

WIA2002 Software Modelling
Semester 1, 2016/17
Tutorial 7

- 1. Analyse the following text of a use case and identify the classes involved in the use case. For each class, specify the actor and type of analysis classes (boundary, entity or control) which the class belongs to.**

The customer clicks the Register button on the Login page. The system displays the Registration page. The customer enters his or her personal details and then clicks the Submit button. The system validates the username and email account against the persistent Account data, and the returns the customer to the Login Page.

- 2. Analyse the following scenario of a use case and identify the classes involved in the use case. For each class, specify the actor and type of analysis classes (boundary, entity or control) which the class belongs to. Draw a sequence diagram for this scenario (include entity, boundary and control classes).**

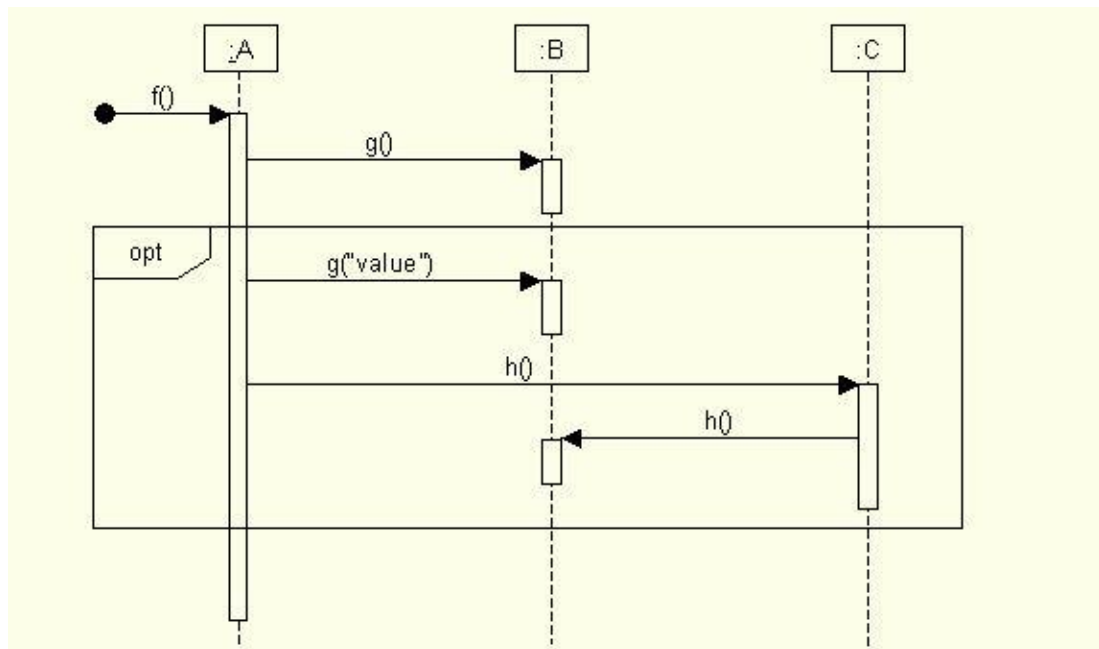
The Customer specifies an author on the Search Page and then press the Search button. The system validates the Customer's search criteria. The system searches the Catalogue for books associated with the specified author. When the search is complete, the system displays the search results on the Search Results Page. If the Customer did not enter the name of an author before pressing the Search button, the system displays an error message to that effect and prompts the Customer to re-enter an author name.

- 3. You are asked to develop a library information system (LIS) for a public library. The LIS will be used to handle book loan and return procedures. Draw a design sequence diagram for a book return scenario in which the book is overdue. You need to include entity, boundary and control classes in this diagram.**

Return Book Scenario:

Books are returned through a collection box at the library entrance. When the Library assistants check in the returned books with the LIS administrator terminals, he/she has to scan the barcode of the book. The system displays the book details and the due date of the book loan. If a book is returned late, an overdue fine is calculated and a new fine record is created. Overdue fine is \$1 per day. Once the book return is completed, the system will display a message "Check in Finish".

4. Is there anything wrong with or missing from the following sequence diagram? If so, what is wrong or missing? How to improve the diagram?



5. Consider a Sale Process Scenario for a cashier processing sale activities in a supermarket. The cashier is required to process any purchasing (sale) activity by the customer. The customer will firstly put their goods on the cash counter. The cashier will then enter the item ID (could be the barcode) and its quantity, so that the system can give the description of the item and total price. The process is repeated until all items are scanned and recorded as sale. The process will end afterwards, and will require the customer to pay the amount indicated.

The Use Case has been thoroughly analyzed and identified below:

- Cashier starts a new sale.
- Cashier enters item identifier (should be barcode of item).
- System records sale item and presents item description, price and running total.
- Cashier repeats steps 2 to 3 until all items processed.
- System presents total.
- Cashier informs the customer the total and asks for payment.
- Customer pays and System handles payment.

In this question, you need to:

- Translate the flows into corresponding system events (Actor input and System response).
- Model a **system-level sequence diagram** based on the flows.
- Consider that the cashier is required to process all sale items until completed. Refine your system-level sequence diagram with a **loop fragment**.

6. Draw an UML sequence diagram to express the dynamic behavior of the following program.

```
import java.util.Vector;

public class Driver {
    private StringContainer b = null;

    public static void main(String[] args) {
        Driver d = new Driver();
        d.run();
    }

    public void run() {
        b = new StringContainer();
        b.add("One");
        b.add("Two");
        b.remove("One");
    }
}

class StringContainer {
    private Vector v = null;

    public void add(String s) {
        init();
        v.add(s);
    }

    public boolean remove(String s) {
        init();
        return v.remove(s);
    }

    private void init() {
        if (v == null)
            v = new Vector();
    }
}
```