Software Engineering Task-7

Kavya Aggarwal

HU22CSEN0100288

AIM: Try simple architecture and design of modules. Represent in activity, sequence, collaboration diagrams(UML).

Introduction

UML is a standardized modeling language that provides a versatile, flexible, and user-friendly method for visualizing system design. It helps software engineers, system architects, and businesses model, design, and analyze systems effectively.

Types of UML Diagrams

UML diagrams can be categorized into two main types:

- **Structural Diagrams** (Capture the static aspects of a system)
  - Class Diagram
  - Object Diagram
  - Component Diagram
  - Deployment Diagram
- **Behavioral Diagrams** (Illustrate the dynamic aspects of a system)
  - Use Case Diagram
  - Sequence Diagram
  - Collaboration Diagram
  - State Chart Diagram
  - Activity Diagram

Structural Diagrams

Class Diagram

A class diagram represents the static structure of a system by showing its classes, attributes, methods, and relationships.

Example: Online Shopping System

- **Classes:** Customer, Order, Product, Payment
- **Relationships:** A Customer can place multiple Orders, an Order contains multiple Products.

Object Diagram

Object diagrams show instances of classes and their relationships at a specific point in time.

Example: A snapshot of an Online Shopping System

- **Objects:** John:Customer, Order123:Order, Laptop:Product
- **Relationships:** John has placed Order123 containing a Laptop.

Component Diagram

Component diagrams illustrate how different software components interact with each other.

Example: Hospital Management System

- **Components:** Patient Management, Billing, Appointment Scheduling
- **Interactions:** Billing interacts with Patient Management for generating invoices.

Deployment Diagram

Deployment diagrams visualize system hardware and software deployment.

Example: Web Application Deployment

- **Nodes:** Web Server, Database Server

- **Software Components:** Frontend (React.js), Backend (Node.js), Database (MySQL)

Behavioral Diagrams

Use Case Diagram

Use case diagrams illustrate system functionalities and interactions with external actors.

Example: Library Management System

- **Actors:** Librarian, Member
- **Use Cases:** Borrow Book, Return Book, Pay Fine

Sequence Diagram

Sequence diagrams show the interaction between objects in a sequential manner.

Example: ATM Transaction

1. The User inserts a card.
2. The ATM requests authentication.
3. The User enters a PIN.
4. The ATM verifies and processes the transaction.

Collaboration Diagram

Collaboration diagrams focus on objects and their interactions.

Example: Food Delivery System

- **Objects:** Customer, Restaurant, Delivery Service
- **Interactions:** Customer places an order -> Restaurant prepares food -> Delivery Service delivers food.

State Chart Diagram

State diagrams define different states of an object and how transitions occur based on events.

Example: Traffic Light System

- **States:** Red, Green, Yellow
- **Transitions:** Green -> Yellow -> Red

Activity Diagram

Activity diagrams illustrate the flow of control in a system.

Example: Online Order Process

1. User selects items.
2. User makes payment.
3. Order is confirmed.
4. Order is shipped.

Conclusion

UML diagrams provide a comprehensive way to visualize both the static and dynamic aspects of a system. By understanding and using these diagrams effectively, developers can ensure better system design, analysis, and communication among stakeholders.