SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Software Engineering - Experiment 5

SAP ID: **60004200155, 60004210166**
Name: **Jigar Siddhpura, Aman Nambisan**
Div: **C2**
Batch: **C22**

Aim

Develop Sequence and Collaboration diagram for the project.

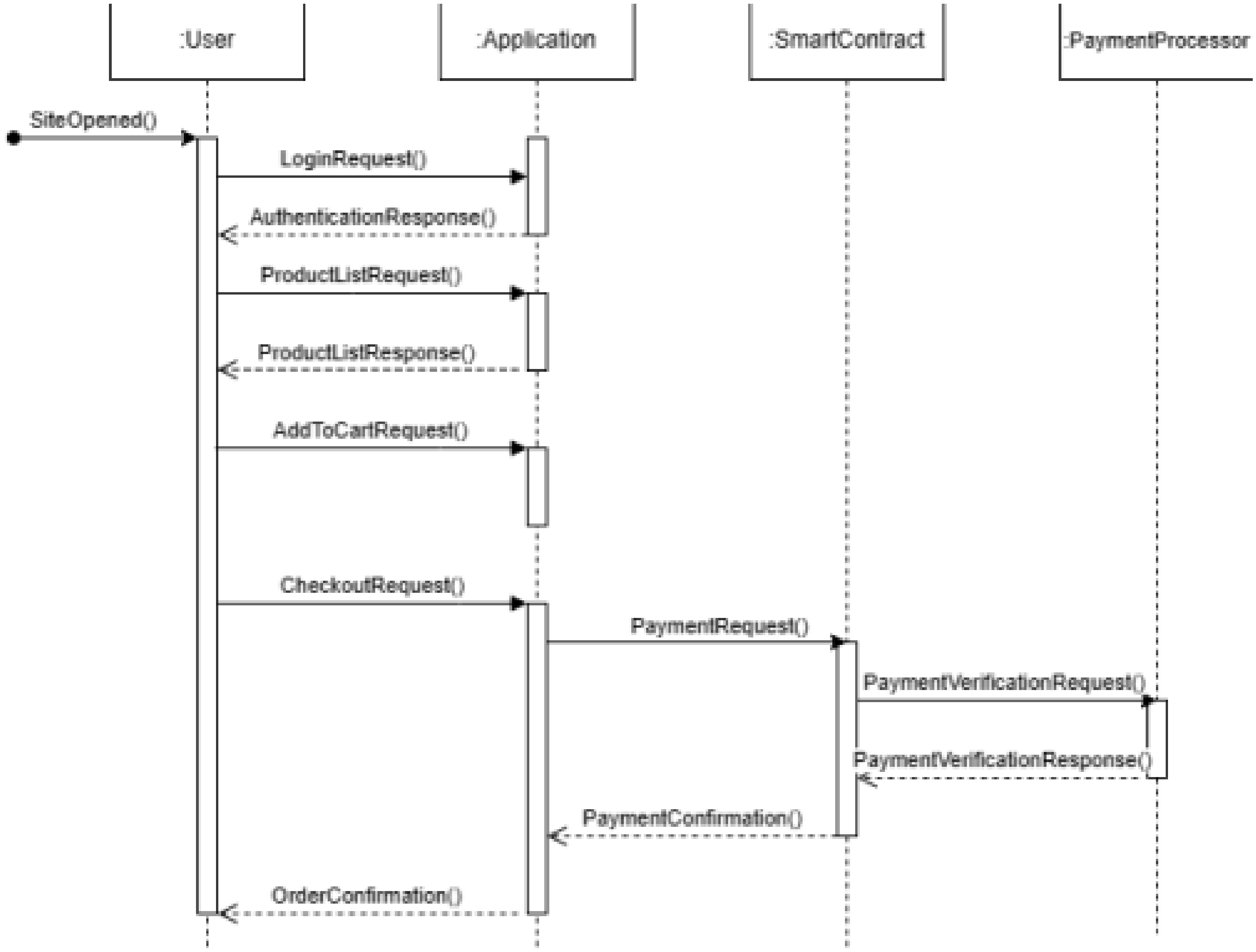
Theory

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.

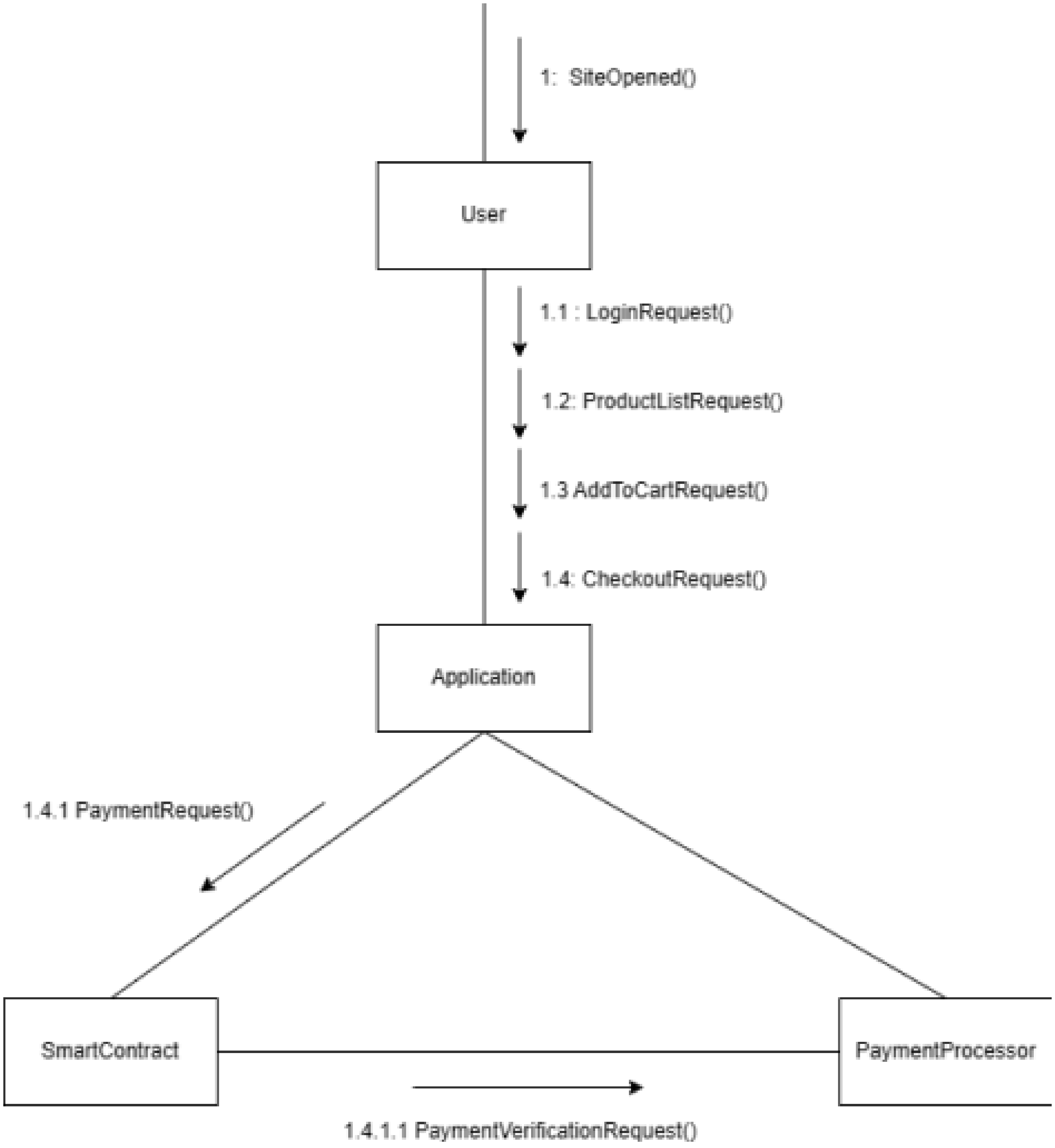
A sequence diagram shows method calls using horizontal arrows from the caller to the callee, labeled with the method name and optionally including its parameters, their types, and the return type. Each box in the row at the top of the diagram usually corresponds to an object, although it is possible 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. 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. 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

A collaboration diagram 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.

Sequence Diagram



Collaboration Diagram



Conclusion

In conclusion, sequence and collaboration diagrams are both powerful tools for modelling and visualising the behaviour of software systems, particularly in the context of object-oriented programming.

Sequence diagrams show the temporal ordering of messages between objects, providing a clear representation of the flow of control and the interactions between objects over time. This makes them useful for modelling complex scenarios involving multiple objects or actors, and for identifying potential issues or bottlenecks in the system.

Collaboration diagrams, on the other hand, show the relationships between objects and their interactions in a static view, allowing developers to see how objects collaborate to accomplish specific tasks. Collaboration diagrams are especially useful for identifying complex relationships between objects, and for visualising how objects interact with one another to implement a particular functionality.