EXPERIMENT - 7

Software Engineering Lab

Aim

To perform the behavioral view diagram for the suggested system: Sequence diagram, Collaboration diagram.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 7

Aim:

To perform the behavioral view diagram for the suggested system: Sequence diagram, Collaboration diagram.

Theory:

Sequence diagrams show potential interactions between objects in the system being defined. Normally these are specified as part of a use case or use case flow and show how the use case will be implemented in the system.

They include:

Objects - oblong boxes or actors at the top - either named or just shown as belonging to a class, from, or to which messages are sent to other objects.

Messages - solid lines for calls and dotted lines for data returns, showing the messages that are sent between objects including the order of the messages which is from the top to the bottom of the diagram.

Object lifelines - dotted vertical lines showing the lifetime of the objects.

Activation - the vertical oblong boxes on the object lifelines showing the thread of control in a synchronous system.

Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case. They are almost self-explanatory; they show the calls between the different objects in their sequence and can show, at a detailed level, different calls to different objects.

A sequence diagram has two dimensions:

- The vertical dimension shows the sequence of messages/calls in the time order that they occur

- The horizontal dimension shows the object instances to which the messages are sent.

Collaboration Diagram

They are the same as sequence diagrams but without a time axis:

- Their message arrows are numbered to show the sequence of message sending.

- They are less complex and less descriptive than sequence diagrams.

These diagrams are very useful during design because you can figure out how objects communicate with each other.

Performance Instruction:

To draw Sequence Diagram

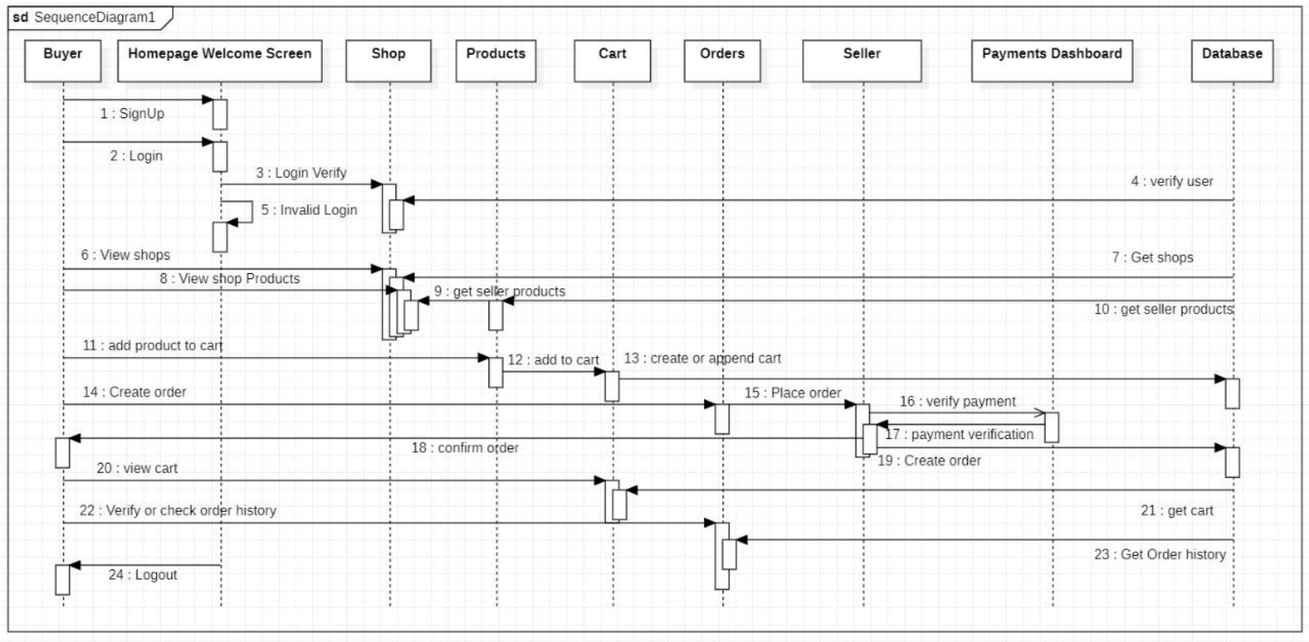
- Identify the class instances (objects) by putting each class instance inside a box.
- If a class instance sends a message to another class instance, draw a line with an open arrowhead pointing to the receiving class instance; place the name of the message/method above the line.
- Optionally, for important messages, you can draw a dotted line with an arrowhead pointing back to the originating class instance; label the return value above the dotted line.

To draw collaboration Diagram

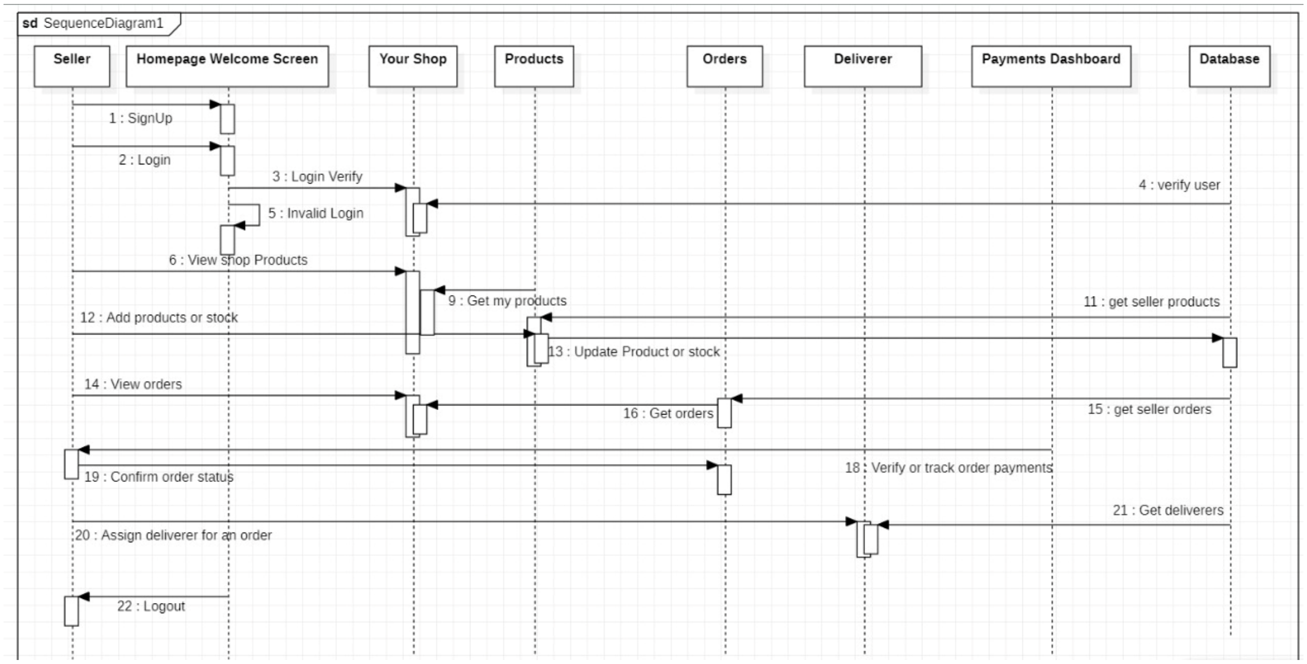
1. By simply pressing combination of keys, we can design collaboration diagram from sequence diagram.

Output:

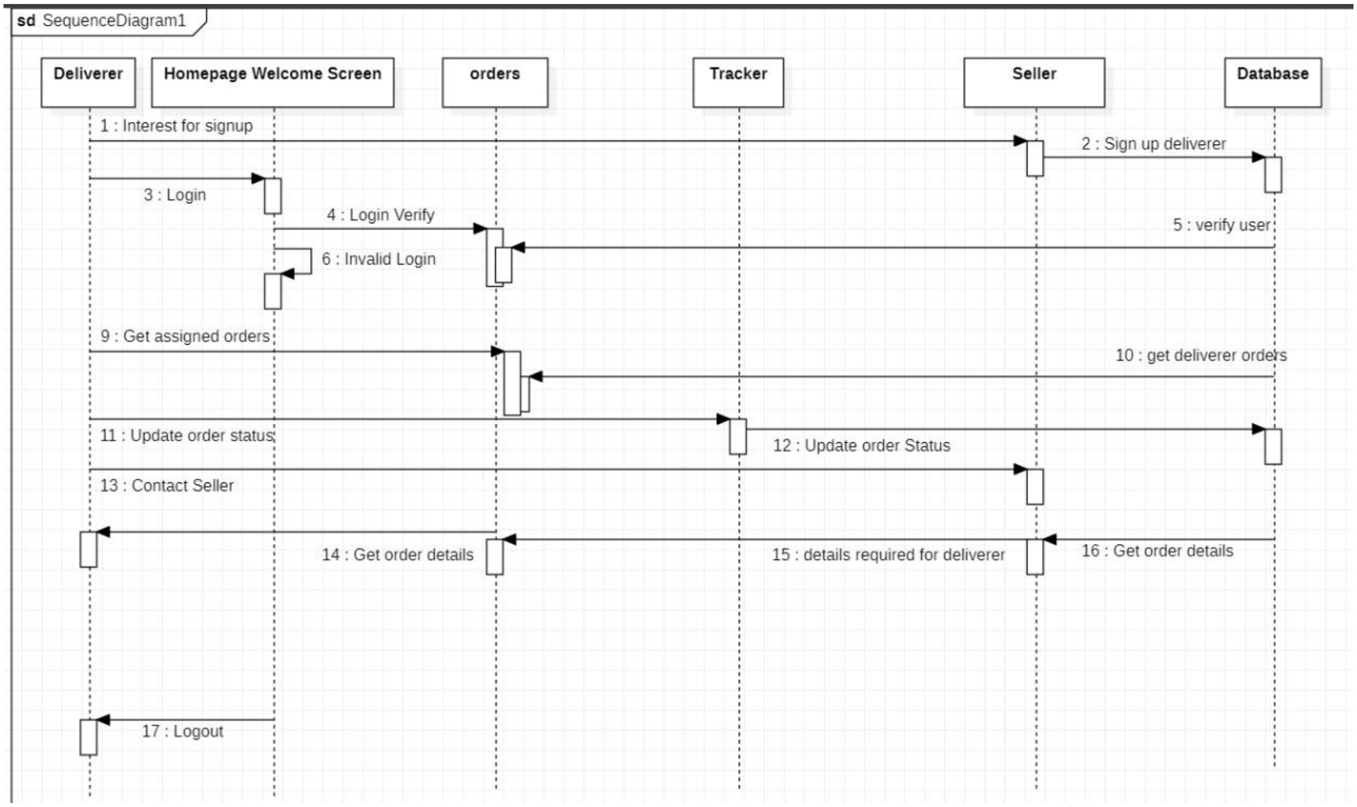
Sequence Diagram for Buyer



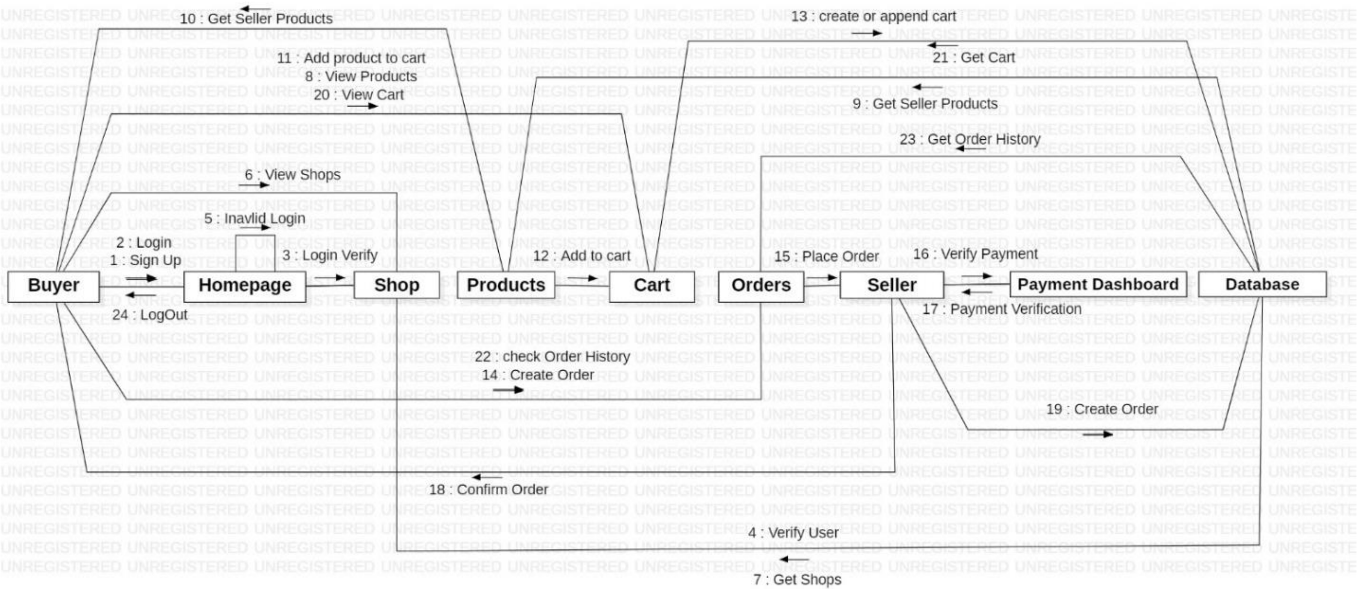
Sequence Diagram for Seller



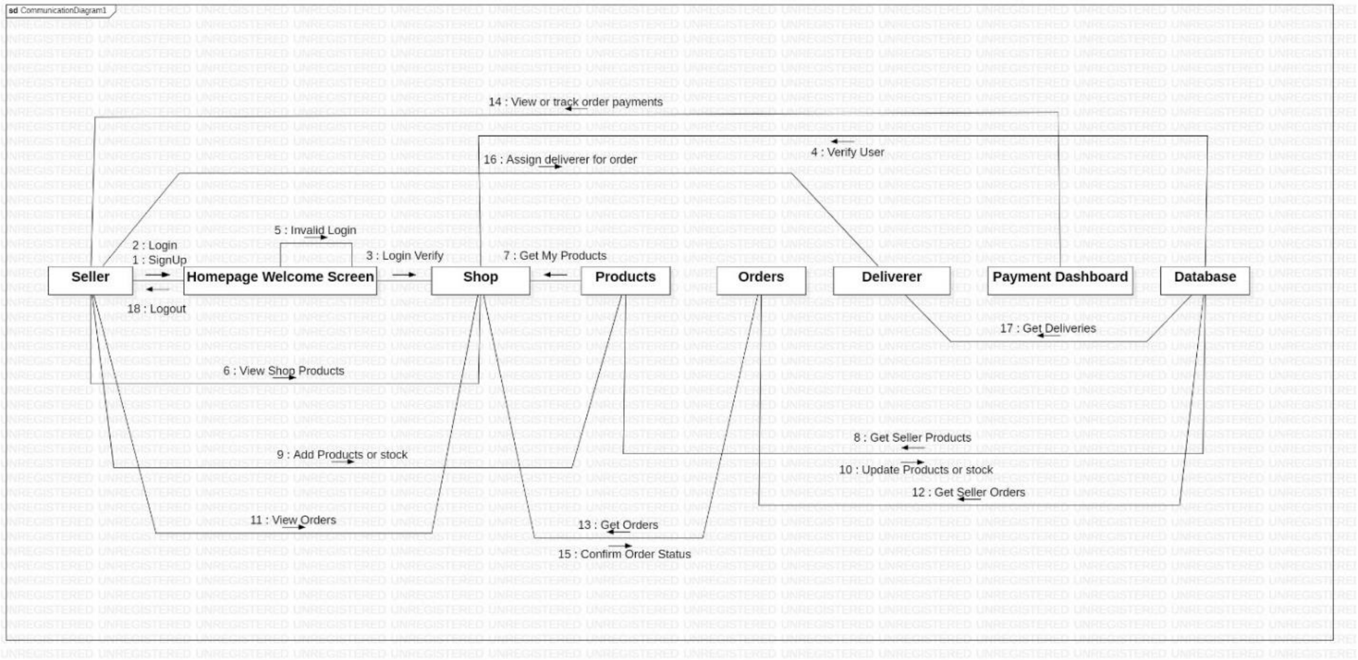
Sequence Diagram for Deliverer



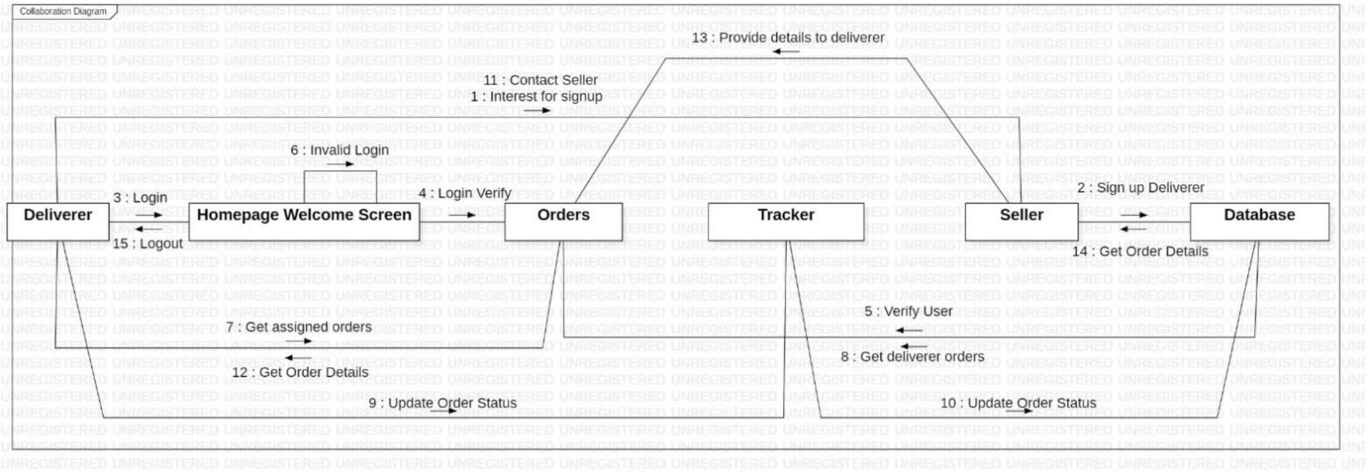
Collaboration Diagram for Buyer



Collaboration Diagram for Seller



Collaboration Diagram for Deliverer



Conclusion:

Sequence diagram and Collaboration Diagram were made successfully by following above steps.

Viva Questions

1. Explain use of sequence diagram?

Ans.

The following scenarios are ideal for using a sequence diagram:

- **Usage scenario:** A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system.
- **Method logic:** Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to explore the logic of any function, procedure, or complex process.
- **Service logic:** If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.
- **Sequence diagram Visio** - Any sequence diagram that you create with Visio can also be uploaded into Lucidchart. Lucidchart supports .vsd and .vdx file import and is a great Microsoft Visio alternative. Almost all of the images you see in the UML section of this site were generated using Lucidchart.

2. Explain steps to draw sequence diagram?

Ans.

1. Identify the class instances (objects) by putting each class instance inside a box.
2. If a class instance sends a message to another class instance, draw a line with an open arrowhead pointing to the receiving class instance; place the name of the message/method above the line.
3. Optionally, for important messages, you can draw a dotted line with an arrowhead pointing back to the originating class instance; label the return value above the dotted line.

3. Explain need of collaboration diagram?

Ans.

1. Modeling collaborations, mechanisms or the structural organization within a system design.
2. Providing an overview of collaborating objects within an object-oriented system.
3. Exhibiting many alternative scenarios for the same use case.
4. Demonstrating forward and reverse engineering.
5. Capturing the passage of information between objects.
6. Visualizing the complex logic behind an operation.

4. Explain steps to draw collaboration diagram?

Ans.

By simply pressing combination of keys, we can design collaboration diagram from sequence diagram.

5. Explain terms- entity objects, interface objects and control objects?

Ans.

Entity Objects : Entity objects are classes that encapsulate the business model, including rules, data, relationships, and persistence behavior, for items that are used in your business application.

Interface objects: Interface objects provide all details on how one process component exchanges data with another, for example, the mode of communication and the data structures.

Control objects : Control objects are also known as widgets or gadgets and they can be used in windows and dialog boxes. They cannot exist outside a window or dialog box, so you have to define a window or dialog box and select it before you can define controls.