CSc 322 - Software Engineering

Lab – Fall 2017

Teaching Assistant: Isatou Sanneh
Email: isanneh00@citymail.cuny.edu

UML Class and Sequence Diagrams

October 20th, 2017

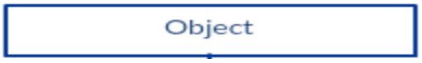
Sequence diagram

It represents the behavioral aspects of a system. Sequence diagram shows the interactions between the objects by means of passing messages from one object to another with respect to time in a system.

Elements in a sequence diagram

Sequence diagram contains the objects of a system and their life-line bar and the messages passed between them

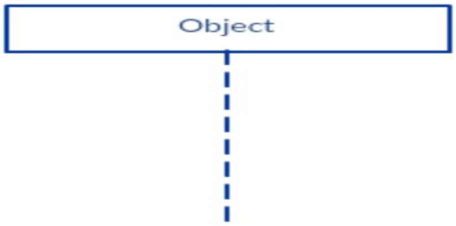
Object



Objects appear at the top portion of a sequence diagram. Object is shown in a rectangular box. Name of object precedes a colon ’:’ and the class name from which the object is instantiated. The whole string is underlined and appears in a rectangular box. Also, we may use only class name or only instance name.

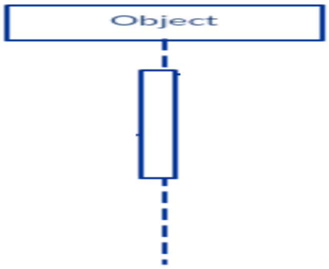
Lifeline

A downward vertical line from object-box is shown as the lifeline of the object.



Activation Bar

A rectangular bar on the lifeline indicates that it is active or instantiated at that point. The height of the rectangle indicates the duration the object is active. It starts when the message is received and ends when the object is done handling the message.



Messages

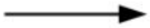
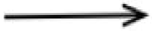
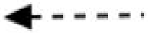
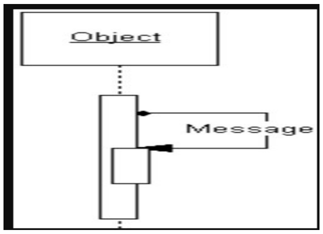
Message Type	Notation
Synchronous Message	
Asynchronous Message	
Return Message	

Figure 7: Sequence Diagram Notations

Messages are shown as an arrow from the lifeline of the sender object to the lifeline of the receiver object and labelled with the message name. Chronological order of the messages passing throughout the object's lifeline show the sequence in which they occur. An interaction between two objects occur when one object sends a message to another. There may exist different types of messages:

- *Synchronous messages:* Receiver starts processing the message after receiving it and sender needs to wait until it is made. The sender cannot send other messages to the objects in the system until it receives a return message from the receiver. In a synchronous message, we always imply that a return message is sent from the receiver to the sender. A solid line with a filled arrowhead from the sender life-line bar to the receiver represents a synchronous message. Return messages are optional so we do not have to display it unless a value is returned.

- *Asynchronous messages:* For an asynchronous message, sender does not need to wait for the receiver to process the message. A function call that creates a thread can be represented as an asynchronous message in a sequence diagram. A solid line with an open arrowhead from the sender life-line bar to the receiver represents an asynchronous message
- *Return message:* A return message is used to indicate that the message receiver is done processing the message and is returning control over to the message caller. A return message is sent back from the receiver to the sender once the receiver finishes processing the information. A dashed line with a filled arrowhead from the sender life-line bar to the receiver represents that message. *Return messages are an optional part of a sequence diagram* - only use them if a value is returned.
- *Response message:* One object can send a message to itself. We use this message when we need to show the interaction between the same object



How to Draw a Sequence Diagram

A sequence diagram represents the flow of events in one single use case.

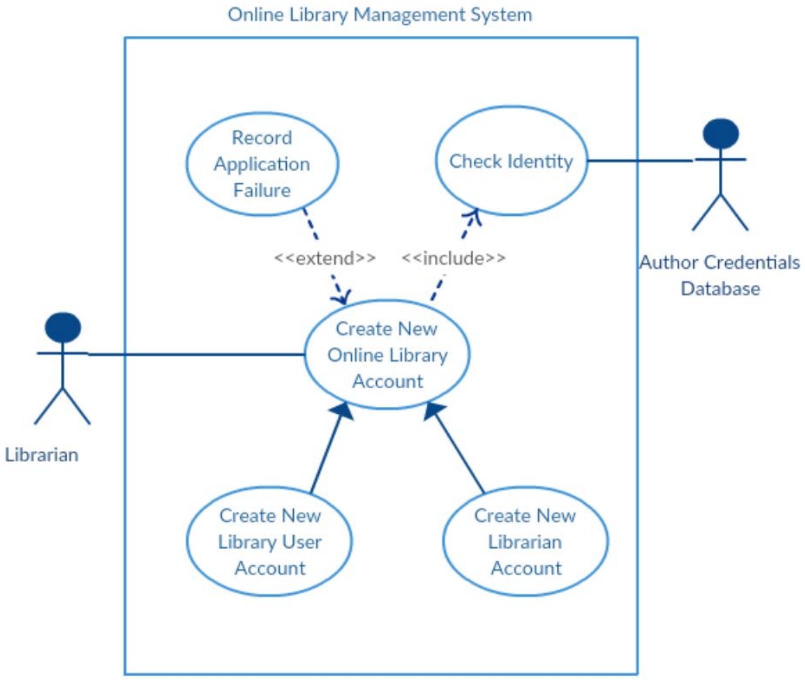

Online Library Management System
Record Application Failure
Check Identity
<<extend>>
<<include>>
Create New Online Library Account
Create New Library User Account
Create New Librarian Account
Librarian
Author Credentials Database

Figure 8: Use Case Diagram

We will draw a sequence diagram for creating a new library user account. We need to identify the objects or actors that that will be captured in our sequence diagram.
They include:

- *Librarian*
- *Online Library Management system*
- *User credentials database*
- *Email system*

We need to list out the interactions that would occur between the objects.
Here are the steps that occur in the use case named ‘Create New Library User Account’:

- The librarian request the system to create a new online library account
- The librarian then selects the library user account type
- The librarian enters the user’s details
- The user’s details are checked using the user Credentials Database
- The new library user account is created
- A summary of the of the new account’s details are then emailed to the user

From each of these steps, we can easily specify what messages should be exchanged between the objects in the sequence diagram.
The sequence diagram below shows how the objects in the online library management system interact with each other to perform the function ‘Create New Library User Account’.

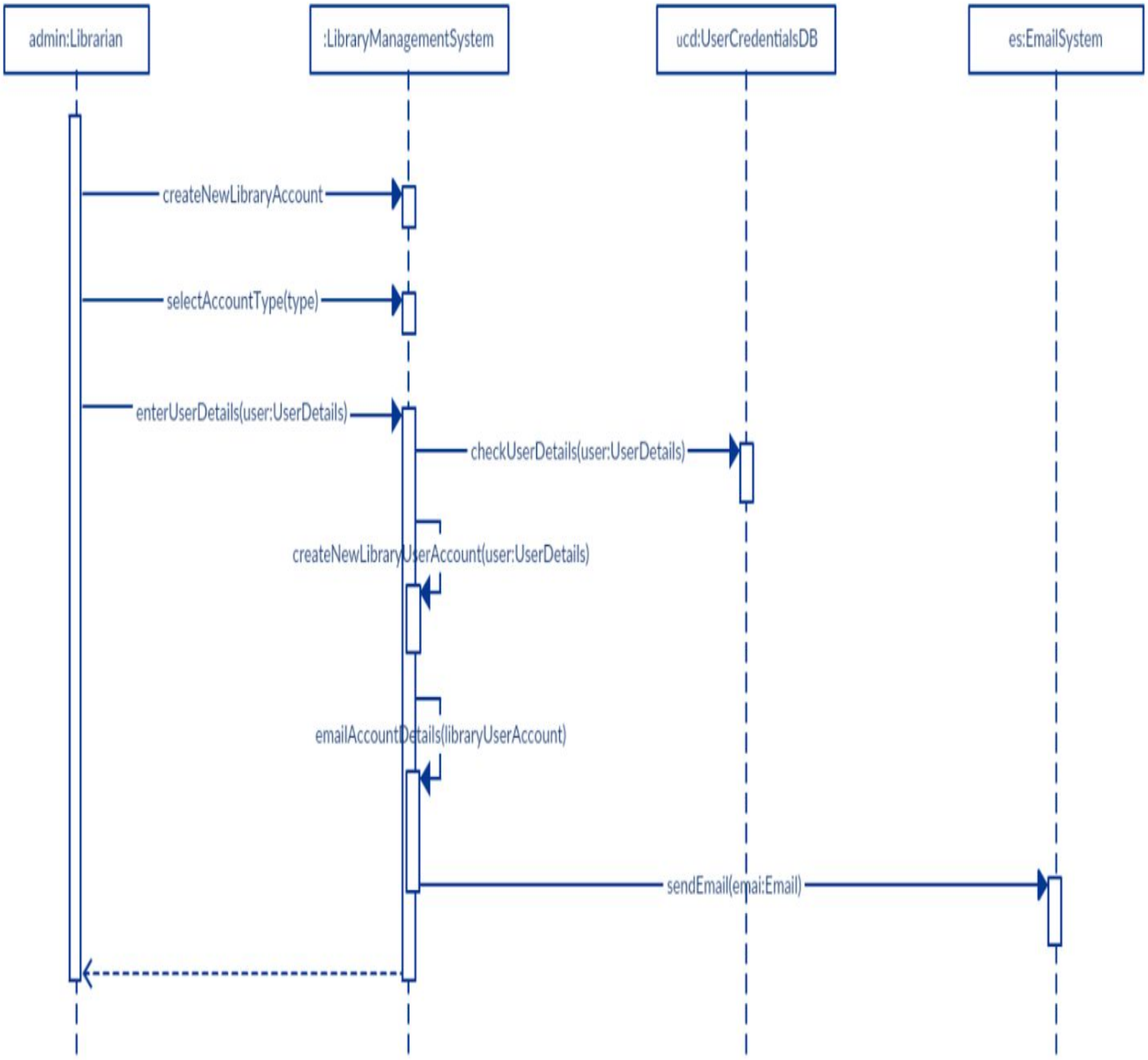
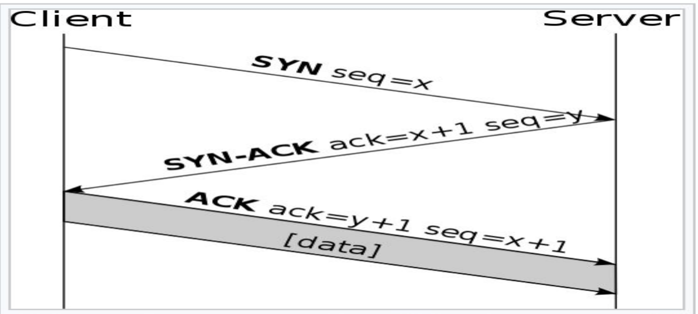

admin:Librarian
:LibraryManagementSystem
ucd:UserCredentialsDB
es:EmailSystem
createNewLibraryAccount
selectAccountType(type)
enterUserDetails(user:UserDetails)
checkUserDetails(user:UserDetails)
createNewLibraryUserAccount(user:UserDetails)
emailAccountDetails(libraryUserAccount)
sendEmail(emai:Email)

Figure 8: Sequence Diagram

Practice Tasks

Part 1 Represent the three-way handshaking mechanism of TCP with a sequence diagram



Schematic representation of the three steps that make a network connection.

Part 2 Sequence Diagram of a Web Browser

A web browser is a software that is used to access a resource (web page) available on the World Wide Web and identified by a URL. Whenever a user types in the URL of a web page in the browser’s address bar and clicks the "Go" button, the browser sends a HTTP request to the concerned web server. If the requested resource is available and accessible, the web server sends back a HTTP response to the requesting web browser. In case of any error, a HTTP response is sent indicating the error.

When the web browser receives a HTTP response, it displays the web page to the user. In very simple terms, a web browser can be thought of consisting the following sub-components: rendering engine and browser control.

Once a HTTP response has been obtained from the server, the rendering engine decides the layout of the contents and actually displays the requested page. This is done keeping in mind the different HTML elements that are present in the page, and corresponding CSS rules, if any.

The browser control provides facilities like navigating across pages (by following hyperlinks), reloads a page, and handles other events related to the window display, for example, resizing the browser window.

Instruction: Identify sequence of activities and draw the sequence diagrams.

Part 3 Draw a sequence diagram for your team project (Coding Turk System)

******

.