

# **Assignment 6**

## **Prepare and Implement Sequence Model**

### **Aim:**

Identify at least 5 major scenarios (sequence flow) for your system. Draw Sequence Diagram for every scenario by using advanced notations using UML2.0. Implement these scenarios by taking reference of design model implementation using suitable object-oriented language.

### **Problem Statement:**

- Prepare Sequence Model
- Identify at least 5 major scenarios (sequence flow) for DEFINITION your system. Draw Sequence Diagram for every scenario by using advanced notations using UML2.0.
- Implement these scenarios by taking reference of design model implementation using a suitable object-oriented language.

### **Objective:**

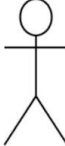




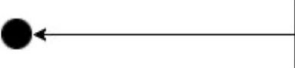

- To study and use communication.
- Draw a sequence diagram.
- To implement a sequence diagram.

### **1. Relevant Theory:**

#### **1.1. Sequence Diagram**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

#### **1.2. Sequence Diagram Notations**

Message Type	Description	Symbol
Actors	An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.	
Lifelines	A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format – Instance Name : Class Name	
Synchronous messages	Synchronous messages wait for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message. The caller continues only when it knows that the receiver has processed the previous message i.e. it receives a reply message.	
Asynchronous Messages	An asynchronous message does not wait for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not. We use a lined arrow head to represent an asynchronous message.	
Reply Message	Reply messages are used to show the message being sent from the receiver to the sender. We represent a return/reply message using an open arrowhead with a dotted line. The interaction moves forward only when a reply message is sent by the receiver.	
Lost Message	A Lost message is used to represent a scenario where the recipient is not known to the system. It is represented using an arrow directed towards an end point from a lifeline. For example: Consider a scenario where a warning is generated.	
Found Message	A Found message is used to represent a scenario where an unknown source sends the message. It is represented using an arrow directed towards a lifeline from an endpoint. For example: Consider the scenario of a hardware failure.	

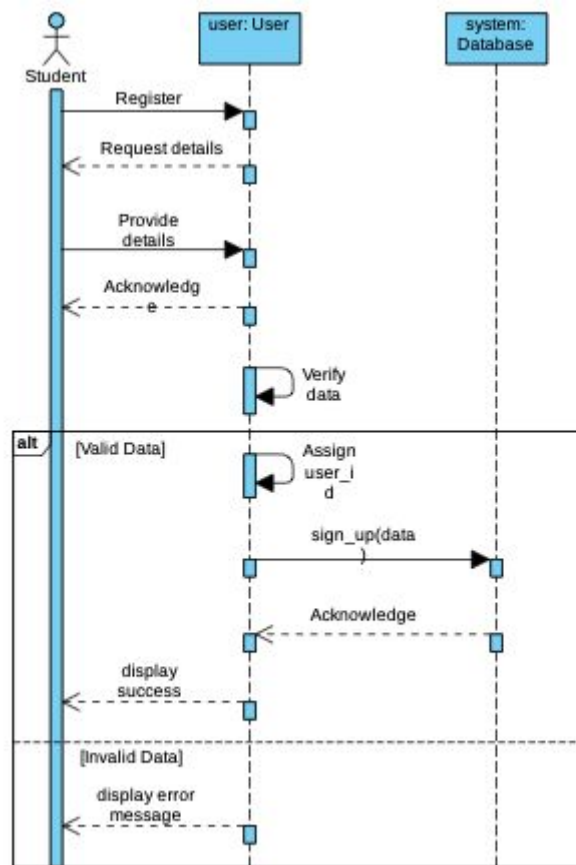


Fig 1. Sign Up

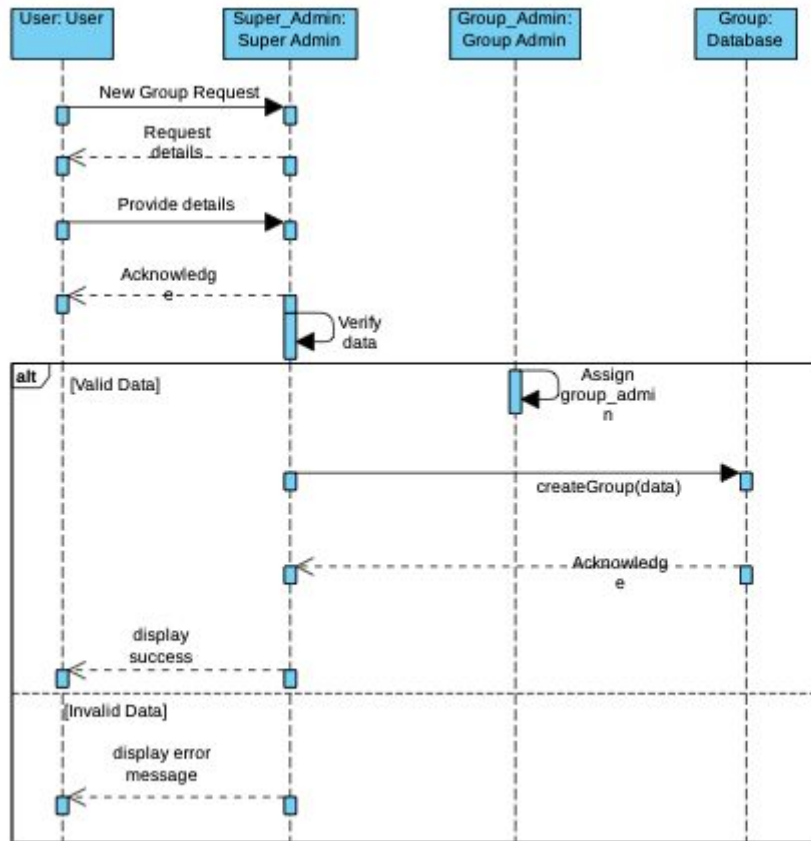


Fig 2. New Group Request

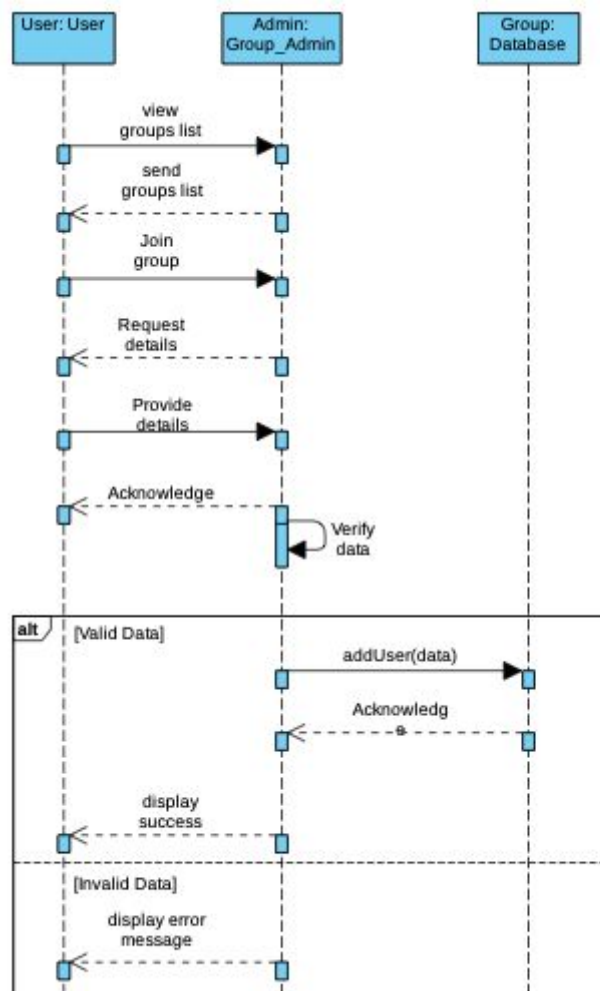


Fig 3. Join Existing Group

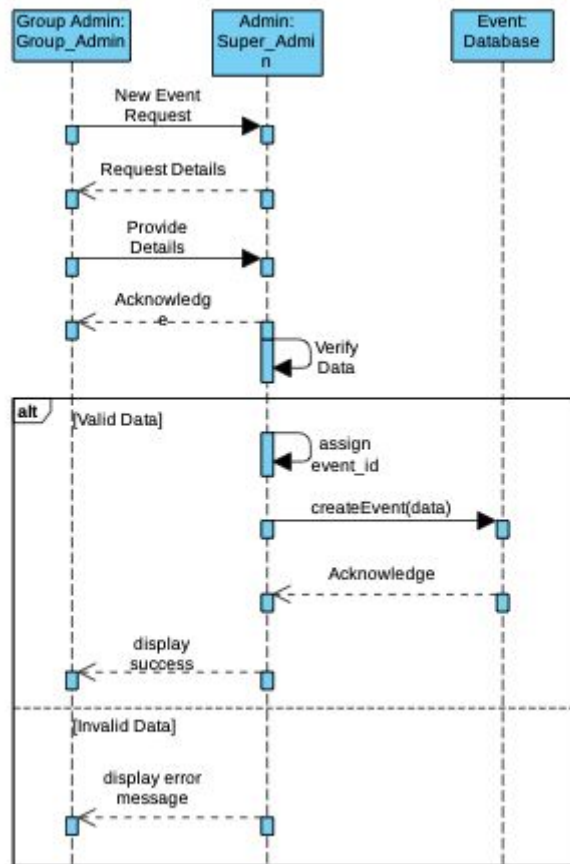


Fig 4. New Event Request

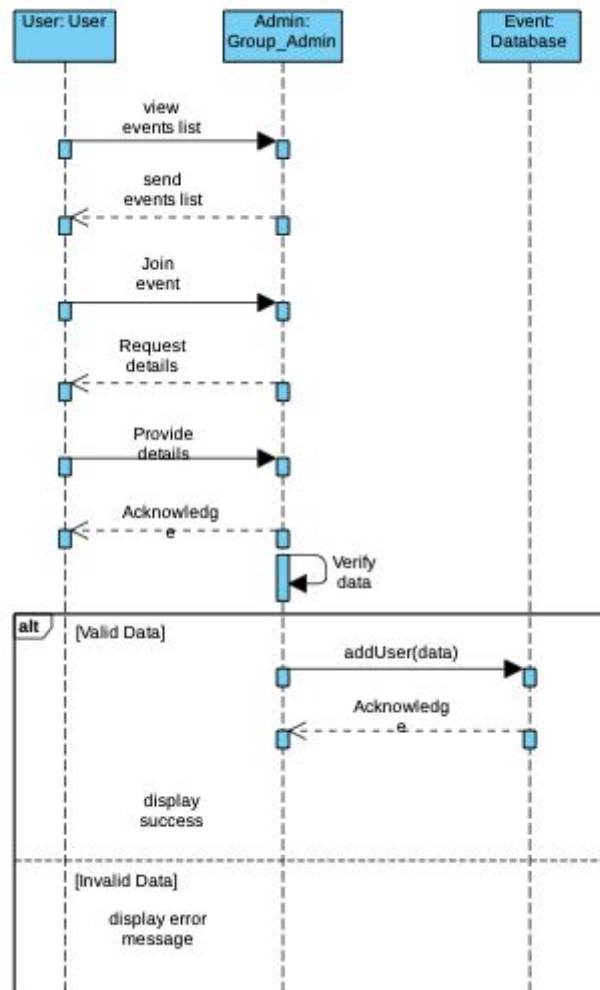


Fig 5. Be a part of an event

## 2. Implementation:

### 2.1 Sign Up

- i. As shown in Fig.1, users will have to sign up/ register.
- ii. The data will get validated by the user object
- iii. If the data provided is valid, then the user is provided with a valid user\_id and a success message is returned.
- iv. If the data provided is not valid, then an error message is returned.

### 2.2 New Group Request

- i. The new group request is validated by the super admin. The user who wants to create a new group, provides the prerequisite details to the super admin.

- ii. The super admin validates this data, and accepts or rejects the request.
- iii. If the request is accepted, then the user is elevated to the status of group admin, and a valid group\_id is given to the group.
- iv. In case the request is not accepted, then an error message is displayed.

### **2.3 Join Existing Group**

- i. If a user wants to join an existing group, then this request needs to be validated by the group\_admin.
- ii. The user provides the necessary details. These details are validated by the group\_admin.
- iii. Suppose the details provided are valid, then the user is added to the group.
- iv. In case the details are not valid, then an error message is displayed.

### **2.4 New Event Request**

- i. The new event request is validated by the super admin. The group\_admin who wants to create a new event, provides the prerequisite details to the super admin.
- ii. The super admin validates this data, and accepts or rejects the request.
- iii. If the request is accepted, then a success message is displayed.
- iv. In case the request is not accepted, then an error message is displayed.

### **2.5 Join Existing Event**

- i. If a user wants to join an existing event, then this request needs to be validated by the group\_admin.
- ii. The user provides the necessary details. These details are validated by the group\_admin.
- iii. Suppose the details provided are valid, then the user is added to the group.
- iv. In case the details are not valid, then an error message is displayed.



**Conclusion:**

Thus in this Assignment we have successfully Identified and implemented sequence diagrams for 5 major scenarios of our system.