ITSS SOFTWARE DEVELOPMENT
Lab 02-Use case Specification

Lecturer: NGUYEN Thi Thu Trang, trangntt@soict.hust.edu.vn

1. SUBMISSION GUIDELINES

You are required to push all your work to the valid GitHub repository complying with the naming convention:

"<FacebookGroupName>-<StudentID>.<StudentName>".

For this lab, you have to turn in your work twice before the following deadlines:

- **Right after class**: Push all the work you have done during class time to Github.
- **10 PM the day after the class**: Create a branch named "***release/lab02***" in your GitHub repository and push the full submission for this lab, including in-class tasks and homework assignments, to this branch.

2. IN-CLASS TASKS

In this lab, we continue with the requirement modeling process and practice with flows of events, activity diagrams, and the use case (UC) specification for the Case Study AIMS.

You are asked to work individually for this section, and then put all your file(s) and directories to a parent directory, namely "***Requirement Analysis***". After that, push your commit to your individual repository before the announced deadline. Remember to submit astah file(s).

2.1. USE CASE SPECIFICATION

In use case models, a use case describes a **flow of events** which is *performed by the software* and *yields an observable result of value to a particular actor*.

In this subsection, we would use the two use cases, UC "Place Order" and UC "Pay Order", to demonstrate how we can make a flow of events or use case specification. Please note that there does not exist the best answer, but a reasonable answer.

2.1.1. Use Case Specification for UC "Pay Order"

In this part, you are asked to fill in the provided template for the use case specification of UC "Pay Order" by using the following solution. When you finish the task of this part, please export your work to a PDF file, namely "*Use case specification – Pay Order*".

This use case describes the interactions between the AIMS software with the customer and Interbank when the customer desires to pay an order. The precondition of this use case is that the AIMS software has calculated the total amount of money that the customer has to pay. On the other hand, there is no need for the specification of output data or the postcondition. An incomplete example of the basic flow of events is listed as follows.

Step 1. The AIMS software displays the payment screen
Step 2. The customer enters credit card info and confirm to pay order
Step 3. The AIMS software check if the credit card info is in the correct format
Step 4. The AIMS software asks the Interbank to pay order
Step 5. The Interbank processes the payment transaction
Step 6. The AIMS software saves the payment transaction.

The alternative flows of events of this use case are illustrated in the following table.

No	Location	Condition	Action	Resume location
1.	At Step 4	If the card info has wrong format	▪ The AIMS software notifies that the card info has wrong format	At Step 1
2.	At Step 5	If the card info is invalid	▪ The AIMS software notifies that the card info is valid	At Step 1
3.	At Step 5	If the balance is not enough	▪ The AIMS software notifies that the balance is not enough	At Step 1

The input data of payment form is shown as follows.

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Card holder name		Yes		DO MINH HIEU
2.	Card number		Yes		1234 5678 9123 4567

3.	Expiration date		Yes	Consist of month and last 2 digits of year only	01/23
4.	Security code		Yes		123

Template of Use Case Specification is shown as below.

Use Case "Name of use case"

1. Use case code

UC00X

2. Brief Description

This use case describes the interaction between <actor(s)> and <name_of_the_system> when <actor(s)> wish(es) to ...

3. Actors

3.1 Name of Actor 1

4. Preconditions

5. Basic Flow of Events

1. The actor(s) …

i. The software displays … (see Table T).

6. Alternative flows

Table N-Alternative flows of events for UC Place order

No	Location	Condition	Action	Resume location
1.	At Step S	If …	▪ Action 1	Resumes at Step Q
2.	At Step O	If …	▪ Action 2	Use case ends

7. Input data

Table A-Input data of …

No	Data fields	Description	Mandatory	Valid condition	Example
1.					

8. Output data

Table B-Output data of …

No	Data fields	Description	Display format	Example
1.				

9. Postconditions

2.1.2. Use Case Specification for UC "Place Order"

In the AIMS Project, UC "Place Order" describes the interaction between customers and AIMS software when the customer wishes to place an order. Naturally, from the use case diagram, we describe how the use case starts and ends to gain the purpose of a use case, and we may think of a basic flow of the events for UC "Place Order" as follows.

Step 1. The customer requests to place an order on the view cart screen
Step 2. The AIMS software checks the availability of products in the cart
Step 3. The AIMS software displays the form of delivery information
Step 4. The customer enters and submits delivery information
Step 5. The AIMS software checks the validity of delivery information
Step 6. The AIMS software calculates shipping fees
Step 7. The AIMS software displays the invoice
Step 8. The customer confirms to place order
Step 9. The AIMS software calls UC "Pay order"
Step 10. The AIMS software save order
Step 11. The AIMS software makes the cart empty
Step 12. The AIMS software displays the successful order notification.

We specify what happens when, for each action. Remember this text will be used to identify test cases. However, things hardly ever go as planned. Taking the above flow as our initial basic flow, we would try to analyze and improve our flow of events for UC "Place Order".

Question: Should we trust any input to an action/event?

It depends on the question or the request and the way it has been asked or responded. If the response must comply with a rule or a regulation, we must question the validity of the input. In the case of humans, they tend to follow the given instructions in an incorrect manner even after several trials. In cases of non-human actors, there are a lot of factors

that heavily affect the input from other systems, e.g., noise in transmission. Hence, we need to give them the chance to try repeatedly until their input is valid or within an acceptable limitation to be at least. However, there are cases in which we must take the input for granted since it is unnecessary or beyond the control of the system. To illustrate, in the form of delivery information, we should not check if the user has filled the real name of the receiver in the field for receiver name, yet we must not let the user left it blank.

Consequently, after each of Step 1 and Step 5, the AIMS software should validate the input of the customer and ask he or she to redo the step until the input is valid. Just then the software proceeds to next step.

<Customer may also go back to previous step to update the information>

Question: What data is exchanged between actor and use case and between use case and use case?

In this use case, there are two inputs we need from the customer up to now: information of media in the cart (for displaying cart, order confirmation, shipping fees, and payment) and delivery information (for shipping and shipping fees). Some or all attributes of these information may play critical roles in input validation, so we need to specify the attributes of the input. To illustrate, the input data of delivery information may include these data fields:

Table 1- Input data of delivery information

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Receiver Name		Yes		Do Minh Hieu
2.	Phone Number		Yes		0987654321
3.	Province	Choose from a list	Yes		Hanoi
4.	Address		Yes		12, 34 Alley of Tran Thai Tong street, Cau Giay district
5.	Shipping instructions		No		

We also need to specify the output to the actor(s). For instance, the output data when displaying the invoice or the cart is shown in the following tables (the rows with green shading are repeated for all media products in the cart/invoice).

Table 2-Output data of displaying invoice

No	Data fields	Description	Display format	Example
1.	Title	Title of a media product		DVD Phim Vượt ngục
2.	Price	Price of the corresponding media product	▪ Comma for thousands separator ▪ Positive integer ▪ Right alignment	123,000
3.	Quantity	Quantity of the corresponding media	▪ Positive integer ▪ Right alignment	2
4.	Amount	Total money of the corresponding media	▪ Comma for thousands separator ▪ Positive integer ▪ Right alignment	246,000
5.	Subtotal Before VAT	Total price of products in the cart before VAT	▪ Comma for thousands separator ▪ Positive integer ▪ Right alignment	2,106,000
6.	Subtotal	Total price of products in the cart with VAT		2,316,600
7.	Shipping fees			30,000
8.	Total	Sum of subtotal and shipping fees		2,346,600
9.	Currency			VND
10.	Name			Do Minh Hieu
11.	Phone number			0987654321
12.	Province	Choose from a list		Hanoi
13.	Address			12, 34 Alley of Tran Thai Tong street, Cau Giay district
14.	Shipping instructions			

Table 3-Output data of displaying cart

No	Data fields	Description	Display format	Example
1.	Title	Title of a media product		CD Em về tinh khôi – Hà Trần
2.	Price	Price of the corresponding media product	▪ Comma for thousands separator ▪ Positive integer ▪ Right alignment	120,000
3.	Quantity	Quantity of the corresponding media	▪ Positive integer ▪ Right alignment	2
4.	Amount	Total money of the corresponding media	▪ Positive Integer ▪ Right alignment	240,000
5.	Subtotal Before VAT	Total price of products in the cart before VAT	▪ Comma for thousands separator ▪ Positive integer ▪ Right alignment	2,106,000
6.	Subtotal	Total price of products in the cart after VAT		2,316,600
7.	Currency			VND

Note that we do not describe the details of the user interface unless it is necessary to understand the behavior of the system. Specifying user interface details too early will limit design options.

Now, we can finally validate the data. For the list of media in the cart, we need to check if a media is out-of-stock. For delivery information, we need to check if a mandatory field is left blank and the validity of the phone number field. Thus, we need to insert at least two more events into the flow so as to validate the two corresponding inputs.

Additionally, depending on how you design, you might want to check the input from UC "Pay order" at Step 10.

After validation, in case there is an exception, the flow cannot continue normally. Consequently, we need alternative flows or sub-flows for the next events in these cases. For instance, the sub-flows for UC "Place Order" is shown as follows.

Table 4-Alternative flow of events for UC "Place Order"

No	Location	Condition	Action	Resume location
1.	At Step 2	If there is media of which quantity in the stock is less than the ordered quantity	▪ The AIMS software asks the customer to update the cart. ▪ The customer updates the cart.	Resumes at Step 2
2.	At Step 5	If a mandatory field is left bank	▪ The AIMS software asks the customer to fill all the mandatory blank.	Resumes at Step 3
3.	At Step 5	If the phone number is invalid	▪ The AIMS software asks the customer to enter a valid phone number.	Resumes at Step 3

The last questions are *what we should save and when we save it*.

By saving the data, we can save a lot of time and effort for us, the system, and the users. To illustrate, the customer cannot finish placing order for some reason. Thus, we can save some information for later such as the list of media in the cart, so that the customer does not have to add them to the cart again.

Finally, we may provide the pre-condition and the post-condition. For example, the pre-condition for UC "Place Order" can be "there is an active network connection to the Internet." A post-condition can be "the logs have been updated accordingly" in the case of a failure condition.

For this part, given the above suggestion, you are asked to make a use case specification for UC "Place Order" by using the template as illustrated in subsection 2.1. Remember to validate data and save information if need be. When you finish the task of this part, please export your work to a PDF file, namely "*Use case specification – Place Order*". Then put both files in the directory "***Use case specification***".

2.2. ACTIVITY DIAGRAM

2.2.1. Activity diagram with Astah

Please see the following link to know how to make an activity diagram with Astah.

https://astah.net/support/astah-pro/user-guide/activity-diagram/

2.2.2. Activity diagram of UC "Pay Order"

In this part, you are asked to redraw the following activity diagram for UC "Pay Order" by using Astah.

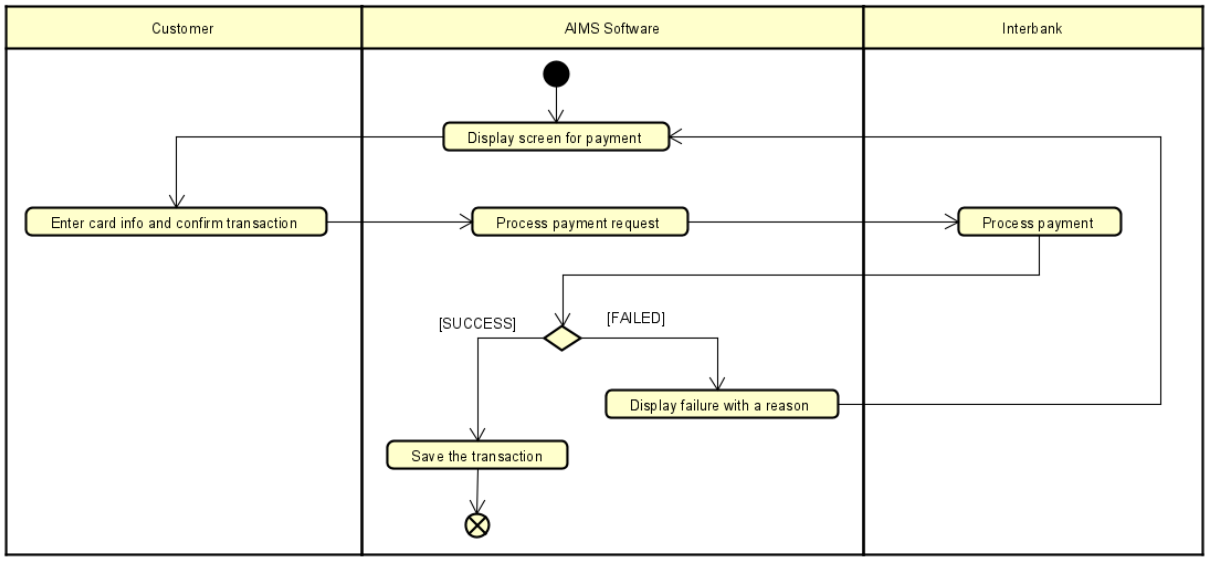

Figure 1-Activity diagram of UC "Pay Order"

When you finish the task of this part, please export your work to a PNG file, namely "*Activity Diagram – Pay order*", and save your work in a directory, namely "***Activity diagrams***".

2.2.3. Activity diagram(s) of UC "Place Order"

As can be seen, the incomplete flow of events in subsection 2.1.2 is clearly represented in the activity diagram in Figure 1. We can convert an activity diagram into flows of events with very little effort.

In this part, you are asked to draw activity diagram(s) of UC "Place Order" by using Astah. When you finish the task of this part, please export your work to a PNG file, namely "*Activity Diagram – Place order*" and save your work in a directory, namely "***Activity diagrams***".

3. HOMEWORK ASSIGNMENTS

3.1. USE CASE SPECIFICATION FOR "PLACE RUSH ORDER"

For this assignment, you are asked to update the use case specification for UC "Place order" and/or ***to make another use case specification by using the provided template*** (e.g., you can model the relationship between UC "Place Rush Order" and UC "Place Order" as an extension) **for the additional UC "Place Rush Order"**.

When you finish, please export your work to a PDF file, namely "Use case specification – Place Order with Place Rush Order." Then put your work and the exported file in the directory "Use case specification".

3.2. ACTIVITY DIAGRAM(S) OF UC "PLACE ORDER" WITH "PLACE RUSH ORDER"

For this assignment, you are asked to update the activity diagram for UC "Place order" with the option "Place rush order". If you consider "Place Rush Order" as an extension use case, you need two activity diagrams: one for UC "Place Order", and another one for UC "Place Rush Order". Otherwise, one activity diagram is enough.

When you finish, please export your work to a PNG file, namely "*Activity Diagram – Place order with Place rush order*" (or 2 separate activity diagrams), and save your work in a directory, namely "**Activity diagrams**".