


- **Systems Analysis and Design** **Object-Oriented Systems**
 - **Course Code: CSE 305** **Analysis and Design Using UML**
(Sequence Diagram)
- 

Instructor:
Dr. Shah Murtaza Rashid Al Masud
Assoc. Prof. CSE Dept., UAP

Sequence Diagrams

Sequence diagram is one of the best ways to identify the requirements and processes of a system.

This is an interaction diagram that illustrates how objects in a system communicate with each other.

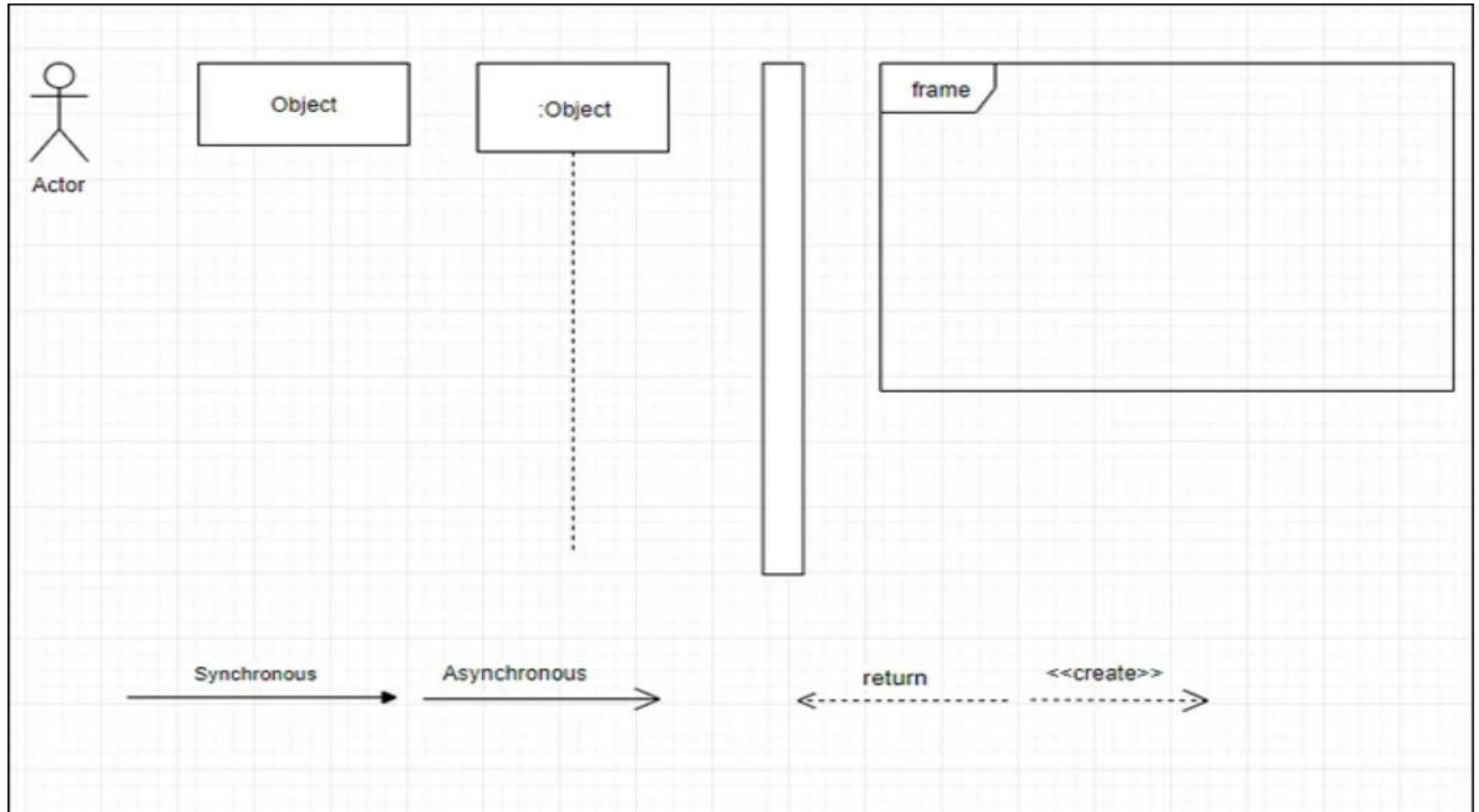
Each object passes on some messages to another in a sequential time or order of steps in a process.

Sequence Diagrams

A sequence diagram is composed of **elements and icons** which you need to be familiar with in order to create one.

There are **basic symbols, components, and types of message arrows** that may occur in a UML sequence diagram example.

Sequence Diagrams Notation



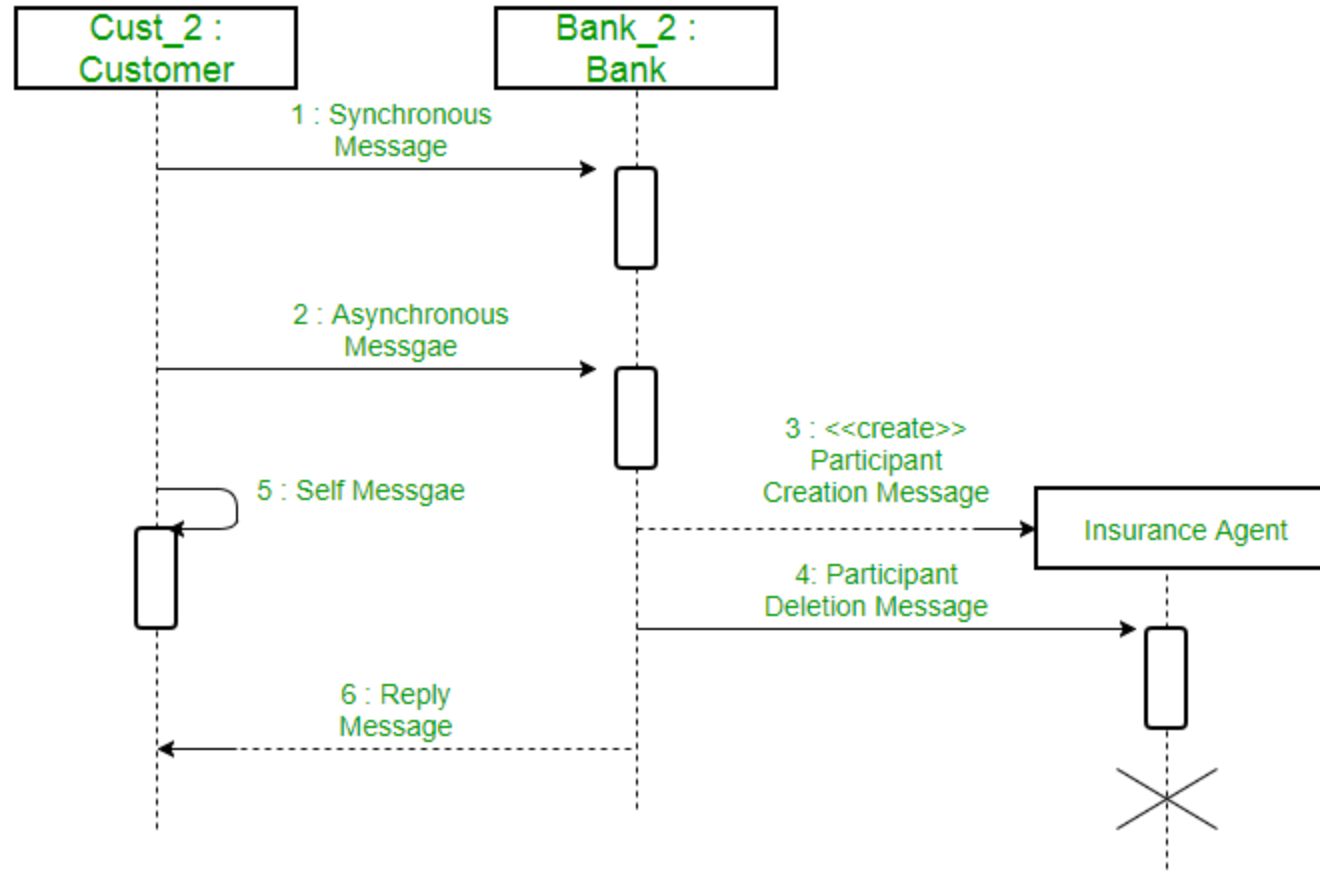
Sequence Diagrams Notation

- **Actor** – An external entity that interact with the system.
- **Object** – Refers to the symbol in a system that describes how a class role will behave.
- **Activation Bar** – A rectangle shape placed on a lifeline that indicates the time or duration an object needs to finish a task.
- **Lifeline** – Is a vertical dashed line that represents the presence of an object over time.
- **Option loop** – Rectangle symbol that represents a repetitive sequence in a system sequence diagram example. It executes a task when a certain condition is met.
- **Synchronous** – Represents a message where the sender waits for the response of the receiver before the message continues. This illustrates a line with a solid arrowhead.
- **Asynchronous** – A type of message that does not need a response from the receiver before the message gets back and carries another message. This is usually drawn with a solid line and an open arrowhead.
- **Return Message** – It represents a symbol of a dashed line with an open arrowhead in response to the calls from the original lifeline.
- **Create Message** – Used to create an object in a sequence diagram example that is usually drawn using a dashed line and an open arrowhead pointing to the object created.

Sequence Diagrams Notation

Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows.

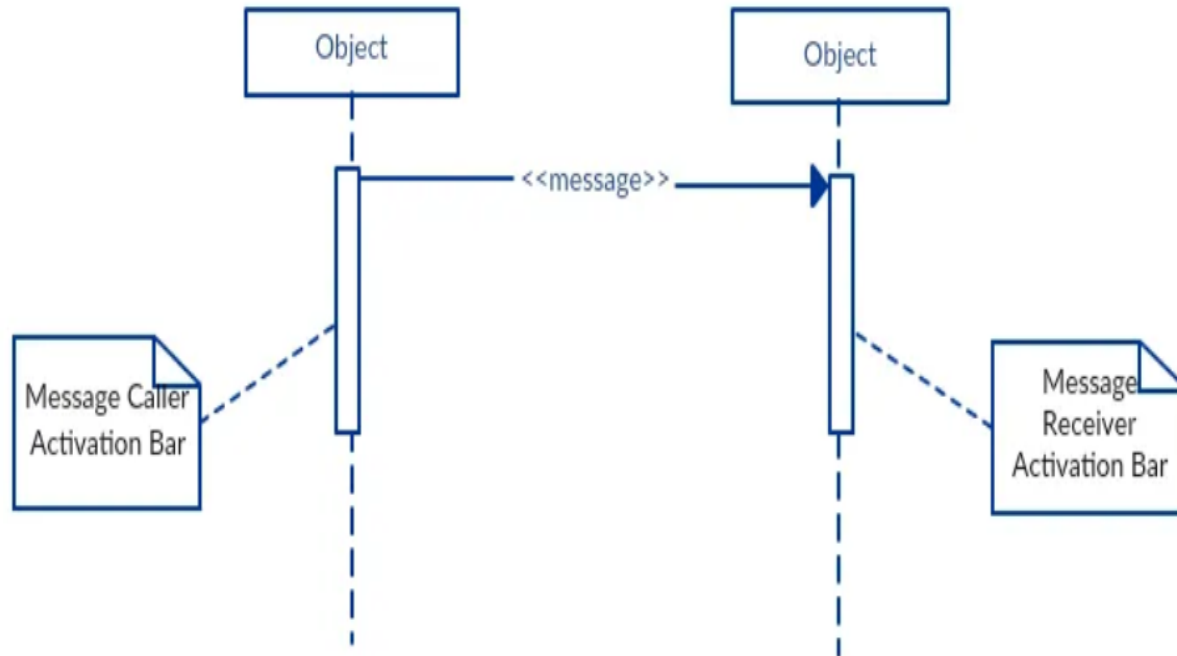
Lifelines and messages form the core of a sequence diagram.



Sequence Diagrams Notation

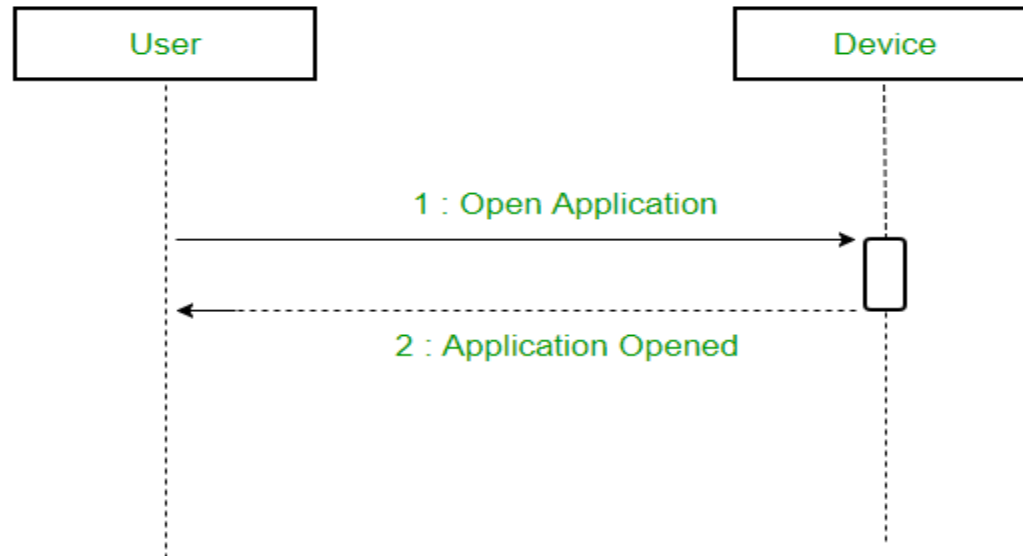
Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows.

Lifelines and messages form the core of a sequence diagram.



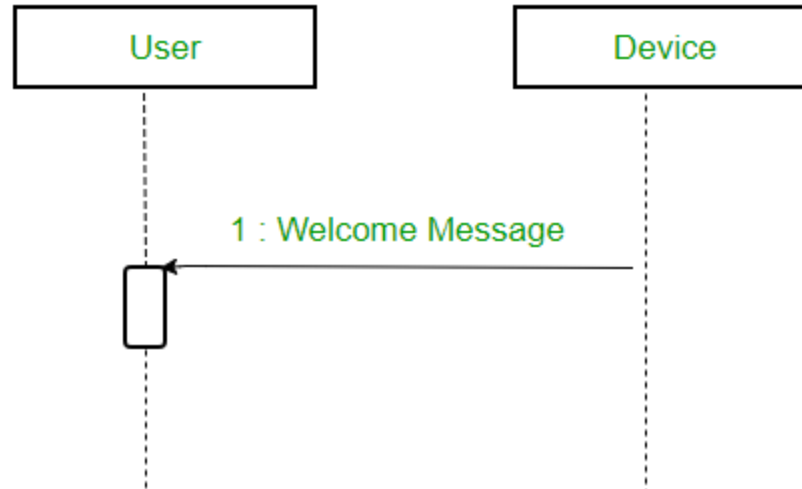
Sequence Diagrams Notation

Synchronous messages – A synchronous message waits 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. A large number of calls in object oriented programming are synchronous. We use a solid arrow head to represent a synchronous message.



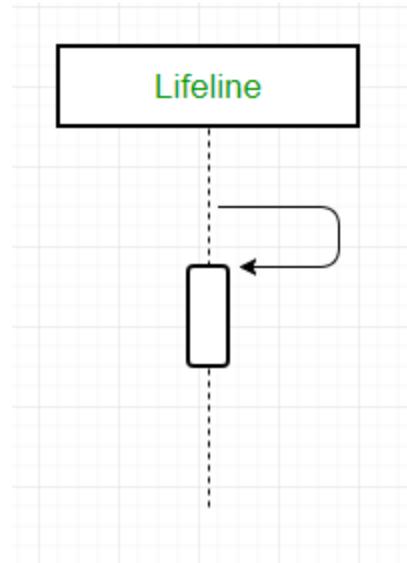
Sequence Diagrams Notation

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.



Sequence Diagrams Notation

Self Message – Certain scenarios might arise where the object needs to send a message to itself. Such messages are called Self Messages and are represented with a U shaped arrow shaped arrow.



Sequence Diagrams Notation

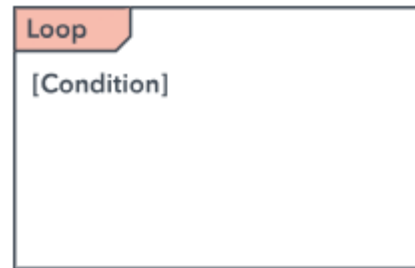
Reply Message – Reply messages are used to show the message being sent from the receiver to the sender. We represent a return/reply message using an open arrowhead with a dotted line. The interaction moves forward only when a reply message is sent by the receiver.



Sequence Diagrams Notation

Option loop symbol: Used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions.

Alternative symbol: Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed



Option loop
symbol



Alternative
symbol

UML Sequence Diagram for User Login Module

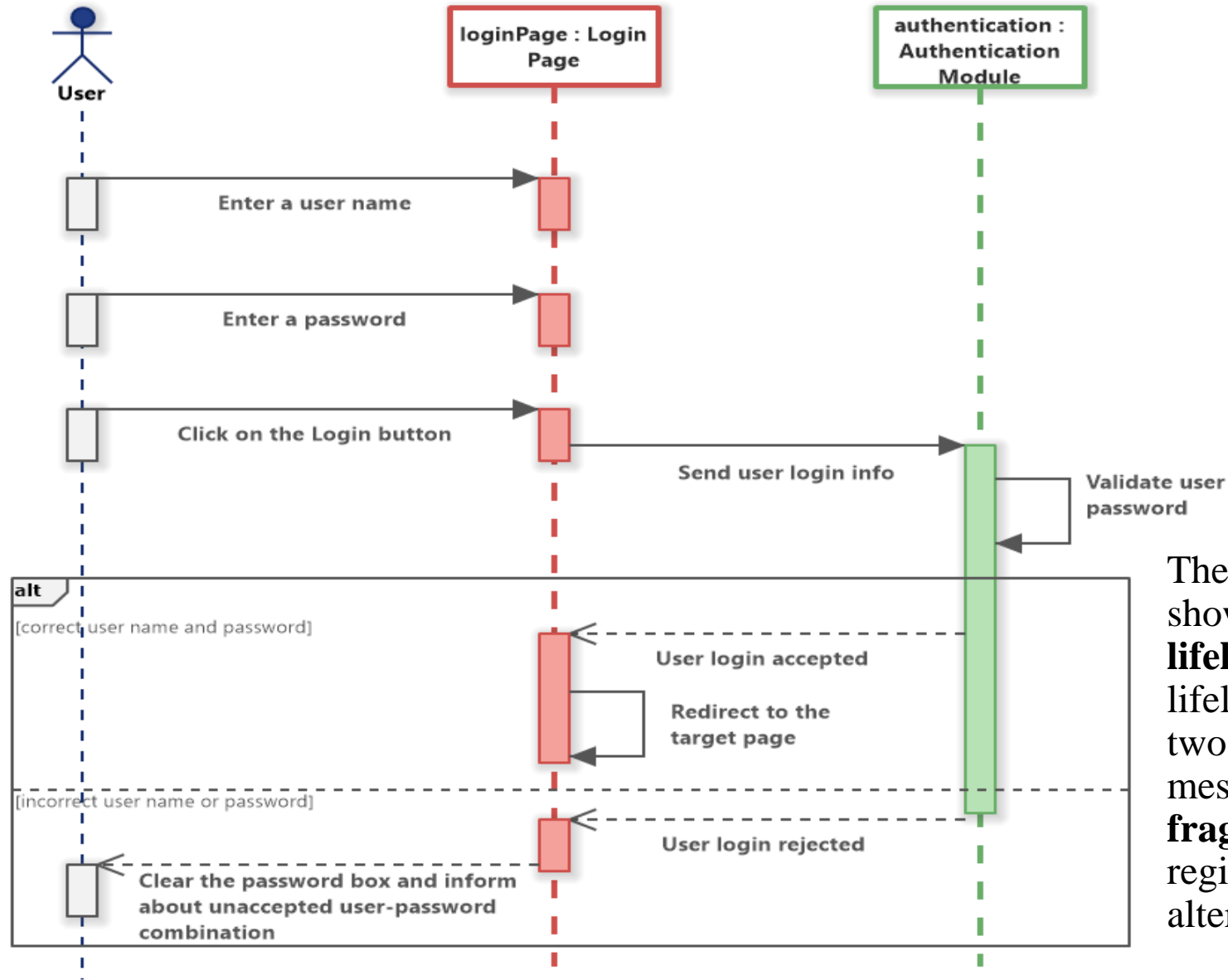
The UML sequence diagram shows these lifelines:

- User
 - Login Page
 - Authentication Module
-

It models the following actions:

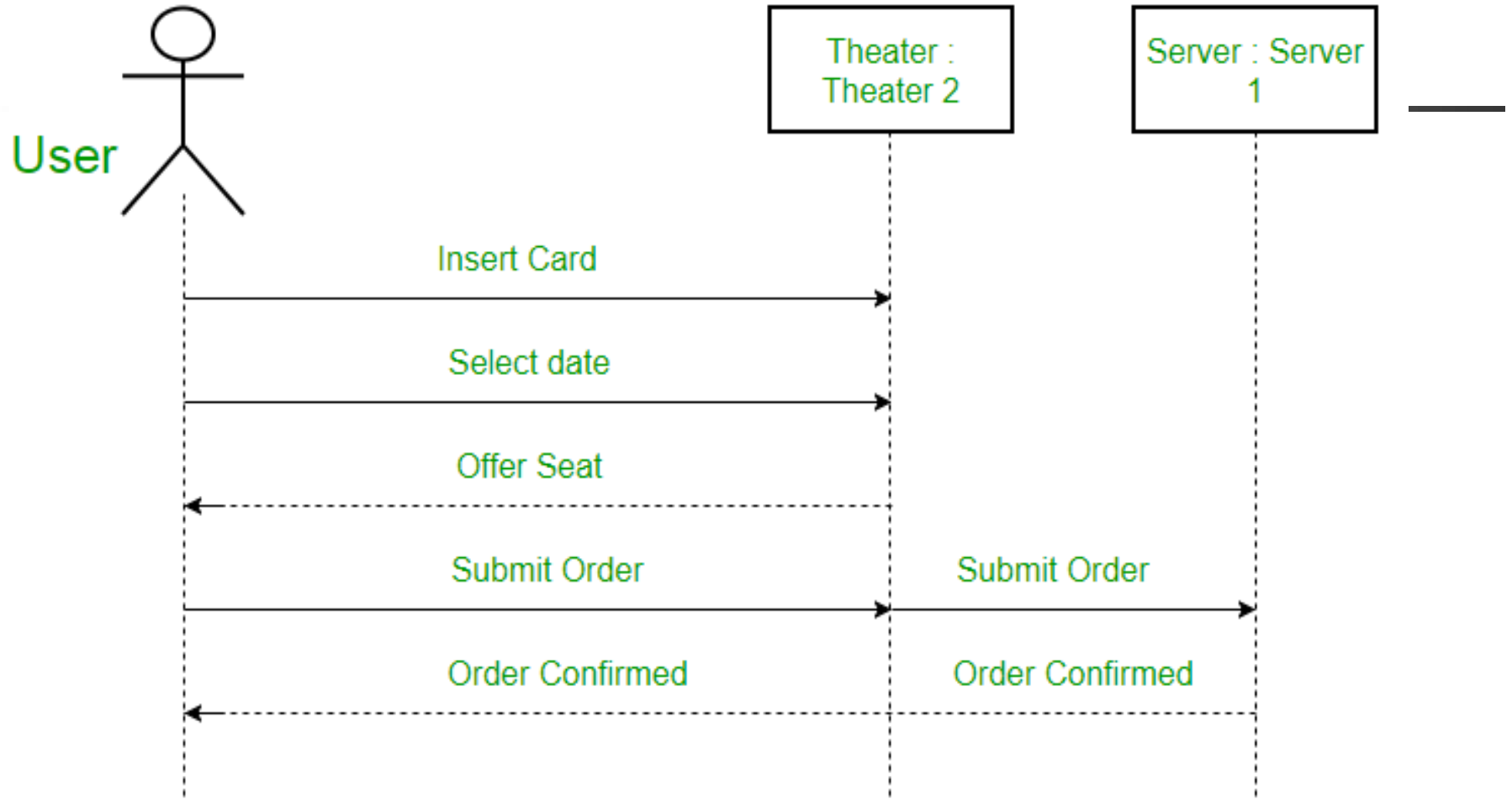
- Enter a user name
- Enter a password
- Click on the Login button
- Send user login info
- Validate user password
- User login accepted
- Redirect to the target page
- User login rejected
- Clear the password box and inform the user about an unaccepted user-password combination

UML Sequence Diagram for User Login Module

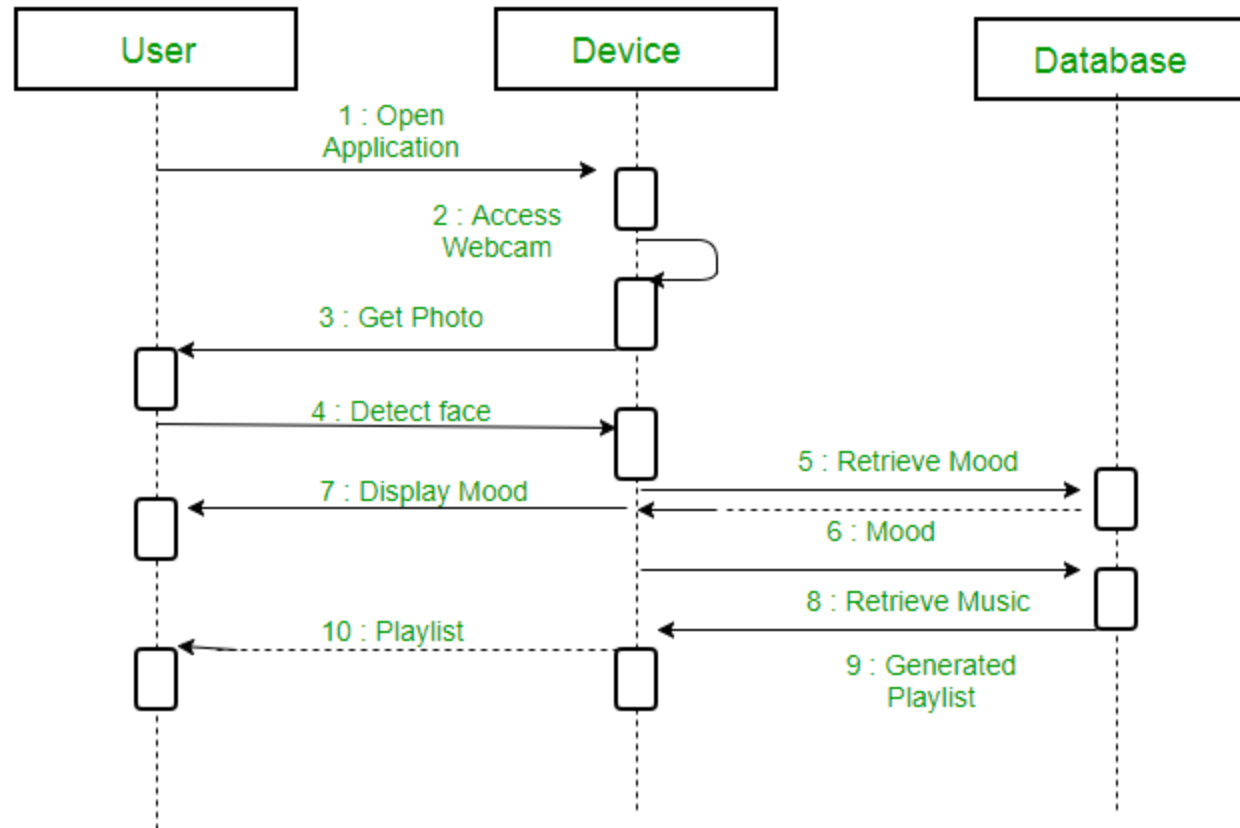


The example diagram shows an **actor lifeline** and two standard lifelines, call messages, two **self messages**, reply messages, and an **alt fragment** with two regions for two alternative options.

An actor interacting with a seat reservation system



A sequence diagram for an emotion based music player –



A sequence diagram for an emotion based music player –

The above sequence diagram depicts the sequence diagram for an emotion based music player:

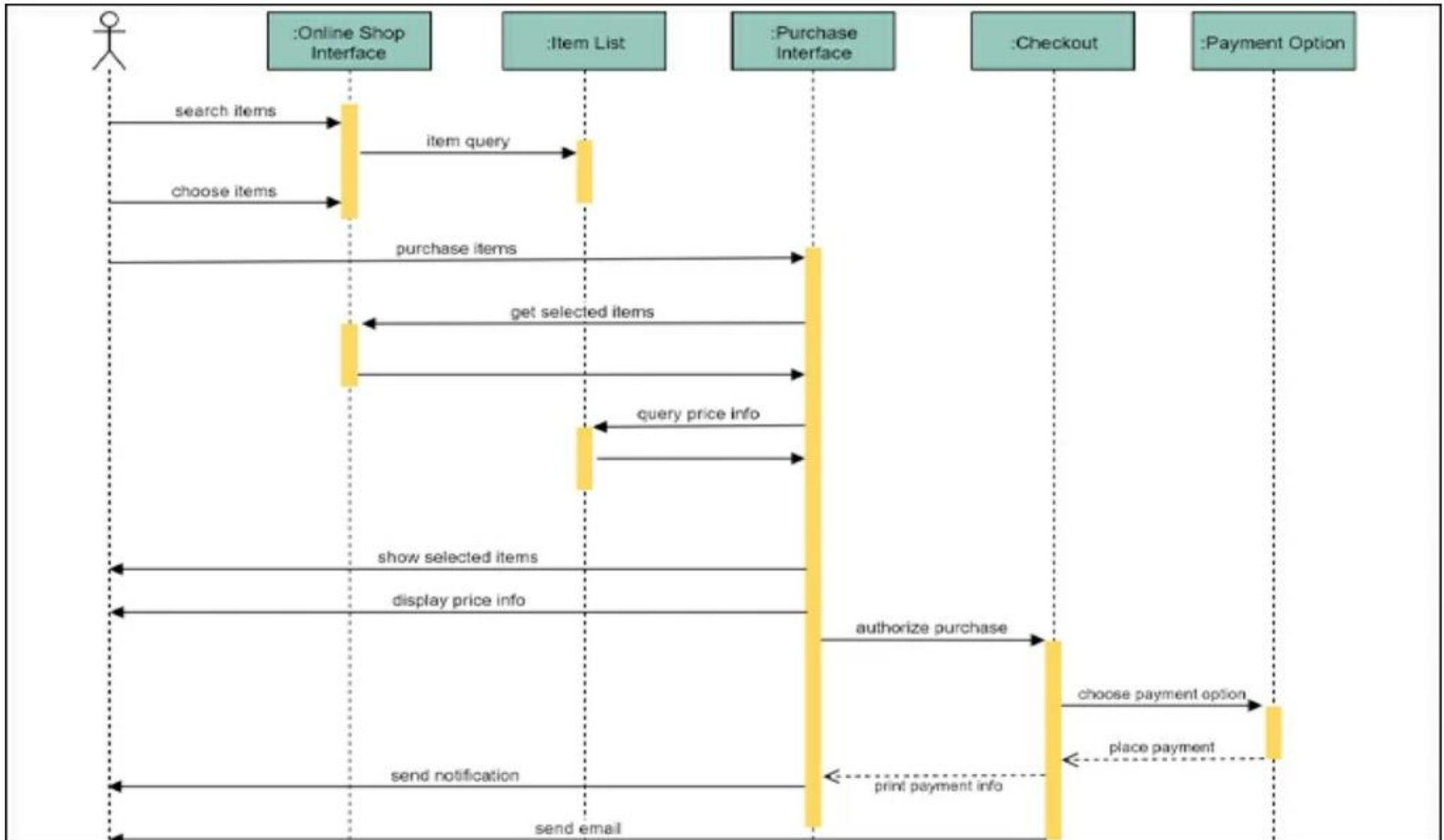
- ❖ Firstly the application is opened by the user.
- ❖ The device then gets access to the web cam.
- ❖ The webcam captures the image of the user.
- ❖ The device uses algorithms to detect the face and predict the mood.
- ❖ It then requests database for dictionary of possible moods.
- ❖ The mood is retrieved from the database.
- ❖ The mood is displayed to the user.
- ❖ The music is requested from the database.
- ❖ The playlist is generated and finally shown to the user.

Sequence diagram example for Online Shopping

A sequence diagram is used in various fields that involve processes just like in online shopping.

In this UML sequence diagram example, you will see the flow of ordering items online from browsing items to the confirmation of the order via email as shown in the illustration.

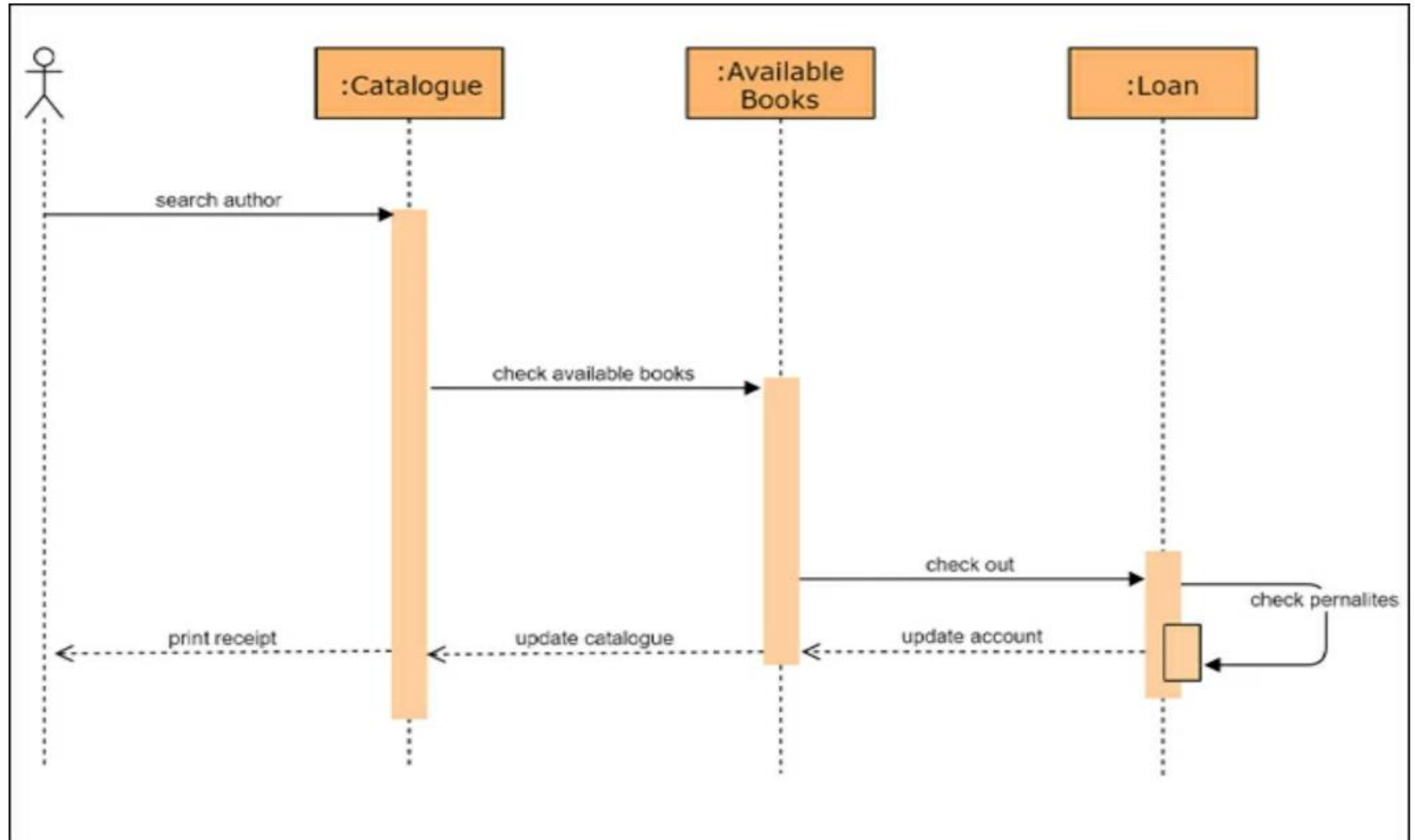
Sequence diagram example for Online Shopping



Sequence diagram example for Library Management System

There is a volume of students renting books from the library. To regulate this, it's essential for students to have online access to the record of available books. This system sequence diagram example is just made for this purpose. Also, the system will also inform the student if they exceeded the allotted time for renting a book and a penalty will apply accordingly.

Sequence diagram example for Library Management System

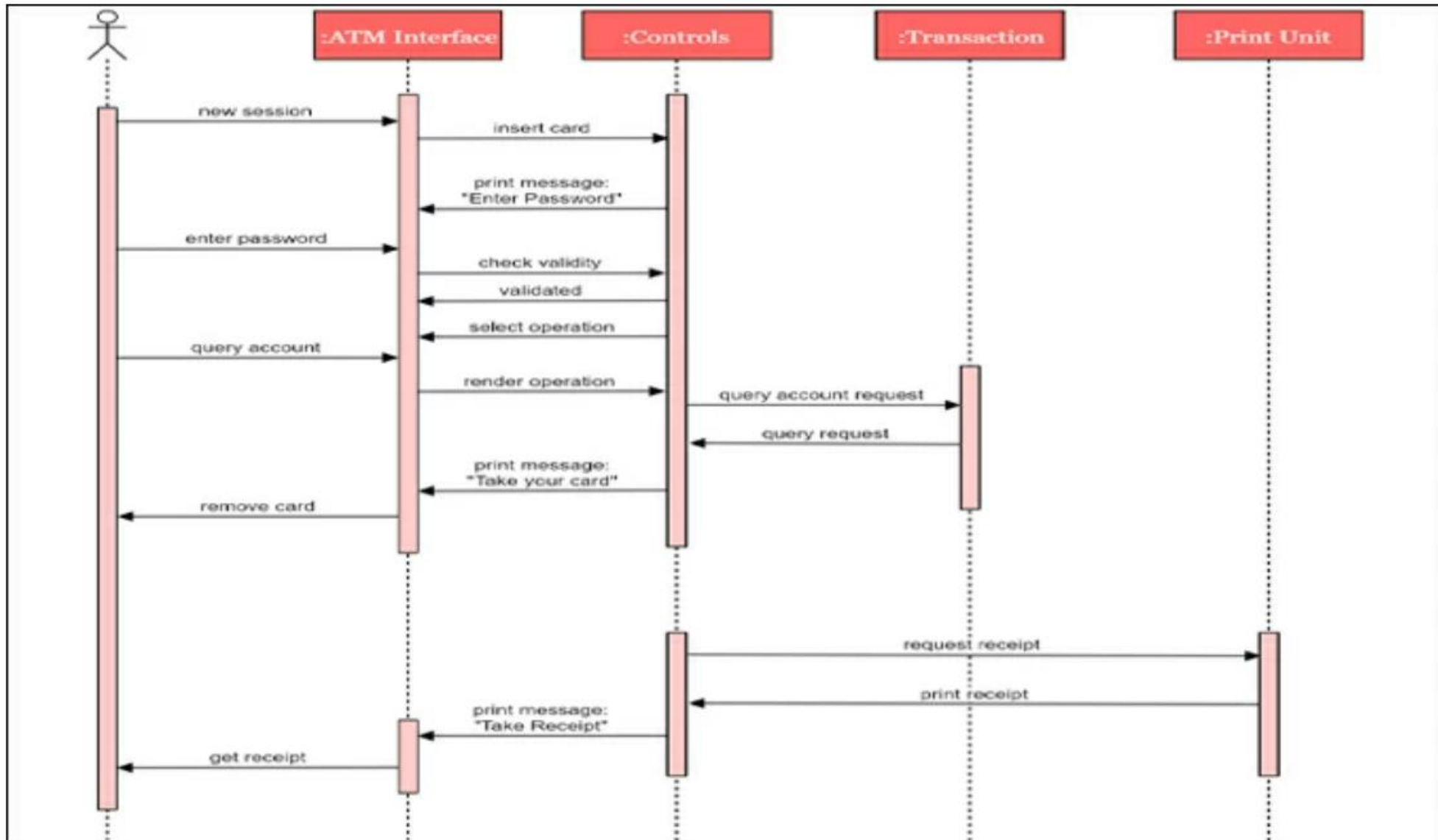


Sequence diagram example for ATM

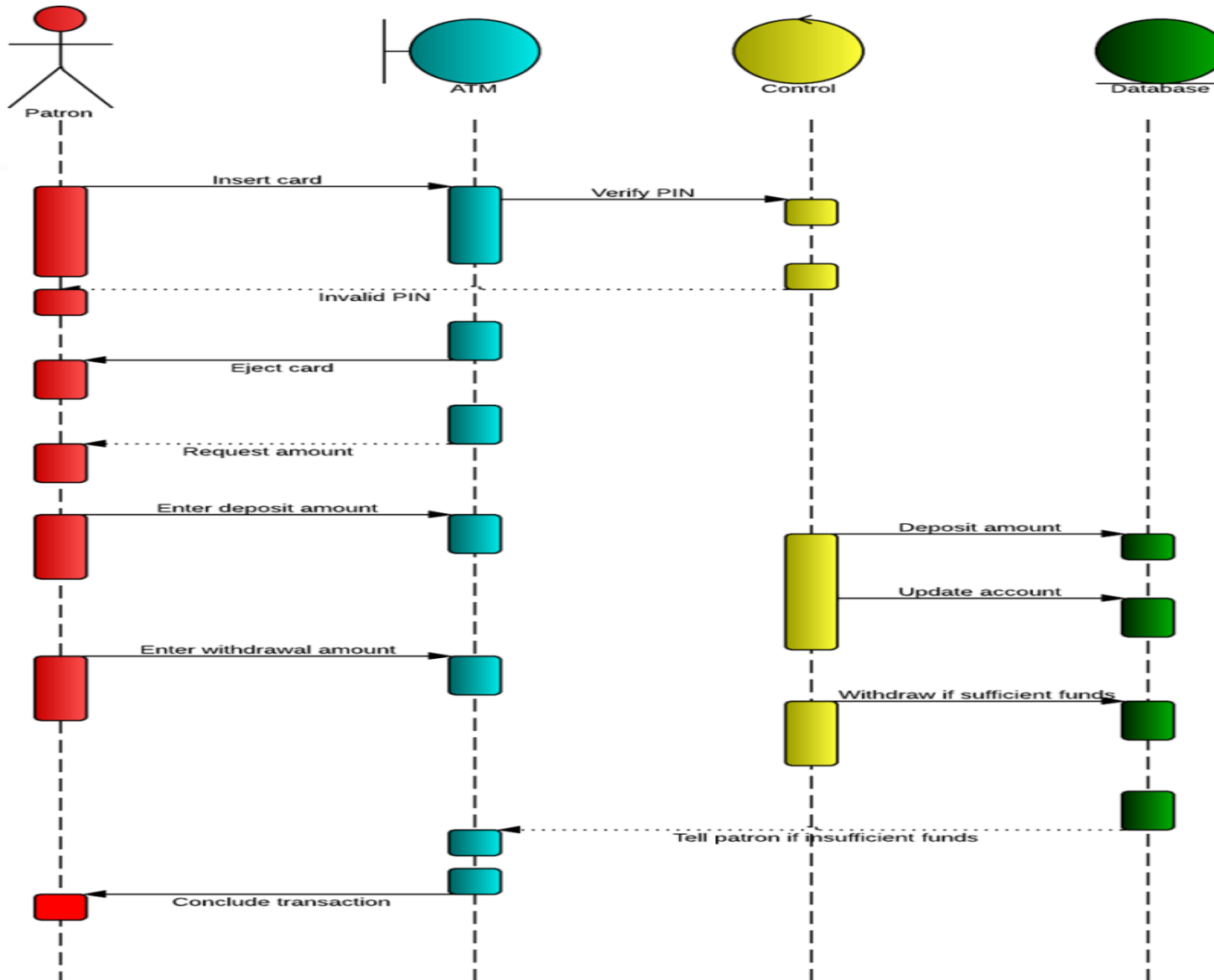
An automated teller machine lets you access your bank account.

With it, you can check your balance, enter the amount to withdraw, and many more. The sequence diagram example of an ATM below shows the processes in sequential order. From the interface access, ATM control, transaction unit, and printer.

Sequence diagram example for ATM



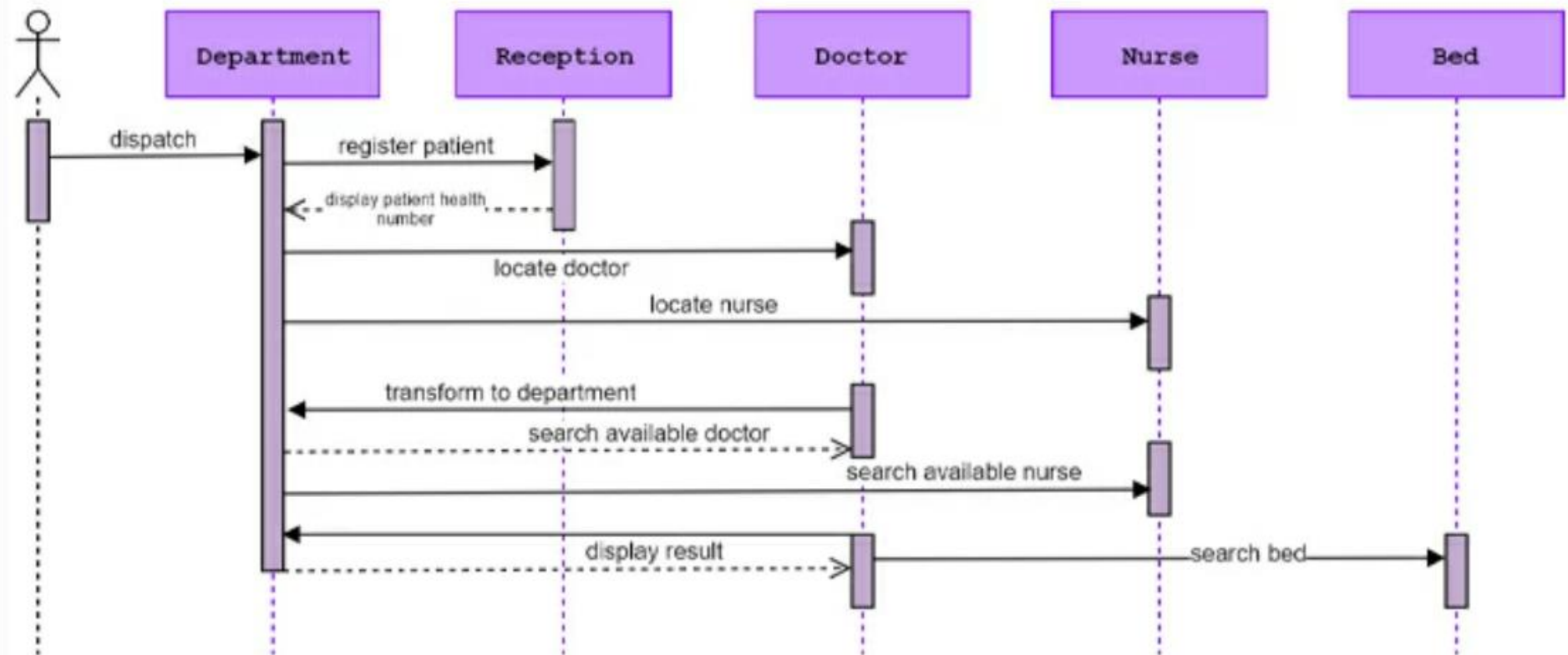
Sequence diagram example for ATM



Sequence diagram example for Hospital Management System

Through the help of a hospital management system, the establishment will be able to manage the volume of information. Moreover, doctors or nurses will be able to track the status, and health history of patients. That is illustrated in the system sequence diagram example below.

Sequence diagram example for Hospital Management System

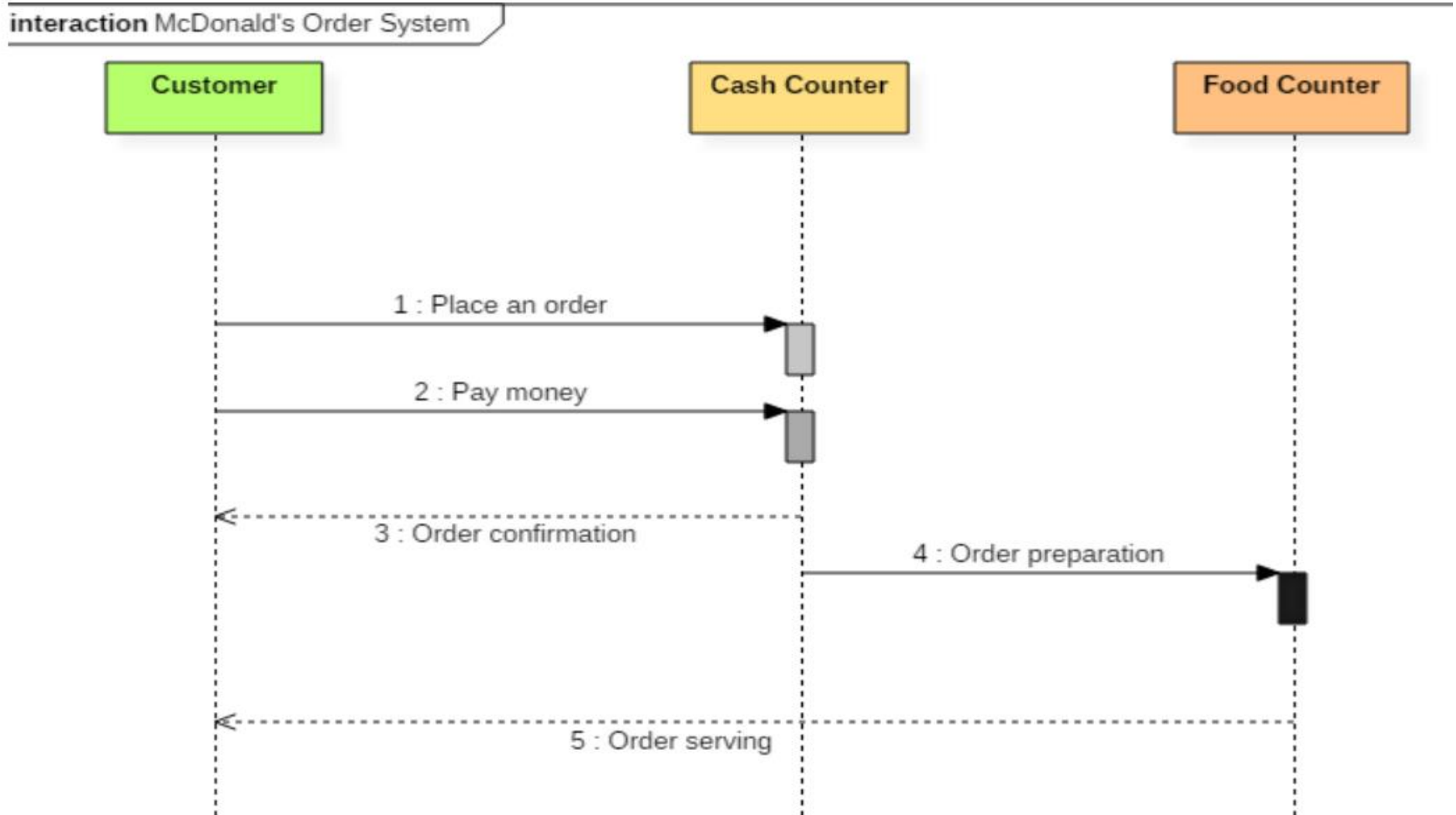


The following sequence diagram example represents McDonald's ordering system:

The ordered sequence of events in a given sequence diagram is as follows:

- Place an order.
- Pay money to the cash counter.
- Order Confirmation.
- Order preparation.
- Order serving.

The following sequence diagram example represents McDonald's ordering system:



Benefits of a Sequence Diagram

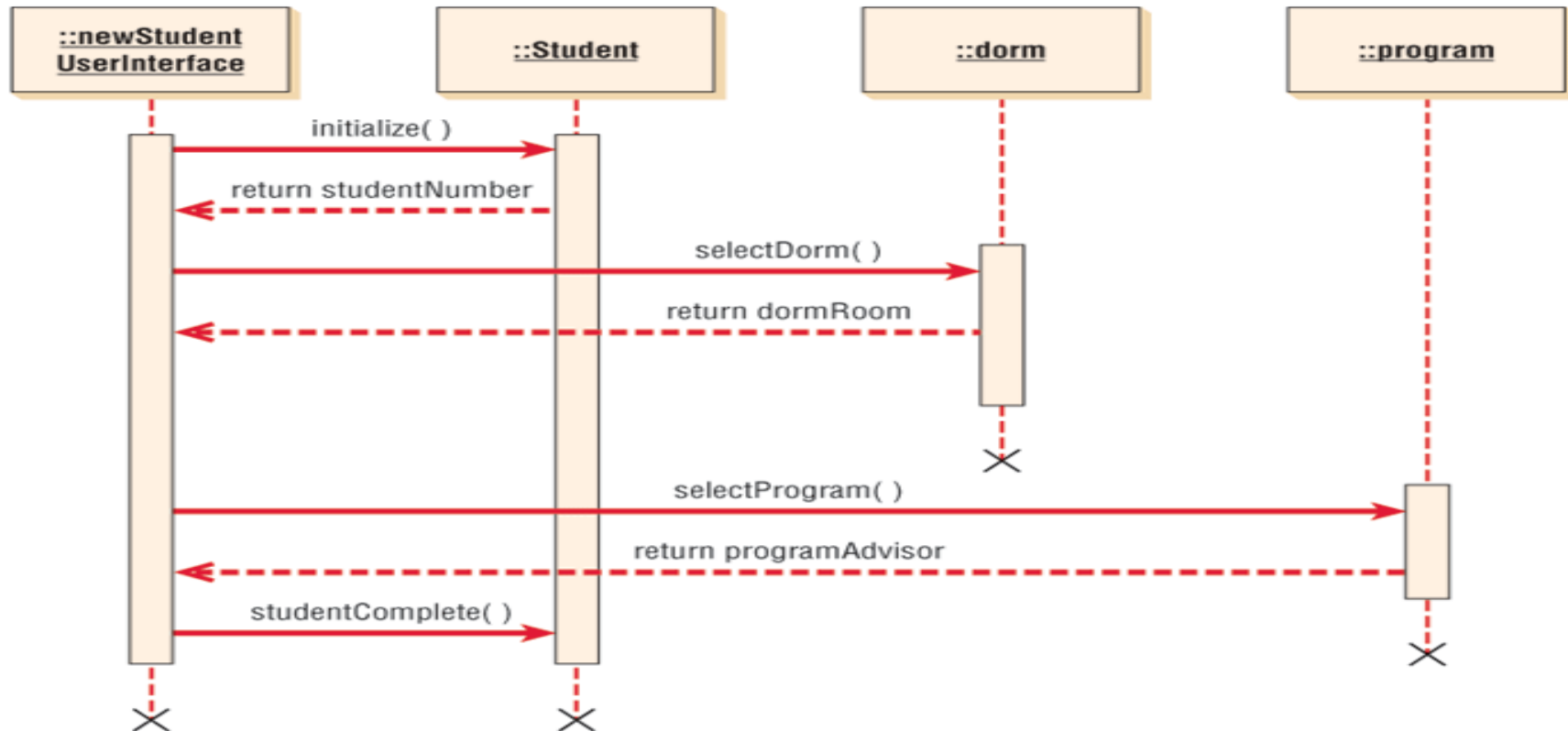
- Sequence diagrams are used to explore any real application or a system.
- Sequence diagrams are used to represent message flow from one object to another object.
- Sequence diagrams are easier to maintain.
- Sequence diagrams are easier to generate.
- Sequence diagrams can be easily updated according to the changes within a system.
- Sequence diagram allows reverse as well as forward engineering.

Drawbacks of a sequence diagram

- Sequence diagrams can become complex when too many lifelines are involved in the system.
- If the order of message sequence is changed, then incorrect results are produced.
- Each sequence needs to be represented using different message notation, which can be a little complex.
- The type of message decides the type of sequence inside the diagram.

Sequence Diagram Example

Figure 18.13 A sequence diagram for student admission. Sequence diagrams emphasize the time ordering of messages.



Steps Used in UML

The steps used in UML are:

- Define the use case model.
- Continue UML diagramming to model the system. during the systems analysis phase.
- Develop the class diagrams.
- Draw statechart diagrams.
- Begin systems design by refining the UML diagrams.
- Document your system design in detail.