ITSS SOFTWARE DEVELOPMENT/ SOFTWARE DESIGN AND CONSTRUCTION Lab 04-Sequence Diagram

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

1. SUBMISSION GUIDELINE

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

"<MSTeamName>-<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 before the next class**: Create a branch named "***release/lab04***" in your GitHub repository and push the full submission for this lab, including in-class tasks and homework assignments, to this branch. Remember to export your diagrams to PNG files and push them with .astah files to GitHub.

2. ARCHITECTURAL DESIGN AND SEQUENCE DIAGRAM

In this section, we get started with the architectural design process by drawing sequence diagrams for the Case Study.

You are asked to work individually for this section, and then put all your files (including both .astah files and exported PNG files) and sub-directories in a parent directory, namely "***Architectural Design***". After that, push your commit to your individual repository before the announced deadline. Remember to submit astah file(s) also.

We will use our Software Requirement Analysis (SRS) in the previous lab as the input for the architecture design process.

In this subsection, you would get familiar with the components of a sequence diagram in Astah. Please see the following links to know how to make a sequence diagram with Astah.

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

https://www.youtube.com/embed/Qi2CsTY4LSk

Some small tips:

https://astahblog.com/2015/10/15/search-models-in-diagram/

https://astahblog.com/2015/10/28/hide-sequence-message-number/

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

3. SEQUENCE DIAGRAM FOR UC "PLACE ORDER"

This subsection demonstrates how to create an interaction diagram for UC "Place Order" step by step. In the end of this subsection, we achieve a sequence diagram and a communication diagram for UC "Place Order", which are used to create the analysis class diagram in the next lab.

3.1. FIND ANALYSIS CLASSES

The steps to analyze classes for UC "Place Order", i.e., the steps to find analysis classes from use-case behavior (flow of events), are illustrated as follows.

Step 1. Find boundary classes:

a) User interface classes: CartScreen, DeliveryForm, SuccessOrderScreen
b) System/device interface classes: None

Step 2. Find entity classes: Cart, Product, DeliveryInfo, Order, PaymentTransaction

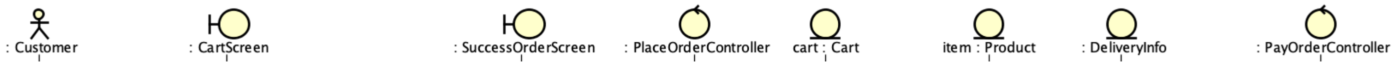
Step 3. Find control classes: PlaceOrderController

3.2. DISTRIBUTE USE-CASE BEHAVIOUR TO CLASSES

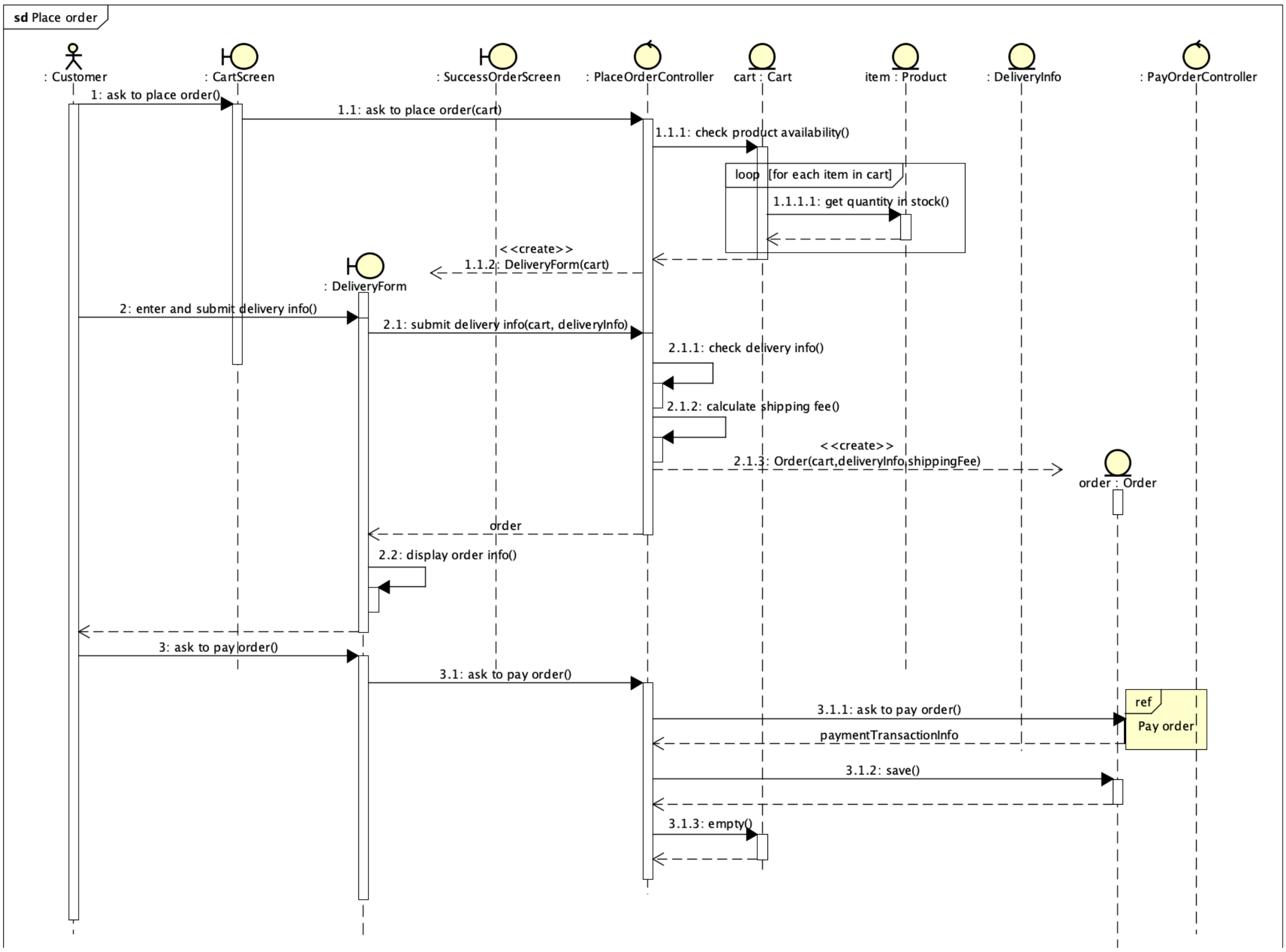
In this section, we design sequence diagrams to allocate responsibilities to classes and model analysis class interactions. All below images/diagrams in this guideline are examples, it may be lack or inaccurate with the prolem statement. You need to check again the statement and your use case specification to ensure that your sequence diagram(s) are consitent with your use case model.

Step 1. Create a new sequence diagram, name "Place Order"

Step 2. Drag all the classes and related actor(s) from the structure tree and drop it on the newly created diagram.



Step 3. Allocating responsibilities to classes


sd Place order
: Customer
: CartScreen
: SuccessOrderScreen
: PlaceOrderController
cart : Cart
item : Product
: DeliveryInfo
: PayOrderController
1: ask to place order()
1.1: ask to place order(cart)
1.1.1: check product availability()
loop [for each item in cart]
1.1.1.1: get quantity in stock()
<<create>>
1.1.2: DeliveryForm(cart)
: DeliveryForm
2: enter and submit delivery info()
2.1: submit delivery info(cart, deliveryInfo)
2.1.1: check delivery info()
2.1.2: calculate shipping fee()
<<create>>
2.1.3: Order(cart,deliveryInfo,shippingFee)
order : Order
order
2.2: display order info()
3: ask to pay order()
3.1: ask to pay order()
3.1.1: ask to pay order()
ref
Pay order
paymentTransactionInfo
3.1.2: save()
3.1.3: empty()

Step 4. Export to PNG file and submit it to GitHub.

4. SEQUENCE DIAGRAM FOR UC "PAY ORDER"

This subsection demonstrates how to create an architectural design for UC "Pay Order" step by step. At the end of this subsection, we achieve a sequence diagram and a communication diagram for UC "Pay Order", which is used to create the analysis class diagram in the next lab.

4.1. FIND ANALYSIS CLASSES

The steps to analyze classes for UC "Pay Order", i.e., the steps to find classes from use-case behavior, are illustrated as follows:

Step 1. Find boundary classes:

a) User interface classes: InvoiceScreen

b) System/device interface classes: VNPayBoundary

Step 2. Find entity classes: Invoice, TransactionInfo

Step 3. Find control classes: PayOrderController

4.2. DISTRIBUTE USE-CASE BEHAVIOUR TO CLASSES

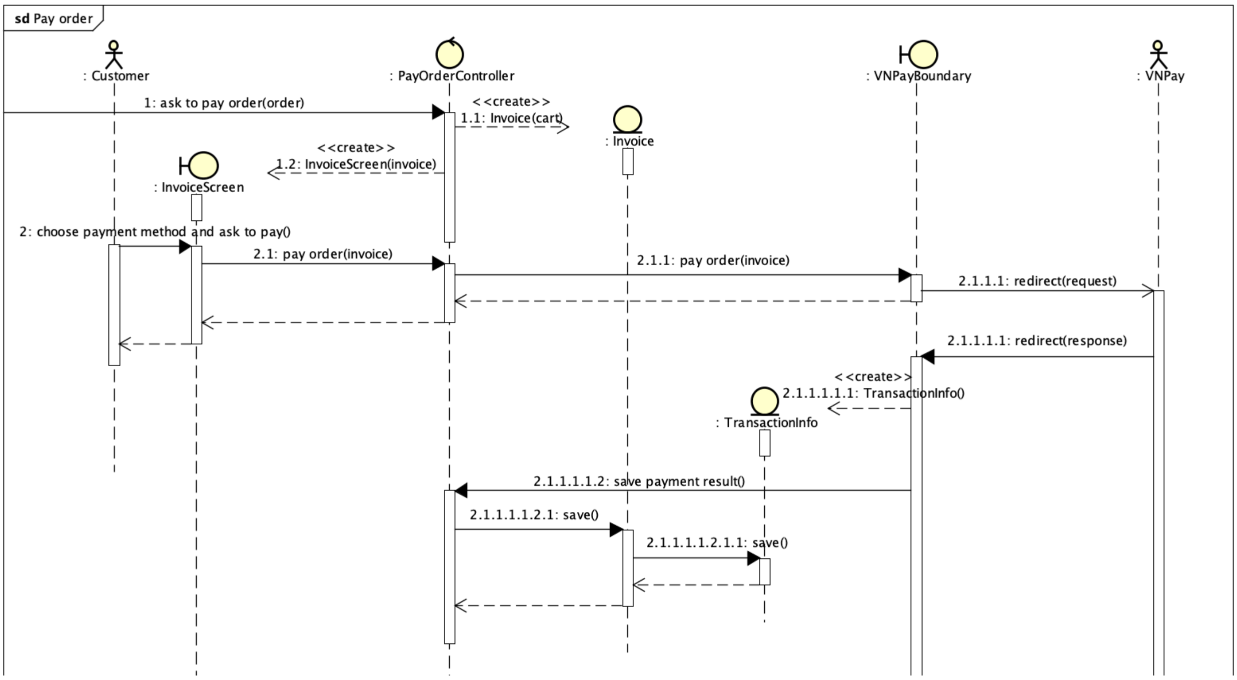
We use interaction diagram(s), i.e., sequence diagram and/or communication diagram, to allocate responsibilities to classes and model analysis class interactions.

Sequence diagram for UC "Pay Order"

Step 1. Create a new sequence diagram.

Step 2. Drag all the classes and related actor(s) from the structure tree and drop it on the newly created diagram

Step 3. Allocating responsibilities to classes


sd Pay order
: Customer
: PayOrderController
: VNPayBoundary
: VNPay
1: ask to pay order(order)
<<create>>
1.1: Invoice(cart)
: Invoice
<<create>>
1.2: InvoiceScreen(invoice)
: InvoiceScreen
2: choose payment method and ask to pay()
2.1: pay order(invoice)
2.1.1: pay order(invoice)
2.1.1.1: redirect(request)
2.1.1.1.1: redirect(response)
<<create>>
2.1.1.1.1.1: TransactionInfo()
: TransactionInfo
2.1.1.1.1.2: save payment result()
2.1.1.1.1.2.1: save()
2.1.1.1.1.2.1.1: save()

Step 4. Save your work.

5. SEQUENCE DIAGRAM FOR UC "PLACE RUSH ORDER"

> **In this assignment, you are asked to design the sequence diagram(s) for UC "Place Rush Order". Please remember to modify the sequence diagram(s) for UC "Place Order" with this new use case.**

In case you model the relationship between UC "Place Rush Order" and UC "Place Order" as an extension, think of where and when the extension use case is inserted in the base use case (i.e., at which message of which class under which conditions the extension use case starts). Then use the event flow in SRS to create sequence diagrams.

If you consider flows of events for "Place rush order" is in the UC "Place order", you can design them into the sequence diagram of UC "Place order" and then you don't need to do this task.

When you complete the assignment, please export your diagram to PNG file(s) and push it with .astah file(s) to GitHub.