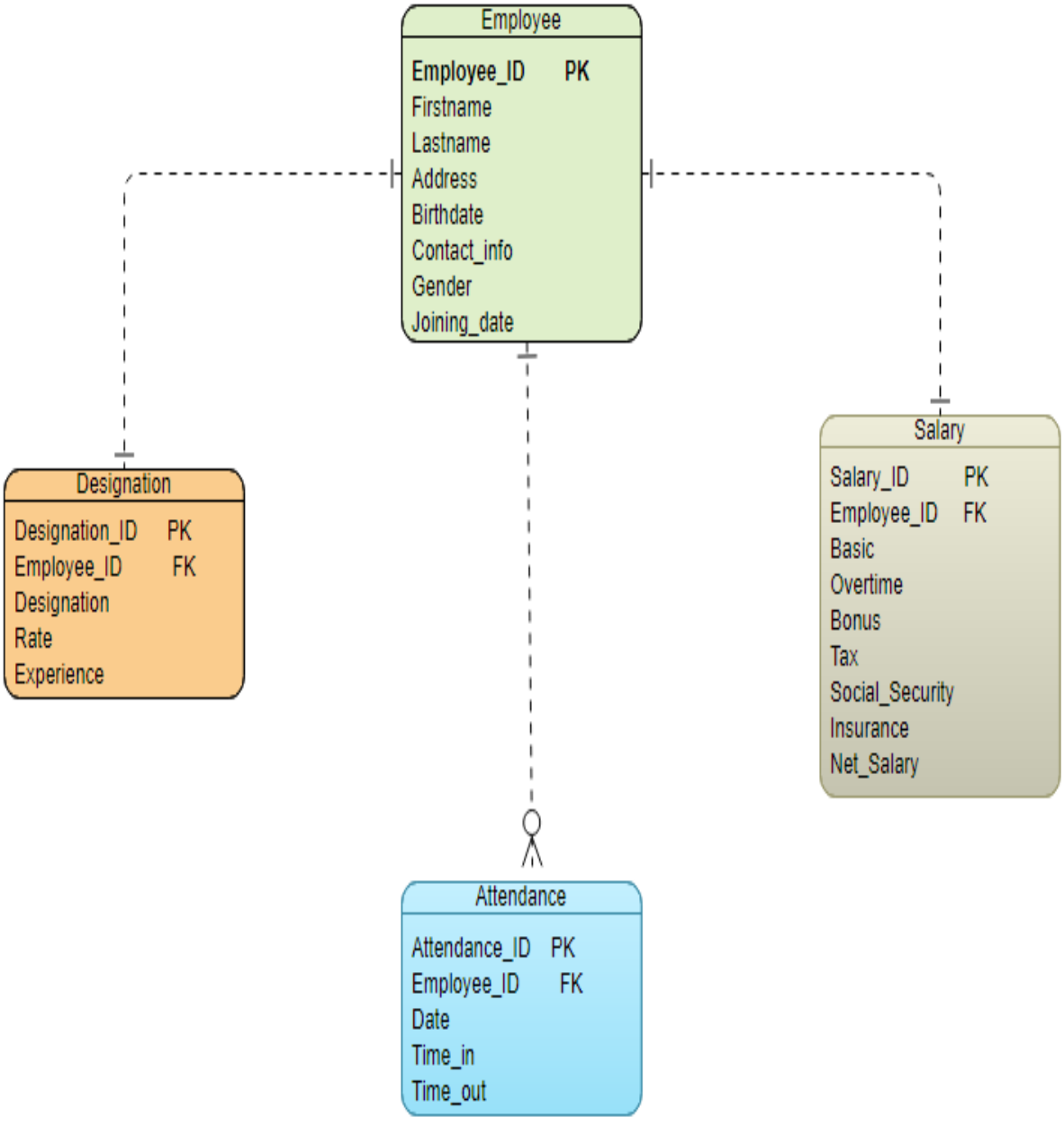
Track of entrance and exit is kept by the company for each employee and daily attendance is also monitored by HRM.

17 percent of tax is deducted from basic salary from every employee, furthermore 5 percent Insurance deduction and 7 percent social_security deduction is also done by the company for each employee. Apart from basic salary, company also pays for bonus based upon achievements and overtime plus fuel expense.

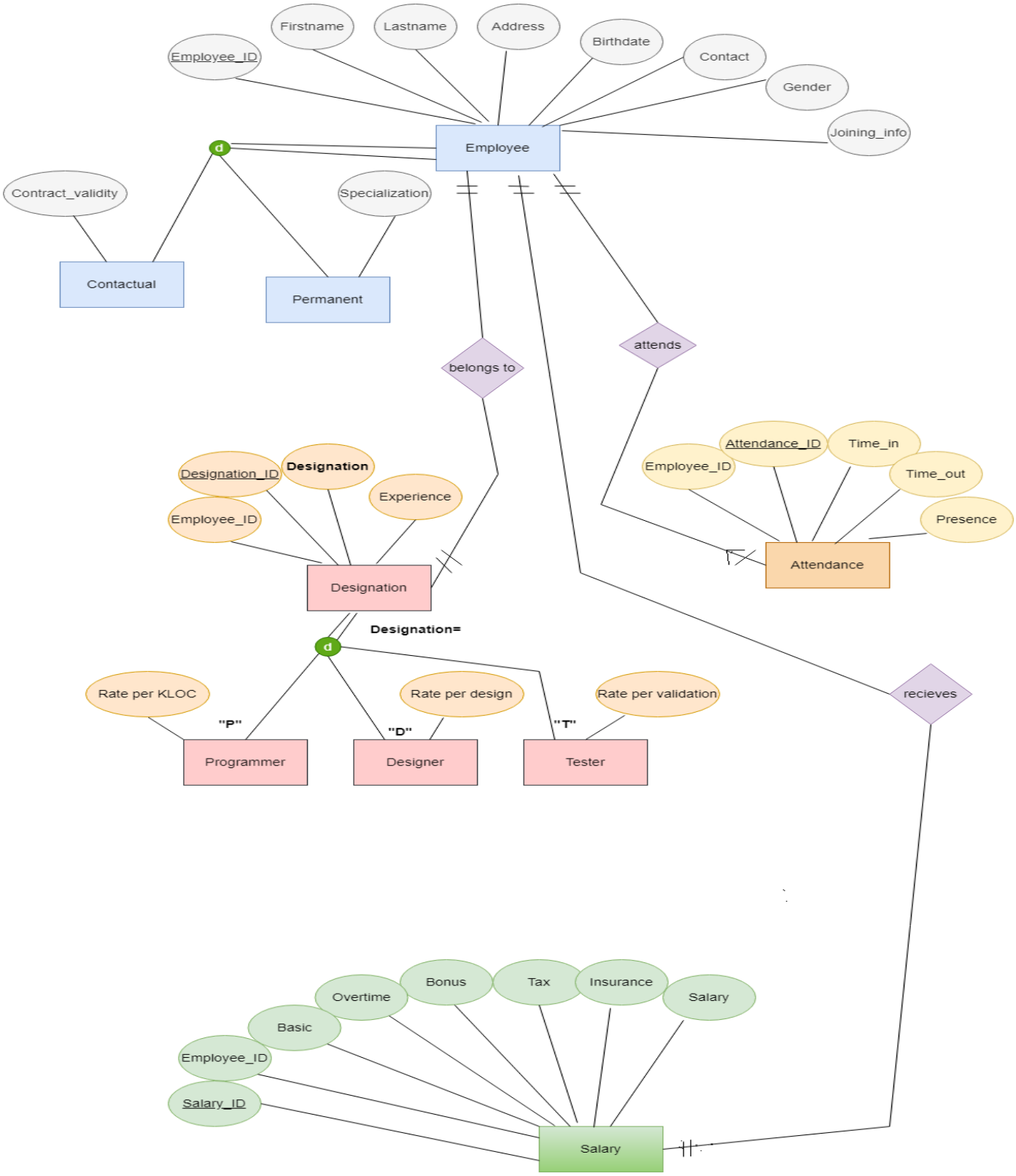
## Business Rules:

| | | | | |
|---|---|---|---|---|
| **Employee** | **must** | **have** | **one** | **designation** |
| **Salary** | **may** | **be drawn by** | **many** | **employees** |
| **Employee** | **may** | **have** | **many** | **specializations** |
| **Attendance** | **may** | **be marked by** | **at most one** | **Employee** |

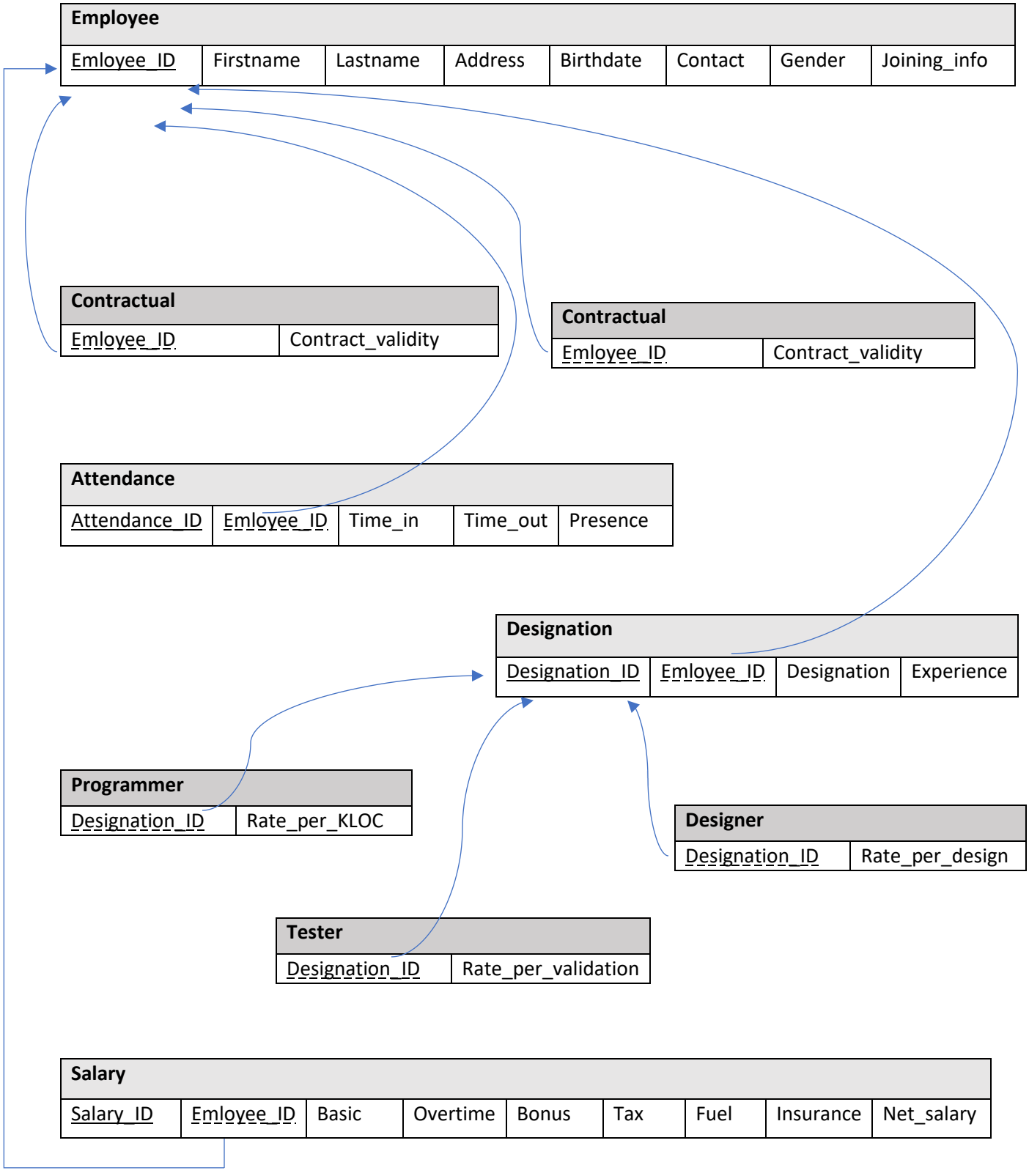
## Entity Relationship Diagram:



## Enhanced Entity Relationship Diagram:



## Relational Mapping from EERD:



## Payroll Database Creation

### Employee Table:

#### DDL:

```
create database payroll;

use payroll;

CREATE TABLE `employee` (


employee_id varchar(15) NOT NULL primary key,

firstname varchar(50) NOT NULL,

lastname varchar(50) NOT NULL,

address text NOT NULL,

birthdate date NOT NULL,

contact_info varchar(100) NOT NULL,

gender varchar(10) NOT NULL,

joining_date date NOT NULL

);
```

#### DML:

```
Insert into
employee(employee_id,firstname,lastname,address,birthdate,contact_info,gender,j
oining_date)

Values


("12290","Sersei","Lannister","Islamabad","1975-10-
05","03002587412","Male","2021-05-10");
```

**Note: Insertion & Virtual Insertion is done through system built**

## Attendance Table:

### DDL:

```
CREATE TABLE `attendance` (

  employee_id varchar(15),
  date date NOT NULL,
  time_in varchar(25) NOT NULL,
  time_out varchar(25) NOT NULL,
  presence varchar(25),

    CONSTRAINT FK_employee FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);
alter TABLE payroll.attendance
ADD
attendance_id varchar (25)UNIQUE KEY NOT Null;
```

### DML:

```
Insert into attendance(employee_id,date,time_in,time_out,presence)
Values ();
```

**Note: Insertion & Virtual Insertion is done through system built**

## Salary Table:

### DDL:

```
use payroll;
create table salary(
employee_id varchar(15) Not Null,
basic int (25) Not Null,
overtime int (25) Not Null,
bonus int (25) Not Null,
fuel int(25) Not Null,
gst int(25) AS (basic * 17/100),
social_security int(25) AS (basic * 5/100),
insurance int (25) AS (basic * 7/100),
net_salary int (35) AS (basic+overtime+bonus+fuel-gst-social_security-insurance),
CONSTRAINT FK_salary FOREIGN KEY (employee_id)
   REFERENCES employee(employee_id)
);

ALTER TABLE salary ADD sal_id varchar(25) NOT Null;
```

### DML:

```
INSERT INTO salary (employee_id,basic,overtime,bonus,fuel)
VALUES
(12290,90000,5000,10000,7000);
```

**Note: Insertion & Virtual Insertion is done through system built**

## Designation Table

### DDL:

```
use payroll;
create table designation(
employee_id varchar(25),
designation varchar(25),
rate varchar(25),
experience varchar (25),
  CONSTRAINT FK_designation FOREIGN KEY (employee_id)
    REFERENCES employee(employee_id)
);
ALTER TABLE designation
ADD desig_id varchar(25) NOT Null;
```

### DML:

```
insert into designation(employee_id,designation,rate,experience)
VALUES
();
```

**Note: Insertion & Virtual Insertion is done through system built**

## DQL:

**Fetch record of those employee whose joining date is in 2015 regardless of month & day.**

```
Select *
From payroll.employee
Where joining_date like '%2015%'
ORDER BY  joining_date ASC;
```

**Fetch record of those employee whose Basic Salary is between 50k and 100k in reverse order.**

```
Select *
From payroll.salary
Where basic Between 50000 And 100000
Order by net_salary DESC;
```

**Generate Minimum and Maximum Salary from the given record, also generate average tax from overall data and group the output by total salary only for those employees whose insurance is greater than 5000.**

```
SELECT min(net_salary),max(net_salary), avg(gst)"Average Tax",
count(employee_id)
from salary
Where insurance > 5000
GROUP BY net_salary;
```

## Data Normalization:

In order to maintain data consistency and to avoid data redundancy, following approach can be done to normalize data.

***designation***

| ***employee_id*** | **desig_id** | **designation** | **rate** | **experience** |
|---|---|---|---|---|
| *12250* | D01 | Programmer | 25000 | 5 years |
| *12251* | D02 | Designer | 50000 | 10 years |
| *12252* | D03 | Tester | 30000 | 12 years |
| *12253* | D01 | Programmer | 25000 | 7 years |

As rate and designation is independent of employee_id and fully dependent on desig_id so we'll create another relation.

***Designation***

| ***employee_id*** | **experience** | **desig_id** |
|---|---|---|
| *12250* | 5 years | D01 |
| *12251* | 10 years | D02 |
| *12252* | 12 years | D03 |
| *12253* | 7 years | D01 |

***Assoc***

| ***desig_id*** | **designation** |
|---|---|
| *D01* | Programmer |
| *D02* | Designer |
| *D03* | Tester |

In this way data will be almost normalized.

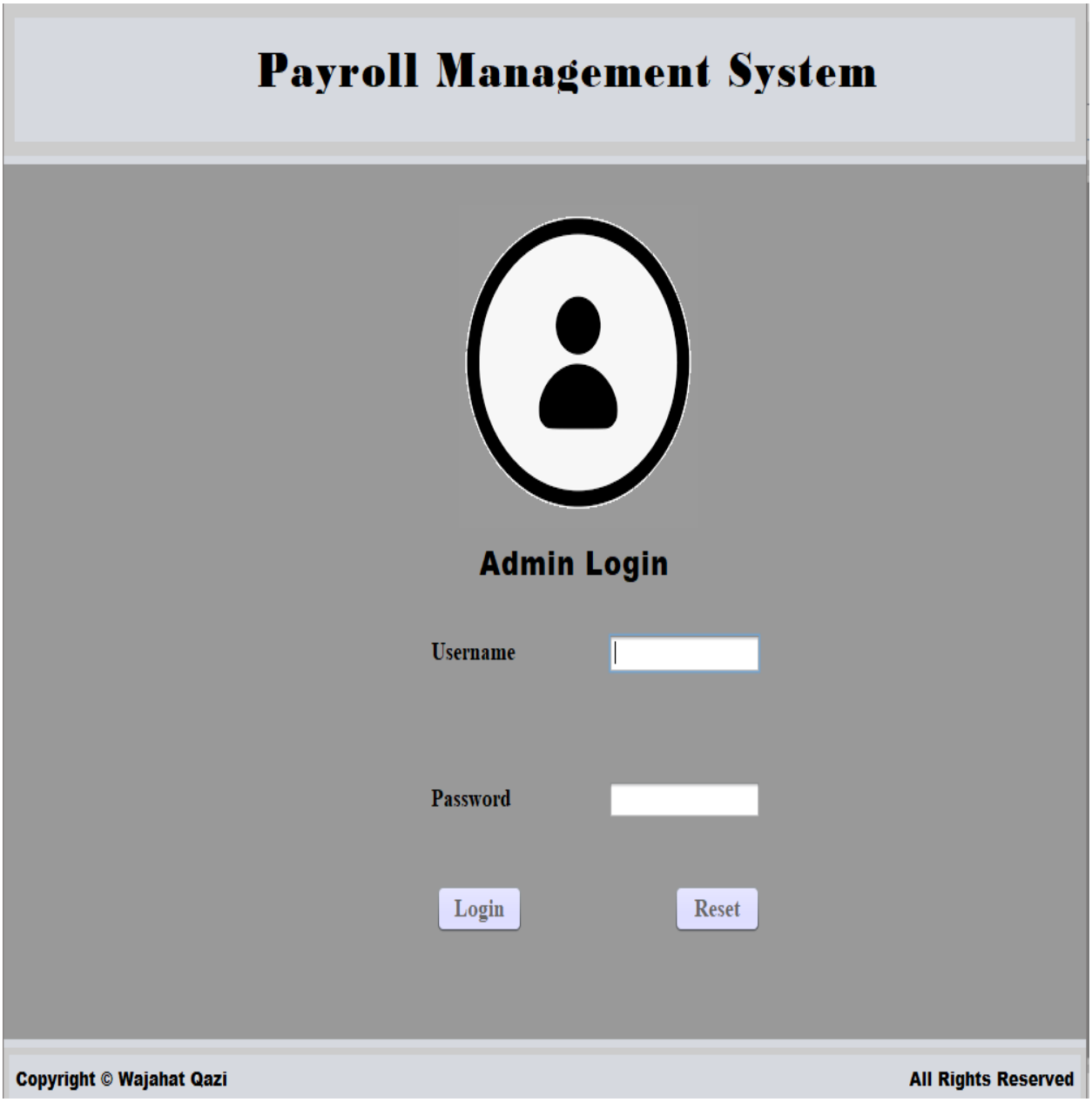
desig_id is Primary Key in Assoc table and is acted as foreign key in designation table.

## Connecting Mysql with Java:

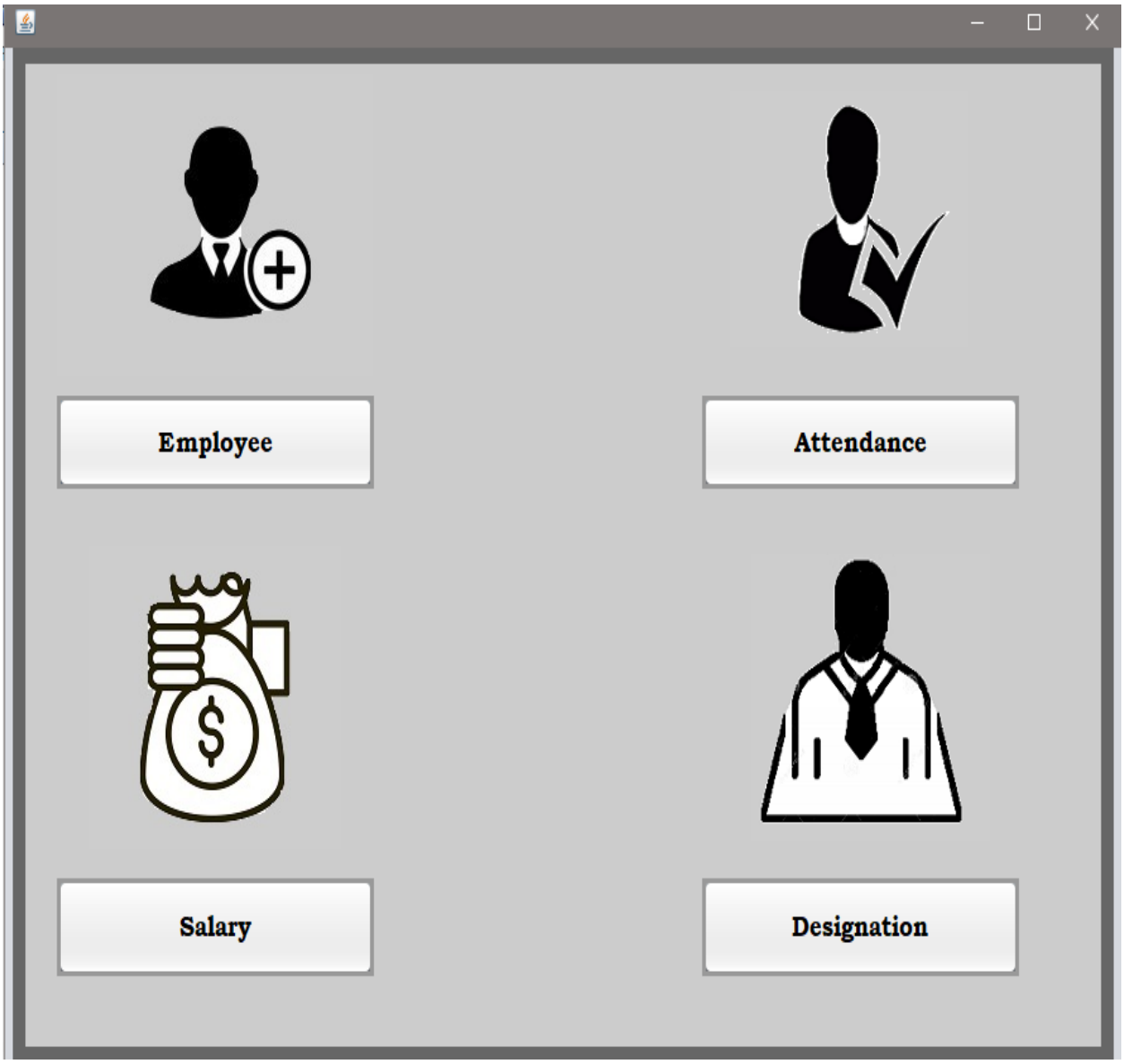
```
 try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3307/payroll","root","");
       String sql= "Insert into employee values(?, ?, ?, ?, ?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id.getText());
        pstmt.setString(2, fname.getText());
        pstmt.setString(3, lname.getText());
        pstmt.setString(4, address.getText());
        pstmt.setString(5, birth.getText());
        pstmt.setString(6, contact.getText());
       String dept="0";
     if(male.isSelected())
        {
        dept = male.getText();  }
     if(female.isSelected())
     {
        dept = female.getText();  }
        pstmt.setString(7, dept);
        pstmt.setString(8, jdate.getText());
        pstmt.executeUpdate();
        JOptionPane.showMessageDialog(null, "Data Entered Successfully");
        conn.close();
```

## Graphical User Interfaces:

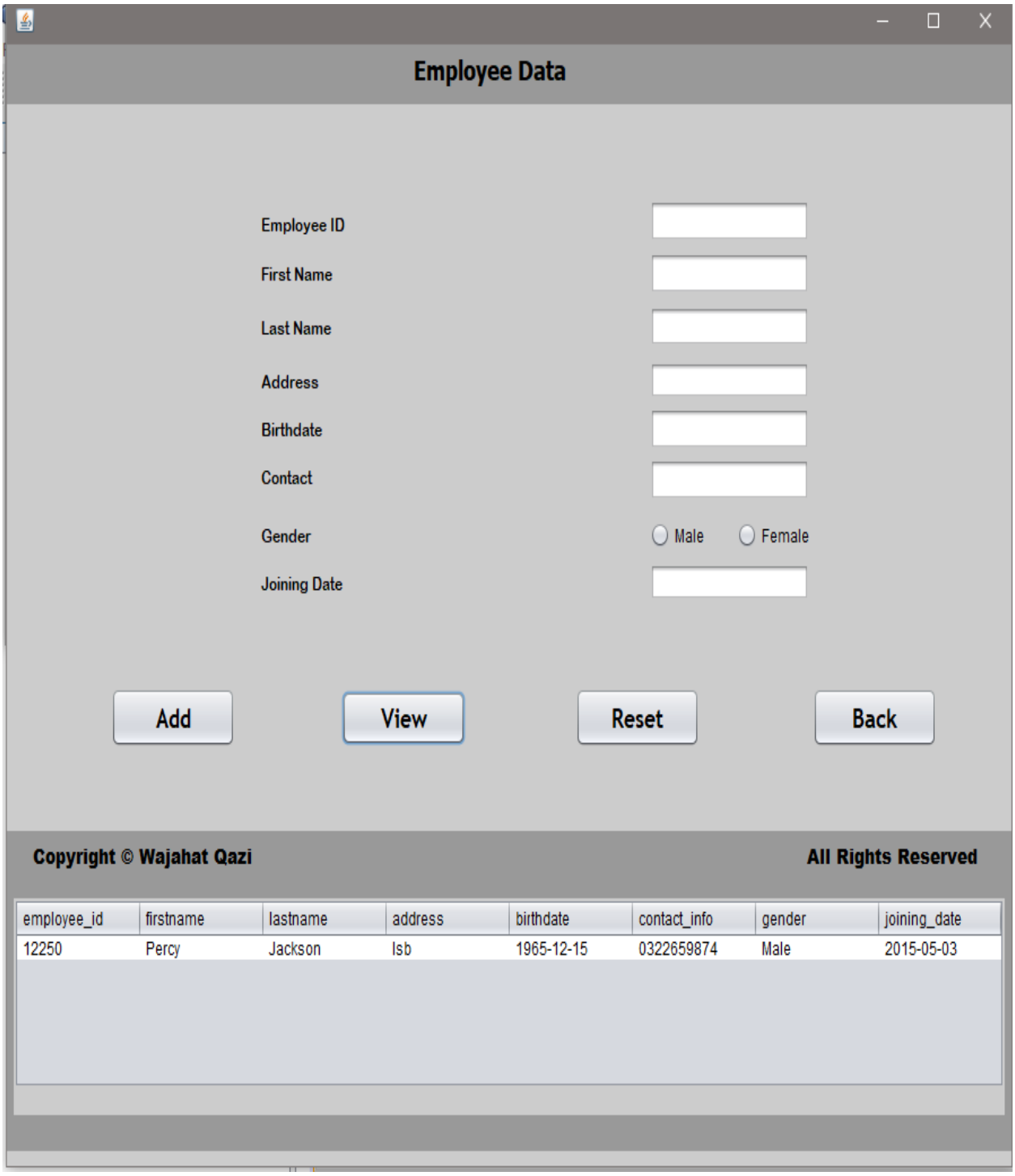
### Login:



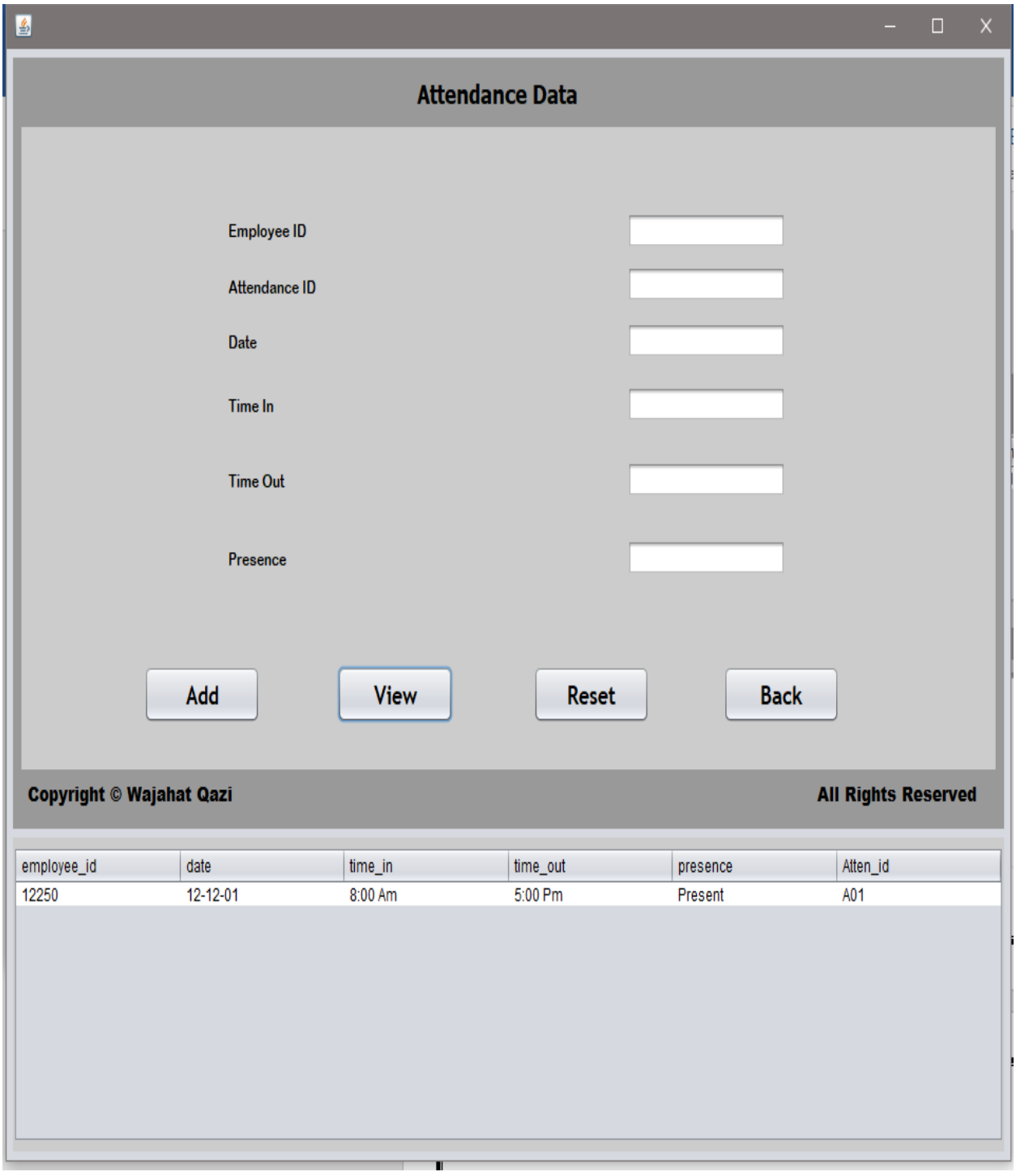
Front Page:



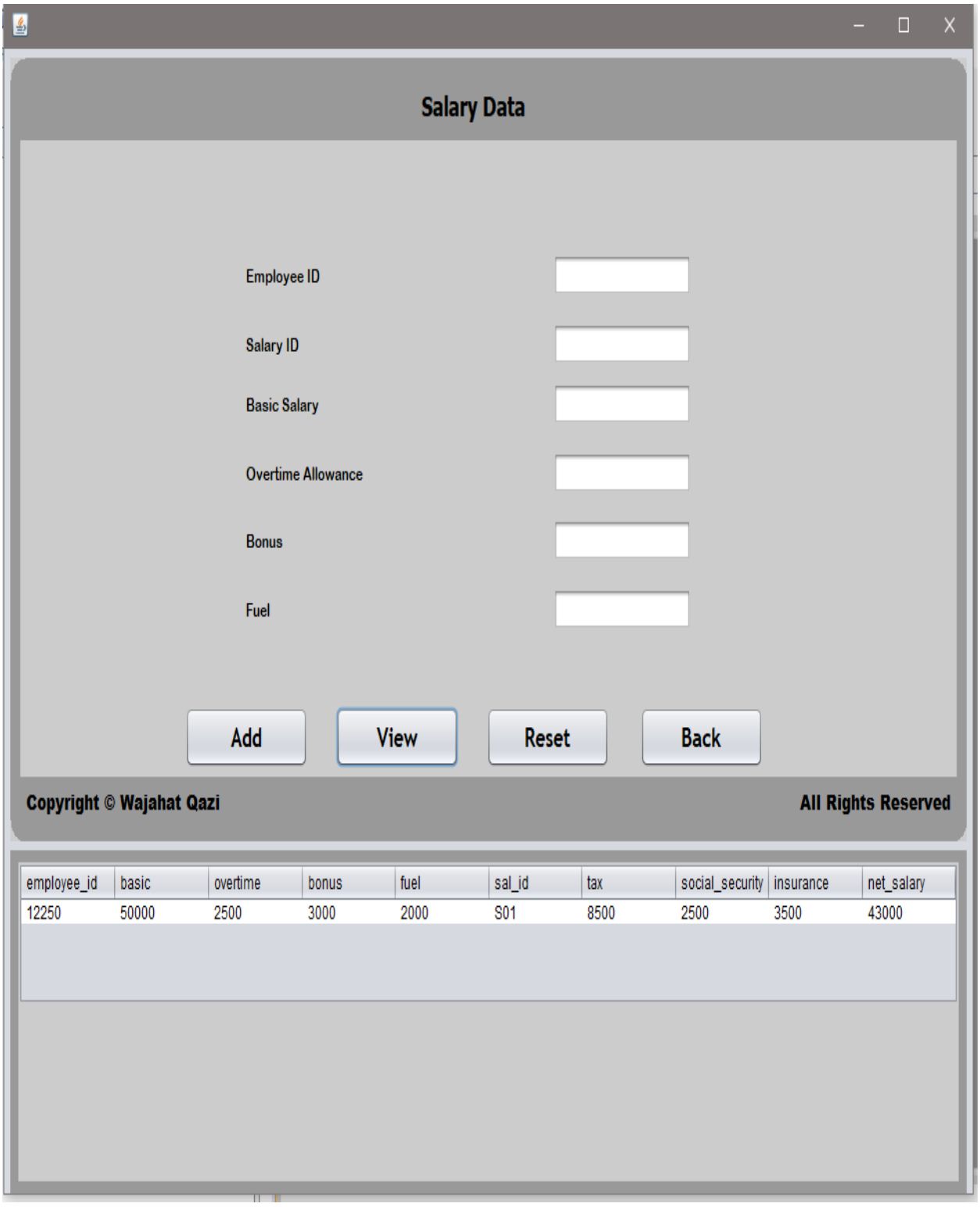
## Employee:


Employee Data
Employee ID
First Name
Last Name
Address
Birthdate
Contact
Gender
Male
Female
Joining Date
Add
View
Reset
Back
Copyright © Wajahat Qazi
All Rights Reserved
employee_id
firstname
lastname
address
birthdate
contact_info
gender
joining_date
12250
Percy
Jackson
lsb
1965-12-15
0322659874
Male
2015-05-03

## Attendance:

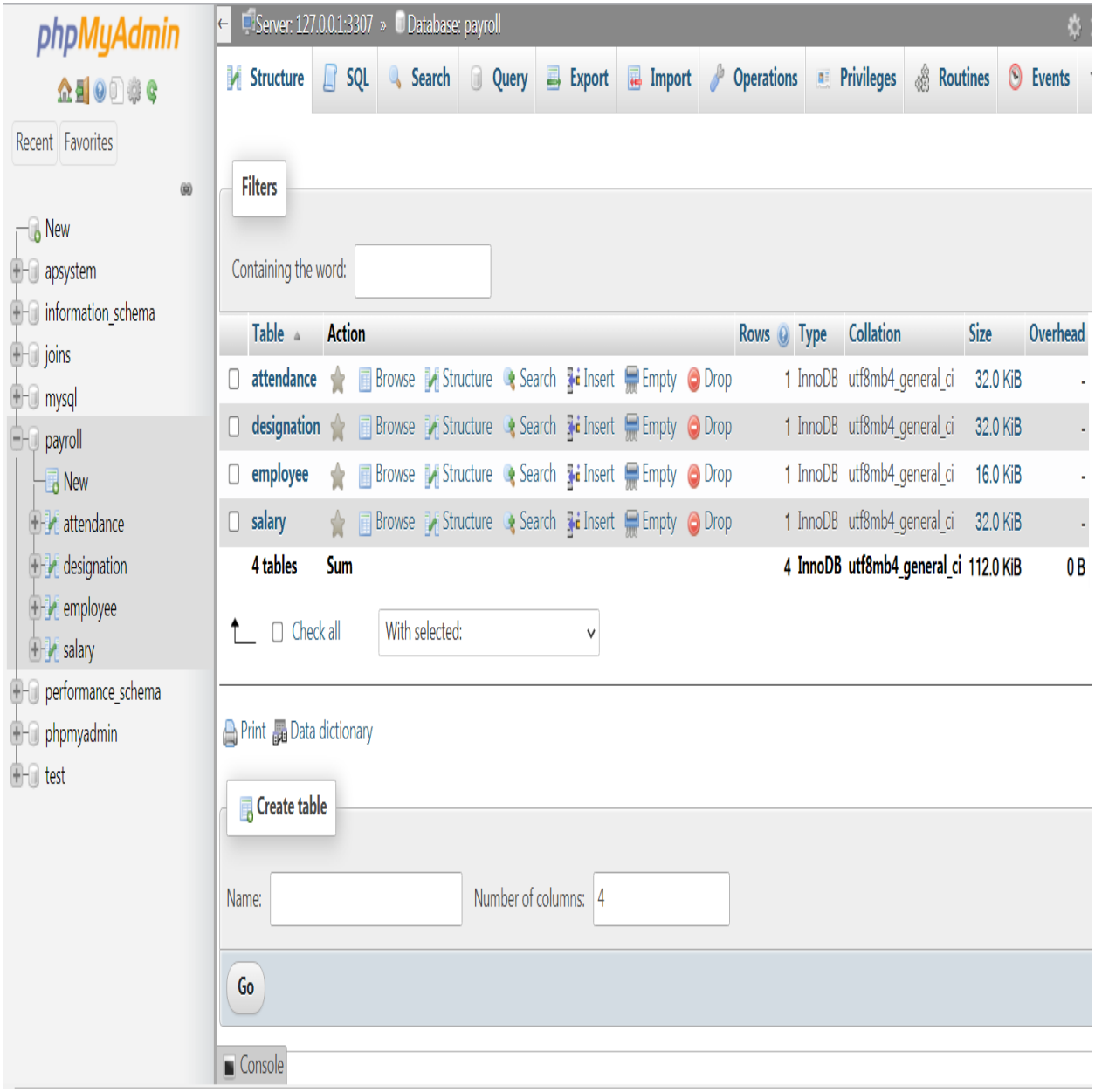

Attendance Data
Employee ID
Attendance ID
Date
Time In
Time Out
Presence
Add
View
Reset
Back
Copyright © Wajahat Qazi
All Rights Reserved
employee_id
date
time_in
time_out
presence
Atten_id
12250
12-12-01
8:00 Am
5:00 Pm
Present
A01

## Salary:


Salary Data
Employee ID
Salary ID
Basic Salary
Overtime Allowance
Bonus
Fuel
Add
View
Reset
Back
Copyright © Wajahat Qazi
All Rights Reserved
employee_id basic overtime bonus fuel sal_id tax social_security insurance net_salary
12250 50000 2500 3000 2000 S01 8500 2500 3500 43000

## Designation:

## Database:


phpMyAdmin
Recent
Favorites
New
apsystem
information_schema
joins
mysql
payroll
New
attendance
designation
employee
salary
performance_schema
phpmyadmin
test
Server: 127.0.0.1:3307 » Database: payroll
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Filters
Containing the word:
Table
Action
Rows
Type
Collation
Size
Overhead
attendance Browse Structure Search Insert Empty Drop 1 InnoDB utf8mb4_general_ci 32.0 KiB -
designation Browse Structure Search Insert Empty Drop 1 InnoDB utf8mb4_general_ci 32.0 KiB -
employee Browse Structure Search Insert Empty Drop 1 InnoDB utf8mb4_general_ci 16.0 KiB -
salary Browse Structure Search Insert Empty Drop 1 InnoDB utf8mb4_general_ci 32.0 KiB -
4 tables Sum 4 InnoDB utf8mb4_general_ci 112.0 KiB 0 B
Check all
With selected:
Print
Data dictionary
Create table
Name:
Number of columns: 4
Go
Console