

## **Experiment 8**

**Aim:** To create a sequence diagram for the project Object Detection Solutions

### **Requirements:**

Hardware:

- Intel 7<sup>th</sup> Gen CPU
- Graphics Card
- Storage – Hard Disk

Software:

- OS: Ubuntu, Windows, MacOS
- Umbrello UML Modeler

### **Theory:**

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Sequence Diagrams captures:

- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction)

Sequence Diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically):

- Object Dimension
- The horizontal axis shows the elements that are involved in the interaction
- Conventionally, the objects involved in the operation are listed from left to right according to when they take part in the message sequence. However, the elements on the horizontal axis may appear in any order
- Time Dimension
- The vertical axis represents time proceedings (or progressing) down the page.

Note that:

Time in a sequence diagram is all about ordering, not duration. The vertical space in an interaction diagram is not relevant for the duration of the interaction.

### **Benefits of sequence diagrams**

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

### **Use cases for sequence diagrams**

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.

### **Components of Sequence Diagram:**

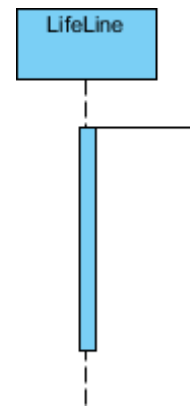
#### **Lifeline**

- A lifeline represents an individual participant in the Interaction.



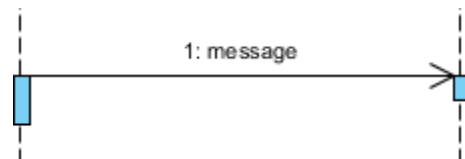
### Activations

- A thin rectangle on a lifeline) represents the period during which an element is performing an operation.
- The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



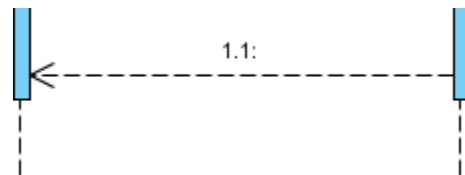
### Call Message

- A message defines a particular communication between Lifelines of an Interaction.
- Call message is a kind of message that represents an invocation of operation of target lifeline.



### Return Message

- A message defines a particular communication between Lifelines of an Interaction.
- Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.



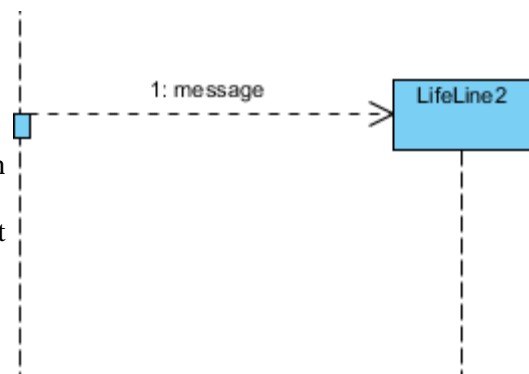
### Self-Message

- A message defines a particular communication between Lifelines of an Interaction.
- Self-message is a kind of message that represents the invocation of message of the same lifeline.

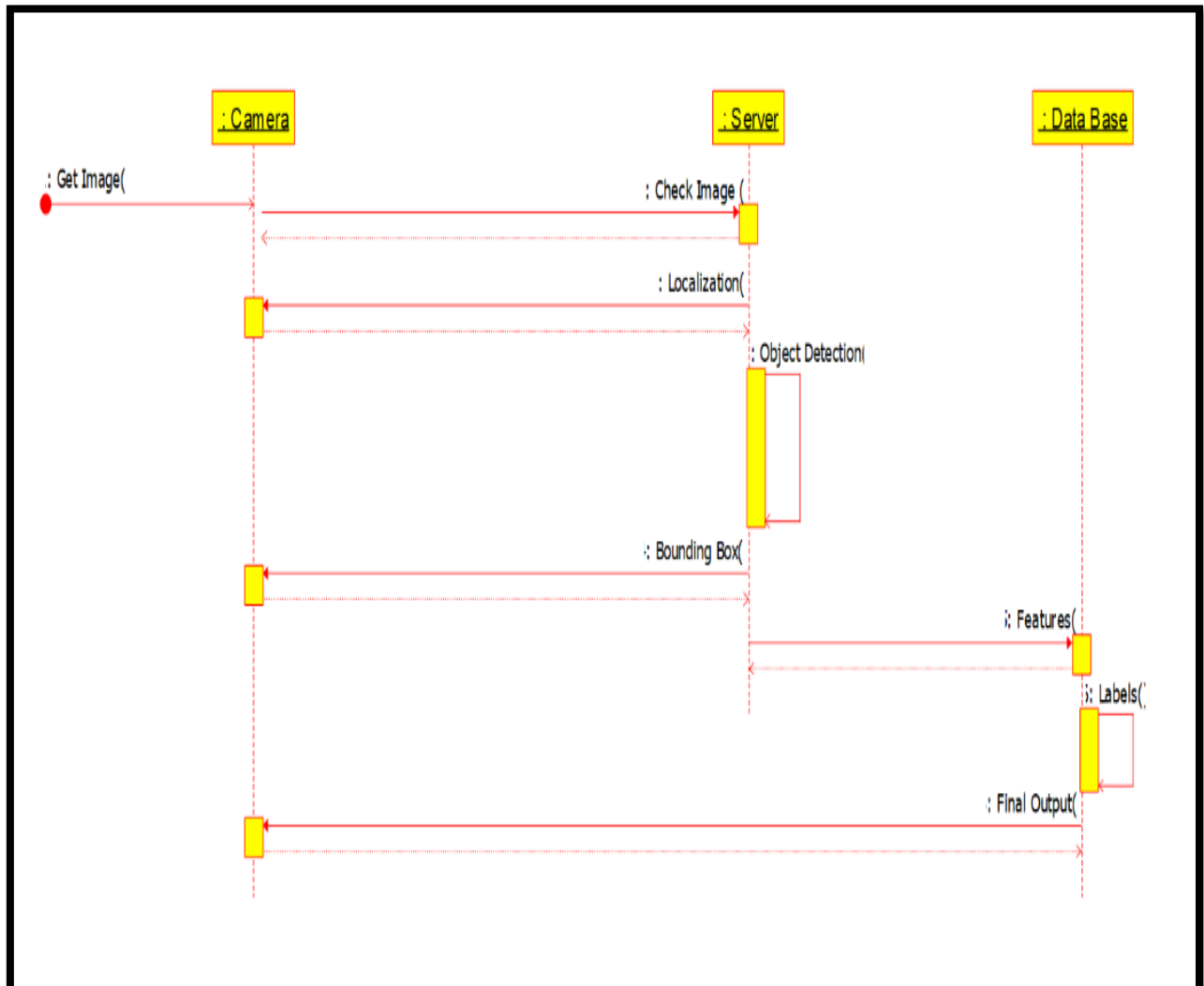


### Create Message

- A message defines a particular communication between Lifelines of an Interaction.
- Create message is a kind of message that represents the instantiation of (target) lifeline.



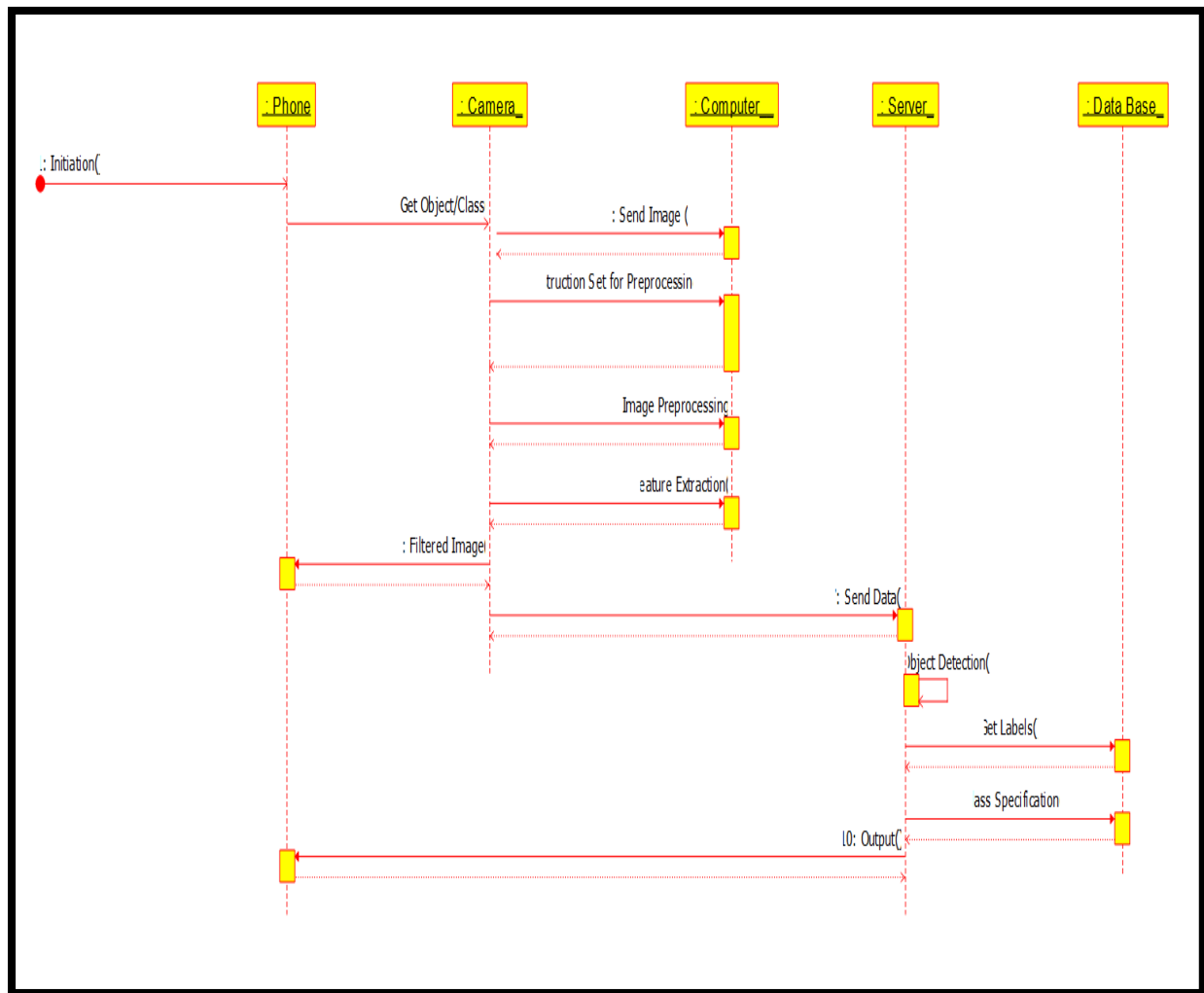
Sequence Diagram 1:



Object Detection methodology

Sequence Diagram for Object Detection Methodology. It tells about the general sequence of flow of data/process in any Object Detection. Where ever it is used.

Sequence Diagram 2:



### Real Time Object Detection -Lens

Sequence Diagram for Real Time Object Detection, this is an in-depth analysis of an Example Case of Object Detection, that is used by Google in it's Google Lens that tells us about the Sequence of processes an Object Detection Model has to undergo to determine whatever object is there in front of it as well as Process the output

### Conclusion:

The Sequence Diagram for the project Object Detection Solution has been made.