Software Design Description  
*<9Meal>*

1. Apply UML
2. Each component and connector in diagram must have name/description
3. Each diagram must have an ID
4. Ensure no spelling mistake!

# Summary

This document describes architectural design for *<9Meal>* via 4+1 View approach

The intended audiences are software development team and deployment team.

## Reference

* *<9Meal Description document>*

# Architecture Style Selection

Provide justification for choosing architecture style, relevant to FRs, NFRs known in product description

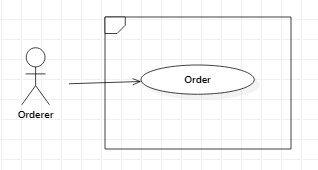
| **Selected Architecture Style** | **Justification of Selection** | **Affected FRs** | **Affected NFRs** |
| --- | --- | --- | --- |
| **MVC** | Used to separate the representation of internal information from the presentation of the information and acceptance from the user. It separates components and allows for efficient code reuse. It can both process requests from users and display notifications to users. | - Choose restaurant or food  - Check out  - Accept order  - Cancel order |  |
| **Repository** | Because the nature of a large system is database, it will manage and retrieve data stored on the database. | - Manage accounts  - Manage user details  - Manage shipper details  - Manage restaurant details  - Manage restaurant rates and reviews |  |
| **Client – Server** | The client means the shipper continuously sends information about their location every 4 seconds. The server will continuously save this location and display it on the system and the orderer can see the actual location of the shipper. | - Contact between orderer with shipper.  - Contact between restaurant with shipper.  - Track location of shipper. |  |
| **Pipe-Filter** | During processing, the first component must be done before the next component. After a successful output, must is store data on DataSink (Database). | - CRUD food in cart  - Show menu |  |

# Architectural Design

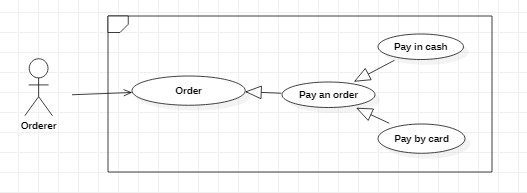
## Scenario View

*[Usecase diagram and use case description of key (not all) use cases*

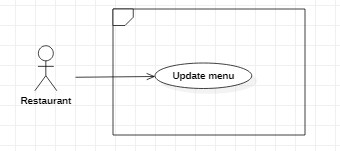
| **UC ID** | **Use case** | **Actors** | **Pre-condition** | **Post-condition** | **Logical Processing** |
| --- | --- | --- | --- | --- | --- |
| **UC1** | Order | Orderer | \_ Orderer is already login. | Success: Return checkout page  Fail: Show a warning message | \_ The orderer select restaurant and food then add the selected items to cart. |
| **UC2** | Pay an order | Orderer | \_ Orderer is already login.  \_ Current screen is “Checkout page” | Success: Order checkout successful.  Fail: System shows a warning message or terminates the session. | \_ The order has been confirmed and the total bill has been displayed on the screen to the orderer. Orderer decides to go ahead with the order. |
| **UC3** | Update Menu | Restaurant | Restaurant has to login to 9Meal’s system. | Success: Update menu successful.  Fail: System shows a warning message or terminates the session. | \_ CRUD everything displayed in the menu (food, discount code, info...) |
| UC4 | Cancel Order | Orderer Shipper Restaurant | The order has been ordered from the orderer. | Success: Cancel successful.  Fail: The order is continued processing. | \_ Orderer can only cancel the order when it’s looking for shipper.  \_ Restaurant and shipper can cancel the order then send the reason to orderer. |



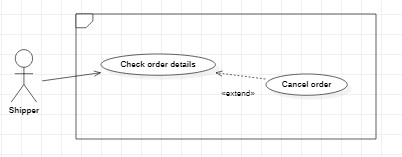
|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC1 Order** | | |
| Created By: |  | Date Created: |  |
| Primary Actor: | Orderer | Secondary Actors: |  |
| Description: | Orderer can make the process of order. | | |
| Trigger: | Orderer enter location | | |
| Preconditions: | PRE-1:Orderer has to login. | | |
| Post-conditions: | POST-1:  ● Success: Return checkout page.  ● Fail: System shows a warning message or terminates the session. | | |
| Normal Flow: | 1. Orderer enter location to find available restaurants or food in that area.  2. System display list of available restaurants or food.  3. Orderer select restaurants or food.  4. System display menu.  5. Orderer can select food and drink or combo deal.  6. Orderer tap button “add” to add the selected item to cart.  7. Orderer tap button “Checkout” & select delivery address.  8. System return checkout page.  9. Orderer select one of two payment methods: cash or credit card. | | |
| Alternative Flows: |  | | |
| Exceptions: | 1. Item can not add to cart because restaurant ran out of ingredients to cook that dish.  1.1. Orderer select item & tap the button “add”.  1.2. System send a message “This dish is temporarily unavailable”.  2. Can not checkout because no item in cart.  2.1. Orderer tap the button “checkout”.  2.2. System send a message “Please add item to checkout”.  3. Internet connection lost.  3.1. System send a message “Something wrong with the internet. Please check your network connection”.  4. Database transaction failed.  4.1. System send a message “Server failed to complete your request, please try again later”. | | |
| Priority: |  | | |
| Frequency of Use: |  | | |
| Business Rules: | BR-1: Orderer must connect to the Internet.  BR-2: Should there be a connectivity problem that lasts at least 5 second, terminate the session and move it to another dedicated server. | | |
| Other Information: |  | | |
| Assumptions: |  | | |

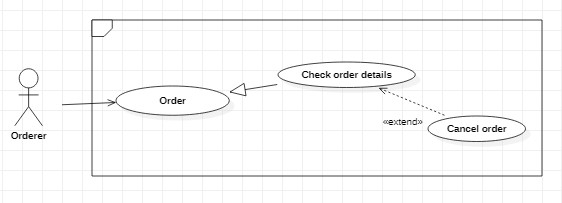
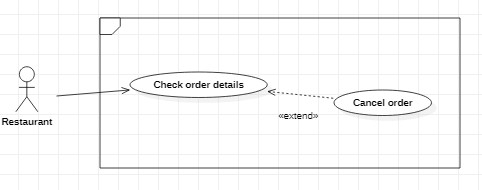


|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC2 Pay an order** | | |
| Created By: |  | Date Created: |  |
| Primary Actor: | Orderer | Secondary Actors: |  |
| Description: | Orderer can choose payment method and checkout. | | |
| Trigger: | Orderer select payment method. | | |
| Preconditions: | PRE-1:Orderer has to login.  PRE-2: Current screen is “checkout page”. | | |
| Post-conditions: | POST-1:  ● Success: Order checkout successful.  ● Fail: System shows a warning message or terminates the session. | | |
| Normal Flow: | 1. Orderer select one of two payment methods: cash or card.  1.1. Select cash payment.  1.2. Select card payment.  1.2.1. Orderer select card to checkout  2. Orderer tap the button “Order”.  3. System send a message “The order is processing” | | |
| Alternative Flows: | 1. Card not yet linked  1.1. System move to card detail page.  1.2. Orderer enter card information & tap the button “Finish”.  1.3. Orderer enter OTP code.  1.4. System return checkout page. | | |
| Exceptions: | 1. Payment failed  1.1. System send a message “Payment failed, please choose another card or another payment method”.  2. Internet connection lost.  2.1. System send a message “Something wrong with the internet. Please check your network connection”.  3. Database transaction failed.  3.1. System send a message “Server failed to complete your request, please try again later”. | | |
| Priority: |  | | |
| Frequency of Use: |  | | |
| Business Rules: | BR-1: Orderer must connect to the Internet.  BR-2: Payment amount must be numeric | | |
| Other Information: |  | | |
| Assumptions: |  | | |

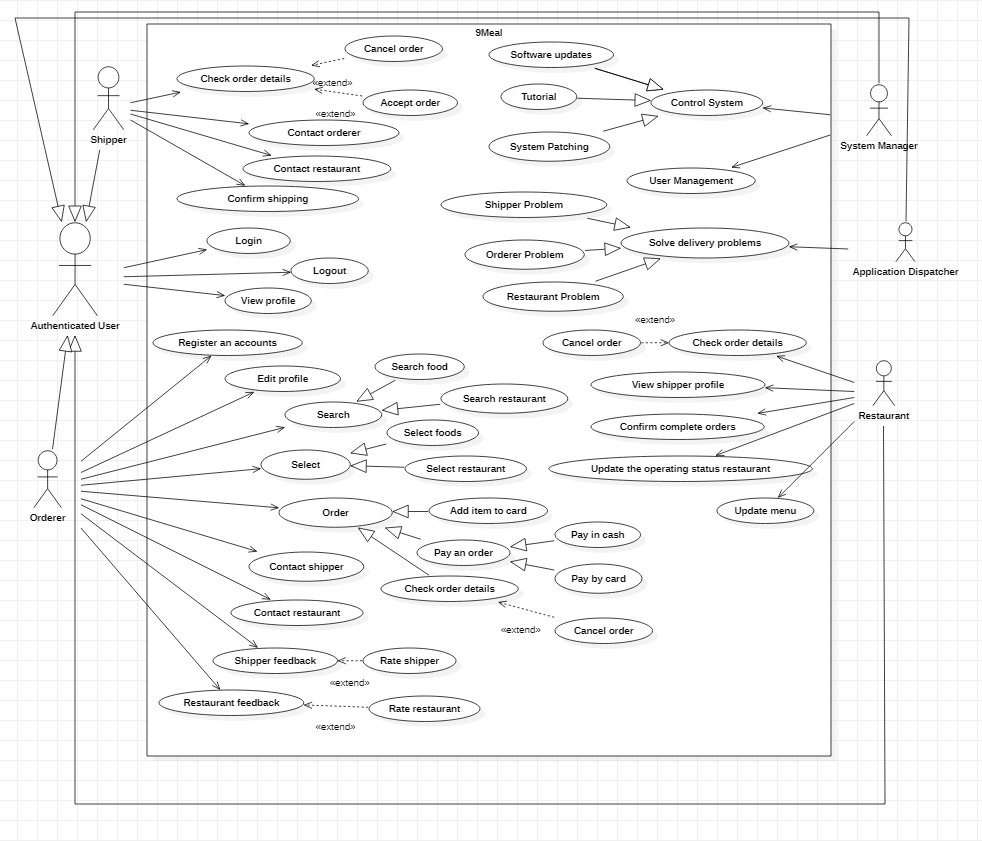


|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC3 Update Menu** | | |
| Created By: |  | Date Created: |  |
| Primary Actor: | Restaurant | Secondary Actors: |  |
| Description: | Restaurant can edit menu | | |
| Trigger: | Restaurant tap “Edit” button | | |
| Preconditions: | PRE-1:Restaurant has to login to 9Meal’s system. | | |
| Post-conditions: | POST-1:  ● Success: Update menu successful.  ● Fail: System shows a warning message or terminates the session. | | |
| Normal Flow: | 1. Restaurant enter account to login to 9Meal’s system  2. System authenticate account & move to edit menu page  3. Restaurant tap “Edit” button.  4. System return message “Is change require?”  3.1. Restaurant tap “Yes” button  3.1.1. Restaurant enter food information.  3.1.2. Restaurant tap “Finish” button.  3.1.3. System return edit menu page.  3.2. Restaurant tap “No” button  3.2.1. System return edit menu page.  4. Restaurant tap “Save” button. | | |
| Alternative Flows: | NA | | |
| Exceptions: | 1. Login failed  1.1. System send a message “Incorrect username or password, please check again”.  2. Internet connection lost.  2.1. System send a message “Something wrong with the internet. Please check your network connection”.  3. Database transaction failed.  3.1. System send a message “Server failed to complete your request, please try again later”.  4. Food ID already exists.  4.1. System send a message “ID is already exist”. | | |
| Priority: |  | | |
| Frequency of Use: |  | | |
| Business Rules: | BR-1: Restaurant must connect to the Internet. | | |
| Other Information: |  | | |
| Assumptions: |  | | |



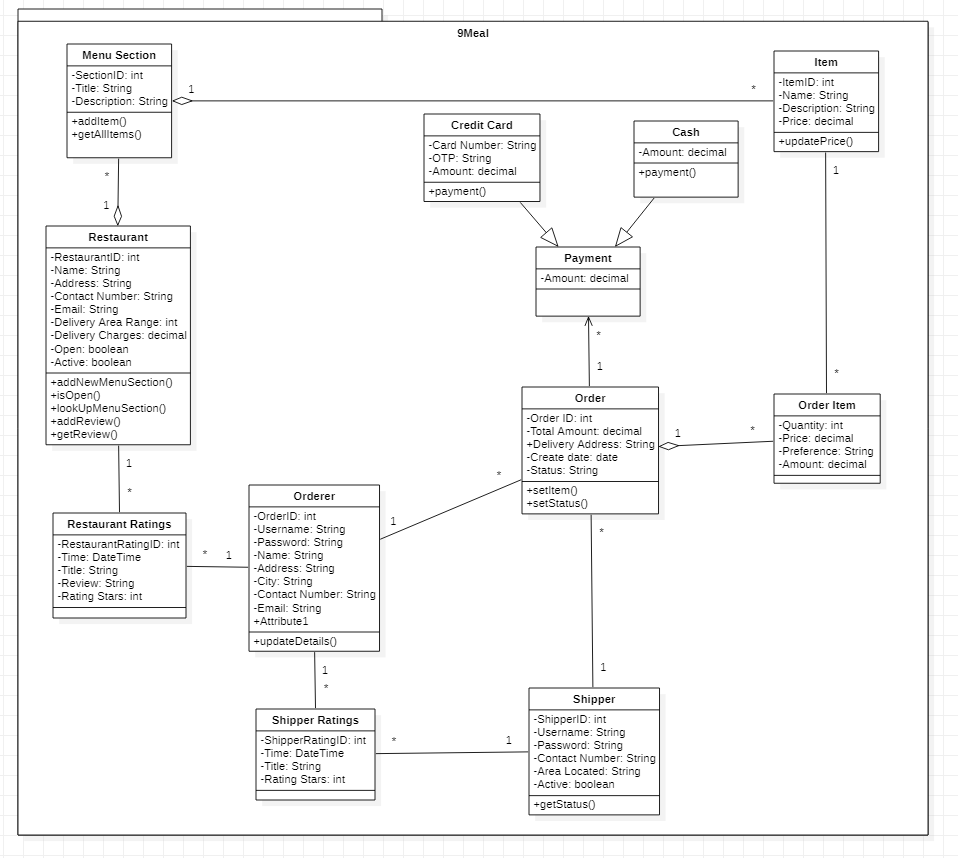


|  |  |  |  |
| --- | --- | --- | --- |
| ID and Name: | **UC4 Cancel Order** | | |
| Created By: |  | Date Created: |  |
| Primary Actor: | Restaurant, Shipper  Orderer | Secondary Actors: |  |
| Description: | Restaurant and shipper can cancel order.  Orderer can cancel order. | | |
| Trigger: | The actor tap “Cancel order” button. | | |
| Preconditions: | PRE-1:The order has been ordered from the orderer. | | |
| Post-conditions: | POST-1:  ● Success: Cancel successful.  ● Fail: The order is continued processing. | | |
| Normal Flow: | **Restaurant and shipper:**  1. Tap “Cancel order” button.  2. System pop-up a window to choose the reason cancel.  3. Send the reason back to the system.  4. Send notifice of cancel order to orderer.  **Orderer:**  1. Tap “Cancel order” button.  2. System send a message to confirm cancel  2.1. Tap “Yes” button.  2.1.1. System return checkout page.  2.2. Tap “No” button.  2.2.1. Continue looking for shipper. | | |
| Alternative Flows: | NA | | |
| Exceptions: | 1. Internet connection lost.  1.1. System send a message “Something wrong with the internet. Please check your network connection”.  2. Database transaction failed.  2.1. System send a message “Server failed to complete your request, please try again later”. | | |
| Priority: |  | | |
| Frequency of Use: |  | | |
| Business Rules: | BR-1: The actor must connect to the Internet.  BR-2: Shipper can only cancel up to 3 times in a day. | | |
| Other Information: |  | | |
| Assumptions: |  | | |



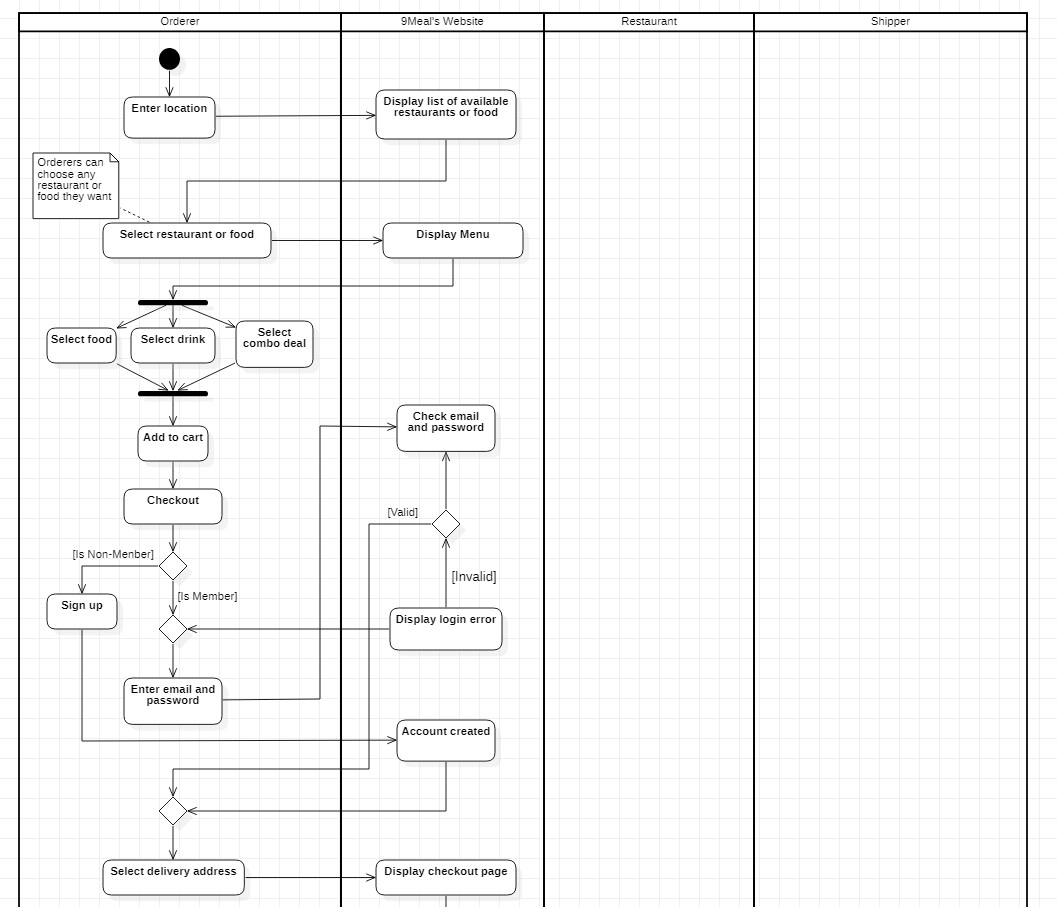
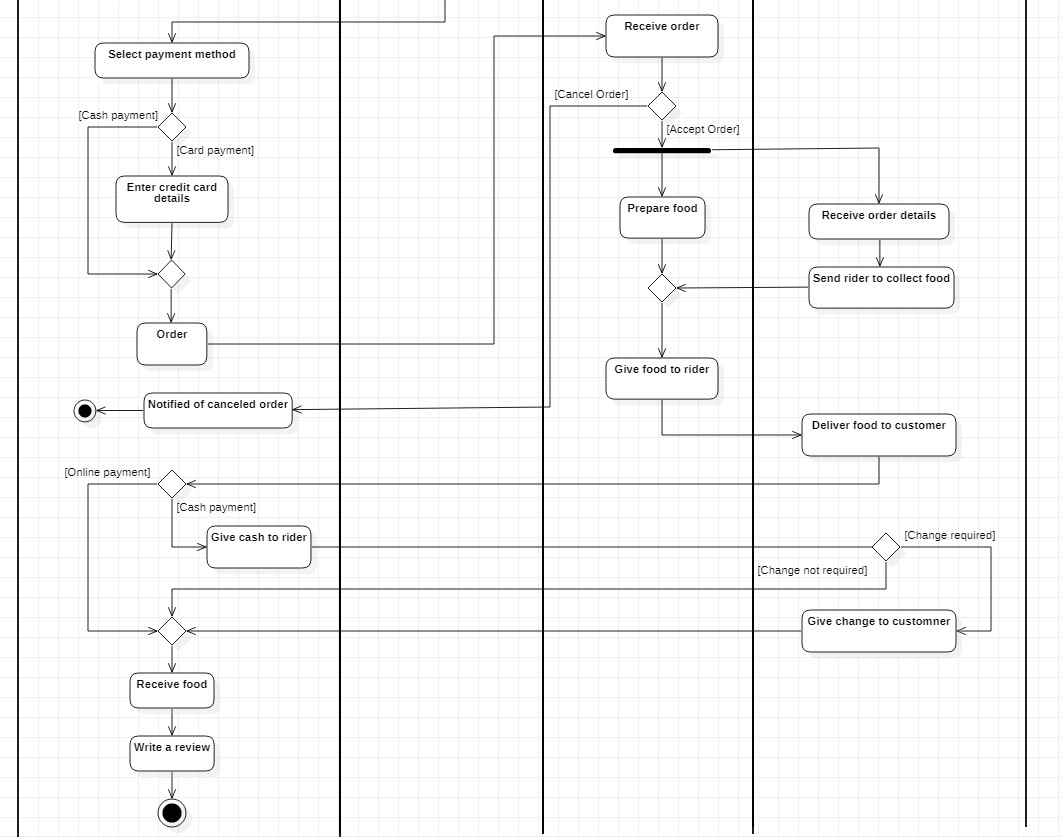
(Use case diagram for 9Meal’s App )

## Logical View

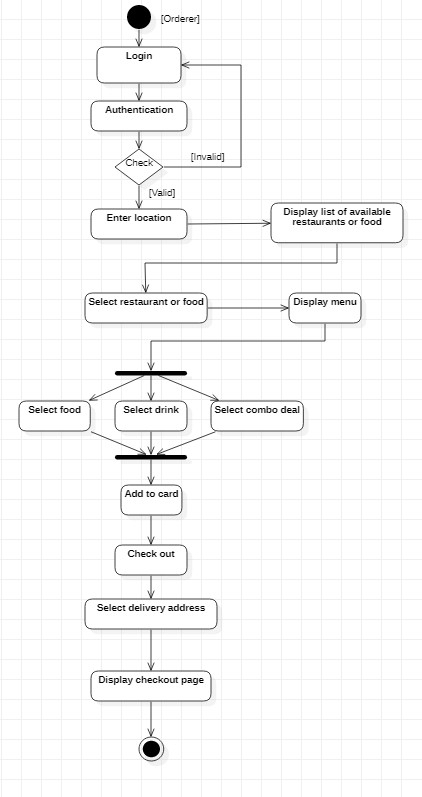


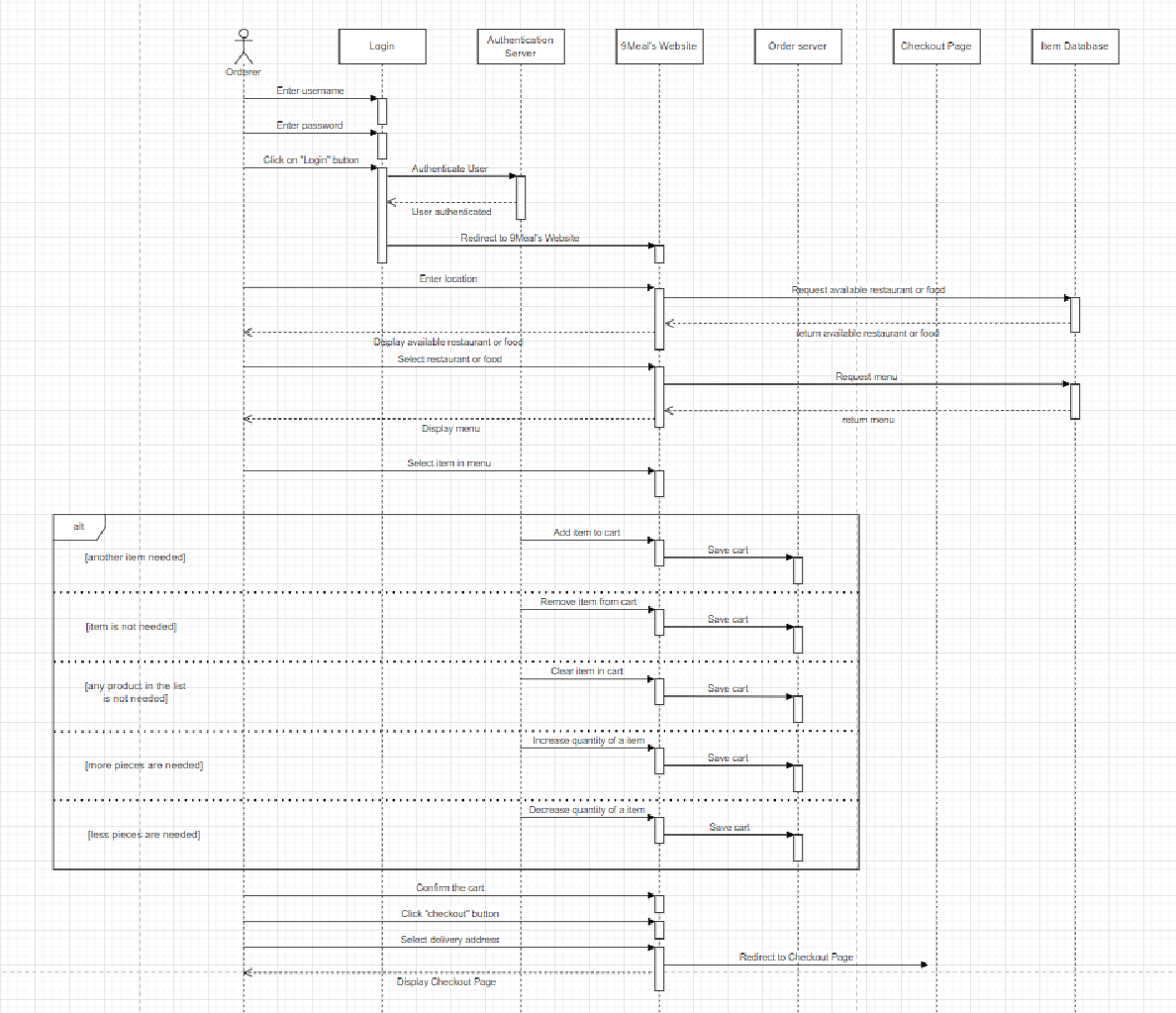
(Class diagram for 9Meal’s App)

## Process View

(Activity diagram for 9Meal’s App)

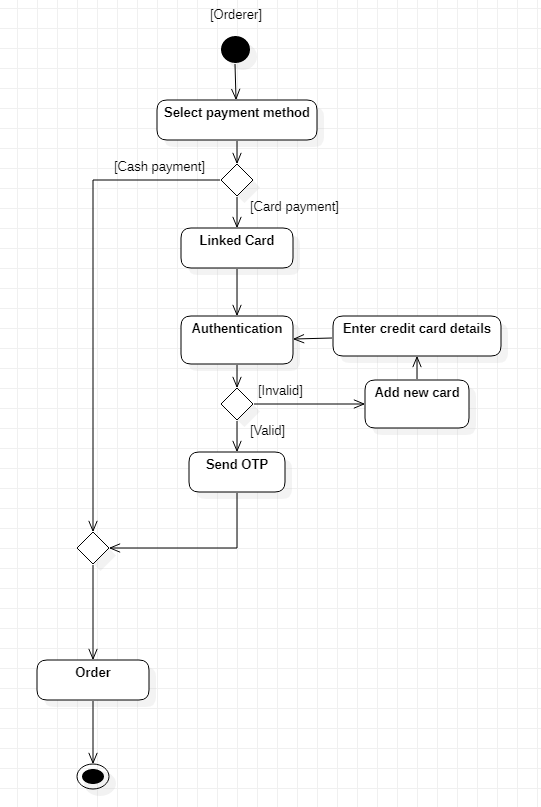
**Order**

(Activity diagram for “order” function)

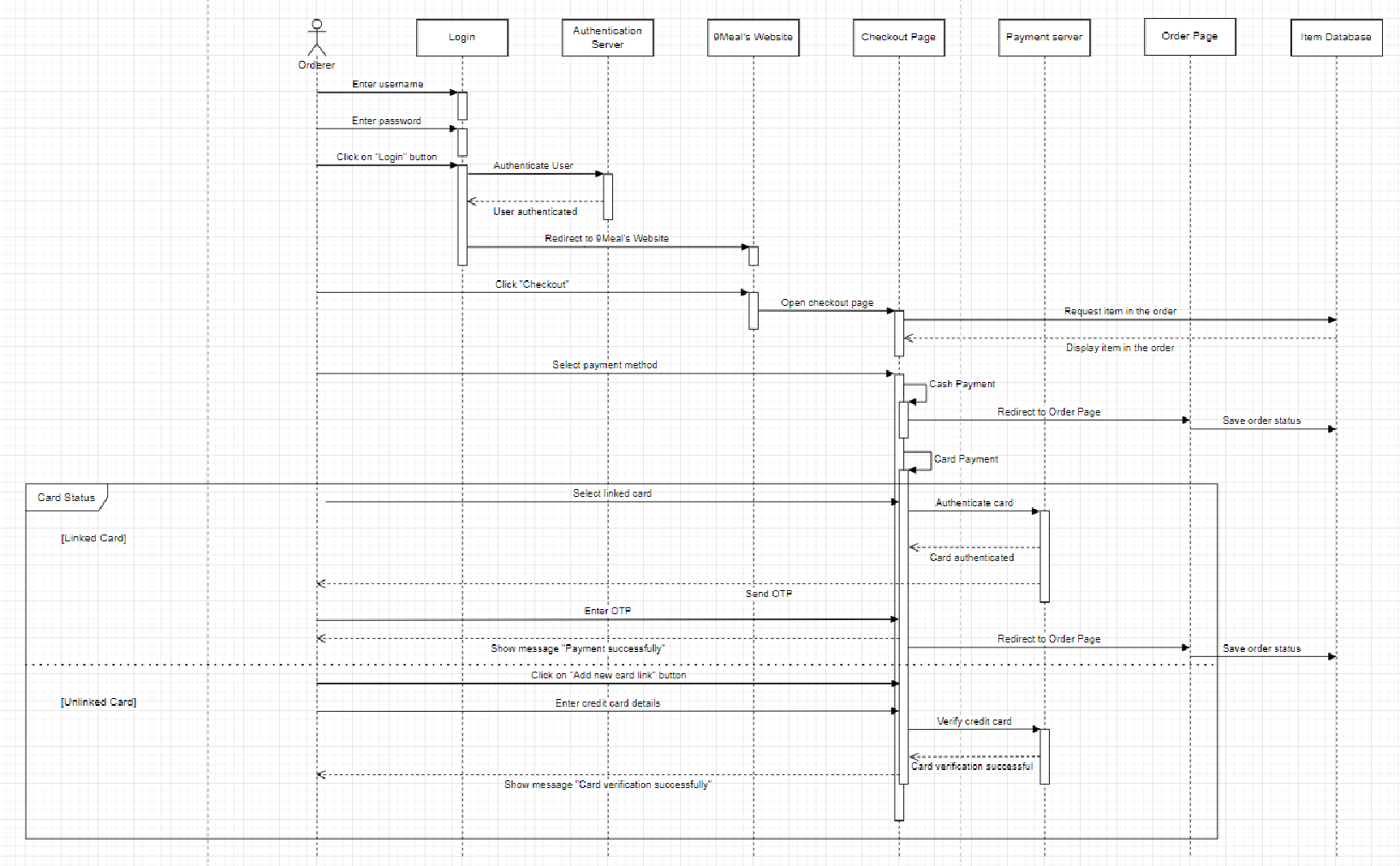
****

(Sequence diagram for “order” function)

**Pay an order**

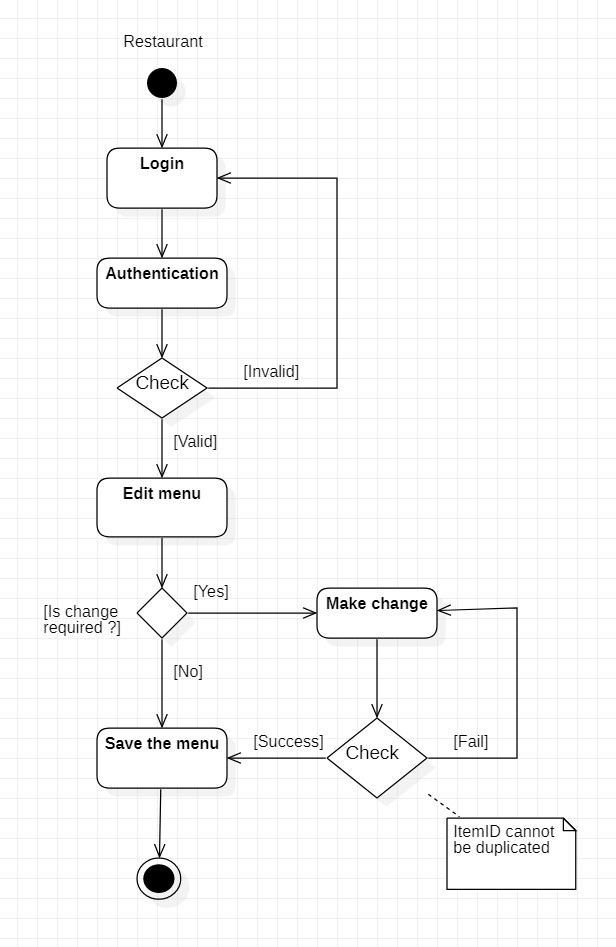
****

(Activity diagram for “pay an order” function)

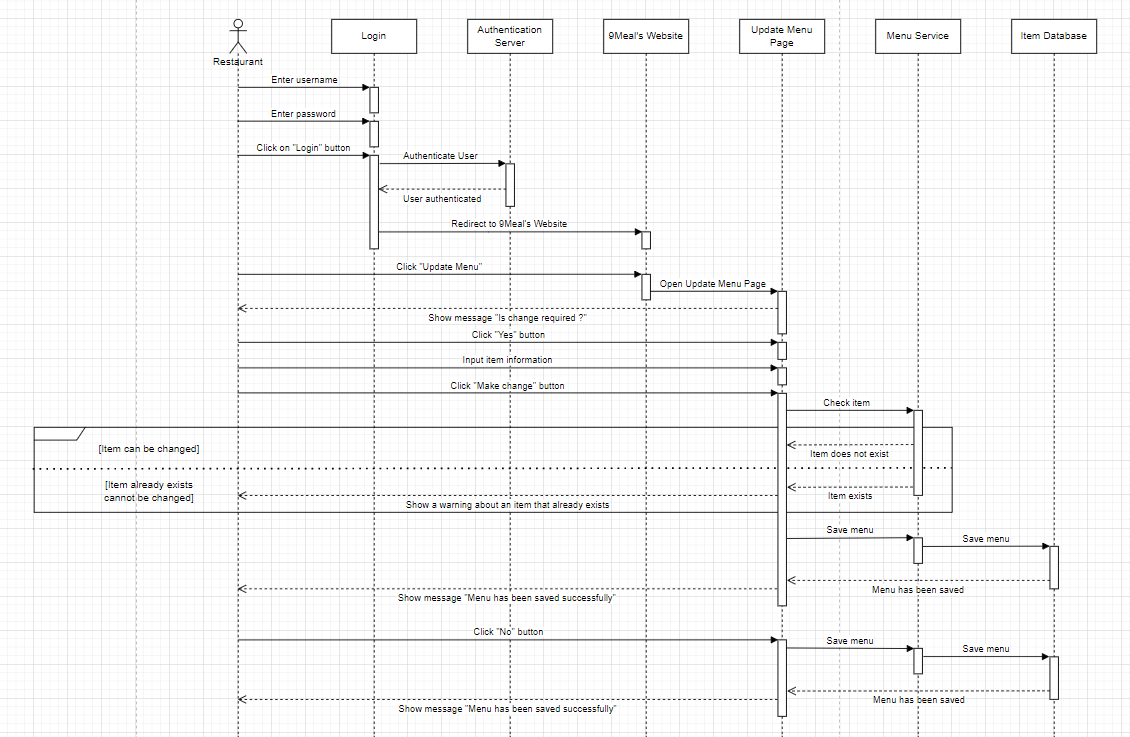
****

(Sequence diagram for “pay an order” function)

**Update Menu**

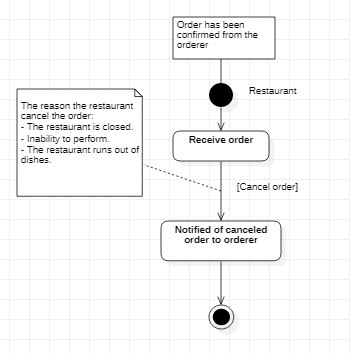


(Activity diagram for “Update Menu” function)

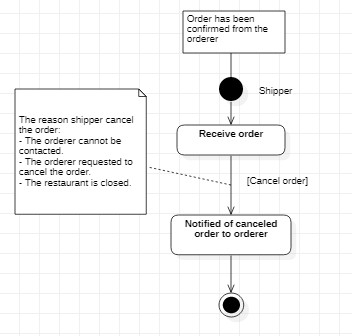


(Sequence diagram for “Update Menu” function)

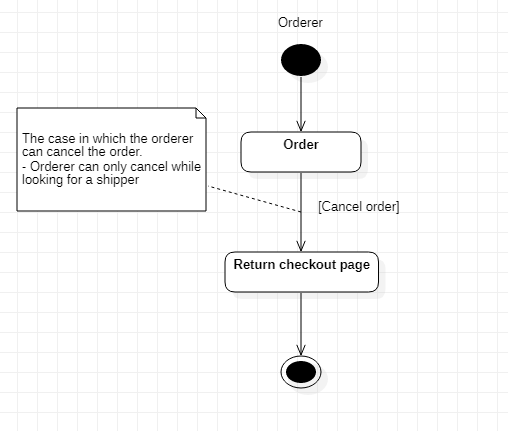
**Cancel**



(Activity diagram for “Restaurant’s cancel” function)



(Activity diagram for “Shipper’s cancel” function)



(Activity diagram for “Orderer’s cancel” function)

## Physical View

