NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. VIth Sem

Subject Code: CS-315

Subject Name: Object-Oriented Design Lab

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

Under the Guidance of:

Dr. Malaya Dutta Borah

Assistant Professor

Department of Computer Science and Engineering

LIST OF EXPERIMENTS

AIM I: TO DRAW THE UML DIAGRAM FOR A CONFERENCE MANAGEMENT SYSTEM

THEORY:

Scientific conferences publish the latest innovations and outcomes in research work spanning several domains. However, before a research paper is published – or even selected for publication – it must go through a rigorous peer-review process. A Conference Management System (CMS) is software that helps to manage and streamline such review processes. EDAS is an example of a CMS.

Organizing a conference is a huge task, and the responsibilities are shared by many people divided into different committees. In general, there are two large committees – the program committee (PC) and the organizing committee (OC). The PC consists of several people (both from academia and industry) who are experts in their respective areas. The role of PC members includes pre-screening the submissions to ensure that appropriate qualities of papers are considered. Subsequently, the PC members assign reviewers for each paper under consideration. Any given paper must have at least two reviewers assigned. However, if a reviewer declines to review a particular paper, new reviewer(s) must be assigned for it. Once all the reviews for all the papers are available, the PC members select a fraction of them for publication. The PC chair(s) oversee the entire review phase and help in tie-breaking if required.

The OC has 1-2 general chair(s) who oversee the entire conference organization process. The OC consists of smaller subcommittees with 1-5 members such as publication chair(s), publicity chair(s), registration chair(s), web chair(s), and so on.

Once a conference has been planned, the OC chair(s) has to request the CMS to create an entry for the corresponding conference. The CMS provides a special link to make such requests online. After an entry for the conference is created, the OC chairs are given admin privileges, who, in turn, are responsible for adding the other relevant members. The URL linking the conference to the CMS is advertised. Authors access that URL, and upload their papers along with all other relevant inputs. After a pre-published deadline, the decision of a paper (accept or reject) together with reviewers 39; comments are visible to the authors. In case a paper is accepted, the authors should make changes as per the review comments, if relevant, and upload the camera-ready version within a specified deadline.

INPUTS:

- Conference information (title, date, venue, topics covered, and deadlines)
- OC & PC information (email addresses of the members)
- User information (name, password, email address, affiliation, and areas of expertise [optional])
- Paper information (title, abstract, authors, affiliations, and PDF file)
- Review comments:
    - Rate the paper on a scale of 1-5
    - Detailed comments
    - Reviewer 39 opinion on whether to accept/reject the paper
- Final selection decision (paper id, accept/reject)

OUTPUTS:

- Details of papers submitted
- Review decision (see the format of review comments)
- List of papers accepted
- Percentage of papers accepted

CONSTRAINTS:

- All users must register with the CMS to access the portal
- A reviewer assigned for a paper may not be an existing user of the CMS; he/she must register in order to submit the review comments
- Login information consists of email address and password
- Each submitted paper is assigned a unique ID
- It is sufficient if only one author of a paper is registered to the CMS; he/she will be responsible for submitting the paper
- An author can view the review decision for his/her paper(s) only
- A reviewer can view information only about those papers for which he/she has been assigned a reviewer
- Only PC members can view all reviews for all papers

CASE DIAGRAM:

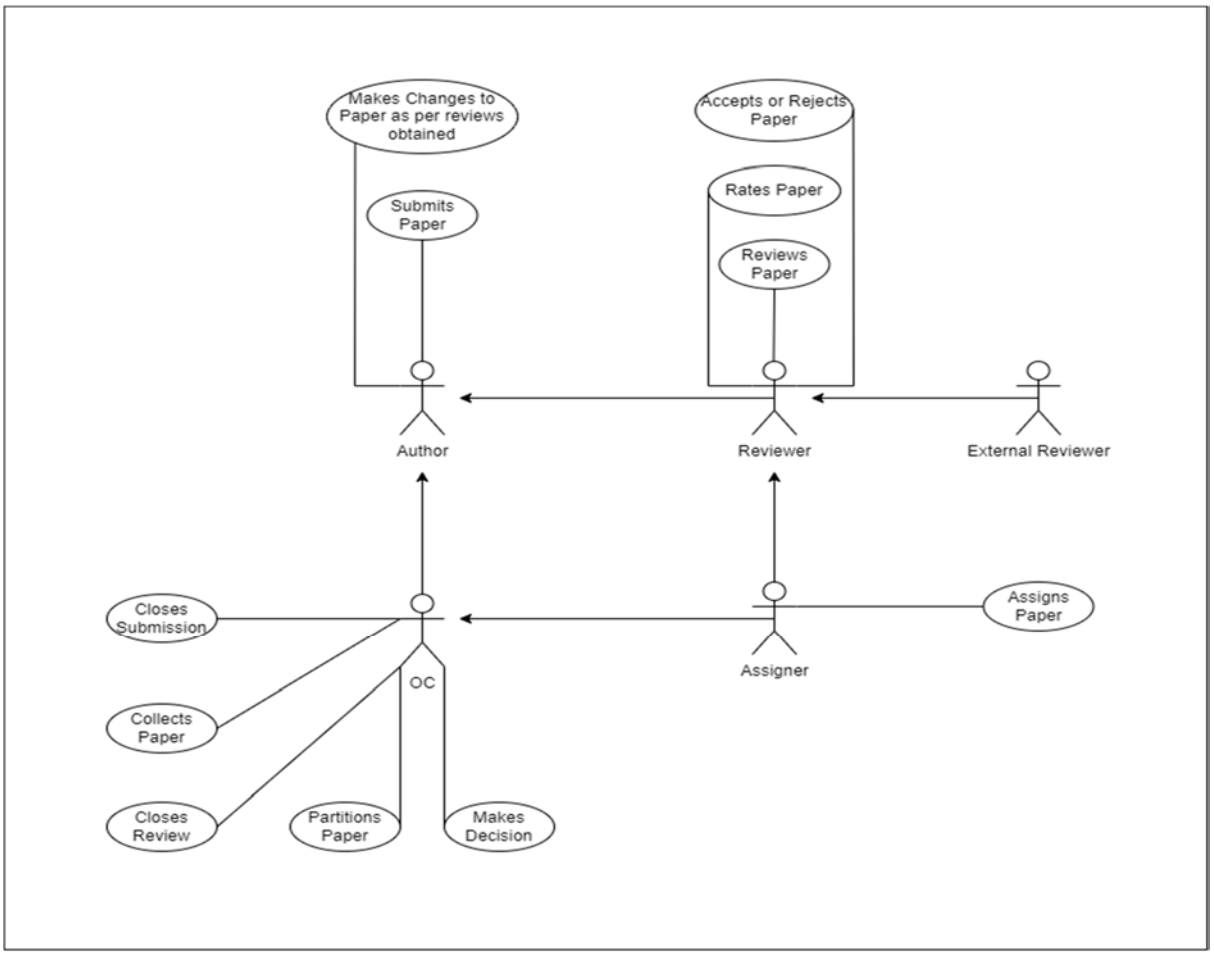


Fig 1.1: Case Diagram for a Conference Management System

EXPLANATION:

We have 5 actors here namely, Assigner, OC (organising committee), Author, Reviewer and External Reviewer. The assigner assigns the paper. The organising committee is in charge of making decision, collecting paper, closing submission, closing reviews etc. The author then submits paper as per the instructions received from the organising committee. If the submission has not been closed yet, the author can still make changes to their submitted paper. After that, the Reviewer reviews paper, either accepts or rejects the paper, and rates the paper. The external reviewer can also rate and review the paper.

CLASS DIAGRAM:

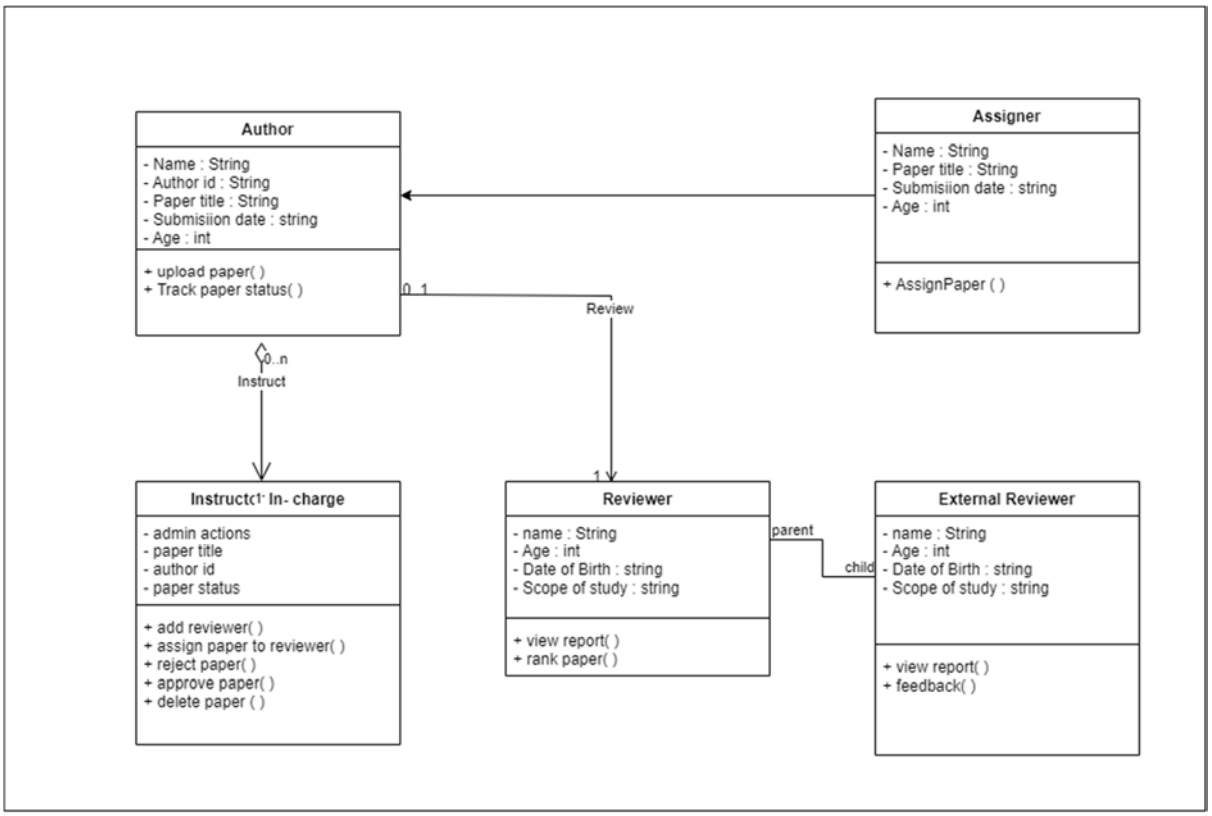


Fig 1.2: Class Diagram for a Conference Management System

SEQUENCE DIAGRAM:

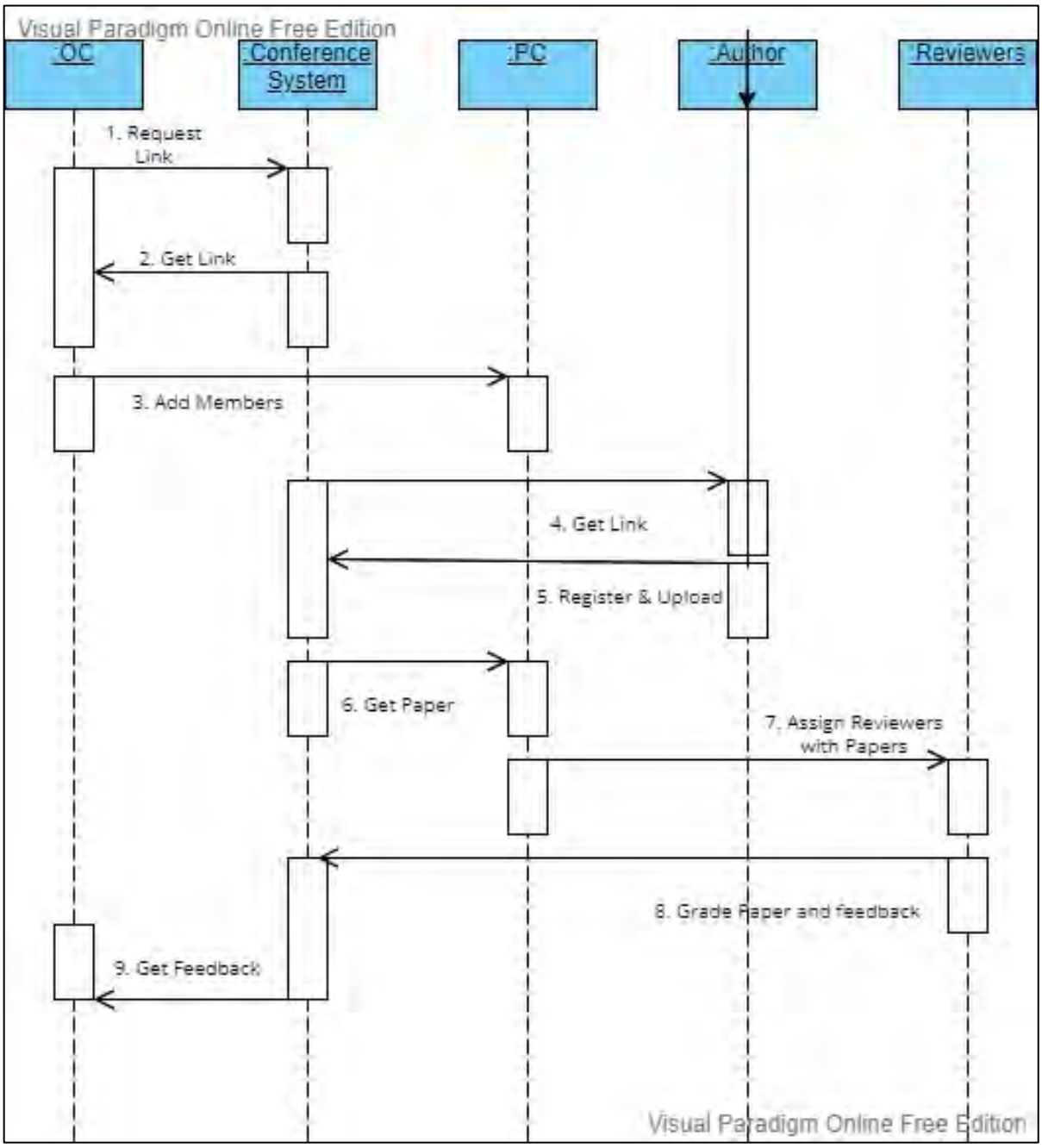


Fig 1.2: Sequence Diagram for a Conference Management System

STATE DIAGRAM:

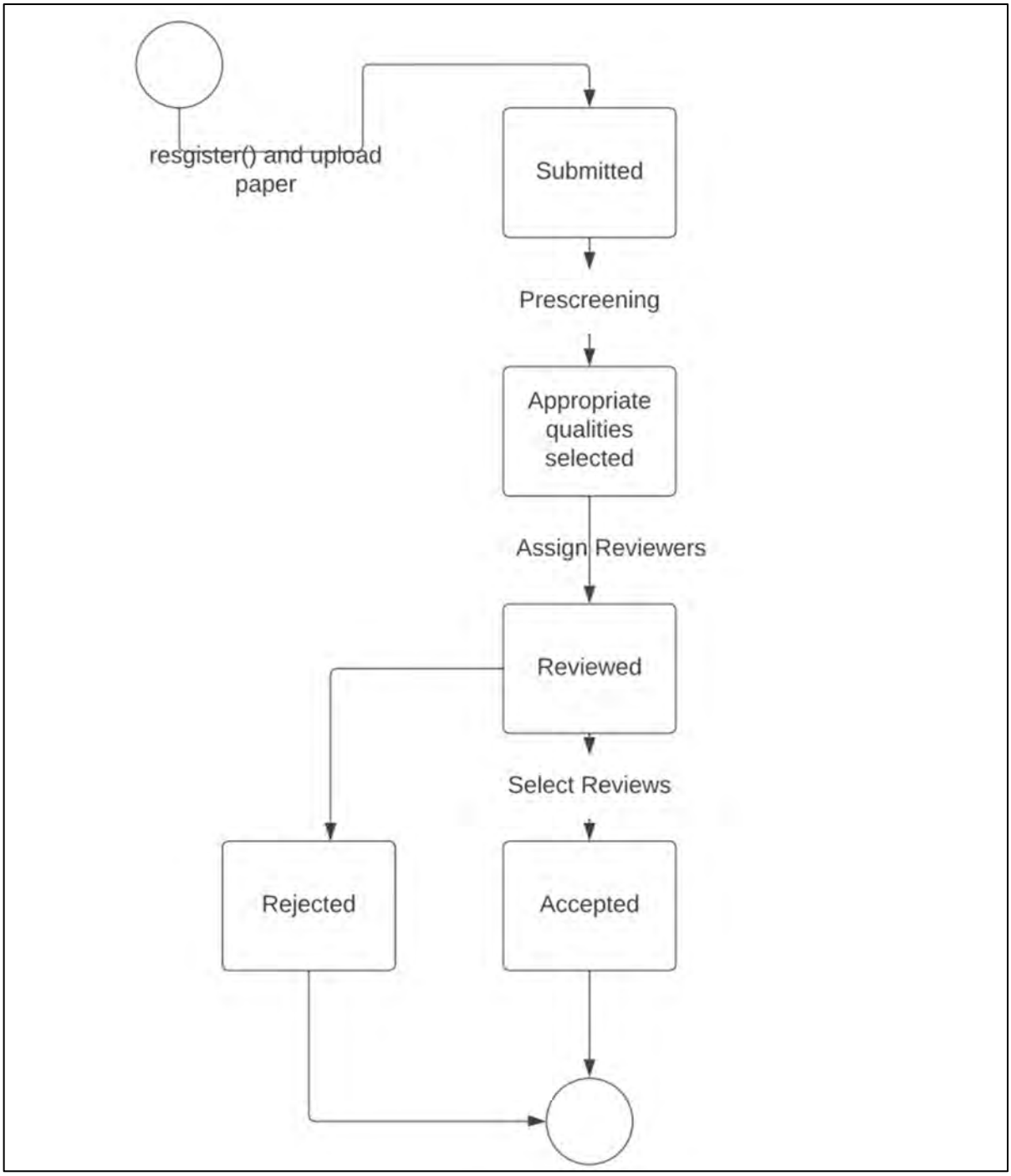


Fig 1.2: State Diagram for a Conference Management System

COLLABORATION DIAGRAM:

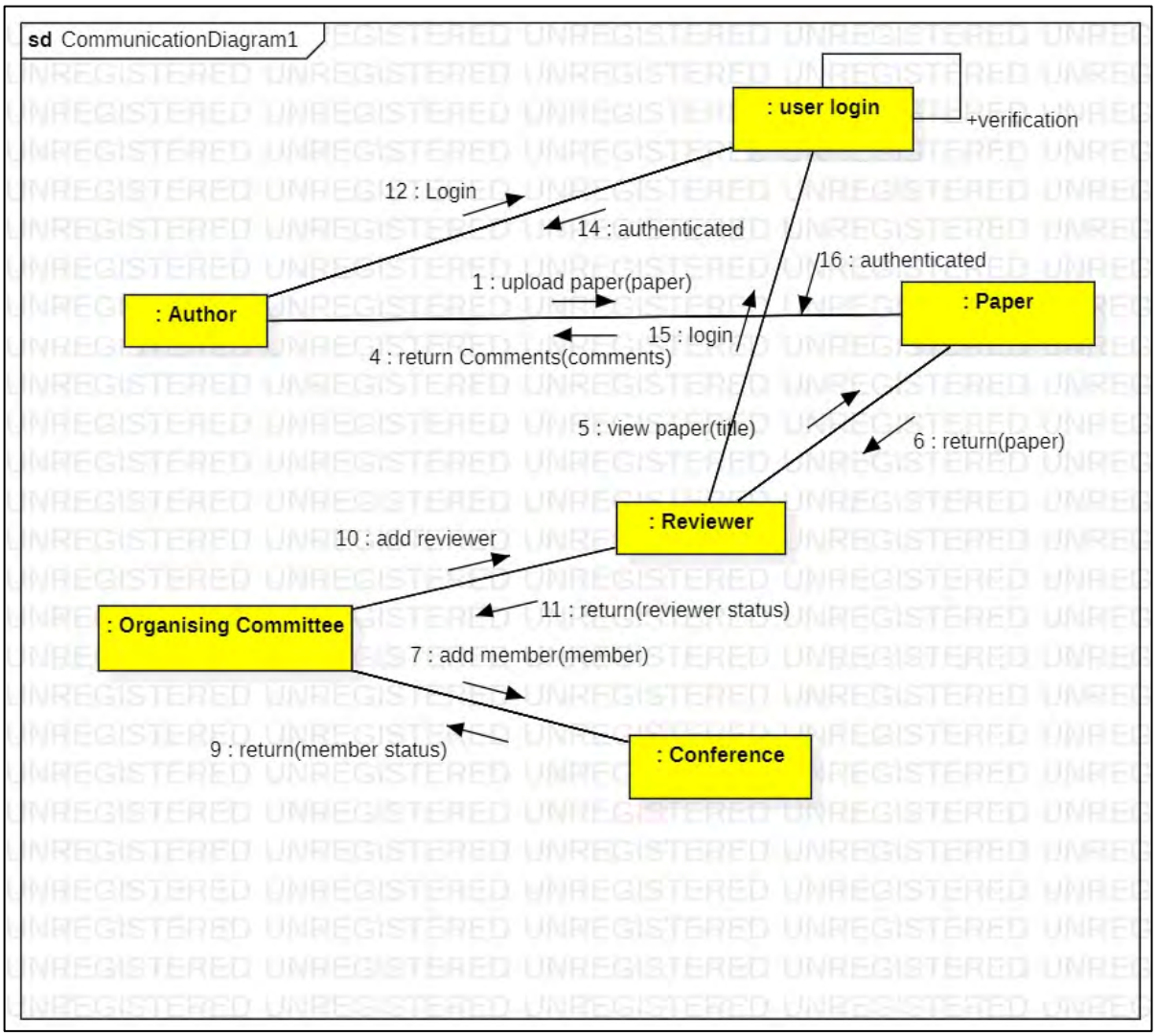


Fig 1.2: Class Diagram for a Conference Management System

PACKAGE DIAGRAM:

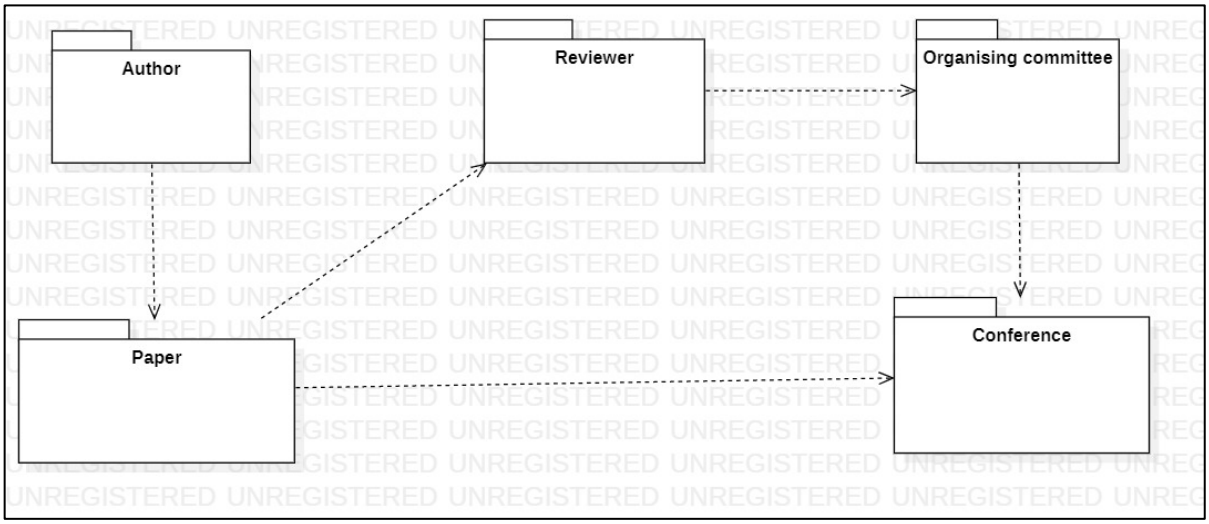


Fig 1.2: Class Diagram for a Conference Management System

COMPONENT DIAGRAM:

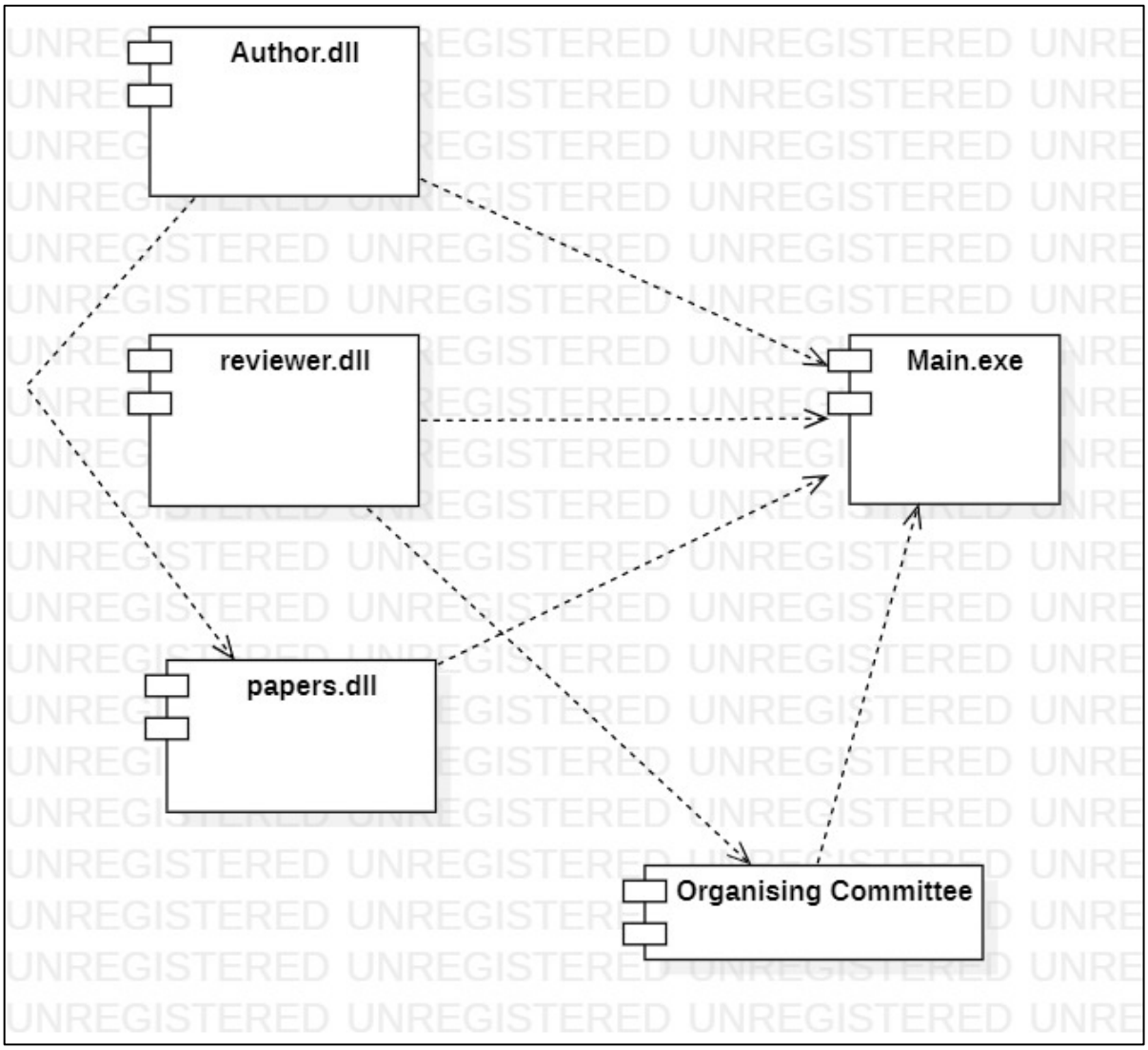


Fig 1.2: Class Diagram for a Conference Management System

DEPLOYMENT DIAGRAM:

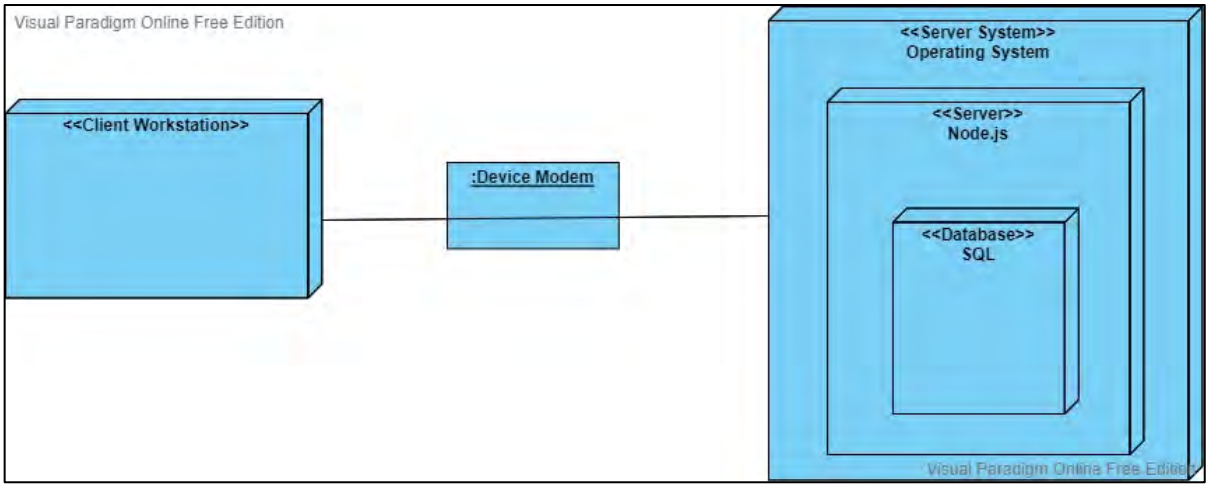


Fig 1.2: Class Diagram for a Conference Management System

AIM II: TO DRAW THE UML FOR A DELIVERY AGENT SYSTEM

THEORY:

There are many online shopping portals such as Flipkart, Amazon, Snapdeal, etc. are active in the Indian market. One major task is to deliver online books products to the customers as first as possible in a cost-effective (cheapest) manner. A delivery agent system, which would automatically receive a delivery request from an online portal and identify the couriers, whom the delivery job can be assigned deadline.

INPUTS:

- Shipping details (source and destination) locations
- Couriers' details in different localities.
- Service offering for each courier company.

OUTPUTS:

- Booking delivery
- Status of delivery
- Cancellation of booking
- Delivery rescheduling
- Reward and penalty calculation
- Archiving record of past services

CONSTRAINTS:

- Booking confirmation, if booking is successful.
- Reporting delivery status
- Cancellation of booking confirmation
- Update record on delivery rescheduling
- Update record and intimation on reward and penalty calculation
- Query generation on record of past service, given a courier agency

CASE DIAGRAM:

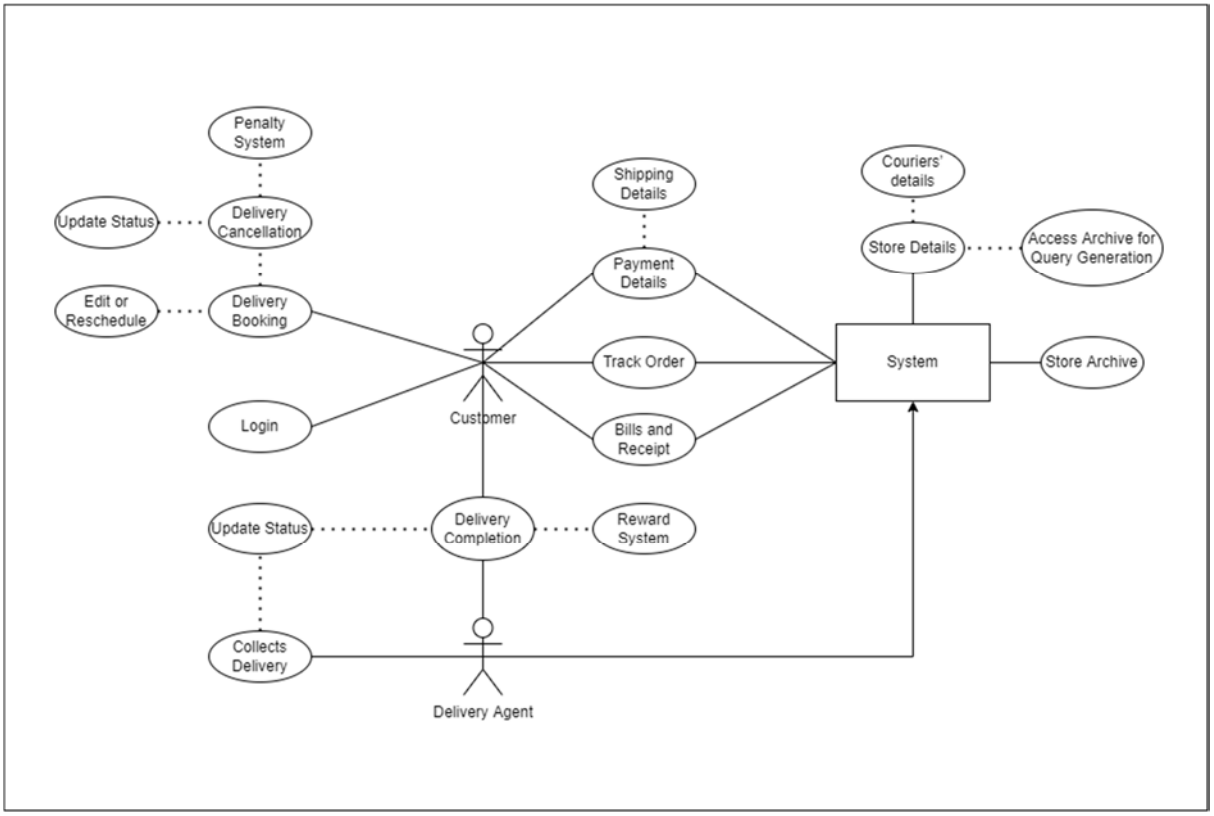


Fig 2.1: Case Diagram for a Delivery Agent System

EXPLANATION:

Firstly, we have two actors- Customer and Delivery Agent, because it is a Delivery Agent System. Since it is online, I have included an online system "System" which can be either an actor (Admin) or an automated system (System). The customer will login, and book a delivery. He can also edit his delivery or reschedule his delivery, so the Delivery Booking is extended to include these constraints. After the delivery is booked, the customer will proceed to payment details which will include his shipping address. Once the delivery has been booked, he gets receipt of his transaction or order, also the customer can track his order. The system will have store details as well as the details of delivery agents near to those stores. So, the system will alert the delivery agent that a delivery has to be made. The delivery agent will collect delivery and updates the status. After the delivery agent has completed his delivery, he again updates the status and the customer will get some reward. The entire order and delivery details will be archived in the system and stored there for future references.

CLASS DIAGRAM:

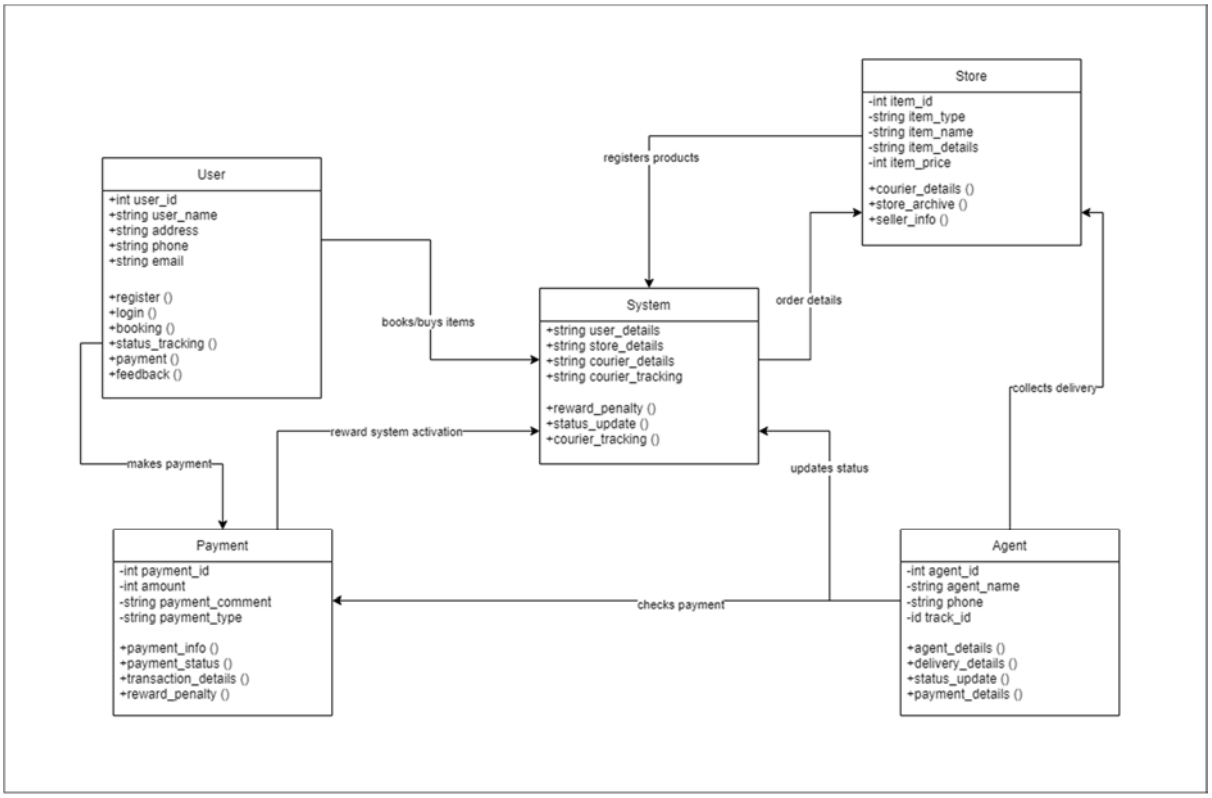
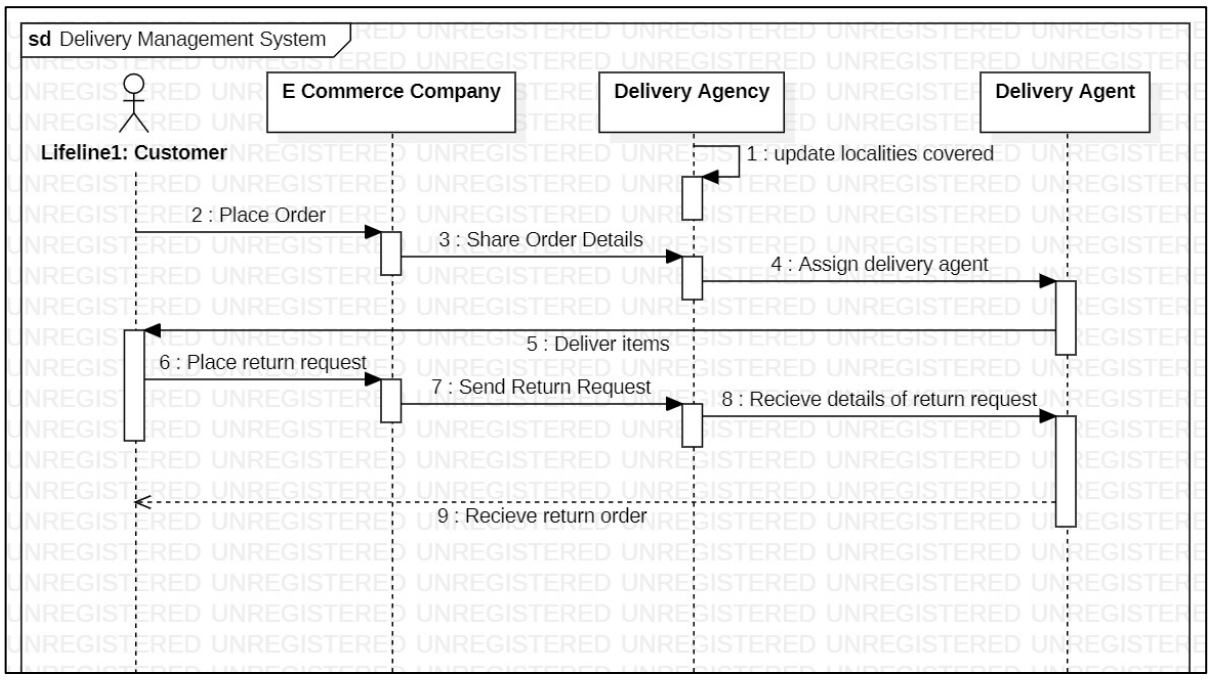
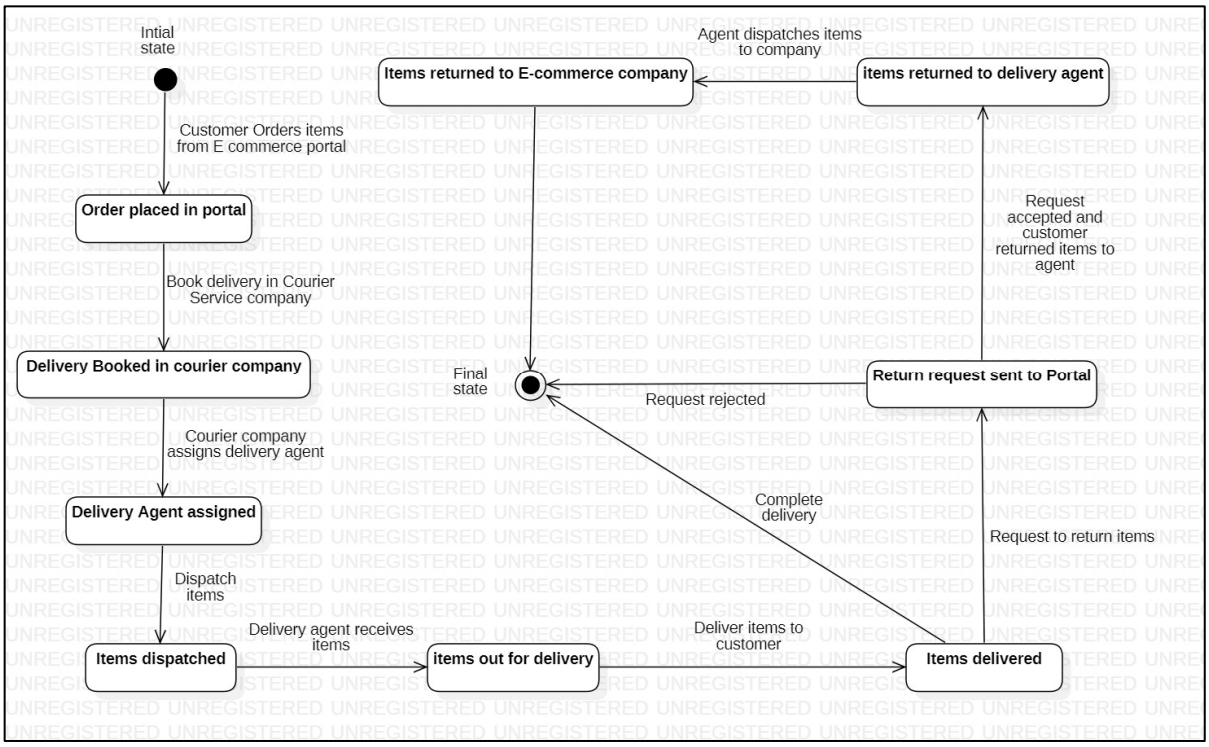


Fig 2.2: Class Diagram for a Delivery Agent System

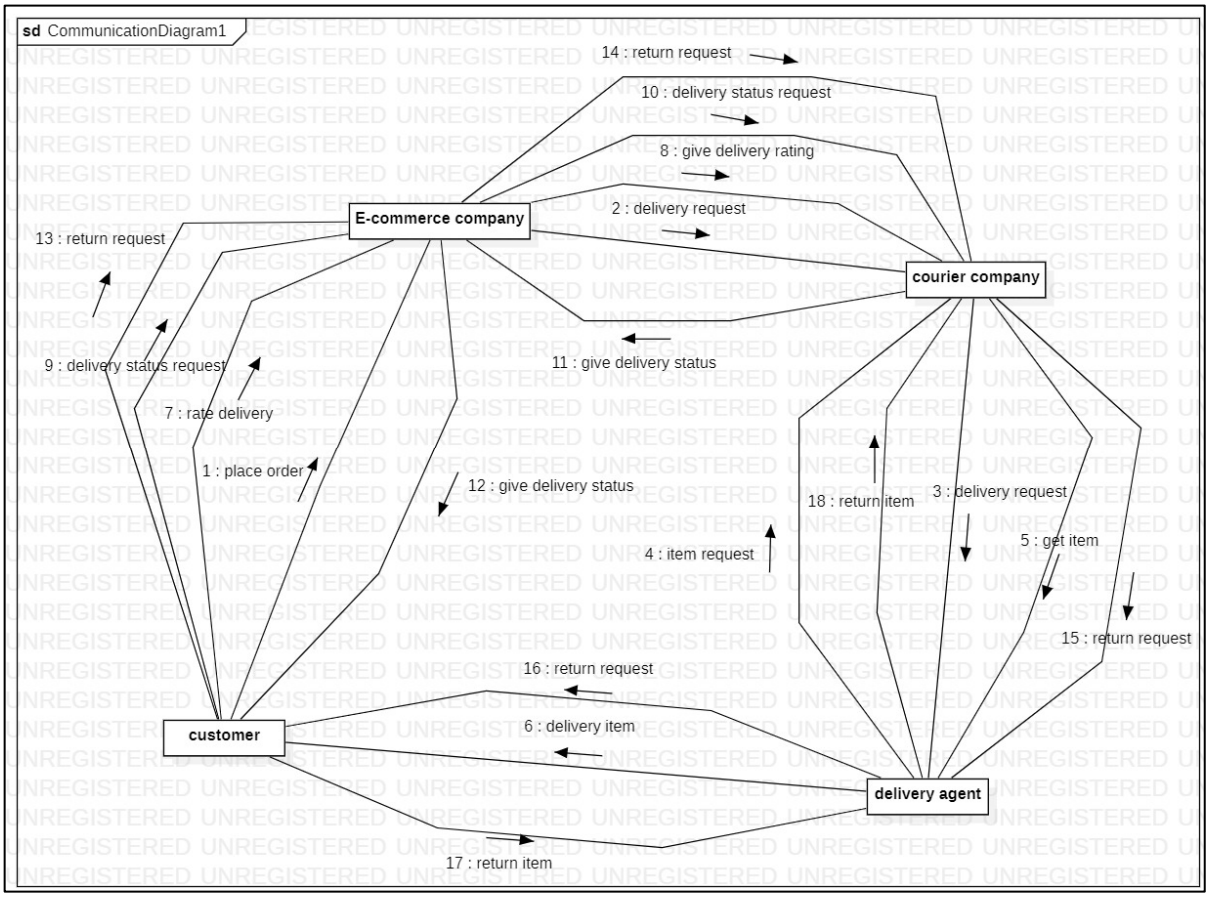
SEQUENCE DIAGRAM:



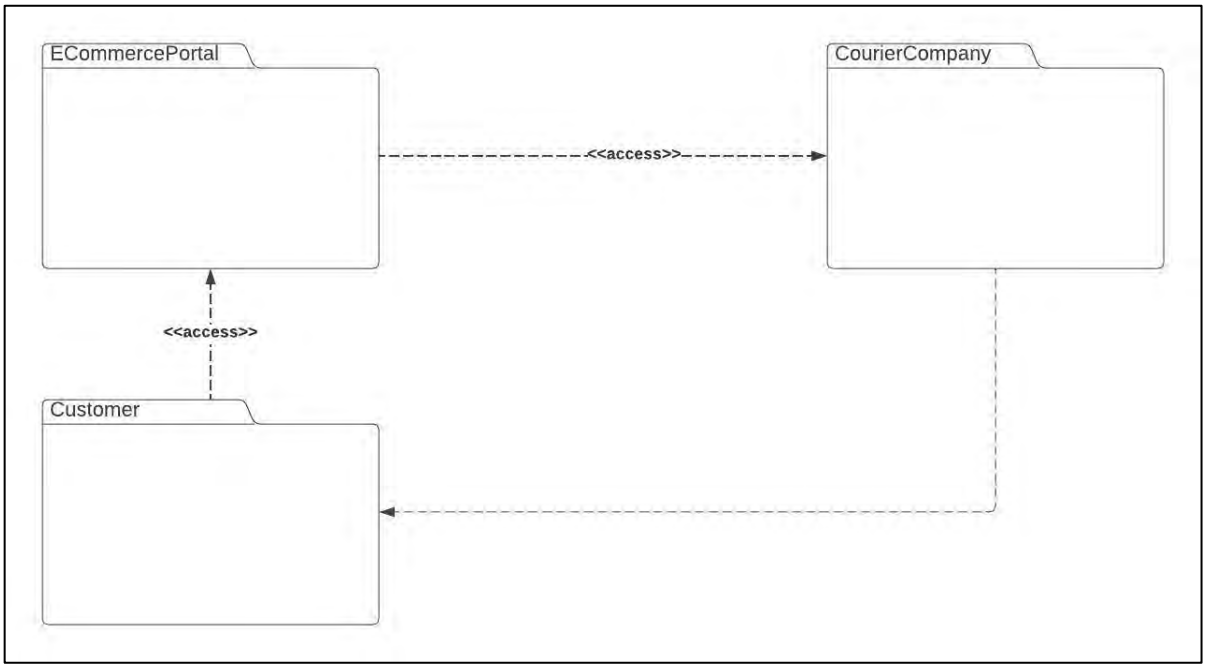
STATE DIAGRAM:



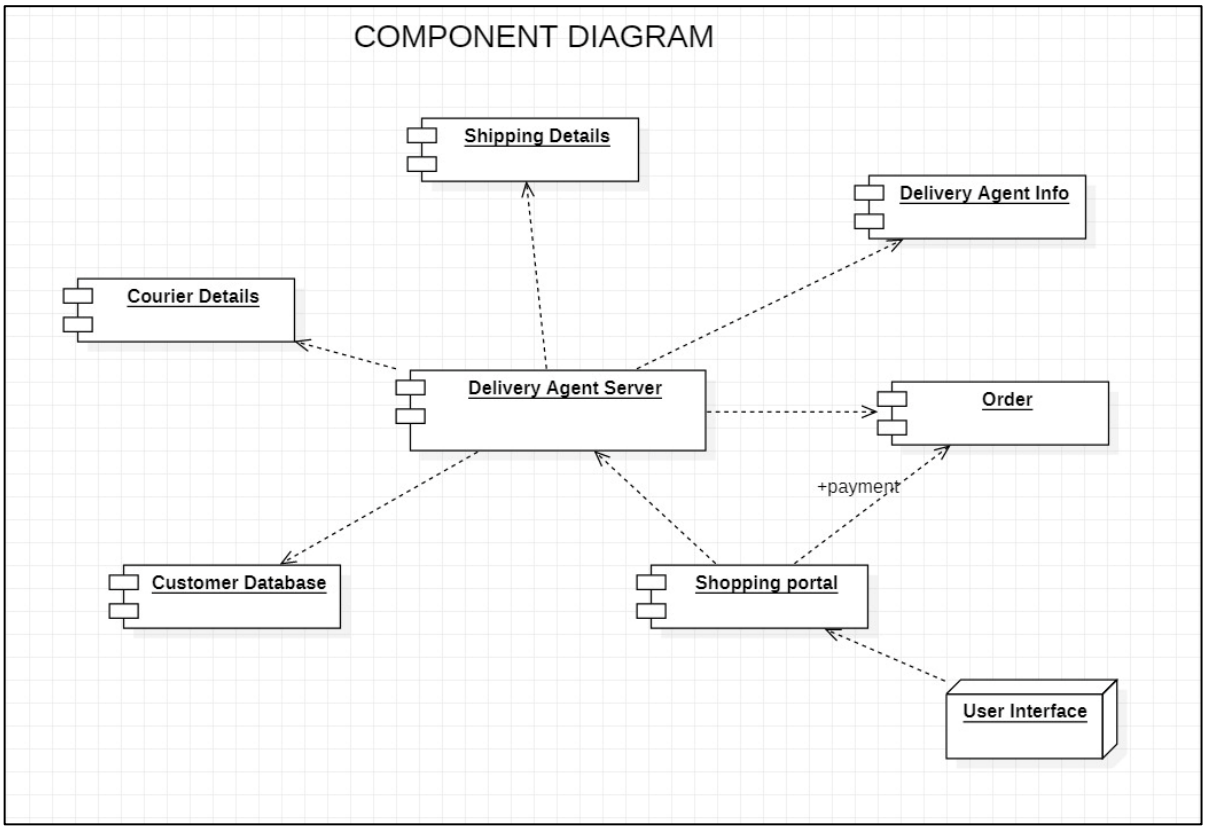
COLLABORATION DIAGRAM:



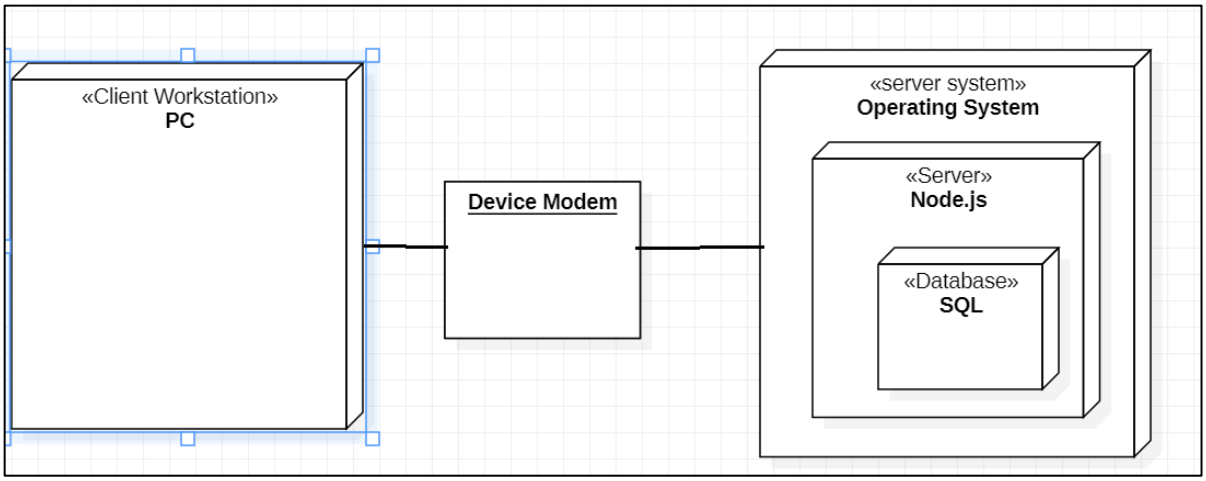
PACKAGE DIAGRAM:



COMPONENT DIAGRAM:



DEPLOYMENT DIAGRAM:



AIM III: TO DRAW THE UML FOR AN ONLINE CAMPUS SECURITY MANAGEMENT SYSTEM

THEORY:

XYZ Company 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 through which entire security system within the campus can be controlled in an efficient manner.

INPUTS:

- User Information
    - Security – (Name, Identity Number, Password)
    - Total number of security persons
    - Manager – (Name, Identity, Password)
- Place Information
    - Number of places identified by unique numbers

OPERATIONS:

- Security Person
    - Log-In
    - View duty date, place, start time, end time (upcoming 7 days schedule can be viewed)
    - Request manager to take leave or to do over duty
    - Request approved/declined
    - Number of leaves taken/ number of allowed leaves remaining
    - Log-Out
- Manager
    - Log-In
    - Create routine for upcoming 7 days for all persons considering leave requests
    - Approve/decline leave request
    - Monitoring
    - Log-Out

OUTPUTS:

- Salary at the end of the month
- View routine

CONSTRAINTS:

- All users MUST register themselves into the system.
- A security person can only check his/her own routine.
- Manager can check the status of all security persons.
- A fixed number of leaves are allowed. Beyond that, fixed amount will be deducted as fine.
- All security persons get same number of duties in a month.
- Adequate number of security persons are there to cover all places considering leave requests.
- A security person must not have duties in two different places at the same date and time.

CASE DIAGRAM:

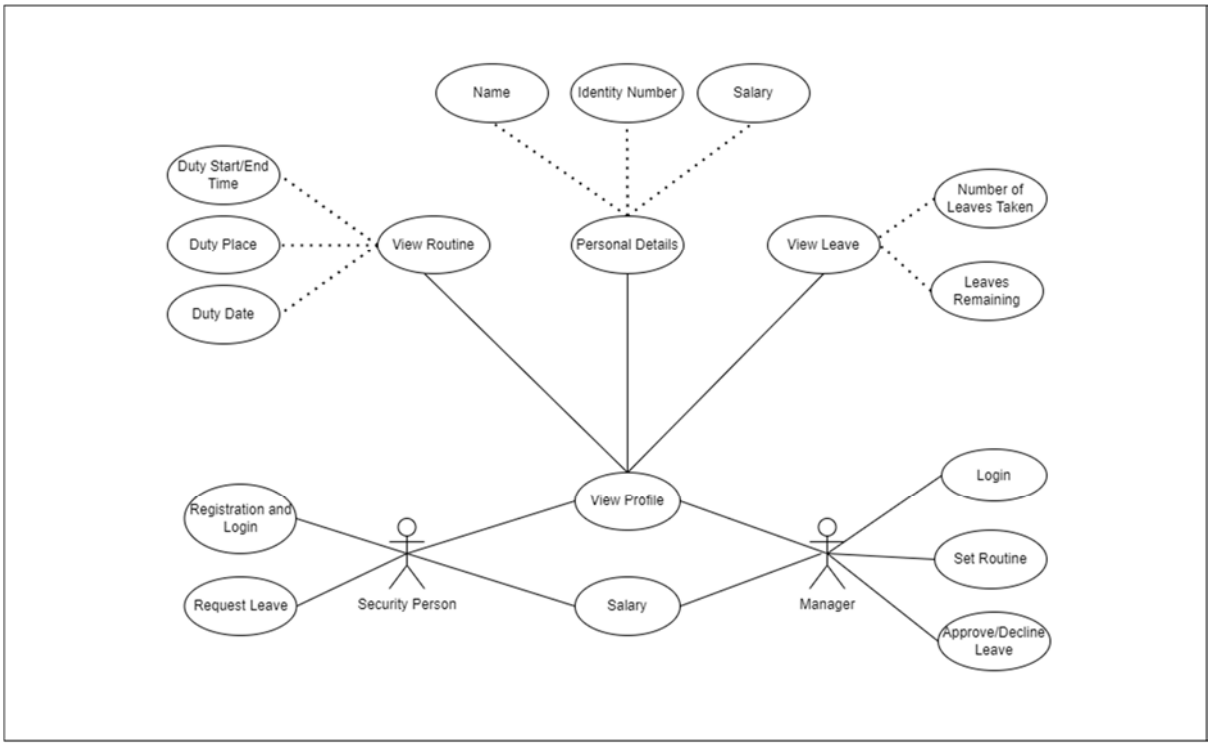


Fig 3.1: Case Diagram for an Online Campus Security Management System

EXPLANATION:

Here, we have two actors- Security Person and Manager. Manager manages the schedule and salary of the security person. Security person can access his routine containing details of his duty. Security person can also request leave in case of any occasion or emergency. All these are reflected in View Leave section and the salary is calculated accordingly.

CLASS DIAGRAM:

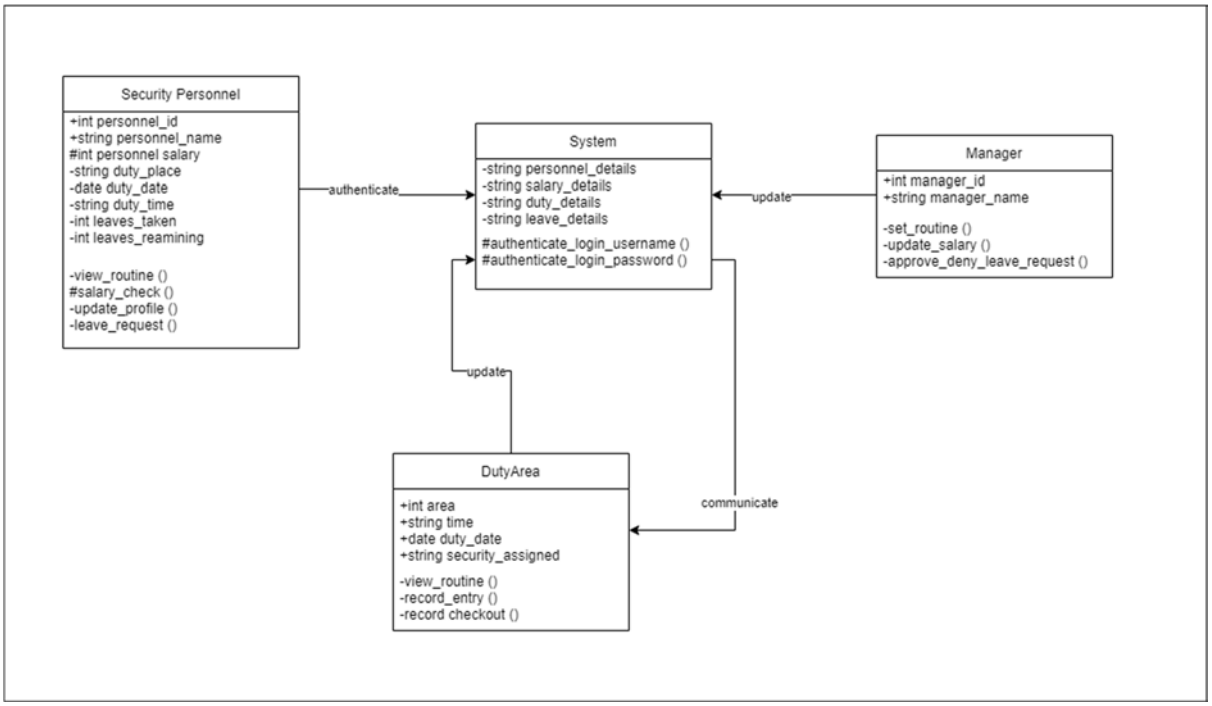
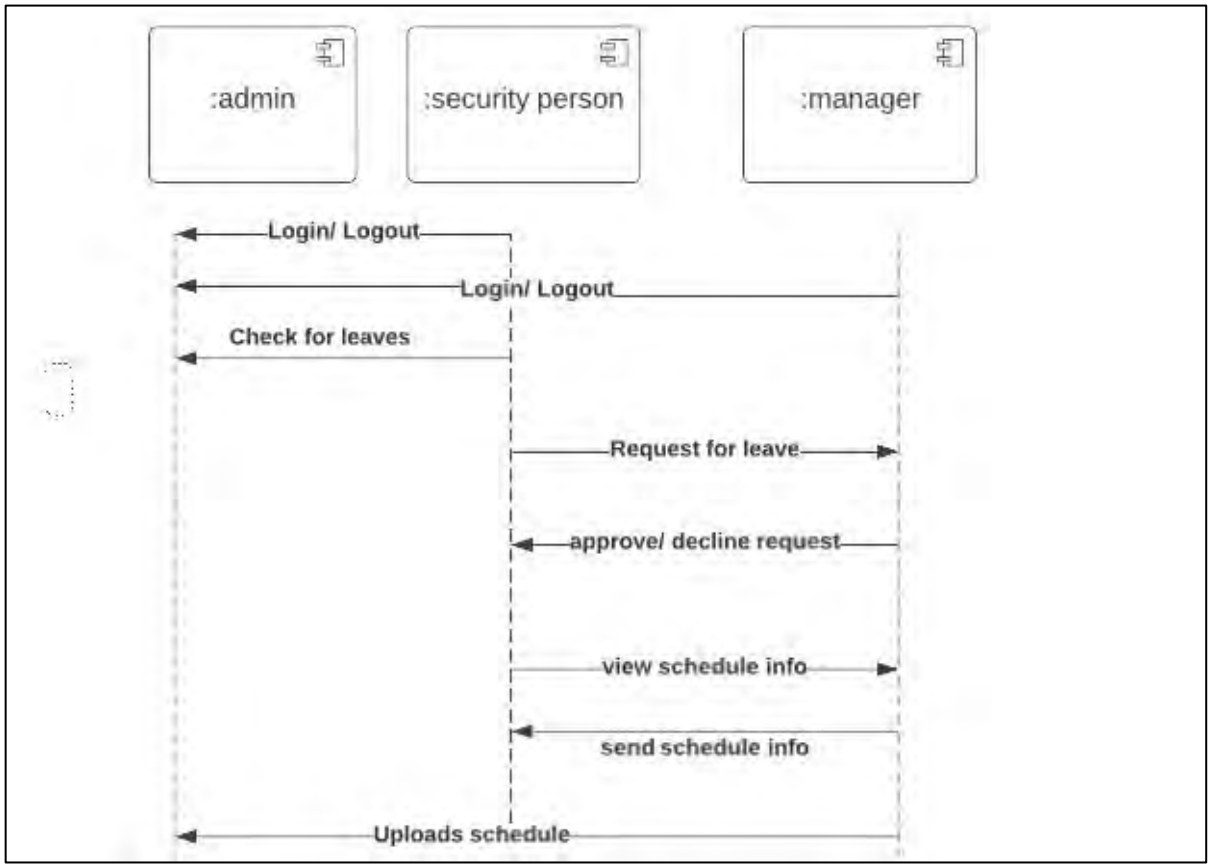


Fig 3.2: Class Diagram for an Online Campus Security Management System

SEQUENCE DIAGRAM:



STATE DIAGRAM:

COLLABORATION DIAGRAM:

PACKAGE DIAGRAM:

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM:

AIM IV: TO DRAW THE UML DIAGRAM FOR A HOSPITAL MANAGEMENT SYSTEM

THEORY:

Hospital Management System (HMS) is powerful, flexible, and easy to use system, which is very helpful in a hospital environment, in order to maintain a hospital efficiently. Hospital Management System designed for multi-specialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System (HMS) that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

In an HMS need to have entry of all the data related to patients, doctors, staffs, and administrator(s). Each patient admitted in the hospital needs to enter their personal information, which are accessible by authorized doctors and staffs. On the other hand, a patient should have access permission to download any medical report associated with him/her. Billing process should be included in the system, such that it can easily be update after each payment.

INPUTS:

- Hospital information (name, address, branch, department, facilities)
- Employee information (name, employee id, address, contact details)
- Doctor information (name, employee id, address, contact details, specialization, degrees)
- Patient information (name, id, address, age, payment, others)

OUTPUTS:

- Patient details
- Patient's list
- Receipt generation
- Report generation

CONSTRAINTS:

- All the users must register into the system though the web interface.
- All the users must login into the web interface through userid (must be unique) and password
- Patients should allow to get login him/her-self in order to collecting own information
- On login into the system, a doctor must be able to see the patients list under his/her observation
- One patient must not access the information of some other patients
- All the patients must be able to download all the medical report and bill as per his requirement
- by login into the system
- Add, delete, and update must be reflected immediately in the system

CASE DIAGRAM:

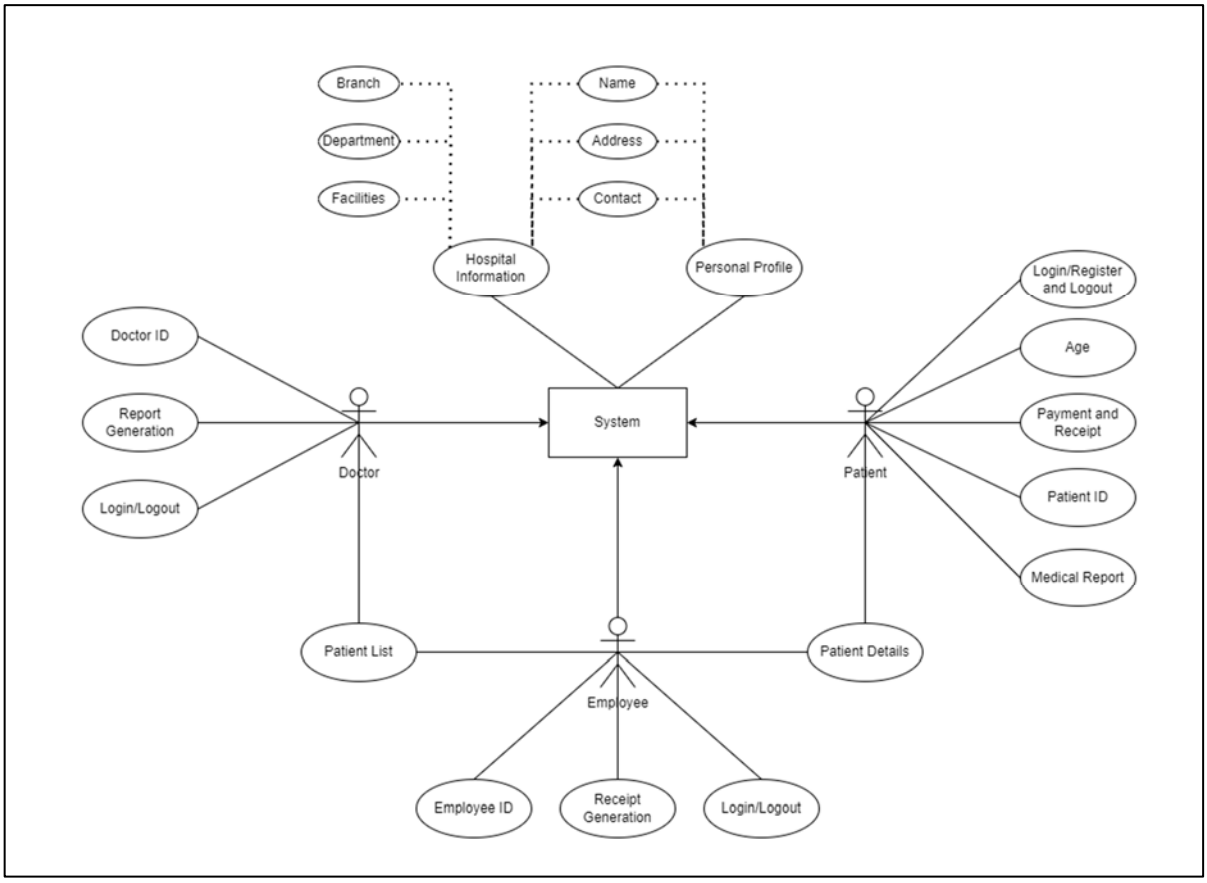


Fig 4.1: Case Diagram for a Hospitable Management System

EXPLANATION:

In this case diagram, we have three actors- doctor, patient and employee. The patient will see which doctors are available and make an appointment after making a payment. The employee of the hospital will take care of all these appointments and payment related activities, and accordingly update doctor's schedule. The doctors will be able to see their schedule, their appointments and the patient details as well. After the check-up, the doctor will generate a medical report which can also be viewed by the patient.

SEQUENCE DIAGRAM:

STATE DIAGRAM:

COLLABORATION DIAGRAM:

PACKAGE DIAGRAM:

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM:

CLASS DIAGRAM:

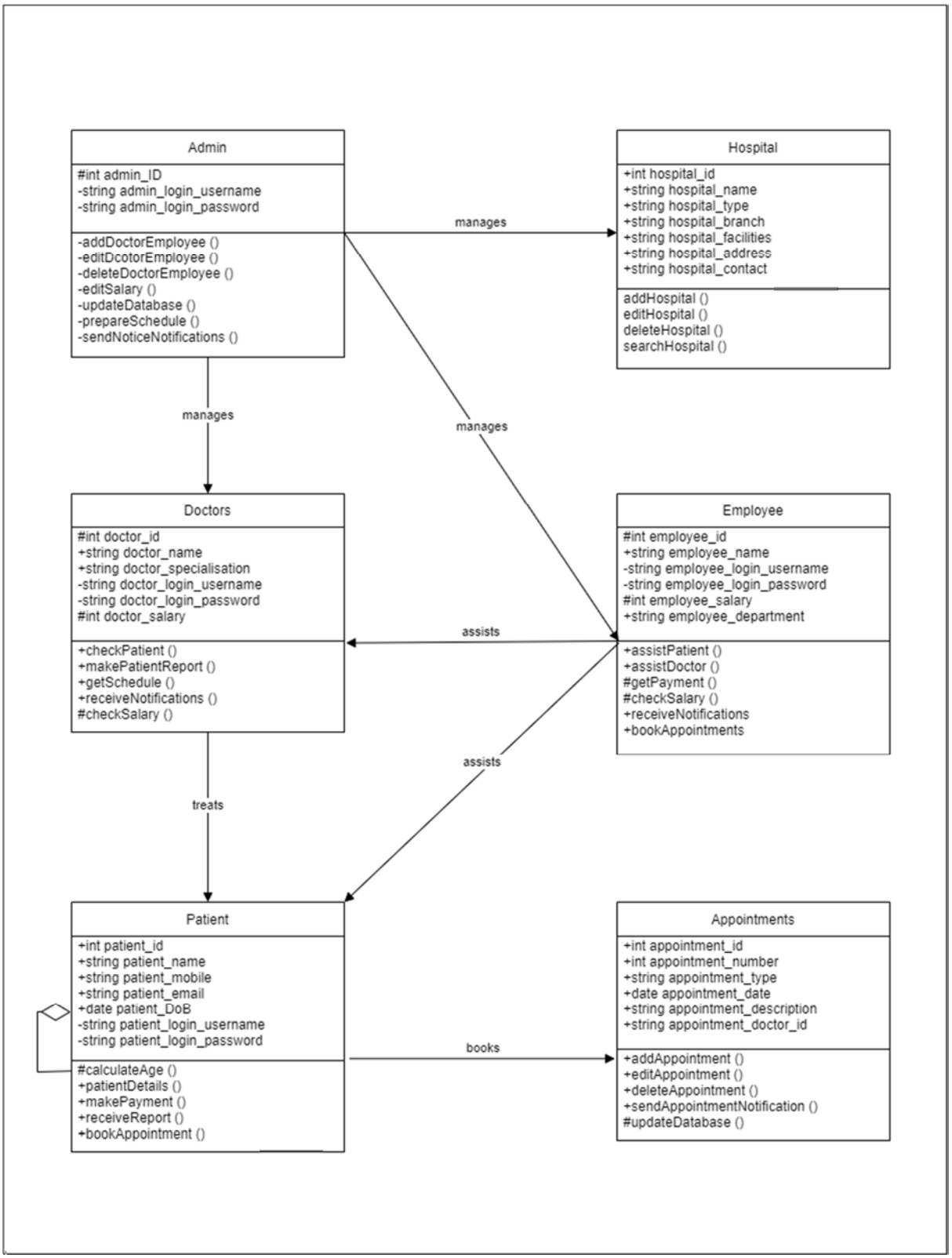


Fig 4.2: Class Diagram for a Hospitable Management System

AIM V: TO DRAW THE UML DIAGRAM FOR A PAYROLL MANAGEMENT SYSTEM

THEORY:

The Employee and Payroll Systems objective is to provide a system which manages the employee details, the payroll activity done in a company depending upon the employee's attendance and its calculation which is very huge. The users will consume less amount of time through computerized system rather than working manually. The system will take care of all the payroll activities like managing each employee's attendance, the number of leaves taken by that particular employee and calculation in a very quick manner and it avoids.

Data storing is easier. Paper work will be reduced and the company staffs spend more time on monitoring the progress. The system is user friendly and easy to use. All the important data's will be stored in the database and it avoids any miscalculation.

The "Employee and Payroll System "is based on maintaining each employee records and calculating his/her salary depending on the workdays. The first activity is based on saving the employees details where each employee will be given a unique Employee ID. Now based on the no of days an employee attended per month, salary will be calculated by checking the no of workdays of a company and his/her basic salary and a separate salary slip will be provided for reference.

INPUTS:

- Employee details (employee no, name, address, designation, department, achievements)
- Accounts details (salary of each employee, deduction, TA, DA, HRA, other allowance, PF)
- Leave information (no of leave taken by each employee)

OUTPUTS:

- Salary slip
- Detailed salary report
- Deduction details
- Leave information

CONSTRAINTS:

- Admin must have the permission to update the employee information manually
- Manager should be able to see the details of an employee
- Salary slip, detailed salary report, deduction details, and leave and information of a particular employee must not be accessible to another employee
- The leave information must be update automatically, if an employee takes leave.

CASE DIAGRAM:

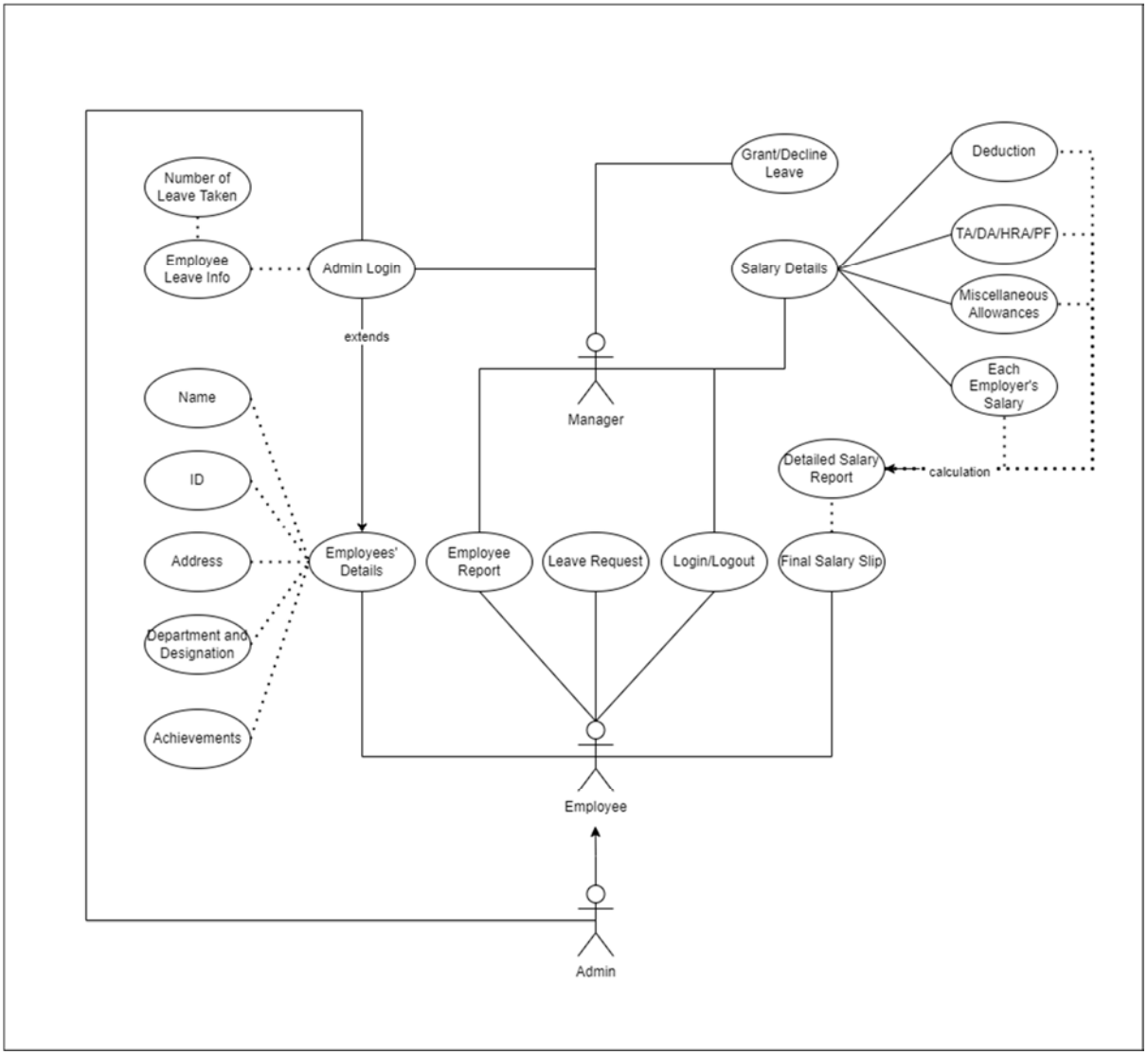


Fig 5.1: Case Diagram for a Payroll Management System

EXPLANATION:

Firstly, we have three actors- manager, employee and admin. Similar to Online Campus Security Management System as discussed in the previous experiment, the manager can see employee's details and achievements. The employee can request for leave and the manager can either grant or deny the request. The manager can also view the number of leaves taken by an employee. The manager then calculates the salary of the employee after taking in account the deductions, TA/DA/HRA/PF, miscellaneous allowances etc. and produce a detailed salary report. The employee can view his final salary slip. Admin also is an employee with additional access to employees' leave information and employees' details, which can be edited and modified by the admin.

CLASS DIAGRAM:

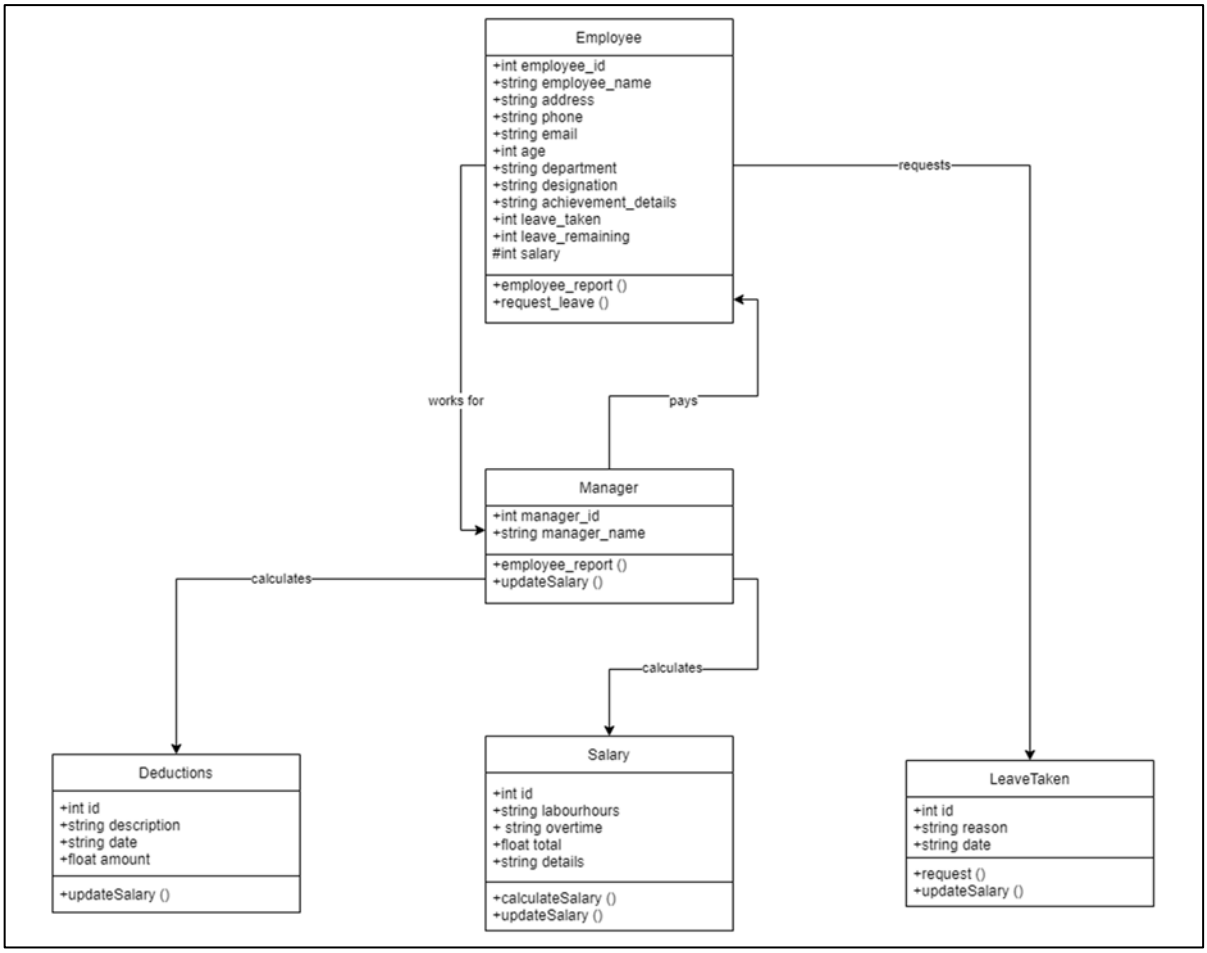


Fig 5.2: Class Diagram for a Payroll Management System

SEQUENCE DIAGRAM:

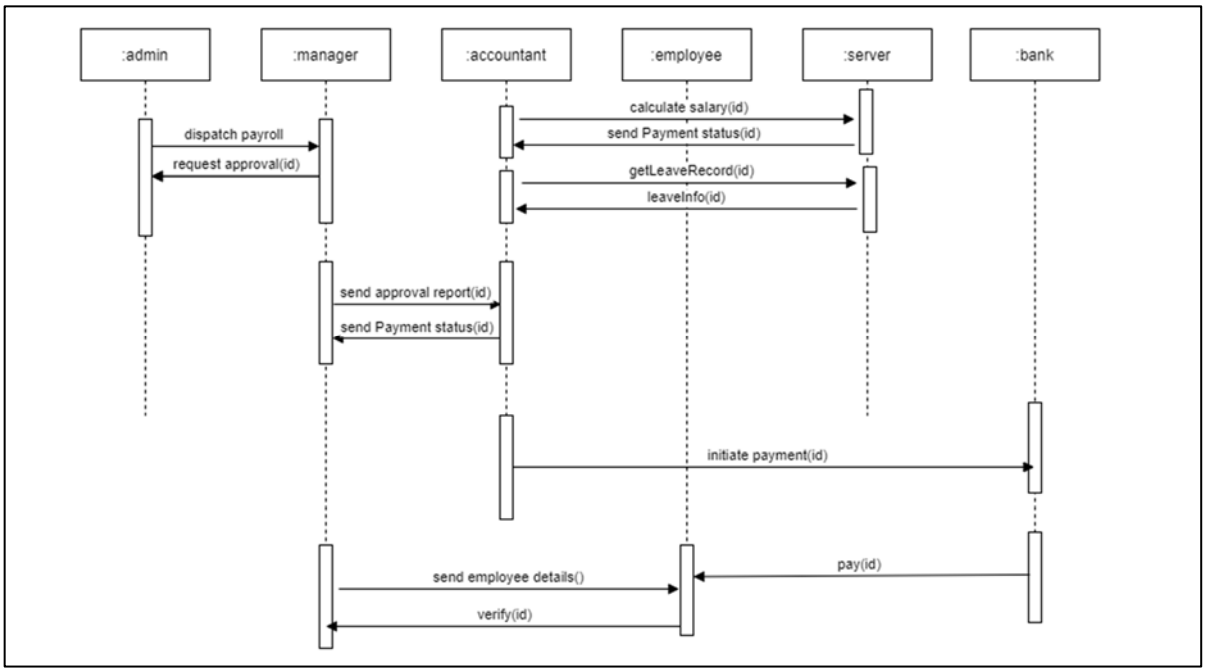


Fig 5.2: Sequence Diagram for a Payroll Management System

STATE DIAGRAM:

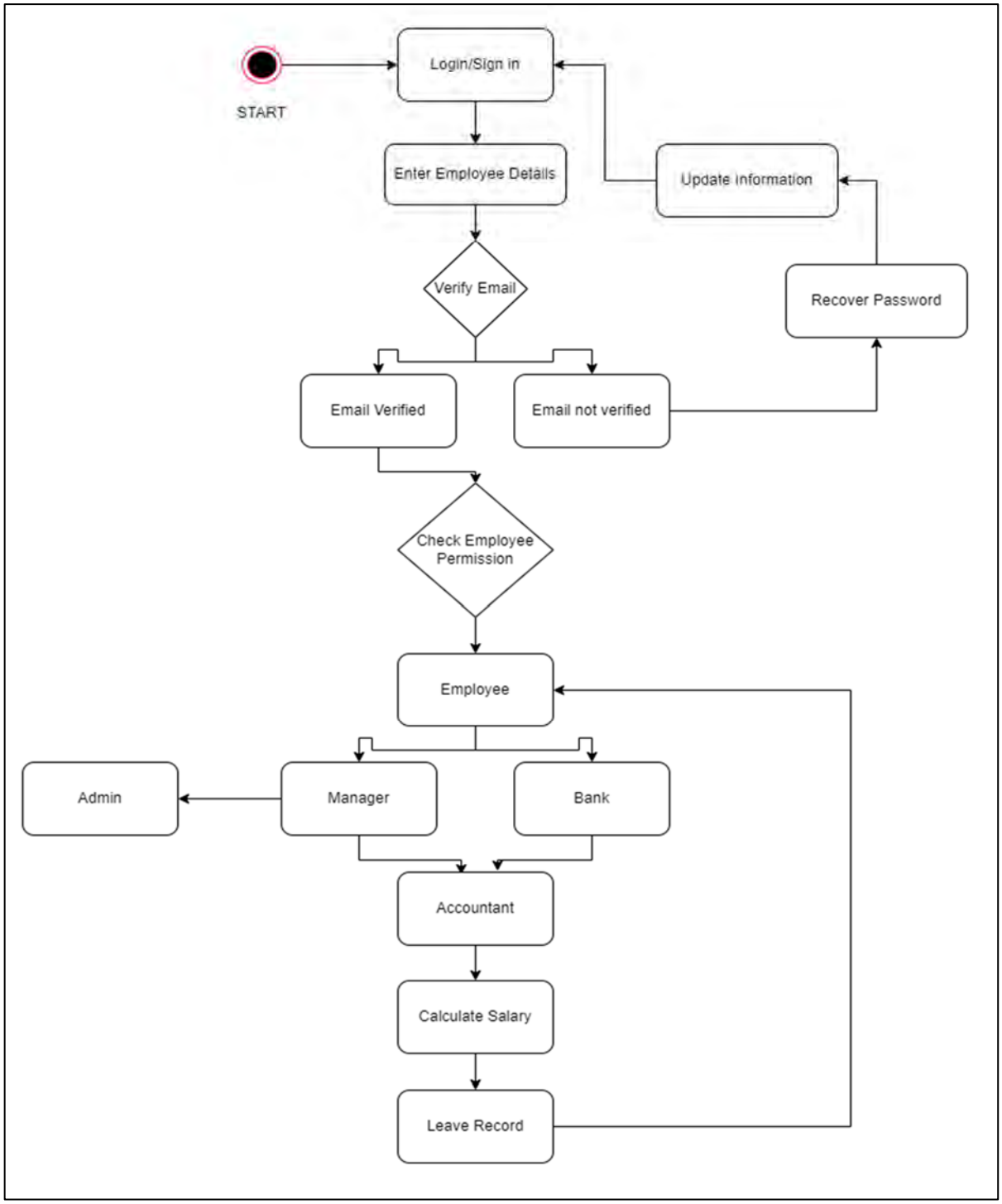


Fig 5.3: State Diagram for a Payroll Management System

COLLABORATION DIAGRAM:

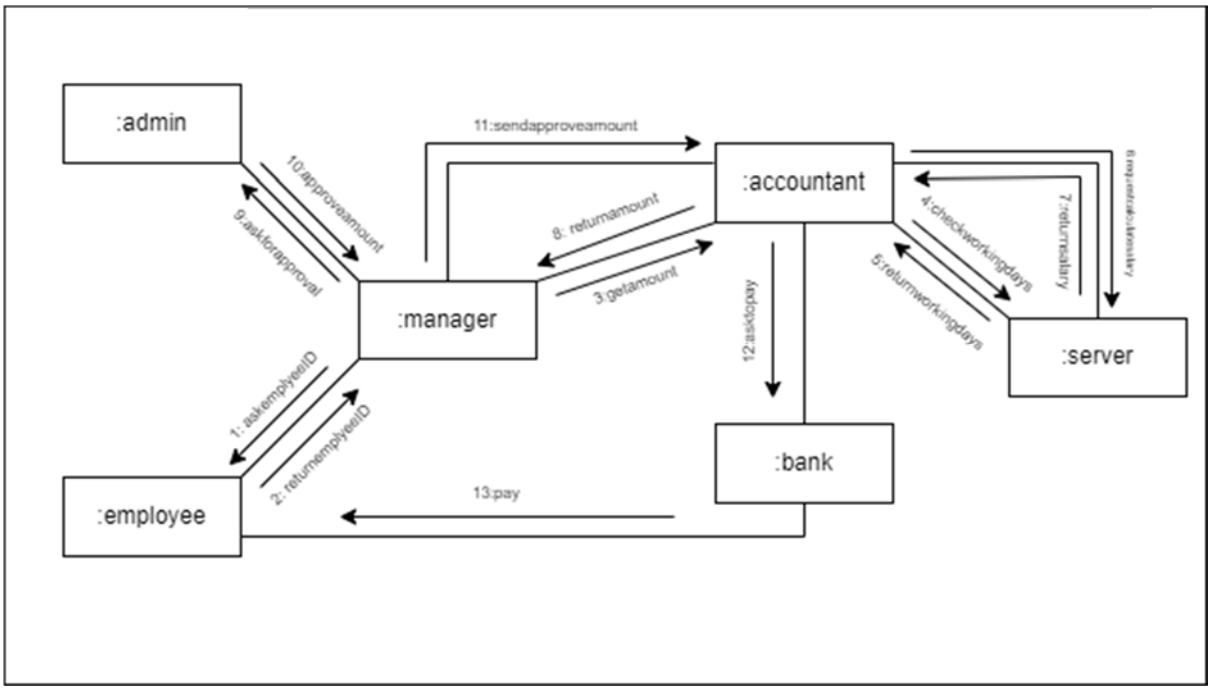


Fig 5.3: Collaboration Diagram for a Payroll Management System

PACAKGE DIAGRAM:

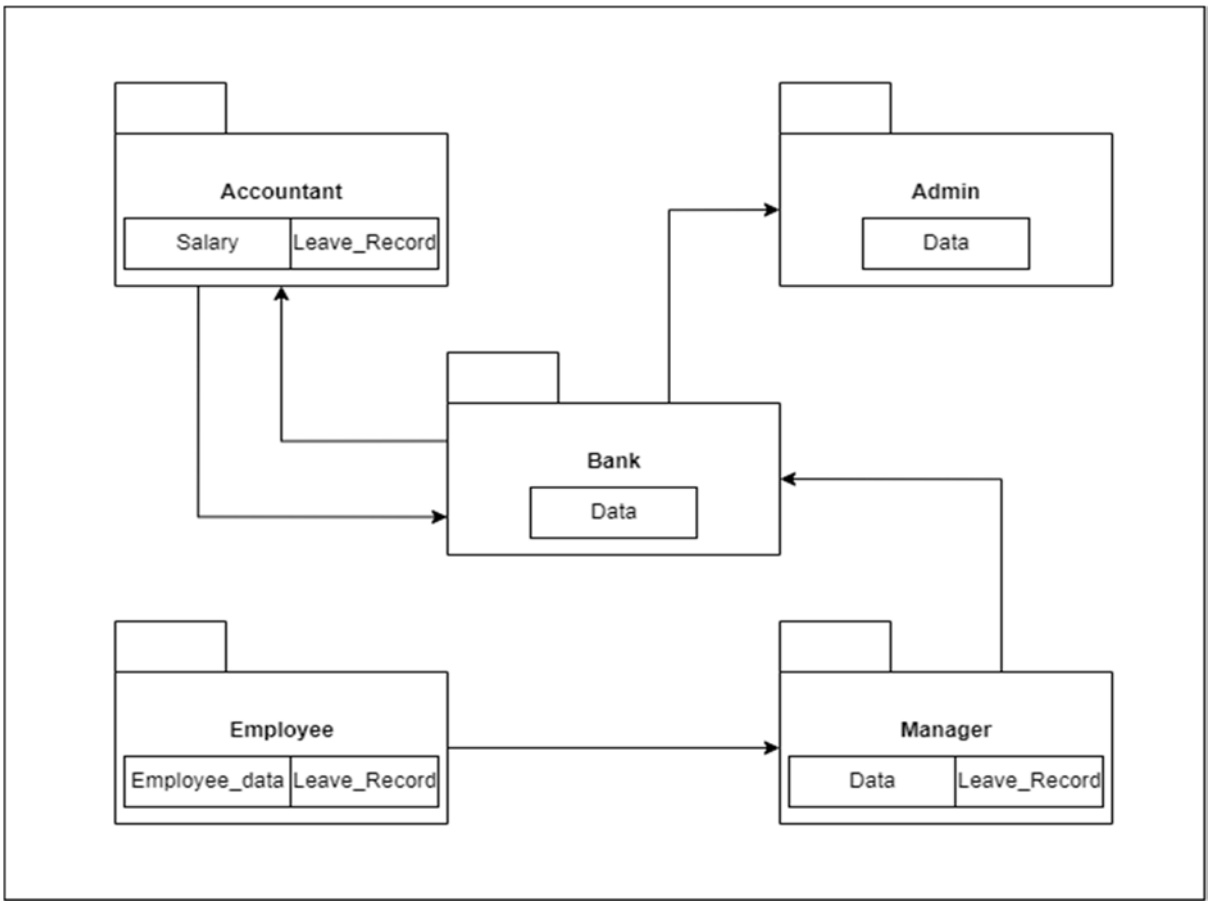


Fig 5.3: Package Diagram for a Payroll Management System

COMPONENT DIAGRAM:

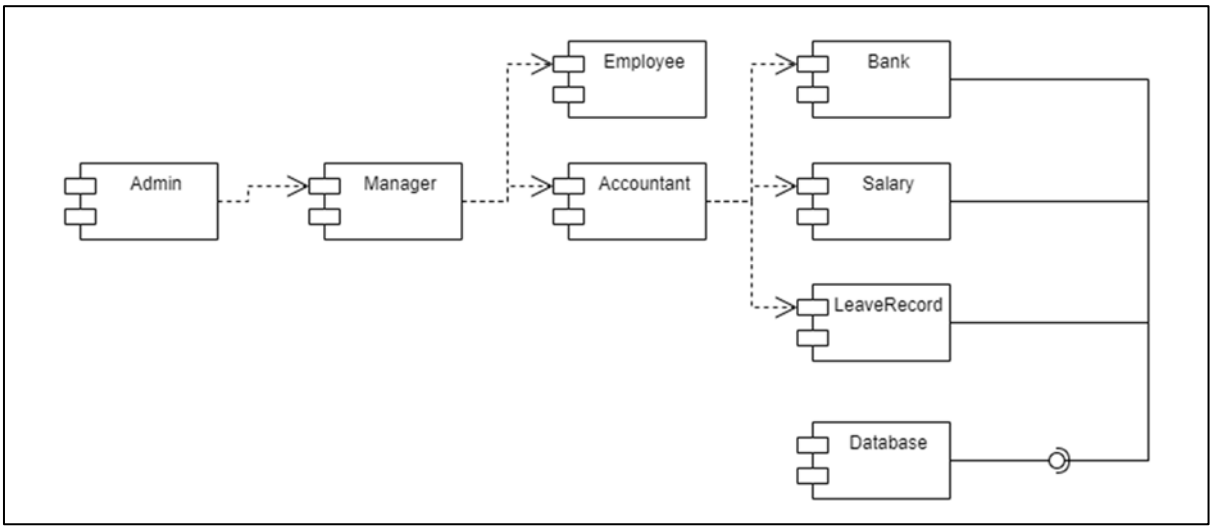


Fig 5.3: Component Diagram for a Payroll Management System

DEPLOYMENT DIAGRAM:

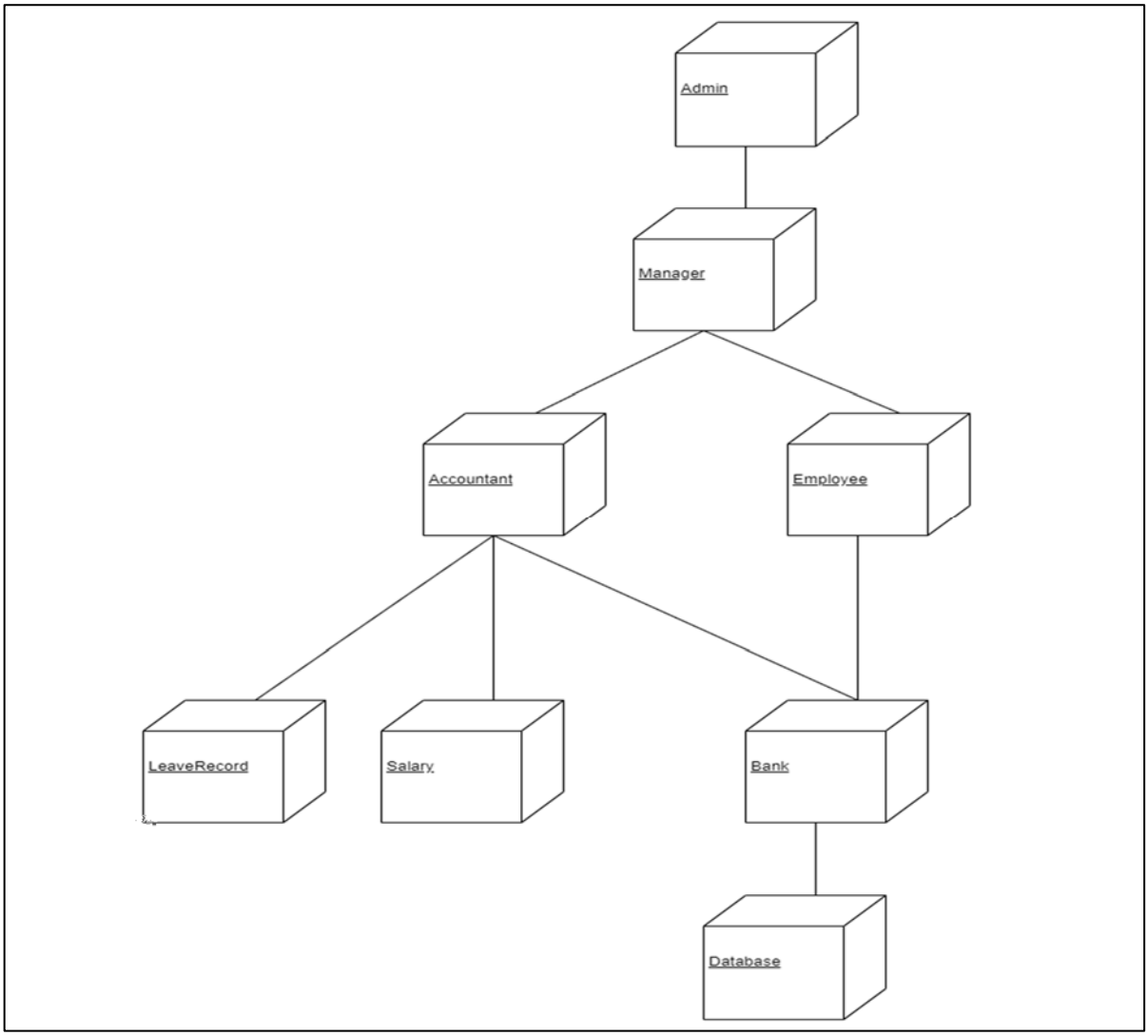


Fig 5.3: Deployment Diagram for a Payroll Management System

AIM VI: TO DRAW THE UML DIAGRAM FOR A SMART HOME MANAGEMENT SYSTEM

THEORY:

"Smart Home" is the term commonly used to define a residence that has smart devices which are able to communicate with the central controller. The smart home management system is a system which enabled the automation of smart home. Typically, it includes following Four Components:

- **Electronics Devices and Sensors:** It includes devices such as air conditioning, TVs, computers, entertainment audio & video systems, and camera systems. In addition to that, it can include sensors (temperature, humidity, light, and sound) for environment sensing.
- **Wireless network:** It is a medium of communication between the central controller and devices/sensors. E.g., Wi-Fi.
- **Central Controller:** The SHMS can operate in two modes (automatic and manual). In automatic mode, it receives streaming data from sensors and on/off the appliances according to the User Environmental Comfort range. However, in manual mode, it on/off the appliances according to the user request.
- **User Interface (e.g., Android App):** It enables user and administrator to sign up and login before availing the services. After login user can give its Environmental Comfort range as per the offered services. The user can also see the available appliances and change its status (on/off) as per the preferences. There should be an option of registering a complaint in the case of system failure. After login the administrator can add new devices/sensors to the existing system. In addition to that he/she gets notified regarding the registered complaint. The billing calculation should be based on the energy consumed by the devices/ sensors. Moreover, the user should be notified through SMS in case of a general event (when operating in manual mode) and through SMS and Call in the case of any emergency.

INPUTS:

- User Personal Information: (name, email, age, gender, mobile number, security question)
- User Environmental Comfort range (Temperature, Humidity, Lighting, sound)
- System Administrator Details: (mailed, contact no.)
- Operating Mode (Automatic or manual)
- Appliances Information (Type, Power, Unique Id, Status)
- Sensor Information (Type, Id, operating range)

OUTPUTS:

- Electricity Billing Details.
- Complaint and Feedback
- Alert by SMS
- Alert by SMS and Call, in case of Emergency

CONSTRAINTS:

- User/ Administrator must register with the SHMS to access the portal.
- The mobile number must be a 10-digit number.
- Login information consists of email address and password
- Each appliance and sensors assigned a unique ID
- User environmental comfort range must be in the operating range of sensors.
- Enable power saver mode only if operating mode is automatic.
- In manual mode, every time the operating status of appliances change system must show the current billing status.
- Every time user logins system must show current environmental condition.

CASE DIAGRAM:

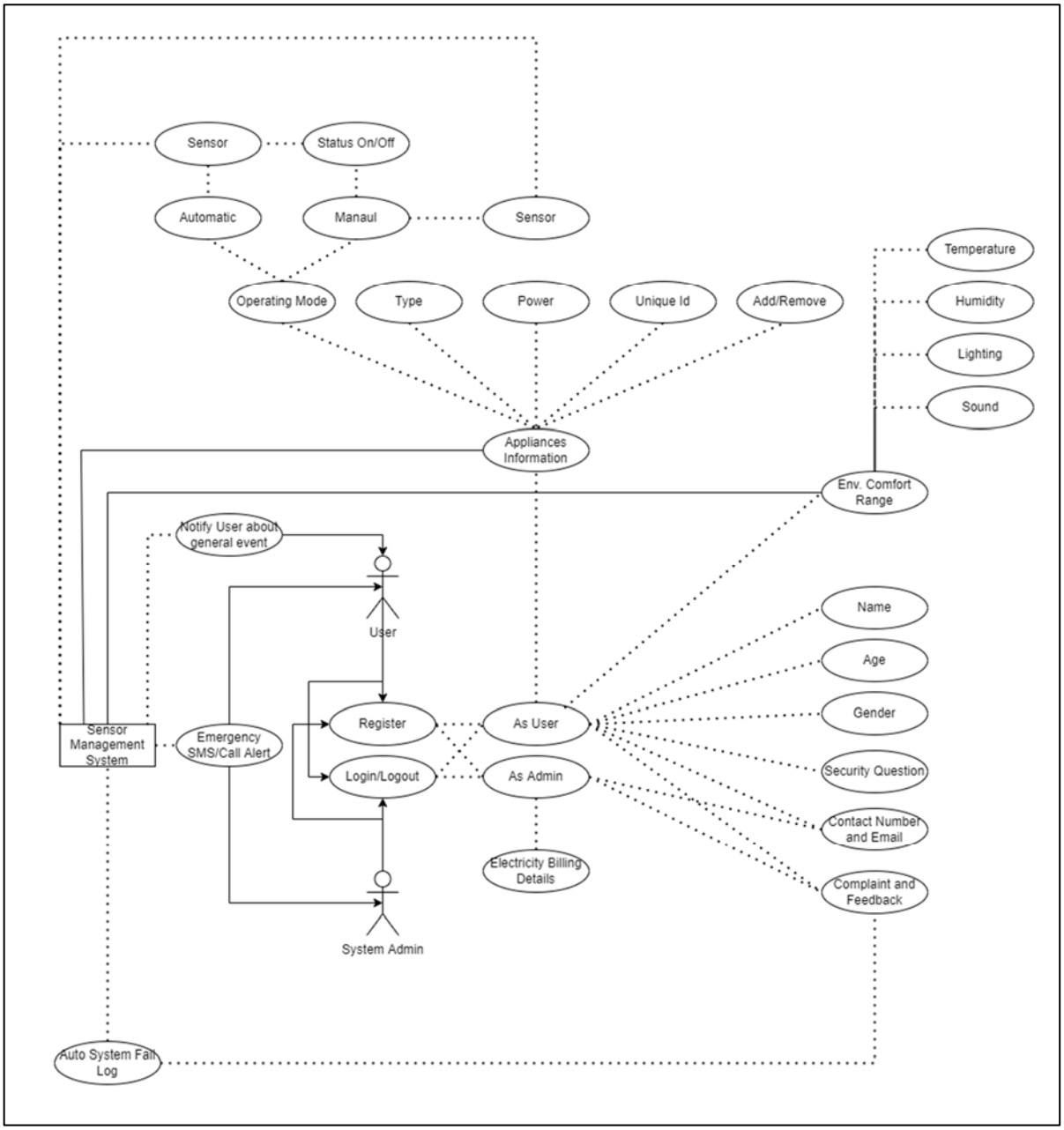


Fig 6.1: Case Diagram for a Smart Home Management System

EXPLANATION:

There are two actors- User and System Admin, and one sensor management system. The user can set the appliances information, select the operating mode for each appliance, add or remove the appliances from the smart home network. The user can also adjust the environment comfort range. Moreover, the user can issue complaint and feedbacks to the system admin. The system admin can see the complaints and feedbacks and respond to those. In addition, the system admin takes care of calculating and checking electricity billing details. The sensor management system has access to all the appliances details that the user has entered and the user's comfortable environment range details as well. In case of automatic operating mode, the sensor takes over the appliance management control. In case of system failure or other emergencies, the sensor system will immediately notify both the user and the system admin through SMS and an automated phone call. In addition, the sensor management system can also register automated feedback to the system admin in case of minor system fail.

CLASS DIAGRAM:

SEQUENCE DIAGRAM:

STATE DIAGRAM:

COLLABORATION DIAGRAM:

PACKAGE DIAGRAM:

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM:

AIM VII: TO DRAW THE UML DIAGRAM FOR AN ONLINE EXAMINATION SYSTEM

THEORY:

Now-a-days, Online examination system has become popular for competitive examinations because of its unique features such as auto-evaluation, speed and accuracy. Moreover, it also helps environments by reducing the use of paper. In such a system, students are asked to select answers from multiple options given for a single question. Likewise, there are several questions which appear in the students' systems. The questions and multiple options are saved in a database along with desired answers. Typically, a student can edit an answer after saving it, however, editing cannot be done after submitting the answer. Another user is also there – administrator. The administrator can create, modify and delete questions and accordingly, the question is updated in the system.

INPUTS:

- User Information with Code, so that all subjects can be identified using unique codes.
- User Information
- If Student- Student Information (Name, Roll No, Email Address, Contact Number, Password)
- If Administrator (Email Address, Password)
- Set of Questions with multiple answers for each stored in a database along with desired answers.

OPERATIONS:

- Administrator
    - Log-In
    - CREATE, MODIFY or DELETE questions. Accordingly, the question set must be updated.
    - Log-Out
- Student
    - Log-In (Time starts)
    - Answer the questions – SAVE and SUBMIT
    - Log-Out (Automatically logged out after Timeout)

OUTPUTS:

- Display the result in DESCENDING order according to obtained marks with Roll Number.
- The result is also saved into a database for future use.

CONSTRAINTS:

- All users MUST register themselves into the system.
- Same set of questions should appear to all students.
- Log-In information contains only Email Address and Password.
- After clicking on SUBMIT, selected answers cannot be changed.
- After clicking on SAVE, selected answers can be changed.

CASE DIAGRAM:

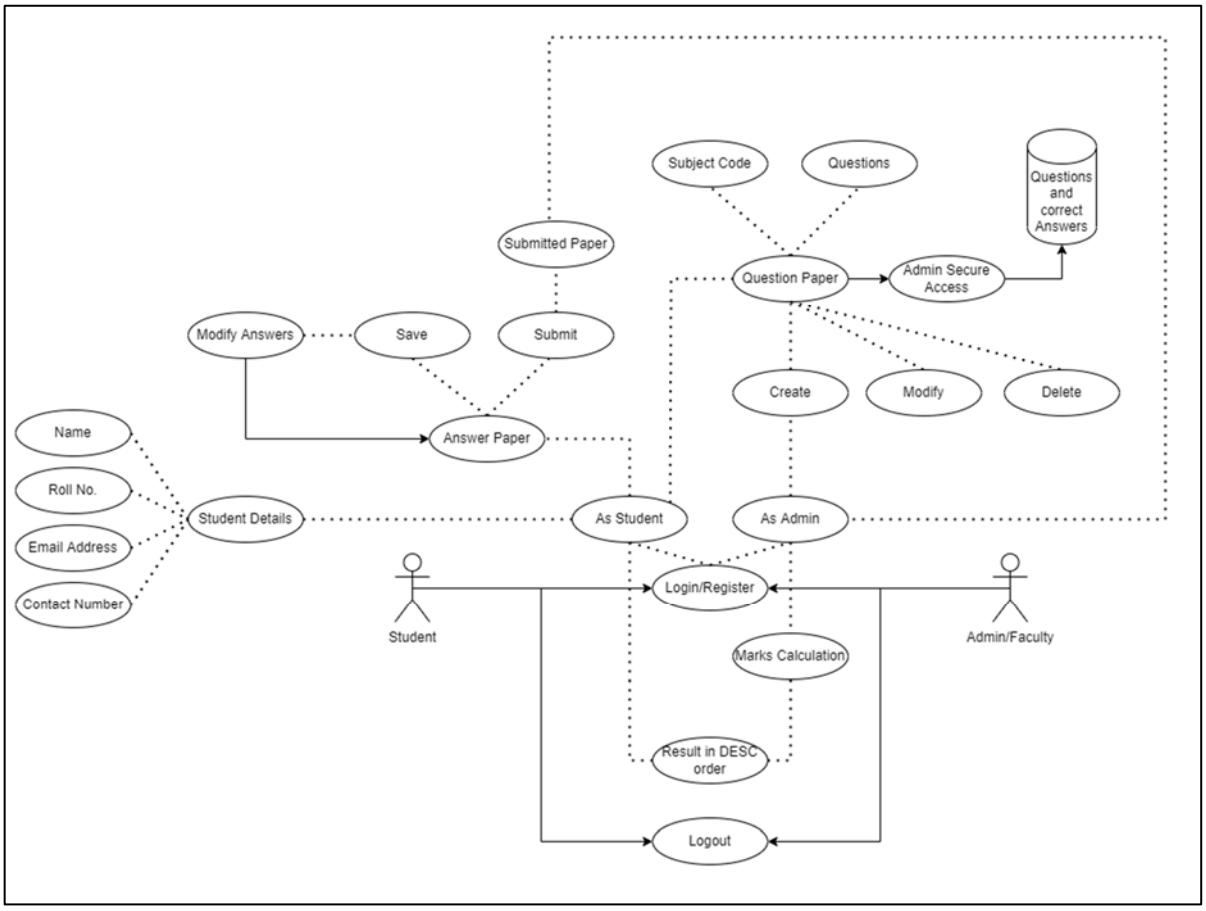


Fig 7.1: Case Diagram for an Online Examination System

EXPLANATION:

In online examination system, we have only two actors- student and faculty, and a database system that stores question paper and corresponding correct answers. The faculty is an admin themselves. The faculty can create, modify or delete a question or a question paper. Through admin secure access, the faculty can update correct answers to the questions in the database system. The student can then view the question paper and fill in the answer paper accordingly. The student can either save or submit the answer sheet. Once submitted, the answers on the answer paper cannot be modified, and the answer paper will be visible to the faculty for checking. The faculty then evaluates the answer paper and calculates the marks. The final published result is then visible to the students as well, in descending order.

CLASS DIAGRAM:

SEQUENCE DIAGRAM:

STATE DIAGRAM:

COLLABORATION DIAGRAM:

PACKAGE DIAGRAM:

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM:

AIM VIII: TO DRAW THE UML DIAGRAM FOR AN ONLINE TOLL PLAZA SYSTEM

THEORY:

Now-a-days, cashless transaction is becoming popular among the users because it is easy to handle, and it does not require to carry cash in hand. Typically, in India, road tolls are collected from cars manually for which the cars need to stop to pay the toll fee. In contrast, the objective is to make the system Online, so that the toll fee is automatically deducted from the user. Therefore, users credit their Online account (consider this as e-Wallet), and money is automatically deducted when the cars pass the toll system. As a result, the users do not have to wait for manual toll fee payment. Concurrently, administrator can also view all transactions from anywhere. Finally, the administrator can view the total income in a day-to-day basis, and can also analyse the traffic pattern as well.

INPUTS:

- User Information (Name, Car Number, Email Address, Password, Money in e-Wallet)
- Administrator Information (Email Address, Password)

OPERATIONS:

- Patient
    - Log-In
    - Credit in e-Wallet
    - Check e-Wallet Balance
    - Log-Out
- System
    - Check the car number
    - Required Fee Available
        - Allow the car to pass
        - Deduct money from e-Wallet
    - Required Fee NOT Available
        - Do Not Allow the car to pass
        - Fee Payment is done manually
    - Allow the car to pass
    - Total Income is stored in a database
- Administrator
    - Log-In
    - View transactions
    - View total income

OUTPUTS:

- Display day-wise transactions to administrator
- User can view his/her own transactions

CONSTRAINTS:

- All users MUST register themselves into the system with their car numbers.
- A user can only view his/her own transactions
- Administrator can view all transactions

CASE DIAGRAM:

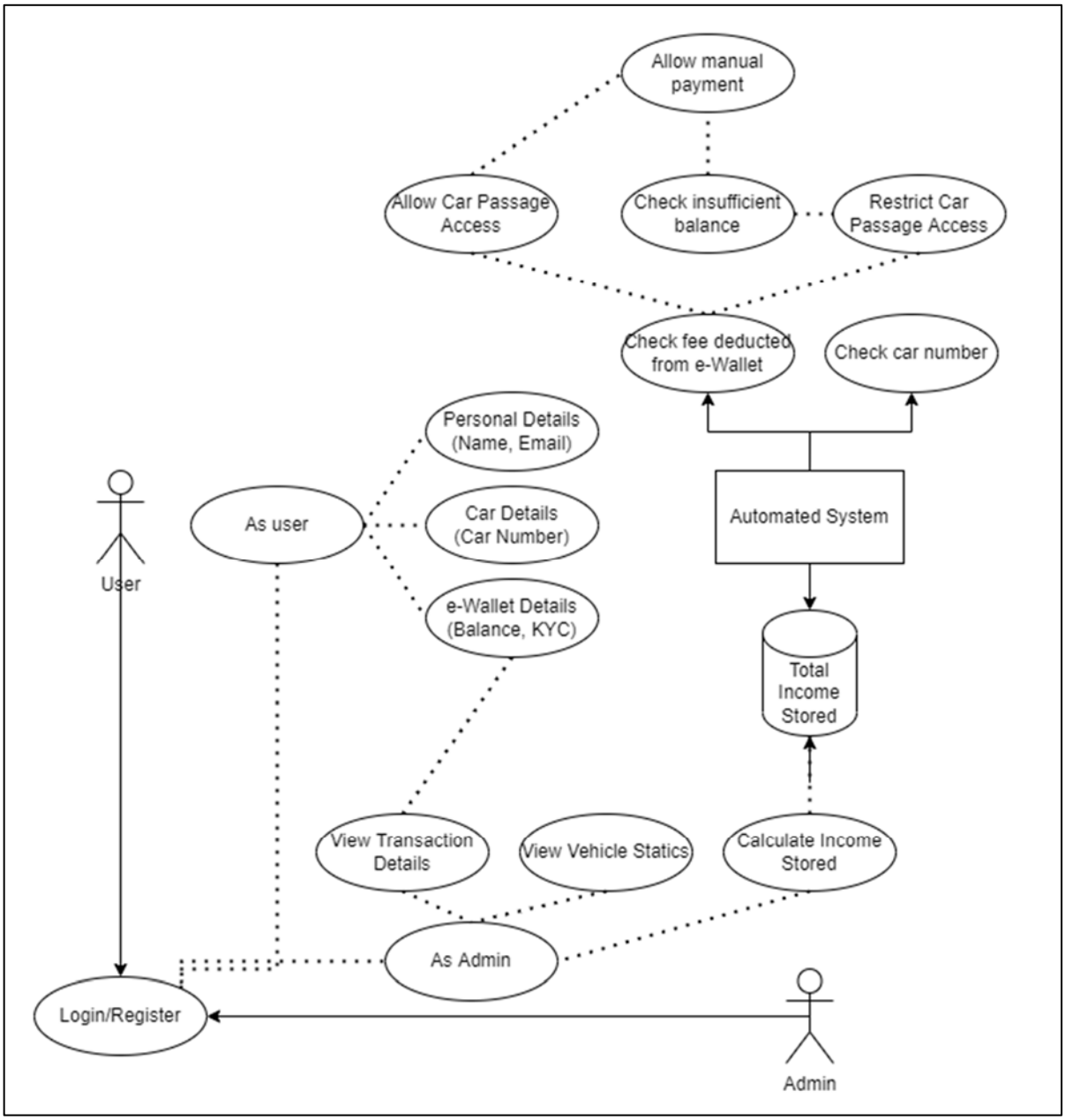


Fig 8.1: Case Diagram for an Online Toll Plaza System

EXPLANATION:

The online toll plaza system has two actors- user and admin, and an automated system, as well as a database system to store the total income. The user updates personal details, vehicular details and e-Wallet details. The admin then checks vehicle status and calculates income stored. The automated system reads values from the car details to recognise the car and checks e-Wallet transaction details to decide whether to grant passage access to the vehicle or not. If the fee deducted from the e-Wallet is the sufficient amount, the passage is granted to the vehicle, but if the fee deducted is not sufficient, it allows for the user to pay the remining amount manually and then grants passage access to the vehicle. All this transaction details are then updated in the database by the automated system.

CLASS DIAGRAM:

SEQUENCE DIAGRAM:

STATE DIAGRAM:

COLLABORATION DIAGRAM:

PACKAGE DIAGRAM:

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM:

AIM IX: TO DRAW THE UML DIAGRAM FOR AN ONLINE HEALTH MONITORING SYSTEM

THEORY:

"Smart Online health monitoring or Online patient monitoring system (OHMS) is a promising technology to enable patient monitoring outside the conventional clinical system, i.e., the patient can be monitored remotely. Consequently, such system increases the access to care the patients and decreases the delivery cost related to healthcare.

Typically, in OHMS, two types of users are there – doctors and patients. Different physiological parameters of the patient are monitored (using sensors), and the monitored data is stored in a server. The stored data is accessible from anywhere through user authentication. On the other hand, doctors can check the health status of a patient registered with the doctor. Therefore, the doctor can only access the physiological data of a patient if and only if he/she is registered with the doctor. Depending on the monitored values, adequate measures can be taken by the doctors. The patient can also view his/her health status. For authenticity, both the users need to login into the system.

INPUTS:

- User Information
- Doctor – (Name, Email Address, Password)
- Patient – (Name, Email Address, Password, DOB)
- Predefined Sensors (such as temperature, blood pressure and heart rate)

OPERATIONS:

- Patient
    - Log-In
    - View health status
    - Ask doctor(s) to consult
    - Payment (Consultancy Fee)
    - Log-Out
- Doctor
    - Log-In
    - Monitor health status of registered patients to him/her
    - ◦ Ask patient(s) to consult
    - Log-Out

OUTPUTS:

- Display health status
- Consult with doctors/patients

CONSTRAINTS:

- All users MUST register themselves into the system.
- Age of patients automatically calculated using DOB information.
- Doctors can access the health information of patients registered to them only.
- Nobody can modify the stored information.
- A patient can only view his/her health status.

CASE DIAGRAM:

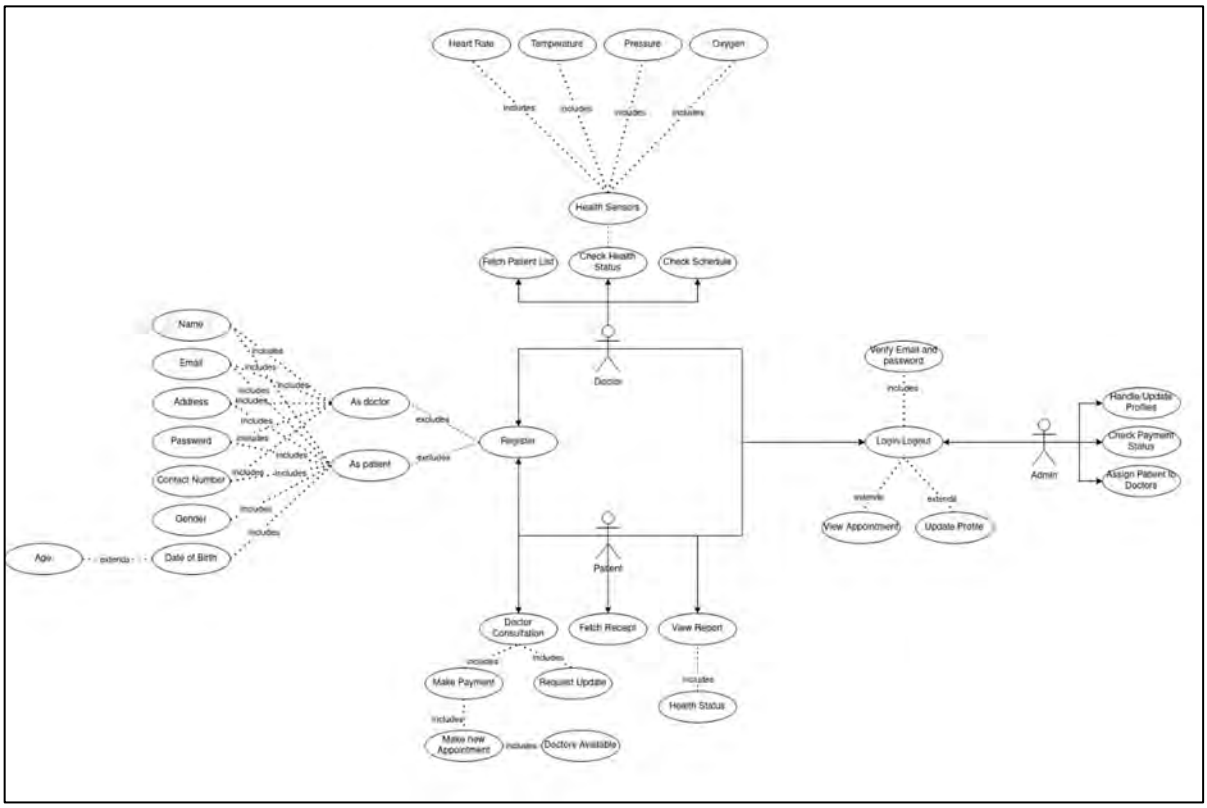


Fig 9.1: Case Diagram for an Online Health Management System

EXPLANATION:

Firstly, we have three actors- patient, doctor and admin. The admin's work is to handle and update the profiles of both the patient and the doctor, check payment status and assign patients to the doctors. As a doctor, they have access to the patient's health status through the health sensors that provide live update on patient's heart rate, temperature, pressure, oxygen level and many more. The doctor can fetch patient list and check their own appointment schedule. The patient can request doctor consultation by checking the doctors' schedule or which doctors are available and register an appointment after making the payment. The health report updated by the doctor can be viewed by the patient.

CLASS DIAGRAM:

SEQUENCE DIAGRAM:

STATE DIAGRAM:

COLLABORATION DIAGRAM:

PACKAGE DIAGRAM:

COMPONENT DIAGRAM:

DEPLOYMENT DIAGRAM: