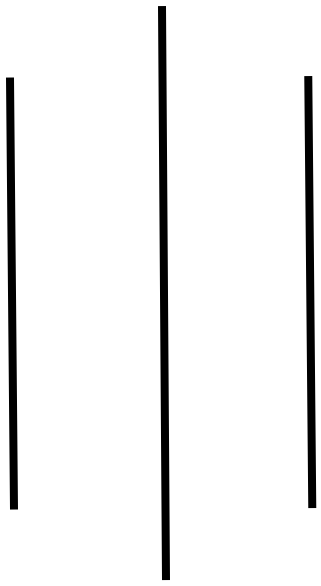


LAB REPORT ON

SYSTEM ANALYSIS AND DESIGN

SUBMITTED BY : SAMYAK MANANDHAR

SUBMITTED TO : HIRANYA BASTAKOTI

FACULTY : BSC.CSIT 5TH SEMESTER

TABLE OF CONTENT

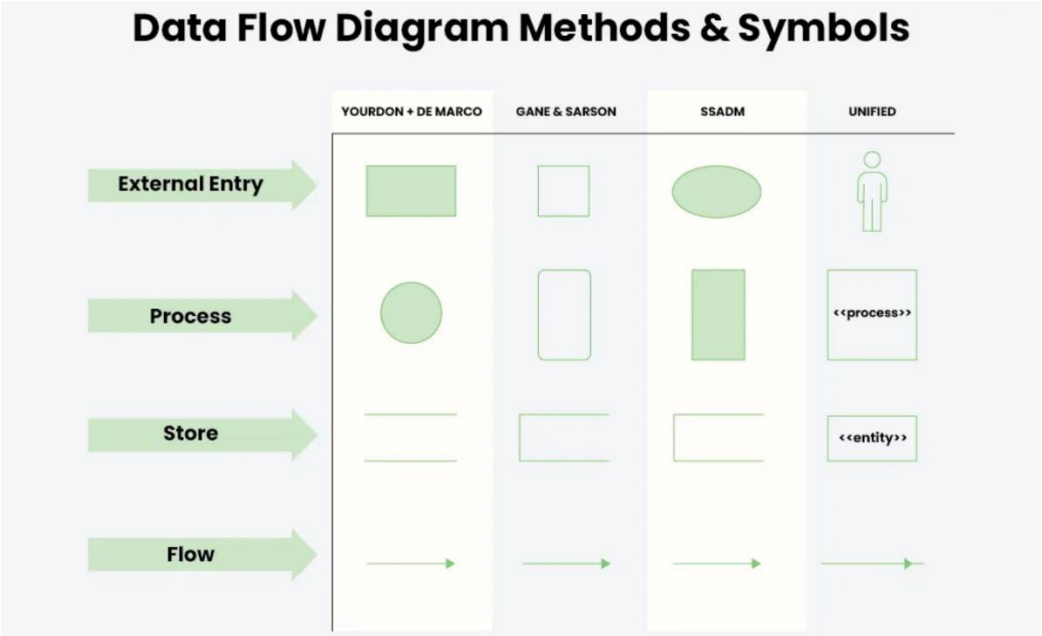
S.N	TOPIC
1.	Data Flow Diagram (DFD) for an Online Shopping System.
2.	Data Flow diagram for Food Ordering System
3.	E-R Diagram for a Library Management System.
4.	Use Case Diagram for an ATM System.
5.	Class Diagram for a Hotel Management System.
6.	Sequence Diagram for a Hospital Management System.
7.	Use Case Diagram for a University Registration System.
8.	Activity Diagram for an Airline Reservation System.
9.	Class Diagram for an Airline Reservation System.
10.	Use Case Diagram for a Food Ordering System.
11.	Component Diagram for an Online Banking System.
12.	Deployment Diagram for an E-Commerce Website.
13.	State Diagram for an Online Examination System.
14.	Collaboration Diagram for a Customer Support System.
15.	Sequence Diagram for an Online Ticket Booking System.

LAB 1: Data Flow Diagram (DFD) for an Online Shopping System.

INTRODUCTION:

A Data Flow Diagram (DFD) is a graphical representation of the flow of data through a system, the inputs to the system, processing, storage, and outputs. The four main components of a DFD are external entities, which are users or outside systems that communicate with the system; processes, which transform input data into output; data stores, which hold information to be used later; and data flows, which indicate the movement of data between parts. DFDs are segmented into levels, starting with Level 0 (Context Diagram), a high-level representation of the overall system, to Level 1, which breaks down the system into major processes, and more detailed levels (Level 2+), which decompose processes into smaller actions. Widely used in software development, system analysis, and database design, DFDs help in visualizing the workflows, system efficiency, and identifying potential data processing issues.

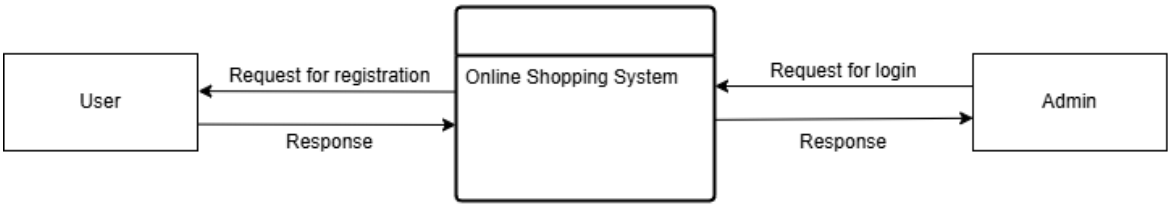
NOTATIONS USED:



Online Shopping System:

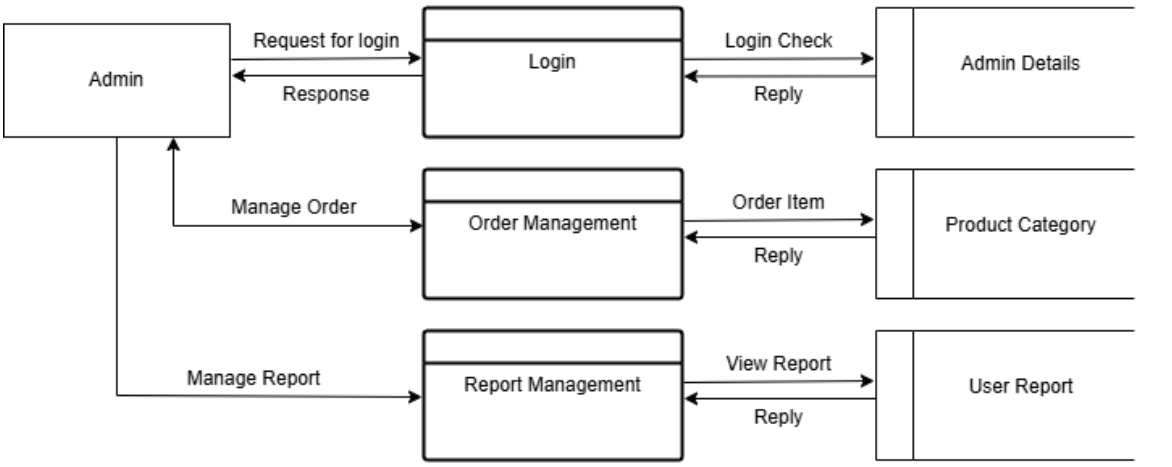
An Online Shopping System is a platform that allows users to browse, select, and purchase products over the internet. It typically consists of multiple interconnected components, including a user interface, a product catalog, a shopping cart, a payment gateway, and an order management system. In a Data Flow Diagram (DFD), the system's processes and data flow are visually represented to show how information moves through the system. For example, users interact with the system by searching for products, adding them to their cart, and proceeding to checkout. The system processes these actions by verifying product availability, calculating prices, and processing payments. The DFD helps in understanding how data is entered, processed, stored, and retrieved within the system, highlighting key entities like users, administrators, products, and payment systems. This diagram is essential for designing the system's architecture and ensuring efficient interaction between the various components involved in the online shopping experience.

Level 0: DFD for Online Shopping System



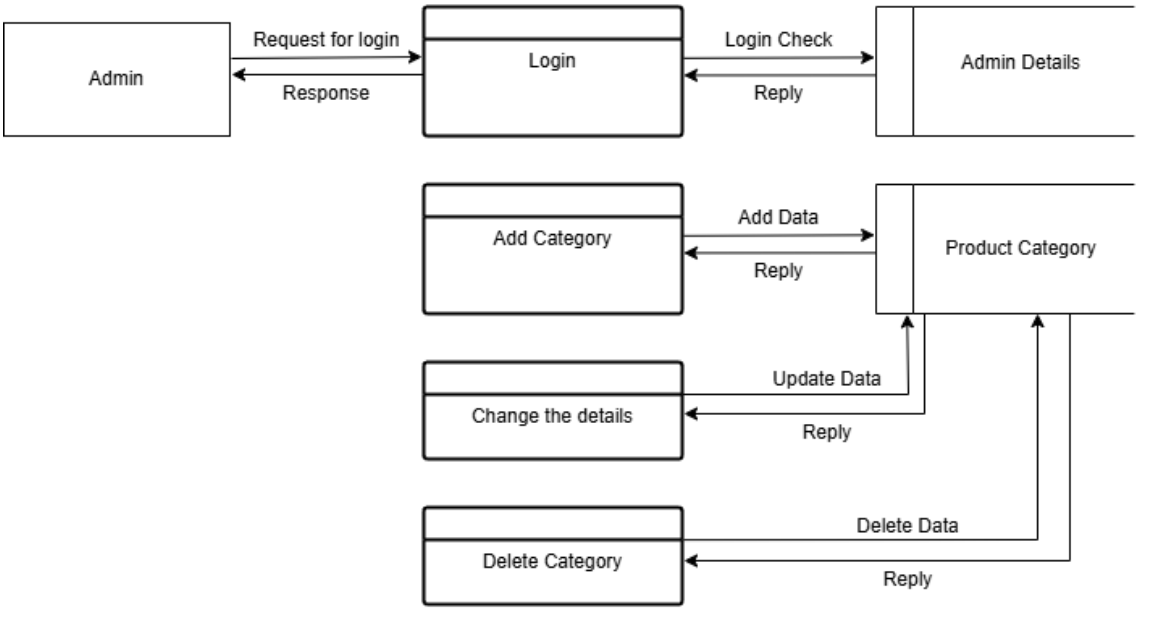
Samyak Manandhar

Level 1: DFD for Admin Side



Samyak Manandhar

Level 2: DFD for Product Catagory on Admin Side



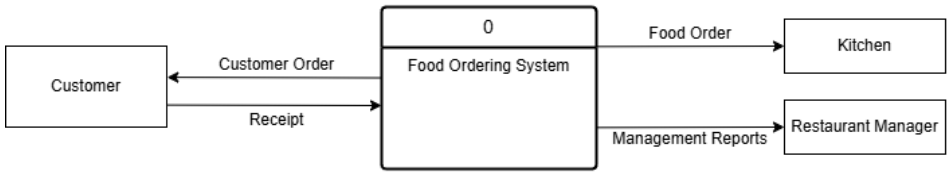
Samyak Manandhar

LAB 2: Data Flow diagram for Food Ordering System.

Food Ordering System:

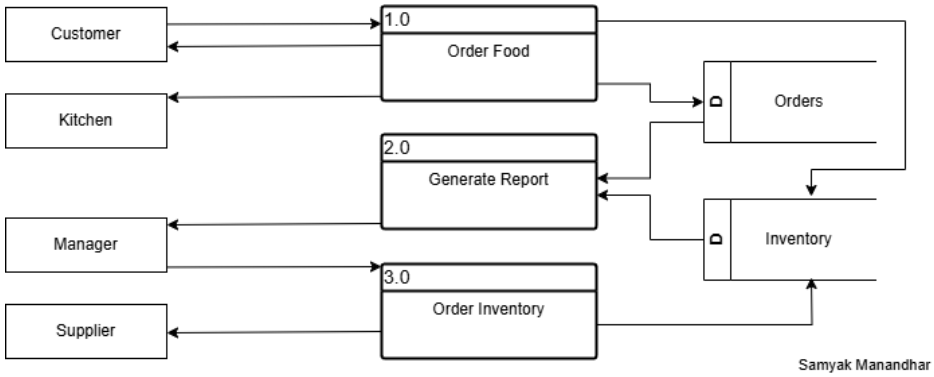
A Food Ordering System allows customers to browse menus, place orders, and make payments online, streamlining the ordering process for both users and restaurants. In a Data Flow Diagram (DFD), the system's data flow is visually represented, showing how information moves between users, restaurants, and the system's database. Key processes include user registration, menu selection, order placement, payment processing, and order tracking. External entities such as customers, restaurants, and delivery personnel interact with the system, while data stores manage user details, orders, and payment records.

Level 0: DFD for Food Ordering System



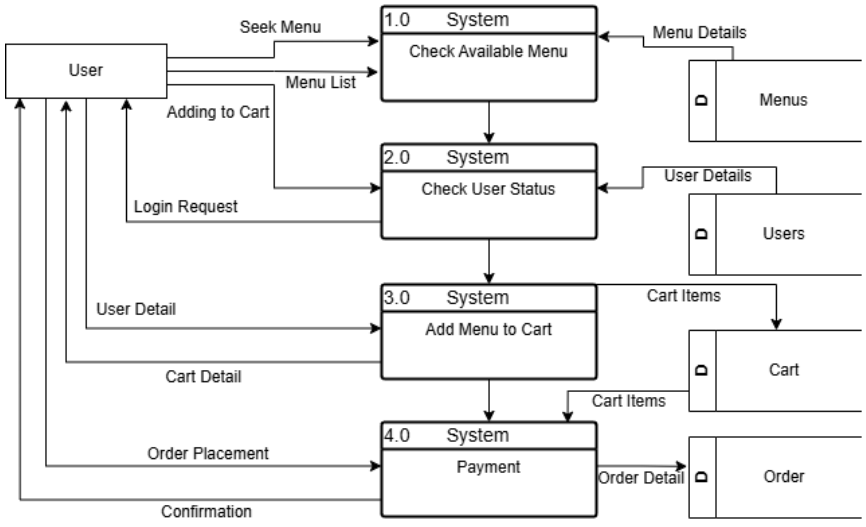
Samyak Manandhar

Level 1: DFD for Order Processing Procedure



Samyak Manandhar

Level 2: DFD for Customer, Kitchen, Manager, Supplier



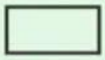
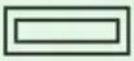
Samyak Manandhar

LAB 3: E-R Diagram for a Library Management System.

INTRODUCTION:

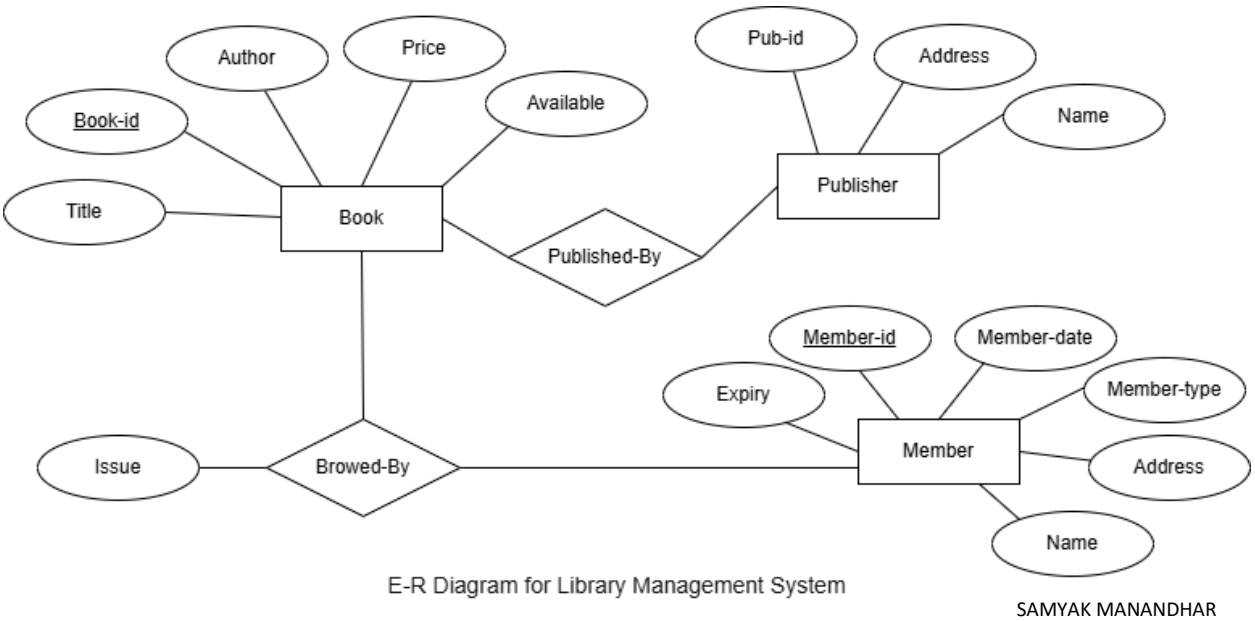
An ER (Entity-Relationship) Diagram is a graphical representation of a database that illustrates how entities (objects) relate to each other. It is widely used in database design to structure data before implementation. The diagram consists of entities, which represent real-world objects such as students, books, or employees, along with their attributes, which define their characteristics. Relationships indicate how entities interact, such as a student borrowing a book. Each entity has a primary key, a unique identifier, and relationships can have different cardinalities, such as one-to-one, one-to-many, or many-to-many. ER diagrams use specific symbols: rectangles for entities, ellipses for attributes, diamonds for relationships, and lines to connect them. These diagrams help developers and database designers visualize data organization, making it easier to create efficient relational databases.

Symbols used in ER-Diagram:

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

Library Management System:

A Library Management System (LMS) is designed to efficiently handle the management of books, members, and transactions within a library. The system helps automate key processes such as book cataloging, member registration, book borrowing, and return tracking, ensuring smooth library operations.To represent the data flow and relationships between various entities in an LMS, an Entity-Relationship Diagram (ERD) is used. The ERD visually illustrates how different entities (such as books, members, librarians, and transactions) are interconnected.



E-R Diagram for Library Management System

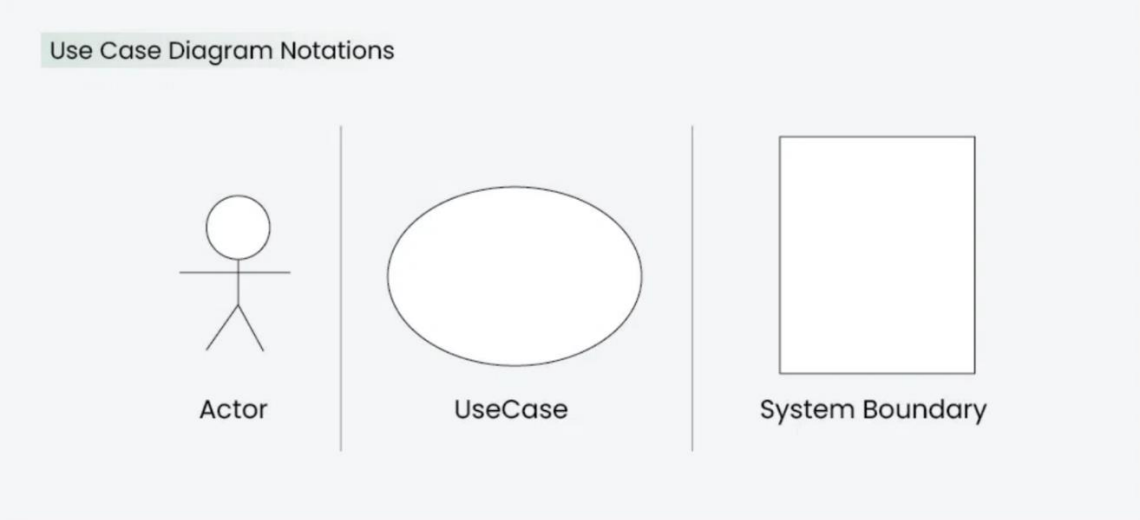
SAMYAK MANANDHAR

LAB 4: Use Case Diagram for an ATM System.

INTRODUCTION:

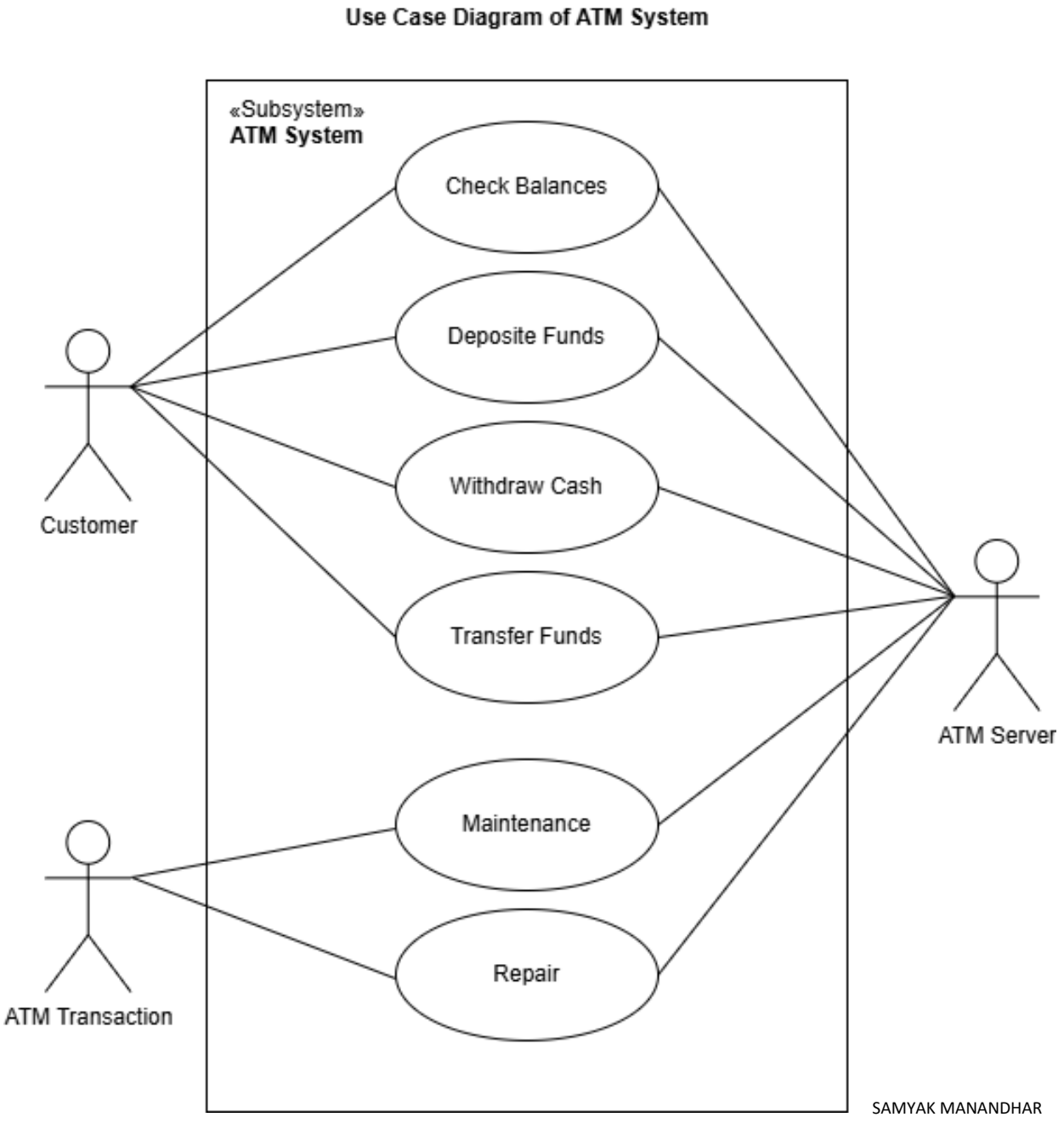
An use case diagram is another type of graphical representation used in system analysis and design, focusing on interaction between users (actors) and a system. It is visually depicted how various users interact with the functionality of the system rather than showing the flow of data like DFD. The main components of an use case diagram include actors, representing users or external systems; Use cases that are specific actions or systems that perform the system; The Sangh, which shows relationships between actors and use matters; And system limitations, which define the scope of the system. Use of case diagrams helps to understand system requirements, identify user interactions and design functional aspects of software.

Symbols used in Use Case Diagram:



Automated Teller Machine (ATM) System:

An Automated Teller Machine (ATM) System is designed to provide banking services such as cash withdrawal, balance inquiry, fund transfers, and deposit transactions to customers. It acts as an interface between the user and the bank's database, enabling secure and efficient financial transactions. Use Case Diagram is used to represent the interactions between users and the ATM system, illustrating the system's functionality in a simplified visual format. It helps in understanding how different actors (users) interact with the system's various features.

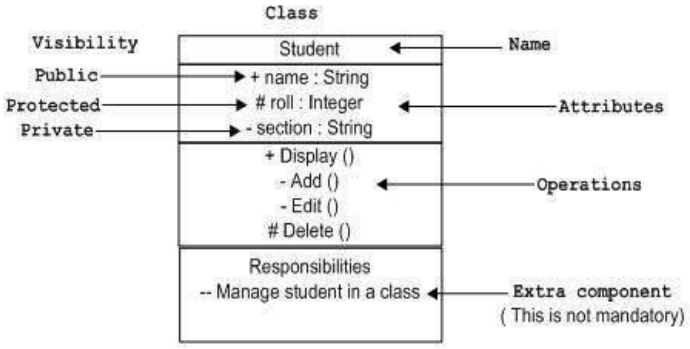


SAMYAK MANANDHAR

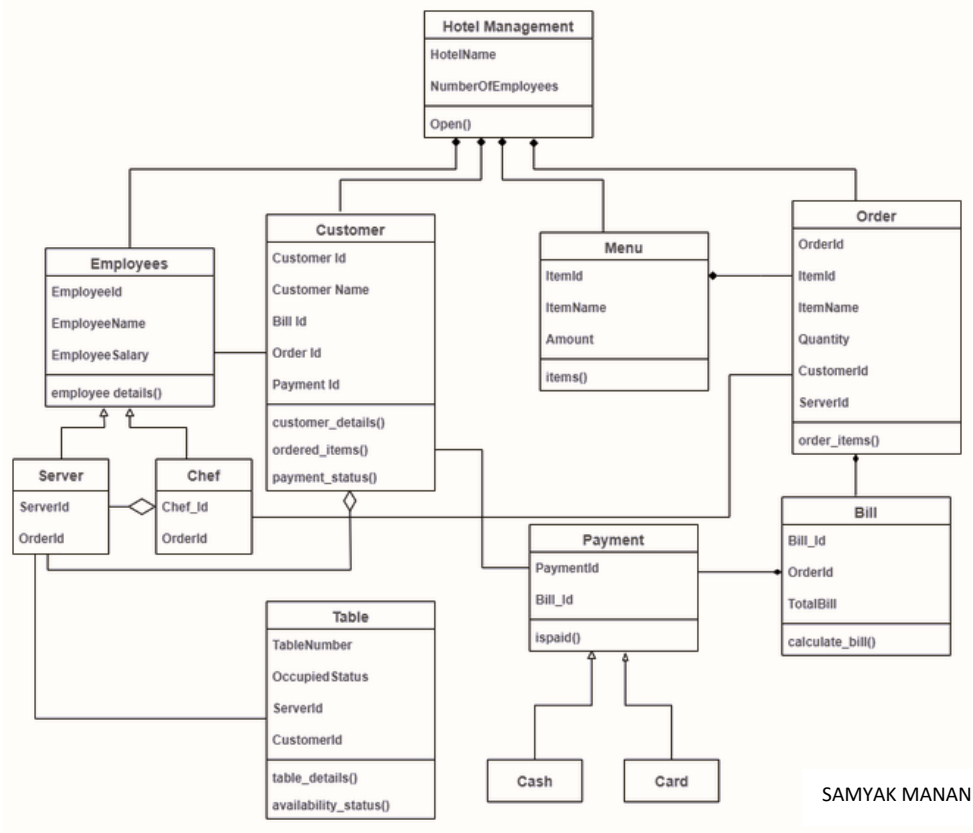
LAB 5: Class Diagram for a Hotel Management System.

INTRODUCTION:

A Class Diagram is a type of static structure diagram in Unified Modeling Language (UML) used to represent the structure of a system by showing its classes, their attributes, methods, and the relationships between them. In a class diagram, classes are represented by rectangles divided into three parts: the top section contains the class name, the middle section lists the class's attributes (properties), and the bottom section shows the class's methods operations. Relationships between classes, such as associations, generalizations (inheritance), and dependencies, are depicted with lines connecting the classes. These diagrams help developers understand the overall design of a system, its components, and how they interact, forming a blueprint for software development.



Class Diagram for a Hotel Management System:

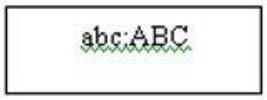
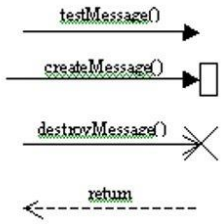


SAMYAK MANANDHAR

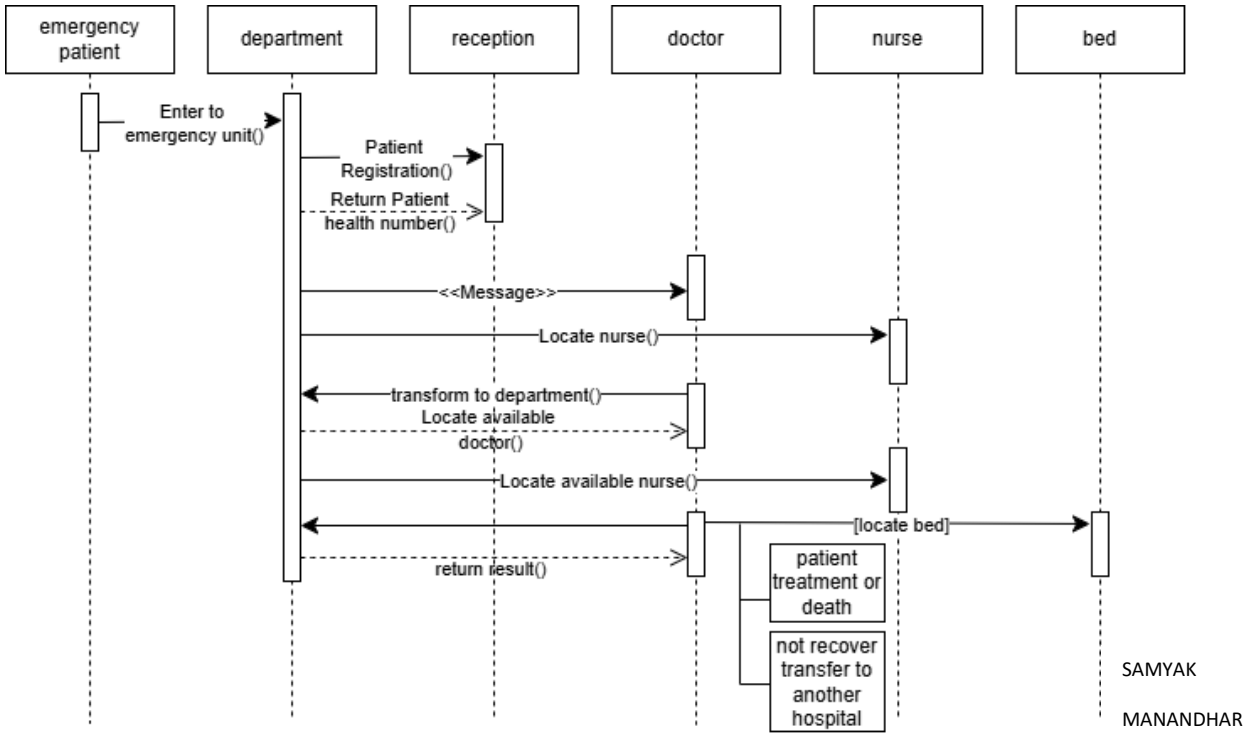
LAB 6: Sequence Diagram for a Hospital Management System.

INTRODUCTION:

A Sequence Diagram is a type of interaction diagram in Unified Modeling Language (UML) that models the flow of messages between objects or components in a system over time. It illustrates how processes or actions are carried out in a particular sequence. In a sequence diagram, objects or entities are represented by vertical dashed lines, and the messages exchanged between them are depicted by horizontal arrows. The diagram also highlights lifelines (the existence of an object over time) and can include conditions and loops to represent decision points or repetitive actions. Sequence diagrams are particularly useful for visualizing how different components of a system interact to achieve specific functionality in response to a particular scenario.

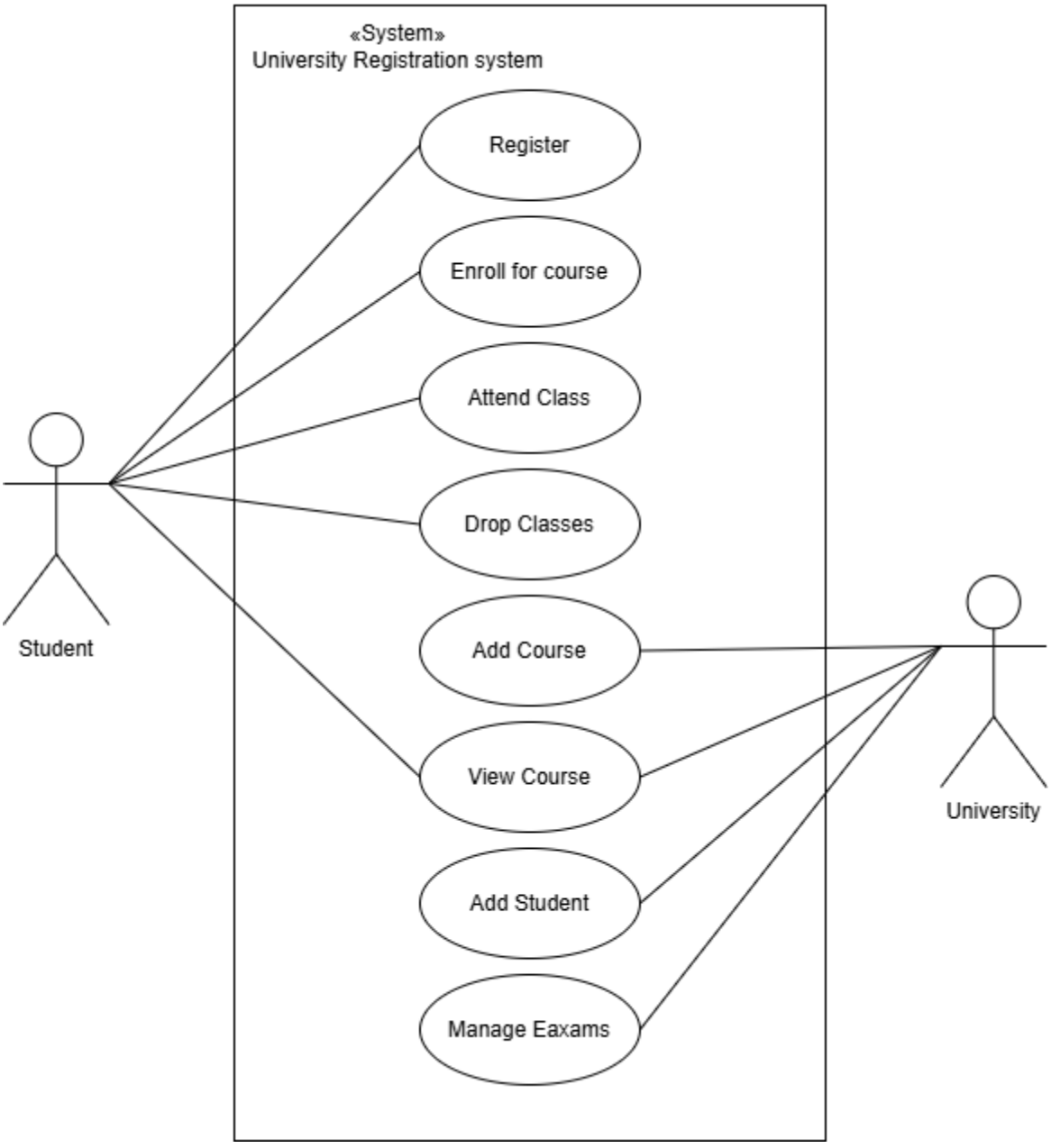
	Object: The primary element involved in a sequence diagram is an Object. A Sequence diagram consists of sequences of interaction among different obje cts over a period of time.	
	Message: The interaction between different objects in a sequence diagram is represented as messages. A messages is represented by a directed arrow.	

Hospital Management System:



LAB 7: Use Case Diagram for a University Registration System.

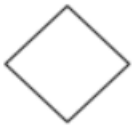
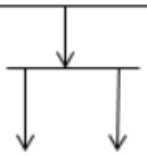
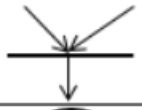
A University Registration System is designed to facilitate the enrollment process for students by automating course registration, fee payments, schedule management, and administrative tasks. It helps students, faculty, and administrators efficiently handle academic activities while ensuring smooth operations. A Use Case Diagram provides a visual representation of how different users interact with the system, defining the system's functionalities and user roles. It helps in understanding the system's scope and workflow.



LAB 8: Activity Diagram for an Airline Reservation System.

INTRODUCTION:

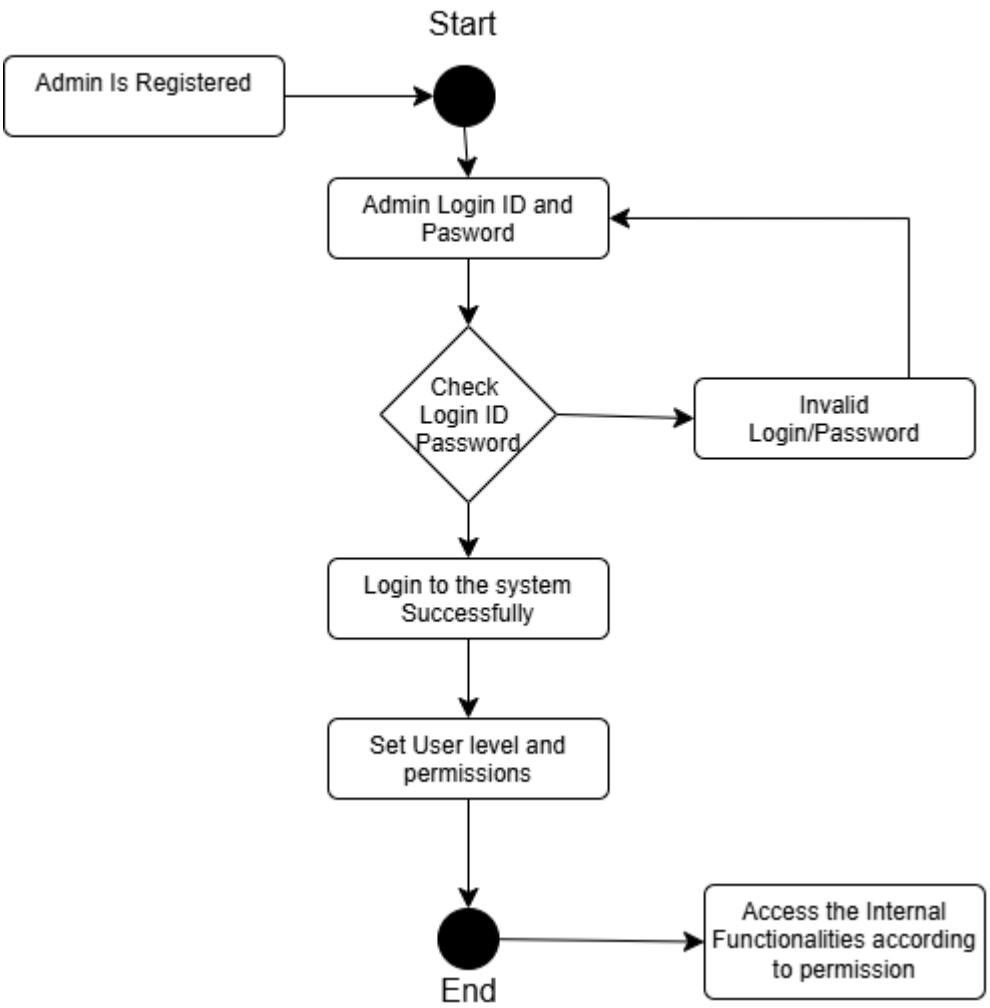
An Activity Diagram is a type of behavioral diagram in Unified Modeling Language (UML) used to model the workflow or the sequence of activities within a system. It illustrates the flow of control from one activity to another, showcasing the dynamic aspects of a system. In an activity diagram, activities are represented by rounded rectangles, and the flow of control is shown as arrows connecting these activities. Decision nodes (diamond shapes) represent points where the flow diverges based on conditions, while forks and joins indicate parallel execution paths. The diagram may also include start (filled circle) and end (bullseye) nodes to mark the beginning and completion of the workflow. Activity diagrams are useful for modeling business processes, use case scenarios, and complex workflows, providing a clear visual representation of how a system's processes are organized and executed.

Sr. No	Name	Symbol
1.	Start Node	
2.	Action State	
3.	Control Flow	
4.	Decision Node	
5.	Fork	
6.	Join	
7.	End State	

Activity diagram for Airline Reservation System :

An Airline Reservation System is designed to facilitate the process of booking, modifying, and managing flight reservations for passengers. It automates essential tasks such as searching for flights, selecting seats, making payments, and generating tickets, ensuring a smooth and efficient user experience. An Activity Diagram visually represents the flow of operations within the system. It outlines the sequence of activities from flight selection to ticket confirmation, helping to understand the system's workflow and identify potential optimizations.

Airline Reservation System:

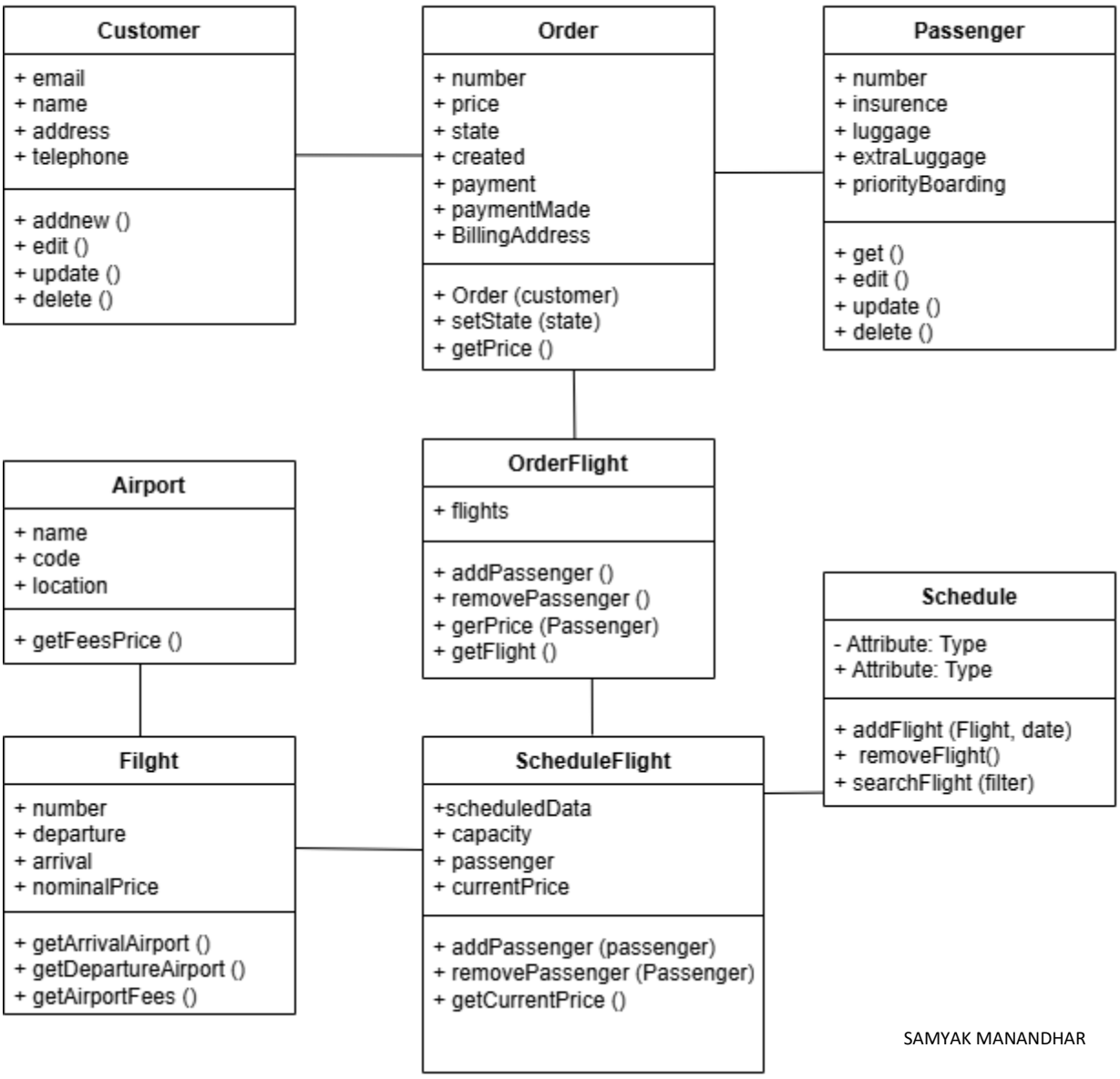


SAMYAK MANANDHAR

LAB 9: Class Diagram for an Airline Reservation System.

An Airline Reservation System is designed to manage flight bookings, passenger details, ticketing, and scheduling efficiently. It enables airlines to handle reservations, cancellations, seat assignments, and payments while ensuring a seamless experience for passengers. A Class Diagram is used to represent the structure of the system by defining its classes, attributes, methods, and relationships. It provides a blueprint for the system's object-oriented design and helps in understanding how different components interact.

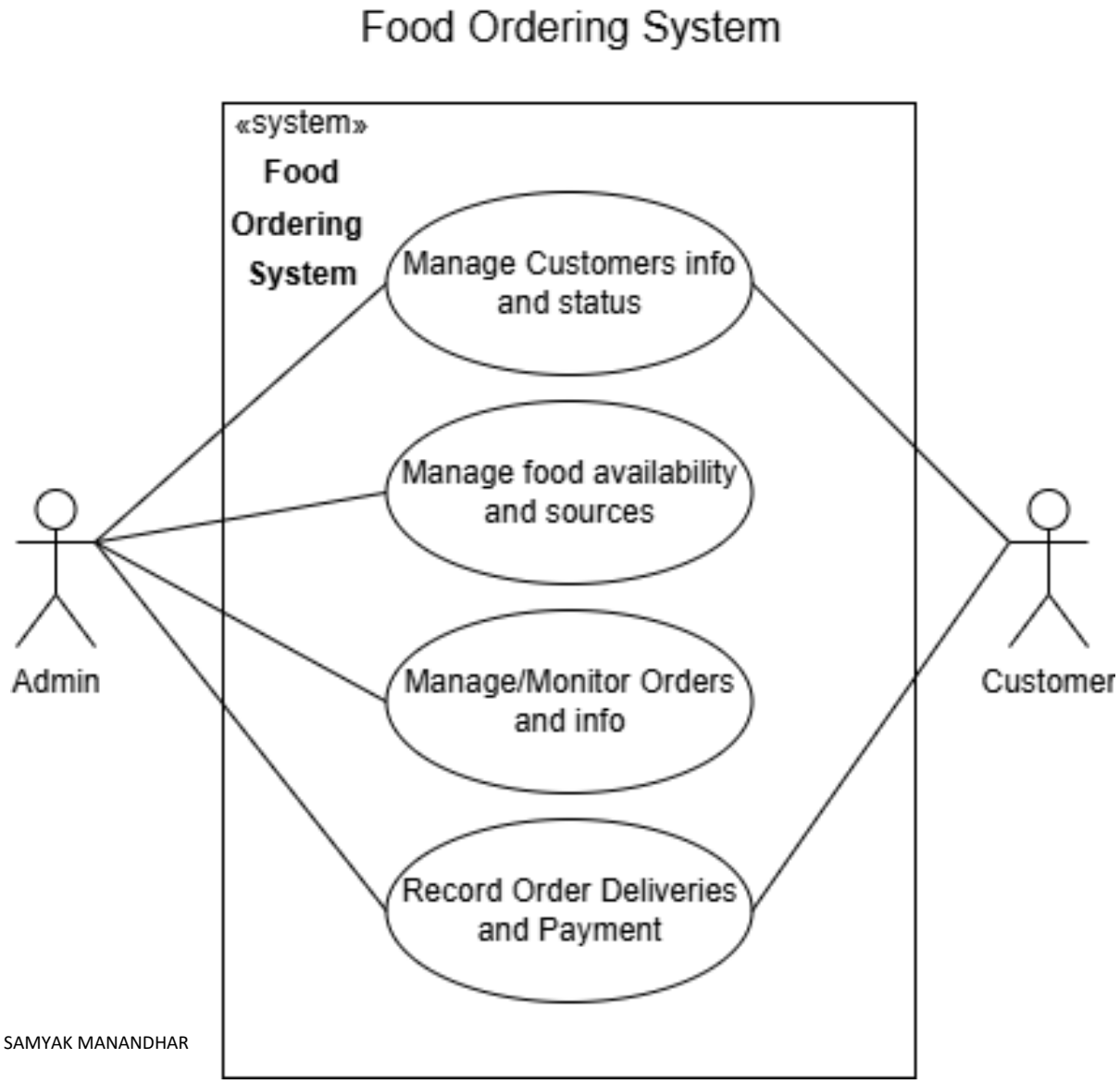
Class Diagram for an Airline Reservation System.



SAMYAK MANANDHAR

LAB 10: Use Case Diagram for a Food Ordering System.

A Food Ordering System is designed to streamline the process of ordering food from restaurants, whether for dine-in, takeaway, or delivery. It enables users to browse menus, place orders, make payments, and track delivery status while allowing restaurants to manage orders efficiently. A Use Case Diagram visually represents the interactions between different users (actors) and the system's functionalities. It helps in understanding the system's scope, identifying key operations, and defining how users interact with various processes.



LAB 11: Component Diagram for an Online Banking System.

INTRODUCTION:

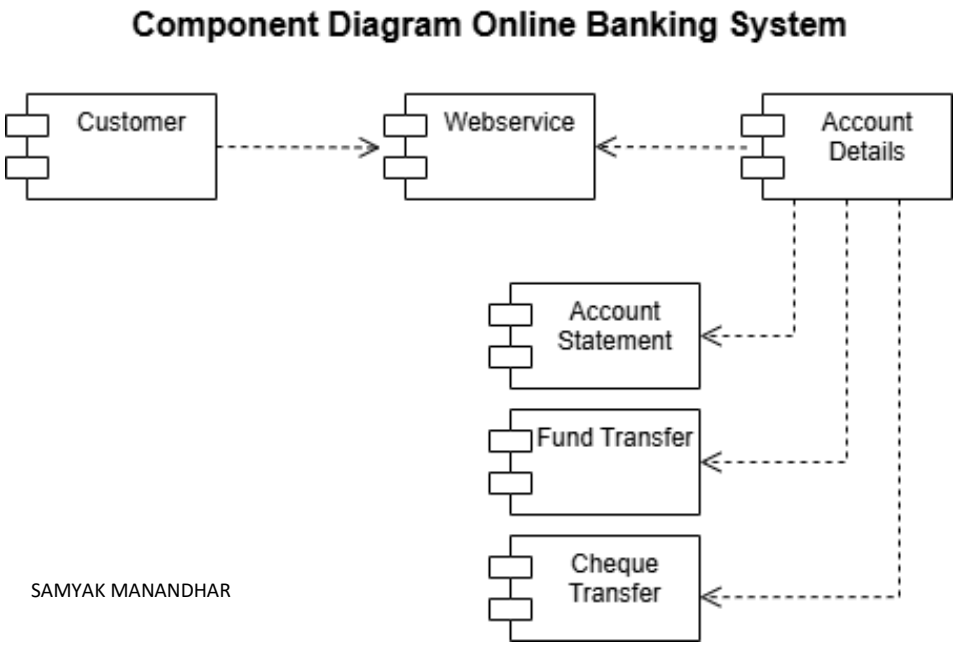
A Component Diagram is a structural diagram in UML (Unified Modeling Language) used to visualize the organization and relationships between various components in a system. It focuses on modeling the physical aspects of a system, such as software components, their interfaces, and how they interact with one another. Components represent modular parts of the system, such as libraries, executables, or modules, and are depicted as rectangles with optional interfaces. Interfaces define the contract or behavior that a component provides or requires, and they are represented using lollipop symbols for provided interfaces and socket symbols for required interfaces. Dependencies between components are shown using dashed arrows, indicating that one component relies on another. Ports, represented as small squares on the boundary of a component, specify interaction points for communication. Connectors illustrate the links between components or between components and their interfaces. Component diagrams are particularly useful during the design phase to plan the system's architecture, identify reusable components, and communicate the system's structure to stakeholders.

Symbol used in Component diagram:

Element	Symbol/Notation	Explanation
Component	«component» Name	Symbol for modules in a system (interaction and communication occur via interfaces).
Package	package	A package combines multiple elements in a system (e.g. classes, components, interfaces) into a group.
Artifact	<<artifact>> manual.jar	Artifacts are physical units of information (e.g. source code, *.exe* files, scripts, documents) that are created or required during a system's development process or at runtime.
Provided interface	)	Symbol for one or more clearly defined interfaces that provide functions, services or data to the outside (the open semi-circle is also called a socket).
Required interface	○	Symbol for a required interface that receives functions, services or data from the outside (the circle-with-stick notation is also referred to as a lollipop).
Port	□	This symbol indicates a separate point of interaction between a component and its environment.
Relationship	———	Lines act as connectors and show relationships between components.
Dependency	------>	This special connector indicates a dependency between two parts of a system (not always explicitly shown).

Online Banking System:

An Online Banking System is designed to provide customers with secure and convenient access to banking services such as fund transfers, account management, bill payments, and transaction history. It enables users to perform financial operations remotely via a web or mobile interface while ensuring security and efficiency.

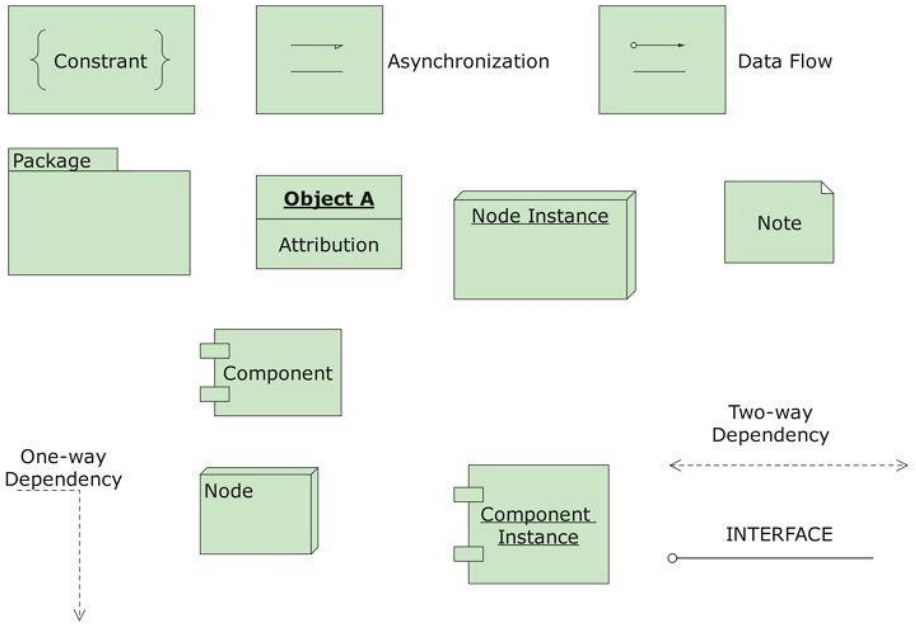


LAB 12: Deployment Diagram for an E-Commerce Website.

INTRODUCTION:

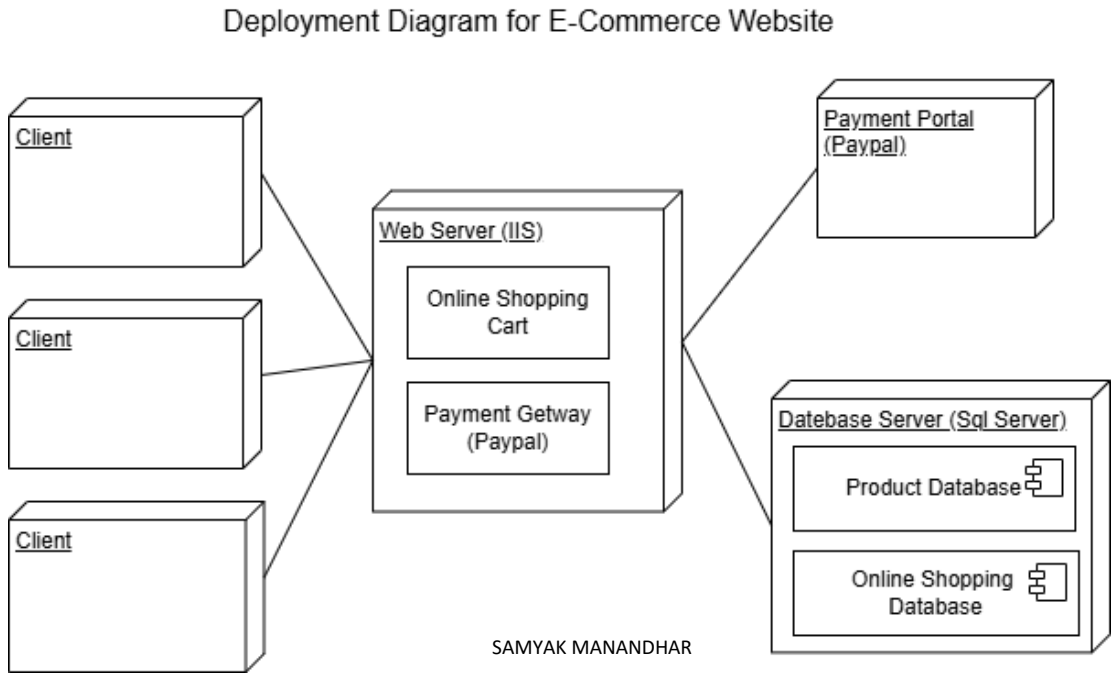
A Deployment Diagram is a type of UML (Unified Modeling Language) diagram used to model the physical deployment of software components or artifacts across hardware nodes in a system. It provides a visual representation of how software is distributed and executed on hardware, such as servers, devices, or machines, and how these nodes interact with each other. Key elements in a deployment diagram include nodes (representing hardware or execution environments), artifacts (representing software components like executables or libraries), and communication paths (showing how nodes are connected). Deployment diagrams are particularly useful for planning system architecture, understanding the distribution of software across hardware, and ensuring efficient communication between nodes. They are commonly used during the design and deployment phases of a project to visualize the system's topology and ensure that all components are properly allocated and connected.

Symbols used in Deployment Diagram:



E-Commerce Website:

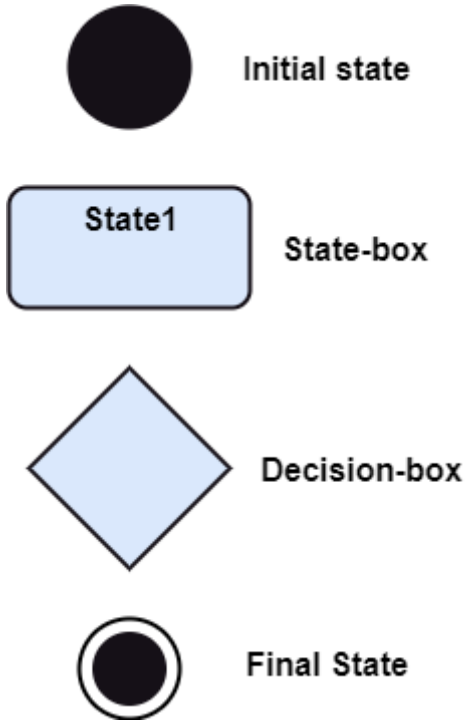
An E-Commerce Website enables users to browse products, place orders, make payments, and track deliveries. It consists of multiple components working together to ensure a smooth shopping experience.



LAB 13: State Diagram for an Online Examination System.

INTRODUCTION:

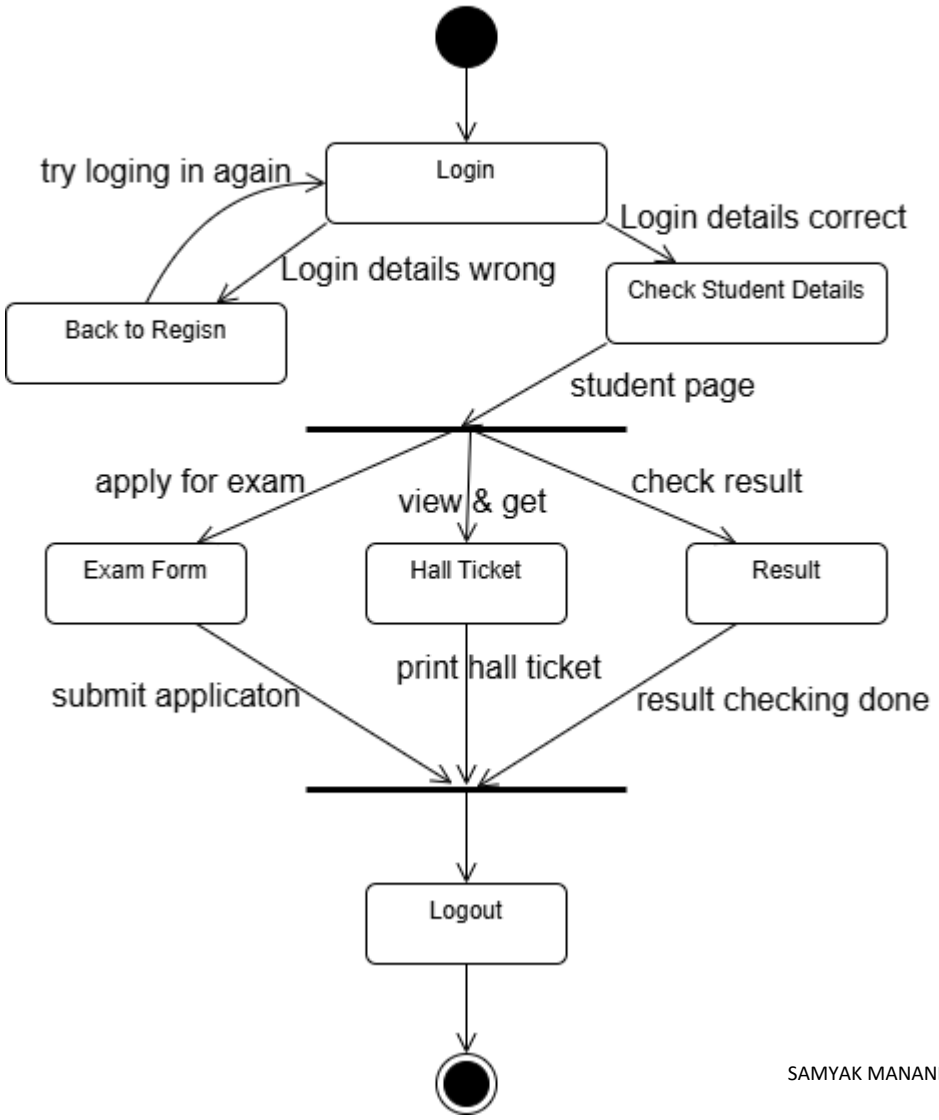
A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. It visually represents the states of an object or system and the transitions between these states in response to events. Each state is depicted as a node (usually a rounded rectangle), and transitions are shown as arrows connecting these nodes, labeled with the event or condition that triggers the change. State diagrams are particularly useful for modeling the dynamic behavior of systems, such as software applications, hardware systems, or business processes, by illustrating how the system progresses from one state to another. They help in understanding, designing, and documenting complex systems by providing a clear and structured view of their behavior over time.



Online Examination System:

An Online Examination System allows students to take exams digitally while enabling administrators to manage test creation, scheduling, and result processing. It ensures a seamless and secure assessment process. A State Diagram represents the different states an exam session goes through, showing transitions between states based on user interactions. This State Diagram for the Online Examination System helps in understanding the system's behavior, ensuring smooth transitions, and improving user experience.

State Diagram for an Online Examination System.

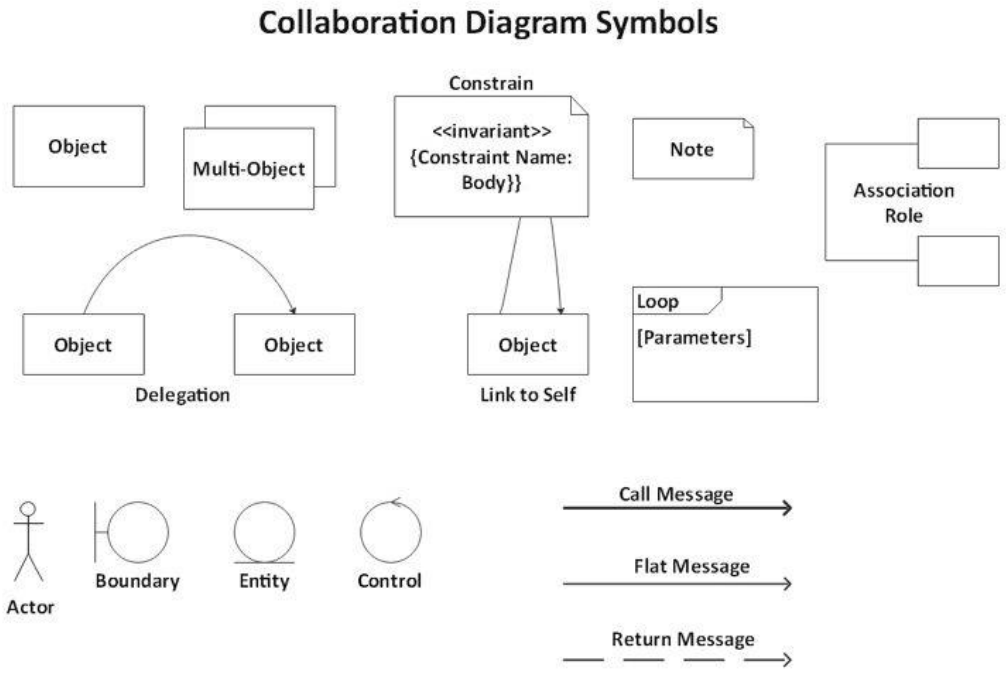


SAMYAK MANANDHAR

LAB 14: Collaboration Diagram for a Customer Support System.

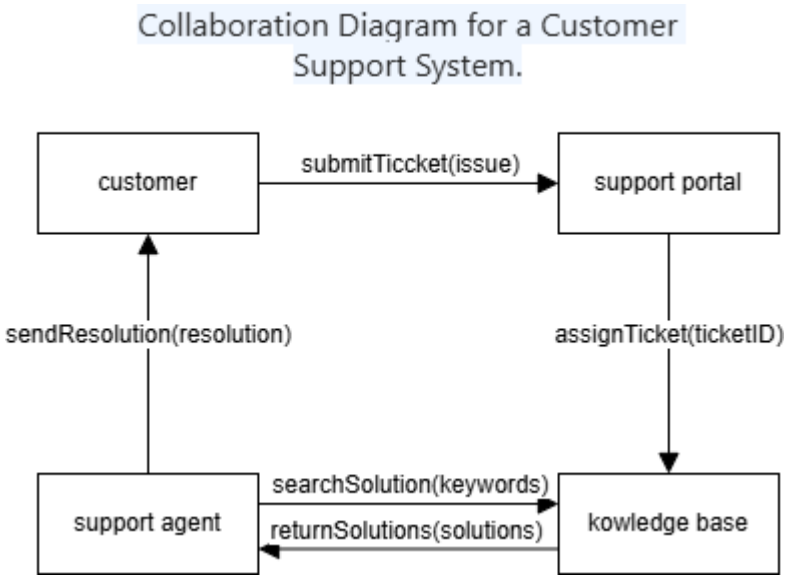
INTRODUCTION:

A Collaboration Diagram, also known as a Communication Diagram in UML 2.x, is a type of interaction diagram that illustrates how objects or roles within a system interact and collaborate to achieve a specific task or use case. It focuses on the relationships between objects and the sequence of messages exchanged between them. In a collaboration diagram, objects are represented as rectangles, and the connections between them, called links, are shown as solid lines. Messages, depicted as labeled arrows along these links, indicate the communication between objects, with sequence numbers (e.g., 1, 2, or 1.1, 1.2) specifying the order in which the interactions occur. Collaboration diagrams are particularly useful for visualizing the flow of control and the relationships between objects in a system, providing a clear understanding of how objects work together to accomplish a specific functionality. They are often used alongside sequence diagrams to offer a different perspective on object interactions, emphasizing the structure of relationships rather than the explicit order of messages. This makes collaboration diagrams a valuable tool for designing and analyzing the dynamic behavior of a system.



Customer Support System:

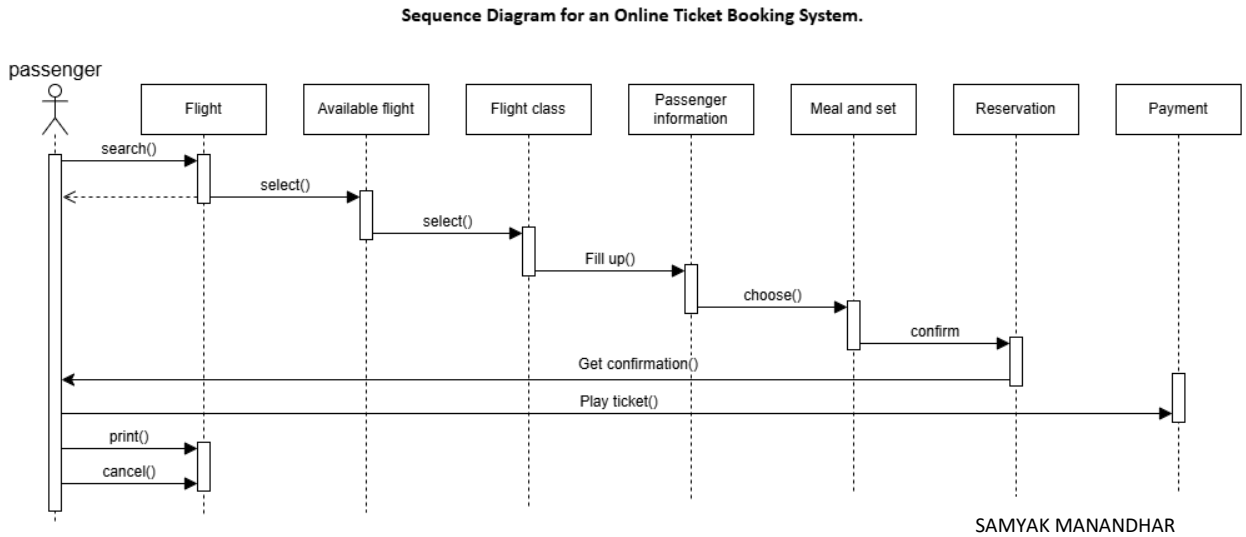
A Collaboration Diagram for a Customer Support System visually represents interactions between various system components, illustrating how objects communicate to handle customer queries and issues. It highlights the roles of key entities such as customers, support agents, ticketing system, knowledge base, and admin while showcasing message exchanges and workflows in resolving support requests efficiently. This diagram helps in understanding system behavior and improving service operations.



SAMYAK MANANDHAR

LAB 15: Sequence Diagram for an Online Ticket Booking System.

A Sequence Diagram for an Online Ticket Booking System illustrates the step-by-step interactions between users and system components during the booking process. It showcases the flow of messages between key entities such as customers, the booking interface, payment gateway, database, and ticketing system to ensure a smooth transaction. This diagram helps in understanding the order of operations, ensuring efficient ticket reservations and payment processing.



SAMYAK MANANDHAR