

QUES1. You are part of a development team working on a complex software system. The team is experiencing difficulties in understanding and coordinating different modules of the system. How can UML diagrams help in this situation?

- a. Which types of UML diagrams would be most effective in clarifying the architecture and interactions within the software system?
- a. The UML diagrams that would be most effective in clarifying the structure and architecture within the software system are:

CLASS DIAGRAM: Shows the static structure of the system, including classes, attributes, methods and relationships. This clarifies the building blocks and their interconnections.

COMPONENT DIAGRAM: Includes physical system components and their dependencies. This helps understand how modules interact at a broader level.

SEQUENCE DIAGRAM: Captures the dynamic flow of messages and interactions between objects in a specific use case. This clarifies how components collaborate for particular functionalities.

ACTIVITY DIAGRAM: Depicts the overall workflow of a complex process, breaking it down into activities and decisions. This aids in understanding the bigger picture and potential bottlenecks.

- b. Can you explain how a sequence diagram could be used to illustrate the flow of control between different components of the system?

A sequence diagram visualizes the interaction between objects across time. Each vertical line (Lifelines) in the diagram represents an object / component, and the horizontal arrows depict messages exchanged between them. Activation bars depict the active period of an object during the interaction.

By following the sequence of messages, the developer can visualize the order of interactions and understand how control flows b/w components during execution.

- c. How would you ensure that the UML diagrams are kept up-to-date as the software evolves during development?

To ensure this, it is essential to integrate diagram maintenance into the form of automated tools to generate UML diagrams based on code changes, regular reviews, version control etc. into the development process.

This can be done by assigning responsibility for updating diagrams to specific team members, conducting regular reviews and updates during Sprint and iteration planning meetings.

- d. What strategies would you use to communicate the UML diagrams effectively within the development team?

Effective communication of UML diagrams can be done by using the following strategies:

→ **REGULAR REVIEWS** : conduct regular reviews of UML diagrams during team meetings to ensure everyone understands the system architecture and interactions.

→ **DOCUMENTATION** : Incorporate UML diagrams into project documentation to serve as a reference for developers and stakeholders.

- TRAINING: Provide training sessions to familiarize team members with UML notation and implementation of diagrams.
- TOOL SUPPORT: Utilize UML modeling tools that facilitate collaboration and version control; allow team to update diagrams easily.

QUES2. You are tasked with designing a new online shopping platform for a retail company. Describe how you would use UML diagrams to model the system architecture.

a. How would you begin the UML modeling process for this project?

a. Beginning the UML process involves:

i.) REQUIREMENT ELUCIDATION / ANALYSIS :-

Conduct thorough interviews and workshops with stakeholders (like clients, users, devs) to understand functionalities, features and non-functional requirements.

ii.) IDENTIFY ACTORS AND USE CASES :-

Depict the actors (like customer, admin, delivery service) and their interactions with the system through use cases (browse products, add to cart, checkout, etc.)

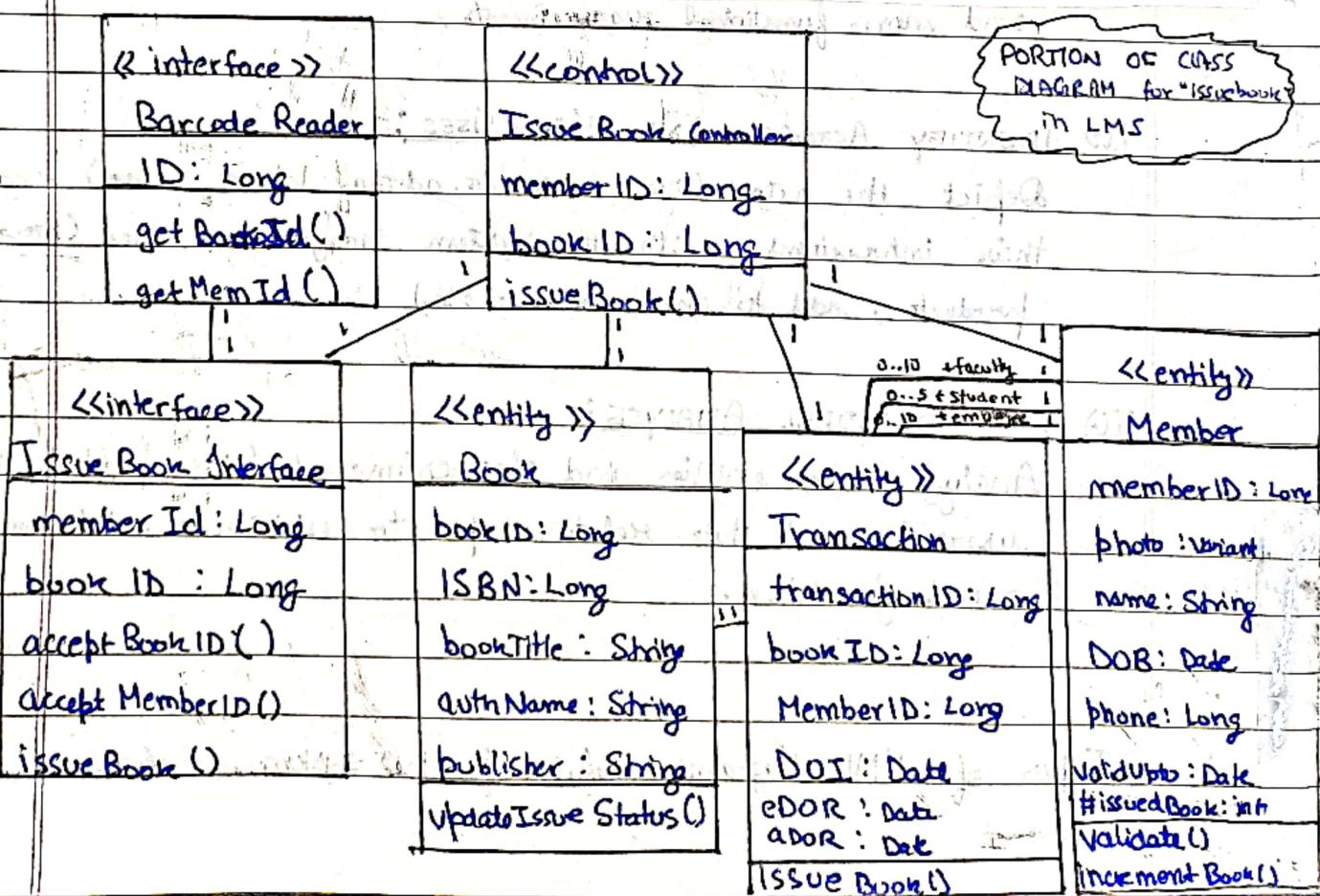
iii.) DOMAIN MODEL ANALYSIS :-

Analyze core entities and classes involved (like product, order, user) and their relationships to establish a solid domain model foundation.

b. What types of UML diagrams would be most useful in designing the online shopping platform?

b. Types of UML diagrams useful for the system include:

- i) Use Case Diagrams: To represent user interactions and system functionality.
- ii) Class Diagrams: To depict the structure of the system, including classes, attributes and relationships.
- iii) Sequence Diagrams: To illustrate the flow of interaction b/w objects over time.
- iv) Activity Diagrams: To model the workflow and business process within the platform.
- v) Components Diagram: To show the components of the system and their relationships.
- c. Can you provide an example of how you would use a class diagram to represent the entities and relationships in the system?



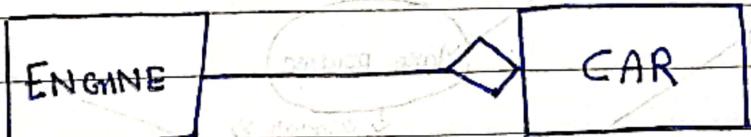
- d. How would you ensure that the UML diagrams accurately capture the requirements and functionalities of the online shopping platform?

- **VALIDATION AND ITERATION:** Regularly review diagrams with stakeholders to ensure they reflect requirements and are clear, concise and consistent.
- **TOOL SUPPORT:** Utilize UML modelling tools that enforce rules, check syntax and generate code or documentation, reducing manual errors.
- **MAINTAIN DOCUMENTATION:** Track changes, decisions, and rationale behind design choices in documentation linked to the diagrams for future references.

QUES 3.

- a. How is an aggregation relationship represented in a class diagram?

In a class diagram, an aggregation relationship is depicted using a hollow diamond notation at one end of a connecting line between two classes. The line typically points from the "whole" class (the aggregate) to the "part" class (the component).

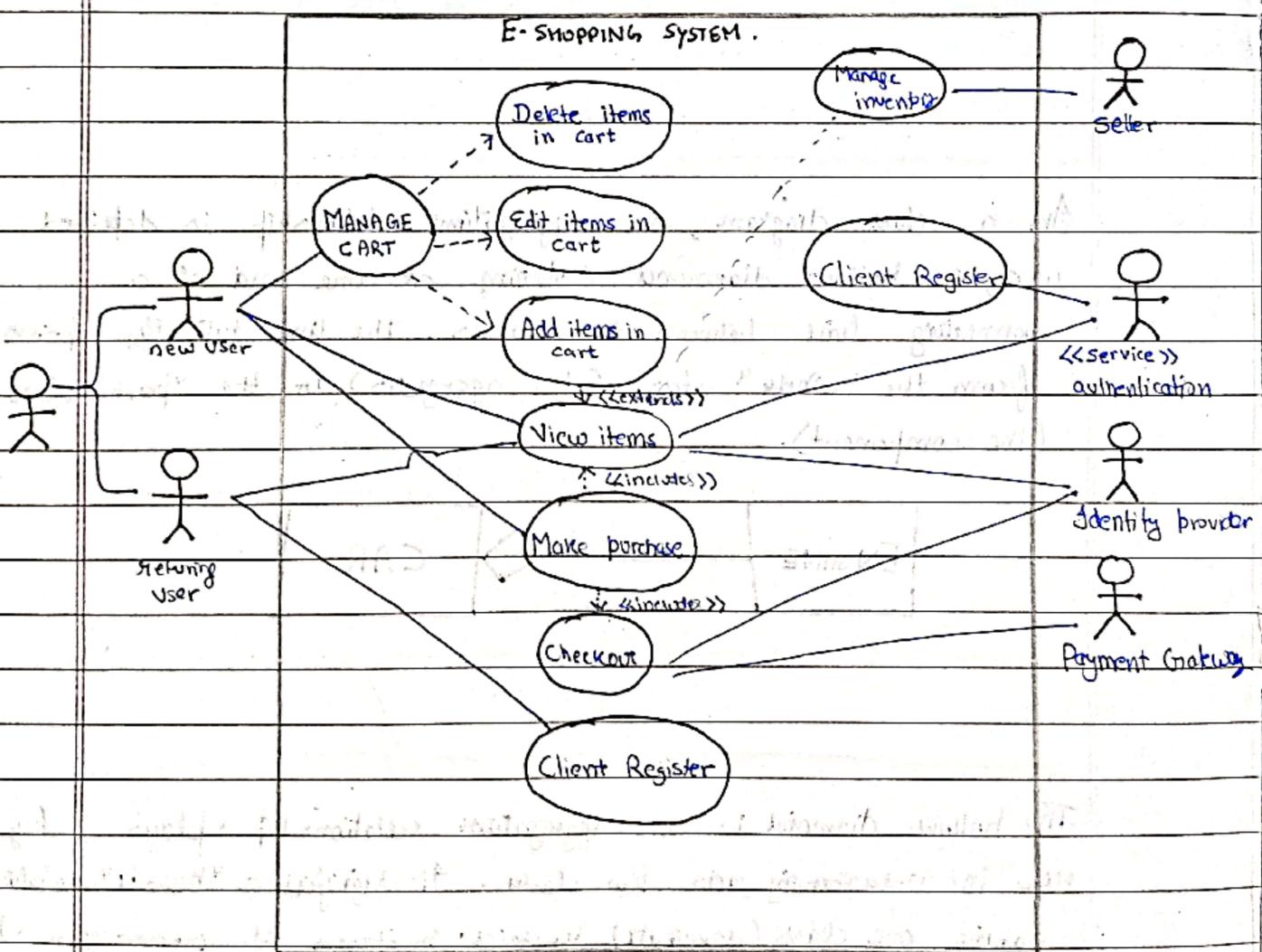


- b. What is the significance of the hollow diamond notation in an aggregation relationship?

The hollow diamond in an aggregation relationship plays a key role in understanding relationships between classes. It signifies a "has-a" relationship, meaning one class (aggregate) "contains" instances of another class (part).

- "Has-a" relationship : Aggregate class contains parts (Eg. Car has wheels).
- Independent life cycles : Parts can exist outside the aggregate. (Car deleted, wheel remain).
- Weaker connection than composition (filled diamond).

QUES4. Draw a use case diagram of e-shopping system.



QUES5. Write the use case description for cash withdrawal from an ATM machine?

INTRODUCTION: This use case allows bank customers to withdraw cash from their accounts using ATM.

ACTORS:

Bank Customer

Banking System (ATM and infrastructure)

PRECONDITIONS: ATM is optional and customer has a valid bank card.

POST CONDITIONS: Customer receives cash and receipt.

Bank debits customer's account and record transaction.

EVENT FLOW:

BASIC FLOW:

1. Customer inserts card and enter PIN.
2. System verifies card and PIN.
3. Customer selects "Withdraw" option.
4. System verifies account balance and ATM cash availability.
5. Customer selects or enters withdrawal amount.
6. System checks if amount complies with limits.
7. System debits account and dispenses cash.
8. Customer takes cash and card.

ALTERNATE FLOWS

- **INVALID CARD/PIN:** Card ejected, user notified.
- **INSUFFICIENT FUNDS:** Transaction rejection, contact bank advised.
- **DAILY WITHDRAWAL LIMIT EXCEEDED:** Lower amount requested.
- **ATM OUT OF CASH:** Max. available amount offered.
- **CARD GET STUCK:** ATM detained, bank notified.
- **CUSTOMER FORGETS CARD:** ATM retains card, bank notified.

SPECIAL REQUIREMENTS:

- Comply with security and privacy regulations.
- Meet accessibility standards for users with disabilities.
- Handle network connectivity issues gracefully.

ASSOCIATED USE CASES:

- Check Account Balance
- Deposit Cash
- Transfer Funds

END