PROJECT REPORT

ONLINE EMPLOYEE MANAGEMENT SYSTEM

BONDALAPATI BHAVISHYA & SHAIK SUHELA

(RA2111026010153) & (RA2111026010161)

18CSC202J-OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY

II YEAR/III SEMESTER

Academic Year: 2022 - 2023

AIM: To design an object oriented model for online employee management system using Star UML Software.

PROJECT DESCRIPTION:

Employee management system is a system developed with an aim to solve the problem faced by organizations while calculating salary of each employee. This system aims to maintain proper automatic attendance so that no cheating in attendance can be done by any one. This system makes sure that all the important calculations should be done to calculate salary properly.

EMS promotes automatic use and ensures employees that their salary would be calculated properly without any cheating. Proper holidays, week offs and festival holidays granted by organizations are kept for each employee. It provides flexibility of choosing different days off for different employees and accordingly salary would be calculated at the end of month, because of this feature salary may vary according to week days off given to employees if varying week offs. Develop a system that helps an organization in handling their employees. The system should provide a facility to maintain attendance and at the end of the month should give the calculated salary, leaves and working hours of their employees. The main task of this application is to record daily attendance and give the calculated salary, present days leaves taken and working hours. At the end of each month the HR team used to invest a lot of time on this task[8], because of which many times employees don't get their salary on time, which results in destroying the employee and employers relation, which directly affects the growth of Company.

The challenge to this project is developing a system to make this process smooth, better, reliable and simple.

Employee Management System Class Diagram:

Employee Management System Class Diagram describes the structure of a Employee Management System classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the Employee Management System are Employee, Salary, Leaves, Experience, Login, Attendance.

Classes of Employee Management System Class Diagram:

- Employee Class: Manage all the operations of Employee
- Salary Class: Manage all the operations of Salary
- Leaves Class: Manage all the operations of Leaves
- Experience Class: Manage all the operations of Experience
- Login Class: Manage all the operations of Login
- Attendance Class: Manage all the operations of Attendance

Classes and their attributes of Employee Management System Class Diagram:

Employee Attributes: employee_id, employee_name, employee_mobile, employee_email, employee_username, employee_password, employee_address

Salary Attributes: salary_id, salary_employee_id, salary_amount, salary_total, salary_type, salary_description

Leaves Attributes: leave_id, leave_employee_id, leave_type, leave_status, leave_to, leave_from, leave_description

Experience Attributes: experience_id, experience_employee_id, experience_year, experience_type, experience_description

Login Attributes: login_id, login_user_id, login_role_id, login_username, login_password, login_lastlogin

Attendance Attributes: attendance_id, attendance_employee_id, attendance_student_id, attendance_type, attendance_description

Classes and their methods of Employee Management System Class Diagram:

Employee Methods: addEmployee(), editEmployee(), deleteEmployee(), updateEmployee(), saveEmployee(), searchEmployee()

Salary Methods: addSalary(), editSalary(), deleteSalary(), updateSalary(), saveSalary(), searchSalary()

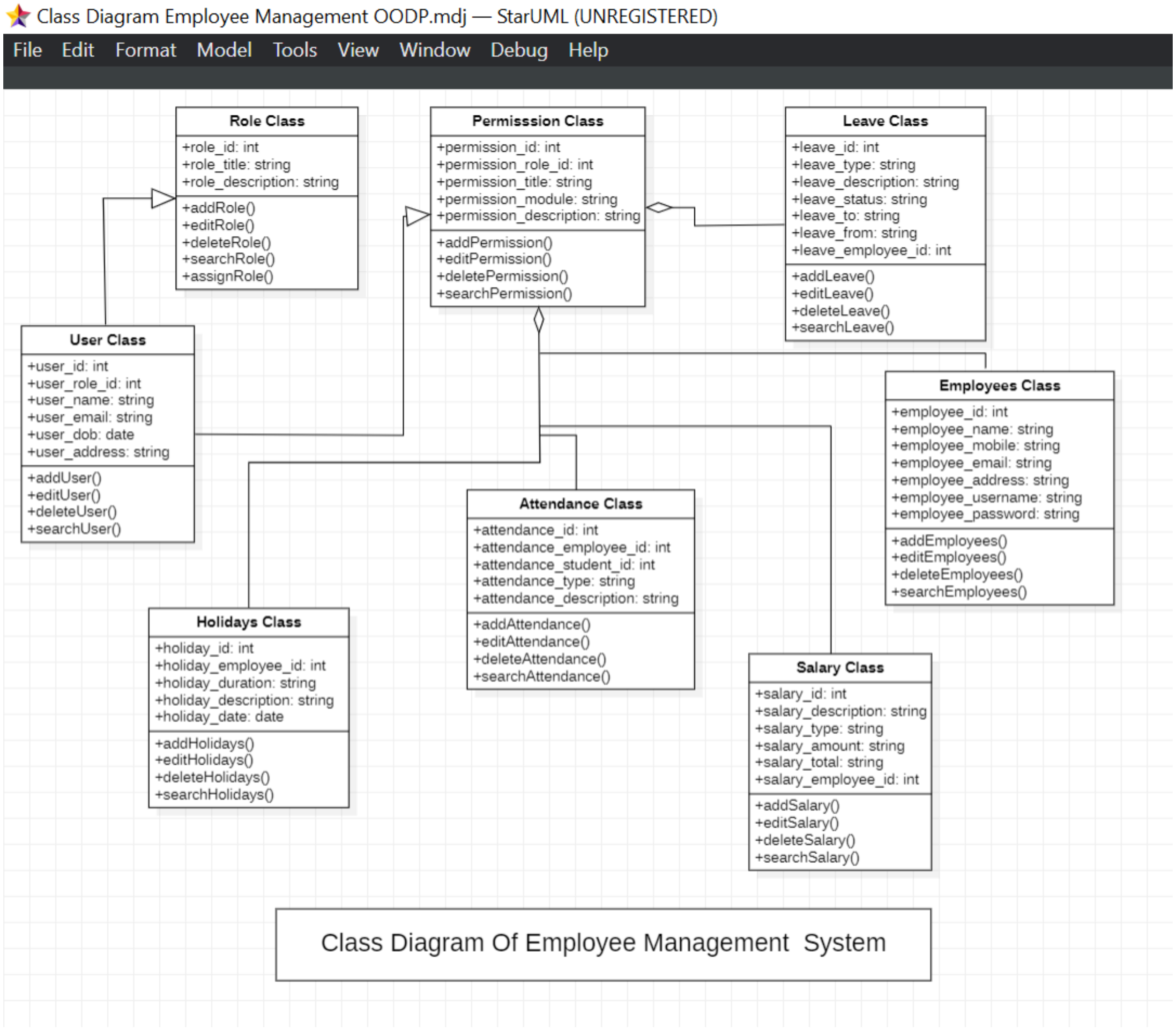
Leaves Methods: addLeaves(), editLeaves(), deleteLeaves(), updateLeaves(), saveLeaves(), searchLeaves()

Experience Methods: addExperience(), editExperience(), delete Experience(), updateExperience(), saveExperience(), searchExperience()

Login Methods: addLogin(), editLogin(), deleteLogin(), updateLogin(), saveLogin(), searchLogin()

Attendance Methods: addAttendance(), editAttendance(), deleteAttendance(), updateAttendance(), saveAttendance(), searchAttendance()

Class Diagram of Employee Management System:



CODE GENERATED:

```
/**

 * Project Untitled

 */

#include "Attendance Class.h"

/**

 * Attendance Class implementation
 */
void Attendance Class::addAttendance() {
}
void Attendance Class::editAttendance() {
}
void Attendance Class::deleteAttendance() {
}
void Attendance Class::searchAttendance() {
}
/*** Project Untitled*/
#include "Leave Class.h"
/*** Leave Class implementation*/
void Leave Class::addLeave() {
}
void Leave Class::editLeave() {
}
void Leave Class::deleteLeave() {
}
void Leave Class::searchLeave() {
}
/**
 * Project Untitled
 */
#include "Employees Class.h"
/**
 * Employees Class implementation
 */
void Employees Class::addEmployees() {
}
void Employees Class::editEmployees() {
}
void Employees Class::deleteEmployees() {
}
void Employees Class::searchEmployees() {
}
/**
 * Project Untitled
 */
#include "Permisssion Class.h"


/**
 * Permisssion Class implementation
 */
void Permisssion Class::addPermission() {
}
void Permisssion Class::editPermission() {
}
void Permisssion Class::deletePermission() {
}
void Permisssion Class::searchPermission() {
}
/**
 * Project Untitled
 */
#include "Salary Class.h"
/**
```

```
 * Salary Class implementation
 */
void Salary Class::addSalary() {
}
void Salary Class::editSalary() {
}
void Salary Class::deleteSalary() {
}
void Salary Class::searchSalary() {
}
/**
 * Project Untitled
 */
#include "User Class.h"
/**
 * User Class implementation
 */
void User Class::addUser() {
}
void User Class::editUser() {
}
void User Class::deleteUser() {
}
void User Class::searchUser() {
}
/**
 * Project Untitled
 */
```

```
#include "Holidays Class.h"
/**
 * Holidays Class implementation
 */
void Holidays Class::addHolidays() {
}
void Holidays Class::editHolidays() {
}
void Holidays Class::deleteHolidays() {
}
void Holidays Class::searchHolidays() {
}
/**
 * Project Untitled
 */
#include "Role Class.h"
/**
 * Role Class implementation
 */
void Role Class::addRole() {
}
void Role Class::editRole() {

}
void Role Class::deleteRole() {
}
void Role Class::searchRole() {
}
```

```
void Role Class::assignRole() {
}
```

Employee Management System Use Case Diagram:

This Use Case Diagram is a graphic depiction of the interactions among the elements of Employee Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Employee Management System. The main actors of Employee Management System in this Use Case Diagram are: Super Admin, System User, Employee, Manages, who perform the different type of use cases such as Manage Employee, Manage Salary, Manage Leaves, Manage Experience, Manage Login, Manage Users and Full Employee Management System Operations. Major elements of the UML use case diagram of Employee Management System are shown on the picture below.

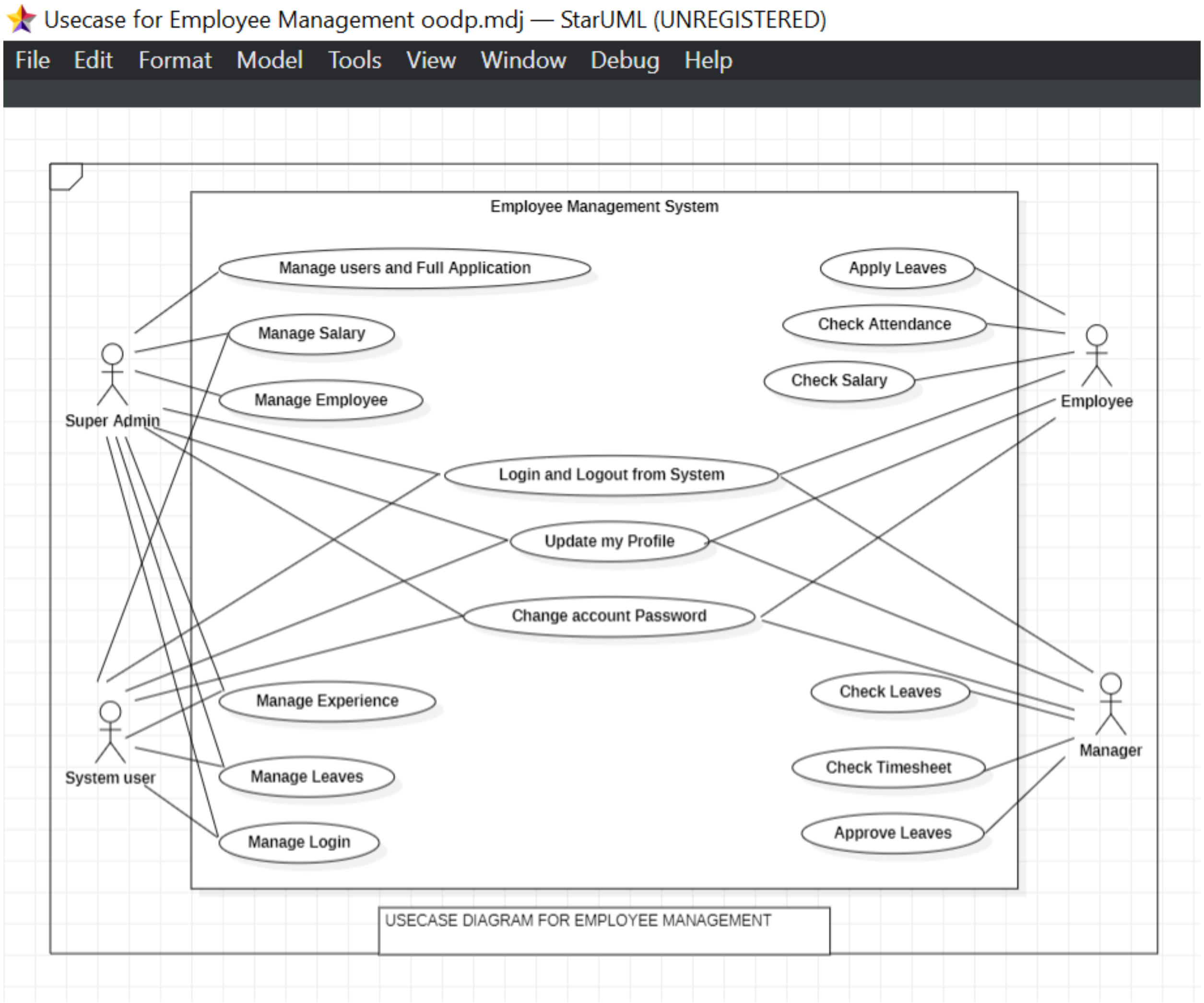
The relationships between and among the actors and the use cases of Employee Management System: Super Admin Entity : Use cases of Super Admin are Manage Employee, Manage Salary, Manage Leaves, Manage Experience, Manage Login, Manage Users and Full Employee Management System Operations

System User Entity : Use cases of System User are Manage Employee, Manage Salary, Manage Leaves, Manage Experience, Manage Login

Employee Entity : Use cases of Employee are Apply Leaves, Check Salary, Check Attandance

Manages Entity : Use cases of Manages are Check Leaves, Approve Leaves, Check Timesheet, Approve Timesheet

Use Case Diagram of Employee Management System :



Employee Management System Sequence Diagram:

This is the UML sequence diagram of Employee Management System which shows the interaction between the objects of Employee, Experience, Salary, , Leaves.

The instance of class objects involved in this UML Sequence Diagram of Employee Management System are as follows:

Employee Object

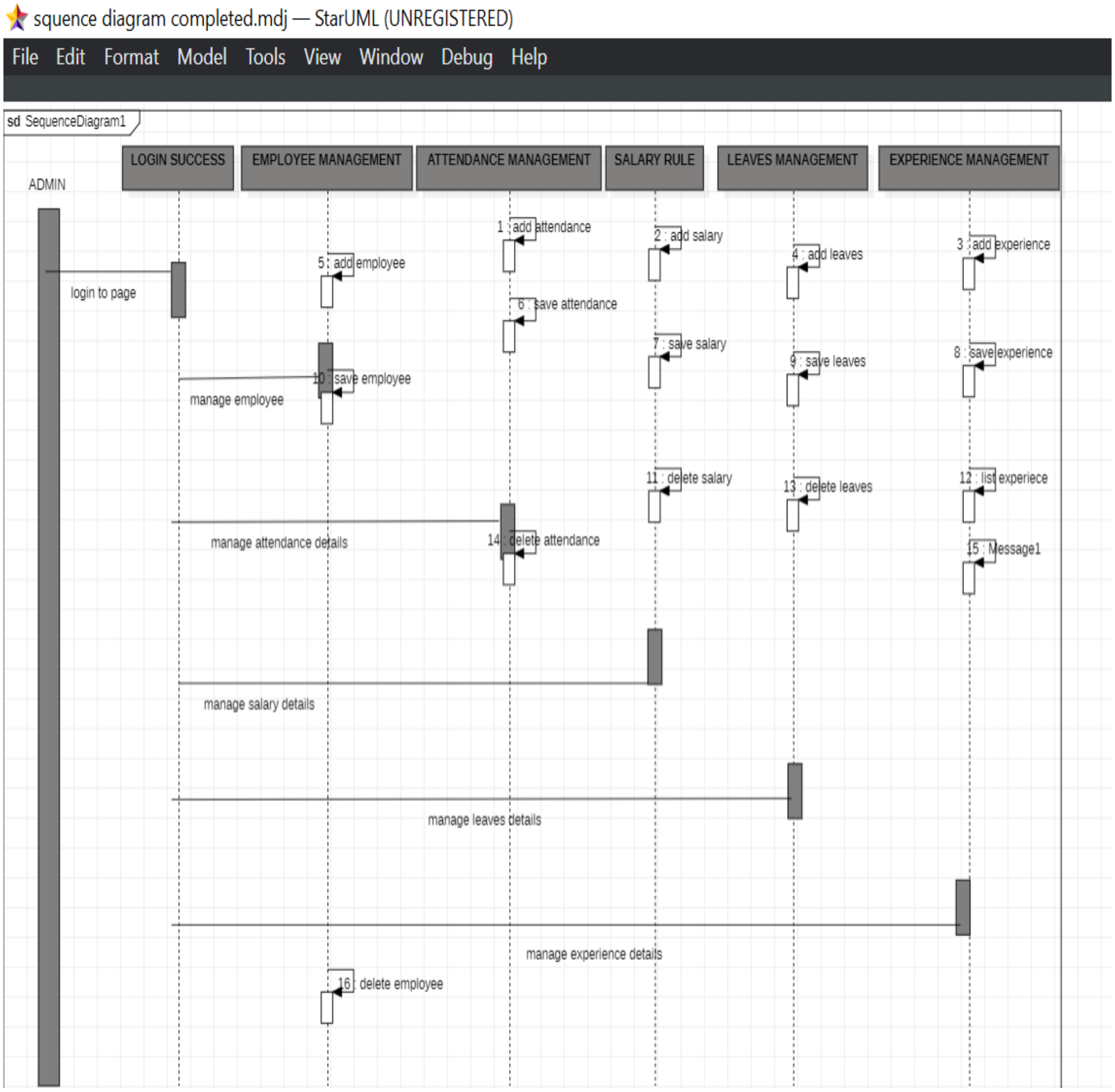
Experience Object

Salary Object

Object

Leaves Object

Sequence Diagram Of Employee Management System:



Employee Management System Communication Diagram:

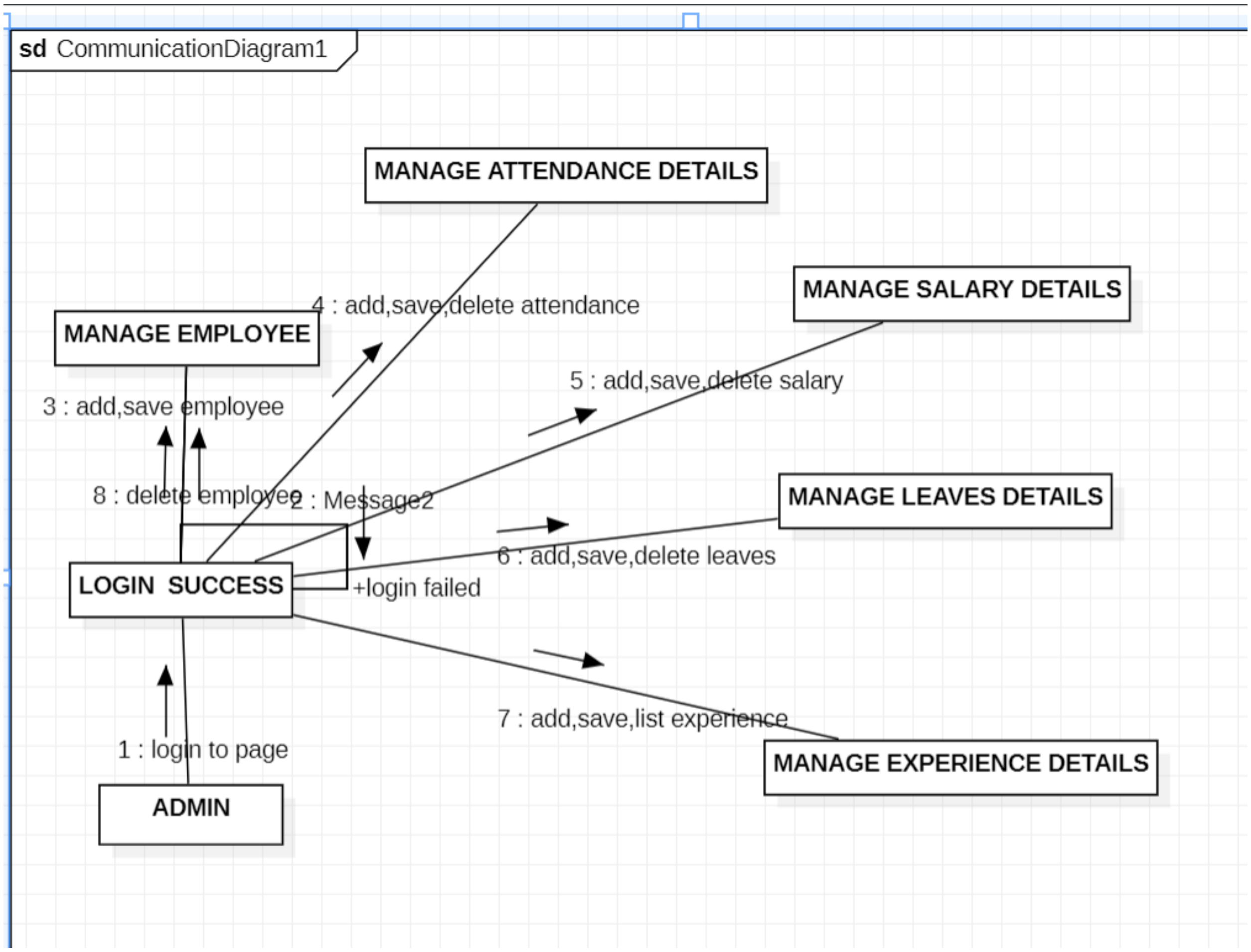
A Communication diagram models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.

UML communication diagrams, like the sequence diagrams - a kind of interaction diagram, shows how objects interact. A communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, communication diagram shows the messages the objects send each other.

Purpose of Communication Diagram

- Model message passing between objects or roles that deliver the functionalities of use cases and operations
- Model mechanisms within the architectural design of the system
- Capture interactions that show the passed messages between objects and roles within the collaboration scenario
- Model alternative scenarios within use cases or operations that involve the collaboration of different objects and interactions
- Support the identification of objects (hence classes), and their attributes (parameters of message) and operations (messages) that participate in use cases.

Communication Diagram:



Employee Management System Package Diagram:

The structure of organization units is important for the internal view of the business system. In UML, organization units are depicted as packages, which can contain employees, business objects, and other organization units.

Organization units can be responsible for the execution of business-process activities. Organization units are abstractions of individual jobs within an organization.

In UML an organization unit spans workers, business objects, other organization units, and their relationships. As a basic principle, organization units are located within business systems. Organization units that are located outside business systems are actors.

In package diagrams we work with the following elements:

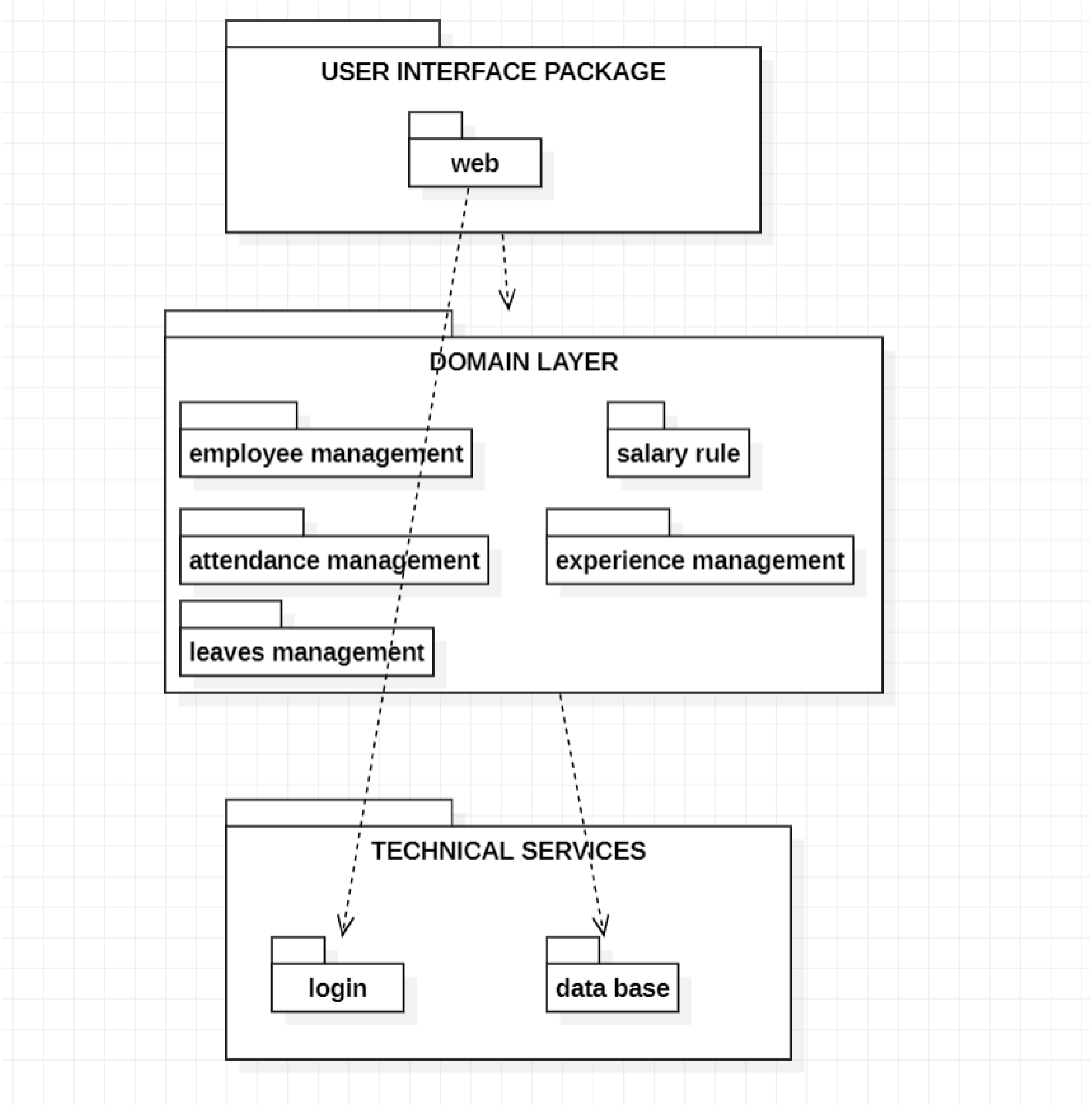
Package «Organization Unit»

Class «Worker»

«Business Object»

Reading Package Diagrams

Package Diagram:



UML Deployment Diagram for Employee Management System:

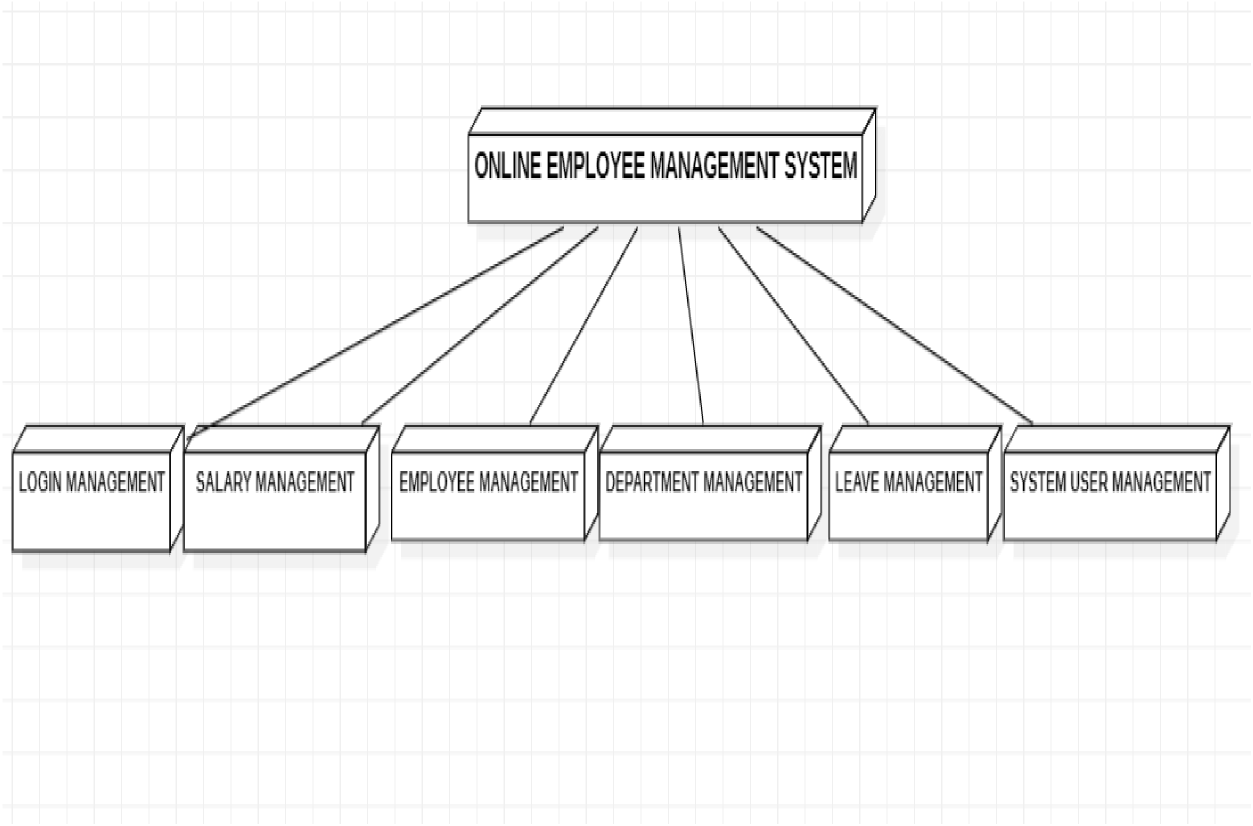
A deployment diagram for employee system in UML is used to illustrate its' physical architecture. In UML, deployment diagrams can show you how the software and hardware of the learning system work together and where the processing takes place.

The employee management system uses a UML deployment diagram to show how should the developed software be deployed. It clarifies the communications between links(nodes) which helps the project to work according to the design given to it. Deployment diagrams depict the setup of run-time processing nodes and the components that reside on them.

The designed deployment diagram for employee system shows the components (nodes) included to carry out the process. Nodes are represented by boxes that are labeled as software or hardware that specify the included components to carry out the employee management process. The boxes will then be connected and labeled to declare the type of connection they have with the other components.

This deployment diagram design helps in the system development. This leads to scenarios where companies gain insights into their employees using an employee management system allowing them to better plan and manage work hours, lowering labor expenses and increasing productivity. This strategy will assist employees in doing their best work each day in order to meet the organization's wider goals.

Deployment Diagram for Employee Management System:



Employee Management System Activity Diagram:

This is the Activity UML diagram of Employee Management System which shows the flows between the activity of Employee, Experience, , Salary, Leaves.

The main activity involved in this UML Activity Diagram of Employee Management System are as follows:

Employee Activity

Experience Activity

Activity

Salary Activity

Leaves Activity

Features Of The Activity UML Diagram Of Employee Management System

→ Admin User can search Employee, view description of a selected Employee, add Employee, update Employee and delete Employee.

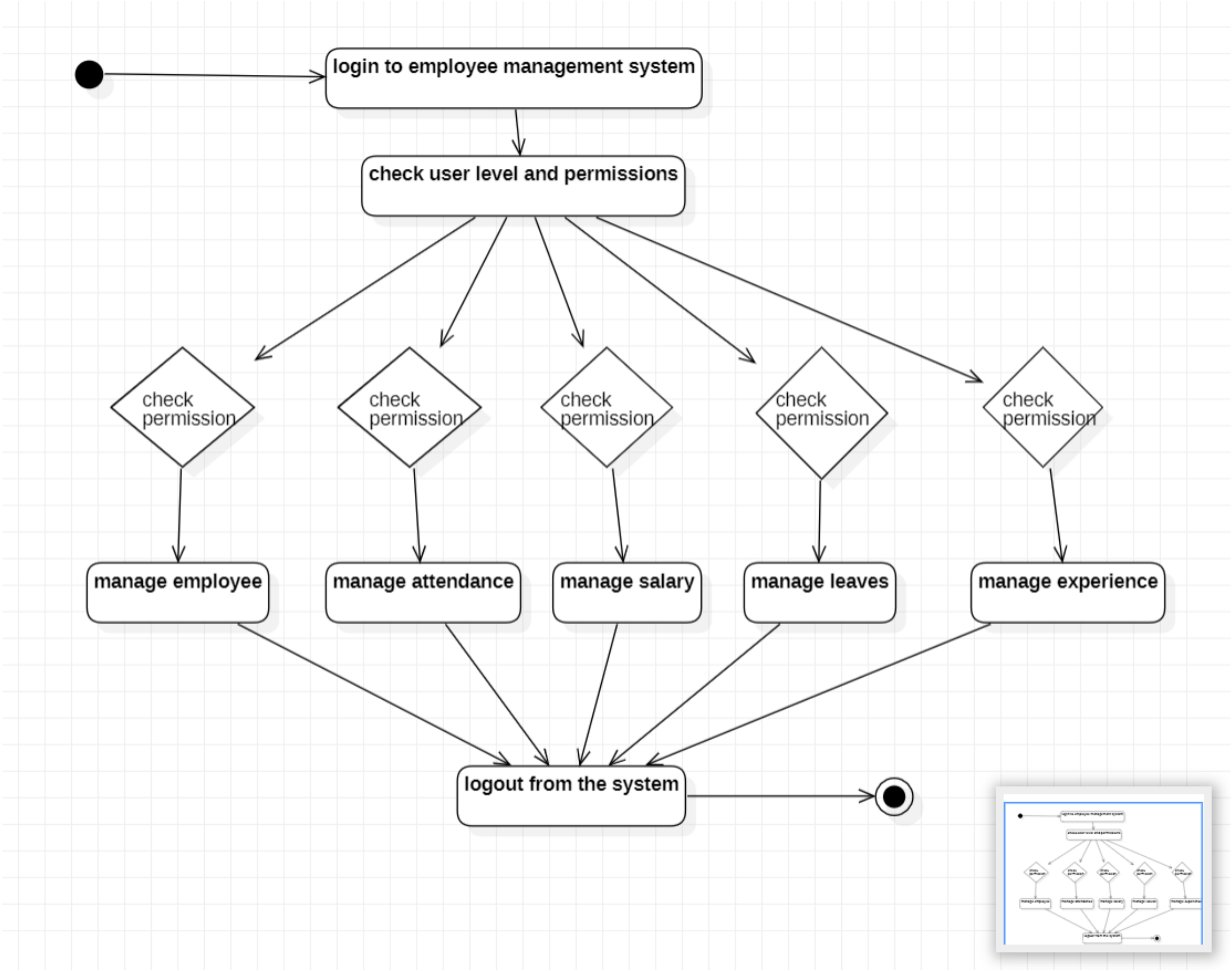
→Its shows the activity flow of editing, adding and updating of Experience

→User will be able to search and generate report of , Salary, Leaves

–>All objects such as (Employee, Experience, Leaves) are interlinked

→Its shows the full description and flow of Employee, Salary, Leaves&Experience.

Activity Diagram:



Employee Management System Component Diagram:

This is a Component diagram of Employee Management System which shows components, provided and required interfaces, ports, and relationships between the , Salary, Employee, Leaves and Experience. This type of diagrams is used in Component-Based Development (CBD) to describe systems with Service-Oriented Architecture (SOA). Employee Management System UML component diagram, describes the organization and wiring of the physical components in a system.

Components of UML Component Diagram of Employee Management System:

Component

Salary Component

Employee Component

Leaves Component

Experience Component

Features of Employee Management System Component Diagram:

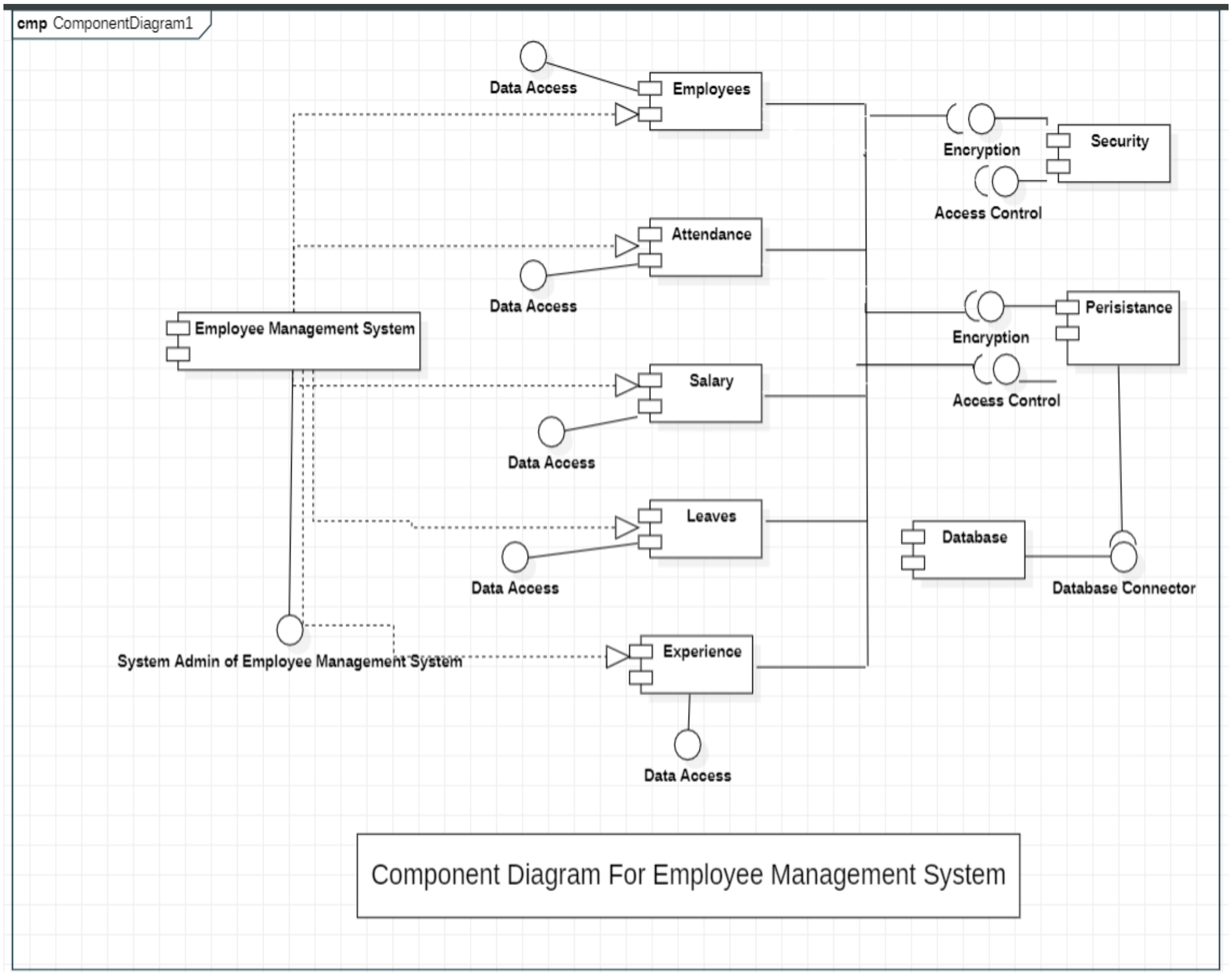
→You can show the models the components of Employee Management System.

→Model the database schema of Employee Management System

→Model the executables of an application of Employee Management System

→Model the system's source code of Employee Management System

Component diagram:



Employee Management System State Chart Diagram:

A state diagram shows the behavior of classes in response to external stimuli. Specifically a state diagram describes the behavior of a single object in response to a series of events in a system. Sometimes it's also known as a Harel state chart or a state machine diagram. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.

Notation of a State Machine Diagram

Following are the notations of a state machine diagram enlisted below:

Initial state: It defines the initial state (beginning) of a system, and it is represented by a black filled circle.

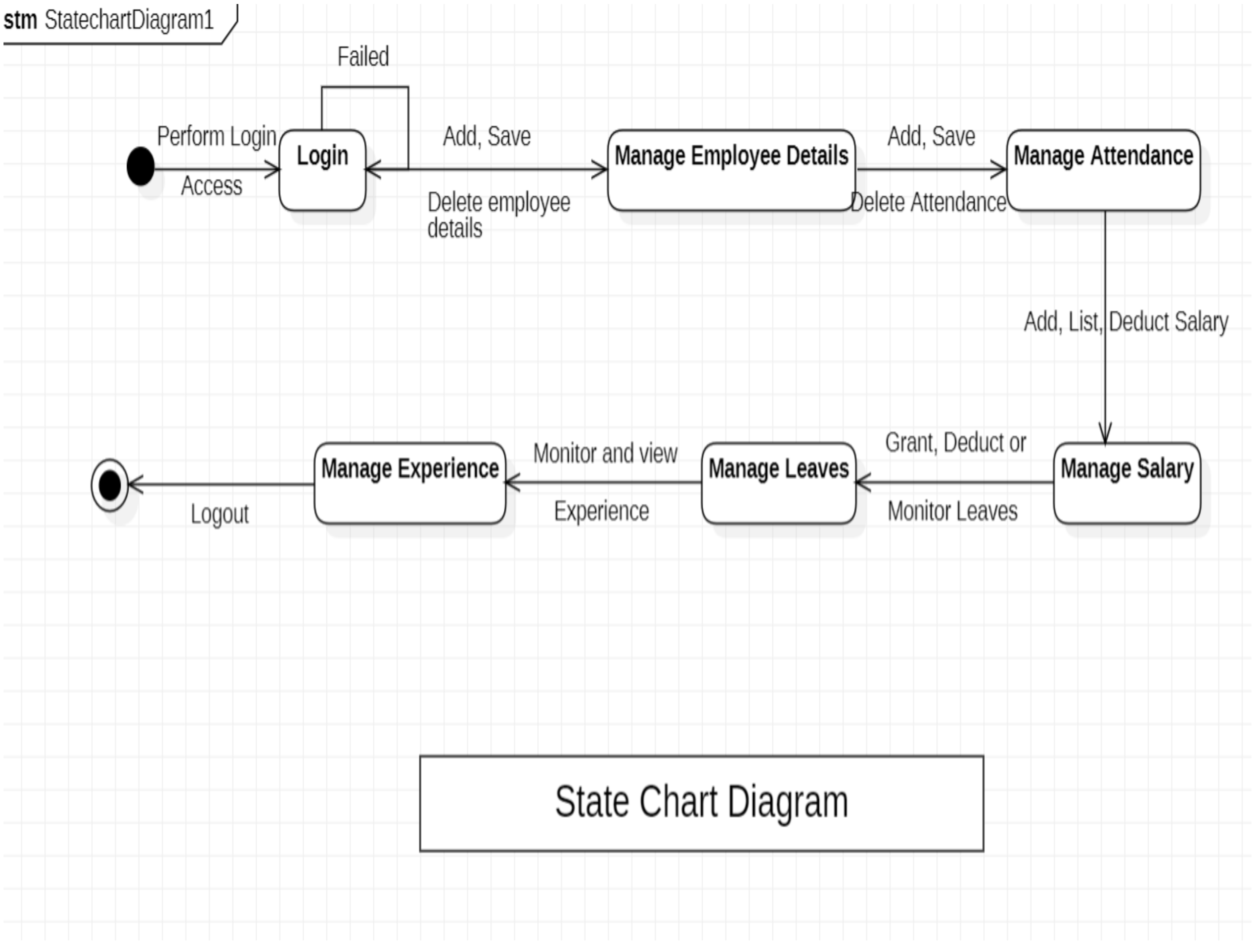
Final state: It represents the final state (end) of a system. It is denoted by a filled circle present within a circle.

Decision box: It is of diamond shape that represents the decisions to be made on the basis of an evaluated guard.

Transition: A change of control from one state to another due to the occurrence of some event is termed as a transition. It is represented by an arrow labeled with an event due to which the change has ensued.

State box: It depicts the conditions or circumstances of a particular object of a class at a specific point of time. A rectangle with round corners is used to represent the state box.

State Chart Diagram:



Conclusion:

As a whole, the UML Diagrams works together to achieve the most desired functions of an Employee Management Project System. All of these were designed to guide programmers and beginners about the behavior and structure of Employee Management System.

By completing all the given Diagrams, the Employee Management Project System development would be much easier and attainable. So those UML diagrams were given to teach you and guide you through your project development journey. You can use all of the given UML diagrams as your reference, or have them for your project development. The ideas presented in UML Diagrams were all based on Employee Management System requirements.

References

https://en.wikipedia.org/wiki/Use_case_diagram

https://en.wikipedia.org/wiki/Class_diagram

https://en.wikipedia.org/wiki/Sequence_diagram#:~:text=A%20sequence%20diagram%20or%20system,to%20carry%20out%20the%20functionality.

https://en.wikipedia.org/wiki/Communication_diagram

https://en.wikipedia.org/wiki/State_diagram

https://en.wikipedia.org/wiki/Activity_diagram

https://en.wikipedia.org/wiki/Component_diagram

https://en.wikipedia.org/wiki/Package_diagram

https://en.wikipedia.org/wiki/Deployment_diagram

https://itsourcecode.com/uml/employee-management-system-uml-diagrams/