1ICPC317

AY 2024-25

SDLC Laboratory

Quality Laboratory Manual

Experiment No. 06

To draw structural view diagram: Class diagram, Objet diagram.

Course Instructor –

Mr. Sharanabasava Raddi

ASSISTANT PROFESSOR

Experiment No. 06

Title of Experiment: To draw structural view diagram: Class diagram, Object diagram

Aim of Experiment: To visualize the structure of a system and the relationships between its components and draw class diagram, Object diagram for better understanding.

System Requirements – Win 10 and above OS, 4GB RAM, 2.33 GHz Processor

Software/s Requirement – StarUML

Experiment Objectives:

- To visualize the structure of the system
- To identify and understand the components of the system.
- To understand the relationship between the components of the system.
- To identify and understand the functions and attributes of each components.

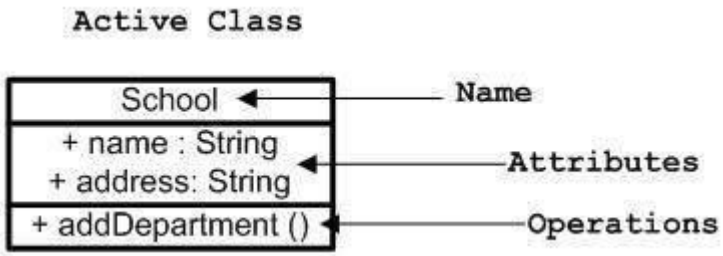
Experiment Outcomes:

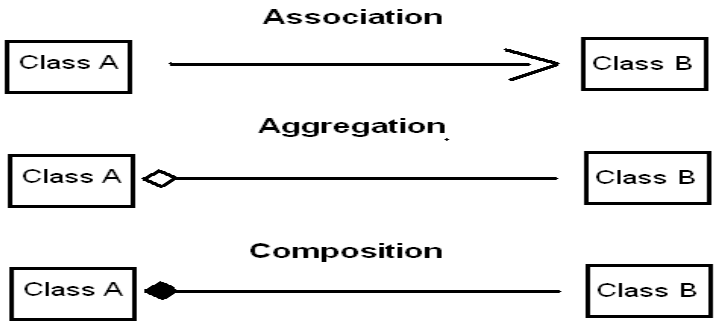
- Class diagram based on the requirement and better visualization.
- Represent the system with Class and Object diagrams.
- Object oriented system design and modeling.

Theory:

Class Diagram:

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects.





Class Name:

- The name of the class appears in the first partition.

Class Attributes:

- Attributes are shown in the second partition.
- The attribute type is shown after the colon.

Class Operations (Methods):

- Operations are shown in the third partition. They are services the class provides.
- The return type of a method is shown after the colon at the end of the method signature.
- The return type of method parameters are shown after the colon following the parameter name. Operations map onto class methods in code

Class Visibility:

The +, - and # symbols before an attribute and operation name in a class denote the visibility of the attribute and operation.

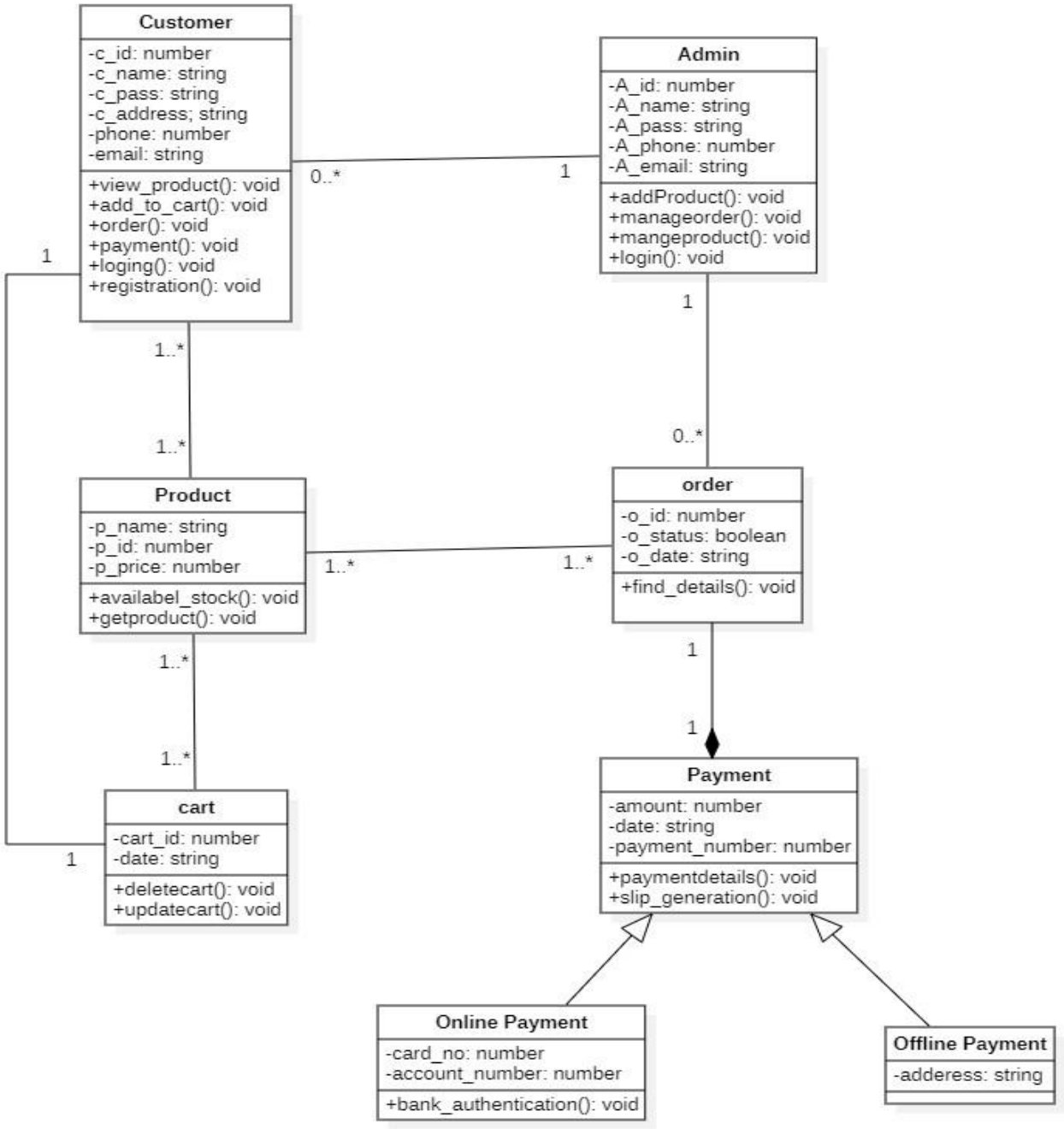
- + denotes public attributes or operations
- - denotes private attributes or operations
- # denotes protected attributes or operations

Object Diagram:

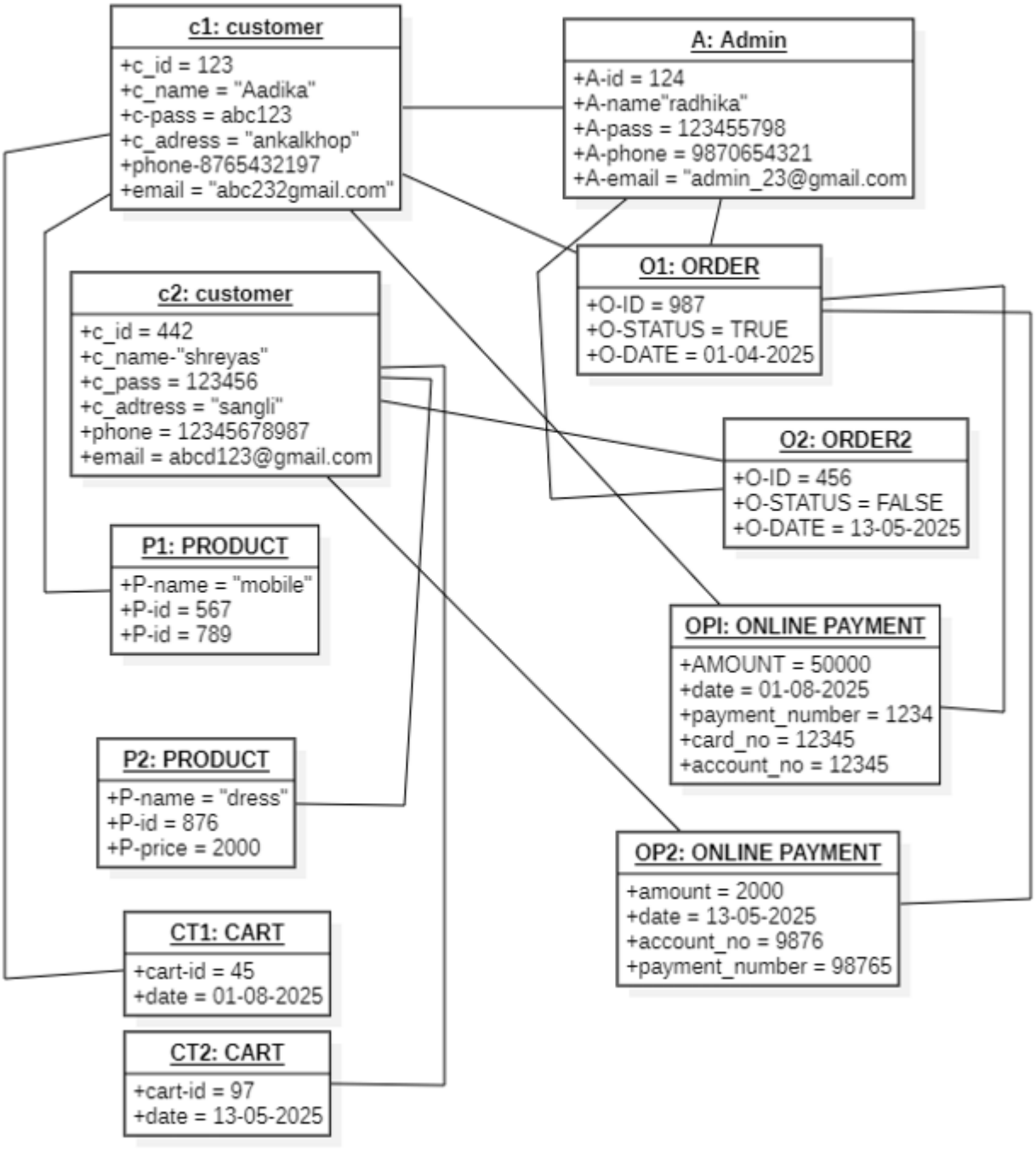
The object diagram holds the same purpose as that of a class diagram. The object diagram is actually similar to the concrete (actual) system behaviour. The main purpose is to depict a static view of a system.

Following are the purposes enlisted below:

- It is used to perform forward and reverse engineering.
- It is used to understand object behaviour and their relationships practically.
- It is used to get a static view of a system.
- It is used to represent an instance of a system.



Class Diagram



Object Diagram

Observations:

- Class diagram helps to represent the structural view of the system
- Class diagram gives clear information on the attributes and methods of a class
- Helps to understand the relationship between two or more classes
- Also helps how to impose access restrictions for particular attribute and a method.

Conclusion:
The experiment successfully demonstrated, the way of representing the structural view of a software application for better understanding of the system.

Expected Oral Questions:

1. What is class?
2. Define attribute?
3. What is an Object?
4. Define a method?
5. Difference between Aggregation and Composition?
6. What are the different access specifiers available for an attribute and method?
7. What is inheritance?

FAQs in Interview:

1. What is class diagram?
2. Define the objective of Class and Object diagram?
3. What is an object?
4. What is the use of association?
5. Define composition?
6. Define aggregation