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 securityperson takes leave, manual entry is done against that person. Finally, at the end of amonth, 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 duty schedules, leave records, and salary calculations within a college campus.

PROCEDURE :

1. Identify Key Classes

- **SecurityPerson**: Represents security personnel with personal and duty details.
- **Duty**: Represents assigned duties with details like time, place, and duration.
- **LeaveRecord**: Tracks leaves taken by security personnel.
- **Salary**: Handles salary calculation based on duties performed and leaves taken.
- **Place**: Represents locations where duties are assigned.

2. Define Attributes for Each Class

- **SecurityPerson**
  - `securityID`: Unique identifier for each security person.
  - `name`: Name of the security person.
  - `contactNumber`: Contact information.
  - `address`: Address details.
  - `totalLeaves`: Total leaves taken.
  - `totalDuties`: Total duties performed.
- **Duty**
  - `dutyID`: Unique identifier for each duty.
  - `securityID`: Associated security person.
  - `date`: Date of the duty.
  - `startTime`: Start time of the duty.
  - `endTime`: End time of the duty.
- **LeaveRecord**
  - `leaveID`: Unique identifier for leave records.
  - `securityID`: Associated security person.
  - `date`: Date of leave.
  - `reason`: Reason for leave.

- **Salary**
    - `salaryID`: Unique identifier for salary details.
    - `securityID`: Associated security person.
    - `totalLeaves`: Leaves considered for salary calculation.
    - `totalDuties`: Duties considered for salary calculation.
- **Place**
    - `placeID`: Unique identifier for each place.
    - `name`: Name of the place.
    - `location`: Details of the location.

3. Establish Relationships Between Classes

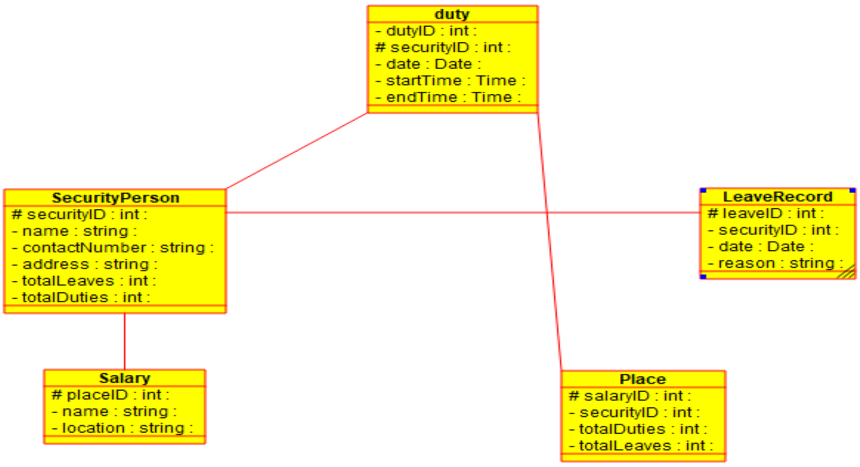
- **SecurityPerson → Duty**: One-to-Many (One security person is assigned multiple duties).
- **SecurityPerson → LeaveRecord**: One-to-Many (One security person can have multiple leave records).
- **SecurityPerson → Salary**: One-to-One (Each security person has a salary record).
- **Duty → Place**: Many-to-One (Many duties can occur at one place).

4. Use CASE Tool to Draw the Diagram

- Add the identified classes and their attributes.
- Connect classes using associations to represent relationships.
- Include multiplicity (e.g., 1..*, 0..1) to show the cardinality between classes.

OUTPUT:

CLASS DIAGRAM:



RESULT:

The **Class Diagram** for the **Online Security Management System** is successfully created.
It provides a structured representation of security personnel, their duty schedules, leave management, and salary processing, ensuring an efficient and automated approach to managing security operations within the college campus.

This diagram effectively models the security system workflow and enhances operational efficiency.