# Systems Analysis and Testing

# Phase 2 – Individual Submission

**[D22125495]**

**[Yuanshuo Du]**

# Table of Contents

Table of Content

# 1. Test Case

## 1.1. Acceptance testing

Acceptance testing is a type of testing where the software's behavior is compared with the requirements of the end user to ensure their satisfaction. If the acceptance testing is completed successfully, it will result in the customer accepting the software and paying for it.

This type of testing is conducted after the software has passed the development testing phase. It is not always necessary to have a formal acceptance test, especially if the end users have been involved throughout the development lifecycle and are satisfied with the other system tests.

The acceptance test case process involves several steps to ensure that the software meets the necessary criteria for acceptance by the customer. The first process involves defining the acceptance criteria by identifying the key requirements and expectations of the customer. A test plan is then created to outline the approach, scope, and resources required for acceptance testing. The acceptance test cases are executed to validate the software against the acceptance criteria, and the results are documented. If there are any discrepancies between the expected and actual results, negotiation may be required to address any issues or concerns. Based on the results of the acceptance testing and any negotiations, the customer can choose to accept or reject the software.

## 1.2. Acceptance Criteria

Possible Acceptance Criteria for the "Place an Order" Functionality of Irish Food Ordering Software:

Shopping cart（place an order）

- User should be able to view selected desired item in cart.
- Users should be able to customize their order by adding or removing any selected items.
- Users should be able to specify a shipping address and contact information.
- The system should notify the user if the restaurant's items are unavailable or the restaurant is closed

Order system and Payment system

- Users should be able to select the desired payment method.
- The user should receive an order confirmation including details such as estimated delivery time, total cost and order number.
- Users should be able to view order history, including details such as order date, items ordered, and cost.
- If there is any problem with the order, such as delivery delay or payment failure, the system should notify the user.

Since I have chosen the shopping cart place an order system for this asignment, the order system and payment system are not involved for the time being. So the acceptance criteria for place an order here will probably be the four listed above.

## 1.3. Run acceptance tests

To successfully drive and run acceptance tests, there are four important steps to follow. The first step is to select what needs to be tested by analyzing the completeness of requirements, evaluating the cohesion of the design, and testing the source code implementation. After that, it is necessary to decide how the testing will be done. This can involve review or code inspection, proofs using Design by Contract, and black-box or white-box testing. Additionally, it is essential to select an integration testing strategy, such as big bang, bottom up, top down, or sandwich.

Once the testing approach has been decided, the next step is to develop test cases, which involve a set of test data or situations to exercise the unit being tested or measure a particular attribute. Lastly, it is crucial to create a test oracle, which comprises expected results for a set of test cases. It is important to create the test oracle before actual testing takes place to have a baseline for comparison of actual results.

Before running acceptance tests, it is important to clearly define the objective and intention of the testing. This involves identifying the specific outcomes that the tests are designed to achieve and the criteria that will be used to evaluate the success or failure of each test case. Additionally, determining the appropriate format for the tests, such as using a specific testing framework or tool, is also crucial to ensure that the testing process is efficient and effective. Only by carefully considering these factors can organizations be confident that their acceptance testing efforts will yield meaningful results.

## 1.4. Objective and intention

The test objective and intention is to verify the functionality of the "Place Order" feature in an online food ordering system. The test cases cover various scenarios, such as adding items to the cart, applying coupons, selecting delivery options, entering delivery address and phone number, selecting payment method, and handling errors. The post-conditions ensure that the system has recorded the order details correctly and connected with the payment system successfully. By executing these tests, the system's quality attributes such as usability, reliability, and availability can be evaluated.

## 1.5. Format

The appropriate format for this test is a test case document or a test script. The document includes the test objective, the steps to be taken, the expected system response, and a pass/fail column to record the actual results. It also contains any other relevant information related to the test such as pre-conditions, data setup, post-conditions, and any necessary notes. The test case document will be organized in a logical sequence to make it easy to follow and understand. Before execution, the document should be reviewed and approved by all relevant stakeholders to ensure that all aspects of the testing have been properly covered.

## 1.6. Addressed the creation of data and its use

The data is primarily created and used in the following ways:

- Shopping cart items and their quantities are displayed to the customer (Step 1)
- The customer checks the items and quantities to ensure they are correct (Step 2)
- The customer clicks "delete" to remove an item (Step 3)
- The customer clicks "add items" to view the menu (Step 4)
- The customer selects an item and its quantity to add to the cart (Step 5)
- The customer modifies the quantity of an item in the cart (Step 7)
- The customer applies a valid coupon to the cart (Step 10)
- The customer adds a new delivery address and phone number (Steps 20-22)
- The customer selects a delivery option (Steps 26-27)
- The customer proceeds to pay (Step 29)

To create the necessary data for testing these steps, a test database or mock database could be created with the following information:

- A list of valid and invalid menu items and their prices
- A list of valid and invalid coupons and their discounts
- A list of valid and invalid delivery addresses and phone numbers
- A list of valid and invalid delivery options and their fees

This data can be used to simulate various scenarios and test cases, such as testing the system's ability to handle out-of-stock items, invalid coupons, and invalid delivery addresses. Additionally, mock data can be used to test the system's ability to handle different types of inputs and to ensure that the system records and displays the correct information in the database.

## 1.7. Test Case form – place an order

| | |
|---|---|
| **Test Case #:** 1<br>**System:** shopping cart system<br>**Designed by:** Yuanshuo Du<br>**Executed by:** | **Test Case Name:** place an order<br>**Subsystem:** payment system<br>**Design Date:** 28/04/2023<br>**Execution Date:** |

**Short Description:** This test case verifies the functionality of the "Place an Order" feature in the shopping cart system.

**Pre-conditions:**

- The customer has added items to their shopping cart.
- The customer has logged in to their account.
- The customer has an account on the online food ordering system.

**Data setup:**

Items:

- Item 1: Steakhouse Stack, Quantity: 2, Price: $5 per each
- Item 2: 9Chicken Nuggets, Quantity: 1, Price: $6 per each
- Item 3: Spice Wrap, Quantity: 1, Price: $4 per each

Coupon: discount 10%

Coupon Num: 789789789 (valid)

Invitation sharing code: abc123

Delivery Address: 123 Main St, Dublin 1, Ireland

Delivery fees: $3

Priority delivery: $2

Phone Number: 555-1234

| Step | Action | Expected System Response | Pass/ Fail | Comment |
|---|---|---|---|---|
| 1 | Customer enter the shopping cart to prepare "Place an Order". | System displays the shopping cart items and their quantities to the customer: Steakhouse Stack price in total €10 , Quantity: 2; 9Chicken Nuggets, price in total €6, Quantity: 1 | | |
| 2 | Customer checks the items and quantities to ensure they are correct | | | |
| | **Test customers delete the items** | | | |
| 3 | Customer clicks “delete” button after the item to delete the item “9Chicken Nuggets” | System romove the item “9Chicken Nuggets”and display the cart: Steakhouse Stack price in total €10 , Quantity: 2 | | |

|  |  |  |  |  |
|---|---|---|---|---|
|  | **Test customers add the items** |  |  |  |
| 4 | Customer clicks “add items”button | System return the “view menu” page |  |  |
| 5 | Customer click the item “Spice Wrap” and default select 1 quantity | System add the item “Spice Wrap”and the quantity 1. |  |  |
| 6 | Customer clicks “add to the cart” | System displays the cart: Steakhouse Stack price in total €10 , Quantity: 2; Spice Wrap, price in total €4, Quantity: 1 |  |  |
|  | **Test customers modify the quantity** |  |  |  |
| 7 | Customer modifies the item Spice Wrap’ quantity and select 2 | System modifies the item Spice Wrap’ quantity and display the cart: Steakhouse Stack price in total €10 , Quantity: 2; Spice Wrap, price in total €8, Quantity: 2 |  |  |
|  | **Test customers use the valid coupon** |  |  |  |
| 8 | Customer click the “checkout” button from cart to checkout page | System goes to checkout page |  |  |
| 9 | Customer clicks the button “add a promo” to go to “coupon page” | System checks the coupon is valid and display valid coupon. Valid coupons are marked in color, expired or invalid coupons are marked in gray |  |  |
| 10 | Customer select the valid coupon “discount 10% 789789789 ” | System displays selected coupon. |  |  |
| 11 | Customer clicks “add the coupon” | System return to cart page and displays the price : Steakhouse Stack price in total €10 , Quantity: 2; Spice Wrap, price in total €8, Quantity: 2. Total price: 18-10% = €16.2 |  |  |
|  | **Test customers invite friends successfully** |  |  |  |
| 12 |  | After step 9, if there is no valid coupon, system displays “invite friends” button |  |  |
| 13 | Customer clicks “invite friends” | Systems instruct to another page, system displays share code, and “email” “whatsapp” “more options” buttons |  |  |
| 14 | Customer copies the code | System copies the code to the clipboard |  |  |
| 15 | Customer clicks the share buttons | System instructs to the corresponding platform |  |  |
| 16 | Customer clicks “back”button to “coupon page” | The system checks whether the invitation is successful, and if successful, generates and displays a 10% coupon; the page goes to step 10 |  |  |
|  | **Test customers invite friends unsuccessfully** |  |  |  |

| | | | | |
|---|---|---|---|---|
| 17 | After step 15, Customer clicks “back”button to “coupon page” | The system checks the invitation is unsuccessful, system still in the “coupon page” and there is no valid coupon, and system still displays “invite friends” button. | | |
| | **Test customers use invalid coupon** | | | |
| 18 | Customer clicks the button “add a promo” to go to “coupon page” | System checks the coupon is invalid. The invalid coupons are marked in gray, customers couldn’t select. | | |
| | **Test customers have the delivery address and phone number** | | | |
| 19 | Customer clicks “address” button to go to address page | System checks the database and displays the existing address “123 Main St, Dublin 1, Ireland” and phone number “555-1234” | | |
| 20 | Customer clicks the existing address and phone number | System records the new delivery address and phone number, the page return back | | |
| | **Test customers add the delivery address and phone number** | | | |
| 21 | Customer clicks “address” button to go to address page | System checks the database and displays address search box and phone number input box | | |
| 22 | Customer inputs the address and click search | System checks the address on map and display the result | | |
| 23 | Customer inputs the phone number | System record the new phone number and display on the screen | | |
| | **Test customers add the invalid delivery address and phone number** | | | |
| 24 | Customer inputs the invalid address and click search | System checks the address on map, and couldn’t find the specific address, system will display nothing, and customers couldn’t select the address | | |
| 25 | Customer inputs the invalid phone number | System checks the phone number format is wrong, and prompt “Please use Irish phone number” | | |
| | **Test delivery options** | | | |
| 26 | Customers do not need any action | The system selects the standard option by default. | | |
| 27 | Customer selects “Priority” option | The system record the selection and add €2 priority fees to the total price | | |
| | **Test proceed to pay** | | | |

| | | | | |
|---|---|---|---|---|
| 28 | Customer checks the item and quantity again | | | |
| 29 | Click “proceed to pay”button | System goes to payment system | | |
| | **Test the meal is out of stock** | | | |
| 30 | Repeat the step 1 and 2, | System checks the item whether is valid, if the meal is not valid(out of stock). The items in the shopping cart will be displayed in red "out of stock" | | |
| 31 | Click to “proceed to pay” | The system will display "Error, items in the shopping cart has changed" | | |
| | **Testing out of operating hours** | | | |
| 32 | Repeat the step 1, enter the shopping cart | The system displays "The restaurant is closed, please choose another restaurant and try again", system prompts the button “find another shop” | | |
| **Post-conditions** | | | | |
| Post-conditions 1 | | The place and order system will further connect the payment system | | |
| Post-conditions 2 | | The selection should be recorded in the database | | |

# 2. Sequence Diagram

## 2.1. Use case flow：

Main Flow:

1. Customer selects items and adds them to the shopping cart.
2. The shopping cart system displays the selected items and their quantities.
3. The customer checks the items and their quantities.
4. The customer can modify the cart by adding or removing items.
5. The shopping cart system returns the updated items and their quantities.
6. The shopping cart system checks the validity of any coupons.
7. If the coupon is valid, the shopping cart system displays the coupon for the customer to select.
8. The customer selects the coupon.
9. The shopping cart system generates a share link for the coupon.
10. If the coupon is not valid, the shopping cart system prompts the customer to share an invitation for a valid coupon.

11. The customer shares the invitation.
12. If the sharing is successful, the shopping cart system generates a valid coupon and displays it for the customer to select.
13. The customer selects the coupon.
14. The shopping cart system displays the delivery information and prompts the customer to check the address and phone number.
15. The customer checks the delivery information and can modify it if needed.
16. The shopping cart system returns the modified delivery information.
17. The shopping cart system displays the available delivery methods.
18. The customer selects the delivery method.
19. The shopping cart system displays the selected items and their order details for the customer to confirm.
20. The customer confirms the order is correct.
21. The shopping cart system proceeds to checkout.

Alternate Flows:

- If the customer decides not to proceed with the order at any point, they can choose to exit the system.
- If the shopping cart system is unable to display the items and quantities, it can display an error message.
- If the customer checks the items and quantities and finds an error, they can modify the cart by adding or removing items.
- If the coupon is not valid and the customer declines to share the invitation, the shopping cart system will not generate a valid coupon and will proceed without one.
- If the customer selects a delivery method that is not available or incompatible with the items, the shopping cart system can display an error message and prompt the customer to select a different delivery method.
- If the customer confirms the order is not correct, they can modify the cart by adding or removing items or changing the delivery information before proceeding to checkout.

## 2.2. Message labelled with a method signature

Loop start(confirm order is right or not)
shopping cart system to customer
1. displayShoppingCart(items, qty)

Customer to shopping cart system
2. checkItems(items,qty)
3. modifyCart(items,qty)modifyCart can be clear all the items in the cart

shopping cart system to customer

3.1. return(items,qty)

shopping cart system to shopping cart system

3.2. checkCouponValidity

shopping cart system to customer

Alt if coupon valid

3.3. displayCoupon

Customer to shopping cart system

4. selectCoupon

shopping cart system to shopping cart system

4.1. generateShareLink

Customer to shopping cart system

if coupon not valid

5. shareInvitation

shopping cart system to shopping cart system

Alt if sharing successful

5.1. generateValidCoupon

shopping cart system to customer

5.2. displayCoupon

Customer to shopping cart system

6. selectCoupon

if sharing successful

7. notUseCoupon

End alt Sharing Successful

End alt Coupon valid

shopping cart system to customer

7.1. displayDeliveryInformation

Customer to shopping cart system

8. checkAddressPhone

9. modifyInformation

shopping cart system to customer

9.1. return()

9.2. displayDeliveryMethod

Customer to shopping cart system

10. selectDeliveryMethod

shopping cart system to shopping cart system

10.1. displaySeletItemsAndOrderDetails

Customer to shopping cart system

Loop end(confirm order is right or not)

11. proceedToCheckout

## 2.3. The functionality the sequence diagram is intended to model

The sequence diagram is intended to model the functionality of the shopping cart system during the checkout process. It shows the interactions between the customer and the system for selecting items, applying coupons, selecting delivery options, and confirming the order before proceeding to checkout. The loop in the sequence diagram represents the customer's ability to confirm the order details and make changes as necessary before proceeding to checkout.

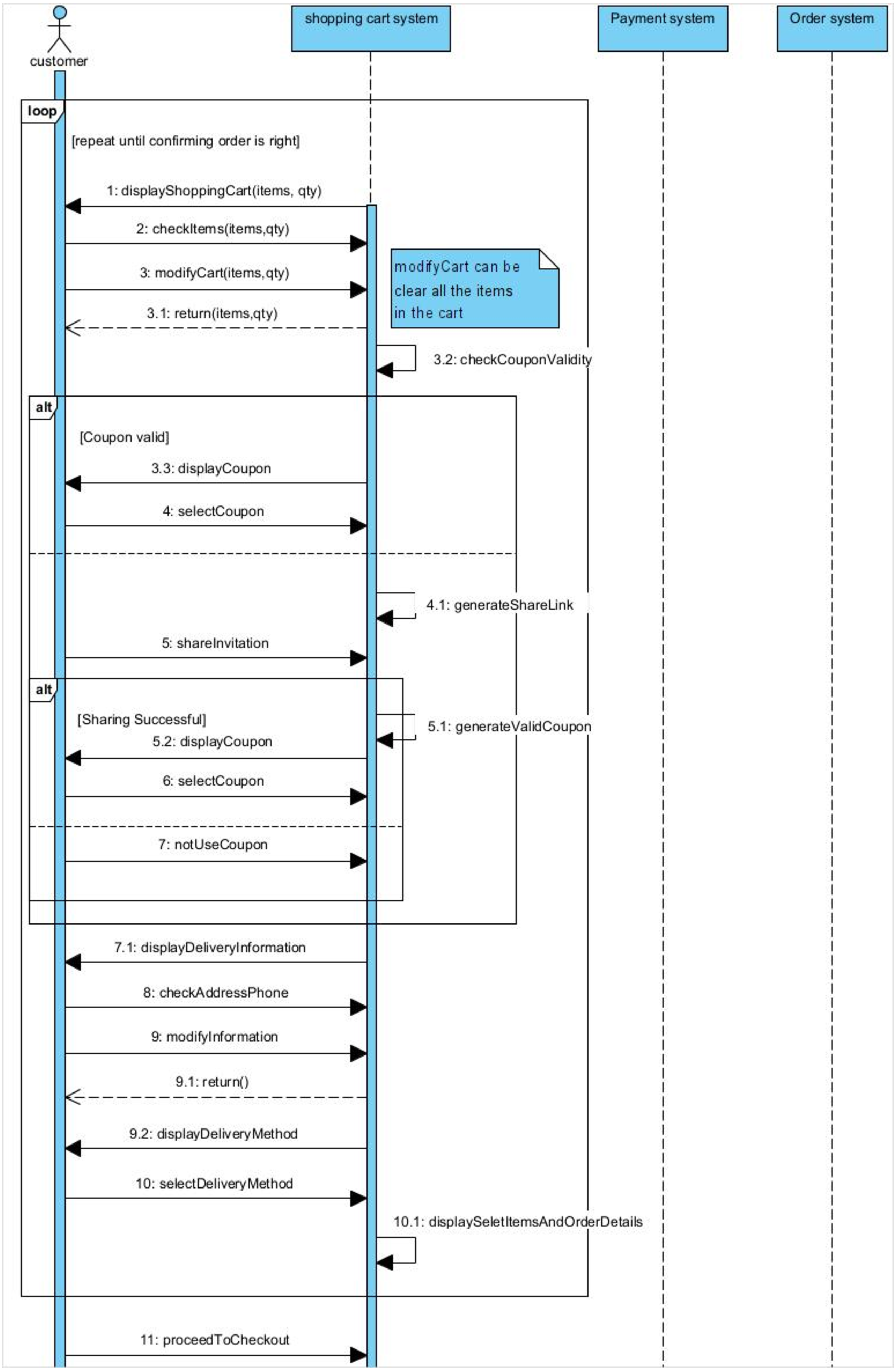
In order to clarify the function of place an order, I will distinguish the function or system that follows after place an order. After that is the order system and payment system. The order system is responsible for assigning orders to delivery staff, sending information to restaurants, updating order tacking information, sending confirmation emails to customers, and confirming successful payment. Payment system includes functions such as payment method selection, adding/updating payment methods, confirming successful payment, etc.

Modeling the place an order functionality in conjunction with MVC, would be the Shopping Cart System represents the Model, the Customer represents the View, and the interactions between the two are mediated by the Controller. It can be inferred that the Controller is responsible for handling the logic of updating the Model based on the user input from the View and updating the View based on changes in the Model.

## 2.4. Format

For this sequence diagram, i have used robustness stereotypes to depict the interactions between the customers, and shopping cart systems. Customer actions, system feedback and system self messages will be included in the diagram and labelled with method. Payment system and order system are not included in the “Place an order” functionality. Please see the additional chapter in the end of the assignment.

## 2.5. Sequence diagram



Combine robustness stereotype method to **describe the sequence diagram** interactively

- The shopping cart system receives a request from the customer to display the shopping cart items and their quantities using the displayShoppingCart(items, qty) method.

- The customer checks the items and quantities in the cart by using the checkItems(items,qty) method.

- If the customer needs to modify the cart, they can use the modifyCart(items,qty) method to make changes.

- If the customer clears all the items in the cart, the shopping cart system returns an empty cart by using the return(items,qty) method.

- The shopping cart system checks the validity of any coupon that the customer may have used, by using the checkCouponValidity method.

- If the coupon is valid, the shopping cart system displays the coupon to the customer by using the displayCoupon method.

- The customer selects the coupon by using the selectCoupon method.

- If the coupon is not valid, the shopping cart system generates a share link for the customer to share with their friends using the generateShareLink method.

- The customer shares the link with their friends using the shareInvitation method.

- If the sharing is successful, the shopping cart system generates a valid coupon for the customer to use by using the generateValidCoupon method.

- The shopping cart system displays the coupon to the customer using the displayCoupon method.

- The customer selects the coupon by using the selectCoupon method.

- If the sharing is not successful or the coupon is not used, the customer proceeds to checkout by using the notUseCoupon method.

- The shopping cart system displays the delivery information to the customer by using the displayDeliveryInformation method.

- The customer checks and modifies the address and phone number information by using the checkAddressPhone and modifyInformation methods.

- The shopping cart system returns the updated information to the customer using the return() method.

- The shopping cart system displays the delivery method options to the customer using the displayDeliveryMethod method.

- The customer selects the delivery method by using the selectDeliveryMethod method.

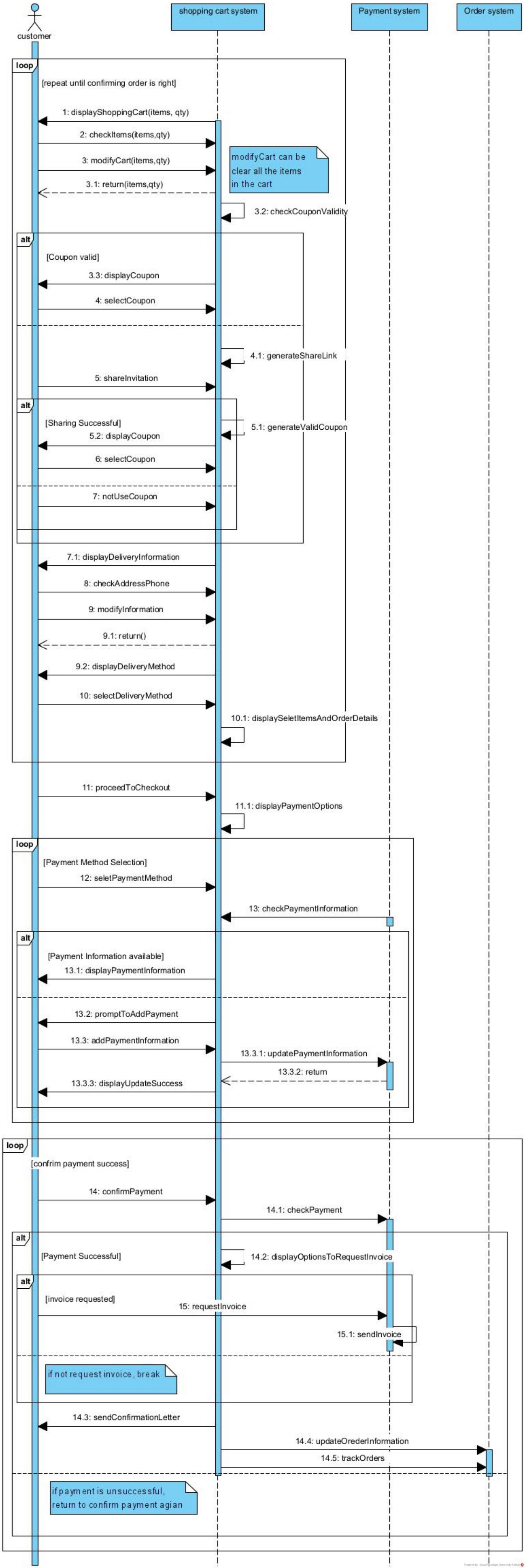
- The shopping cart system displays the selected items and order details to the customer using the displaySeletItemsAndOrderDetails method.

- The customer confirms the order and proceeds to checkout by using the proceedToCheckout method.

- The loop starts again to confirm the order is correct or not.

# Additional Chapter

To better understand the role of shopping cart system and payment system and order system, the above diagram is an example, payment and order functions are not in the scope of this assignment “place an order” case.


customer
shopping cart system
Payment system
Order system
loop
[repeat until confirming order is right]
1: displayShoppingCart(items, qty)
2: checkItems(items,qty)
3: modifyCart(items,qty)
modifyCart can be clear all the items in the cart
3.1: return(items,qty)
3.2: checkCouponValidity
alt
[Coupon valid]
3.3: displayCoupon
4: selectCoupon
4.1: generateShareLink
5: shareInvitation
alt
[Sharing Successful]
5.1: generateValidCoupon
5.2: displayCoupon
6: selectCoupon
7: notUseCoupon
7.1: displayDeliveryInformation
8: checkAddressPhone
9: modifyInformation
9.1: return()
9.2: displayDeliveryMethod
10: selectDeliveryMethod
10.1: displaySeletItemsAndOrderDetails
11: proceedToCheckout
11.1: displayPaymentOptions
loop
[Payment Method Selection]
12: seletPaymentMethod
13: checkPaymentInformation
alt
[Payment Information available]
13.1: displayPaymentInformation
13.2: promptToAddPayment
13.3: addPaymentInformation
13.3.1: updatePaymentInformation
13.3.2: return
13.3.3: displayUpdateSuccess
loop
[confrim payment success]
14: confirmPayment
14.1: checkPayment
alt
[Payment Successful]
14.2: displayOptionsToRequestInvoice
alt
[invoice requested]
15: requestInvoice
15.1: sendInvoice
if not request invoice, break
14.3: sendConfirmationLetter
14.4: updateOrederInformation
14.5: trackOrders
if payment is unsuccessful, return to confirm payment agian