EXPERIMENT-9

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 a Class Diagram for an Online Security Management System that efficiently manages security personnel, their duties, leave records, and salary calculations within a college campus.

Procedure:

1. Identify Main Entities (Classes)

1. **Security Person**
    - Attributes: securityID, name, contact, assignedDuty
    - Methods: requestLeave(), markAttendance()
2. **Duty Schedule**
    - Attributes: scheduleID, date, startTime, endTime, dutyPlace, assignedSecurity
    - Methods: assignDuty(), updateDuty()
3. **Leave Management**
    - Attributes: leaveID, securityID, leaveDate, status
    - Methods: applyLeave(), approveLeave()
4. **Salary Management**
    - Attributes: salaryID, securityID, totalDuties, totalLeaves, finalSalary
    - Methods: calculateSalary(), processPayment()
5. **Admin (Security Manager)**
    - Attributes: adminID, name, role
    - Methods: assignDuties(), approveLeave(), generateSalaryReports()

2. Establish Relationships

- **Security Person** ↔ **Duty Schedule**: One security person can have multiple duty assignments.

- **Security Person ↔ Leave Management**: A security person can apply for leave, which gets recorded.
- **Security Person ↔ Salary Management**: The salary is calculated based on duty attendance and leaves.
- **Admin ↔ Duty Schedule & Leave Management**: Admin assigns duties and manages leave approvals.

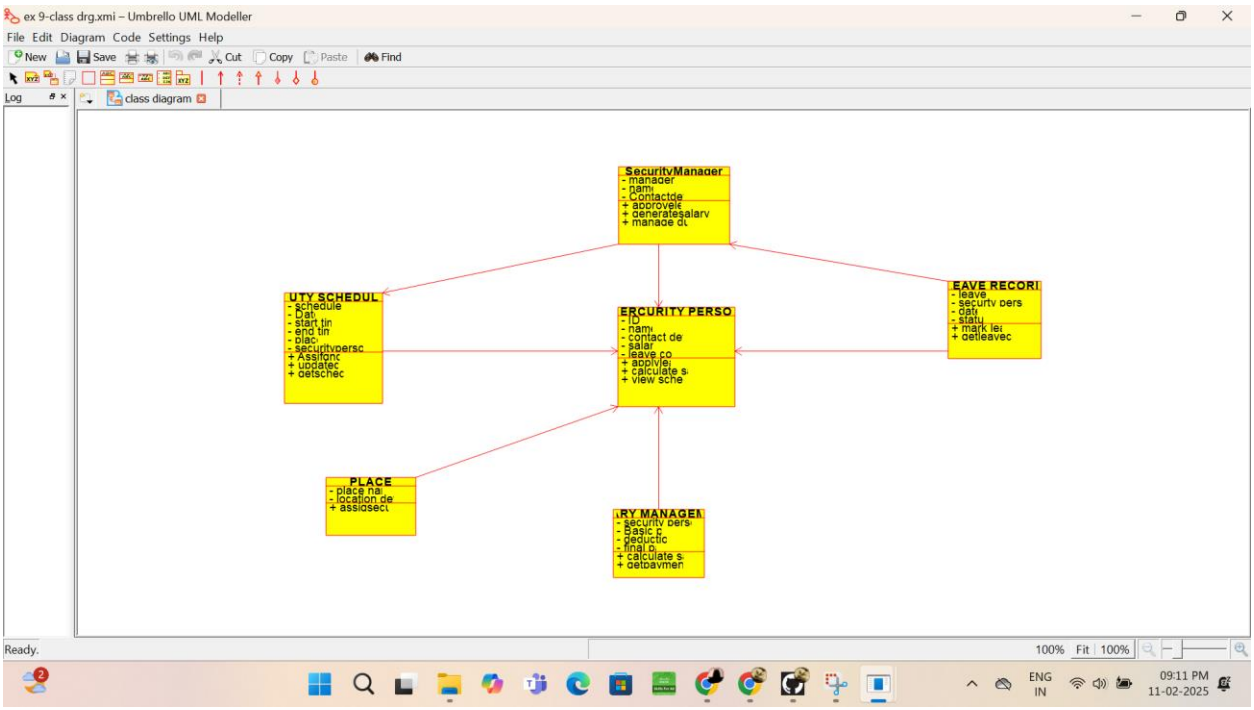
3. Draw the UML Class Diagram

- Represent **classes** as rectangles with attributes and methods.
- Use **associations** to connect related classes.
- Apply **aggregation and composition** where required.

4. Verify the Diagram

- Ensure that all functionalities (duty allocation, leave tracking, salary processing) are correctly represented.

Diagram:



Result:

A **Class Diagram** for the **Online Security Management System** was successfully designed, helping automate duty scheduling, leave tracking, and salary calculations efficiently.