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 automates duty scheduling, leave management, and payroll processing for security personnel within a college campus. This system will replace manual record-keeping with an efficient digital solution.

Procedure (Step-by-Step Process)

Step 1: Identify the Key Entities (Classes) in the System

The main components of the system include:

1. **SecurityPerson** – Represents individual security personnel.
2. **DutySchedule** – Maintains records of duty assignments.
3. **Place** – Locations within the campus where security personnel are assigned.
4. **LeaveRecord** – Manages leave applications and approvals.
5. **Salary** – Calculates payments based on attendance.
6. **SecurityManager** – Oversees scheduling, leave approvals, and payroll.

Step 2: Define Class Attributes and Methods

1. SecurityPerson Class

- **Attributes**: `securityID`, `name`, `contact`, `assignedDuties`, `leavesTaken`
- **Methods**: `requestLeave()`, `viewSchedule()`, `checkSalary()`

2. DutySchedule Class

- **Attributes**: `dutyID`, `securityID`, `date`, `startTime`, `endTime`, `placeID`
- **Methods**: `assignDuty()`, `modifyDuty()`, `viewSchedule()`

3. Place Class

- **Attributes**: `placeID`, `placeName`, `securityCoverageStatus`
- **Methods**: `assignSecurity()`, `checkCoverage()`

4. LeaveRecord Class

- **Attributes**: `leaveID`, `securityID`, `date`, `status`
- **Methods**: `applyLeave()`, `approveLeave()`, `rejectLeave()`

5. Salary Class

- **Attributes**: `salaryID`, `securityID`, `basicPay`, `deductions`, `finalPay`
- **Methods**: `calculateSalary()`, `generatePayslip()`

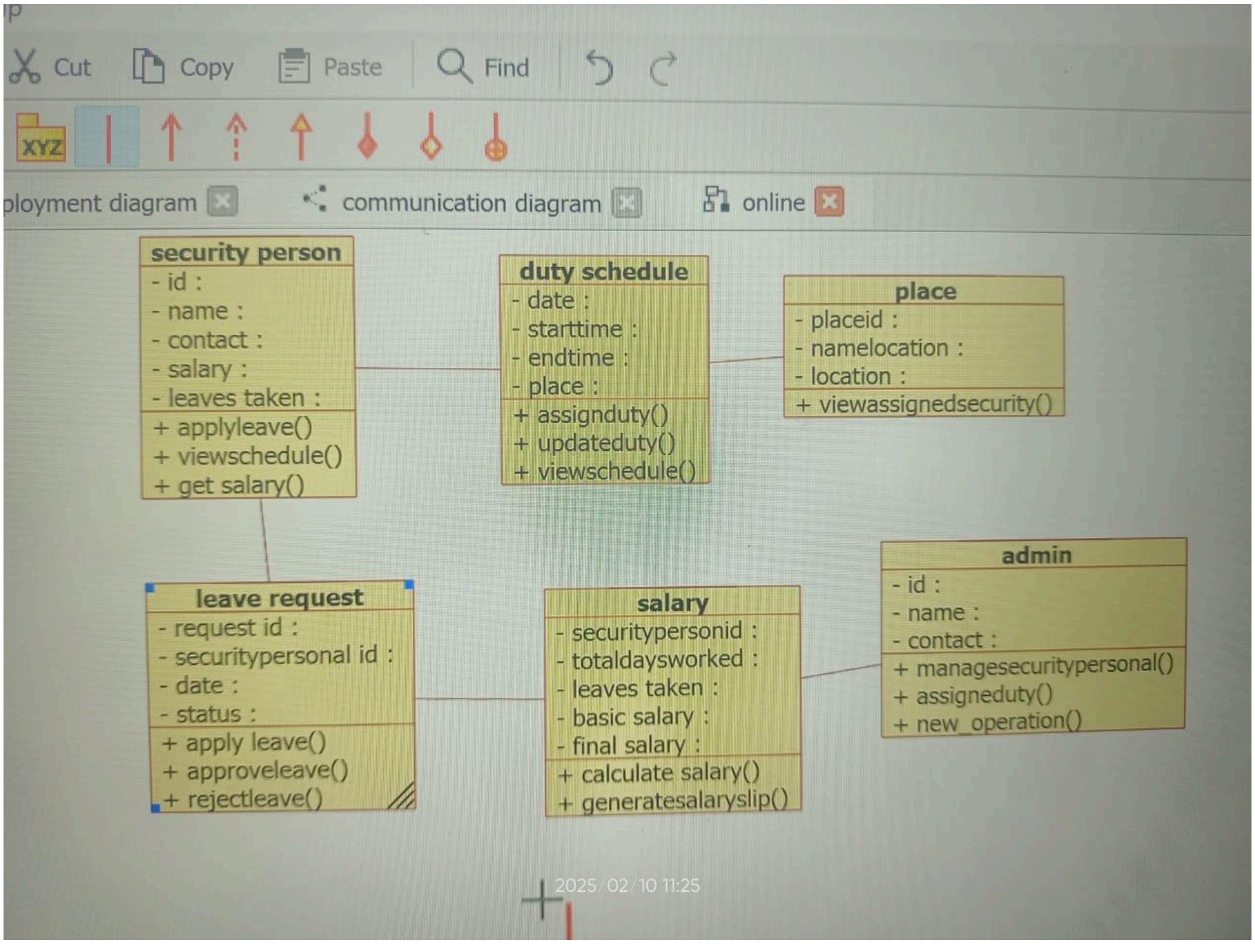
6. SecurityManager Class

- **Attributes**: `managerID`, `name`, `contact`
- **Methods**: `approveLeave()`, `assignDuty()`, `generateSalaryReport()`

Step 3: Define Relationships Between Classes

- **SecurityPerson** is assigned to multiple **DutySchedules**.
- **DutySchedule** is linked to **Places** where security personnel are assigned.
- **SecurityPerson** may have **LeaveRecords**, which affect **Salary** calculations.
- **SecurityManager** manages **LeaveRecords**, **DutySchedules**, and **Salary** processing.

class diagram



Result

The **Online Security Management System** ensures **efficient duty scheduling, leave tracking, and payroll processing** for security personnel within the college campus. The **Class Diagram** helps streamline operations, reducing **manual workload** and improving **accuracy and efficiency** in managing security staff.