

A  
PROJECT REPORT  
ON  
**Payroll Management System**

**Subject**  
System Design Practice

**Bachelor of Technology**  
**In**  
**Computer Engineering**

Submitted by

**Lathiya Parth (CE 54)**  
**Jagwani Abhishek (CE 41)**

Under the Guidance of  
Prof. Mrudang.T.Mehta  
Associate Professor



**DEPARTMENT OF COMPUTER ENGINEERING**  
**FACULTY OF TECHNOLOGY,**  
**DHARMSINH DESAI UNIVERSITY, NADIAD**

**DHARMSINH DESAI UNIVERSITY**  
**NADIAD-387001, GUJARAT**



**CERTIFICATE**

**This is to certify that the project carried out in the subject of Software Design Practices, Semester VI entitled “Payroll Management System” and recorded in this report is a work of**

<b>1) Lathiya Parth.</b>	<b>ROLL NO: CE-54</b>	<b>ID:12CEUOS130</b>
<b>2) Jagwani Abhishek.</b>	<b>ROLL NO: CE-41</b>	<b>ID:12CEUOS033</b>

**Of Department of Computer Engineering. They were involved in Project work during academic year 2014 -2015.**

**Prof. Mrudang T. Mehta**  
**Associate Professor,**  
**Dept. of Computer Engineering,**  
**Faculty of Technology,**  
**Dharmsinh Desai University**

**Prof. C.K. Bhensdadia**  
**Head,**  
**Dept. of Computer Engineering,**  
**Faculty of Technology,**  
**Dharmsinh Desai University**

## ACKNOWLEDGEMENT

With immense pleasure and commitment we would like to present the project assignment. The nature of project on the development of **Payroll Management System** has given us wide opportunity to think, implement and interact with various aspects of management skills as well as the new emerging facilities and the technology used in architecture and the enhancements given to the students with a boon of spirituality and curricular activities.

Every work that one completes successfully stands on the constants encouragement, good will and support of the people around. We hereby avail this opportunity to express our gratitude to number of people who extended their valuable time, full support and cooperation in developing this project.

We express deep sense of gratitude towards our project guide Prof. Mrudang T. Mehta towards their innovative suggestions and efforts to make project a success. It is their sincerity that prompted us throughout the project to do hard work using the industry adopted technologies.

We are sincerely thankful to Head of CE department, Prof. C.K. BHENSDADIA for the unconditional and an unbiased support during the whole session of study and development.

They altogether provide us favorable environment, without them we would not have achieved our goal.

Thanks and regards,  
Lathiya Parth  
Jagwani Abhishek

## **ABSTRACT**

Payroll Management System (PMS) is a system which is used to maintain records of all the users working in an organization, providing an easy GUI to the user. This System is per organization. At the front end of the system we have used NetBeans IDE in JAVA technology and WAMP server for maintaining databases. The project proceeds through a sequence of well designed JSP pages with validation to security, consistency, reliability, etc. Various users are provided which can perform various operations and tasks for which they have been authorized respectively. Basically there are 4 types of users: Employee, Accountant, HR Manager and Admin. Admin is fully authorized. While other users are authorized upto a certain level. Based on the user authorisation, users can perform various operations like Adding information of new Employee, Updating information, View list of employees, generate pay slip of particular employee, Generate report of the organization. This system is very user friendly and easily maintains all the information for all kind off users without any difficulties. The system also generates fault and error messages in case of invalid operation or the user enters invalid or wrong information for some particular cases. Various UML diagrams are prepared which may help a user to understand the flow of any module/task easily. It contains 6 various UML diagrams. Individual modules of the system are tested later and test cases are generated for that. Later all the modules are integrated and then whole system testing is performed to ensure that the system is error free and secured.

## TABLE OF CONTENT

	Abstract.....	iv
1.0	Introduction.....	1
	1.1 Project details and specifications.....	1
	1.2 Purpose.....	2
	1.3 Scope.....	2
	1.4 Product Perspective.....	2
	1.5 Nomenclature.....	3
	1.6 Interfaces.....	3
	1.7 Memory.....	5
	1.8 Site Adaption Requirements.....	5
	1.9 Constraints.....	5
	1.10 Assumptions and Dependencies.....	6
2.0	Requirements.....	7
	2.1 Functional Requirements.....	7
	2.2 Non Functional Requirements.....	10
3.0	System Analysis and Design.....	13
	3.1 Use case Diagram.....	14
	3.2 Class Diagram.....	15
	3.3 Sequence Diagram.....	16
	3.4 Activity Diagram.....	18
	3.5 E-R Diagram.....	19
	3.6 Data Dictionary.....	20
4.0	System Implementation.....	22
	4.1 Modules.....	22
	4.2 Technologies used for Project Implementation.....	27
5.0	Testing.....	28
	5.1 Testing Plan.....	28
	5.2 Testing Strategy.....	28
	5.3 Test Cases.....	29

6.0	Limitations, Conclusion and Future Extensions.....	33
6.1	Limitations.....	33
6.2	Conclusion.....	33
6.3	Future Extensions.....	33
	Bibliography.....	34

# Chapter 1 Introduction

## 1.1 Project Details & Specification

Payroll Management System is the system that maintains all the records of all employees and accountants of an organisation. This system is developed to ease the process of searching and maintaining the valuable records of all type of users (i.e employees and accountants).

### Users of payroll Management System:-

#### 1) Employee

Employee is a user that can view and update its details, can view his own pay slip and can apply for leave.

#### 2) Accountant

Accountant is a user who manages all the information about each employee and can generate pay slips and reports.

#### 3) Admin

Admin is a user who manages information about all the employees as well as about all the accountants.

#### 4) HR Manager

HR Manager is a user who can only add information of new employees via interviews.

## **1.2 Purpose**

The purpose of this SRS is to outline both functional and non-functional requirements of the subject Payroll Management System(PMS). In addition to said requirements, this document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Consequently, the document should act as a foundation for efficient and well-managed project completion and further serve as an accurate reference in the future. It will not only provide an extensive capacity for project planning and progress assessment but it will further assist with developer interactions.

## **1.3 Scope**

We describe what features are in the scope of the software to be developed.

- To view and manage information of all currently working employees and accountants in the organisation.
- To add and update details of all the employees.
- To view and generate pay slip of each employee.
- To generate report of organisation.

## **1.4 Product Perspective**

PMS, a Payroll Management System that is easy to understand, easy to use and offers the simplicity of managing various records of an organisation. This powerful software program is specifically designed for maintaining overall information of an organisation. This intuitive visual interface makes various aspects of performing various operations related with all the kind of users. PMS controls all back-end functionalities like generating pay slips, reports, tax calculations, leave management, etc.



## 1.5 Nomenclature

This subsection presents definition for the terms and acronyms used throughout this SRS as they relate to the subject PMS.

Term	Description
Record	Information about various user
Employee	Developer in an organisation
Accountant	High level authority who manages information
Admin	Highest authority user
Pay slip	Salary information of employee
Report	Performance of organization

**Table 1.5.1 System Terminology**

Acronym	Description
SRS	Software Requirements Specification
PMS	Payroll Management System

**Table 1.5.2 System Acronyms**

## 1.6 Interfaces

The PMS interacts with the user via GUI. This system should be easily operable so that it can return information to the user what is required per their basis. System must generate error messages in case of failure.

## **Software interfaces**

- A computer must be running on Windows XP or above.
- The application is optimized for screen resolution of 1024x768 and above with minimum 16 bit colour.
- PMS interfaces with NetBeans IDE(version 7.4 or above).
- WAMP Server must be installed for MYSQL or ACCESS Database Server.

## **Hardware Interfaces**

- System with Windows XP or higher version.
- 1.8 Ghz processor, 256MB RAM, 80 GB HDD.
- Optical mouse, keyboard, Serial and parallel port.
- Suitable Printer.

## **Communication Interfaces**

Employee of the system is provided the facility of viewing pay slip, in order to communicate with accountant or admin. It communicates with the system also in case of applying leaves. No other means of communication.

## **System Interfaces**

- Interfaces : NetBeans IDE 7.4
- WAMPServer64 2.5
- MS ACCESS or SQL database server
- Drawing tool : Microsoft Visio Office 2013

## **1.7 Memory**

The memory usage of the PMS will have to be constrained by the devices it is intended to run on. In particular, the wireless tablets, as mobile devices, may have limited memory - this should be taken into account when writing the tablet software. Memory constraints upon the server is not likely to be an issue as each will likely have at least a gigabyte of primary memory and hundreds of gigabytes or more of secondary memory.

- RAM : 256 MB
- Hard Disk : 80 GB

## **1.8 Site Adaption Requirements**

User of this system must have following prior requirements:

- A working computer system.
- An Internet Browser installed on that system.
- A PDF Viewer to view the pay slips.

## **1.9 Constraints**

The PMS should be written in an object-oriented language with strong GUI. The primary candidate tool chains is Java/JSP Pages. The system must provide a capacity for parallel operation and system design should not introduce scalability issues with regard to the number of systems connected at any one time. The end system should also allow for seamless recovery, without data loss, from individual system failure. There must be a strong audit chain with all system actions logged. With that in mind, the most adaptable and portable technologies should be used for the implementation. The system has criticality insofar as it is a live system. If the system is down, then users must not notice, or notice that the system recovers quickly (seconds). The system must be reliable enough to run crash and glitch free more or less indefinitely, or facilitate error recovery strong enough such that glitches are never revealed to its end-users

The general descriptions of any other application that will limit developers' options are:

- This system must be available to the user during his/her working time.
- User of the system must be registered to the system.
- User must be aware of all the functionalities of the system.
- Regulatory policies- Copyright will be as per the system.
- All Rights Reserved. Except as permitted under Indian Jurisdiction copyright act.
- No Part of this Software may be reproduced or distributed in any forms or by any means, without the prior written permission of the developing organization.
- All the other issues/disputes regarding the terms and conditions shall be liable to the Indian Jurisdiction
- The system is 100% reliable.
- Improper knowledge of the user about the system leads to the criticality of the system.
- Some ancillaries including the documentation except the user manual will not be available to the user until a prior execution of the application.
- Backup and Recovery system is available for safety and security.

## **1.10 Assumptions and Dependencies**

Since this is a very secure system, security is the main concern. So SRS assumes that this system is registered as a secured system. The SRS assumes that none of the constituent system components will be implemented as standalone program/application deployment and not require customized embedded firmware to be written.

## Chapter 2 Requirements

The following section presents the complete set of functional and performance requirements identified for the subject PMS.

### 2.1 Functional Requirements

This sub section presents the identified requirements for the subject PMS. Initially, general requirements that pertain to the whole system are given. Where possible, subsequent requirements have been demarcated based on their relevance to the users of the system.

#### R1 LOGIN

**Description:** System allows users to get authenticated for the registered users.

##### R1.1 Login based on the type of user

Input : User provides user name,password and type of user.  
Output : System allows access based on the type of user.

##### R1.2 Login for admin

Input : User provides user name and password.  
Output : System allows access to user admin.

## **R2 MANAGE USER INFORMATION**

**Description:** System manages every information related with each user.

### **R2.1 Manage Employee Information**

#### **R2.1.1 Add information of new employee.**

Input : User provides employee details.

Output : System provides confirmation message.

#### **R2.1.2 Update employee information**

Input : User provides employee details which are to be updated.

Output : System provides confirmation message.

#### **R2.1.3 Manage and generate employee pay slip**

Input : User provides employee id along with name of month and year.

Output : System generates the pay slip of the employee.

#### **R2.1.4 Manage employee leave information and salary deductions**

Input : User provides number of leaves, reason and dates.

Output : System grants the leave and updates the salary and leaves of the employee.

#### **R2.1.5 View employee list**

Input : User clicks on corresponding option.

Output : System provides the list of the employees.

### **R2.2 Manage Accountant information**

Input : Admin provides information of accountant.

Output : System provides confirmation message about adding new accountant.

### **R3 REPORT GENERATION**

**Description :** System generates report of the organisation.

Input : User provides name of the month and year.

Output : System generates a report of the organisation based on input details.

## **2.2 Non Functional Requirements**

### **Performance**

- The response time of the PMS should be less than 5 seconds most of the time in any kind of operation performed.
- The system should be loaded very fast initially.
- The system should display confirmation messages in very quick time.

### **Safety Requirements**

No Safety Requirements have been identified.

### **Security Requirements**

- It must be ensured that access will be provided to the authorized users only.
- Accountant user may only be allowed to perform some of the functions and rest of the functions to be performed with the permissions of the admin.
- The system ensures that any change in the information triggered from any user follows a proper hierarchy of verifications and approval before getting reflected in the system.
- The users must be required to log into PMS system to perform all operations.
- The system will handle role based security.

### **Quality Issues**

- System must be updated accurately.
- Data should be locked up and should be secure.
- Different checkpoints should be made at regular intervals to recover from system crashes.
- There should be an immediate restart after the system crash and there should not be any time delay in the restart of the system.
- It is not necessary for system to be portable. It works as a single standalone application.



## **Software System Attributes**

### **Documentation**

**Software Project Management Plan (SPMP):** SPMP provides brief introduction about the project, its requirements, process involved, resource utilized and task between the working team.

**Software Requirements Specification (SRS):** It gives detail introduction of project with specific requirements, process flow involved in the project. It also states the data involved in it and detailed study of pros and cons involved in it. All the organization members are addressed in through the documentation.

### **Error Handling and Extreme Conditions**

- Input errors should be responding by the message box.
- The system should be restarted in order to overcome the extreme condition.
- Automatically backup the files.

### **Reusability**

The PMS has been designed keeping in mind the system as a utility. Hence it can accommodate and extend its function to several organisations with only some change of code.

### **Security**

The System provides well formed security to all the authorized and unauthorized users. It displays error messages in case of failure.

### **Reliability**

The user inputs must be valid and within the given range. Also the program should terminate normally.

### **Maintainability**

The administrator should have a technical knowledge about maintaining software and further enhancements will be undertaken by the developer.

## **Additional Comments**

Whenever a new SRS is contemplated, then changes should be made accordingly. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.

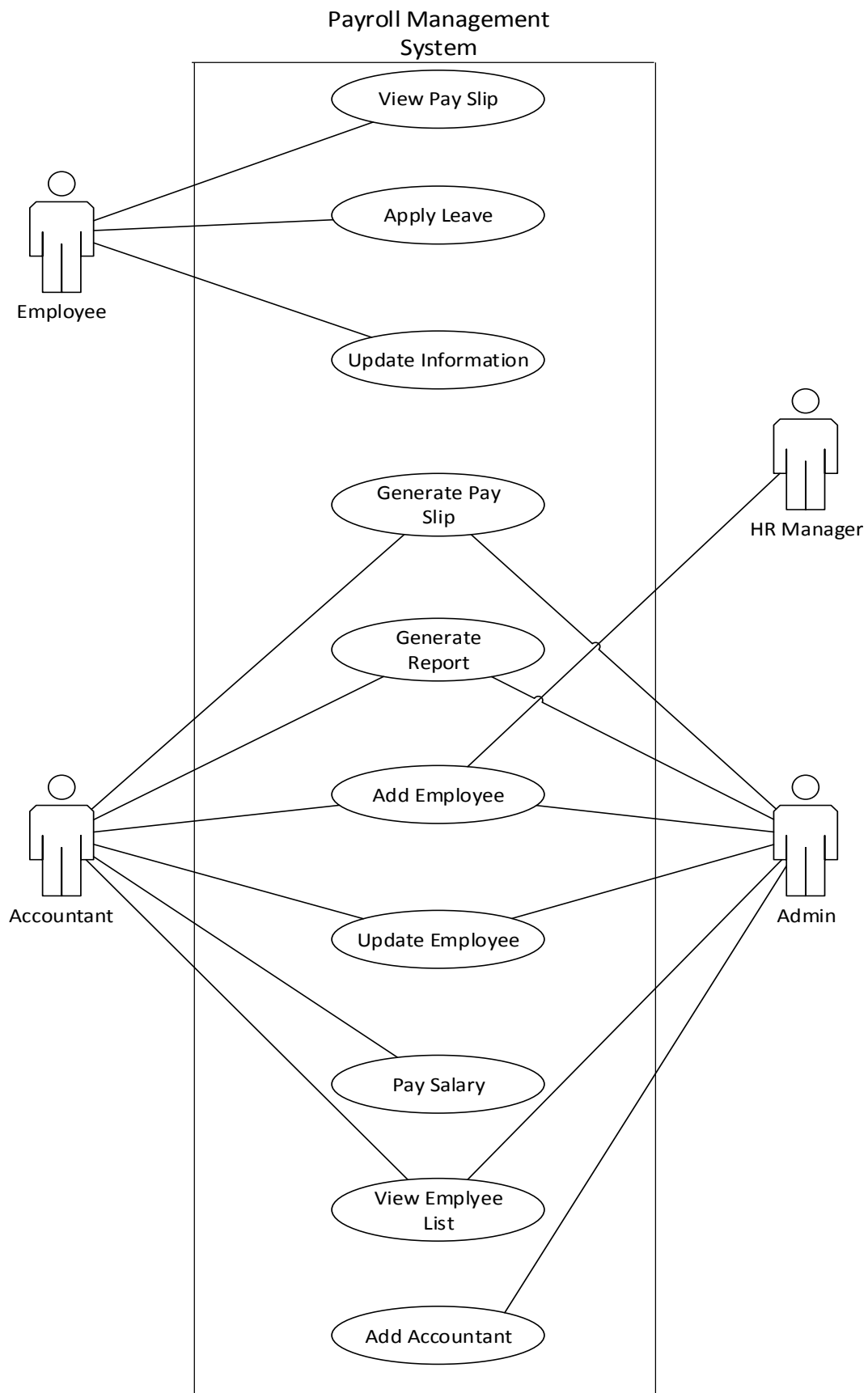
## **Chapter 3 System Analysis and Design**

The Unified Modeling Language (UML) is a general purpose modeling in the field of Software Engineering, which is designed to provide a standard way to visualize the design of a system.

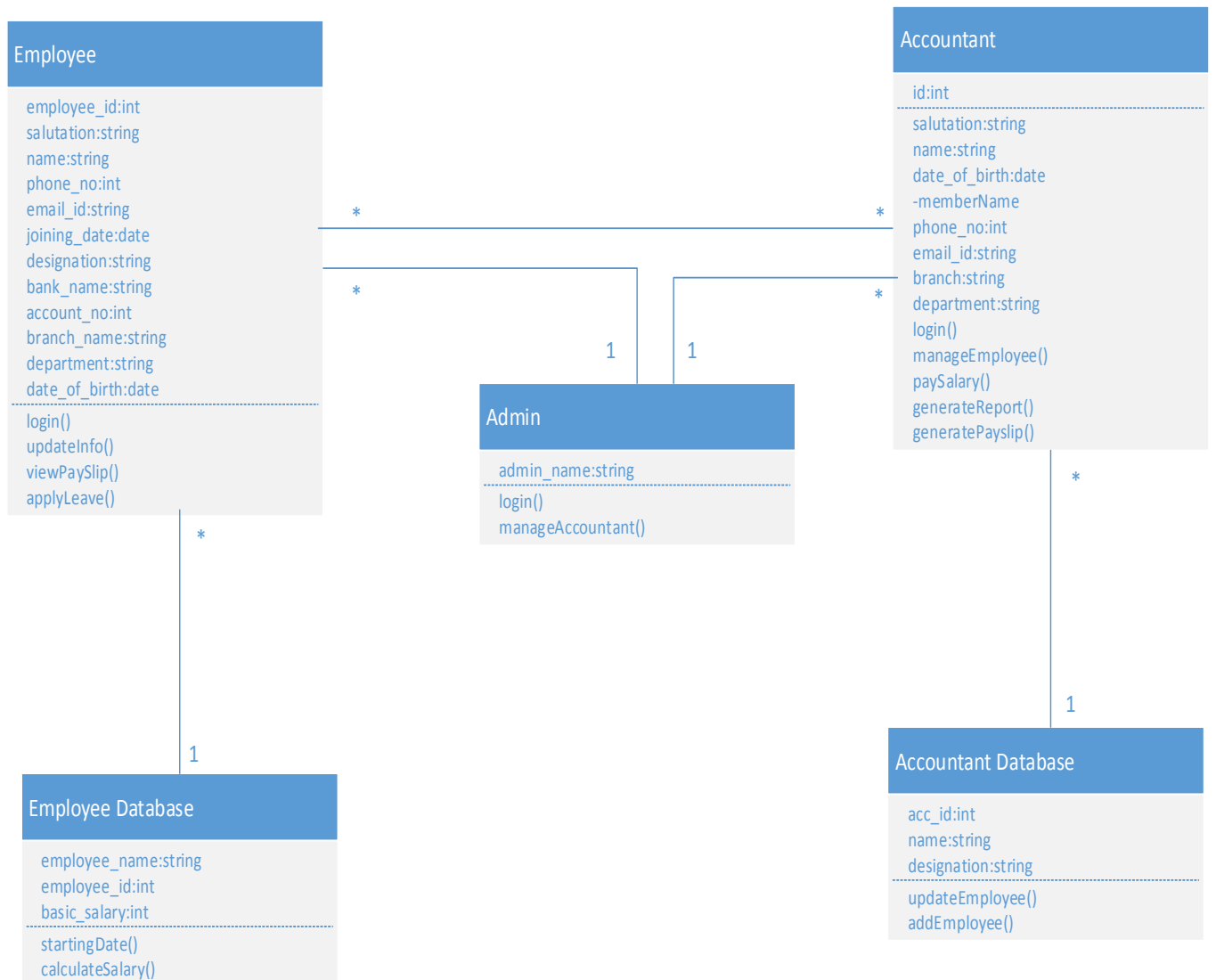
For PMS, we have developed following UML Diagrams:

- Use case Diagram
- Class Diagram
- Sequence Diagram
- Activity Diagram
- E-R Diagram
- Data Dictionary

### 3.1 Use case Diagram:

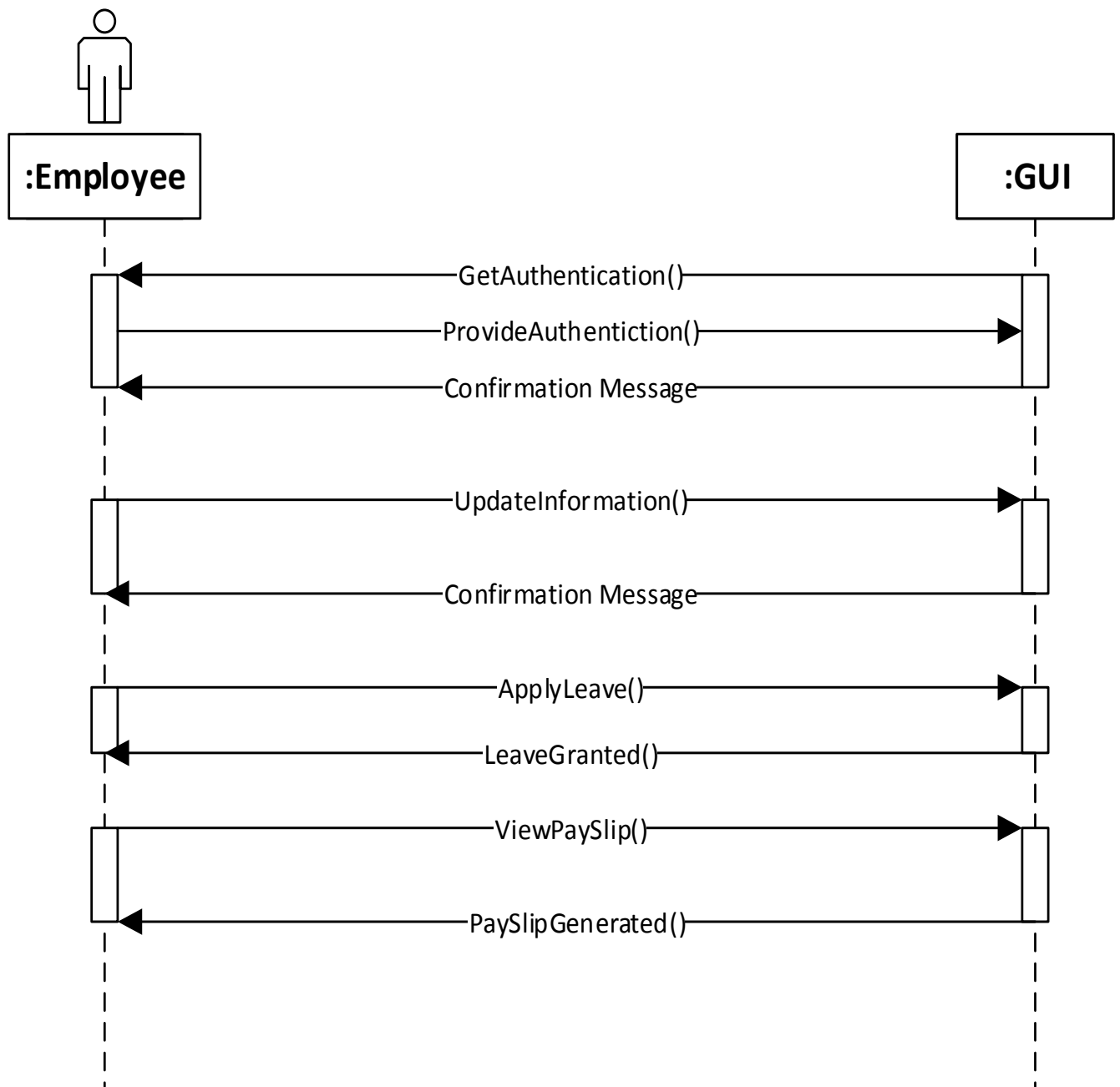


### 3.2 Class Diagram:

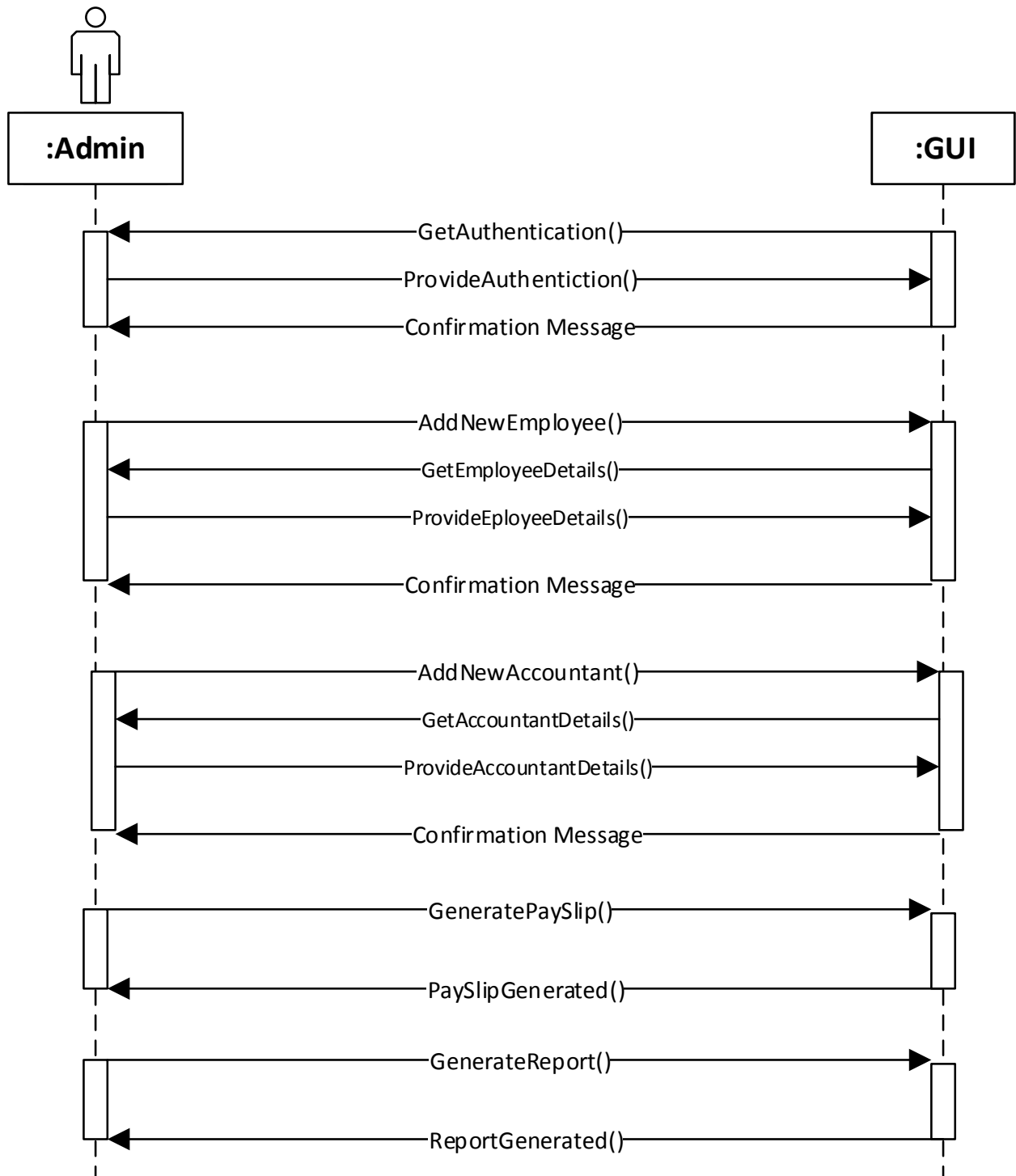


### 3.3 Sequence Diagram:

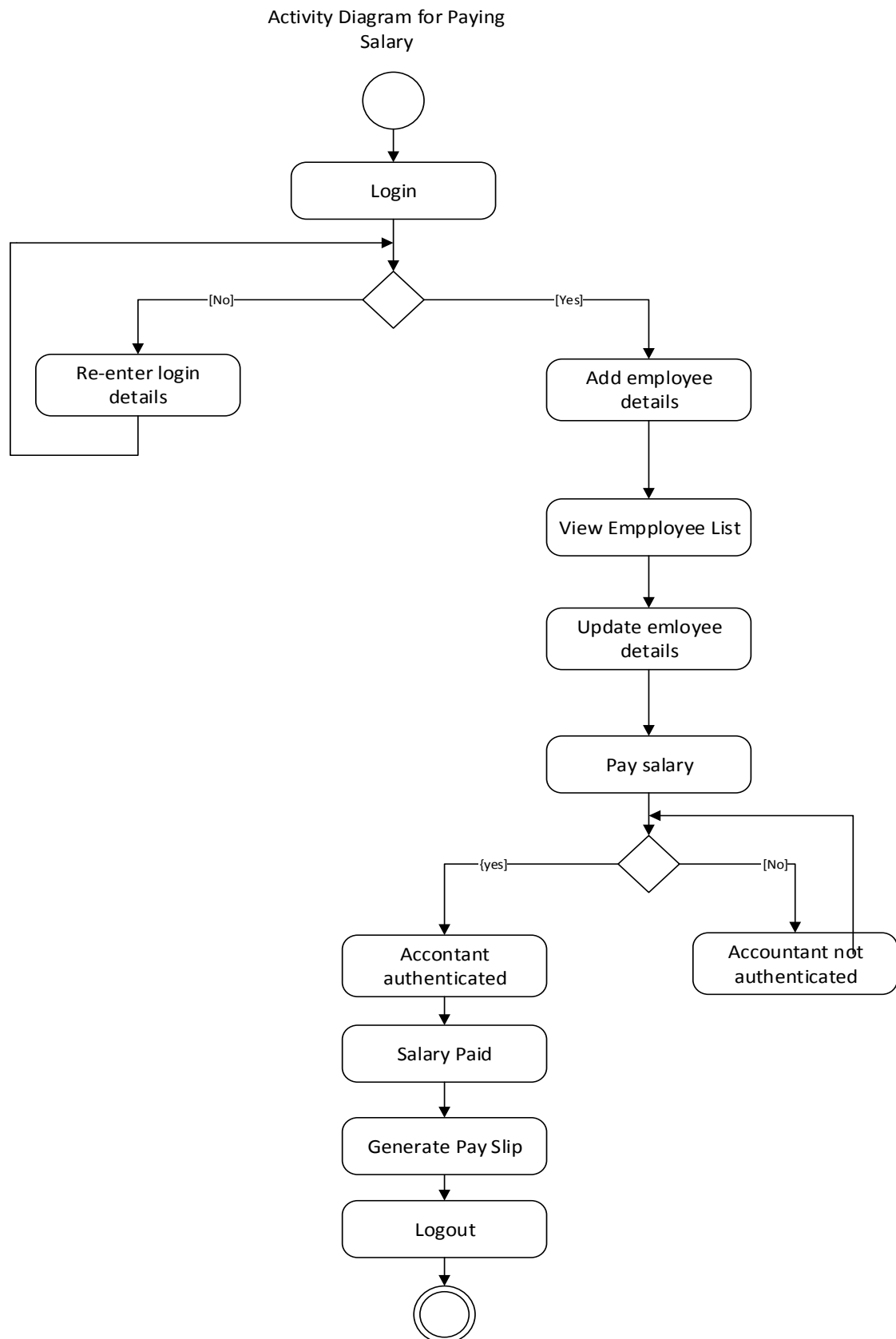
#### 1. For employee



## 2. For Admin

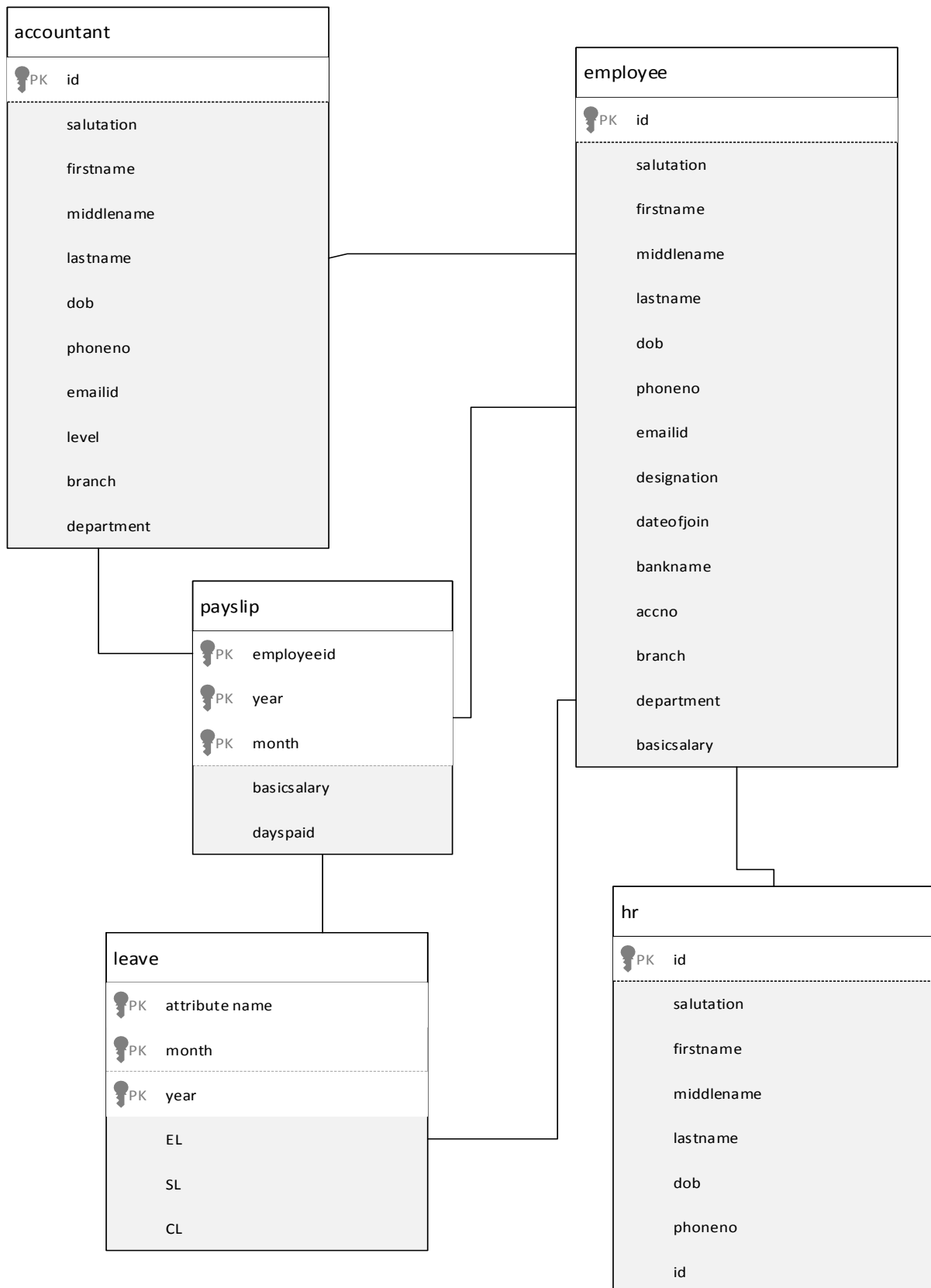


### 3.4 Activity Diagram:





### 3.5 E-R Diagram:



## 3.6 Data Dictionary:

### Accountant:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 <u>id</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	2 <u>salutation</u>	varchar(5)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	3 <u>firstname</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	4 <u>middlename</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	5 <u>lastname</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	6 <u>dob</u>	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	7 <u>phoneno</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	8 <u>emailid</u>	varchar(30)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	9 <u>level</u>	int(1)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	10 <u>branch</u>	varchar(20)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	11 <u>department</u>	varchar(20)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values

☐ Check All    With selected: Browse    Change    Drop    Primary    Unique    Index

### Employee:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 <u>id</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	2 <u>salutation</u>	varchar(5)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	3 <u>firstname</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	4 <u>middlename</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	5 <u>lastname</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	6 <u>dob</u>	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	7 <u>phoneno</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	8 <u>emailid</u>	varchar(30)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	9 <u>designation</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	10 <u>dateofjoin</u>	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	11 <u>bankname</u>	varchar(25)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	12 <u>accno</u>	int(20)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	13 <u>branch</u>	varchar(20)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	14 <u>department</u>	varchar(20)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/>	15 <u>basicsalary</u>	float			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values

☐ Check All    With selected: Browse    Change    Drop    Primary    Unique    Index

## HR:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>id</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 2	<u>salutation</u>	varchar(5)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 3	<u>firstname</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 4	<u>middlename</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 5	<u>lastname</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 6	<u>dob</u>	date			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 7	<u>phoneno</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 8	<u>emailid</u>	varchar(30)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values

☐ Check All    With selected: Browse Change Drop Primary Unique Index

## Payslip:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>employeeid</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 2	<u>month</u>	varchar(10)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 3	<u>year</u>	int(5)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 4	<u>basicsalary</u>	float			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 5	<u>dayspaid</u>	int(3)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values

☐ Check All    With selected: Browse Change Drop Primary Unique Index

## Leave:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>EL</u>	int(3)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 2	<u>SL</u>	int(3)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 3	<u>CL</u>	int(3)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 4	<u>month</u>	varchar(10)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 5	<u>year</u>	int(5)			No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values
<input type="checkbox"/> 6	<u>employeeid</u>	varchar(15)	latin1_swedish_ci		No	None		Change  Drop  Primary  Unique  Index  Spatial  Fulltext  Distinct values

☐ Check All    With selected: Browse Change Drop Primary Unique Index

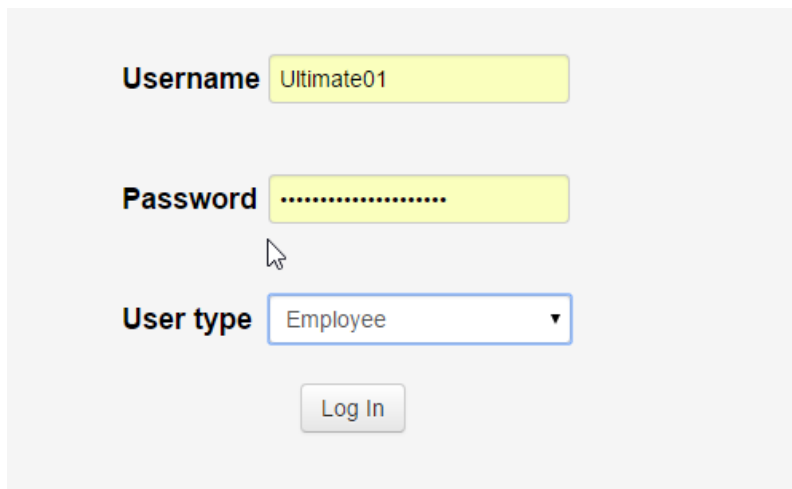
## Chapter 4 System Implementation

### 4.1 Modules

#### 4.1.1 Login Module:

This module provides login of the authorized users on the base of the user types.

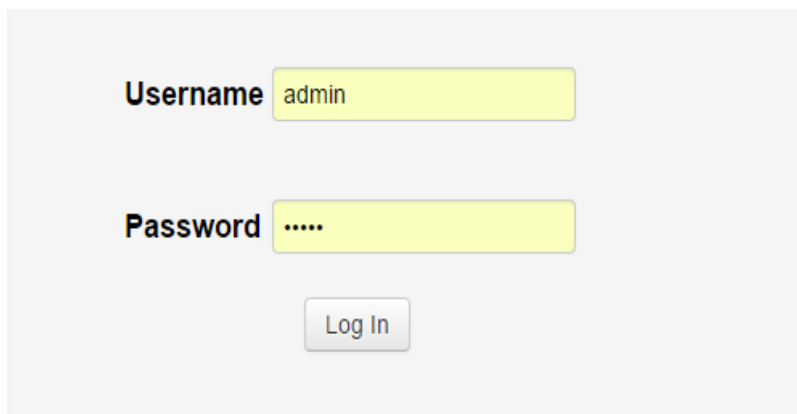
This is the **Login** page for an **Employee**.



The screenshot shows a login form for an Employee. It has three input fields: 'Username' with the value 'Ultimate01', 'Password' with masked characters '.....', and 'User type' with a dropdown menu showing 'Employee'. Below these fields is a 'Log In' button.

Similarly, Login for Accountant and HR manager is done in the same way. Just the main difference between them would be user type and their different username passwords respectively.

This is **Login** page for **Admin**

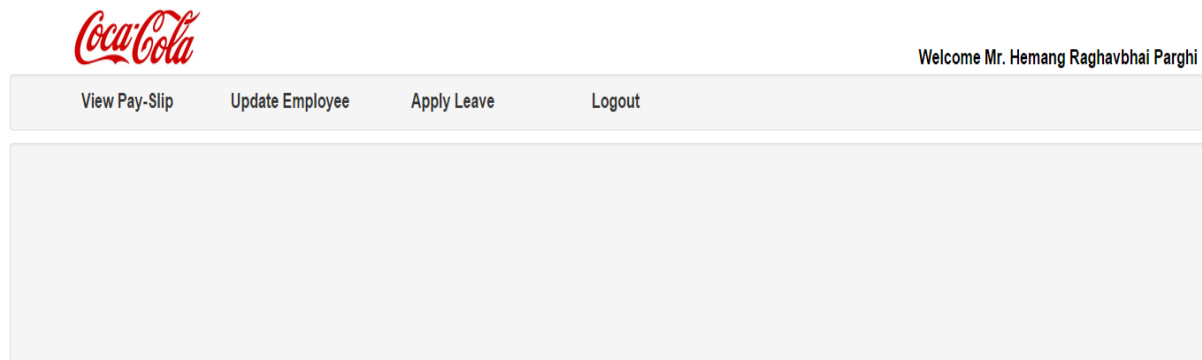


The screenshot shows a login form for an Admin. It has two input fields: 'Username' with the value 'admin' and 'Password' with masked characters '.....'. Below these fields is a 'Log In' button.

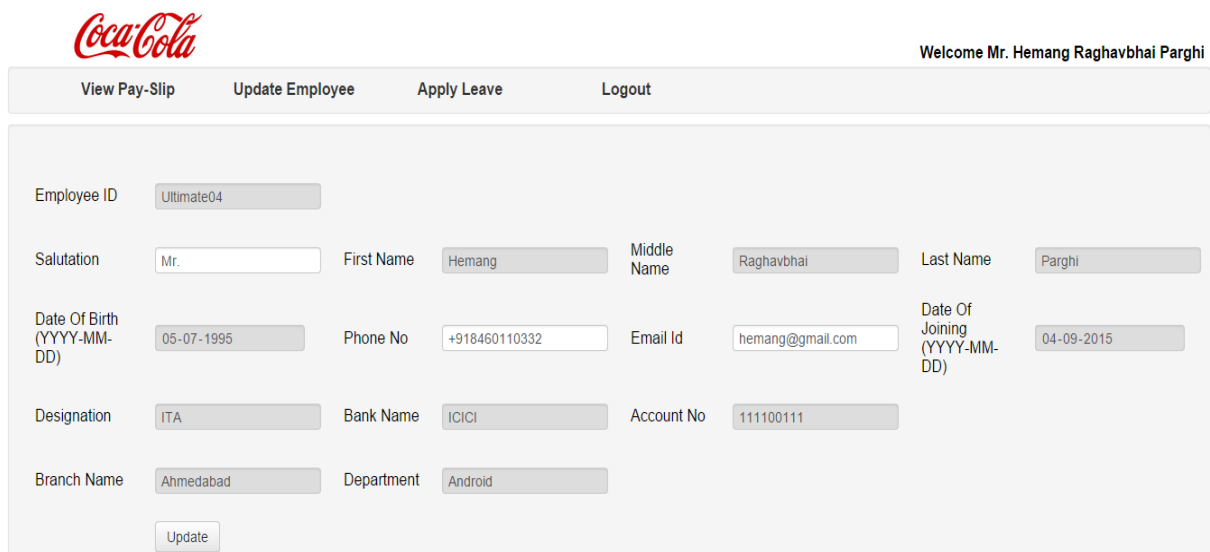
### 4.1.2 Employee Module

This module provides various operations and tasks which a employee is authorized to perform.

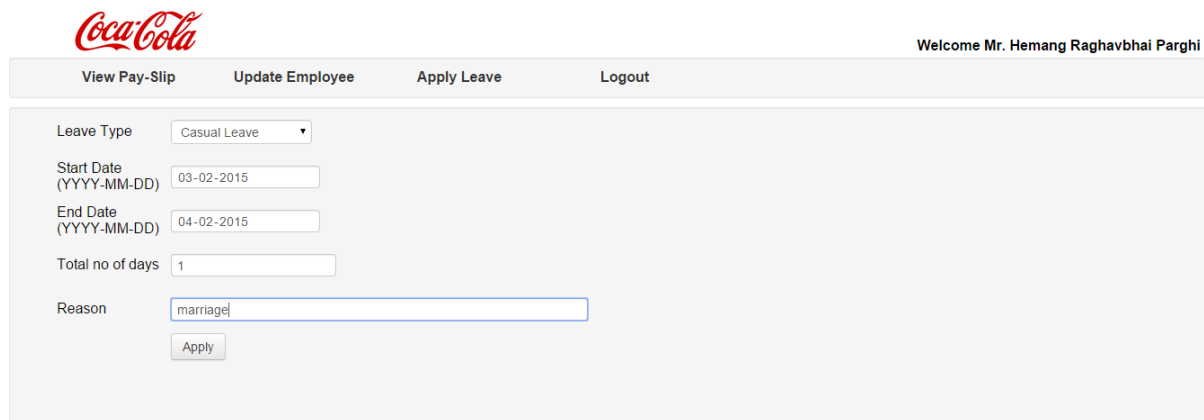
First of all, whenever **Employee** is authenticated, he comes across the main page.



Click **Update Employee** to update employee information by employee himself. Employee is allowed only to update some particular info. And then clicking on **Update** button, information will be updated.



To apply leave, click on **Apply Leave**. It will display the following selection boxes. Leave will be granted on clicking **Apply** option and accordingly changes will be made in the database.



**Coca-Cola** Welcome Mr. Hemang Raghavbhai Parghi

View Pay-Slip Update Employee Apply Leave Logout

Leave Type

Start Date (YYYY-MM-DD)

End Date (YYYY-MM-DD)

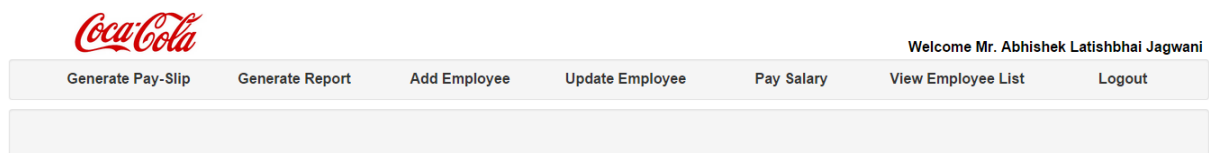
Total no of days

Reason

### 4.1.3 Accountant module

This is a module which provides an accountant to perform various operations.

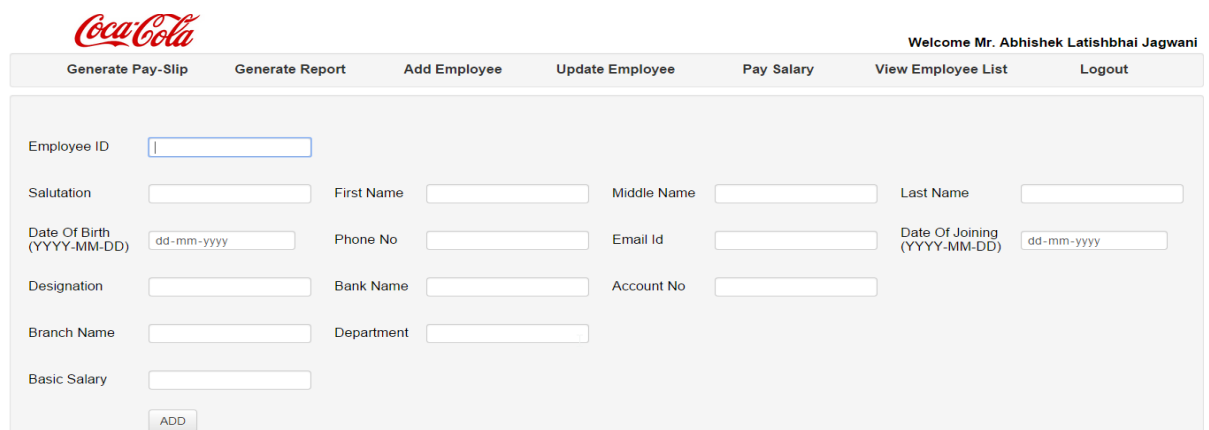
When **Accountant** is logged in, he goes to the main page.



**Coca-Cola** Welcome Mr. Abhishek Latishbhai Jagwani

Generate Pay-Slip Generate Report Add Employee Update Employee Pay Salary View Employee List Logout

If you want to add new employee, click on **Add Employee**. System will ask for certain details of the employee. By providing specific information and then clicking on **Add** button will add new employee.



**Coca-Cola** Welcome Mr. Abhishek Latishbhai Jagwani

Generate Pay-Slip Generate Report Add Employee Update Employee Pay Salary View Employee List Logout

Employee ID

Salutation  First Name  Middle Name  Last Name


Date Of Birth (YYYY-MM-DD)  Phone No  Email Id  Date Of Joining (YYYY-MM-DD)

Designation  Bank Name  Account No

Branch Name  Department

Basic Salary

If you want to view the current working employees, click on **View Employee List**. It will display the list of the employees.




Welcome Mr. Abhishek Latishbhai Jagwani

Generate Pay-Slip   Generate Report   Add Employee   Update Employee   Pay Salary   View Employee List   Logout

### Employee List

Name	ID	DOB	PhoneNo	Designation	Basic Salary(Per Month)	Date Of Join
Mr. Parth	Ultimate01	1995-07-05	+917405500480	IT Analyst	17000	2011-08-09
Mr. Mayur	Ultimate02	1995-08-05	+919662968362	SE	20000	2010-12-09
Mr. Sanyam	Ultimate03	1995-07-05	9913872306	SE	25000	2010-12-30
Mr. Hemang	Ultimate04	1995-07-05	+918460110332	ITA	35000	2015-09-04

If you want to update any information of any employee, click **Update Employee**. It will first ask for an Employee ID.



Welcome Mr. Abhishek Latishbhai Jagwani

Generate Pay-Slip   Generate Report   Add Employee   Update Employee   Pay Salary   View Employee List   Logout

Employee ID

By providing particular Employee ID and the clicking on **Update** button, system will ask for employee details which you want to update. Then clicking on **Update** button to update the information of an employee.

Employee ID

Salutation  First Name  Middle Name  Last Name

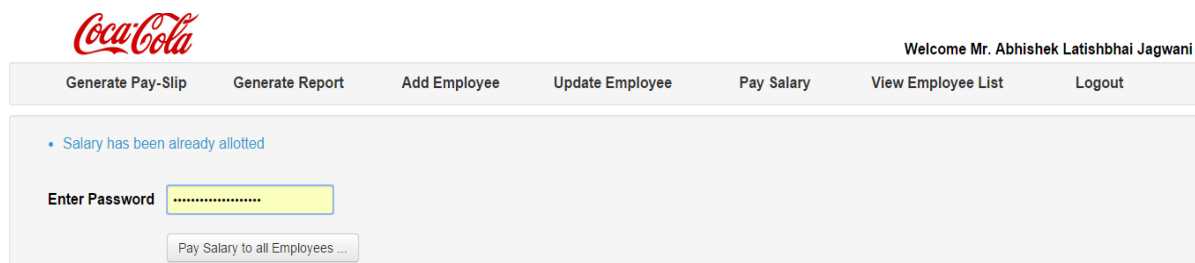
Date Of Birth (YYYY-MM-DD)  Phone No  Email Id  Date Of Joining (YYYY-MM-DD)

Designation  Bank Name  Account No

Branch Name  Department

Basic Salary

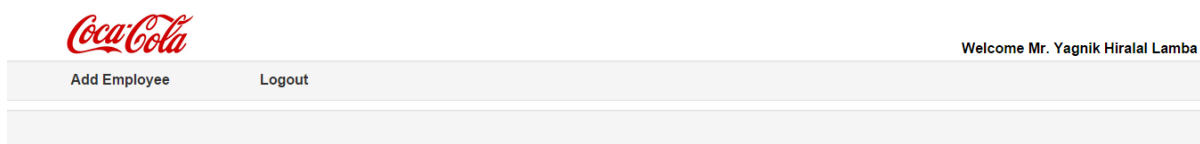
If you want to pay salary to all the employees, click on **Pay Salary**. System will ask the user again to authenticate. Providing password and then clicking **Pay Salary to all employees** will gives us a confirmation message that **Salary has been paid** in case of salary not being paid. But if salary is already been paid, then it will give a message that **Salary has already been allotted**.



#### 4.1.4 HR Module

This module provides a HR manager to perform various operations and tasks to which he/her is authorized.

First of all, when a **HR Manager** is logged in, he/she interacts with the main page.

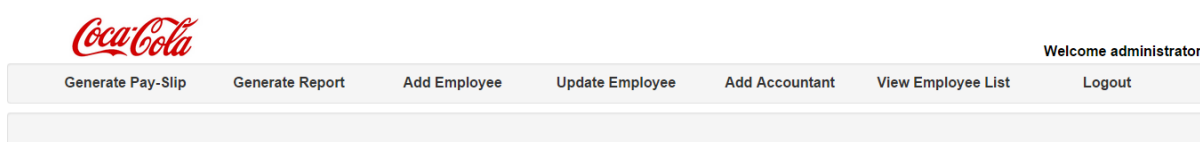


The HR manager is only allowed to add new employee. By clicking on **Add Employee**, it functions like same as we have seen in section 4.1.2 for accountant module.

#### 4.1.5 Admin Module

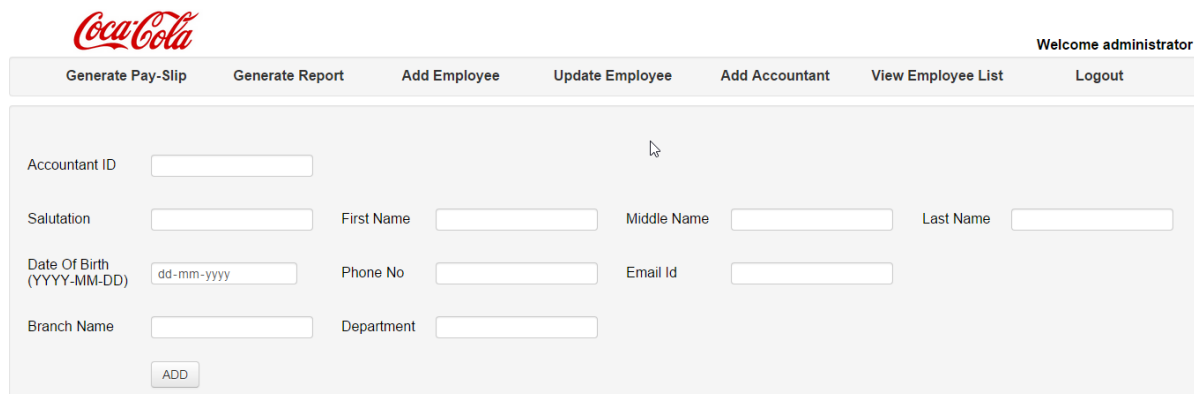
This module provides number of operations which ar authorized to the admin of the organization.

Whenever **Admin** is logged in, he/her comes across through the main page.





If you want to add any new Accountant, then click on the **Add Accountant** option. System will ask for the details of the accountant that you wish to add. After providing necessary details, click on **Add** button and the new accountant will be added to the list.



The screenshot shows the Coca-Cola Admin interface. At the top, there is a navigation bar with the Coca-Cola logo on the left and a 'Welcome administrator' message on the right. Below the logo, there are several menu items: 'Generate Pay-Slip', 'Generate Report', 'Add Employee', 'Update Employee', 'Add Accountant', 'View Employee List', and 'Logout'. The 'Add Accountant' option is highlighted. Below the navigation bar, there is a form for adding a new accountant. The form contains the following fields: 'Accountant ID' (a single-line text input), 'Salutation' (a dropdown menu), 'First Name' (a single-line text input), 'Middle Name' (a single-line text input), 'Last Name' (a single-line text input), 'Date Of Birth (YYYY-MM-DD)' (a date picker), 'Phone No' (a single-line text input), 'Email Id' (a single-line text input), 'Branch Name' (a single-line text input), and 'Department' (a single-line text input). At the bottom of the form, there is an 'ADD' button.

Rest of the functionalities of **Admin** are almost same as that of **Accountant**. There are also functionalities of **Generate Pay-Slip** and **Generate Report**. We will see this functionalities in later section.

## 4.2 Technologies used for Project implementation

### NetBeans IDE 7.4

The **NetBeans IDE** is an award-winning integrated development environment available for Windows, Mac, Linux, and Solaris. The NetBeans project consists of an open-source IDE and an application platform that enable developers to rapidly create web, enterprise, desktop, and mobile applications using the Java platform, as well as PHP, JavaScript and Ajax, Groovy and Grails, and C/C++. It is the first free IDE providing support for JDK 8, JDK 7, Java EE 7 including its related HTML5 enhancements, and JavaFX 2. The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages. Design GUIs for Java SE, HTML5, Java EE, PHP, C/C++, and Java ME applications quickly and smoothly by using editors and drag-and-drop tools in the IDE.

### WAMP Server

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases. WampServer installs automatically all you need to start developing web applications and is very intuitive to use. You will be able to tune your server without even touching the setting files.

## Chapter 5 Testing

Testing is a process carried out on a software to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications.

### 5.1 Testing Plan

The testing technique that is going to be used in the project is **Black Box Testing**. The expected inputs to the system are applied and only the outputs are checked.

### 5.2 Testing Strategy

The developer process repeats this testing sub process a number of times for the following phases.

- Unit Testing
- Integration Testing

Unit Testing tests a unit of code after coding of that unit is completed.

Integration Testing tests whether the previous programs that make up a system, interface with each other as desired.

System testing ensures that the system meets its stated design specifications.

Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification.

Testing is carried out in such a hierarchical manner so that each component is correct and the assembly/combination of component is correct. Merely testing a whole system at end would most likely throw up errors in component that would be very costly to trace and fix.

## **5.3 Test Cases**

### **5.3.1 User login**

**Condition:** Enter Username, password and the type of user.

**Expected Output:** User is authenticated correctly

**Actual Result:** As expected.

### **5.3.2 Add Employee Details**

**Condition:** User enters information about employee

**Expected Output:** Employee is added

**Actual Result:** As expected

### **5.3.3 Pay Salary**

**Condition:** User authenticates again and gives salary.

**Expected Output:** Salary is paid to all the employees.

**Actual Result:** As expected

### 5.3.4 Generate Pay Slip

**Condition:** User enters Employee ID, month and year.

**Expected Output:** Pay slip is generated for particular employee

**Actual Result:** As expected

**Payslip FEB-2015**

Mr. Parth Popatbhai Lathiya

Employee Details		Payment & Leave Details		Location Details	
Emp ID.	Ultimate01	Bank Name	State Bank of India	Location	Ahmedabad
Dsgn.	IT Analyst	Acc No.	XXXXXXXX2110	Base Br.	CC - Ahmedabad
Grade	C2	Days paid	28	Depute Br.	CC - Ahmedabad
PAN	XXXXXXXX792E	Leave Balance	EL -8 SL -1 CL 0	WON/SWON	2616141

Earnings	Arrears (INR)	Current (INR)	Deductions	Amount (INR)
Basic Salary		17000.0	Provident Fund	2040.0
House Rent Allowance		6000.0	Professional Tax	3400.0
Medical Allowance		1000.0	Extra Leaves	5100.0
Travelling Allowance		1000.0		
Personal Allowance		8500.0		
City Allowance		1000.0		
Performance Pay		8500.0		
Total Earnings (Current + Arrears)		43000.0	Total Deductions	10540.0

<b>Net Pay</b>	<b>32460.0</b>
----------------	----------------

### 5.3.5 Generate Pay Slip in case of wrong Information

**Condition:** User enters Employee ID, month and year.

**Expected Output:** Pay slip is not generated due to invalid information

**Actual Result:** As expected

**Coca-Cola**

Welcome Mr. Nilesh Popatbhai Lathiya

Generate Pay-Slip   Generate Report   Add Employee   Update Employee   Add Accountant   View Employee List   Logout

• Payslip of JAN/2015 is not generated OR Record of this Payslip is not available in our Database

Employee ID

Month :

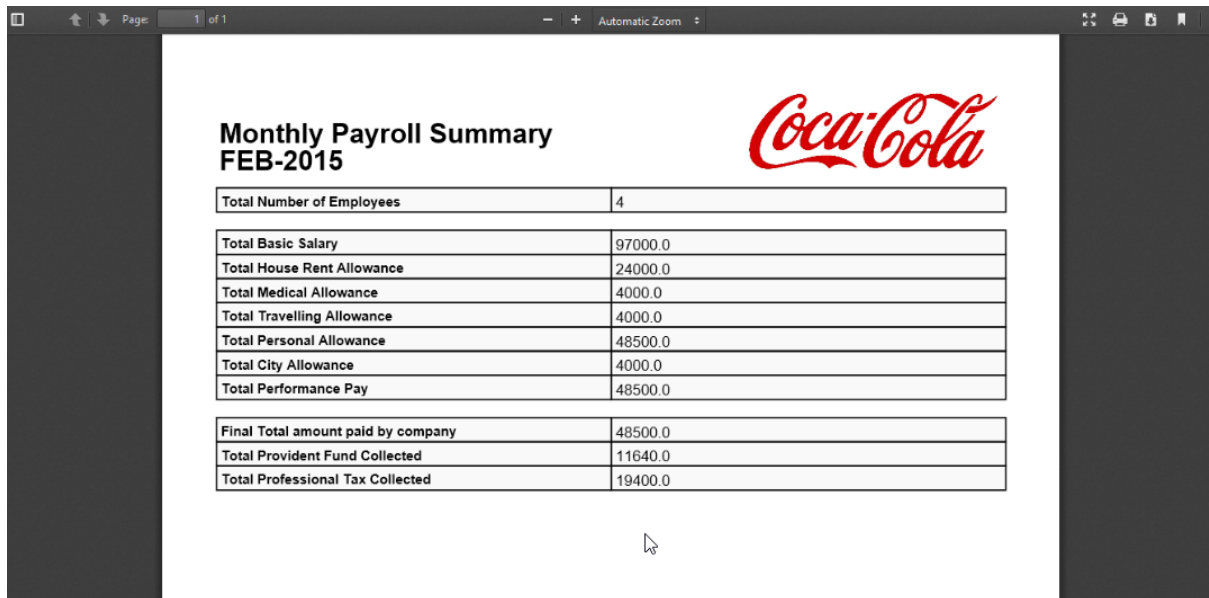
Year :

### 5.3.6 Generate Report

**Condition:** User enters month and year information.

**Expected Output:** Report of organization id generated for particular month

**Actual Result:** As expected



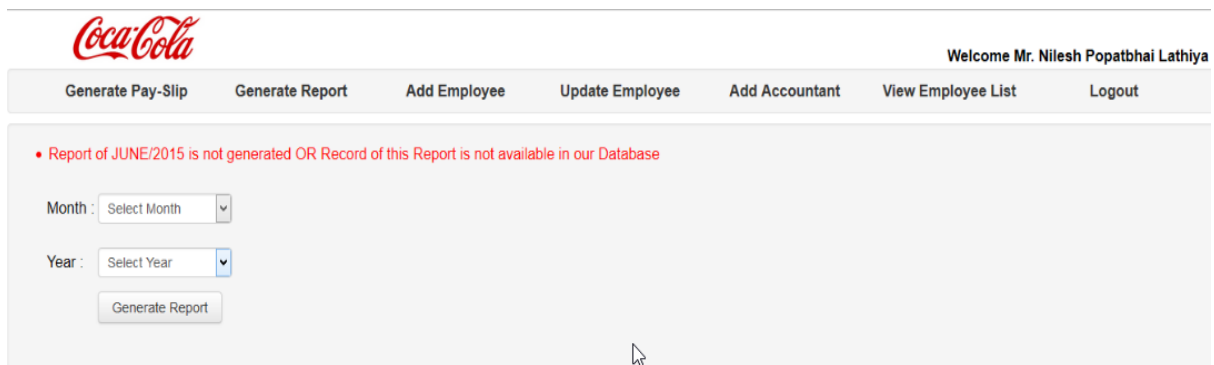
Monthly Payroll Summary FEB-2015	
Total Number of Employees	4
Total Basic Salary	97000.0
Total House Rent Allowance	24000.0
Total Medical Allowance	4000.0
Total Travelling Allowance	4000.0
Total Personal Allowance	48500.0
Total City Allowance	4000.0
Total Performance Pay	48500.0
Final Total amount paid by company	48500.0
Total Provident Fund Collected	11640.0
Total Professional Tax Collected	19400.0

### 5.3.7 Generate Report in case of wrong Information

**Condition:** User enters month and year.

**Expected Output:** Report is not generated due to invalid information

**Actual Result:** As expected



**Coca-Cola** Welcome Mr. Nilesh Popatbhai Lathiya

Generate Pay-Slip Generate Report Add Employee Update Employee Add Accountant View Employee List Logout

• Report of JUNE/2015 is not generated OR Record of this Report is not available in our Database

Month :

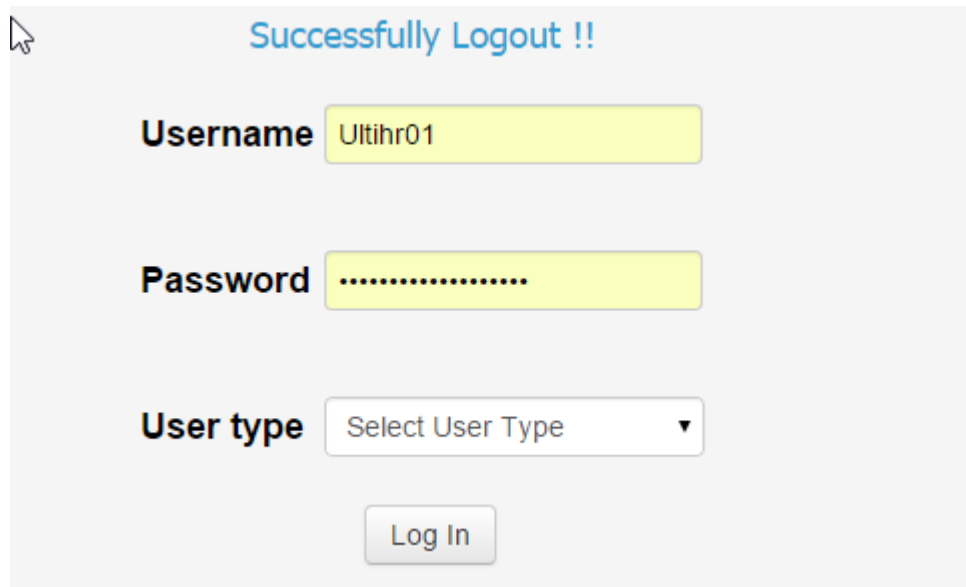
Year :

### 5.3.8 User Logout

**Condition:** User clicks on Logout button

**Expected Output:** User is logged out with a confirmation message.

**Actual Result:** As expected



A screenshot of a web application's login interface. At the top, a blue message 'Successfully Logout !!' is displayed. Below the message are three input fields: 'Username' with the value 'Ultihr01', 'Password' with masked characters '.....', and 'User type' with a dropdown menu showing 'Select User Type'. A 'Log In' button is located at the bottom of the form.

## **6 Limitations, Conclusion and Future Extensions**

### **6.1 Limitations**

- The system doesn't have facility to calculate and give back total provident fund collected from employee at the time of resignation.
- The system can't generate pay slip and report automatically at the last day of the month.
- The system can't generate pay slip of any employee at any point of time.

### **6.2 Conclusion**

We have implemented a project by understanding all the modules of this project. We have checked the feasibility and the requirements of the system. Then we designed the GUI and the flow control roughly on the paper. After this the actual implementation of our modules was started. Initially all the modules were developed separately and lately all were integrated with each other. Next phase we came across was testing. We tested each module individually and later the whole system. And thus through this system, an organization can easily maintain records of all the users efficiently and can perform various operations easily.

We experienced an enhancement in our Designing and technical skills of programming through the development of this Software.

### **6.3 Future Extensions**

- The System may provide facilities to employees to add their investment which is useful for tax reduction and Verification of investment documents will be carried out by accountant accordingly.
- The System may provide a facility to manage presence of employees on daily basis.
- The System may provide facility to increment the salary of employees based on their part time work.

## **Bibliography**

### **Books:**

1. Head first Servlets and JSP

By Bert Bates, Kathy Sierra and Bryan Basham

### **Websites:**

1. [www.stackoverflow.com](http://www.stackoverflow.com)
2. [www.javacodegeeks.com](http://www.javacodegeeks.com)
3. [www.wikipedia.org](http://www.wikipedia.org)