

Terna Engineering College
Computer Engineering Department
Program: Sem V

Course: Software Engineering

LAB Manual

PART A

Experiment No.05

A.1 Aim:

Develop Sequence and Collaboration diagram for the selected mini project.

A.2 Prerequisite:

1. Requirement Modelling

A.3 Outcome:

After successful completion of this experiment students will be able to

- ✓ Able to model requirements using UML

A.4 Theory:

Interaction Diagrams

Show how objects interact with one another, it shows an interaction of set of objects and their relationships, including the messages that they exchanged among them. These set of messages are used by the objects to communicate with each other.

UML supports two types of interaction diagrams

- ✓ Sequence diagrams
- ✓ Collaboration diagrams

Sequence Diagram

Sequence diagram shows an interaction arranged in a time sequence. It is an alternate way to understand the overall flow of the control of the system program.

A UML Sequence Diagram consists of :

- Objects
- Object life line
- Focus of Control
- Messages send by the objects to communicate with each other

Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass. Sequence diagrams show the sequence of how events occur. The horizontal dimension shows the objects participating in the interaction. The vertical arrangement of messages indicates their order.

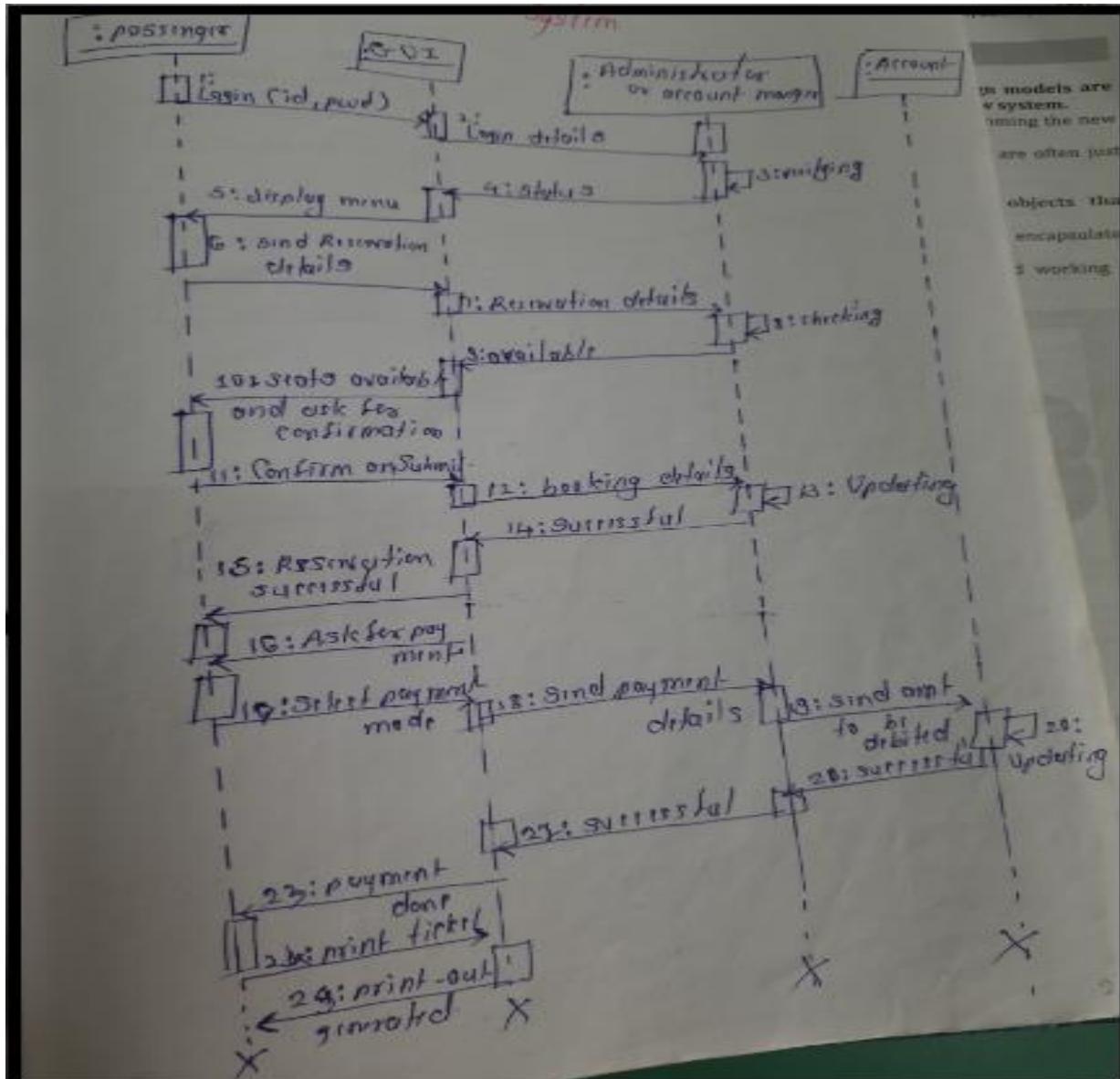
Collaboration Diagram

Collaboration diagrams are equivalent to sequence diagrams. All the features of sequence diagrams are equally applicable to collaboration diagrams.

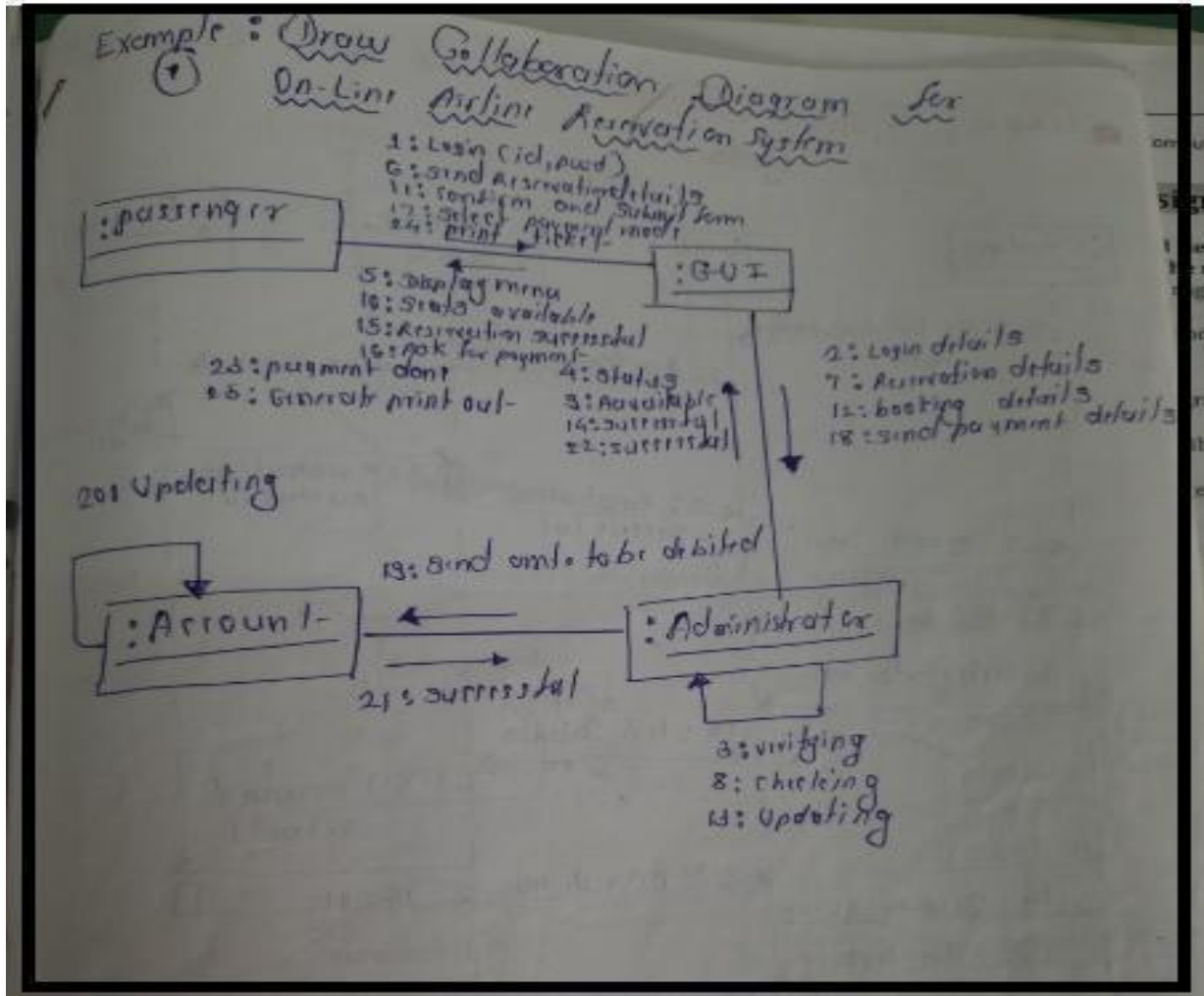
Use a sequence diagram when the transfer of information is the focus of attention.

Use a collaboration diagram when concentrating on the classes.

Example: Sequence Diagram for Online Airline Reservation system



Example: Collaboration Diagram for Online Airline Reservation system



PART B

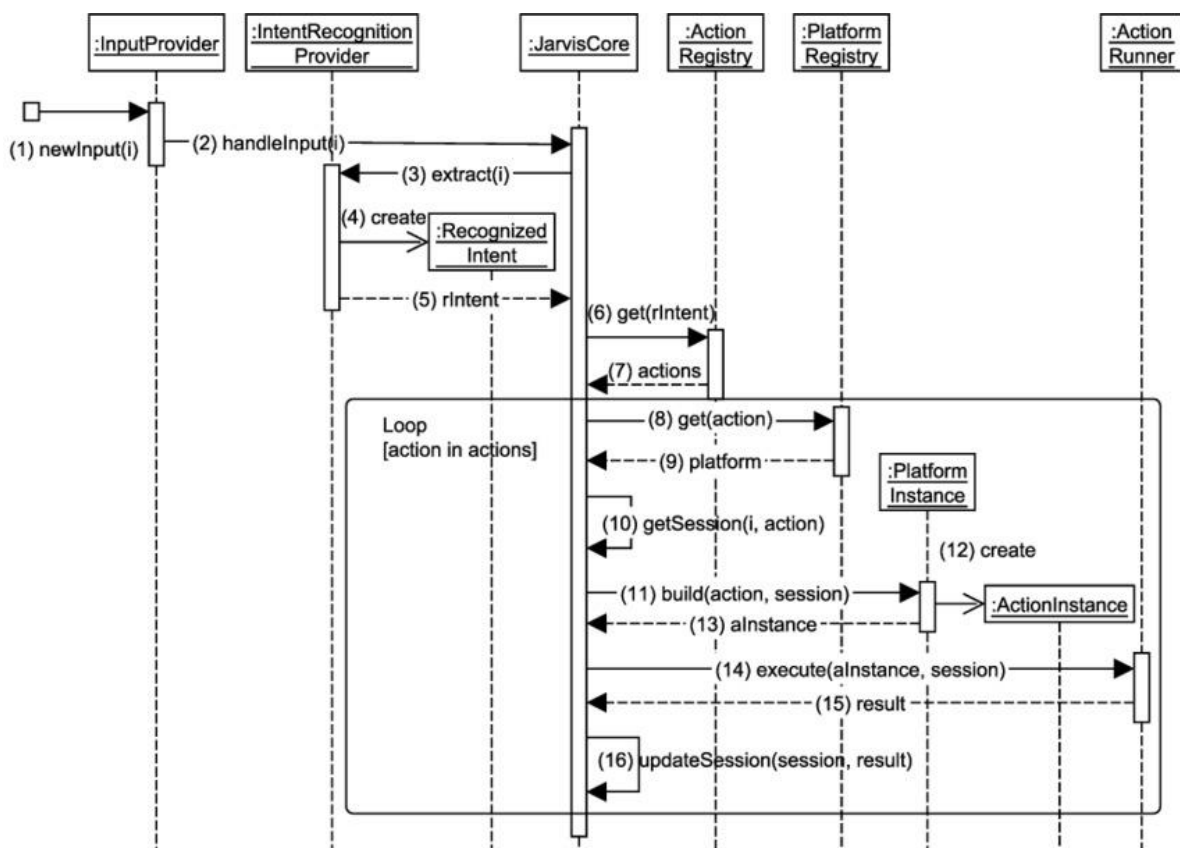
(PART B : TO BE COMPLETED BY STUDENTS)

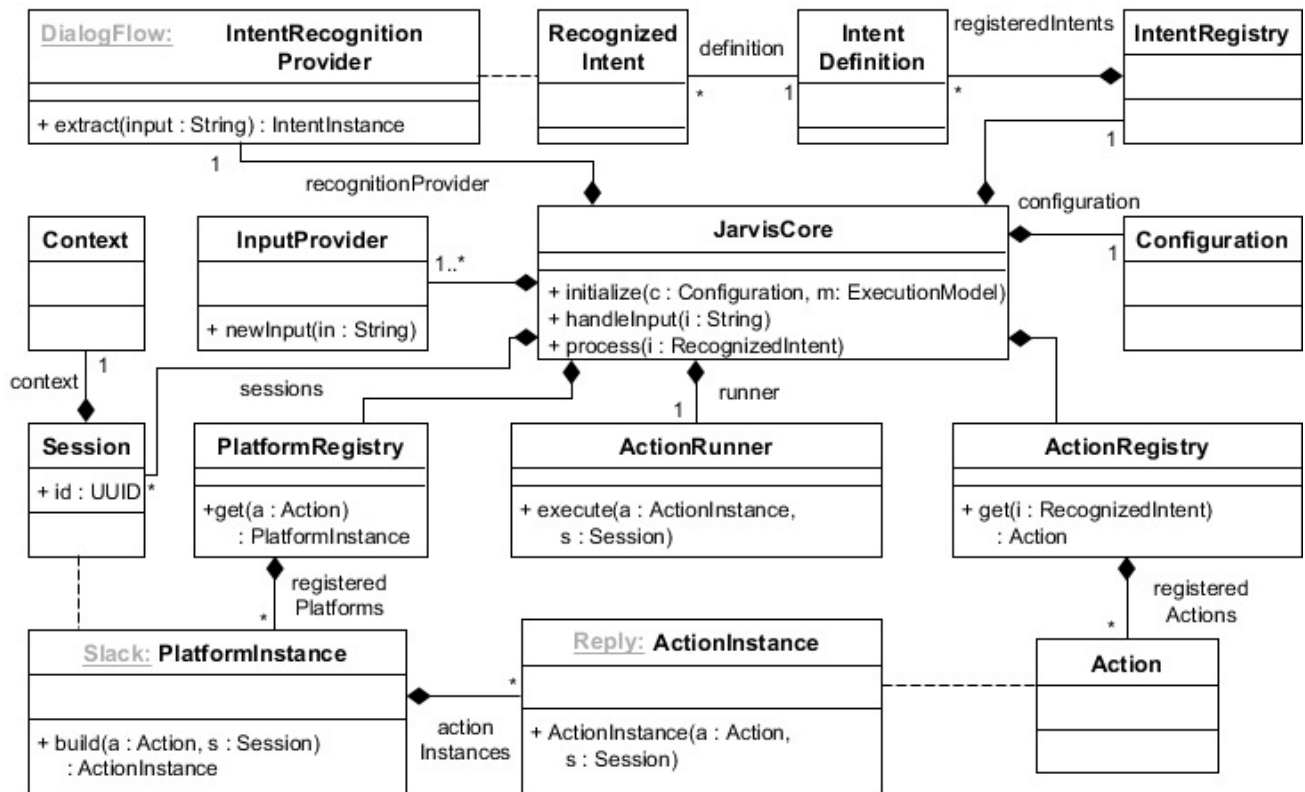
(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll No. B47	Name: Manas Narendra Telavane
Class : B3	Batch : B3 - G1
Date of Experiment: 25 August 2022	Date of Submission : 22 September 2022
Grade :	

B.1 Draw Sequence and Collaboration Diagram for selected mini project

Sequence Diagram:

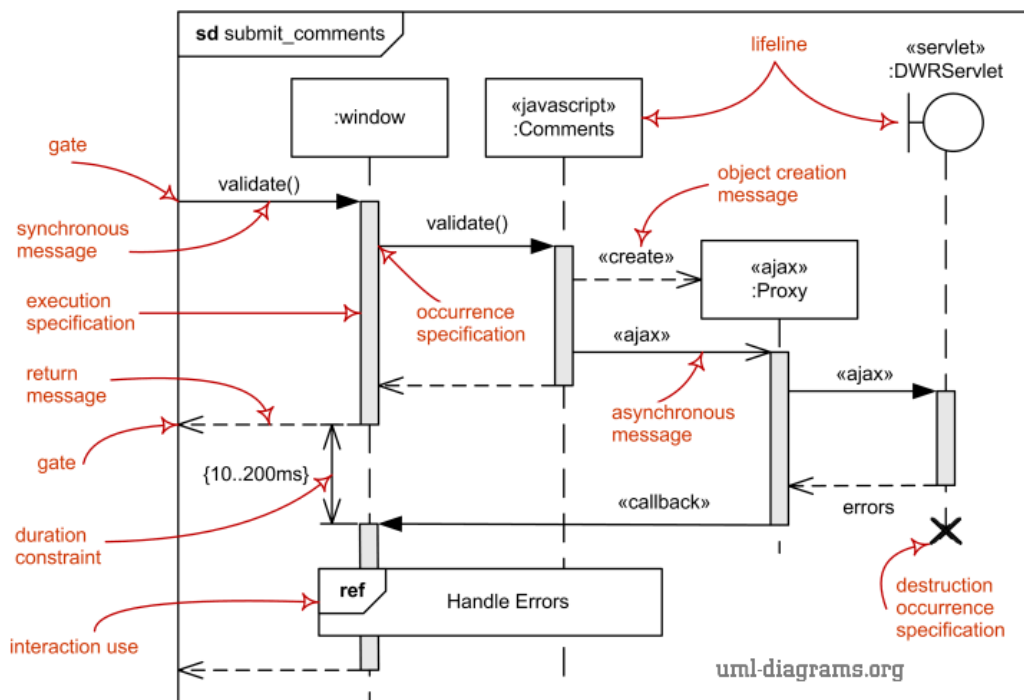


Collaboration Diagram:**B.2 Conclusion:**

- Sequence and Collaboration diagrams are both important steps in the project planning process and help iron out the inner architecture of the project.
- This experiment has shown that a UML sequence diagram is more effective than a UML collaboration diagram in representing the steps in creating a software application, if the architecture is more of a call-and-return based architecture.
- The reason for this is that the sequence diagram provides more detail about each step and is easier to understand than the collaboration diagram.
- However, if the architecture is more of an object oriented type model, then a collaboration diagram is much easier to understand and is useful to explain the working of the project.
- This experiment also showed that it is possible to use both types of diagrams together in order to represent an application's development process.

B.3 Question of Curiosity :

Q1: Explain various elements of the sequence diagram.



- **Lifeline**
 - Lifeline is a named element which represents an individual participant in the interaction. While parts and structural features may have multiplicity greater than 1, lifelines represent only one interacting entity.
 - A lifeline is shown using a symbol that consists of a rectangle forming its "head" followed by a vertical line (which may be dashed) that represents the lifetime of the participant.
- **Gate**
 - A gate is a message end, connection point for relating a message outside of an interaction fragment with a message inside the interaction fragment.
 - The purpose of gates and messages between gates is to specify the concrete sender and receiver for every message. Gates play different roles:
 - ◆ formal gates - on interactions
 - ◆ actual gates - on interaction uses

- ◆ expression gates - on combined fragment
- The gates are named implicitly or explicitly. Implicit gate name is constructed by concatenating the direction of the message ("in" or "out") and the message name, e.g. in_search, out_read.
- Gates are notated just as message connection points on the frame.
- Interaction Fragment
 - Interaction fragment is a named element representing the most general interaction unit. Each interaction fragment is conceptually like an interaction by itself.
 - There is no general notation for an interaction fragment. Its subclasses define their own notation.
 - Examples of interaction fragments are:
 - ◆ occurrence
 - ◆ execution
 - ◆ state invariant
 - ◆ combined fragment
 - ◆ interaction use
- Occurrence:
 - Occurrence (complete UML name - occurrence specification, i.e. "event description") is interaction fragment which represents a moment in time (event) at the beginning or end of a message or at the beginning or end of an execution.
 - An occurrence specification is one of the basic semantic units of interactions. The meanings of interactions are specified by sequences of occurrences described by occurrence specifications.
 - Each occurrence specification appears on exactly one lifeline. Occurrence specifications of a lifeline are ordered along the lifeline.
 - Occurrence specification has no notation and is just a point at the beginning or end of a message or at the beginning or end of an execution specification.
 - Examples of occurrences are:
 - ◆ message occurrence
 - ◆ execution occurrence
- Execution
 - Execution (full name - execution specification, informally called activation) is interaction fragment which represents a period in the participant's lifetime when it is
 - ◆ executing a unit of behavior or action within the lifeline,

- ◆ sending a signal to another participant,
- ◆ waiting for a reply message from another participant.
- Note, that the execution specification includes the cases when behavior is not active, but just waiting for reply. The duration of an execution is represented by two execution occurrences - the start occurrence and the finish occurrence.
- Execution is represented as a thin grey or white rectangle on the lifeline

Q2: Explain various elements of the Collaboration diagram.

- Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.
- Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.
- Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.
- Messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

END