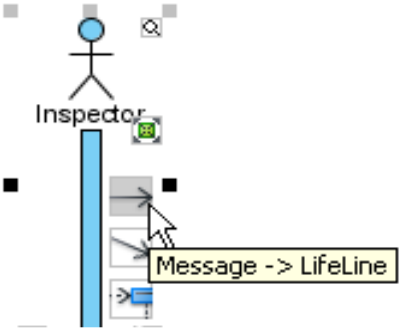
Lab 7: Design Sequence Diagram

- A sequence diagram is used primarily to show the interactions between objects that are represented as lifelines in a sequential order.

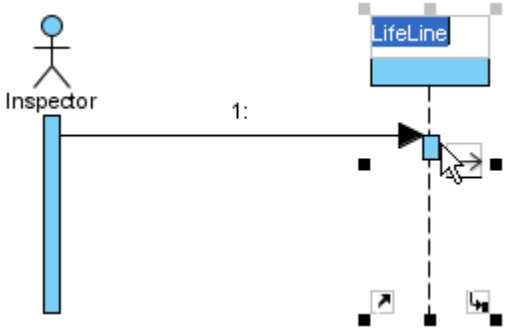
Creating a sequence diagram

1. To create a sequence diagram, select **Diagram** > **New** from the toolbar. In the **New Diagram** window, select **UML Diagrams** > **Sequence Diagram**. Click **OK** to confirm.
2. Enter **name** for the newly created sequence diagram in text field of pop-up box on top left corner.
3. To create actor, click **Actor** on the diagram toolbar and then click on the diagram.

4. To create lifeline, you can click **LifeLine** on the diagram toolbar and then click on the diagram. Alternatively, a much quicker and more efficient way is to use the resource-centric interface. Click on the **Message** > **LifeLine** resource beside an actor/lifeline and drag.



5. Move the mouse to empty space of the diagram and then release the mouse button. A new lifeline will be created and connected to the actor/lifeline with a message.



Student Name: **Student ID:**

System description:
All Phones Shop (APS) is a company that sells three types of phone: *smartphone*, *normal mobile phone*, and *land phone*. APS currently uses a small computer-based system called PoB that manages the payroll of its employees, all payments made by clients, and the financial information of the APS client. PoB is actually a standard, commercial-off-the-shelf application. APS has two types of **clients**: *individual,* and *corporate*.

To **buy** or **sell** a phone, the client must first register if he/she does not have any registration with APS. In order to register, the client has to provide name and address. APS first checks if the client already exists. If not, the system creates a registration, assigns a unique registration number, PoB is advised by APS with the registration number. PoB then opens a financial account of the client with the registration number. All payments of the client are handled by PoB and recorded in the financial account. However, APS keeps only the registration information such as name, address, the registration number, the invoice(s) if the client has bought any item from APS, and the item information if the client sells any item. For all future selling and buying activities with APS, the client has to use the same registration number.
A **registration** can only be removed by corporate clients. In this case, APS checks if there is any unpaid invoice attached with the client registration. PoB is informed if a client registration is removed. The individual client can only create registration but cannot remove any registration. APS does not handle any login operation; it is assumed all login functions are checked before any operation is executed.

To sell an item, the client first selects the type of item that he/she wants to sell. Under smartphone category, there are two sub-categories: *Apple iPhone* and *Samsung Galaxy*. The client then provides model, make, capacity, and color. The client has the option to sell the item either on **auction** (bidding), or on **fixed price sale**. For the auction, a starting price and the end of the auction time is specified; and for the fixed price sale, a price is given. For both types of sale (fixed price and auction), the system assigns an ID to the item, records and attaches the item information with the client registration.

For the catalog entry, APS then automatically creates a display image based on the item description, asks the client (seller) to approve it, and displays the item for sale if approved by the client. Otherwise, the system re-creates another image, and displays it to the client for approval. Once the client approves the display image, APS includes it in its product catalog. Clients can also remove their items after display but before sale; in that case, a **penalty** of 10% of the starting price (for auction) or the fixed price of the item is charged to the client, and an advice on this penalty will be sent to PoB.

Clients can also buy items on auction, or on fixed price sale. For the fixed price items, the client selects item(s). The system creates a shopping basket for the selected item(s). The system saves the shopping basket and attaches this with the client registration. The client then checks out, and each item in the basket is considered "sold" and removed from the product catalog. For auction, clients can bid a price for the selected item before the end of auction time. A bid submitted by a client must be greater than the last offered bid (price). All bids submitted by a client are recorded in the registration of the client. When the auction time ends, the client with the last bid will be notified as the winner. The item is considered "sold" and removed from the product catalog.

For either type of sale, APS prepares an **invoice** (bill) based on the item(s). Once the client confirms the invoice, it is attached with the client registration with the status "unpaid".
For all payments, PoB informs APS with the client registration number and the invoice number. The system finds the invoice and assigns the invoice as "paid". If a client fails to pay to PoB within 7 days, a

penalty of 10% of the price is charged to the client, an advice on this penalty is sent to PoB, and the sale is cancelled by APS.

Task: Develop design sequence diagram (DSD) of the system.
Lab topic(s): Design Sequence diagram (DSD)
Lab objective(s): Understanding of

- Object collaboration in terms of message passing.
- Ordering of message
- Concepts of collection (container/database class)

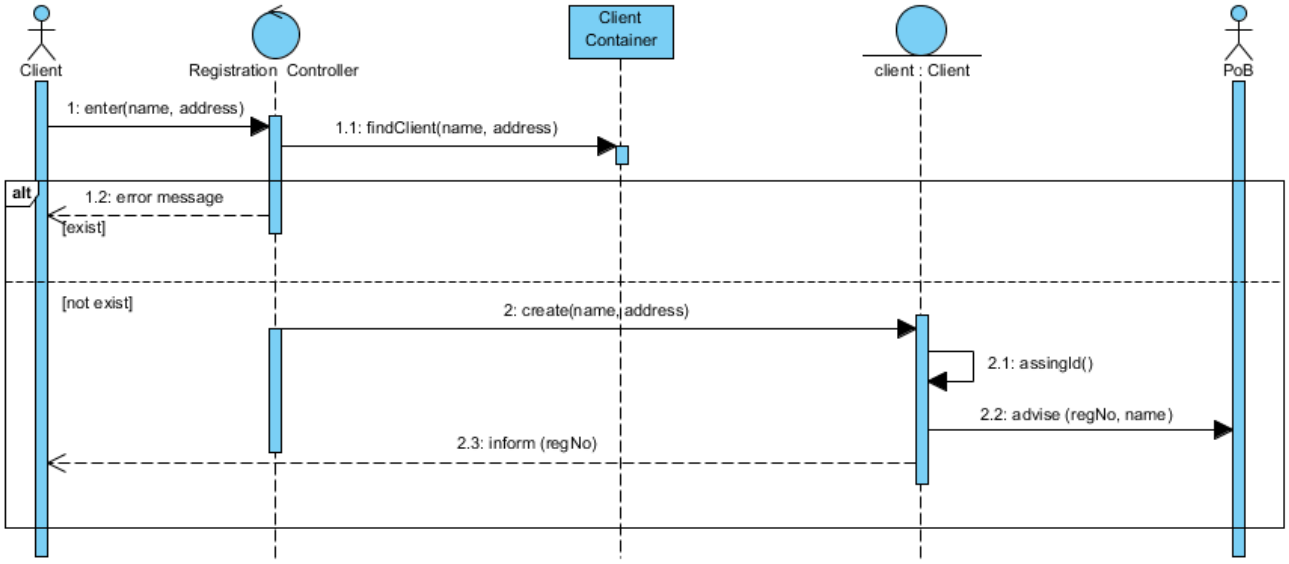
***Lab activities*:**

Activity	Resources and notes	Estimated time
• Students write their name and SID (See first page on where to write these)	• First page of this document	5 minutes
• **Task 1:** The lab instructor will demonstrate the design sequence diagram of the use case "Create registration" (Available in this document). • The students need two documents handy to develop DSD: (i) use case specification and (ii) the design class diagram with methods. The instructor will show the class diagram on the screen. • The instructor will show how the sequence diagram "Create registration" is consistent with its use case specification and the corresponding classes and methods in the design class diagram (available in this document). It is important to note that: o Design sequence diagram must be consistent with the use case specification, o Messages in the DSD must be consistent with the methods in the classes, o Association between corresponding classes must be consistent between the message passing objects.	• This lab document • Data projector • White board • Use case specification of "Create registration" available in this document. • Design Class diagram available in this document.	20 minutes
• **Task 2**: Develop DSD for the two use cases "Sell item" and "Buy on auction" (Group discussion) • **Students must be aware of these**: o Find the actor's actions from the	• Use case of "Sell item" available in this document. • Use case of "Buy on auction" available in this	60 minutes

use case specification. ○ Identify the corresponding system response from the use case specification. ○ Identify the object which will handle the system response. ○ Use the most appropriate message that is consistent with the system response. ○ Make sure that the class of the message receiving object has the corresponding method defined. ○ Finally, the two objects in the message must be represented in the design class diagrams as classes and they have association. • Develop DSD using Visual Paradigm	document. • Design class diagram • System description • Visual Paradigm	
• **Task 3**: Review your DSD by comparing this with the use case specification • Check consistency with your design class diagram. • Update your class diagram if necessary. • Save your DSD and design class diagram	• Visual Paradigm • Use case specification of "Sell item." • Use case specification of "Buy on auction." • Design class diagram	10 minutes
• **Task 4.1:** Copy your DSD "Sell item" and paste it at the end of this document.	• Visual paradigm • This document.	5 minutes
• **Task 4.2:** Copy your DSD "Buy on auction" and paste it at the end of this document.	• Visual paradigm • This document.	5 minutes
• **Task 4.3:** Copy your updated design class diagram and paste it at the end of this document.	• Visual paradigm • This document.	5 minutes
• Submit this document to the lab instructor and leave the lab clean and tidy.		5 minutes

Use case Id: UC**001**	**Create registration**
Brief Description	The client creates a new registration with APS. The system checks if the client already exists, if not it creates a client registration and informs PoB about the new client.
Primary actors	Client, PoB.
Preconditions: 1. The client must not exist.	
Post-conditions: 1. A client registration is created.	
Main Success Scenario: A client registration has been created.	
Actor Action	**System Response**
1. The client provides details	2. Check if the client exits
	3. Create a registration with the client details if the registration does not exist. (See 3.a. for alternative flow)
	4. Assign a unique number to the registration
	5. Advise PoB about the new client.
	6. Inform the client with the registration number.
Alternative flows: 3.a. If the client already exists, inform the client that a new registration cannot be created.	

DSD for "Create registration"



Use case Id: UC**002**	**Sell item**
Brief Description	This use case begins once a client wants to sell an item. The system records the item for sale.
Primary actors	Client.
Trigger	Client selects type of item.
Preconditions: **1.** Client is logged in.	
Post-conditions: **1.** The item is registered for sale.	
Main Success Scenario: The item for sale has been successfully recorded.	
Actor Action	**System Response**
1. The client selects the type of item	
2. The client provides item details	3. Records the item details
	4. Ask to select the sale option
5. The client selects the sale option	6. Get the starting price and the end of sale for the auction (See alternative flow 6.a)
	7. Assign an ID to the item
	8. Attach the item with the client registration
	9. Call use case “Make Catalog Entry”
Alternative flows: 6.a. For the fixed price sale, get the price.	

Task 4.1: Paste your DSD for “Sell item” <<Here>>:

Use case Id: UC**003**	**Buy on auction**
Brief Description	The client can bid for an item on sale. The system finds the highest bidder at the end of sale, and decides the winner.
Primary actors	Client.
Preconditions: 1. The end of auction must be valid	
Post-conditions: 2. A winner was declared.	
Main Success Scenario: The system has successfully declared a winner with highest bid.	
Actor Action	**System Response**
1. The client select the item	
2. Offer a price	3. Records the bid with the client registration (See 4.a)
	4. Find the highest bid at the end of the auction
	5. Inform the client with highest bid
	6. Change the status of the item as "sold"
	7. Remove the item from the catalog
	8. <include: Prepare Invoice>
Alternative flows: 4.a. if the bid is less than the last accepted bid, the bid is rejected, and asks the client to bid again with higher price.	

Task 4.2: Paste your DSD for "Buy on auction" <<Here>>:

Task 4.3: Paste your updated class diagram <<Here>>: