

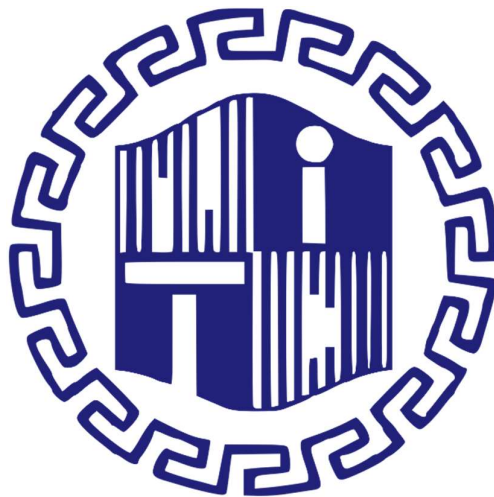
LAB REPORT
OF
SOFTWARE ENGINEERING
(CSBB 254)

A report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Submitted By
Kartik Mittal
(221210056)

Submitted To
DR. SAHIL(ASSISTANT PROFESSOR)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, DELHI

2024

Table of Contents

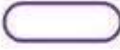

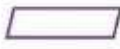




Sr. No.	Experiment Title	Page No.	Remarks
1.	Introduction to Raptor	1.1	
	Tasks:		
	T-1	1) Find the roots of quadratic equation. 2) Print all the numbers until the given number. 3) Find the largest among the given three numbers.	
2.	Raptor	2.1	
	Tasks:		
	T-2	1) Find whether a number is prime. 2) Find the factorial of a given number. 3) Generate pyramid pattern. 4) Find the minimum and maximum element in an array.	
3.	DFD	3.1	
	Tasks:		
	T-3	To draw different levels of DFD.	
4.	ER DIAGRAM(Project)	4.1	
	Tasks:		
	T-4	To draw an ER diagram of Pharmaceutical drone delivery management system.	
5.	USE CASE DIAGRAM (UCD)	5.1	
	Tasks:		
	T-5	1.Use case diagram of ATM system 2.Use case diagram of Online shopping System 3.UML diagram of credit card processing	
6.	SEQUENCE DIAGRAM	6.1	
	Tasks:		
	T-6	Sequence diagram of library management system.	

Sr. No.	Experiment Title		Page No.	Remarks
7.	Gantt Chart		7.1	
	Tasks:			
	T-7	Gantt chart of project		
8.	Class Diagram		8.1	
	Tasks:			
	T-8	Class Diagram of Project		
9.	Component Diagram		9.1	
	Tasks:			
	T-9	Component diagram of Placement Portal		
10.	SRS Documentation and Collaboration Diagram		10.1	
	Tasks:			
	T-10	SRS Documentation of Project		

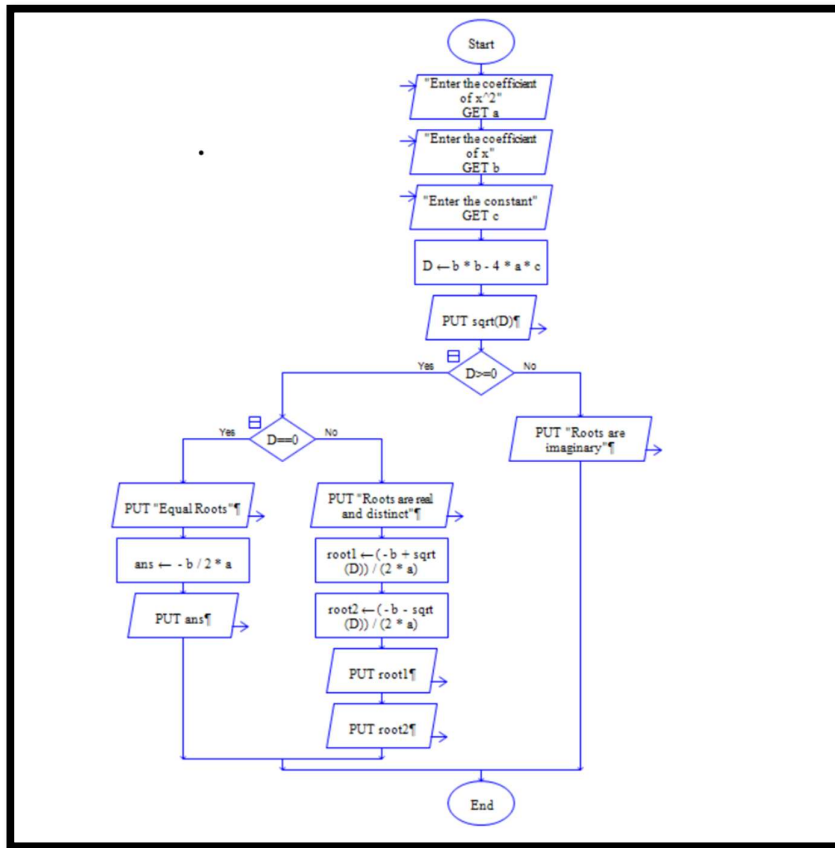
Lab Experiment No. 1: Introduction to Raptor

1. Flowcharts:

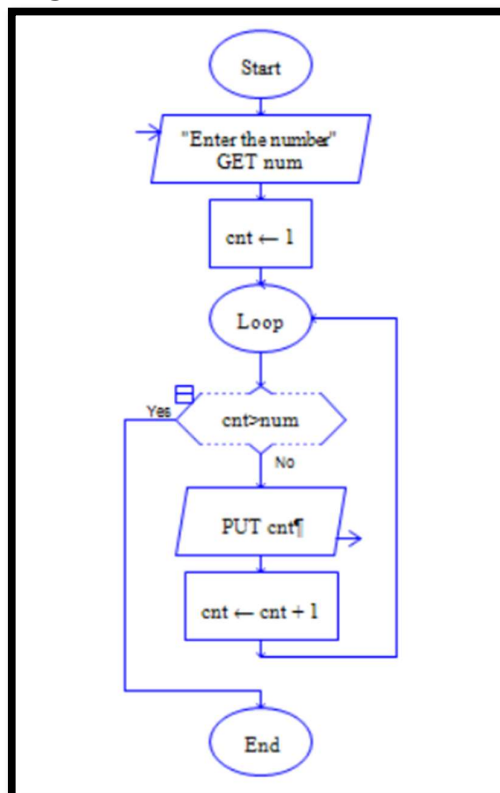
Flowchart is a diagrammatic representation of the sequence of logical steps of a program. Flowcharts use simple geometric shapes to depict processes and arrows to show relationships and process/data flow.

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

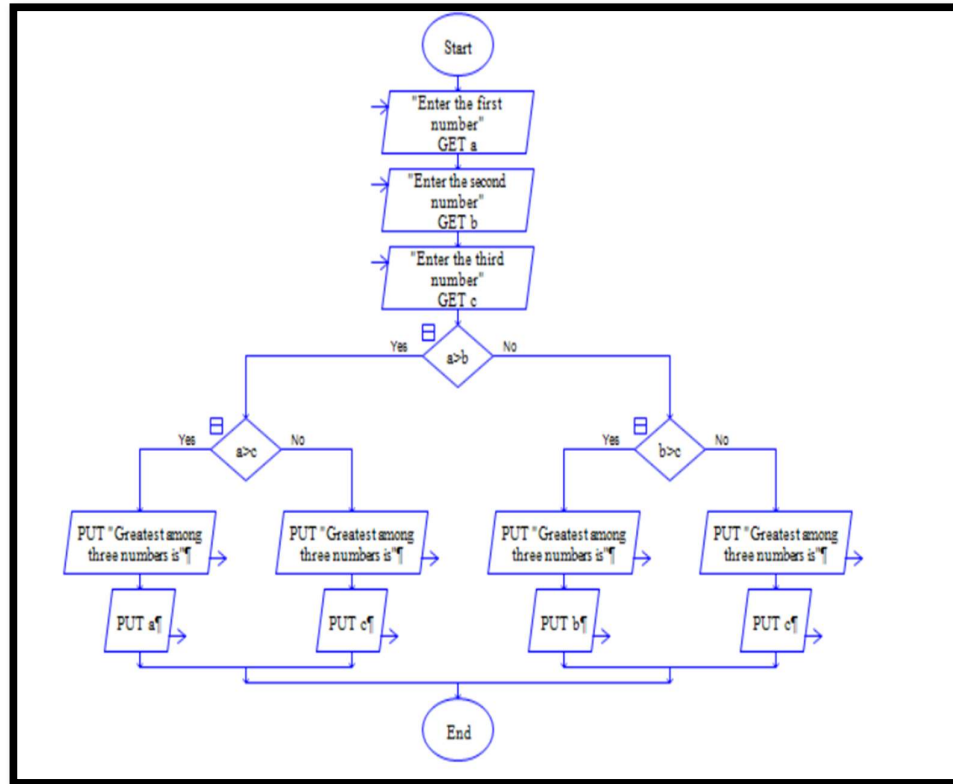
1) Find the roots of quadratic equations.



2) Print all the numbers until the given number.

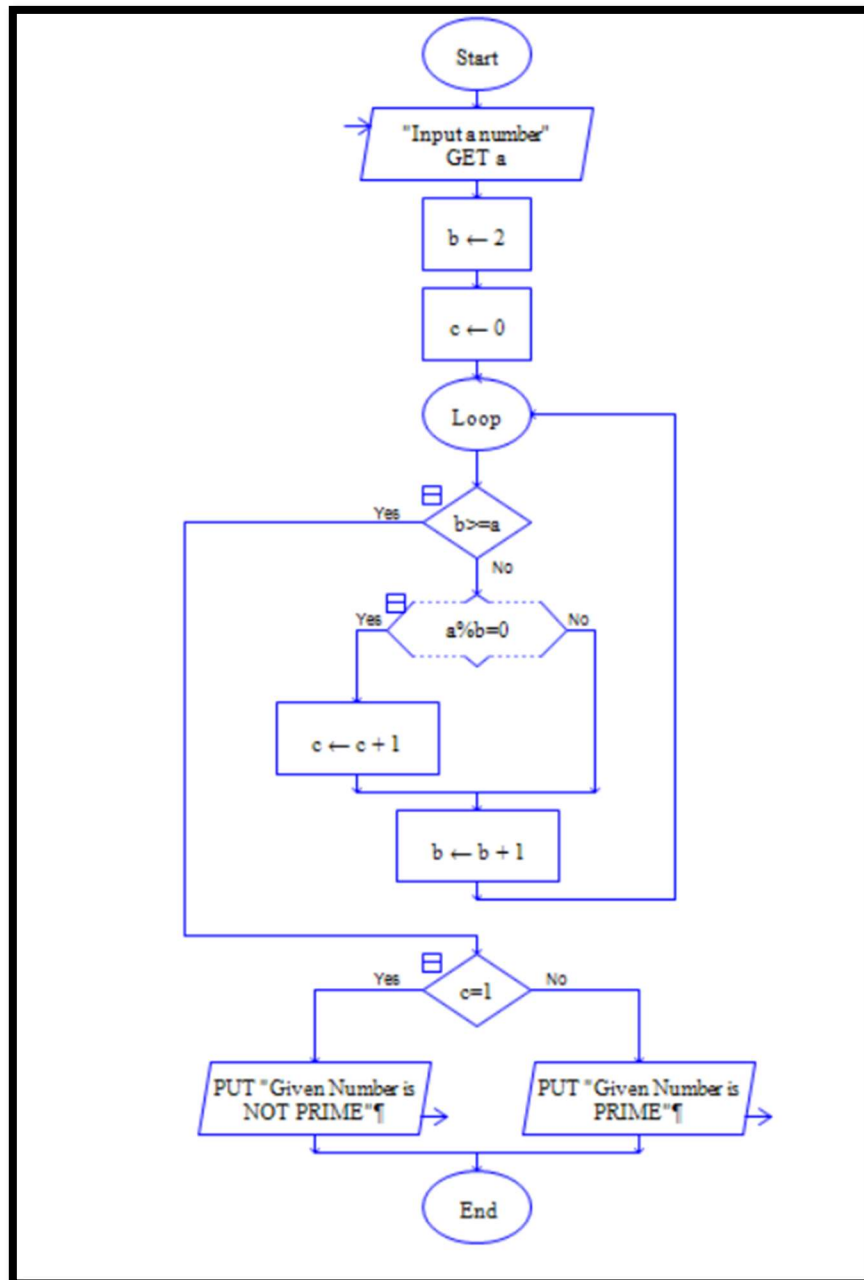


3) Find the largest among the given 3 numbers.

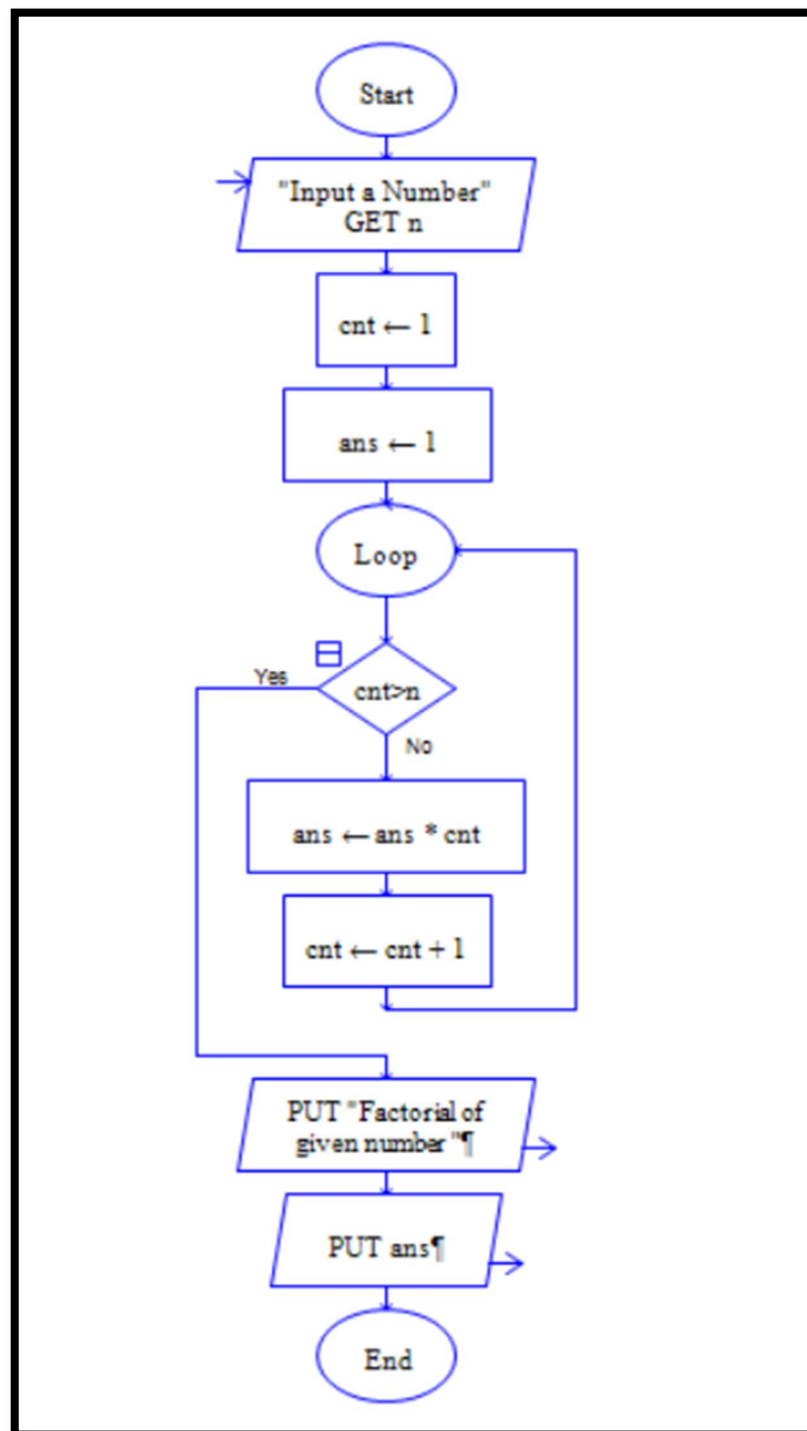


Lab Experiment No. 2 : Raptor

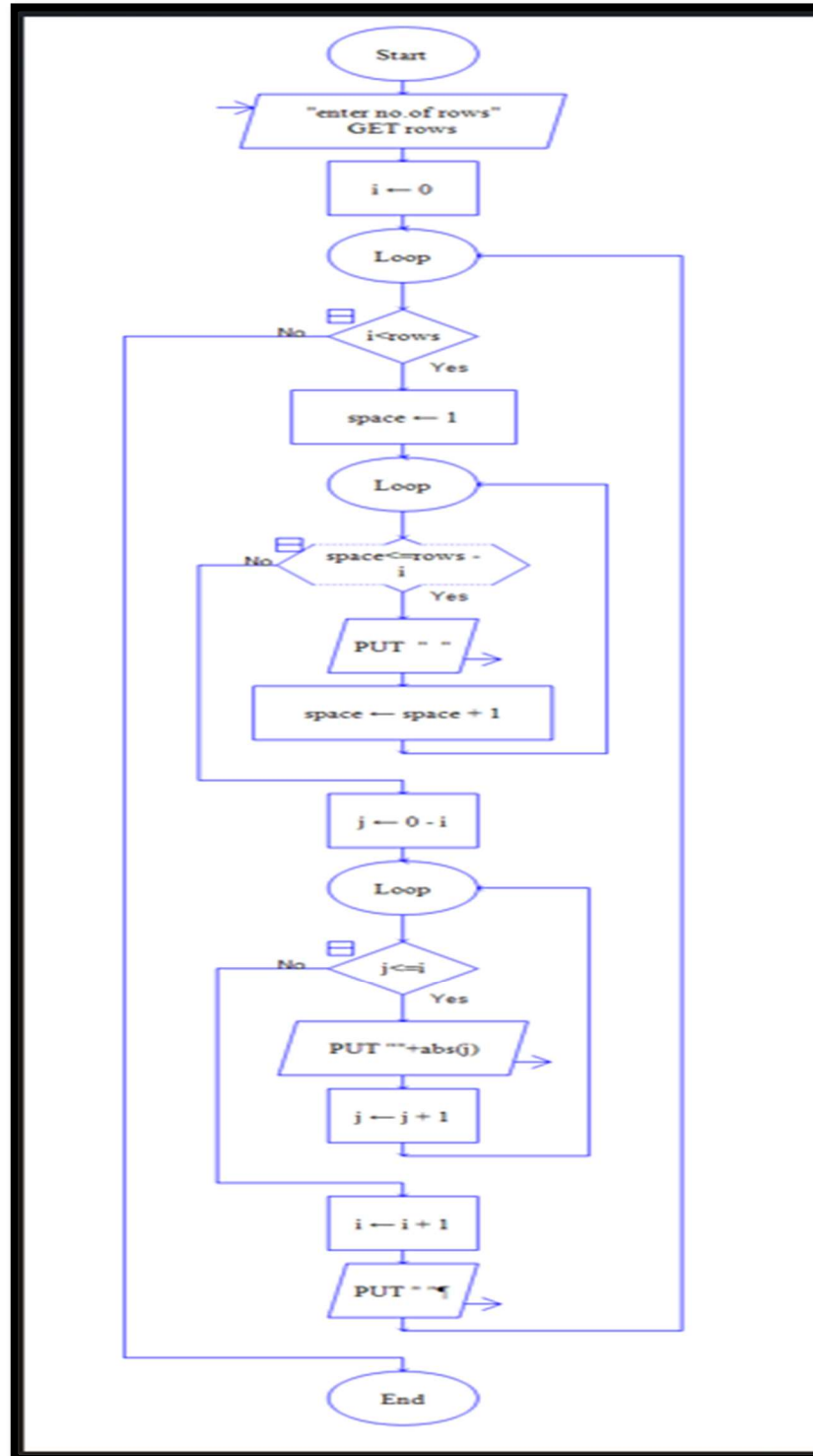
1) Find whether a number is prime.



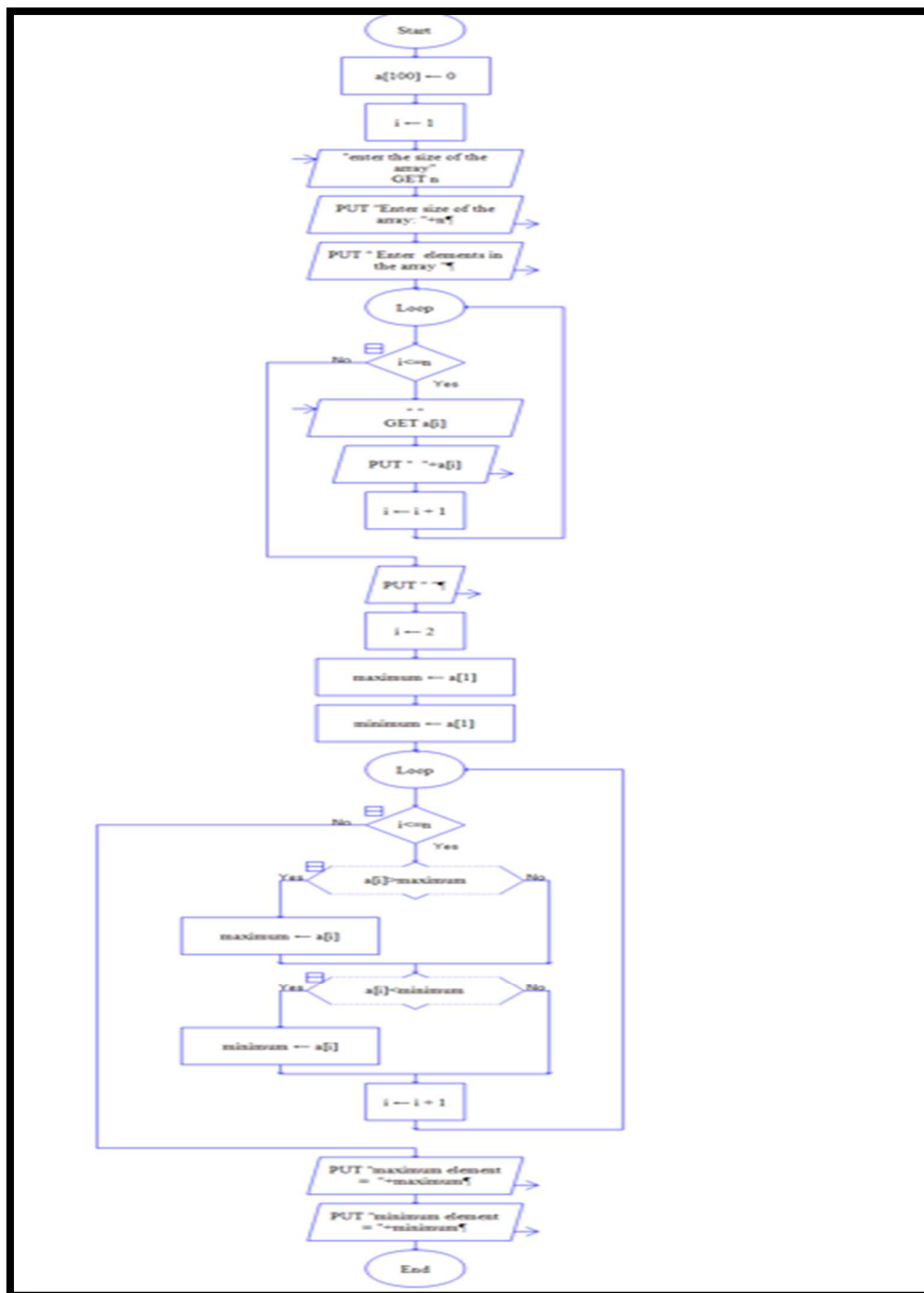
2) Find the factorial of a given number.



3) Generate pyramid pattern.



4) Find the minimum and maximum elements in an array.

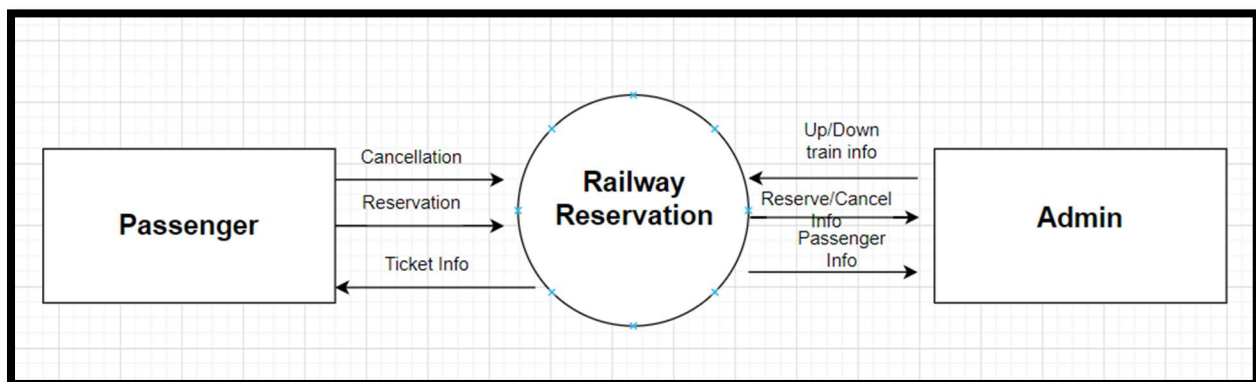


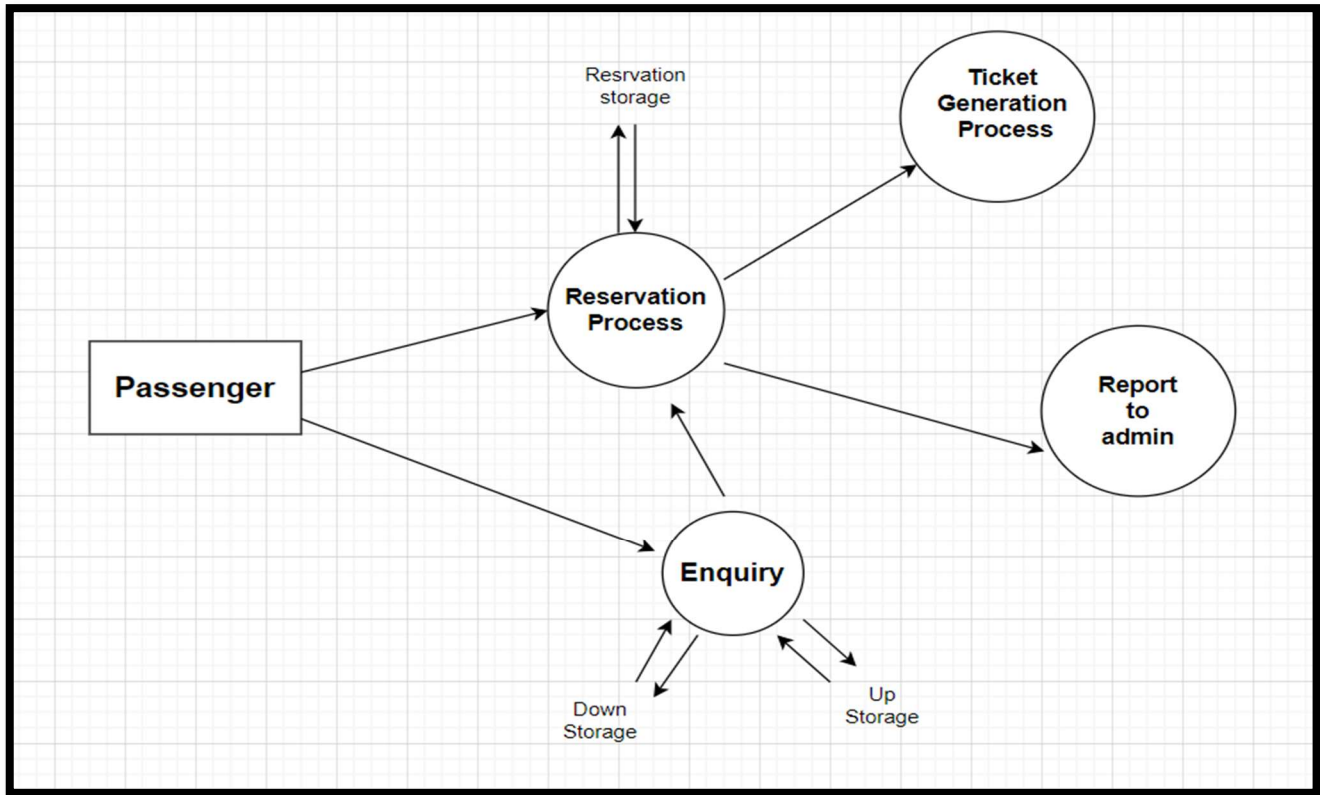
Lab Experiment No. 3 : To different levels of DFD.

In Software engineering DFD(data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher-level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

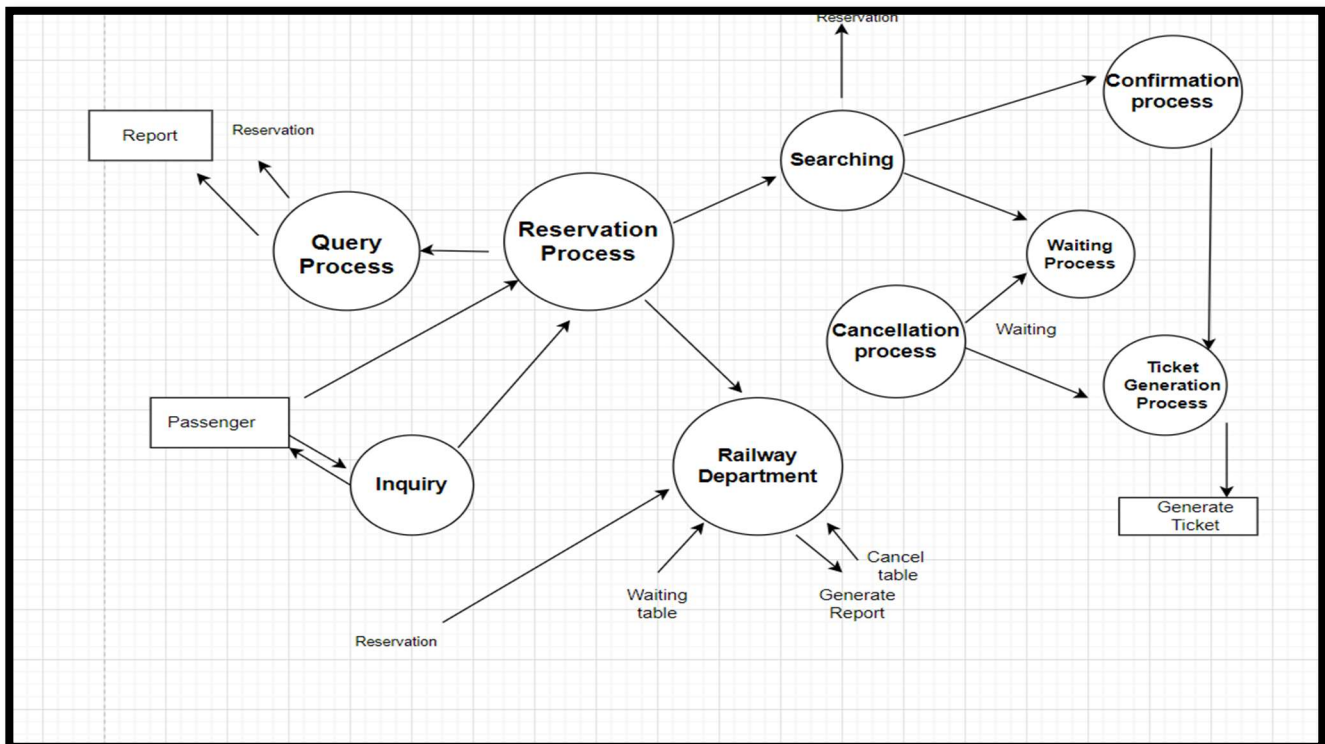
1. **Level 0 DFD:** This is the highest-level DFD, which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.
2. **Level 1 DFD:** This level provides a more detailed view of the system by breaking down the major processes identified in the level 0 DFD into sub-processes. Each sub-process is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub-process are also shown.
3. **Level 2 DFD:** This level provides an even more detailed view of the system by breaking down the sub-processes identified in the level 1 DFD into further sub-processes. Each sub-process is depicted as a separate process on the level 2 DFD. The data flows and data stores associated with each sub-process are also shown.

Level 0 DFD





Level-1 DFD

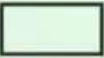




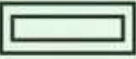


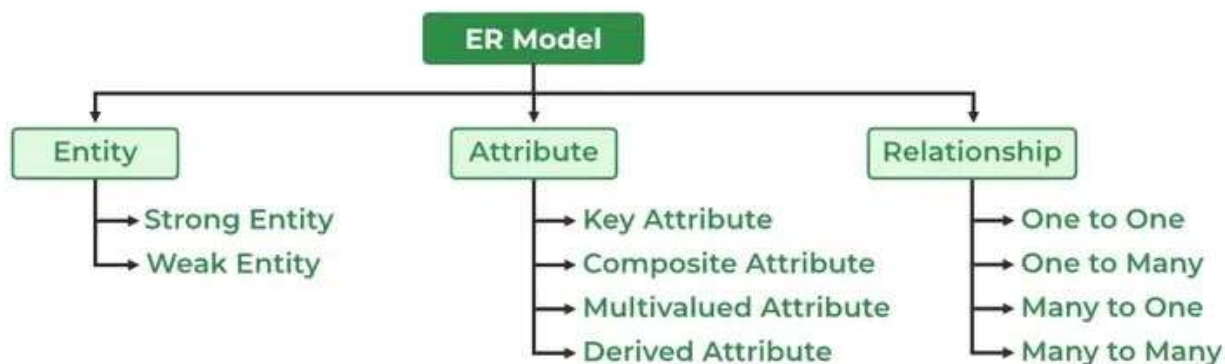
Level-2 DFD

Lab Experiment No. 4 : ER Diagram

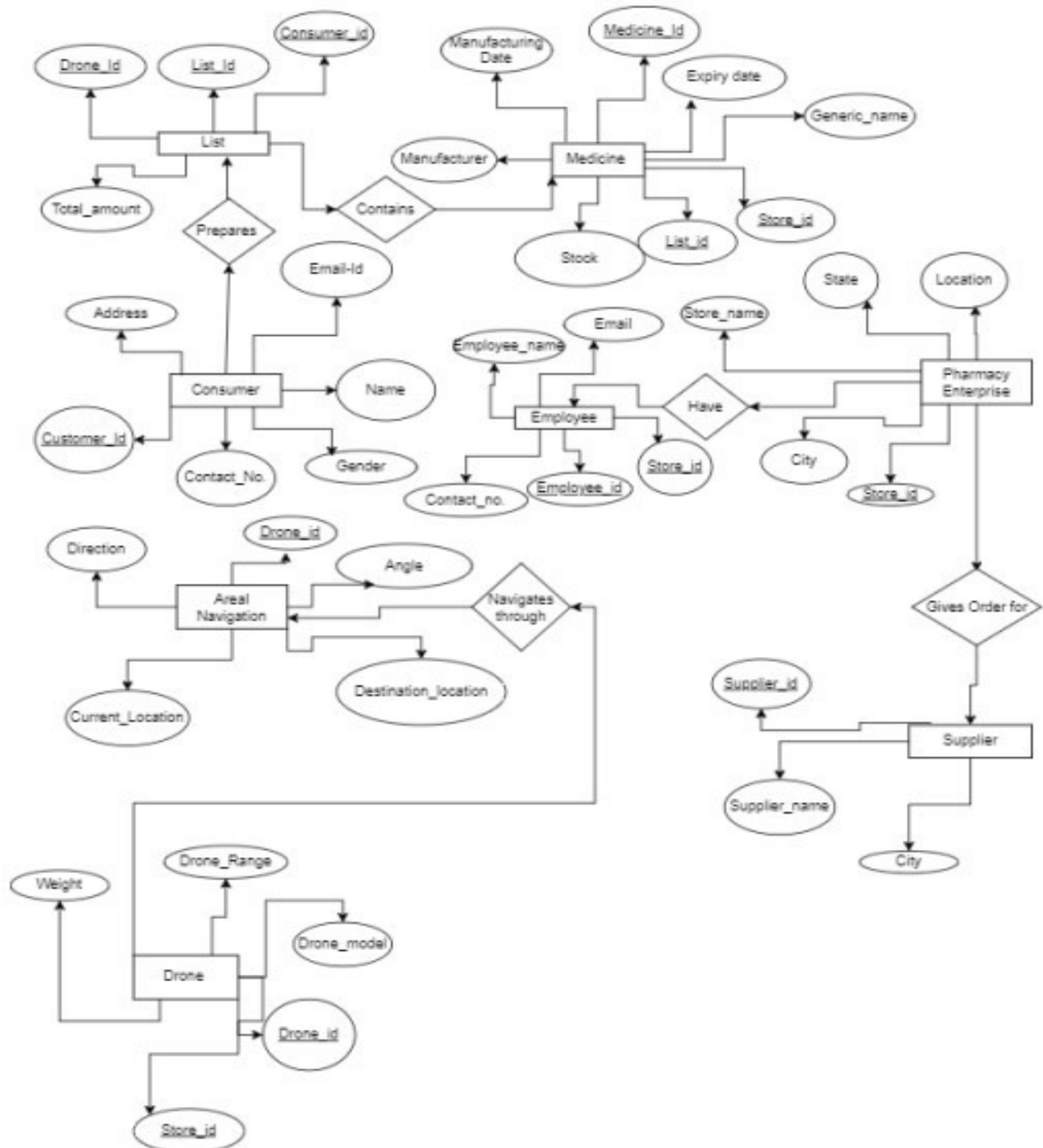
The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.

Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity



Entity Relationship Model of Pharmaceutical Drone Delivery Management System



Lab Experiment No. 5 : Use case diagram

A Use Case Diagram is a type of Unified Modelling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals. It provides a high-level view of the system's functionality by illustrating the various ways users can interact with it.

How to draw use case diagram in UML:-

Step 1: Identify Actors

Determine who or what interacts with the system. These are your actors. They can be users, other systems, or external entities.

Step 2: Identify Use Cases

Identify the main functionalities or actions the system must perform. These are your use cases. Each use case should represent a specific piece of functionality.

Step 3: Connect Actors and Use Cases

Draw lines (associations) between actors and the use cases they are involved in. This represents the interactions between actors and the system.

Step 4: Add System Boundary

Draw a box around the actors and use cases to represent the system boundary. This defines the scope of your system.

Step 5: Define Relationships

If certain use cases are related or if one use case is an extension of another, you can indicate these relationships with appropriate notations.

Step 6: Review and Refine

Step back and review your diagram. Ensure that it accurately represents the interactions and relationships in your system. Refine as needed.

Step 7: Validate

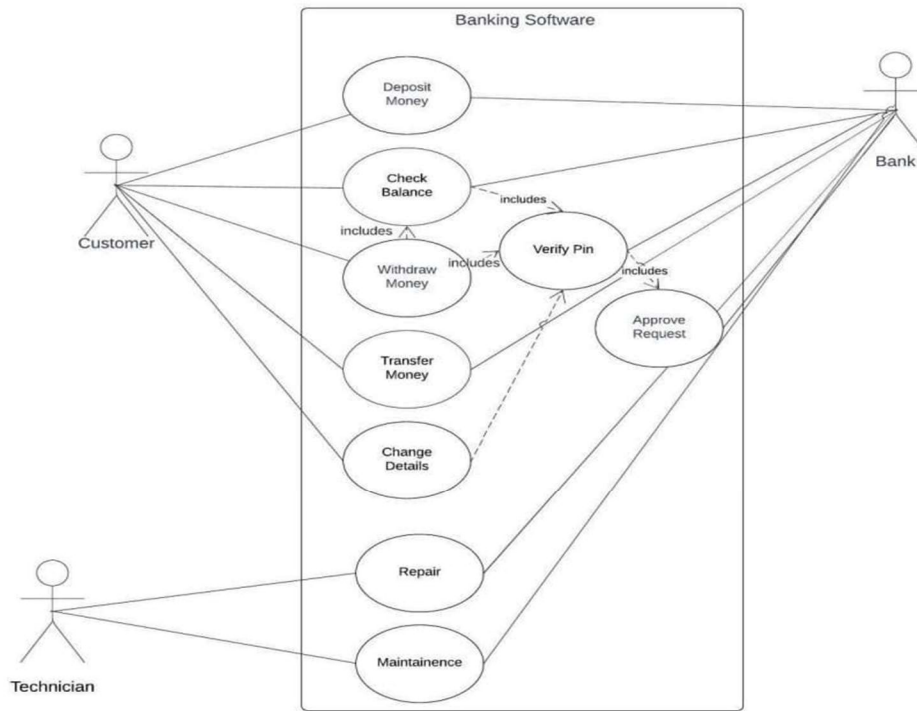
Share your use case diagram with stakeholders and gather feedback. Ensure that it aligns with their understanding of the system's functionality.

Software used for Use Case Diagram:-

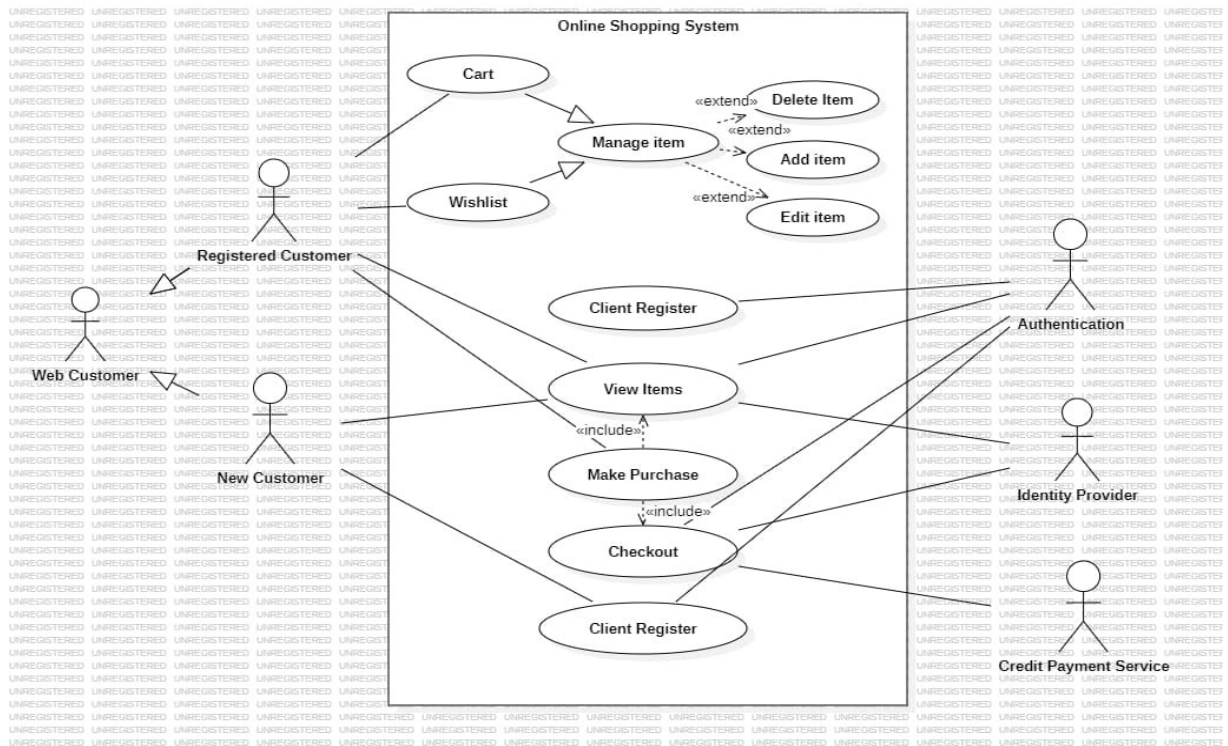
draw.io:-

- Free, open-source diagramming tool.
- Works offline and can be integrated with Google Drive, Dropbox, and others.
- Offers a wide range of diagram types, including Use Case Diagrams.

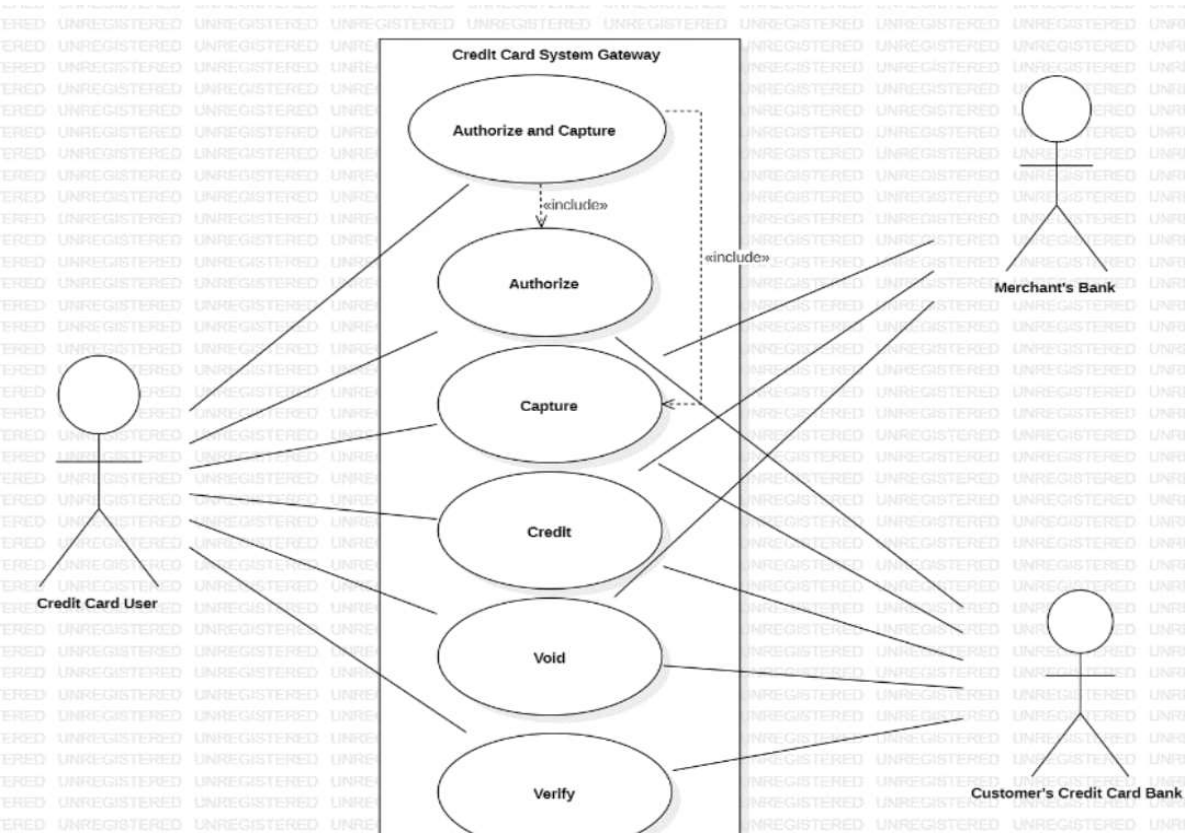
1. Use Case Diagram of ATM System



2. Use Case Diagram of online shopping system



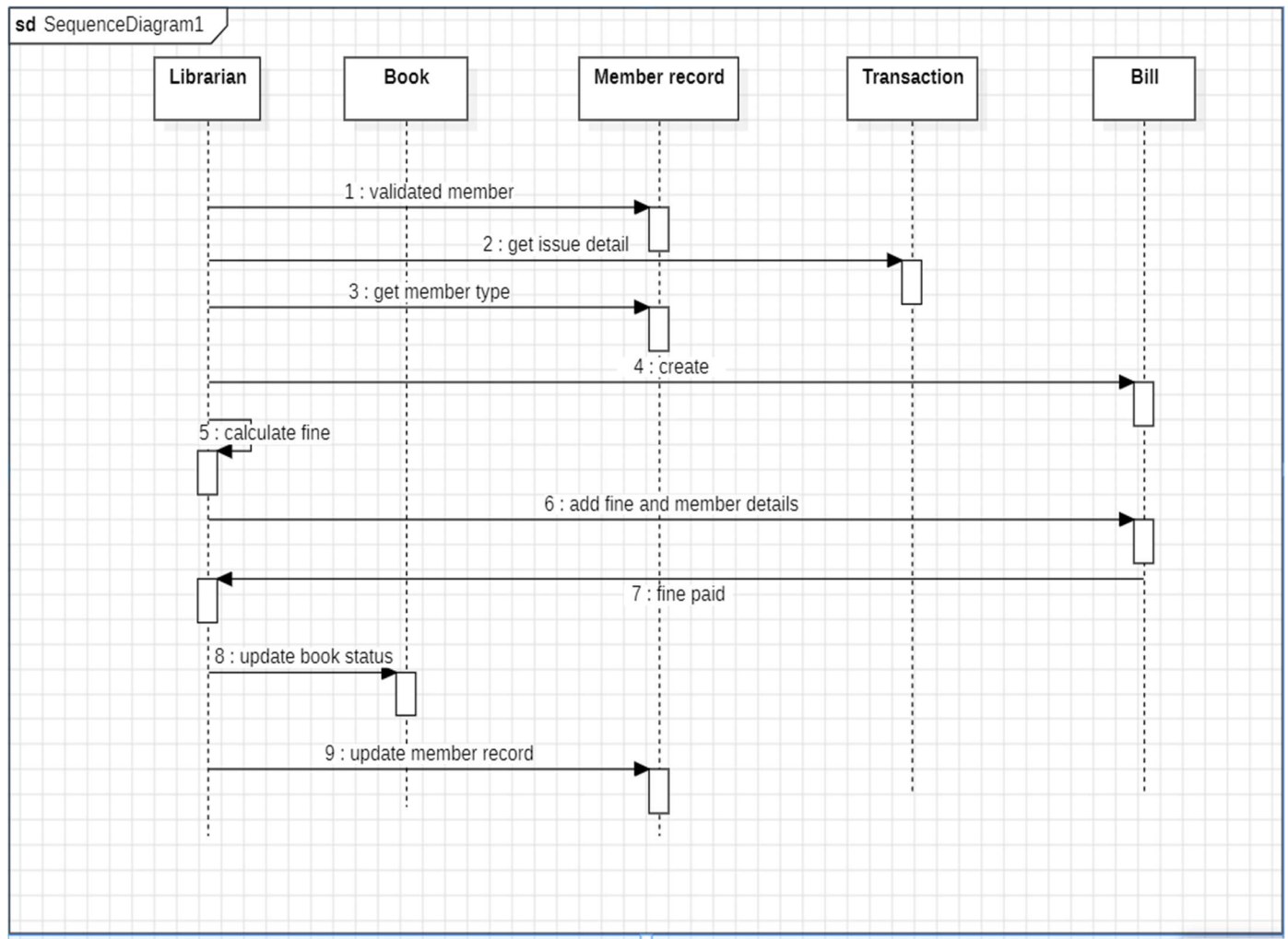
3.UML Diagram of Credit Card Processing



Lab experiment No. 6 : Sequence diagram Library Management system

Sequence or event diagrams help create an overview of how a system works. Sequence diagrams show how all the different parts interact with each other over time, carry out the required actions, and complete processes. A sequence diagram descends from top to bottom showing a sequence of interactions and sequence diagram notations.

The sequence diagram maps out a library management system. First, the member requests a book from the librarian. The librarian checks the availability of the book and validates the request. Before issuing the book, the librarian checks the number of books the member has checked out. Then the member gets the requested book. The librarian creates the transaction and documents the member and book details. After the librarian creates the transaction, the librarian updates the status of the issued book and member record.



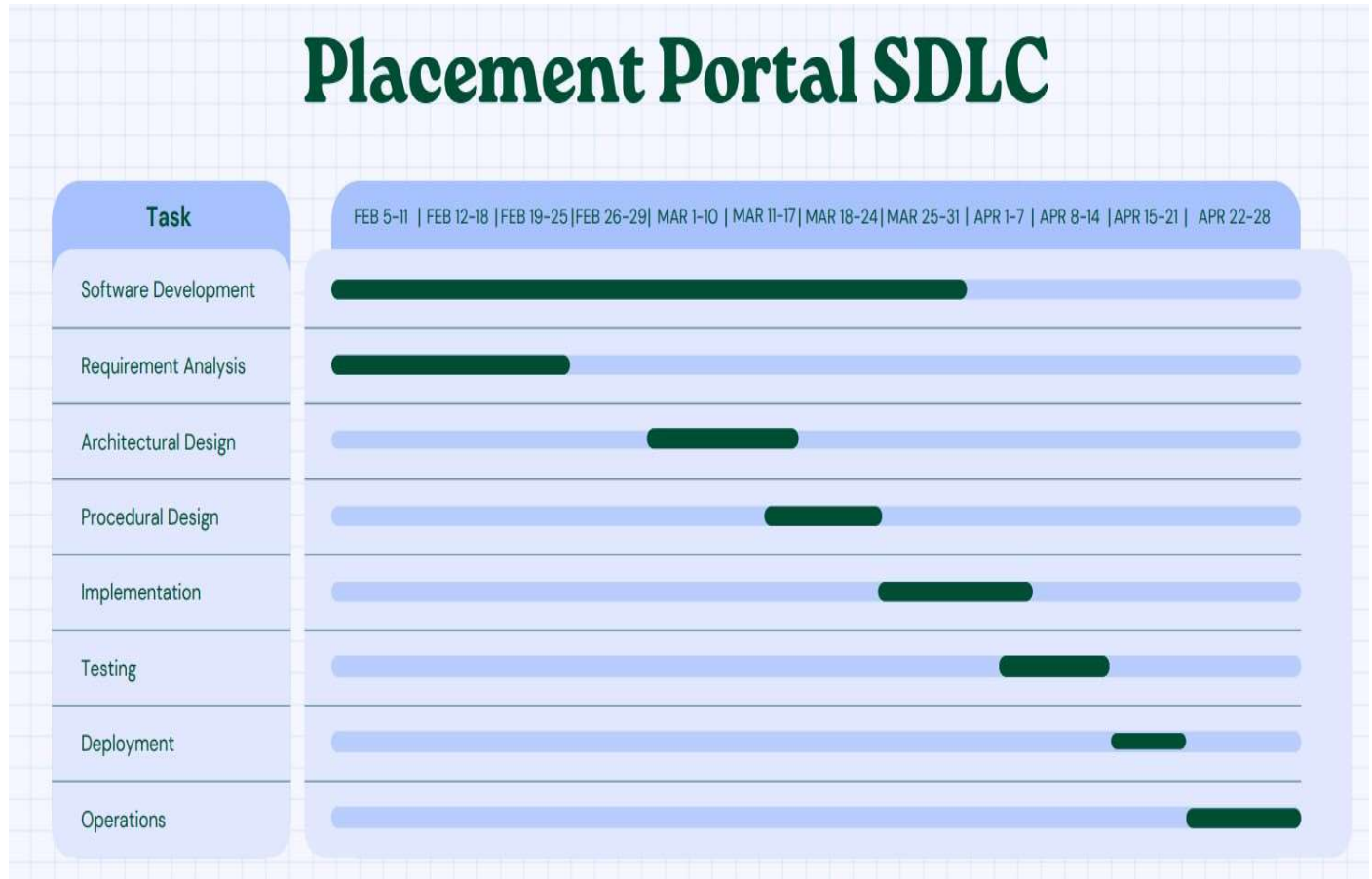
Lab Experiment No. 7 : Gantt Chart (Project)

Generalized Activity Normalization Time Table (GANTT) chart is type of chart in which series of horizontal lines are present that show the amount of work done or production completed in given period of time in relation to amount planned for those projects. It is horizontal bar chart developed by Henry L. Gantt (American engineer and social scientist) in 1917 as production control tool. It is simply used for graphical representation of schedule that helps to plan in an efficient way, coordinate, and track some particular tasks in project.

The purpose of Gantt chart is to emphasize scope of individual tasks. Hence set of tasks is given as input to Gantt chart. Gantt chart is also known as timeline chart. It can be developed for entire project or it can be developed for individual functions. In most of projects, after generation of timeline chart, project tables are prepared. In project tables, all tasks are listed in proper manner along with start date and end date and information related to it.

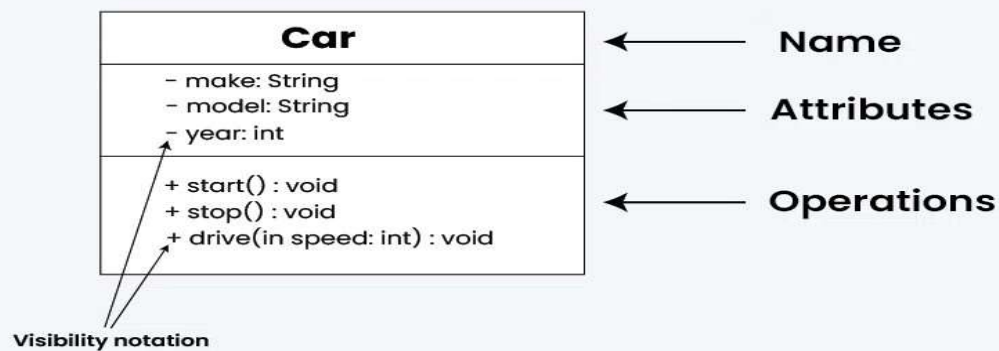
Gantt chart represents following things :

- All the tasks are listed at leftmost column.
- The horizontal bars indicate or represent required time by corresponding particular task.
- When occurring of multiple horizontal bars takes place at same time on calendar, then that means concurrency can be applied for performing particular tasks.
- The diamonds indicate milestones.



Lab Experiment No. 8 : Class Diagram (Project)

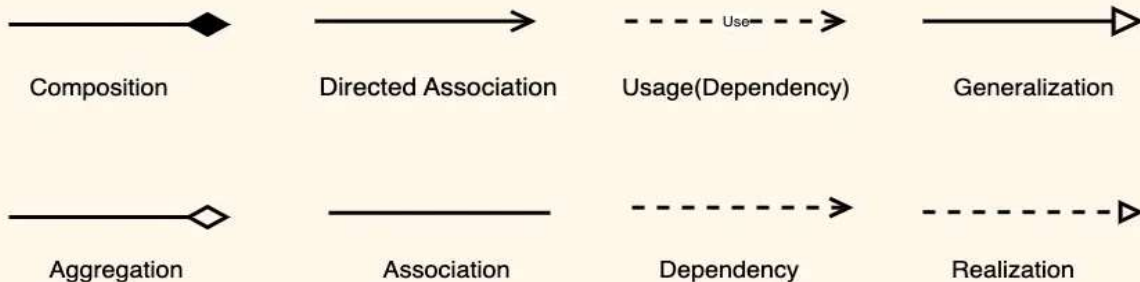
Class diagrams are a type of UML (Unified Modeling Language) diagram used in software engineering to visually represent the structure and relationships of classes in a system. UML is a standardized modeling language that helps in designing and documenting software systems. They are an integral part of the software development process, helping in both the design and documentation phases.

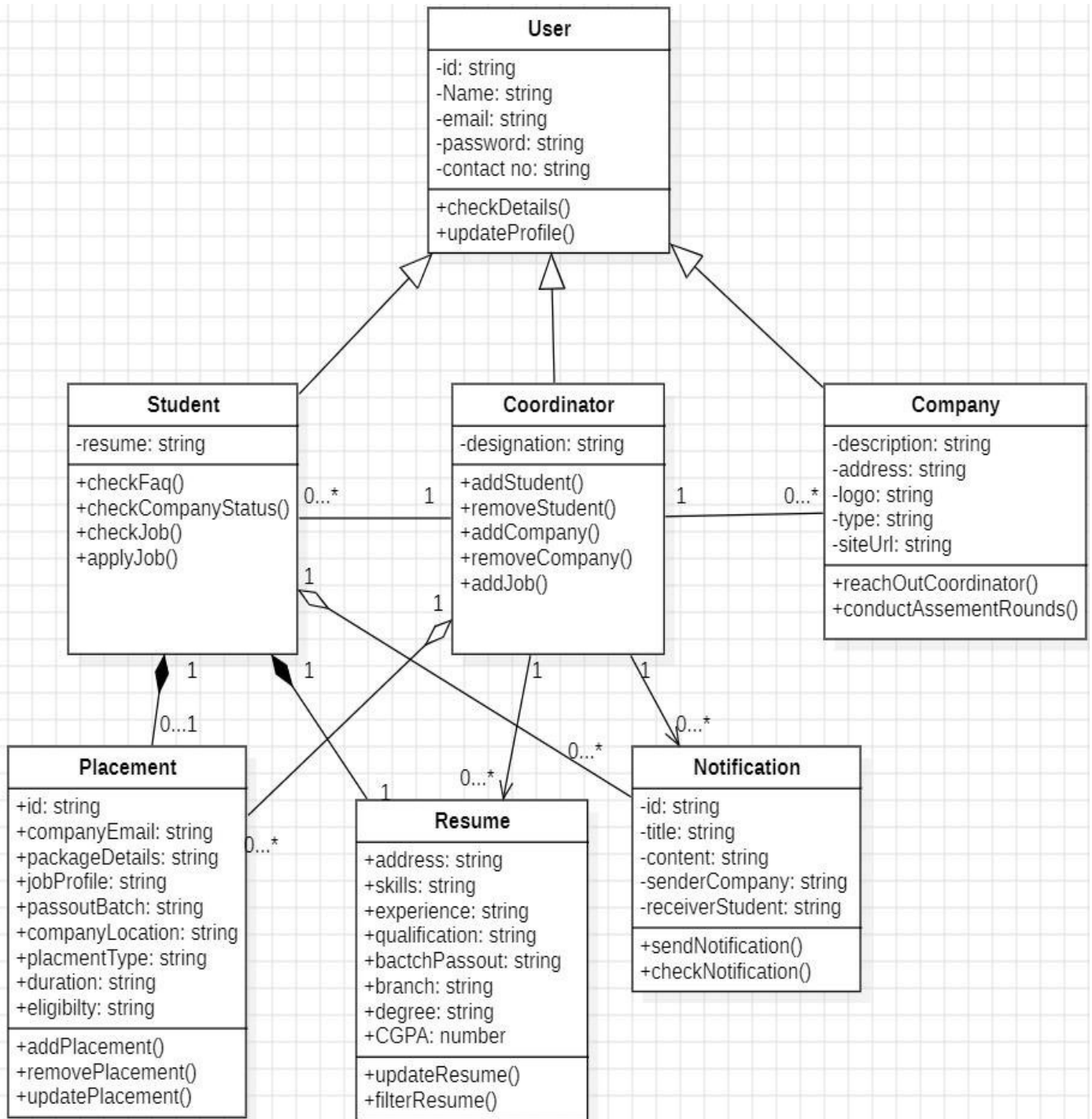


Class Notation



Class Diagram Relationships

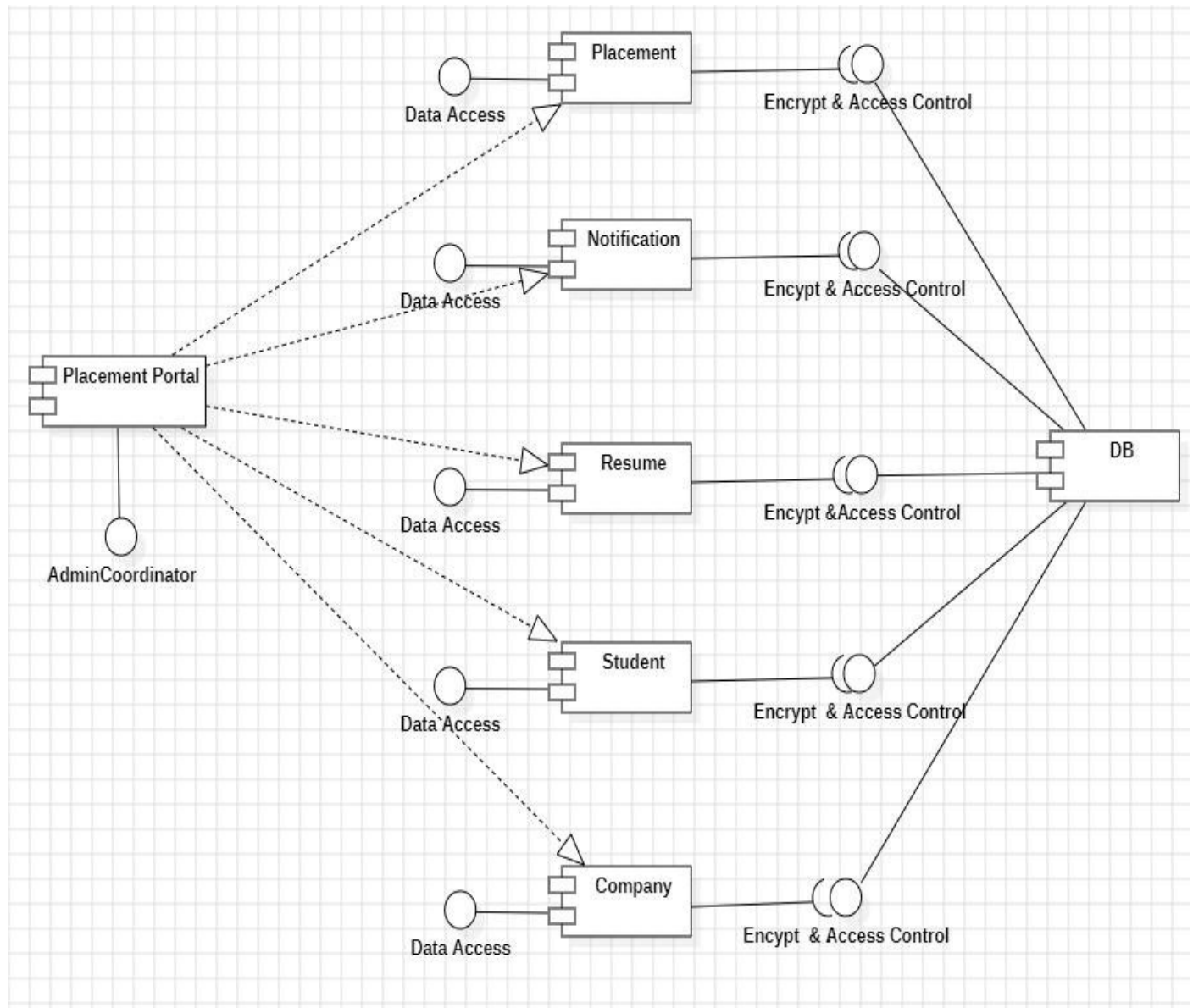




Lab Experiment No. 9 Component Diagram

Component Diagram:

In Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



Lab Experiment -10 SRS Documentation of Project

Software Requirements and Specification Document:

Software Requirement Specification (SRS) document is a complete set of specifications and descriptions of software requirements that must be fulfilled to successfully develop the software system. These requirements can be functional and non-functional depending upon the type of requirement. The interaction between different customers and contractors is done because it is necessary to fully understand customers' needs.

Properties of a good SRS document:

- **Concise:** The SRS report should be concise and at the same time, unambiguous, consistent, and complete. Verbose and irrelevant descriptions decrease readability and also increase error possibilities.
- **Structured:** It should be well-structured. A well-structured document is simple to understand and modify. In practice, the SRS document undergoes several revisions to meet user requirements. Often, user requirements evolve over time. Therefore, to make the modifications to the SRS document easy, it is vital to make the report well-structured.
- **Black-box view:** It should only define what the system should do and refrain from stating how to do these. This means that the SRS document should define the system's external behaviour rather than discuss the implementation issues. The SRS report should view the system to be developed as a black box and should define the externally visible behaviour of the system. For this reason, the SRS report is also known as the black-box specification of a system.
- **Conceptual integrity:** It should show conceptual integrity so that the reader can merely understand it. Response to undesired events: It should characterise acceptable responses to unwanted events. These are called system responses to exceptional conditions.

SRS Document of Project

1. Introduction

1.1 Purpose

Placement Portal primary purpose is to provide clarity and direction for the development and implementation of the portal, ensuring that all stakeholders understand the objectives and expectations associated with the project.

1.2 Need

In today's job market, inefficiencies plague both job seekers and employers. Job seekers struggle to find suitable opportunities amidst a sea of postings, while employers face challenges in identifying the right candidates efficiently. Lengthy recruitment processes compound the issue, leading to delays and frustration on both sides. Limited visibility exacerbates the problem, hindering job seekers from discovering desirable positions and employers from attracting top talent. Moreover, safety concerns in online interactions and environmental sustainability add further complexity.

1.3 Project Scope

Our project has a big scope to do. Students can access previous information about placement. We can store information about all students. Various companies can access their information. Notifications are sent to students about the companies.

1. Easy to collect and manage student data.

2. To increase the accuracy and efficiency of placement procedure.

3. Reduce the paperwork.

4. Analysis of overall placement activities

FUNCTIONAL REQUIREMENTS

1: User Registration

User Requirement:

Users should be able to easily create an account by providing basic information such as username, password, email, and phone number.

System Requirement:

1. The registration form shall include fields for username, password and email with appropriate validation for data accuracy.
2. Upon submission, the system shall verify the uniqueness of the username and email address to prevent duplicate accounts. The system shall encrypt sensitive user data, such as passwords, during transmission and storage using industry-standard cryptographic algorithms.

2: User Login

User Requirement:

Registered users should be able to log in to their accounts using their credentials quickly and securely.

System Requirement:

1. The system shall implement secure authentication mechanisms, using industry-standard cryptographic algorithms, to protect user passwords from unauthorised access.
 2. Upon successful login, the system shall generate a session token and maintain user authentication and authorization state across web sessions on various devices securely.
- ### **3: Profile Management**

User Requirement:

Users should be able to update their profile information, including personal details, contact information, qualification details and profile picture, as needed.

System Requirement:

1. The system shall feature an intuitive and user-friendly interface for effortless profile management, accessible from the user dashboard.
 2. The profile management interface shall display the user's current information, including personal details, contact information, and profile picture, in editable fields.
- ### **4: Company Registration**

User Requirement:

Companies should be able to register with the portal by providing relevant information such as company name, industry, contact details, and job offerings.

System Requirement:

1. The system shall provide a registration form for new company registration, accessible from the homepage or company registration page.
2. The registration form shall include fields for company name, industry, contact person, email, phone number, and job offerings, with appropriate validation for data accuracy.
3. The system shall encrypt sensitive company data, such as contact details and job offerings, during transmission and storage using industry-standard cryptographic algorithms.

5: Job Posting

User Requirement:

Companies should be able to post job openings and internship opportunities with detailed descriptions and requirements.

System Requirement:

1. The system shall provide a job posting form for companies to submit job openings and internship opportunities, accessible from the company dashboard.
2. Upon submission, the system shall review and approve job postings before making them visible to students.
3. Approved job postings shall be displayed in the job board section of the portal, categorized by industry, location, and job type for easy access and navigation by students.

3.3 Student Management

6: Job Search and Application

User Requirement:

Students should be able to search and browse job openings posted by companies and apply for positions of interest.

System Requirement:

1. The system shall provide a job search functionality for students to browse and filter job openings based on criteria such as industry, location, and job type.
2. Students shall have the ability to apply for job openings directly through the portal by submitting their resumes and cover letters.
3. Upon submission, the system shall notify the respective company and update the application status in the student's dashboard.
4. The system shall maintain a record of all job applications submitted by students, along with relevant details such as application date, status, and communication history with the company.

3.4 Notification System

7: Notification Delivery

User Requirement:

Users should receive timely notifications about important events, such as new job postings, application status updates, and upcoming placement events.

System Requirement:

1. The system shall implement a notification system to deliver alerts and updates to users via email, in-app notifications, and SMS.
2. Notifications shall be triggered by specific events, such as new job postings, application status changes, and event reminders.

3. The system shall maintain a log of all notifications sent to users, including details such as notification type, timestamp, and recipient.

3.5 Placement Officer (TPO) Management

8: Company Approval and Management

User Requirement:

Placement officers should have the ability to review and approve company registrations, manage company profiles, and update company information as needed.

System Requirement:

1. The system shall provide placement officers with a dashboard to view pending company registrations and approve them based on predefined criteria.
2. The system shall maintain a log of all company interactions and updates made by placement officers for audit and tracking purposes.

9: Student Verification and Management

User Requirement:

Placement officers should be able to verify student registrations, manage student profiles, and track student engagement in placement activities.

System Requirement:

1. The system shall enable placement officers to verify student registrations by reviewing and approving student profiles based on eligibility criteria.
2. Placement officers shall have access to student profiles to view and update information such as academic records, skills, and placement preferences.
3. The system shall track student engagement in placement activities, including job applications, interview schedules, and placement event attendance, for monitoring and reporting purposes.

4. Non-Functional Requirements

4.1 Performance

1: Scalability

- The system should be able to accommodate a growing number of users and increased activity without significant slowdowns.
- Application architecture should support the addition of more resources to effectively distribute workload.

2: Response Time

- The system should respond to user interactions within acceptable time frames, ensuring a smooth user experience.
- Web pages should load quickly, and actions such as job applications and profile updates should be processed promptly to minimise user wait times.

4.2 Security

3: Data Encryption

- All user data, including personal details and resumes, should be encrypted using robust encryption methods to prevent unauthorised access.
- Secure communication channels, such as HTTPS, should be implemented to protect data transmission over the network.

4: Access Control

- Access to student profiles, company details, and administrative functions should be restricted based on user roles and permissions.
- Authentication mechanisms, including password protection and multi-factor authentication, should be employed to verify user identities securely.

4.3 Reliability

5: High Availability

- The system should maintain high availability, with uptime of at least 99.9% excluding scheduled maintenance windows.

- Redundant servers and failover mechanisms should be in place to ensure continuous service in the event of hardware or software failures.

6: Data Backup

- Regular backups of all system data should be performed to prevent data loss in the event of system failures or disasters.
- Backup procedures should include off-site storage and regular testing to verify data integrity and recovery capabilities.

4.4 Maintainability

7: Modular Design

- The system should be designed with a modular architecture to facilitate easy maintenance and future enhancements.
- Codebase should be well-organized and documented, allowing for efficient troubleshooting and development.

8: Version Control

- Version control systems such as Git should be used to track changes to the codebase and facilitate collaboration among development teams.
- Code changes should be managed through a structured workflow, including code reviews and testing procedures.

4.5 Usability

9: Intuitive Interface

- The user interface should be intuitive and easy to navigate, with clear labeling and consistent design elements.
- Usability testing should be conducted to identify and address any usability issues, ensuring a positive user experience for all stakeholders.

5. Goals and Guidelines

Goals:

- **Efficient Job Matching**

Develop algorithms that efficiently match job seekers with suitable job openings based on their skills, experience, and preferences to optimize job placement success.

- **User-Friendly Interface**

Provide a user-friendly interface that allows job seekers and employers to easily navigate the portal, create profiles, search for jobs or candidates, and manage their accounts effectively.

- **Streamlined Recruitment Process**

Facilitate a seamless recruitment process for both job seekers and employers, from job application submission to interview scheduling and offer negotiation, to expedite hiring decisions.

- **Transparent and Trustworthy Platform**

Maintain transparency and trustworthiness by ensuring accurate and up-to-date information, protecting user data privacy, and fostering a safe and respectful online community for job seekers and employers.

- **Scalability and Expansion**

Build a scalable platform that can accommodate a growing user base and adapt to evolving market trends and technological advancements, allowing for future expansion into new regions or sectors.

Guidelines:

- **Feedback Mechanism**

Implement a feedback mechanism for users to provide input on their experience with the portal, including suggestions for improvements and features they would like to see.

- **Employer Engagement**

Provide employers with tools and resources to actively engage with job seekers, such as hosting virtual career fairs, participating in employer spotlights, or offering internship programs.

- **Diversity and Inclusion Initiatives**

Ensure that the portal's content and messaging reflect a commitment to diversity, equity, and inclusion principles.

- **Partnerships and Collaborations**

Collaborate with local businesses and community organizations to promote job opportunities and support workforce development initiatives.

- **Legal Compliance**

Provide clear disclosures and disclaimers regarding the portal's terms of use, privacy practices, and liability limitations to protect both users and the portal.