

# EXPERIMENT - 2

**AIM:** To know about use-case and sequence diagrams.

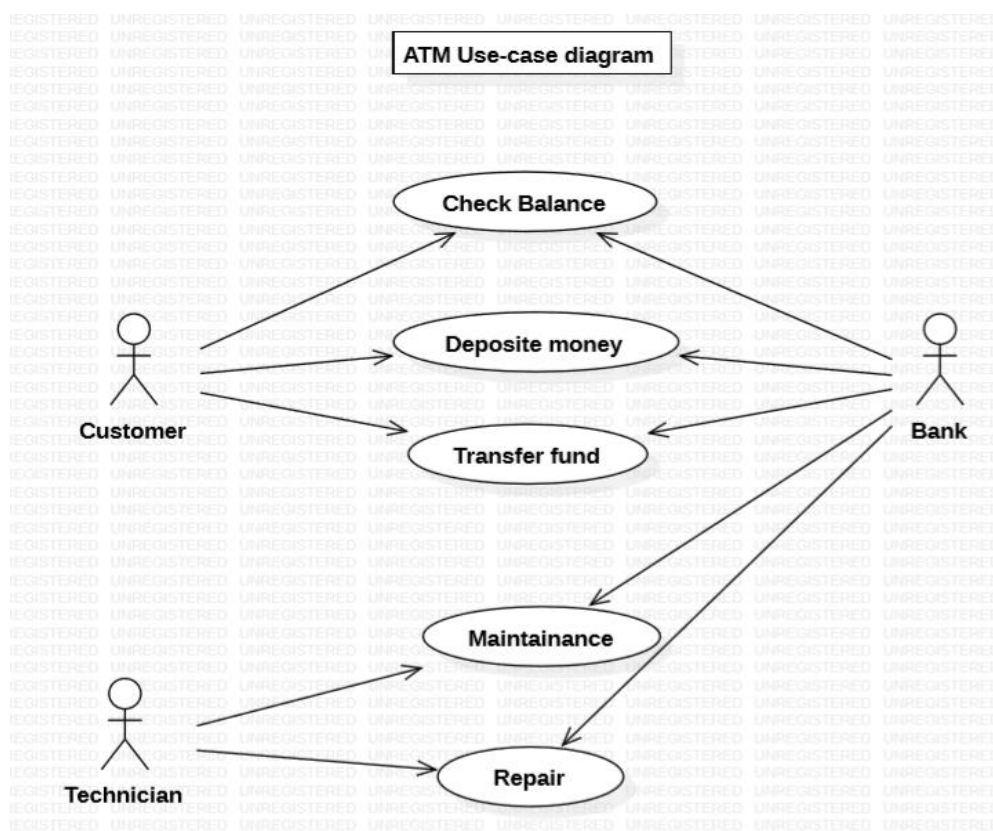
## 1. USE-CASE DIAGRAM

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as use case
- Actors
- Relationships among the use cases and actors.

**Use-case Diagram of ATM:**



## 2. SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

### Sequence Diagram Notations –

1. **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.
2. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram.
3. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

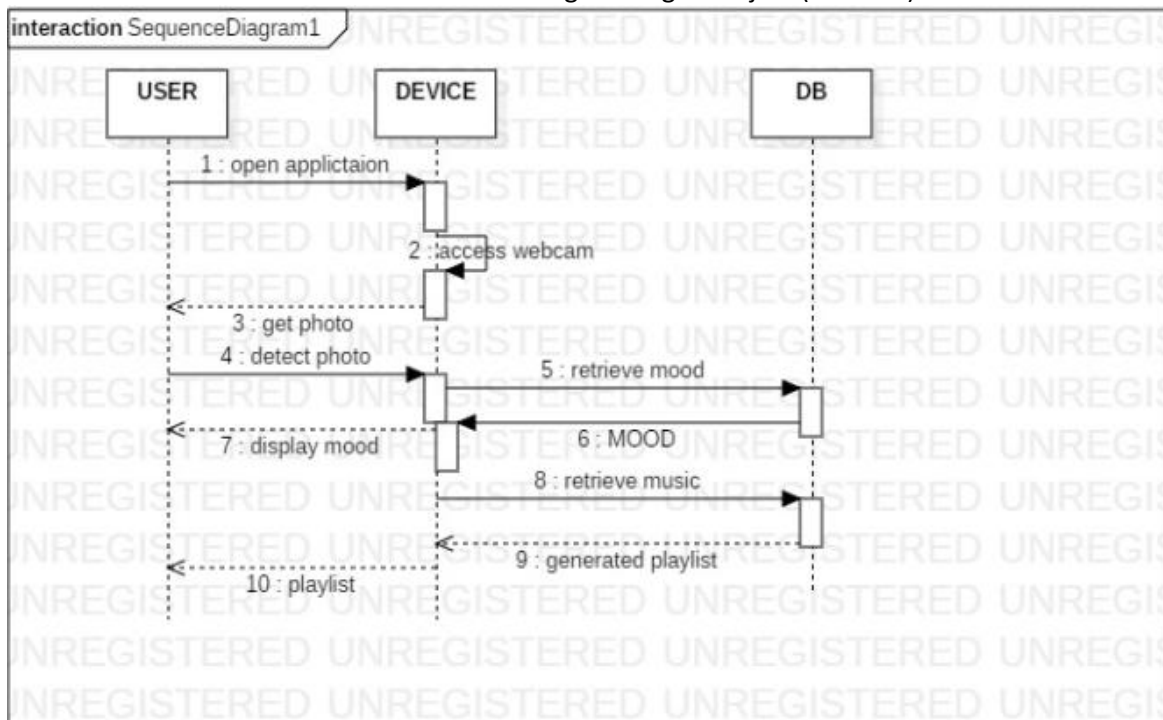
Messages can be broadly classified into the following **categories**:

- **Synchronous messages** – A synchronous message waits for a reply before the interaction can move forward.
  - **Asynchronous Messages** – An asynchronous message does not wait for a reply from the receiver.
  - **Create message** – We use a Create message to instantiate a new object in the sequence diagram.
  - **Delete Message** – We use a Delete Message to delete an object.
  - **Self Message** – Certain scenarios might arise where the object needs to send a message to itself.
  - **Reply Message** – Reply messages are used to show the message being sent from the receiver to the sender.
  - **Found Message** – A Found message is used to represent a scenario where an unknown source sends the message.
  - **Lost Message** – A Lost message is used to represent a scenario where the recipient is not known to the system.
4. **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

### Uses of sequence diagrams –

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.

### Sequence Diagram of user device and database



### Sequence Diagram of validate user and database

