Experiment 9

A college has more than thousand security persons, who are instructed to give duties at different places within the campus. Additionally, they also maintain a routine, which contains all information, such as Date, Duty Start Time, Duty End Time, and Place.Most importantly, all the places are covered by at least one security person. If a security person takes leave, manual entry is done against that person. Finally, at the end of a month, the security persons get paid for their duties, while considering the number of leaves as well. You can see that the manual calculation/operation is a heavy task for the security manager. Therefore, the objective is to build an Online security management system using class diagram through which entire security system within the campus can be controlled in an efficient manner.

Aim:

To design an Online Security Management System using a UML Class Diagram that efficiently manages security personnel duties, schedules, leaves, and payments within a college campus.

Procedure:

Step 1: Identify Key Classes in the System

SecurityPerson

Attributes: ID, Name, Contact, Salary, TotalLeaves

Methods: applyLeave(), viewSchedule(), calculateSalary()

DutySchedule

Attributes: ScheduleID, Date, StartTime, EndTime, Place

Methods: assignDuty(), viewSchedule()

LeaveManagement

Attributes: LeaveID, SecurityID, Date, Reason, Status

Methods: applyLeave(), approveLeave(), rejectLeave()

Payment

Attributes: PaymentID, SecurityID, Month, TotalWorkingDays, SalaryPaid

Methods: calculateSalary(), processPayment(), generatePayslip()

SecurityManager

Attributes: ManagerID, Name, Contact

Methods: assignDuty(), approveLeave(), processSalaries()

CampusLocation

Attributes: LocationID, LocationName

Methods: assignSecurity()

Step 2: Establish Relationships Between Classes

SecurityPerson is assigned to DutySchedule (One-to-Many).

SecurityPerson can request leave, which is managed by LeaveManagement (One-to-Many).

SecurityPerson receives payment, handled by Payment (One-to-One).

SecurityManager supervises DutySchedule, LeaveManagement, and Payment (One-to-Many).

Each DutySchedule is assigned to a CampusLocation (One-to-One).

Step 3: UML Class Diagram Components

Classes: SecurityPerson, DutySchedule, LeaveManagement, Payment, SecurityManager, CampusLocation.

Relationships:

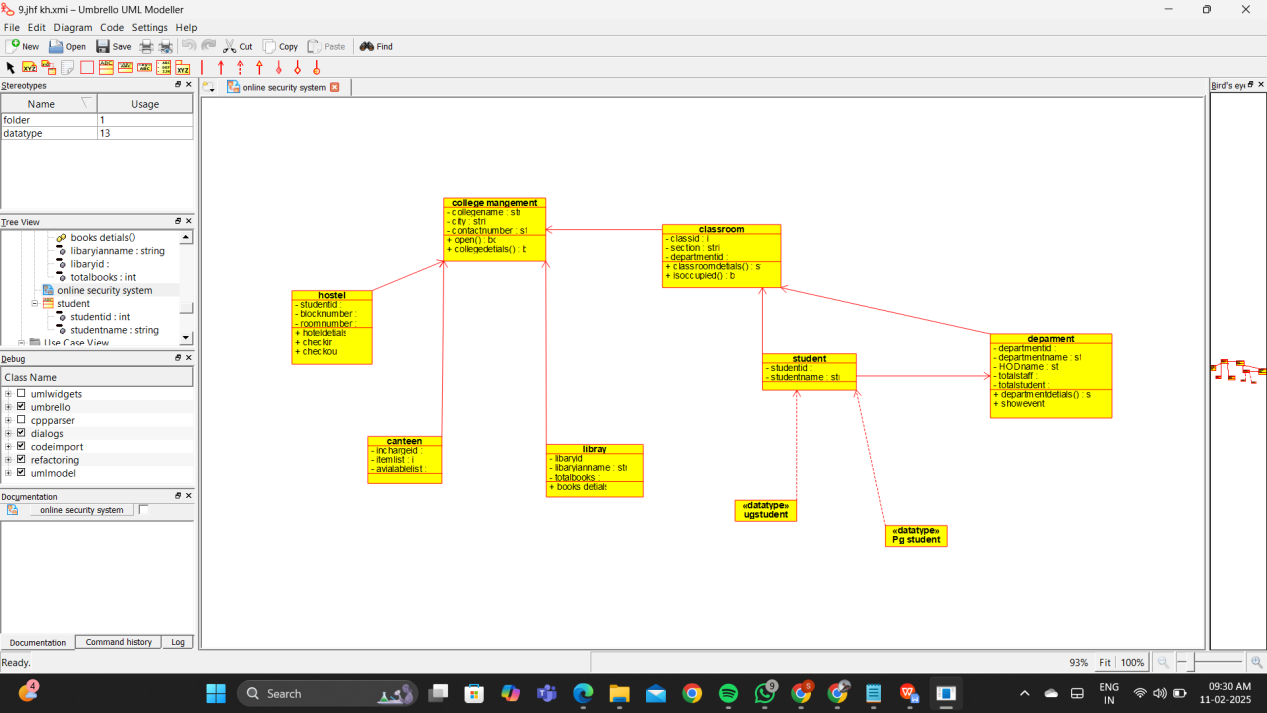
Association between SecurityPerson and DutySchedule.

Association between SecurityPerson and LeaveManagement.

One-to-One between SecurityPerson and Payment.

One-to-Many between SecurityManager and other classes.

Output:



Result :

Thus the UML diagram has been implemented successfully.