**Project**



**SDA (Software Design and Architecture)**

**BSSE- 4 B (Morning)**

**Submit By:**

* **Maheen Nasir Abbasi(12315)**
* **Sundas Javed(12321)**
* **Soma nasir(12322)**
* **Anusha Rubbab Malik(12309)**

**Submit To:**

**Maam Sadia Sabir**

**Date: 03/01/2022**

**Introduction:**

**Objective:**

The task is to build a salary management system. Current salary system is manual therefore the organization wants to switch to an automated computerized salary management system. After building this system we have to integrate it with the existing computerized system. The existing system is dealing with the client registrations, keeping records of clients, client billing etc.

**Project Goals:**

The system should be capable of performing following functions:

* Store basic information regarding employees of the organization.
* Store salary information of employees (entered by the team leaders in each city) such as, working hours, salary per hour, salary before tax, tax percentage, total amount of tax paid, salary after tax, social security fee, on monthly basis.
* System should be able to generate the following salary information:

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

* Social security fee depends upon employee age. Hence, percentage on the salaries would be variable.
* System should be able to generate the bank files, tax files and employee salary slips automatically.
* Salary slips can be sent to the employees upon request.

**System Requirements:**

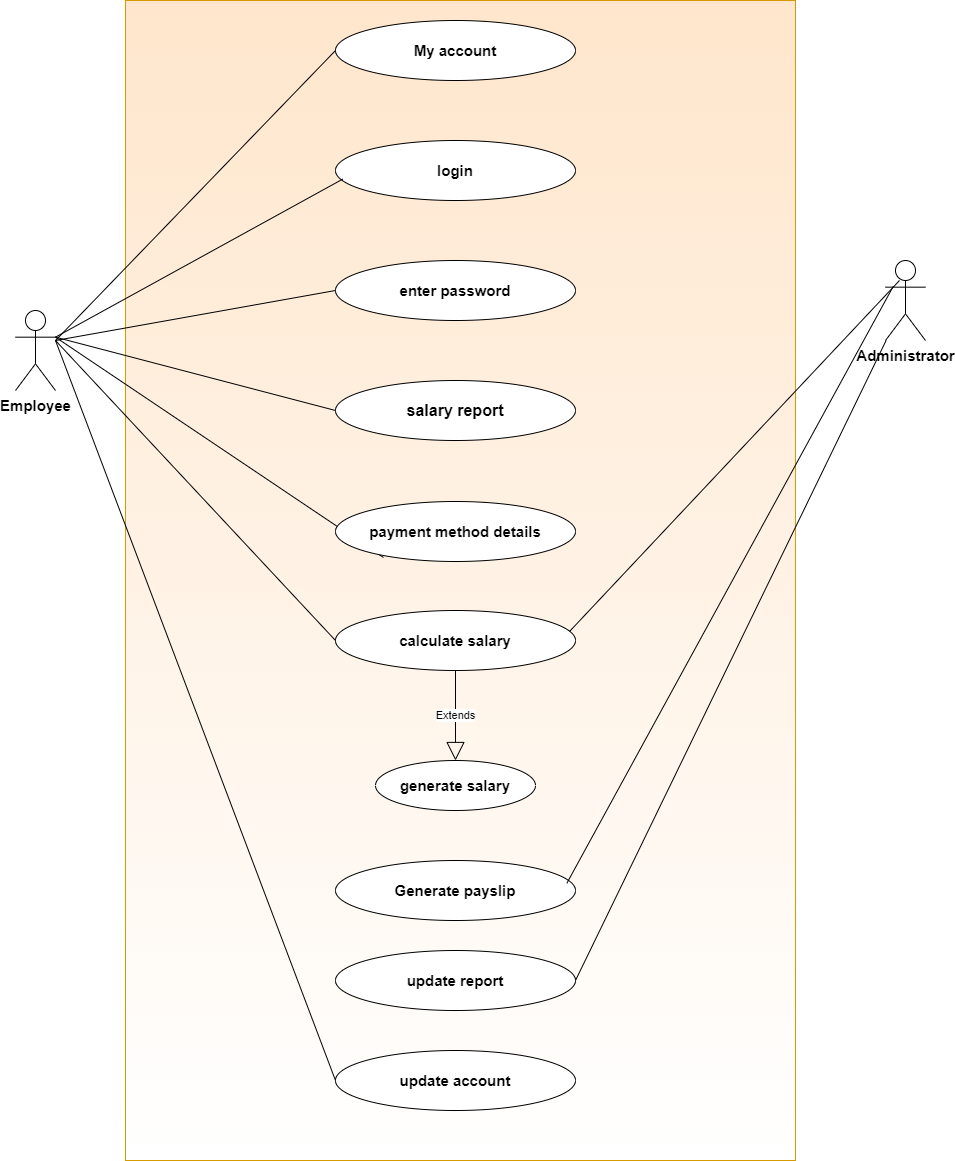
Requirement analysis is an important phase of the system development cycle which provides us all the specification of system in detail which are very essential to build the system and also provides us knowledge about the behavior of the system. Collection of system requirements is a very critical point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

* Functional Requirements
* Non Functional Requirements

**Functional Requirements:**Functional Requirements specifically defines functionalities of the system, behavior of the system and the goals to achieve it. Functions that describe the behavior of the system are considered as behavior requirements and will be shown in the form of use cases

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

**System Use case:**In this section we will discuss the use case of payroll management system. Uses cases show how the users interact with the system. There are two actors in our system, employee, and administrator. Each one has its own responsibilities and level of access to the system.



**Descriptive Use Case:**

In this section we will discuss the descriptive use case of payroll management system. Descriptive Uses cases show how the users interact with the system in descriptive form with all previous actors.

**Login:**

|  |  |
| --- | --- |
| **UC-01** | **SMS/21/001** |
| **Goal** | **Employee wants to login to the system** |
| **Actor** | **Employee** |
| **Pre-requisite/Pre-condition** | **Employee should have an account** |
| **Description** | **User Action System response** |
|  | |  |  | | --- | --- | | * **Open Application** * **Provide Username And password** | * **Display User interface of Application** * **Verify Username And password** * **Login User to the system** * **Display contents of the accounts** | |
| **Alternate** | * **Wrong username or password** |
| **Post Condition** | * **Employee successfully logged in to the system** |

**Calculate Salary:**

|  |  |
| --- | --- |
| **UC-01** | **SMS/21/003** |
| **Goal** | **Employee wants to perform salary calculation** |
| **Actor** | **Employee** |
| **Pre-requisite/Pre-condition** | **Employee should be logged in to the system** |
| **Description** | **User Action System response** |
|  | |  |  | | --- | --- | | * **Tap**   **“calculate salary”** | * **Display salary calculator** | | * **Enter E\_ID** | * **A procedure calls in response and gathers the required information about the user from the database, adding it to the salary calculator.** | | * **Enter salary per month, of tax and of social security** | * **Calculate tax and social security and deduct it from monthly salary** * **Display salary** | |
| **Alternate** | * **Employee enter wrong E\_ID** * **Employee enter negative values** |
| **Post Condition** | * **Employee has calculated his salary** |

**Fully Dressed Use Cases:**

* **Login**

|  |  |
| --- | --- |
| **Use Case Section** | **Comment** |
| **Use Case Name** | **Login** |
| **Brief Description** | **The user will be able to login the Salary management System.** |
| **Scope** | **Salary management System is very efficient with login id and password.**  **The authorized will only have access to online system.** |
| **Primary Actors** | **User (employee),(administrator)** |
| **Secondary Actors** | **Salary Management System** |
| **Pre-Condition**  **(Main Success Scenario)** | **Employee should have an account** |
| **Post Condition** | **Login password has been validated, so the user can continue with system** |
| **Success Guarantee** | **System login successfully** |
| **Extension** | **Login failed** |
| **Level** | **User Goal** |
| **Special Requirements** | **The System shall provide password protection for all accounts** |

* **Calculate Salary:**

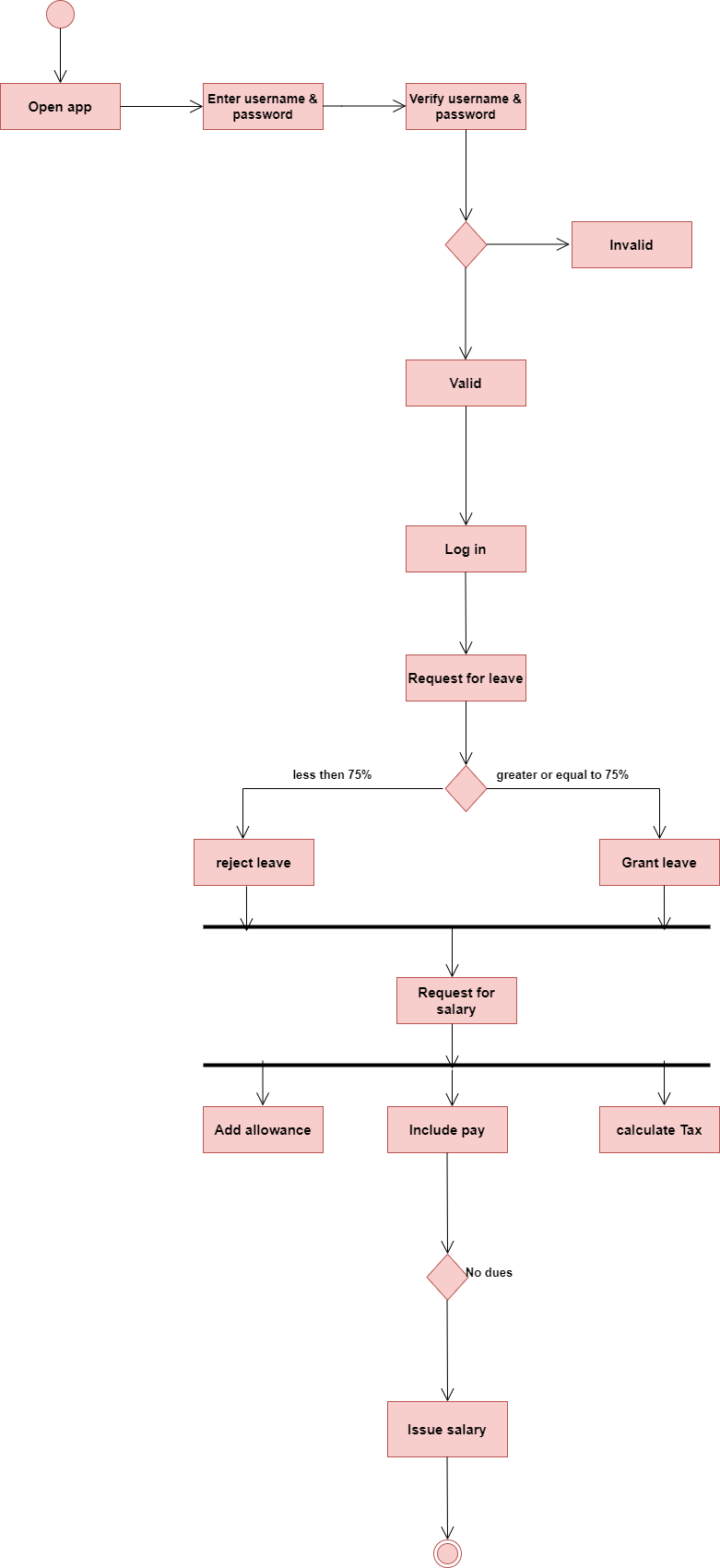
|  |  |
| --- | --- |
| **Use Case Section** | **Comment** |
| **Use Case Name** | **Calculate salary** |
| **Brief Description** | **Employee wants to perform salary calculation** |
| **Scope** | **Salary management System is very efficient with login id and password.** |
| **Primary Actors** | **User (employee),(administrator)** |
| **Secondary Actors** | **Salary Management System** |
| **Pre-Condition**  **(Main Success Scenario)** | **Employee should be logged in to the system** |
| **Post Condition** | **Employee has calculated his salary** |
| **Success Guarantee** | **System login successfully** |
| **Extension** | * **Employee enter wrong E\_ID** * **Employee enter negative values** |
| **Level** | **User Goal** |

* **Update Account:**

|  |  |
| --- | --- |
| **Use Case Section** | **Comment** |
| **Use Case Name** | **Update Account** |
| **Brief Description** | **Employee wants to update his account information** |
| **Scope** | **Salary management System is very efficient system**  **The payroll management system is the core system for any organization.** |
| **Primary Actors** | **User (employee),(administrator)** |
| **Secondary Actors** | **Salary Management System** |
| **Pre-Condition**  **(Main Success Scenario)** | **Employee should have an account** |
| **Post Condition (Success Guarantee)** | **Employee information updated** |
| **Extension** | * **Employee enter wrong E\_ID** * **Employee makes an invalid updating** |
| **Level** | **User Goal** |

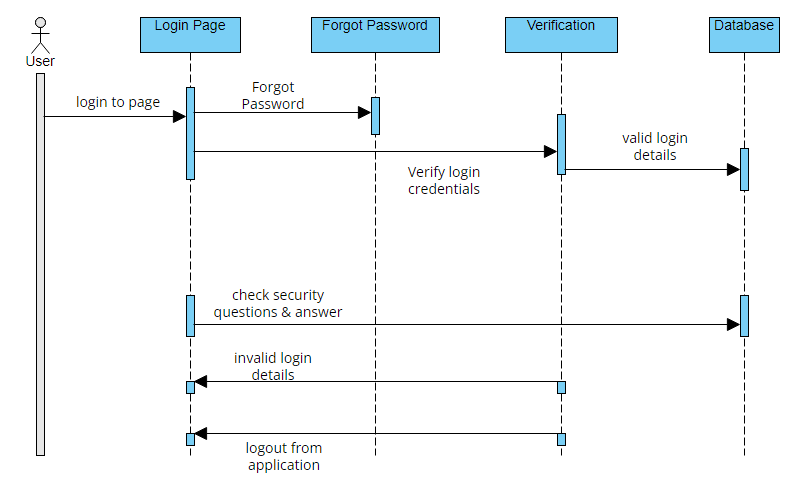
**Activity Diagram:**

In this section we will discuss the activity diagram generated from pre designed use case of payroll management system. Activity Diagram elaborates how use cases carried out in a flow is.

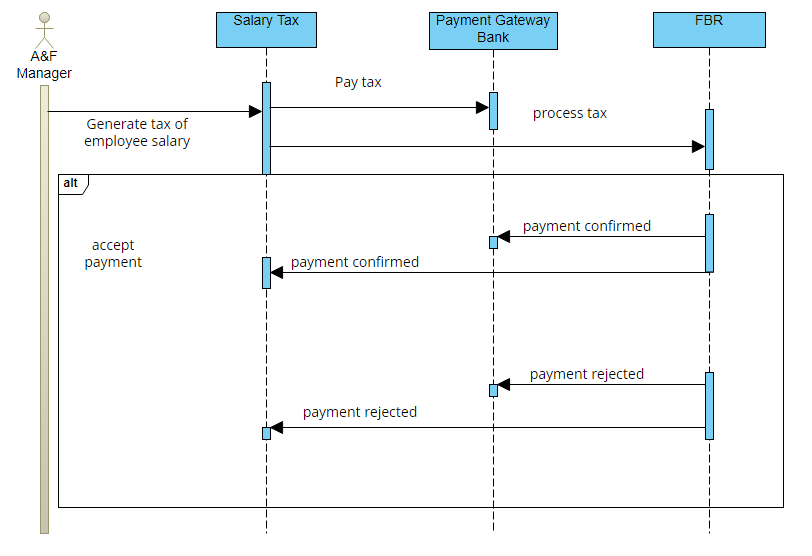


**Sequence Diagram:** A sequence diagram or system sequence diagram shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**Login:**

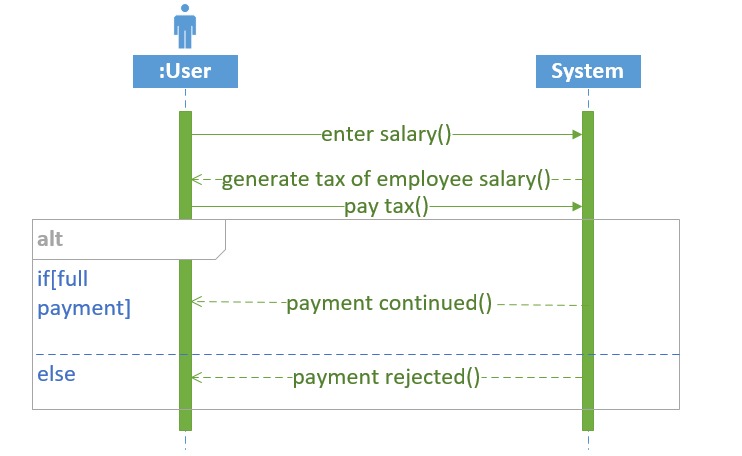


**Tax deposit:**

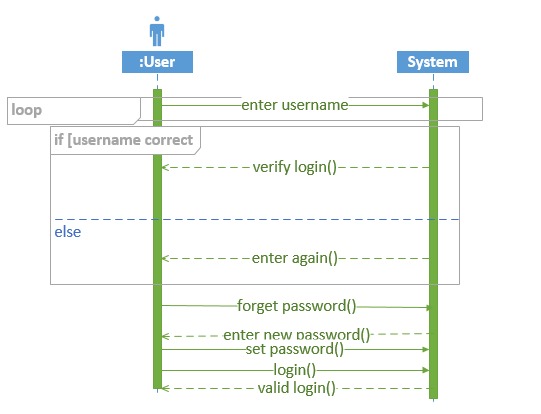


**System Sequence Diagram:**   
 A system sequence diagram shows the interaction between an actor and the system for one particular use case scenario

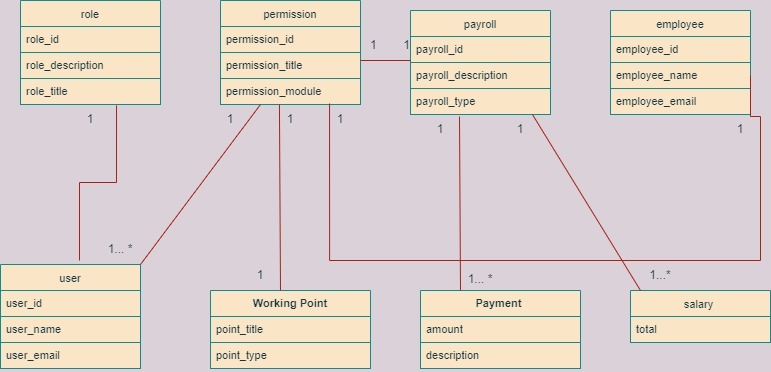
It is a fast and easily created artifact that illustrates input and output events related to the systems under discussion



**Create profile:**

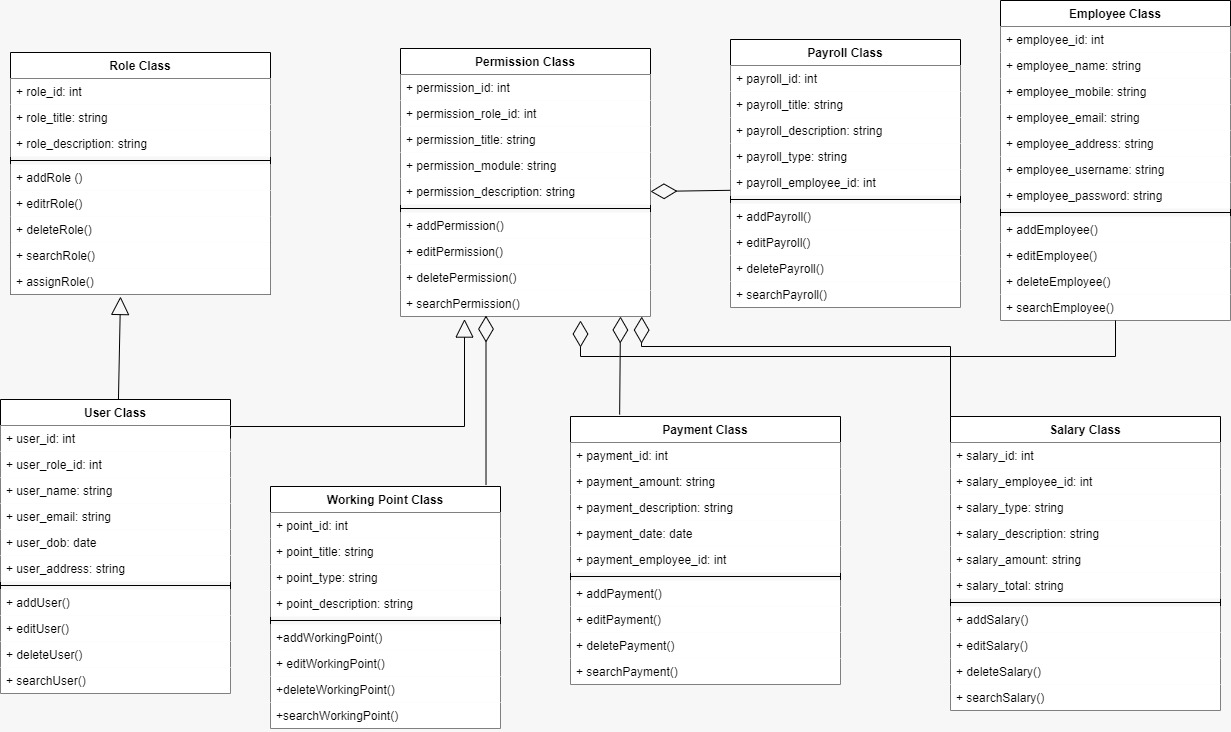


**Domain Model:**



**Class Diagram:**

In this section we will discuss class diagram for system. Class diagram overall divides system into multiple classes dependent upon properties they possess. Furthermore, it also assists to implement classes in program.



**State Machine Diagram:**

In this section we will discuss state machine diagram for system. State Machine Diagram explains states of objects drawn from Classes.

