**EMPLOYEE PAYROLL SYSTEM**

**SOFTWARE ENGINEERING PROJECT REPORT**

**[Submitted in Partial Fulfilment]**

**As a part of the curriculum of**

**B.Sc (H) COMPUTER SCIENCE**

**From**

****

**Mata Sundri College for Women,New Delhi**

**University of Delhi**

Submitted By

Aradhana Sharma-17044570014

Radhika Arora-17044570009

B.Sc.(H) Computer Science

IV SEMESTER

**CERTIFICATE**

This is to certify that the project entitled, “Employee Payroll System”, is an employee payroll management system submitted by Aradhana Sharma and Radhika Arora of Mata Sundri College for Women in partial fulfillment of the requirements of BSc (Hons) Computer Science, IV semester.

It has been completed under the super vision of Ms. Ashema Hasti from Mata Sundri College for Women, Delhi University

Ms. Ashema Hasti

(Asst. Professor)

**ACKNOWLEDGEMENT**

It gives me immense pleasure to present to you this project on Employee Payroll System. We were lucky to get enormous support from extremely talented people, who deserve our great gratitude.

Firstly, we would like to thank our teacher and guide, Ms. Ashema Hasti who gave her valuable suggestions and ideas whenever we needed them. Also, she encouraged us to work on this project tirelessly by giving us numerous consultations.

we are also grateful to our parents for their constant support, guidance and providing us the necessary resources for the project.

Lastly, we would like to thank our classmates for their valuable suggestions for the betterment of this project and everybody who has helped us directly or indirectly in completion of this project.

We are immensely grateful to everyone involved in this project as without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

With sincere thanks,

Aradhana Sharma-17044570014

Radhika Arora-17044570009

**ABSTRACT**

A business,no matter how big or small it is, have employee/s . Employee is to be paid a salary that should be following government’s rules and regulations ,so as to avoid legal consequences.

Manually managing salary expenses accurately is tuff and hectic job.

So, to ease this process, we’ve come up with an employee payroll system software. It manages gross pay, allowances, overtime, bonus, tax deduction, pay-slip , paycheck generation of each and every employee.

And that too, accurately.

Employee Payroll System , with some inputs from admin, calculates the salary to be given to each employee.

Admin adds new employee’s account in the system by entering his/her personal details , salary and login details to be assigned.

Admin can view

* paycheque received status of each employee.
* salary details like gross salary, tax deducted , net pay of each employee.
* Tax structure , salary structure and work-hours.

Admin can update

* Salary and tax structure in general.
* Salary structure of each employee.
* Minimum work-hours.

Employee can view or generate payslip and receive paycheque from the system.

The system calculates and generates payslip and paycheque of each employee.

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Description** | **Page No.** |
| 1 | List of figures | 5 |
| 2 | List of tables | 6 |
| 3 | List of screens | 7 |
| 4 | Problem Statement | 8 |
| 5 | Software lifecycle model | 10 |
| 6 | Contents | 12 |
| 7 | Chapter-1 | 15 |
| 8 | Chapter-2 | 19 |
| 9 | Chapter-3 | 23 |
| 10 | Chapter-4 | 47 |
| 11 | Chapter-5 | 51 |
| 12 | Chapter-6 | 58 |
| 13 | Chapter-7 | 61 |
| 14 | Annexure | 64 |
| 15 | Conclusion | 83 |
| 16 | Bibliography | 84 |

**List of figs**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 3.1 | Level 0 DFD | 27 |
| 3.2 | Level 1 DFD | 28 |
| 3.3 | Level 2(Login) | 29 |
| 3.4 | Level 2(Employee Registration) | 30 |
| 3.5 | Level 2(Calculate Tax) | 31 |
| 3.6 | Level 2(Calculate Work-Hours) | 32 |
| 3.7 | Level 2(Calculate salary) | 33 |
| 3.8 | Level 2(Generate Payslip) | 34 |
| 3.9 | Level 2(Issue Paycheque) | 35 |
| 3.10 | Level 2(Salary Revision) | 36 |
| 3.11 | Use Case Diagram | 37 |
| 4.1 | ER Diagram | 48 |
| 7.1 | Control Flow Graph | 63 |

**List of tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page No.** |
| 4.1 | Data Design(Admin) | 48 |
| 4.2 | Data Design(Employee) | 49 |
| 4.3 | Data Design(Tax) | 49 |
| 4.4 | Data Design( Work Hours) | 50 |
| 4.5 | Data Design(Salary) | 51 |
| 5.1 | Project scheduling Table | 52 |
| 5.2 | Time Line Chart | 53 |
| 5.3 | Function Point Complexity Weight | 56 |
| 5.4 | COCOMO II Complexity weight | 57 |
| 6.1 | Risk Impact | 59 |

**List of screens**

|  |  |  |
| --- | --- | --- |
| **Screen No.** | **Description** | **Page No.** |
| 1 | Login screen | 65 |
| 2 | Login unsuccessful | 66 |
| 3 | Home page for Admin | 67 |
| 4 | Home page for Employee | 68 |
| 5 | Get pay slip screen | 69 |
| 6 | Get pay cheque screen | 70 |
| 7 | Registration of an employee | 71 |
| 8 | Assigning login credentials | 72 |
| 9 | Employee registration successful | 73 |
| 10 | Employee registration unsuccessful | 74 |
| 11 | Updating salary structure | 75 |
| 12 | Salary structure updation | 76 |
| 13 | Salary updation successful | 77 |
| 14 | View salary or tax detail | 78 |
| 15 | Viewing salary details | 79 |
| 16 | Viewing tax details | 80 |
| 17 | Updating tax details | 81 |
| 18 | Tax updation successful | 82 |
| 19 | Enter work hour | 83 |

**PROBLEM STATEMENT**

**INTRODUCTION**

Employee Payroll System is software used to cater payment process which deals with Salary of employees, tax deduction, bonus given, extra hours worked wages, late arrival deduction, printing and delivering cheque and pay-slip.

Admin needs to feed information like hours worked and wage information.

New tax laws are automatically updated on the system.

**DESCRIPTION**

Employee Payroll System deals with financial aspects of employee’s basic salary, other allowances, HRA, DA, increment, gross pay, group insurance and generation of pay-slip. It provides extensive features and reports.

**FUNCTIONALITIES**

ADMIN:

* Login: Admin is provided with unique id and password and permission to update the details.
* Register: Admin can register an employee to the company’s database by entering his/her personal details.
* Calculate tax: Employees tax is calculated according to the grade of his/her salary.
* Calculate work hours: Total number of hours worked are calculated by adding extra hours worked and deducting early leaving or an off day hours from the normally fixed working hours per week.
* Calculate salary: final salary of an employee is calculated by adding basic, da, hr, other allowances which sums up to be the gross pay from which we deduct group insurance and tds to find the net pay.
* Generate pay slip: all the details of the net pay are used to generate pay slip for each employee.
* Collect pay cheque: delivering salary cheque to employees.
* Salary revision: Update salary structure.

EMPLOYEE:

* Login: Employee is provided with unique id and password and permission to view his/her details.
* Check salary calculation, pay slip details, personal details number of hours worked, tax deductions.
* Collect his/her cheque and pay-slip.

**SOFTWARE LIFECYCLE MODEL**

Prototyping model is the best suited model for this software because all our software implementers and designers can obtain feedback from the user early in the project so accommodating the changes is quite easy and cost of changes will also be less. Client and software manager can compare if the software made matches the software specification prior to the delivery of the final product. Stable funding would not be an issue for iterative process as the project is involving government interference. Also because:

* The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
* New requirements can be easily accommodated as there is scope for refinement.
* Missing functionalities can be easily figured out.
* Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
* The developed prototype can be reused by the developer for more complicated projects in the future.
* Flexibility in design.

**What is Prototyping model?**

During Prototyping model, the software development team, make clear requirements and design elements that results in mockups and prototypes of screens, reports, and processes.

Although some of the prototypes may appear to be very substantial, they’re generally similar to a movie set, everything looks good from the front but there’s nothing in the back.

When a prototype is generated, the developer produces the minimum amount of code necessary to clarify the requirements or design elements under consideration.

* The developed prototype can be reused by the developer for more complicated projects in the future.
* Flexibility in design.

**When to use prototyping model?**

The Prototyping Model should be used when the requirements of the product are not clearly understood or are unstable.

It can also be used if requirements are changing quickly.

This model can be successfully used for developing user interfaces, high technology software-intensive systems, and systems with complex algorithms and interfaces.

It is also a very good choice to demonstrate the technical feasibility of the product.

**CONTENTS**

**CHAPTER 1: INTRODUCTION 16**

* 1. Purpose
  2. Project scope

1.2.1 Objectives of the proposed system

* 1. Definition Acronyms Abbreviations
  2. References
  3. Overview

**CHAPTER 2: PROJECT DESCRIPTION 20**

2.1 Product Prospective

2.1.1 System Interfaces

2.1.2 Hardware Interfaces

2.1.3 Communication Interfaces

2.1.4 Memory Constraints

2.1.5 Operations

2.1.6 Site Adaption Requirements

2.2 Product Features

2.2.1 Admin’s Functionality

2.2.2 Employee’s Functionality

2.3 User Classes and Characteristics

2.3.1 User Characteristics

2.3.2 User Classes

2.4 Design and Implementation Constraints

2.5 Assumptions and Dependencies

**CHAPTER 3: SPECIFIC REQUIREMENTS 24**

3.1 External Interfaces

3.2 Functional Requirements

3.2.1 Admin’s Functionality

3.2.2 Employee’s Functionality

3.3 Performance Requirements

3.4 Logical Database Requirements

3.5 Design Constraints

3.6 Security Requirements

3.7 Software System Attributes

3.7.1 Reliability

3.7.2 Availability

3.7.3 Security

3.7.4 Maintainability

3.7.5 Probability

3.8 Data Flow Diagram

3.8.1 Context Level Diagram

3.8.2 Level-1 DFD

3.8.3 Level-2 DFD

3.9 Data Dictionary

3.10 Use Cases

3.10.1 Use Case Diagram

3.10.2 Use Case Description

**CHAPTER 4: DESIGN 48**

4.1 ER Diagram

4.2 Data Design

4.3 Component Level Design

**CHAPTER 5: ESTIMATION AND SCHEDULING 52**

5.1 Project Scheduling

5.2 Timeline Chart

5.3 Size Estimation (Function Based Metrics)

5.4 Cost Estimation (COCOMO II Model)

**CHAPTER 6: RISK ANALYSIS 59**

**CHAPTER 7: TESTING 62**

7.1 Control Flow Graph

7.2 Cyclomatic Complexity

7.3 Basis Path Set

**ANNEXURE** **65**

**CONCLUSION 84**

**BIBLIOGRAPHY 85**

**CHAPTER 1: INTRODUCTION**

An employee payroll management system is one in which the company has a record of its employee their salary details and tax deductions.

* Employee payroll system is a system that will automate the payroll procedure. It will keep a record of the salary details; calculate salary, taxes to be paid to government, cheque receiving status, pay-slip receiving status, extra working hour wages, and tax deduction. It will print cheque and pay-slip of each registered employee. The software will provide the functionalities as explained below:
* Login: Admin is provided with unique id and password and permission to update the details. Employee is provided with unique id and password and permission to view his/her details.
* Register: Admin can register an employee to the company’s database by entering his/her personal details.
* Calculate tax: Employees tax is calculated according to the grade of his/her salary.
* Calculate work hours: Total number of hours worked are calculated by adding extra hours worked and deducting early leaving or an off day hours from the normally fixed working hours per week.
* Calculate salary: final salary of an employee is calculated by adding basic, da, hr, other allowances which sums up to be the gross pay from which we deduct group insurance and tds to find the net pay.
* Generate pay slip: all the details of the net pay are used to generate pay slip for each employee.
* Collect pay cheque: delivering salary cheque to employees.
* Salary revision: Update salary structure. Check salary calculation, pay slip details, personal details number of hours worked, tax deductions.
* Employee can collect his/her cheque and pay-slip.

**1.1 Purpose:**

This is Software Requirement Specification (SRS) intended to describe the software’s functional and non-functional requirements. The requirements stated in the SRS shall be final. Any changes therein will be carried only after a written application from the employee. The SRS will then be revised after the approval of the project manager. The SRS once approved shall serve as the sole proof for the requirements stated by the customer. Further, it shall be the only legal document that shall be used for validation, in case any dispute arises between the employee and the developer.

**1.2 Project Scope:**

It is a web based application, which serves as a tool for the user of this application to experience a fully automated payroll system.

Once an employee has been registered he/she can:

* Login: Employee is provided with unique id and password and permission to view his/her details.
* Check salary calculation, pay slip details, personal details number of hours worked, tax deductions.
* Collect his/her cheque and pay-slip.

Admin can perform following functionalities:

* Login: Admin is provided with unique id and password and permission to update the details.
* Register: Admin can register an employee to the company’s database by entering his/her personal details.
* Calculate tax: Employees tax is calculated according to the grade of his/her salary.
* Calculate work hours: Total number of hours worked are calculated by adding extra hours worked and deducting early leaving or an off day hours from the normally fixed working hours per week.
* Calculate salary: final salary of an employee is calculated by adding basic, da, hr, other allowances which sums up to be the gross pay from which we deduct group insurance and tds to find the net pay.
* Generate pay slip: all the details of the net pay are used to generate pay slip for each employee.
* Collect pay cheque: delivering salary cheque to employees.
* Salary revision: Update salary structure.

**1.2.1 Objectives of the proposed system:**

To ease the payroll system done manually. This system provides following functionalities:

* Employee registration.
* Calculate tax to be deducted.
* Calculate hours worked and overtime or under time of each employee.
* Calculate net salary.
* Generates pay-slip.
* Issue pay cheque.
* Salary revision: Admin can revise salary of either an employee or in general.

**1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS:**

Admin: Person who interacts with the internal system and software both.

Employee: Person who interacts with the software.

**1.4 REFERENCES:**

1. P. Jalote,An integrated approach to software Engineering, Narosa Publishing House, Ed.3,2011
2. R.S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill, Ed 7,2010
3. IEEE Std 1233-1998,IEEE Recommended Practice for Software Requirements Specifications

**1.5 OVERVIEW:**

Employee Payroll System is software used to cater payment process which deals with Salary of employees, tax deduction, bonus given, extra hours worked wages, late arrival deduction, printing and delivering cheque and pay-slip.Admin needs to feed information like hours worked and wage information. New tax laws are automatically updated on the system.

SRS will include two sections:

**Overall Description** will describe major components of the system, interconnection and external interfaces.

**Specific Requirements** will describe the functions of actors, their role in the system and constraint

**CHAPTER 2: PROJECT DESCRIPTION**

**2.1 Product Prospective:**

This software is built for payroll management system. It contains employee’s information in the database and retrieves its salary cheque, pa yslip, taxes which is further stored in the database itself. The database stores the employee’s information including his/her personal details, salary structure, salary delivery details, pay slip receiving details, tax details. It stores any changes made to the existing information by the admin such as personal details, salary structure, addition or deletion of a new record.

This document provides an overview of the system functionality and system interaction with other system. The document also mentions the system constraints and assumptions about the product.

* + 1. **System Interfaces:**

**Operating System:** Windows (for best support and user friendliness)

**Database:** To save customers records, SQL, database is used

**Web Browser:** Chrome, Internet Explorer

**2.1.2 Hardware Interfaces:**

RAM: 512 MB

Disk Space: 2 GB

CPU: Intel processor

Keyboard, Monitor, Mouse, Printer

.

**2.1.3Communication Interfaces**:

None

**2.1.4 Memory Constraints:**

Hardware memory: The growth of company is unpredictable; to resolve the future problems occurs while enhancing the system is controlled by larger memory as possible. So the memory constraint in the server side is extended up to 1TB.

* Memory: 1TB

**2.1.5 Operations:**

Admin and employee need to provide input like option to choose and basic details. Processing of that data is done automatically in the system using basic calculator. Output in the form of data or pay slip or pay cheque is given.

**2.1.6 Site Adaptation Requirements:**

No site adaptation is necessary in this project because the Employee payroll system is portable. The entire system is transported to wherever it is needed. No external dependencies are in place and operation of the system will never change due to location.

* 1. **Product Features:**
     1. **Admin’s Functionality:**
     + Login: Admin is provided with unique id and password and permission to update the details.
  + Register: Admin can register an employee to the company’s database by entering his/her personal details.
  + Calculate tax: Employee’s tax is calculated according to the grade of his/her salary.
  + Calculate work hours: Total number of hours worked are calculated by adding extra hours worked and deducting early leaving or an off day hours from the normally fixed working hours per week.
  + Calculate salary: View net salary, gross salary and tax deducted of each employee.
  + Issue pay cheque: pay cheque received status is notified.
  + Salary revision: Update salary structure. Check salary calculation, pay slip details, personal details number of hours worked, tax deductions.
    1. **Employee’s Functionality:**
       - Login: Employee is provided with unique id and password and permission to view his salary details, pay slip and tax details.
       - Generate pay slip: all the details of the net pay are used to generate pay slip for each employee.
       - Employee can collect his/her cheque and pay-slip.
  1. **User Classes and Characteristics:**
     1. **User Characteristics:**
        + - Admin: Can update tax details, change salary structure and also add new registrations to the database, also can add number of work hours and view salary details of employee.
          - Employee: Can view his/her salary, pay slip and tax details, get hard copy of pay slip and pay cheque.
     2. **User Classes:**

There are two classes of user in this software:

* Admin: He/she is the HR Head of the company and has to have a special username and password to perform aforementioned functions. He/she would not be required to be registered.
* Employee: This class defines the employee of the company who can log in to the system only after they have been registered by the admin i.e. their record exists in the database. They will be allowed to perform limited operations as clarified above.
  1. **Design and Implementation Constraints:**

**Design Constraints:**

* + - * The software is designed for web portal so screen size will be a major constraint

**Implementation Constraints:**

* Internet connectivity is required to allow the software fetch data from the database over the internet.
* Hard disk:20GB
* Memory:4GB
  1. **Assumptions and Dependencies:**

**Assumption**: All the users have a mailing account and have the computer know-how.

**Dependency**: Internet connectivity is must.

**2.6 Apportioning of Requirements**

* It is possible in the future that a few additional features be implemented into this system.
* Management System: This will allow the system to manage effectively the other resources in the easiest way.

**CHAPTER 3: SPECIFIC REQUIREMENTS**

**3.1 External Interfaces:**

* No such external interface is needed.

**3.2 Functional Requirements:**

This software will work on windows to make it easily accessible to the customer. User need to login to his/her eid as username and password. A new user will have to be registered by the admin with all the necessary details.

User can avail for various services like sms alert, salary structure, cheque receiving etc.

**3.2.1 Admin’s Functionality:**

* Login- To login admin needs to enter valid userid and password which is checked for it validation and if valid the admin home page is displayed if not a try again message is displayed.
* Employee registration- To register an employee the admin first needs to enter his/her personal details and then admin assigns the login credentials to the employee which is then check for authenticity uniqueness on basis of which registration becomes successful or unsuccessful
* Calculate tax-Tax calculated using the stored percentages of professional tax, epf, tds etc.
* Calculateworkhours-Number of hours worked by an employee are calculated using min number of hours to be worked, and overtime or early leaving details from the attendance database.
* Calculate salary-Net salary is calculated by using following formulas.

Net pay= gross pay- gis – tds

Gross pay= basic+hr+da+other allowances

* Salary revision-to update salary details of an employee admin first needs to input the eid, post, department, grade and then can edit the current salary details with the revised once

**3.2.2Employee’s Functionality:**

* Login- To login employee needs to enter valid user id and password which is checked for it validation and if valid the employee home page is displayed if not a try again message is displayed.
* Get pay slip- employee needs to enter his eid and can then view or get a print of his pay slip.
* Collect pay cheque- employee needs to enter his eid and then can get his pay cheque printed.
  1. **Performance Requirements:**

The software is said to be working at its utmost best when:

The admin updates the database regularly for new employees.

The software has protection from threats i.e. virus protection.

The software has enough memory for a large number of

Employees

* 1. **Logical Database Requirements:**

The database contains the following information:

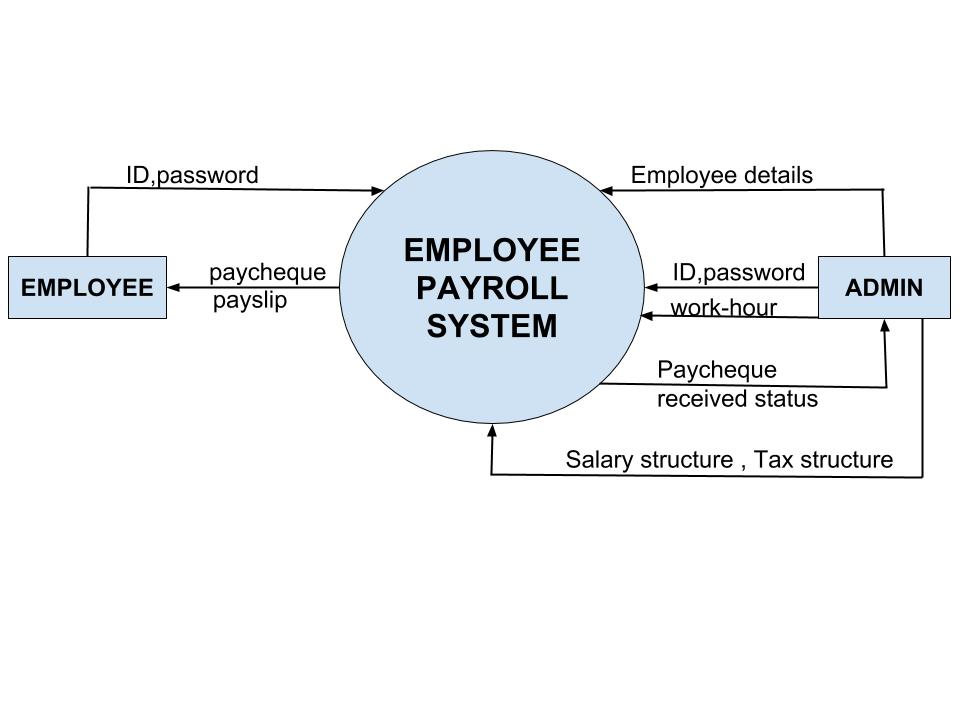
* Emp details: this database contains the personal information of the employee and also his login credentials
* Emp salary: this database contains the salary structure of an employee
* Tax: this database consists of the tax %age tds, epf, p-tax etc.
* Attendance: this database has the attendance of all employees of each day
  1. **Design Constraints:**

The software is designed for web portal so screen size will be a major constraint

* 1. **Security Requirements:**

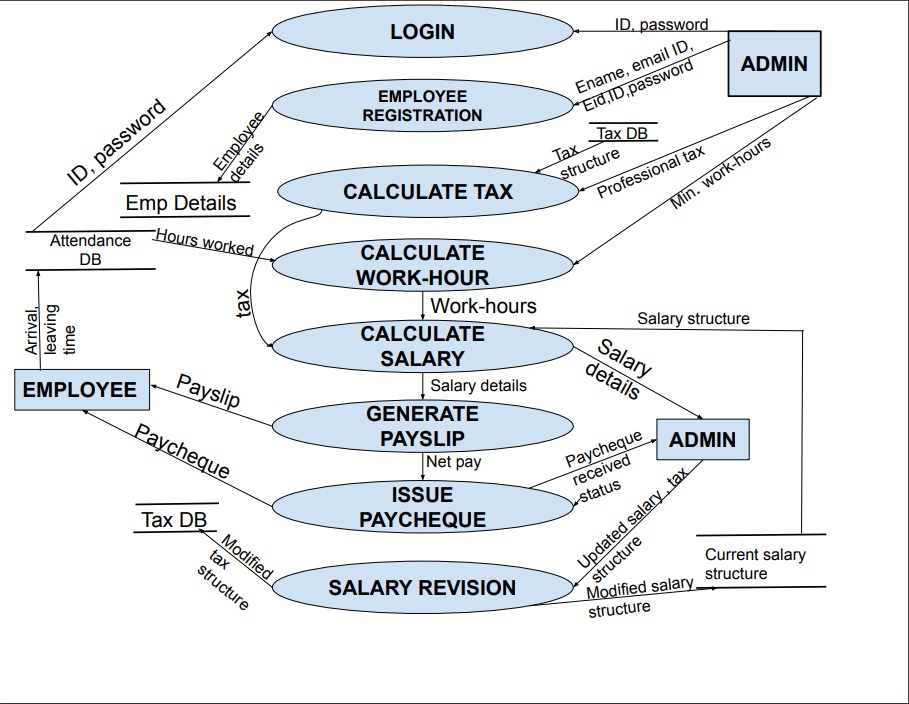
Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Specific requirements in this area could include the need to:

* Utilize certain cryptographic techniques
* Keep specific log or history data sets
* Assign certain functions to different modules
* Restrict communications between some areas of the program
* Check data integrity for critical variables
* Later version of the software will incorporate encryption techniques in the user/license authentication process.
* The software will include an error tracking log that will help the user understand what error occurred when the application crashed along with suggestions on how to prevent the error from occurring again.
* Communication needs to be restricted when the application is validating the user or license. (i.e., using https).
  1. **Software System Attributes:**
     1. **Reliability:** The software is reliable.
     2. **Availability:** Previous pay slip records and cheque date will be available for the employee’s convenience and for maintaining previous records.
     3. **Security:** Software will not affect data stored outside of its server nor will it affect any other application installed in system.
     4. **Maintainability:** Software should maintain correct schedule of the salary cheque delivery and pay slip receiving.
     5. **Portability:** This software is portable, as it can be installed at any system fulfilling hardware requirements.
  2. **Data Flow Diagram:**
     1. **Context Level Diagram(Level-0)**

****

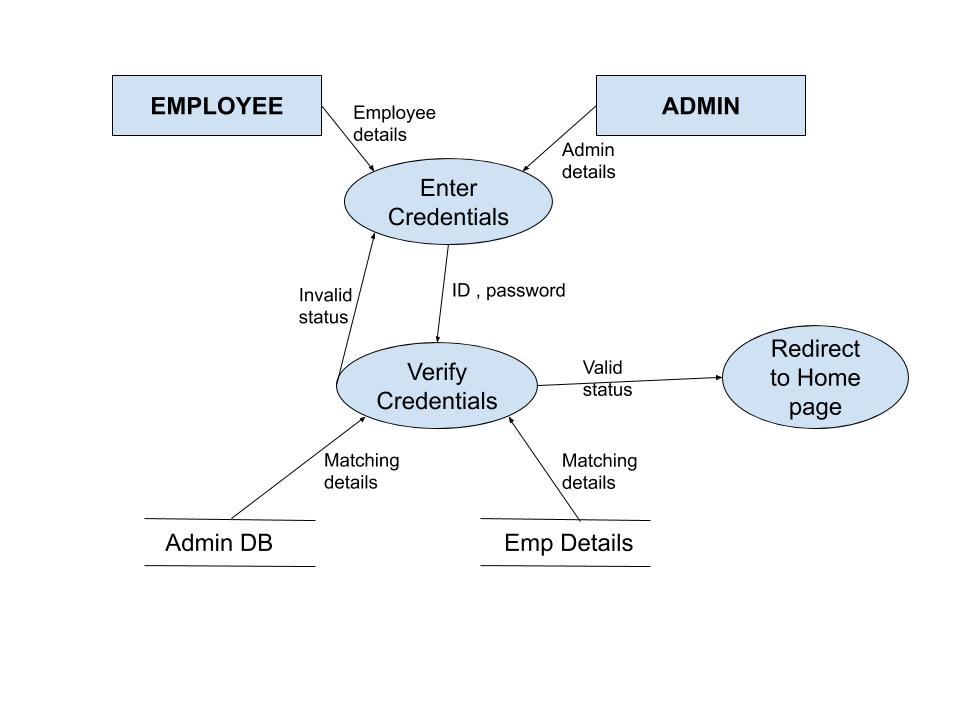
**Fig 3.1. Context Diagram**

* + 1. **Level-1 Data Flow Diagram**

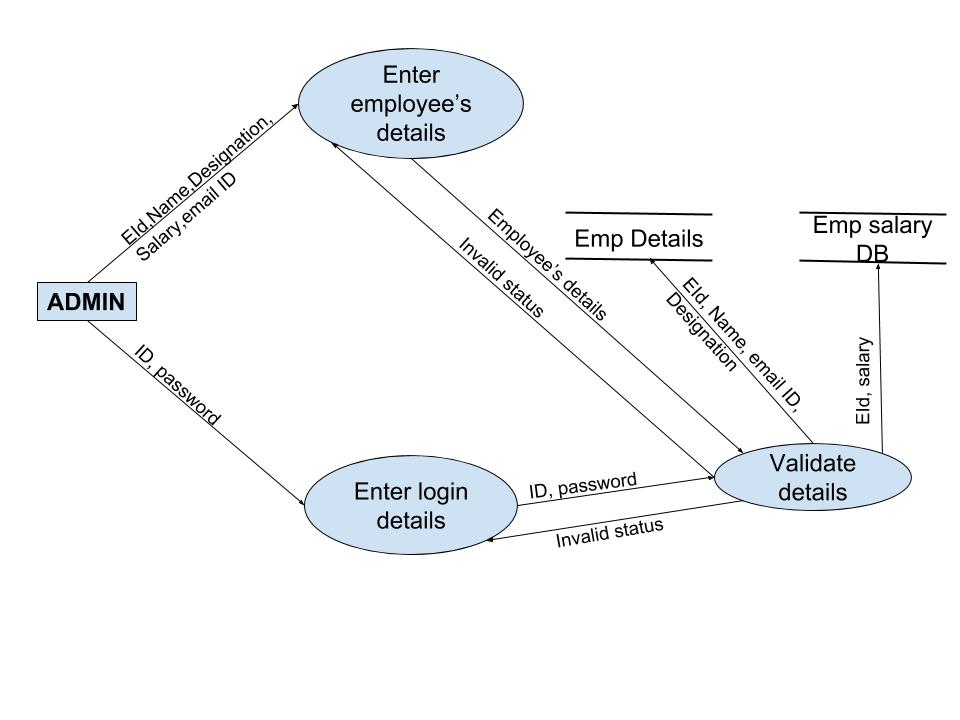
****

**Fig 3.2. Level 1 DFD**

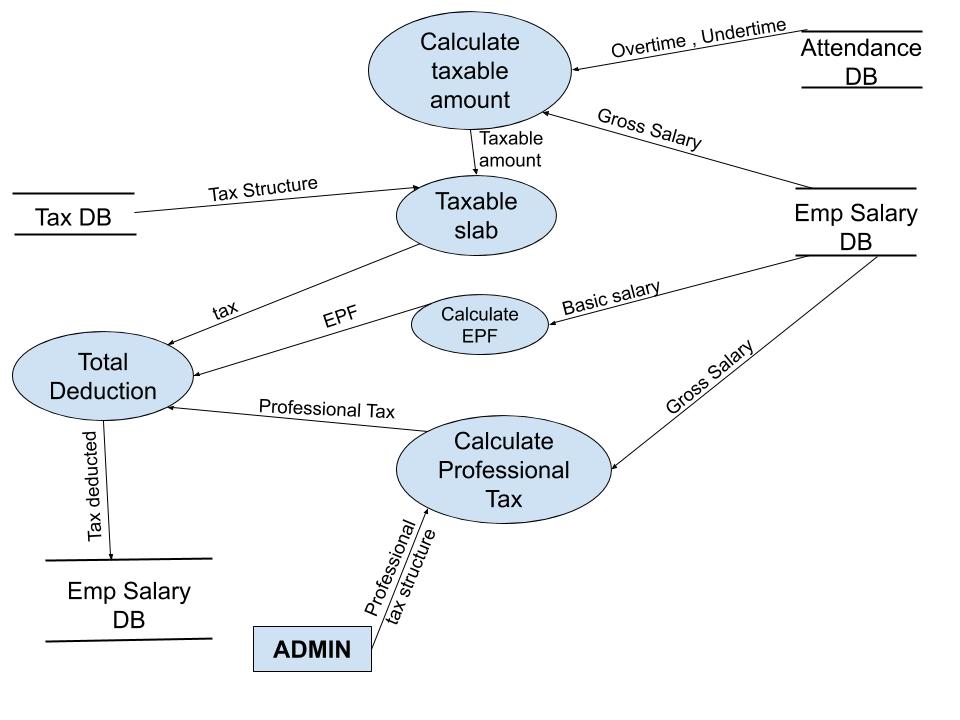
**3.8.3: Level-2 Data Flow Diagram**

**LOGIN:**

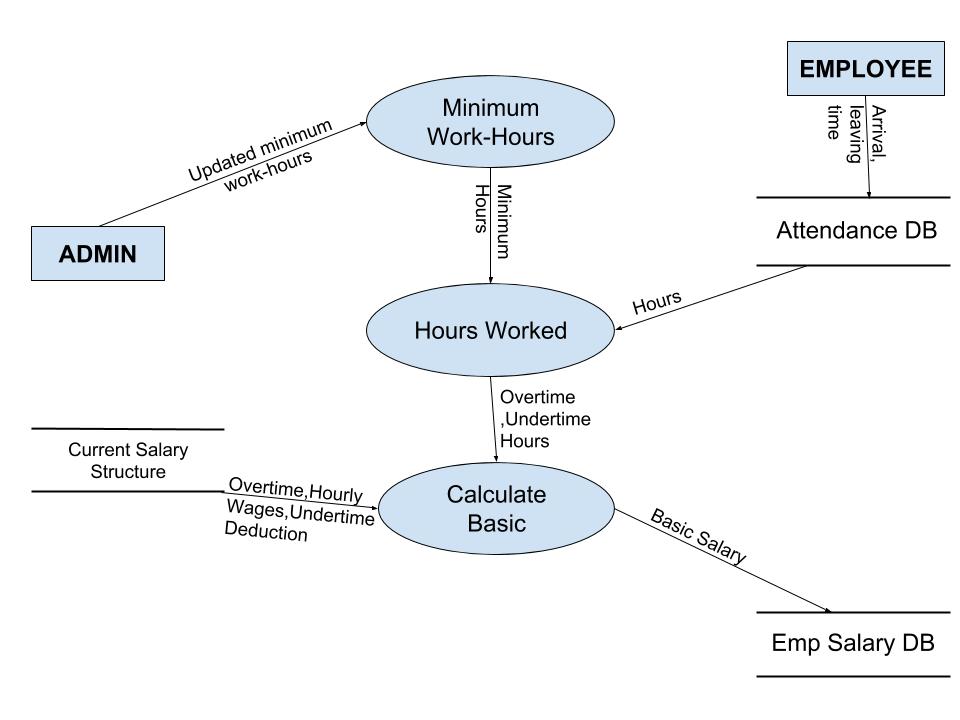
**Fig 3.3. Level 2 (LOGIN)**

****

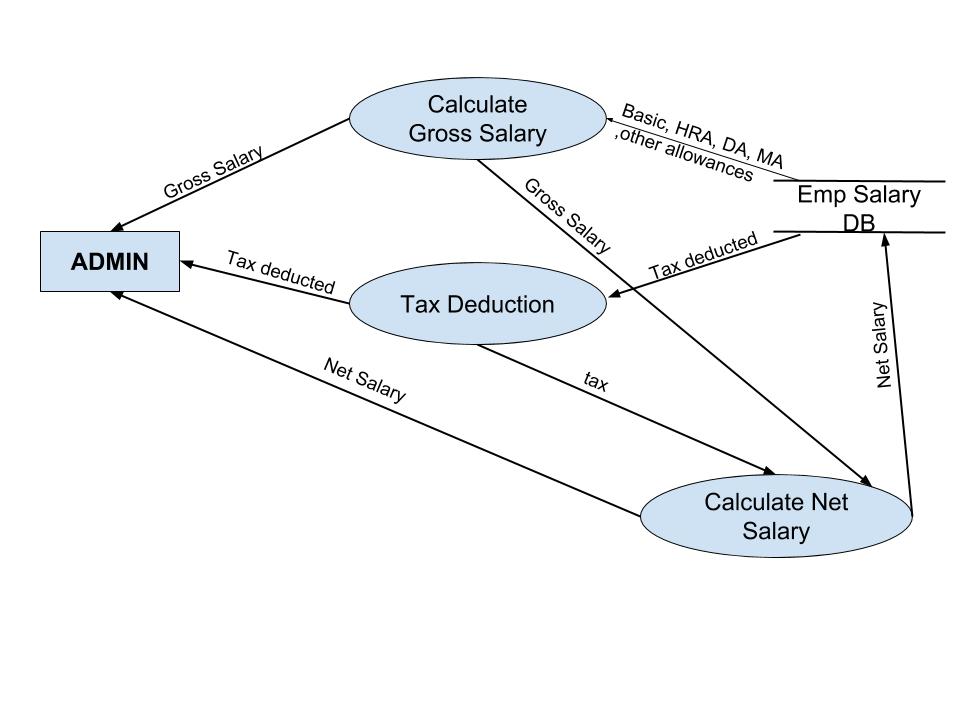
**Fig 3.4.Level 2(Employee registration)**

****

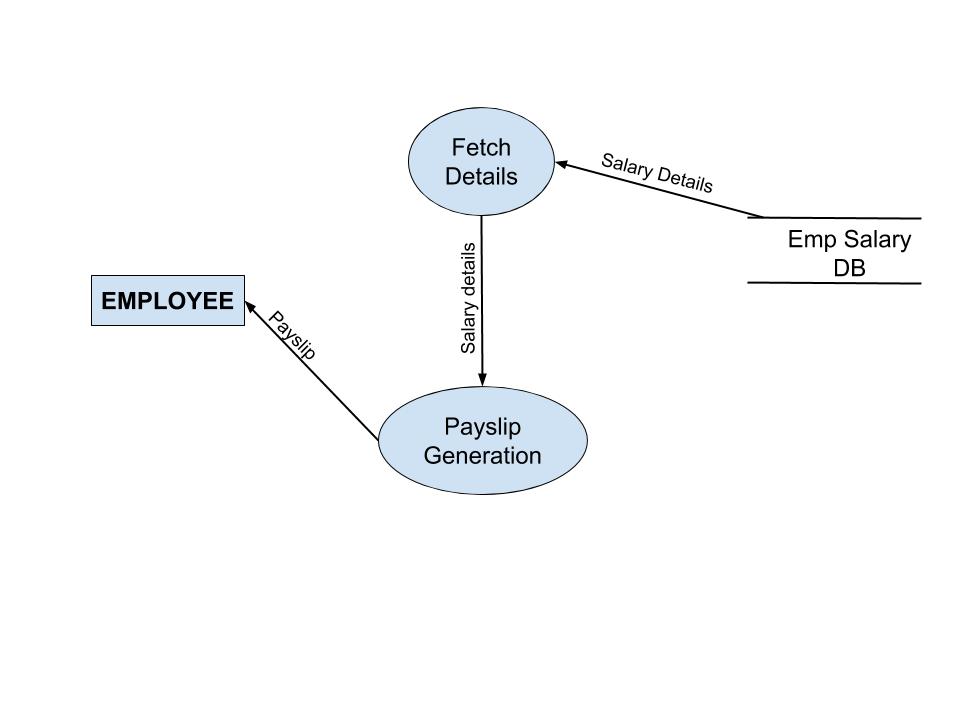
**Fig 3.5.Level 2(Calculate Tax)**

****

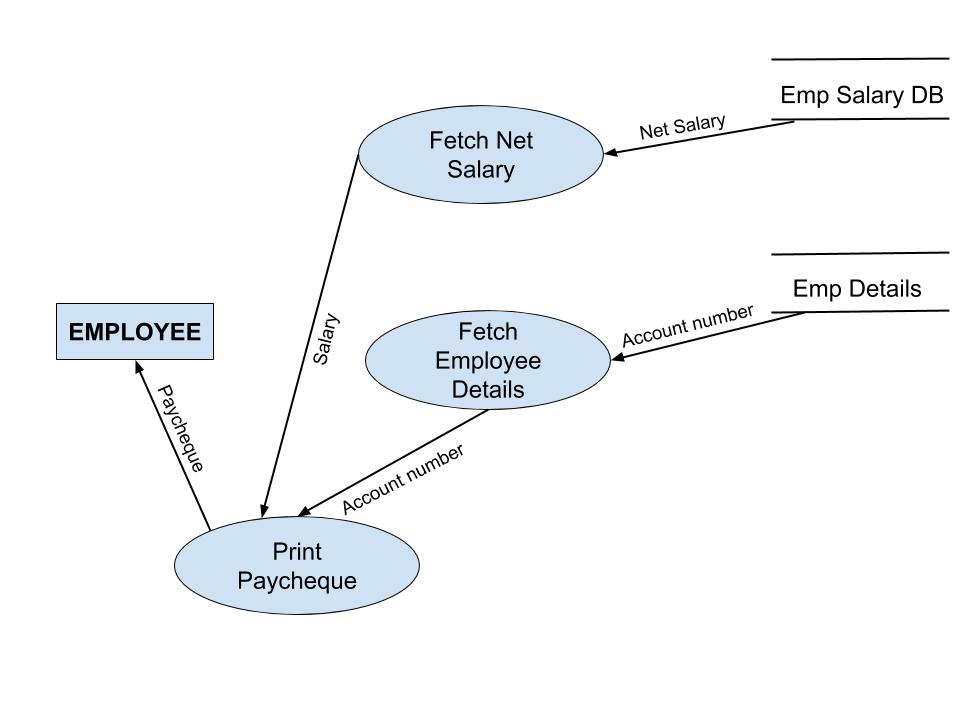
**Fig 3.6.Level 2(Calculate Work-Hours)**

****

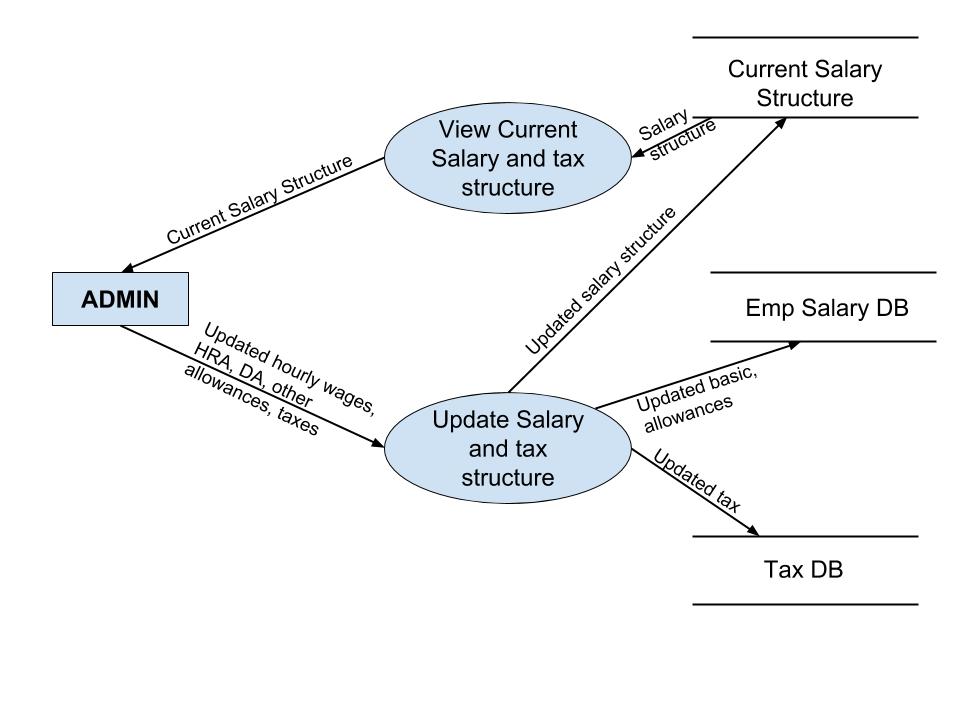
**Fig 3.7.Level 2(Calculate Salary)**

****

**Fig 3.8.Level 2(Generate Pay slip)**

****

**Fig 3.9.Level 2(Issue Pay cheque)**

****

**Fig 3.10.Level 2(Salary Revision)**

* 1. **.Data Dictionary:**

ID = digit + digit + digit + digit + digit

Password = {legal\_character}\*

legal\_character = [A-Z | a-z | \_ | $ | \*]

Ename = FirstName + (MiddleName) + LastName

EId = digit + digit + digit

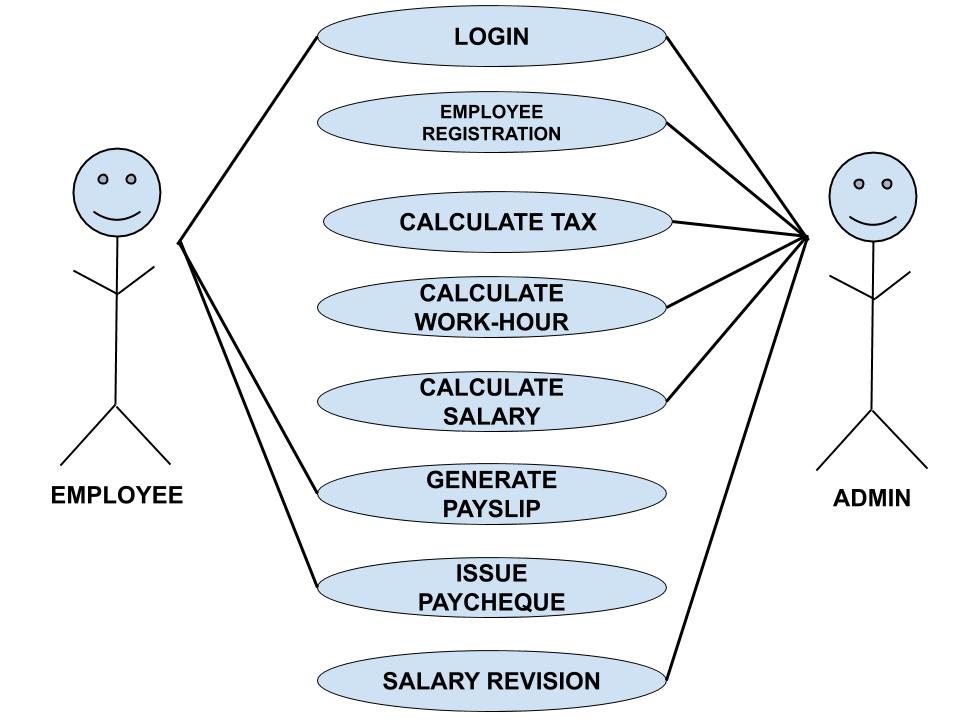
Digit = [0-9]

Employee details = Ename + EId + Email ID

Salary details = Gross salary + tax + net salary

**3.10: Use Cases:**

**3.10.1: Use Case Diagram:**

****

**Fig: 3.11: Use Case Diagram**

**3.10.2: Use Case Description:**

**LOGIN**

* **Brief Description**

           It allows the actors to use the software , Employee Payroll System ,with the help

           of an already assigned unique ID.

* **Actors**

The following actors participates in this use case :

* Admin
* Employee
* **Flow of Events**

**Basic Flow**

This use case starts when the actor wants to sign in.

* The system requests the actor to enter his/her id, password and role (admin or employee).
* The actor enters his/her id, password and role.
* The system verifies whether the entered id matches with the password and role and then allows the actor to login.

**Alternative Flow**

        If the actor enters an invalid id, password and role, then an error message will be displayed asking the user to log into his/her account again or cancel the login  process. If the actor selects the latter option, the use case ends.

* **Special Requirements**

 None

* **Pre-Conditions**

The actors must have an already existing account in the system. In case of no, prior account, actor is not allowed to sign in.

* **Post-Conditions**

If login is successful, actor is logged into the system and redirected to home page. If not, the use case remains unchanged.

If actor has role of the admin, then s/he would have access to Employee registration, Calculate tax, calculate work hour, calculate salary, and salary revision.

If actor has role of an Employee, then s/he would have access to calculate work hour, generate pay slip, and collect pay cheque.

**EMPLOYEE REGISTRATION**

* **Brief Description**

It allows the actor to create a new employee account, so that s/he can avail facilities of the system.

* **Actors**

The following actor participates in this use cas :

* Admin
* **Flow of Events**

**Basic Flow**

This use case starts when a new employee is employed whose account does not exist in the system.

* The system asks for Employee ID, name, email ID, role and salary.
* The actor enters employee’s details.
* The system asks for login details (user ID and password) to be given to employee.
* The actor enters login details.
* The system validates both login and employee’s details.

**Alternative Flow**

          If in the basic flow, the actor does not enter any mandatory details or enters invalid information, the system displays an error message asking the actor to re-enter the information.

* **Special Requirements**
* The Eid and email ID must be unique.
* The password should be strong (must contain at least 8 characters).
* The actor cannot leave any field blank.
* **Pre-Conditions**

None

* **Post-Conditions**

If the register is successful, the employee is able to log into the system. If not, then the use case remains unchanged.

**CALCULATE TAX**

* **Brief Description**

It basically calculates tax to be deducted from basic salary, with the help of details provided by admin.

* **Actors**

The following actor participates in this use case:

* Admin
* **Flow of Events**

**Basic Flow**

This use case starts when tax is to be calculated.

* The system asks for professional tax rate.
* The actor enters professional tax rate.

**Alternative flow**

None

* **Special Requirements**

None

* **Pre-Conditions**

The actor must be logged in as an admin and not employee.

* **Post-Conditions**

None

**CALCULATE WORK-HOUR**

* **Brief Description**

It calculates hours including overtime worked by an employee.

* **Actors**

The following actor participates in this use case :

* Admin
* **Flow of Events**

**Basic Flow**

This use case starts when work-hour is to be calculated.

* The system asks for minimum work-hour.
* The actor enters minimum work-hour.

**Alternative flow**

None

* **Special Requirements**

None

* **Pre-Conditions**

The actor must be logged in as an admin and not employee.

* **Post-Conditions**

None

**CALCULATE NET SALARY**

* **Brief Description**

It provide the actor with information like gross salary ,tax deducted and net salary of an employee.

* **Actors**

The following actor participates in this use case :

* Admin
* **Flow of Events**

**Basic Flow**

This use case starts when actor wants to know the details like  gross salary ,tax deducted and net salary of an employee.

* The system asks for Eid of an employee whose details, an actor wants to view.
* The actor enters the Eid.
* The system shows gross salary, tax deducted, and net salary of that employee.

**Alternative flow**

If actor enters invalid Eid, the use case ends there.

* **Special Requirements**
* The date must be 1st or onwards of a month.
* Net pay should be calculated.
* **Pre-Conditions**

The actor must be logged in as an admin and not employee.

* **Post-Conditions**

None

**GENERATE PAYSLIP**

* **Brief Description**

It generates pay slip of an employee.

* **Actors**

The following actor participates in this use case:

* Employee
* **Flow of Events**

**Basic Flow**

This use case starts when an actor wants to have the pay slip.

* The system asks whether actor wants to view or print the pay slip.
* If print is the option selected, pay slip is printed and collected by an actor.
* If view is the option selected, pay slip is viewed on screen by an actor.

**Alternative flow**

None

* **Special Requirements**
* The date must be 1st or onwards of a month.
* Net pay should be calculated.
* **Pre-Conditions**

The actor must be logged in as an employee and not admin.

* **Post-Conditions**

None

**ISSUE PAYCHEQUE**

* **Brief Description**

It issues pay cheque to an actor.

* **Actors**

The following actor participates in this use case:

* Employee
* Admin
* **Flow of Events**

**Basic Flow**

This use case starts when actor wants to receive a pay cheque or actor wants to know pay cheque received status.

* If the actor is employee, the system issues a pay cheque.
* If the actor is admin, the system shows pay cheque received status.

**Alternative flow**

None

* **Special Requirements**
* The date must be 1st or onwards of a month.
* Net pay should be calculated.
* **Pre-Condition**

None

* **Post-Conditions**

None

**SALARY REVISION**

* **Brief Description**

It lets an actor, modify salary structure and tax structure.

* **Actors**

The following actor participates in this use case:

* Admin
* **Flow of Events**

**Basic Flow**

This use case starts when an actor wants to revise the salary.

* The system shows the actor, current salary structure.
* The system asks for updated hourly wages, HRA, DA, other allowances ,and tax.
* The actor enters the updated details.

**Alternative flow**

None

* **Special Requirements**
* The system should have current salary structure and tax structure available.
* **Pre-Conditions**

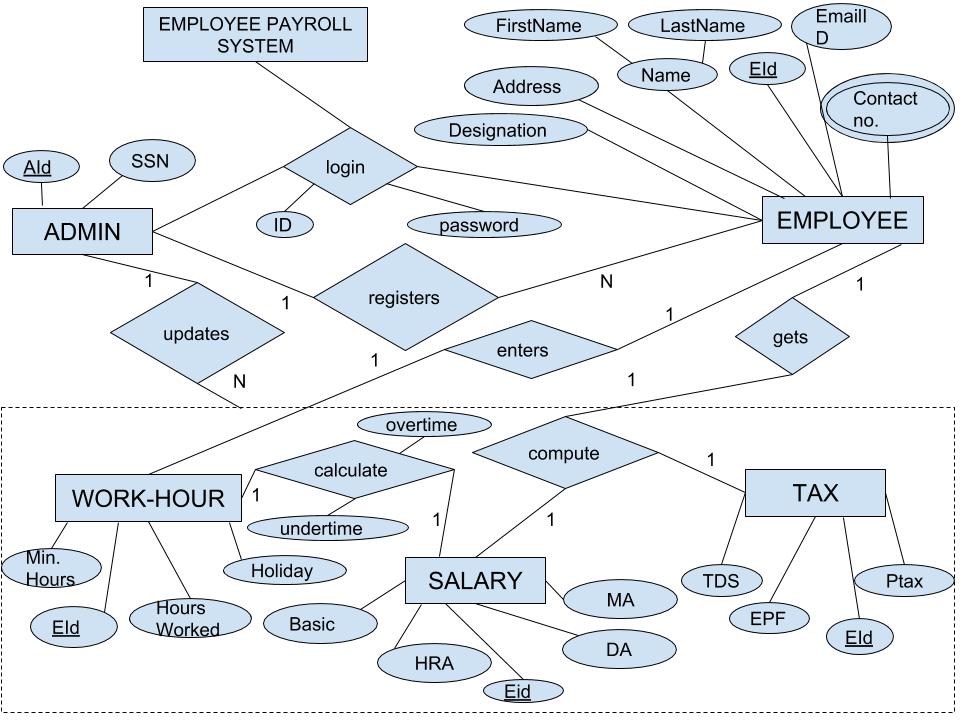
The actor must be logged in as admin and not employee.

* **Post-Conditions**

After updating the details, system should have updated details in its databases.

**CHAPTER 4: DESIGN**

**4.1 ER Diagram:**

****

**Fig 4.1 ER Diagram**

**4.2 Data Design:**

1. Name: ADMIN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Field Name** | **Data Type** | **Field Length** | **Constraints** | **Description** | **Example** |
| 1. | AId | Numeric | 5 | Primary Key | Unique Admin ID | 29045 |
| 2. | SSN | Numeric | 8 | Not Null | System Security Number | 11902378 |

**Table no. 4.1: Data Design(Admin)**

2. Name: EMPLOYEE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Field Name** | **Data Type** | **Field Length** | **Constraints** | **Description** | **Example** |
| 1. | Name | Char | 50 | Not Null | Employee Name | Suresh Kumar |
| 2. | EId | Numeric | 3 | Primary Key | Employee ID | 198 |
| 3. | Contact no. | Numeric | 10 | Not Null | Contact numbers | 9811100000 |
| 4. | Email ID | Char | 50 | Not Null | Employee mail ID | abcd@gmail.com |
| 5. | Address | Char | 50 | Unique | Full address | 11 A, abc society ,Delhi-91 |
| 6. | Designation | Char | 20 | Not Null | Designation assigned | Manager |

**Table no. 4.2: Data Design(Employee)**

3. Name: TAX

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Field Name** | **Data Type** | **Field Length** | **Constraints** | **Description** | **Example** |
| 1. | EId | Numeric | 3 | Primary Key | Employee ID | 239 |
| 2. | TDS | Float | 10 | Not Null | Tax deducted according to slab | 2380 |
| 3. | EPF | Float | 10 | Not Null | Employee PF tax | 235 |
| 4. | Ptax | Float | 10 | Not Null | Professional tax | 540 |

**Table no. 4.3: Data Design(Tax)**

4. Name : Work-Hour

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Field Name** | **Data Type** | **Field Length** | **Constraints** | **Description** | **Example** |
| 1. | EId | Numeric | 3 | Primary Key | Employee ID | 239 |
| 2. | Min.Hours | Numeric | 2 | Not Null | Minimum hours to complete by employee | 6 |
| 3. | Hours Worked | Numeric | 3 | Not Null | Number of hours worked in a month | 30 |
| 4. | Holiday | Numeric | 2 | Not Null | Number of Holidays availed in a month | 3 |

**Table no. 4.4: Data Design(Work Hour)**

5. Name: SALARY

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Field Name** | **Data Type** | **Field Length** | **Constraints** | **Description** | **Example** |
| 1. | EId | Numeric | 3 | Primary Key | Employee ID | 239 |
| 2. | HRA | Float | 5 | Not Null | HRA added on basic | 2300 |
| 3. | DA | Float | 4 | Not Null | DA added on basic | 2378 |
| 4. | MA | Float | 4 | Not Null | MA added on basic | 5678 |
| 5. | Basic | Float | 8 | Not Null | Basic Salary | 30000 |

**Table no. 4.5: Data Design(Salary)**

**4.3 Complex Level Design:**

PSEUDO CODE for employee registration module :

Emp\_reg ( )

1 Enter all the personal details of the employee

2 If (details are valid)

3 Assign login credentials

4 If (login credentials are valid and unique)

5 Registration successful

6 Else

7 Registration unsuccessful

8 Else

9 Registration unsuccessful

10 End function

**CHAPTER 5: ESTIMATION AND SCHEDULING**

**5.1 Project Scheduling:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Work** | **Planned start** | **Actual start** | **Planned completion** | **Actual completion** | **Assigned person** | **Effort allocated** |
| Problem statement | Jan , w1 | Jan, w1 | Jan, w2 | Jan, w2 | Radhika, Aradhana | 2 PW |
| Software lifecycle model | Jan, w2 | Jan, w3 | Jan, w3 | Jan, w3 | Radhika | 1PW |
| Project scheduling | Jan, w4 | Jan, w4 | Jan, w4 | Jan, w4 | Radhika, Aradhana | 2 PW |
| SRS | Jan, w3 | Jan, w3 | Feb, w1 | Feb, w1 | Radhika, Aradhana | 2 PW |
| ER Diagram | Feb, w2 | Feb, w2 | Feb, w2 | Feb, w2 | Radhika, Aradhana | 1 PW |
| Data Dictionary | Feb, w2 | Feb, w2 | Feb, w3 | Feb, w3 | Radhika, Aradhana | 2 PW |
| Level-0 DFD | Feb, w3 | Feb, w4 | Feb, w4 | Feb, w4 | Radhika | 1 PW |
| Level-1 DFD | Feb, w4 | March, w1 | March, w1 | March, w2 | Radhika | 1 PW |
| Level-2 DFD | March, w2 | March, w3 | March, w4 | April, w1 | Radhika, Aradhana | 2 PW |
| Use Case Description | March, w4 | March, w4 | March, w4 | March, w4 | Radhika, Aradhana | 1 PW |
| Function Point Metrics | April, w1 | April, w1 | April, w1 | April, w1 | Aradhana | 1 PW |
| COCOMO II Model | April, w2 | April, w2 | April, w2 | April, w2 | Radhika | 1 PW |
| Risk Analysis | Jan,w4 | Jan ,w4 | April ,w3 | April,w3 | Radhika, Aradhana | 2 PW |
| Testing | April, w2 | April, w3 | April, w3 | April, w3 | Radhika, Aradhana | 2 PW |

**Table 5.1: Project Scheduling Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Work tasks** | **January** | | | | **February** | | | | **March** | | | | **April** | | | |
|  | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 |
| Problem statement |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Software lifecycle model |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SRS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ERD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data Dictionary |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Level-0 DFD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Level-1 DFD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Level-2 DFD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Use Case Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Use Case Description |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Function point metrics |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| COCOMO II Model |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Risk Analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Table 5.2: Timeline Chart**

**5.3 Size Estimation (Function Based Metrics)**

Project metrics are used to control and coordinate software engineering process and to improve quality of the software to be produced. Project specific metrics provide indication of productivity and insight into the technical activities. Project metrics are used by a project manager and a software team to adapt project work flow and technical activities.

**Function Oriented Metrics**

Function oriented metrics use function point as normalization value. Function points are derived using an empirical relationship based on countable (direct) measure of software’s information domain and assessments of software complexity.

Information domain values are defined in the following manner:

* **Number of external inputs (EIs)** - Each external input originates from a user or is transmitted from another application and provides distinct application-oriented data or control information. Inputs are often used to update internal logical files (ILFs). Inputs should be distinguished from inquiries, which are counted separately.
* **Number of external outputs (EOs) -** Each external output is derived data within the application that provides information to the user. In this context external output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.
* **Number of external inquiries (EQs) -** An external inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output (often retrieved from an ILF).
* **Number of internal logical files (ILFs) -** Each internal logical file is a logical grouping of data that resides within the application’s boundary and is maintained via external inputs.
* **Number of external interface files (EIFs)**. - Each external interface file is a logical grouping of data that resides external to the application but provides information that may be of use to the application.

Calculation of Value Adjustment Factor (VAF) based on responses to the following:

|  |  |
| --- | --- |
| **FACTOR** | **VALUE** |
| 1. Does the system require reliable backup and recovery? | 2 |
| 1. Are specialized data communications required to transfer the info to or from the application? | 2 |
| 1. Are there distributed processing functions? | 5 |
| 1. Is performance critical? | 5 |
| 1. Will the system run in an existing heavily utilized operational environment? | 5 |
| 1. Does the system require online data entries? | 4 |
| 1. Does the online data entry require input transaction to be built over multiple screens or operations? | 3 |
| 1. Are ILF’s updated online? | 1 |
| 1. Are input, output, files or enquiry complex? | 2 |
| 1. Is the internal processing complex? | 2 |
| 1. Is the code designed to be reusable? | 3 |
| 1. Are conversion and installation included in the design? | 3 |
| 1. Is the system designed for multiple installations in different organizations? | 0 |
| 1. Is the application designed to facilitate change and ease of use by the user? | 4 |
| Σfi | 45 |

|  |  |  |  |
| --- | --- | --- | --- |
| **MEASUREMENT**  **PARAMETER** | **COUNT** | **WEIGHTING**  **FACTOR** | **WEIGHTING**  **COUNT** |
| EI’s | 50 | 3 4 6 | 150 |
| EO’s | 10 | 4 5 7 | 40 |
| EQ’s | 0 | 3 4 6 | 0 |
| ILF’s | 5 | 7 10 15 | 35 |
| EIF’s | 2 | 5 7 10 | 10 |

**TABLE 5.3 Function Point Complexity Weights**

Count Total = 150 + 40 + 0 + 35 + 10 = 235

Function point = Count Total \* [0.65+{0.01\*Σ(fi)}]

= 235 \* [0.65 + (0.01\*45)]

=235 \* [0.65 + 0.45]= 258.5

**5.4 Effort Estimation (COCOMO II Model)**

Constructive Cost Model (COCOMO)

Constructive cost model is a widely used hierarchy of software estimation models. It addresses the following areas:

1. Application composition model. It was used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount.

2. Early design stage model: Used once requirements have been stabilized and basic software architecture has been established.

3. Post architectural stage model: Used during the construction of the software.

COCOMO II model requires sizing information for which three different sizing options are available as part of the model hierarchy: object points, lines of source code and function points. Like function points, the object point is an indirect software measure that is computed using counts of the number of

1 .Screens (at user interface),

2. Reports, and

3. Components likely to be required to build the application

|  |  |  |  |
| --- | --- | --- | --- |
| **OBJECT TYPE** | **COMPLEXITIY WEIGHT** | | |
| **SIMPLE** | **MEDIUM** | **DIFFICULT** |
| **SCREEN** | 1 | 2 | 3 |
| **REPORT** | 2 | 5 | 8 |
| **3GL COMPONENT** |  |  | 10 |

**TABLE 5.4 COCOMO II Complexity Weights**

Each object instance (e.g., a screen or report) is classified into one of three complexity levels (i.e., simple, medium, or difficult) using criteria suggested by Boehm. In essence, complexity is a function of the number and source of the client and server data that are required to generate the screen or report and the number of views or sections presented as part of the screen or report. The object point count then determined by multiplying the original number of object instances by the weighting factor in the figure and summing to obtain a total object point count. When general software reuse is to be applied, the percent of reuse (%reuse) is estimated and the object point is adjusted:

NOP = (object points) \* [(100 - %reuse)/100]

Where, NOP = new object points

To derive an estimate of effort based on the computed NOP value, a “productivity rate” must be derived.

After productivity rate determination, an estimate of project effort is computes using,

ESTIMATE EFFORT = NOP/PROD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Developer’s experience/capability | Very Low | Low | Normal | High |
| Environment maturity/capability | Very Low | Low | Normal | High |
| PROD | 4 | 7 | 13 | 25 |

**1. SCREENS**

Total Screens = 19

**2. REPORTS**

Total Reports = 2

**3. 3 GL COMPONENTS**

Total 3 GL Components = 0

Considering all of the above have simple complexity, 0% of components are reused and taking the developer experience and environment maturity as normal.

**PRODUCTIVITY RATE=**7 (Low)

**OBJECT POINT =** {19 \* 2} + {2 \* 5} = 38 + 10 = 48

**ESTIMATED EFFORT =** = 48/7 = 6.857 Person-Month.

**CHAPTER 6: RISK ANALYSIS**

**Risk Identification:**

Product Size: The customer’s system should have the capability to fulfill the system’s size.

Business Impact: No risk is associated with constraints imposed by management.

Stakeholder Characteristics: No risk associated with the sophistication of the stakeholder and developer’s ability to communicate with the stakeholder in timely meetings.

Process Definition: The process is well defined and is followed by development organization.

Development Environment: No risk associated with the availability and quality of the tools to be used to build the product.

Staff Size and Experience: The staff size is appropriate for the project and all the members are very experienced in their own field. They all are comfortable with the technology.

|  |  |  |  |
| --- | --- | --- | --- |
| **RISK** | **CATEGORY** | **PROBABILITY** | **IMPACT** |
| 1. Size estimation may be significantly low | PS | 40% | 2 |
| 2. Larger number of users then planned | PS | 60% | 3 |
| 3. Less reuse then planned | PS | 30% | 2 |
| 4.End user resist system | BU | 40% | 3 |
| 5.Delivery date line will be tightened | BU | 60% | 2 |
| 6.Funding will be lost | CU | 30% | 1 |
| 7.Technology will not meet expectations | TE | 5% | 1 |
| 8.Staff turnover will be high | ST | 50% | 2 |

**Table no. 6.1: Risk impact**

Impact values:

1. Catastrophic

2. Critical

3. Marginal

4. Negligible

**Assessing Overall Project Risk:**

1. Have top software and customers, managers formally committed to support the project?

YES

2. Are end users enthusiastically committed to the project and the system/product to be built?

YES

3. Are requirements fully understood by software engineering team and its customers?

YES

4. Have customers been involved fully in the definition of requirements?

YES

5. Do end users have realistic expectations?

YES

6. Is the project’s scope stable?

YES

7. Does the software engineering team have the right mix of skills?

YES

8. Are project requirements stable?

YES

9. Does the project team have experience with the technology to be implemented?

YES

10. Is the number of people on the project team adequate to do job?

YES

11. Do all the customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

YES

**CHAPTER 7: TESTING**

**White box testing:**

White box testing, sometimes called glass box testing is a test case design philosophy that uses the control structure described as a part of component level design to derive test cases. Using white box testing technique we can derive test cases that:

1. Guarantee that all the independent paths within a module have been exercised at least once.

2. Exercise all logical decisions on their true and false sides.

3. Execute all loops at their boundaries and within their operational bounds and

4. Exercise internal data structures to ensure their validity.

**7.1 Control Flow Graph:**

* COMPLEX LEVEL DESIGN:

Emp\_reg ( )

1 Enter all the personal details of the employee

2 If (details are valid)

3 Assign login credentials

4 If (login credentials are valid and unique)

5 Registration successful

6 Else

7 Registration unsuccessful

8 Else

9 Registration unsuccessful

10 End function

* GRAPH:

1

2

8

3

**R3**

4

9

**R1**

6

5

**R2**

7

10

**Fig 7.1 Control Flow Graph**

**7.2 Cyclomatic Complexity**

There are 3 methods to find cyclomatic complexity using control flow graph-

1. V (G) = No. of regions =3

2. V (G) = (No. of predicate nodes) + 1= 2 +1 = 3

3. V (G) = Edges – Nodes + 2 = 11 – 10 + 2 = 3

**7.3Basis Path Set**

The basis set consists of all possible paths during the execution of the code. The no. of paths = V (G). The basis set of this graph is:

1. 1---2---3---4---5---10

2. 1---2---3---6---7---10

3. 1---2---8---9---10

**ANNEXURE (SAMPLE SCREENS)**

**LOGIN SCREEN**

**EMPLOYEE PAYROLL SYSTEM**

**!!!WELCOME USER!!!**

**USER Id:**

**PASSWORD:**

**LOGIN**

LOGIN UNSUCCESSFUL SCREEN

**SCREEN NO. 1(LOGIN)**

**LOGIN UNSUCCESSFUL SCREEN**





**!!!WELCOME USER!!!**

**LOGIN UNSUCCESSFUL**

**INVALID ID OR PASSWORD**

**TRY AGAIN WITH VALID LOGIN DETAILS**

**SCREEN NO.2 (LOGIN UNSUCCESSFUL)**

**HOMEPAGE FOR ADMIN**

**ENTER WORK HOURS**

**UPDATE TAX DETAILS**

**VIEW SALARY/TAX DETAILS**

**UPDATE SALARY STRUCTURE**

**REGISTER AN EMPLOYEE**



**!!!WELCOME ADMIN!!!**

**CHOOSE AN ACTION TO PROCEED:**

**SCREEN NO. 3(HOMEPAGE FOR ADMIN)**

**HOME PAGE FOR EMPLOYEE**



**!!!WELCOME EMPLOYEE!!!**

**CHOOSE AN ACTION TO PROCEED:**

**COLLECT PAY CHEQUE**

**GET PAY SLIP**

**BACK**

**LOGOUT**

**SCREEN NO. 4(HOME PAGE FOR EMPLOYEE)**

**GET PAY SLIP OPTION FOR EMPLOYEE**



**!!!WELCOME EMPLOYEE!!!**

**GET PAY SLIP:**

**ENTER EId:**

**PRINTPAY SLIP**

**VIEW PAY SLIP**

**SCREEN NO. 5(GET PAY SLIP OPTION FOR EMPLOYEE)**

**GET PAY CHEQUE OPTION SCREEN**



**!!!WELCOME EMPLOYEE!!!**

**GET PAY CHEQUE:**

**ENTER EId:**

**PRINT PAY CHEQUE**

**SCREEN NO. 6(GET PAY CHEQUE OPTION SCREEN)**

**REGISTER AN EMPLOYEE SCREEN FOR ADMIN**



**!!!WELCOME ADMIN!!!**

**REGISTER AN EMPLOYEE:**

**ENTER PERSONAL DETAILS:**

* **ENAME: **
* **ADDRESS: **
* **PHONE NO: **
* **EMAIL Id: **
* **POST: **
* **DEPARTMENT:**
* **GRADE: **

**SUBMIT**

**SCREEN NO. 7(REGISTER AN EMPLOYEE SCREEN FOR ADMIN)**

**ASSIGNING LOGIN CREDENTIALS SCREEN**



**!!!WELCOME ADMIN!!!**

**REGISTER AN EMPLOYEE:**

**ASSIGN LOGIN CREDENTIALS:**

* **EId: **
* **PASSWORD:**

**SUBMIT**

**SCREEN NO. 8(ASSIGNING LOGIN CREDENTIALS SCREEN)**

**EMPLOYEE REGISTERATION UNSUCCESSFUL SCREEN**



**!!!WELCOME ADMIN!!!**

**REGISTER AN EMPLOYEE:**

**COULD NOT REGISTER EMPLOYEE**

**TRY AGAIN WITH VALID CREDENTIALS**

**SCREEN NO. 9(EMPLOYEE REGISTRATION UNSUCCESSFUL)**

**EMPLOYEE REGISTERED SUCCESSFULLY SCREEN**



**!!!WELCOME ADMIN!!!**

**REGISTER AN EMPLOYEE:**

**EMPLOYEE REGISTERED SUCCESSFULLY**

**SCREEN NO. 10(EMPLOYEE REGISTRATION SUCCESSFUL SCREEN)**

**UPDATE SALARY STRUCTURE SCREEN**



**!!!WELCOME ADMIN!!!**

**UPDATE SALARY STRUCTURE:**

**ENTER CREDENTIALS OF THE EMPLOYEE**

* **EId: **
* **POST: **
* **DEPARTMENT:**
* **GRADE: **

****

**SCREEN NO. 11(UPDATE SALARY STRUCTURE SCREEN)**

**SALARY UPDATION SCREEN**



**!!!WELCOME ADMIN!!!**

**UPDATE SALARY STRUCTURE:**

**ENTER THE UPDATES OF SALARY:**

* **BASIC: **
* **DA: **
* **HRA: **
* **OTHER ALLOWANCES:**
* **GRP INSURANCE:**
* **GRADE: **

**SAVE CHANGES**

**SCREEN NO. 12(SALARY UPDATION SCREEN)**

**SALARY UPDATION SUCCESSFUL**



**!!!WELCOME ADMIN!!!**

**UPDATE SALARY STRUCTURE:**

**SALARY STRUCTURE HAS BEEN UPDATED**

**SCREEN NO. 13(SALARY UPDATION SUCCESSFUL SCREEN)**

**VIEW SALARY OR TAX DETAILS SCREEN**



**!!!WELCOME ADMIN!!!**

**VIEW SALARY OR TAX DETAILS:**

**ENTER CREDENTIALS OF THE EMPLOYEE**

* **EId: **
* **POST: **
* **DEPARTMENT:**
* **GRADE: **

**VIEW TAX**

**VIEW SALARY**

**SCREEN NO. 14(VIEW SALARY OR TAX DETAILS SCREEN)**

**VIEWING SALARY STRUCTURE SCREEN**



**!!!WELCOME ADMIN!!!**

**VIEW SALARY STRUCTURE:**

* **GROSS PAY: **
* **NET PAY: **

**SCREEN NO. 7**

**SCREEN NO. 15(VIEWING SALARY STRUCTURE SCREEN)**

**VIEWING TAX DETAILS SCREEN**



**!!!WELCOME ADMIN!!!**

**VIEW TAX DETAIL:**

**TDS: **

**EPF: **

**P-TAX: **

**SCREEN NO. 16(VIEWING TAX DETAILS SCREEN)**

**UPDATING TAX DETAILS SCREEN**



**!!!WELCOME ADMIN!!!**

**UPDATE TAX DETAIL:**

**ENTER THE UPDATED VALUES:**

**P-TAX %AGE:** 

**TDS %AGE:** 

**EPF %AGE:**

**APPLY CHANGES**

**SCREEN NO. 17(UPDATING TAX DETAILS SCREEN)**

**TAX UPDATION SUCCESSFUL SCREEN**



**!!!WELCOME ADMIN!!!**

**UPDATE TAX DETAIL:**

**TAX DETAILS HAVE BEEN UPDATED**

**SCREEN NO. 18(TAX UPDATION SUCCESSFUL SCREEN)**

**ENTER WORKING HOURS SCREEN**



**!!!WELCOME ADMIN!!!**

**ENTER MINIMUM WORK HOURS:**

**HOURS: **

****

**SCREEN NO. 19(ENTER WORKING HOURS SCREEN)**

**CONCLUSION**

Employee Payroll System is software to ease salary payment process. This Report is a compilation of requirement process, design process, risk analysis, testing process analysis to achieve efficient product. When used, it will make salary payment process timelier, accurate, and easy as compared to manually performed. Tax structure, salary structure, and minimum work hours as may get revised by government or company after some time, is updated in the system and salary is calculated according to that, then.

Pay slip and Pay cheque is given to employees through a transparent process.

As more requirements or errors may come up in the future, prototype model is used in this system which makes the modifications later on in the system feasible. And so the limitations discovered later on can be over comed.

And more features can be added to make it more easy and efficient.

**ADVANTAGES**

This software i.e. Employee Payroll System makes the salary payment process in offices easy for employer as well as employee in following ways:

* System automates the process of pay cheque and pay slip printing.
* It calculates the salary details of each and every employee accurately.
* Employer doesn’t have to keep a check of employee’s payment details.
* Pay cheque received status is notified to him/her.
* All the necessary details of an employee are safely placed in the databases.
* Security breach is controlled by this system using personal login details.
* Employer (Admin) can easily update tax or salary structure or minimum work hours.

**LIMITATIONS**

This software has following limitations:

* It has only cheque as the option of salary payment way.
* Tax and Salary revised by government is updated by Admin in the system. There is no way of automatically updating these details.

**BIBLIOGRAPHY**

**Books:**

1. P. Jalote, An integrated approach to software Engineering, Narosa Publishing House, Ed.3, 2011

2. R.S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill, Ed 7, 2010