

Campus Security Staffs Management System



Lecturer
Nguyen Hai Vinh
Cuong



GROUP MEMBERS

Nguyen Thi Thanh Mai

Van Thi Huong Duyen

Luong Anh Kiet

Nguyen Duc Trong

Table of Contents



Objective

Objective of our software product



System Modelling Diagrams

Use Case Diagram, and
Class Diagram



Functionalities

Functionalities description &
Non-Functional requirements

Table of Contents



We Have Used
Programming Languages,
Software Tools, Databases



Software Architecture
Which type have we
followed



Detailed Execution
Functionalities Details



01.

Objective

Objective of Our Software Product



Campus Security Staffs Management System



Campus Security Staffs Management System

Build a security management system through which entire security system within the campus can be controlled in an efficient manner



A wide-angle photograph of a modern university campus. In the foreground, there are several large, leafy green trees. Behind them is a long, dark brown building with multiple stories and many windows. The sky is blue with some white clouds.

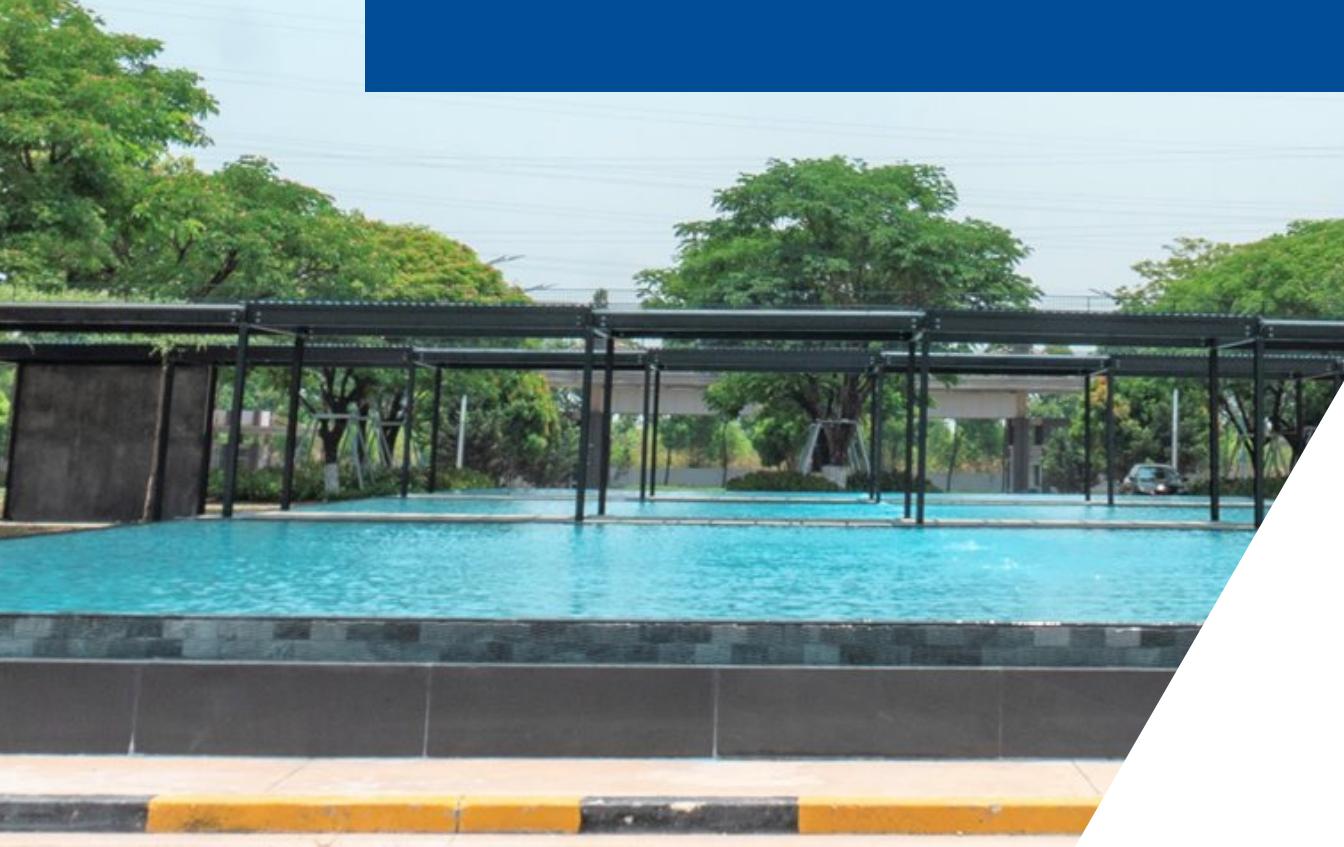
02.

System Modelling Diagrams

Use Case Diagram and Class Diagram



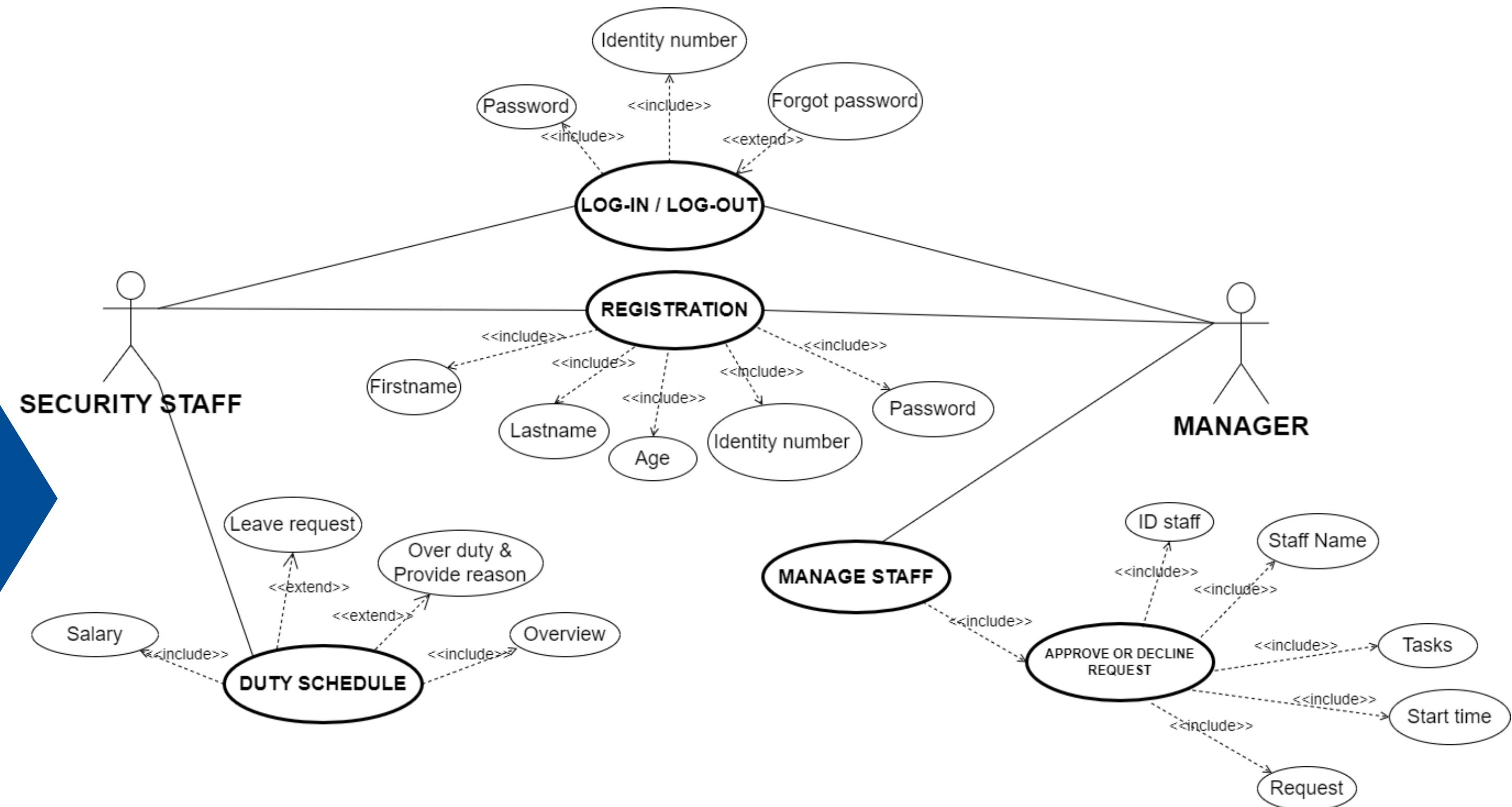
Use Case Diagram



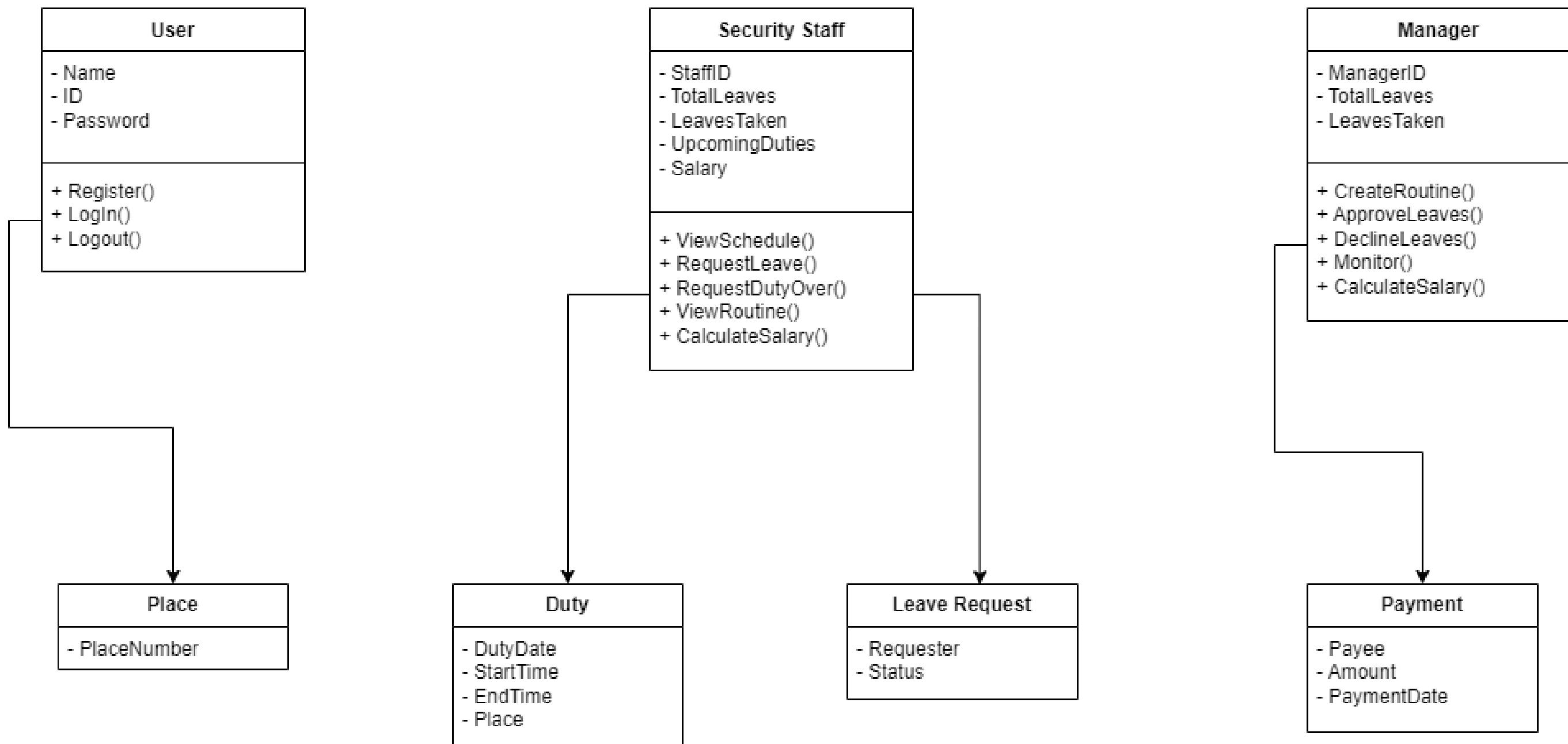
Class Diagram

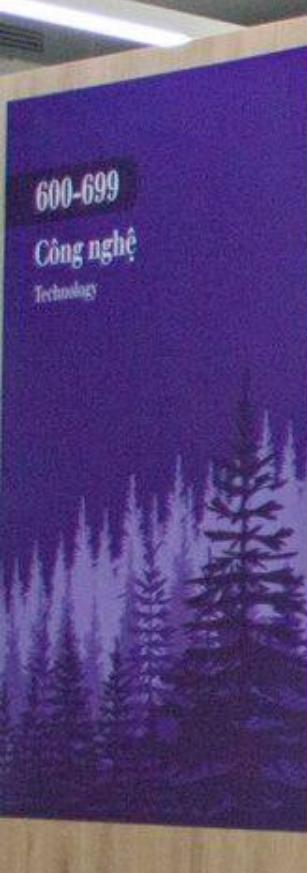


Use Case Diagram



Class Diagram





03.

Functionalities

Functionalities Description & Non-Functional Requirements



Functionalities
Description

Functionalities of Staff

Functionalities of Manager

Functionalities Of Staff

Log In

log in by username, password

View Duty

view date, place, start time, end time

Request Leave/Duty

request manager to take leave
or to do over duties

Request Approved/Declined

request manager to
approved/declined

Number of leaves taken/ allowed leaves remaining

view routine and
salary at the end of the month

Log Out

log out of the staff account

Functionalities Of Manager

Log In

log in by manager id, password

Create Routine

for upcoming 7 days for all persons
considering leave requests

Approve / Decline

approve / decline leave requests

Monitoring Routine /Salary

monitoring staff routine and
salary at the end of the month

View Routine

view staff's detail routine

Log Out

log out of the manager account

Non-Functional Requirements

Security

user passwords must be stored securely using industry-standard encryption

Performance

The system should respond to user actions within 2 seconds

Reliability

The system should not experience more than one critical failure per month

Usability

the system should be usable by users with basic computer skills





04.

We Have Used

Programming Languages, Software Tools, Databases

PROGRAMMING LANGUAGES

JAVA

- Platform Independence
- Object-Oriented Programming (OOP)
- Rich Standard Library
- Security

SOFTWARE TOOLS

APACHE NETBEANS IDE

- Integrated Development Environment (IDE)
- Cross-Platform Support
- Smart Code Editing
- Rich Plugin Ecosystem

DATABASES

MYSQL

- Open Source
- Scalability
- Performance
- Replication and High Availability



05.

Software Architecture

Which Type Have We Followed

Three-Tier Architecture

What is the Three-Tier Architecture?

- Divides an application into three interconnected components or tiers
 - Presentation Tier (User Interface)
 - Application Tier (Logic or Processing)
 - Data Tier (Database)



What are the benefits of the Three-Tier Architecture?

Maintenance

The separation of concerns allows for easier maintenance

Performance

The separation of concerns allows for easier maintenance

Flexibility

Allows for greater flexibility in choosing technologies for each tier

Scalability

Each tier can be scaled independently based on the specific needs of that tier



06.

Detailed Execution

Four Functionalities Details



Thank You For
Your Listening

Do you have any
questions?