SAPID: 60004200082, 60004200066

Names: Pratham Bhoir, Aayushman Gupta

Batch: A2

Experiment No. 6

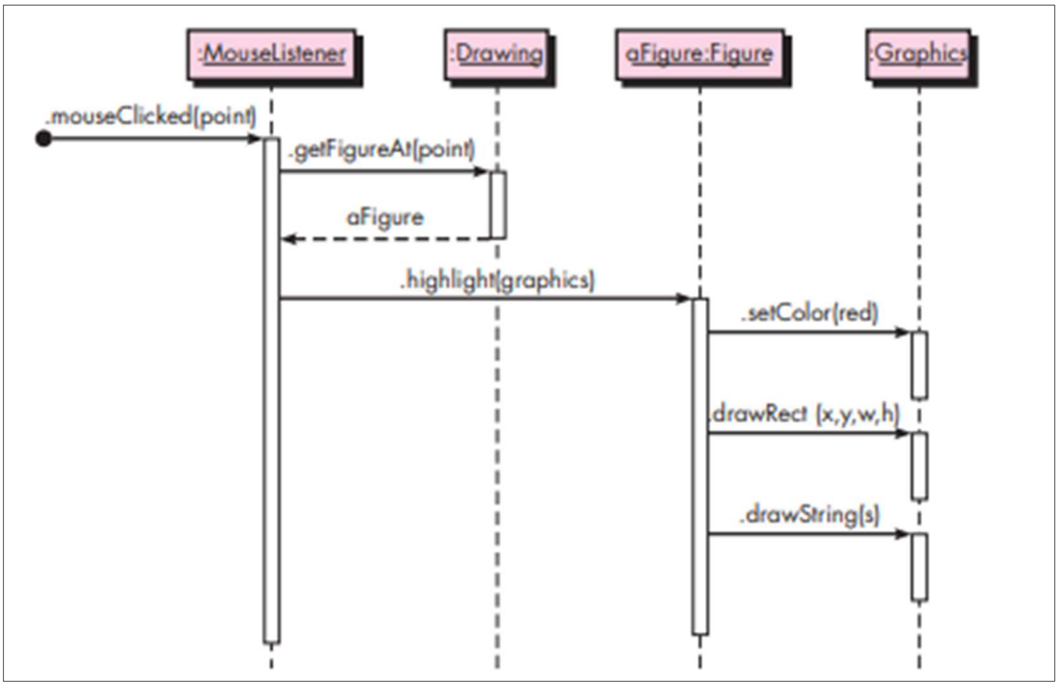
Aim: Develop Sequence and Collaboration diagram for the project.

Theory:

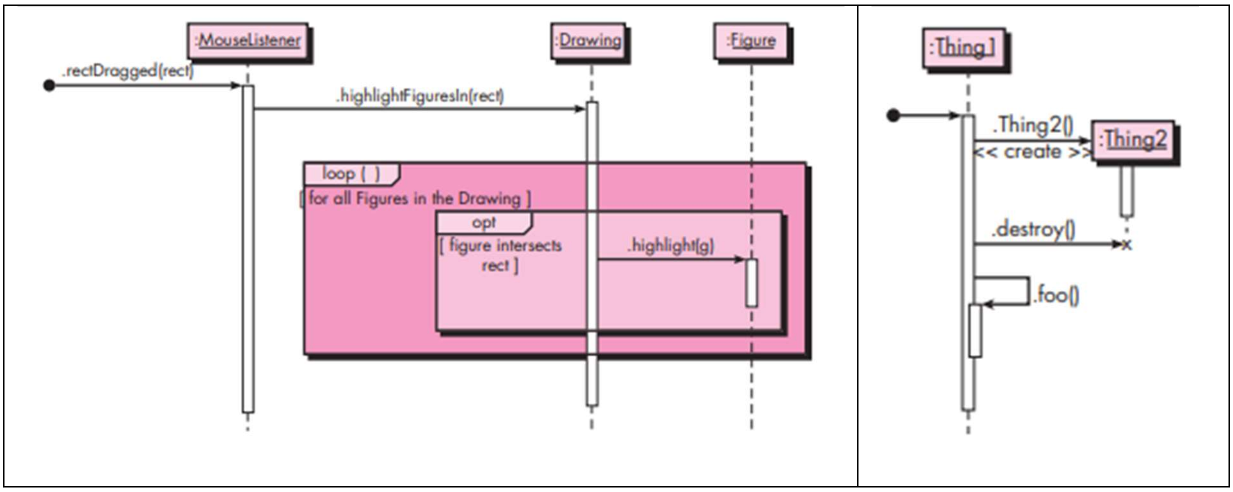
Sequence Diagram:

A sequence diagram is used to show the dynamic communications between objects during execution of a task. It shows the temporal order in which messages are sent between the objects to accomplish that task. One might use a sequence diagram to show the interactions in one use case or in one scenario of a software system.

A sequence diagram shows method calls using horizontal arrows from the caller to the callee, labeled with the method name and optionally including its parameters, their types, and the return type.

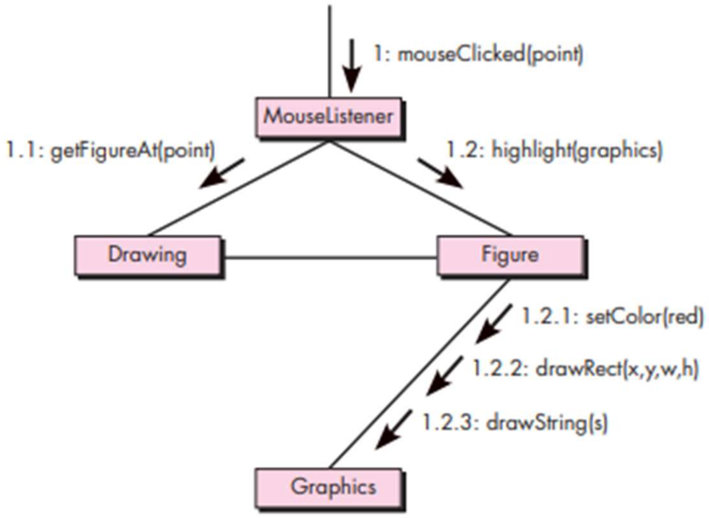


The figure shows a sequence diagram for a drawing program. The diagram shows the steps involved in highlighting a figure in a drawing when it has been clicked. Each box in the row at the top of the diagram usually corresponds to an object, although it is possible to have the boxes model other things, such as classes. If the box represents an object (as is the case in all our examples), then inside the box you can optionally state the type of the object preceded by the colon. You can also precede the colon and type by a name for the object, as shown in the third box in Figure. Below each box there is a dashed line called the lifeline of the object. The vertical axis in the sequence diagram corresponds to time, with time increasing as you move downward. The diagram in Figure is very straightforward and contains no conditionals or loops. If logical control structures are required, it is probably best to draw a separate sequence diagram for each case. That is, if the message flow can take two different paths depending on a condition, then draw two separate sequence diagrams, one for each possibility



Collaboration (communication) Diagrams:

A collaboration diagram, illustrated in Figure below, displays the same actions shown in the sequence diagram in Figure.

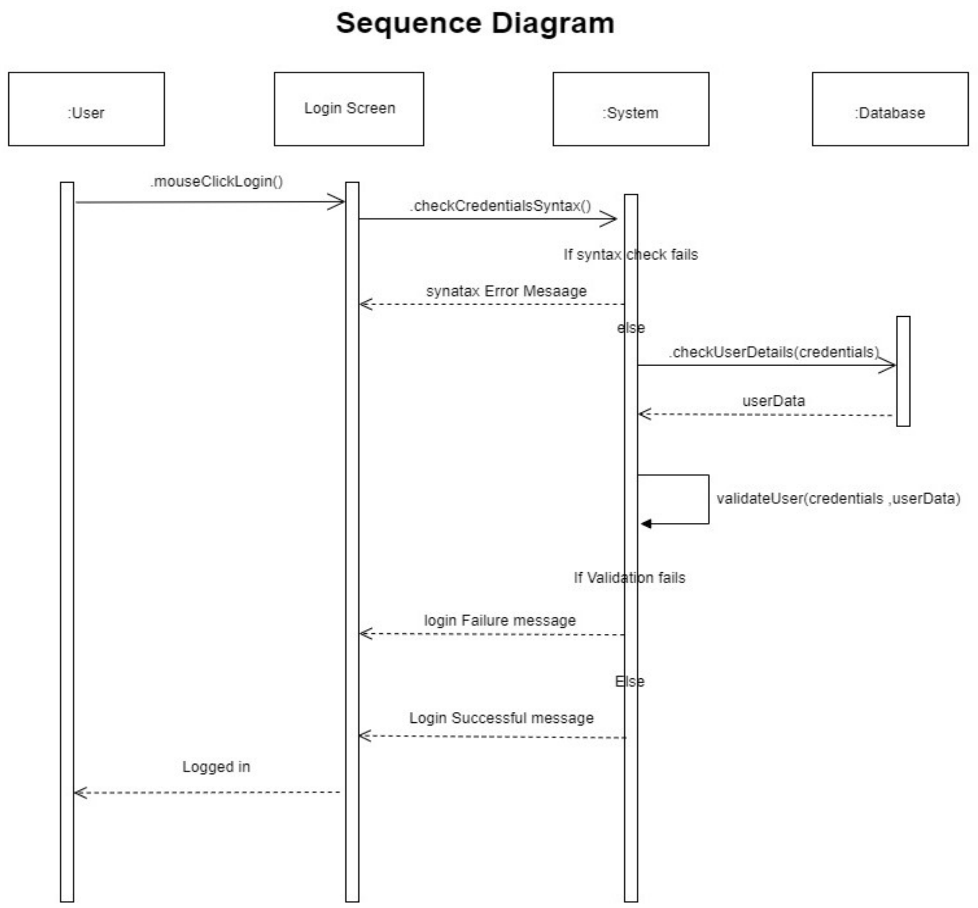


In a collaboration diagram the interacting objects are represented by rectangles. Associations between objects are represented by lines connecting the rectangles. There is typically an incoming arrow to one object in the diagram that starts the sequence of message passing. That arrow is labeled with a number and a message name. If the incoming message is labeled with the number 1 and if it causes the receiving object to invoke other messages on other objects, then those messages are represented by arrows from the sender to the receiver along an association line and are given numbers 1.1, 1.2, and so forth, in the order they are called. If those messages in turn invoke other messages, another decimal point and number are added to the number labeling these messages, to indicate further nesting of the message passing. In Figure, you see that the mouseClicked message invokes the methods getFigureAt() and then highlight(). The highlight() message invokes three other messages: setColor(), drawRect(), and drawstring(). The numbering in each label shows the nesting as well as the sequential nature of each message.

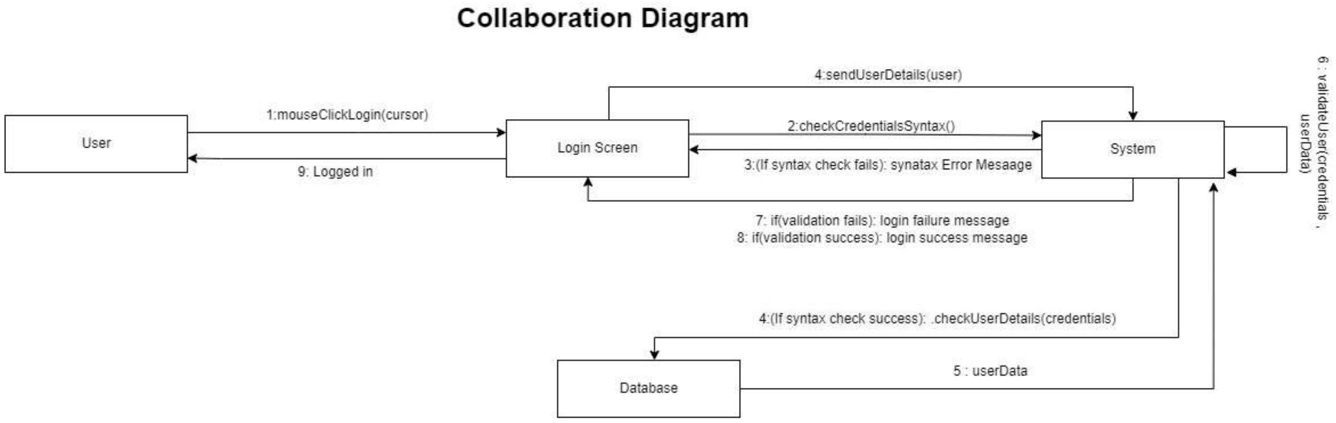
Practical:

For Sequence diagram

We have used the Login Functionality



For Collaboration diagram


Collaboration Diagram
User
Login Screen
System
Database
1:mouseClickLogin(cursor)
9: Logged in
4:sendUserDetails(user)
2:checkCredentialsSyntax()
3:(If syntax check fails): synatax Error Mesaage
7: if(validation fails): login failure message
8: if(validation success): login success message
6 : validateUser(credentials , userData)
4:(If syntax check success): .checkUserDetails(credentials)
5 : userData

Conclusion:

Thus, we are able to draw Sequence and Collaboration diagram for a functionality of our case study.