

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
Computer Engineering Department
Program: B. Tech. AI Semester: VI

Course: Software Engineering
List of Experiments

Faculty: Prof. Payal Mishra

| Exp No. | Title | Prerequisite* | CO# |
|---------|--|--|-----|
| 1 | Finalize the problem statement and conduct the requirement gathering | C, C++, Software Development Life Cycle /Literature survey | CO1 |
| 2. | Estimation of Project Metrics | Software Development Life Cycle /Literature survey | CO4 |
| 3. | Modeling UML Use Case diagrams and capturing Use Case scenarios | Process Model, DFD, Requirement Engineering | CO2 |
| 4 | To draw the behavioral view diagram: Sequence diagram, Collaboration diagram | Process Model, DFD, Requirement Engineering | CO2 |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |
| 11 | | | |

* Students are expected to be ready with the prerequisite before attending the lab

Experiment No.04

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 AIM: - AIM: - To draw the behavioral view diagram: Sequence diagram, Collaboration diagram

A.2 Prerequisite

Determine the desired flow of action and their interaction with each other

A.3 Outcome

After successful completion of this experiment students will be able to -

1. Better understanding of the interaction diagrams.
2. Get familiar with sequence & collaboration diagrams.
3. Practice drawing the interaction diagrams using StarUML

A.4 Theory

Interaction diagrams describe how groups of objects collaborate in some behavior. An interaction diagram typically captures the behavior of a single use case. Interaction diagrams do not capture the complete behavior, only typical scenarios.

Diagram is used to describe some type of interactions among the different elements in the model. Interaction is part of the **dynamic behavior** of the system – snapshot of running system at a particular moment. Sequence diagram emphasizes on time sequence of messages collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

For sequence diagram things to be identified:

- Objects taking part in the interaction – three types of objects – Entity, Control, Boundary objects
- Message flow among objects
- The sequence in which messages are flowing
- Object organization

Sequence Diagram -

Sequence diagrams are a graphical way to illustrate a scenario:

- They are called sequence diagrams because they show the sequence of message passing between objects.
- Another big advantage of these diagrams is that they show when the objects are created and when they are destructed. They also show whether messages are synchronous or asynchronous

Collaboration Diagram -

They are the same as sequence diagrams but without a time axis:

- Their message arrows are numbered to show the sequence of message sending.
- They are less complex and less descriptive than sequence diagrams.
- These diagrams are very useful during design because you can figure out how objects communicate with each other.

A.5 Procedure/Algorithm

A.5.1 Task:

Draw a sequence diagram for the case study.

- Identify objects
- Identify messages between objects.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

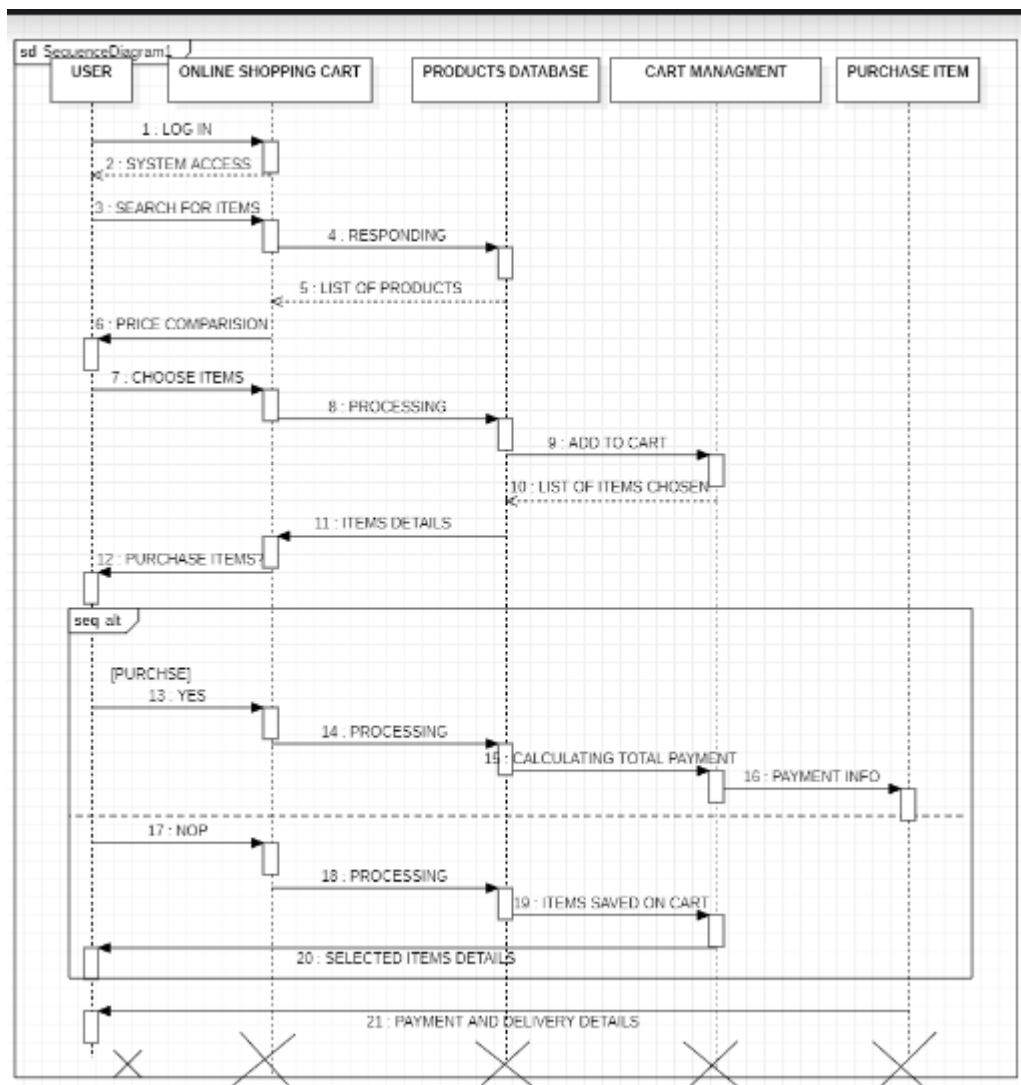
(Students must submit the soft copy as per the following segments within two hours of the practicals. The soft copy must be uploaded on Blackboard LMS or emailed to the concerned Lab in charge Faculties at the end of practical; in case Blackboard is not accessible)

| | |
|---------------------------|----------------------------|
| Roll No: I045 | Name: Nihhaal Raichura |
| Class: BTECH AI | Batch: B3 |
| Date of Experiment:7-2-23 | Date of Submission:13-2-23 |
| Grade: | |

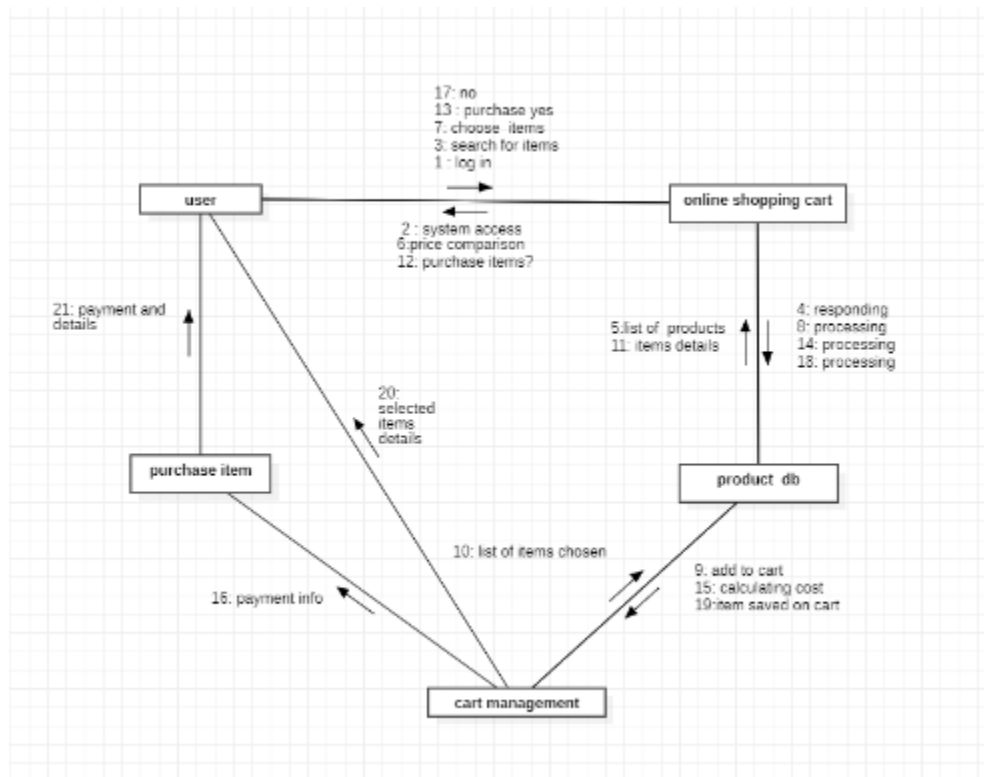
B.1 Objects:

User, Online Shopping Cart, Products Database, cart Management, Purchase Item

B.2 Sequence diagram:



B.3 Collaboration diagram:



B.4 Conclusion

We learnt about sequence and collaboration diagrams, even made the sequence and collaboration diagrams for an online shopping website.

B.5 Questions of Curiosity:

Q1. State the difference between entity, boundary and control objects.

An entity object represents a real-world object or concept and contains data that describes the object. For example, an entity object might represent an employee and contain data such as the employee's name, address, and salary. Entity objects typically have behaviours (methods) that operate on their data.

A boundary object is an object that acts as an interface between the system and the user. It presents information to the user and receives inputs from the user. Boundary objects are responsible for converting user inputs into a form that can be processed by the system, and for presenting the results of processing to the user.

A control object is an object that manages the flow of data and the coordination of the behaviour of other objects in the system. Control objects are responsible for making decisions about what actions to take based on the current state of the system and the inputs received from boundary objects. They also manage the flow of control from one object to another, and enforce any rules or constraints that apply to the system as a whole.

In summary, entity objects represent real-world objects and contain data, boundary objects provide an interface between the system and the user, and control objects manage the flow of data and coordinate the behaviour of other objects.

Q.2 State the difference between sequence and collaboration diagram.

An entity object represents a real-world object or concept and contains data that describes the object. For example, an entity object might represent an employee and contain data such as the employee's name, address, and salary. Entity objects typically have behaviours (methods) that operate on their data.

A boundary object is an object that acts as an interface between the system and the user. It presents information to the user and receives inputs from the user. Boundary objects are responsible for converting user inputs into a form that can be processed by the system, and for presenting the results of processing to the user.

A control object is an object that manages the flow of data and the coordination of the behaviour of other objects in the system. Control objects are responsible for making decisions about what actions to take based on the current state of the system and the inputs received from boundary objects. They also manage the flow of control from one object to another, and enforce any rules or constraints that apply to the system as a whole.

In summary, entity objects represent real-world objects and contain data, boundary objects provide an interface between the system and the user, and control objects manage the flow of data and coordinate the behaviour of other objects.

Q.3 When looping is required in sequence diagram?

An entity object represents a real-world object or concept and contains data that describes the object. For example, an entity object might represent an employee and contain data such as the employee's name, address, and salary. Entity objects typically have behaviours (methods) that operate on their data.

A boundary object is an object that acts as an interface between the system and the user. It presents information to the user and receives inputs from the user. Boundary objects are responsible for converting user inputs into a form that can be processed by the system, and for presenting the results of processing to the user.

A control object is an object that manages the flow of data and the coordination of the behavior of other objects in the system. Control objects are responsible for making decisions about what actions to take based on the current state of the system and the inputs received from boundary

objects. They also manage the flow of control from one object to another, and enforce any rules or constraints that apply to the system as a whole.

In summary, entity objects represent real-world objects and contain data, boundary objects provide an interface between the system and the user, and control objects manage the flow of data and coordinate the behavior of other objects.

