



DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

Lab No. 08

Title: Develop UML Sequence and
Communication Diagram for the given project.

INTEGRATED DESIGN PROJECT I
CSE 324



GREEN UNIVERSITY OF BANGLADESH

1 Objective(s)

- To understand the interactions between objects that are represented as lifelines in a sequential order of a project using Sequence Diagram.

1.1 Sub-Objective(s)

- To depict complex interactions between multiple objects in the sequence diagram by using sequence fragments.
- To show the interactive behavior of a system.
- To model and visualise the logic behind a sophisticated function, operation or procedure.
- To show details of UML use case diagrams.
- To understand the detailed functionality of current or future systems.
- To Visualise how messages and tasks move between objects or components in a system.

2 Problem analysis

Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system. An Sequence diagram commonly known as interaction diagram is used to show the interactive behavior of a system.

Problem statement and motivation for the given project: The project entitled Banking ATM system has a drastic change to that of the older version of banking system, customer feel inconvenient with the transaction method as it was in the hands of the bank employees. In our ATM system, the above problem is overcome here, the transactions are done in person by the customer thus makes the customers feel safe and secure. Thus the application of our system helps the customer in withdrawing money, checking the balance and transaction of the amount with mini-statement and transferring the balance by validating the pin number therefore ATM system is more user friendly.

3 Methodology

To draw an UML Sequence diagram of a system we have to use some symbols and notations are shown in Fig.1, Fig.2 and Fig.3.

(a) 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. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

(b) 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

We display a lifeline in a rectangle called head with its name and type. The head is located on top of a vertical dashed line (referred to as the stem) as shown above. If we want to model an unnamed instance, we follow the same pattern except now the portion of lifeline's name is left blank.

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.

Messages can be broadly classified into the following categories

- 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.

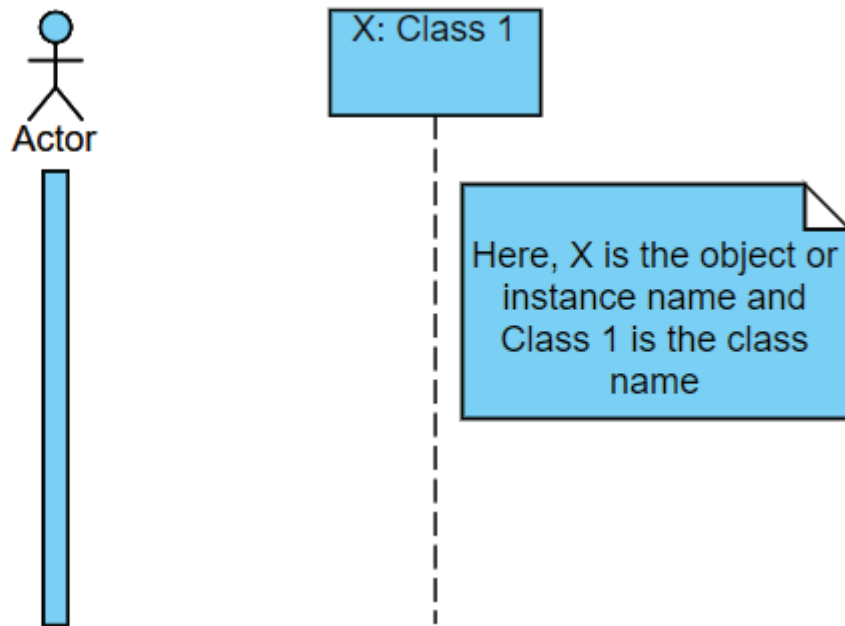


Figure 1: Symbols of Actor, Life Line and Note

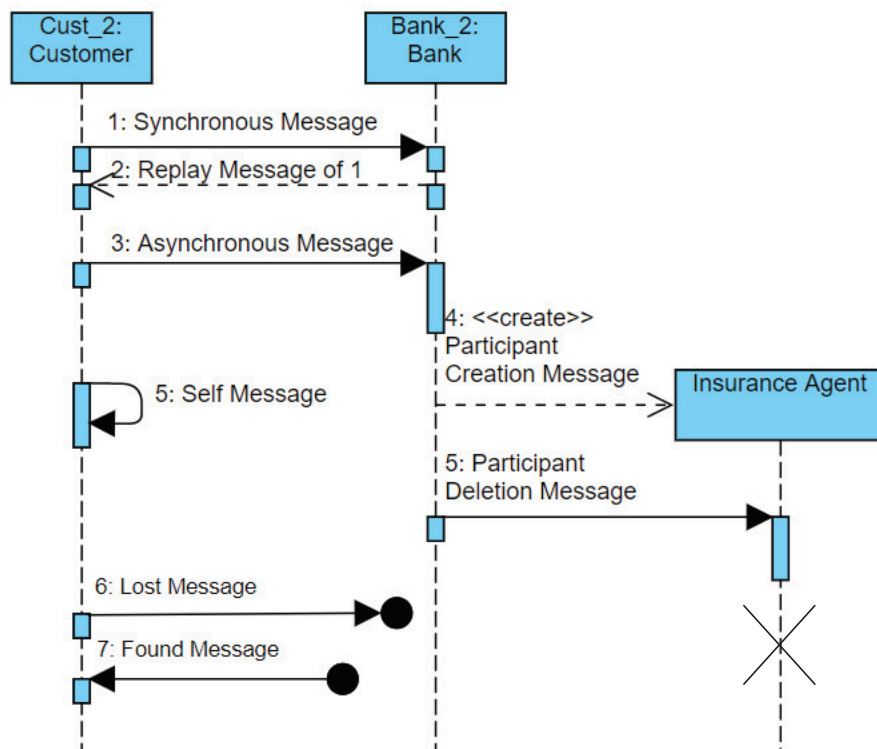


Figure 2: Symbols of different type of messages

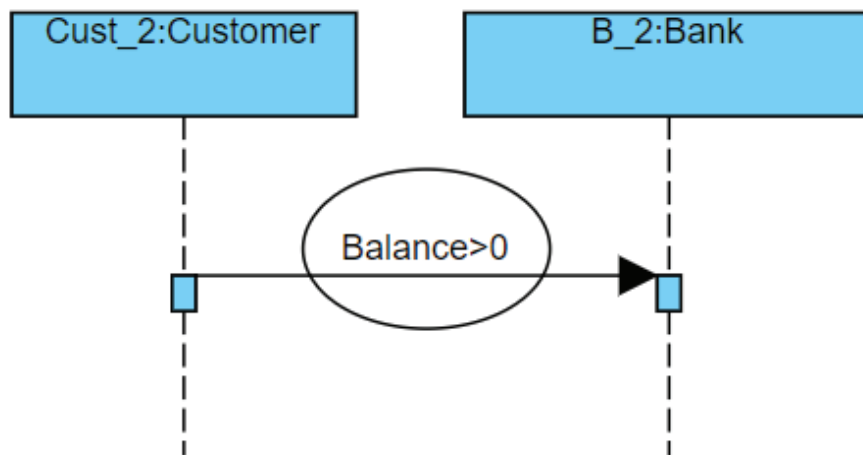


Figure 3: Symbols of Guards

- 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.
- Create message – We use a Create message to instantiate a new object in the sequence diagram. There are situations when a particular message call requires the creation of an object. It is represented with a dotted arrow and create word labelled on it to specify that it is the create Message symbol.
- Delete Message – We use a Delete Message to delete an object. When an object is deallocated memory or is destroyed within the system we use the Delete Message symbol. It destroys the occurrence of the object in the system. It is represented by an arrow terminating with a x.
- 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.
- 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.
- 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 end point. For example: Consider the scenario of a hardware failure.
- 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.

(c) **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.

3.1 Required Software

Anyone of the following tools can be used to draw the UML use case diagram.

1. Visual Paradigm for UML 8.2 (online link: <https://online.visual-paradigm.com/>)
2. StartUML
3. Lucidchart and other drawing tools

3.2 Procedure

You can draw sequence diagrams in VP-UML. A sequence diagram is used primarily to show the interactions between objects that are represented as lifelines in a sequential order. Go through the following link and follow procedure step by step to draw a sequence diagram.

4 Implementation

The Sequence diagram of the Banking ATM system is shown in Fig.4.

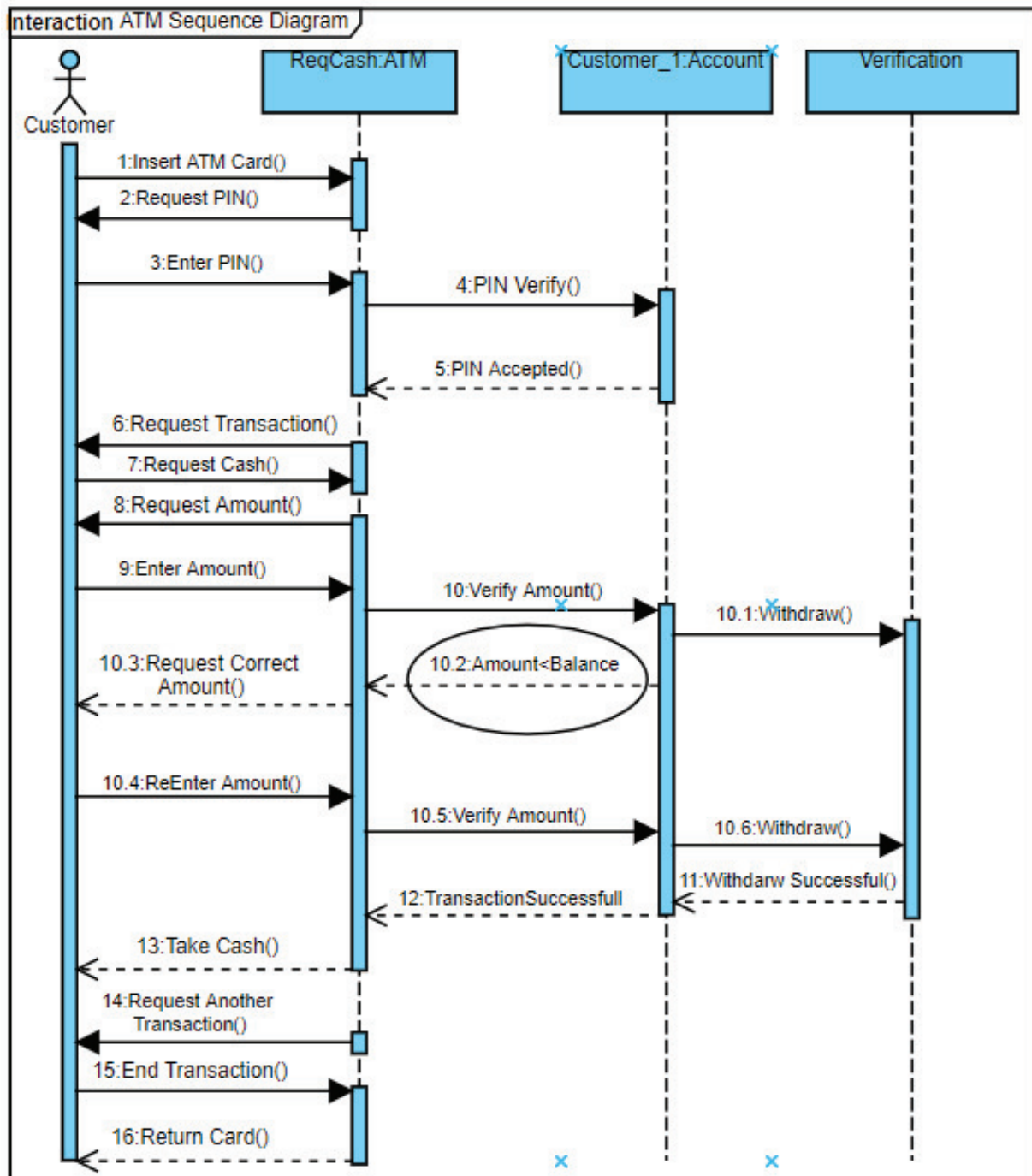


Figure 4: Sequence diagram of the Banking ATM system

5 Discussion & Conclusion

Based on the focused objective(s) to understand about UML sequence diagram, the additional lab exercise made me more confident towards the fulfilment of the objectives(s). This sequence diagram is a visual representation

the flow of functionality through a use case of a system.

6 Lab Task (Please implement yourself and show the output to the instructor)

1. Draw a Sequence diagram for the given project 'Library Management System'.

6.1 Problem analysis

A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books. The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library. Some scenarios of the 'Library Management System' are as follows:

1. User who registers himself as a new user initially is regarded as staff or student for the library system.
 - (i) For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.
 - (ii) After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.
2. After getting the library card, a new book is requested by the user as per their requirement.
3. After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.
4. Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.
5. If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.
6. User can fill the feedback form available if they want to.
7. Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.
8. Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.
9. Updating database is the important role of Librarian.

7 Lab Exercise (Submit as a report)

- Draw an Sequence diagram for the given project for your team.

8 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.