|  |  |
| --- | --- |
| Group Member: | Pham Minh Tan  Pham Thanh Long  Vu Lam Bao Van  Nguyen Thanh Hiep |
| Supervisor: | Mr. Duong Dung Tien |

Introduction

Food Hub, with just a few taps on the screen, we now have access to hundreds or new and delicious restaurants ranging from every cuisine you can think of.

Food Hub

Software Architecture Styles

Contents

[1. Introduction 3](#_Toc77444956)

[1.1 Food Hub Description 3](#_Toc77444957)

[1.2 Scope 3](#_Toc77444958)

[1.3 Role user 3](#_Toc77444959)

[2. Add Food (Manager) 4](#_Toc77444960)

[2.1 Scenario 4](#_Toc77444961)

[2.2 Use Case Description 4](#_Toc77444962)

[2.3 Data-centered (Repository) 4](#_Toc77444963)

[2.3.1 Advantage 4](#_Toc77444964)

[2.4 Data-centered (Repository) style 5](#_Toc77444965)

[2.4.1 Class Diagram 5](#_Toc77444966)

[2.4.2 Sequence Diagram 6](#_Toc77444967)

[3. Add Order (Customer) 7](#_Toc77444968)

[3.1 Scenario 7](#_Toc77444969)

[3.2 Use Case Description 7](#_Toc77444970)

[3.3 Implicit Asynchronous Communication (Non-buffered) 7](#_Toc77444971)

[3.3.1 Advantage 7](#_Toc77444972)

[3.4 Interaction (MVC) 7](#_Toc77444973)

[3.4.1 Advantage 7](#_Toc77444974)

[3.5 Interaction (MVC) style 8](#_Toc77444975)

[3.5.1 Class diagram 8](#_Toc77444976)

[3.5.2 Sequence diagram 8](#_Toc77444977)

[3.6 Implicit Asynchronous Communication (Buffered) style 10](#_Toc77444978)

[3.6.1 Sequence Diagram 11](#_Toc77444979)

[11](#_Toc77444980)

[4. Search Food (Customer) 12](#_Toc77444981)

[4.1 Scenario 12](#_Toc77444982)

[4.2 Use Case Description 12](#_Toc77444983)

[4.3 Hierarchical (Main-Subroutine) 12](#_Toc77444984)

[4.3.1 Advantage 12](#_Toc77444985)

[4.4 Hierarchical (Main-Subroutine) style 13](#_Toc77444986)

[4.4.1 Activity Diagram 13](#_Toc77444987)

[4.4.2 Data Flow Diagram 14](#_Toc77444988)

[5. Feedback (Customer) 15](#_Toc77444989)

[5.1 Scenario 15](#_Toc77444990)

[5.2 Use Case Description 15](#_Toc77444991)

[5.3 Interaction (MVC) style 16](#_Toc77444994)

[5.3.1 Class Diagram 16](#_Toc77444995)

[5.3.2 Sequence Diagram 16](#_Toc77444996)

# Introduction

## Food Hub Description

* Food Hub is a mobile app that allows customers to place a food order from any of the restaurants in an area using their smartphone.
* Food Hub is an online food delivery system that allows you to order food online from your favorite restaurants and get it delivered to your doorstep in a quick and efficient manner.

## Scope

* Help the restaurant to manage their orders
* Make it convenient for the customers to buy food without going to the restaurant
* Provide for the user many functions such as searching for their favorite foods
* Seeing the other feedbacks and comparing the price of the restaurants
* Increased employment opportunities and income for restaurants and shipping companies

## Role user

1. Customers:

* Manage the order (add, cancel the order, view order’s history)
* Search foods or search the restaurants base on location, price, discount, …
* Feedback the restaurant, the shipper and the food
* See the review of the restaurant, the shipper and the food
* Find and apply the discount
* Chat with the Shipper

1. Restaurant (manager, staff):

* Manage the food (add, update the food’s details, delete)
* See customer’s order
* See customer’s feedbacks

1. Shipper

* Manage the order (accept, cancel the order, see the order’s details)
* See customer’s profile
* Chat with the Customer

# Add Food (Manager)

## Scenario

- As a manager, I want to add the new food, so that I can increase incomes and give the various options.

## Use Case Description

UC1: Add the new food to Food Hub system

|  |  |
| --- | --- |
| Use case Name | Add new food to Food Hub system |
| Brief description | This use case allow manager to add the new food to increase incomes and give the various options. |
| Actors | Manager |
| Basic Flow | 1. Manager clicks on the icon of adding the new food 2. System will change the manager current screen to the adding the new food screen 3. Manager chooses the type of food (fish, meat,…) 4. Manager enters information of food (name, image, price,…) 5. Manager click on “Add Food” button 6. System adds a new food to the menu 7. System displays the quantity, type and the price of all foods in the menu |
| Alternative Flows | **Alternative flow: Manager cannot add existed food to the menu**   1. From step #4 of the basic flow, manager add another food 2. Continue step #5 in the basic flow |
| Pre-conditions | Manager is logged into the Food Hub system. |
| Post-conditions | The manager successfully adds new food to the Food Hub system. |

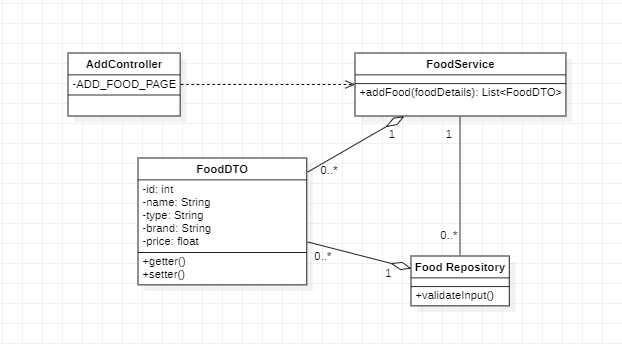
## Data-centered (Repository)

### Advantage

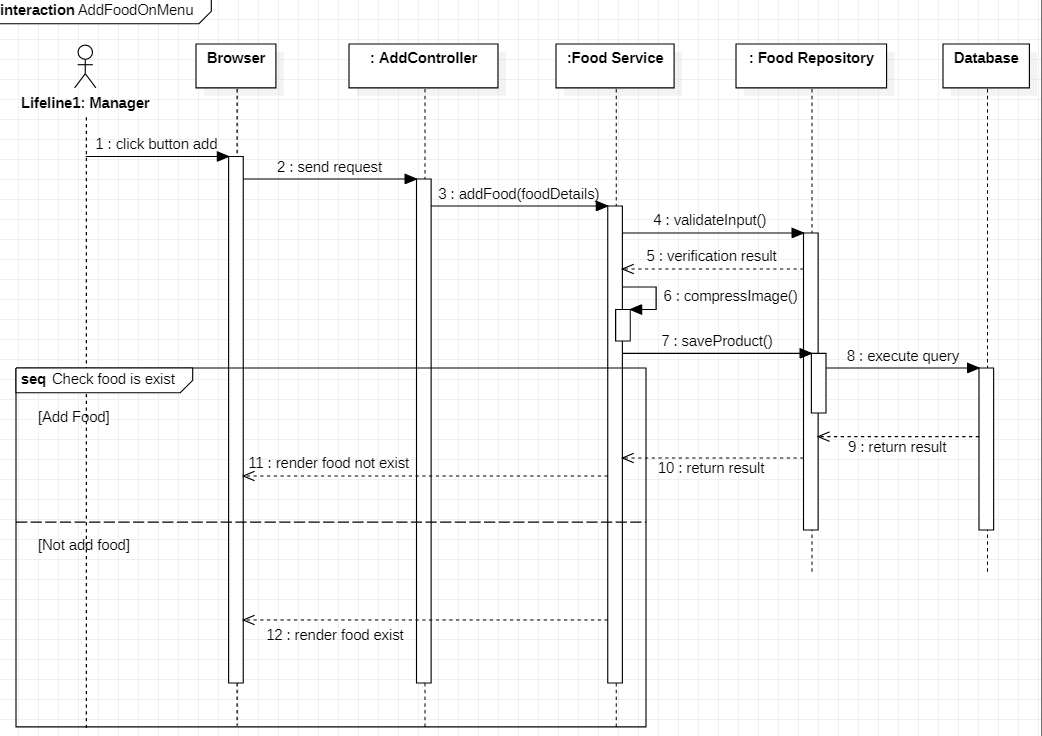
* Data integrity: easy to back up and restore
* System scalability and reusability of agents: easy to add new software components because they do not have direct communication with each other
* Reduces the overhead of transient data between software components

## Data-centered (Repository) style

### Class Diagram



### Sequence Diagram



# Add Order (Customer)

## Scenario

- As a customer, I want to add order, so that I can get my food.

## Use Case Description

UC2: Add Order

|  |  |
| --- | --- |
| Use case Name | Add the order |
| Brief description | This use case describes how the Customer can add order |
| Actors | Customer |
| Basic Flow | 1. At the find food screen 2. User clicks on ‘Add’ button that save to cart 3. System displays notification “Add success” 4. User view their cart 5. User click button “Payment” 6. User choose the method payment 7. User can add one product many times |
| Alternative Flow | **Alternative flow: The product has existed on the cart**   1. From step #2 in the basic flow, system increases the quantity and the price of the food 2. Continue step #3 |
| Pre-conditions | User is logged into the Food Hub system. |
| Post-conditions | The user successfully adds new food can see total price and the quantity of all the food on the cart. |

## Implicit Asynchronous Communication (Buffered)

### Advantage

* Anonymity: provides high degree of anonymity between message

producer and consumer

* Concurrency: supports concurrency both among consumers and

between producer and consumers.

* Supports loose coupling between message producers and consumers

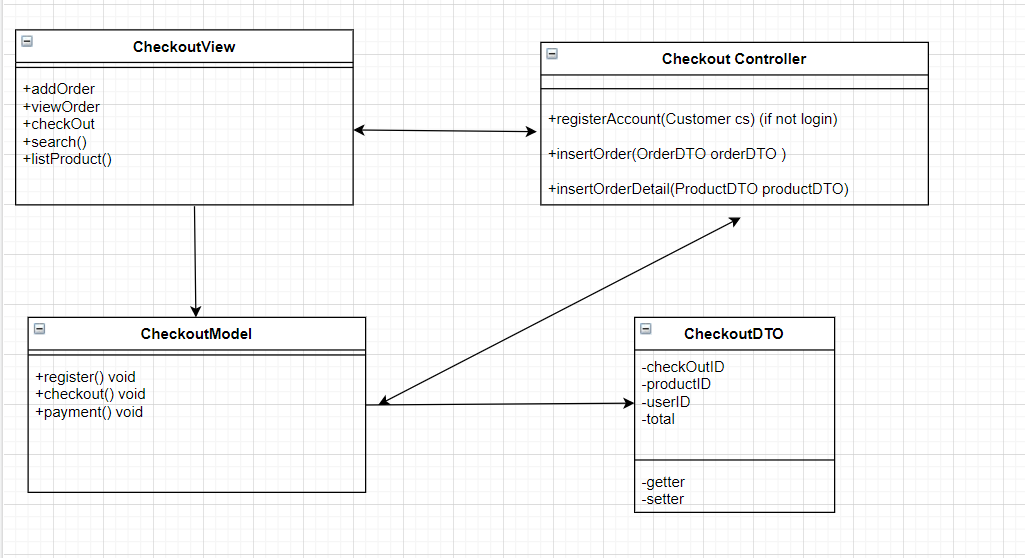
## Interaction (MVC)

### Advantage

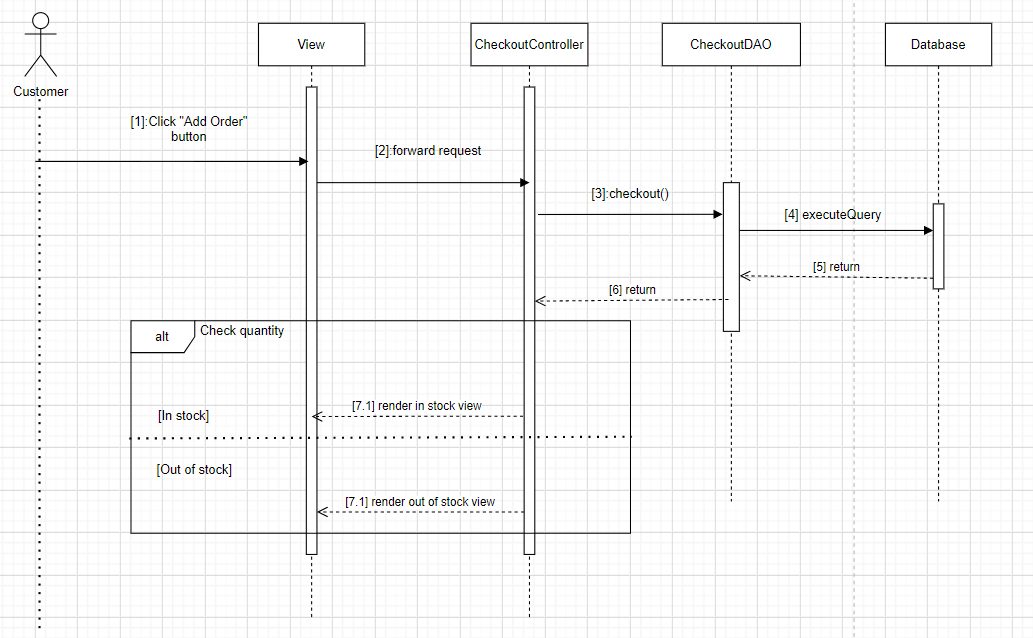
* Development of the application becomes fast.
* Easy for multiple developers to collaborate and work together.
* Easier to Update the application.
* Easier to Debug as we have multiple levels properly written in the application

## Interaction (MVC) style

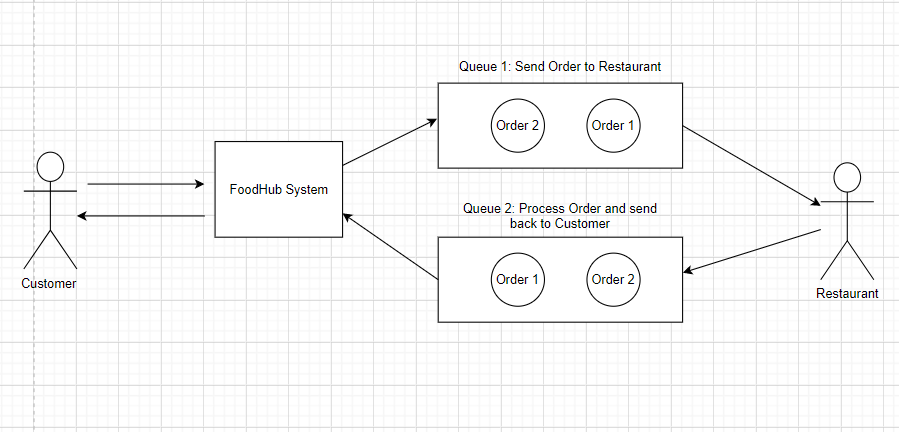
### Class diagram



### Sequence diagram



## Implicit Asynchronous Communication (Buffered) style



### Sequence Diagram

## 

# Search Food (Customer)

## Scenario

- As a customer, I want to search foods or the restaurants based on: the location, the price or the discount, … so that I can get the best meals from the restaurant. Apart from searching, I can view another comment about the food so that it can help me to decide to buy.

## Use Case Description

UC3: Search food to add the order

|  |  |
| --- | --- |
| Use case Name | Search food to add the order |
| Brief description | This use case describes how the Customer can search the expected food and order it. |
| Actors | Customer |
| Basic Flow | 1. At the homepage, the user enters keywords on the ‘Search’ field 2. User clicks on ‘Search’ button to start searching 3. System displays the restaurant 4. User chooses the specific restaurant 5. System displays the foods found 6. User chooses the specific food to see the information 7. User clicks ‘Add to cart’ button on the expected food 8. System adds a new food to the order 9. System displays the quantity and the price of all foods in the order |
| Alternative Flows | **Alternative flow 1: User cannot find products searched**   1. From step #1 of the basic flow, user enters another keyword 2. Continue step #2 in the basic flow   **Alternative flow 2: User can view feedback**   1. From step #4 and step #6 of the basic flow, user views the comment and rating of another user about the food 2. Continue step #7 in the basic flow   **Alternative flow 3: User can choose another search method**   1. From step #1 of the basic flow, user can enter the restaurant name to search base on: the nearest location, the price, the discount 2. Continue step #2 in the basic flow |
| Pre-conditions | User is logged into the Food Hub system. |
| Post-conditions | The user successfully adds new food to the order or increases quantity and price of the existing food in the order. |

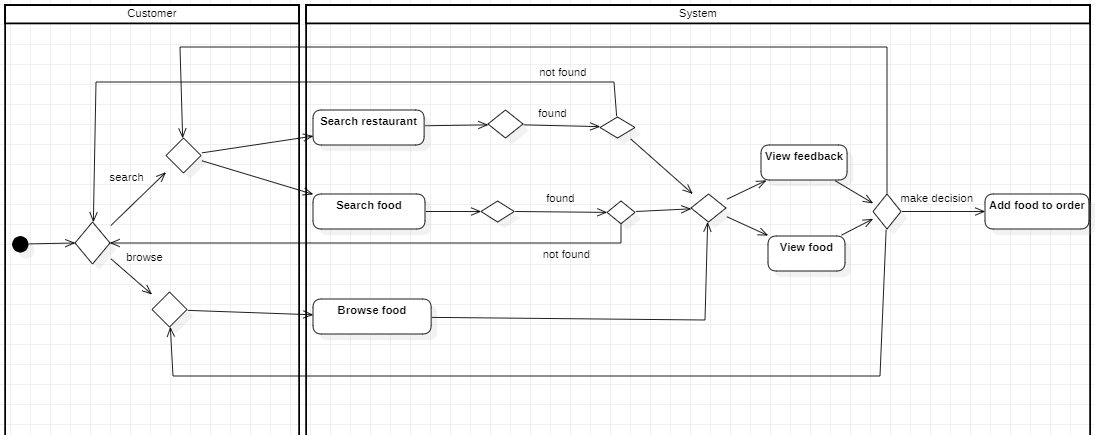
## Hierarchical (Main-Subroutine)

### Advantage

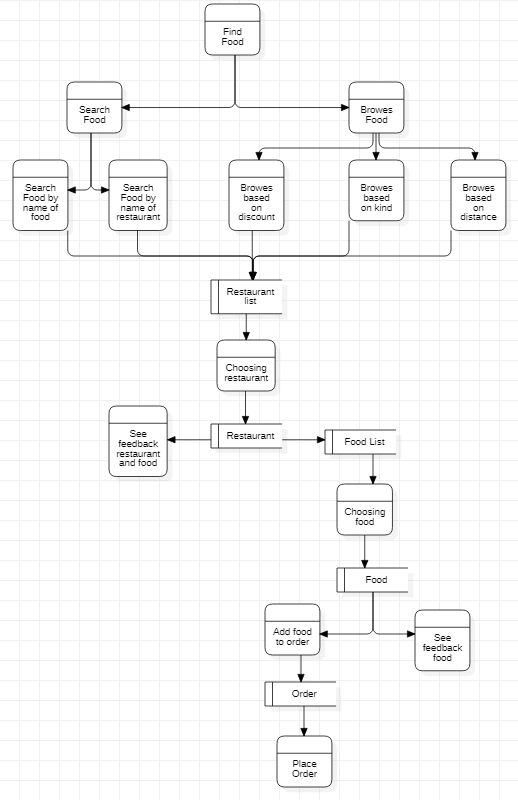
* It is easy to decompose the system based on the definition of the tasks in a top-down refinement manner.
* This architecture can still be used in a subsystem of OO design.

## Hierarchical (Main-Subroutine) style

### Activity Diagram



### Data Flow Diagram



# Feedback (Customer)

## Scenario

- As a customer, I want to feedback the shipper or the food or the restaurant’s service, so that I can share my opinions.

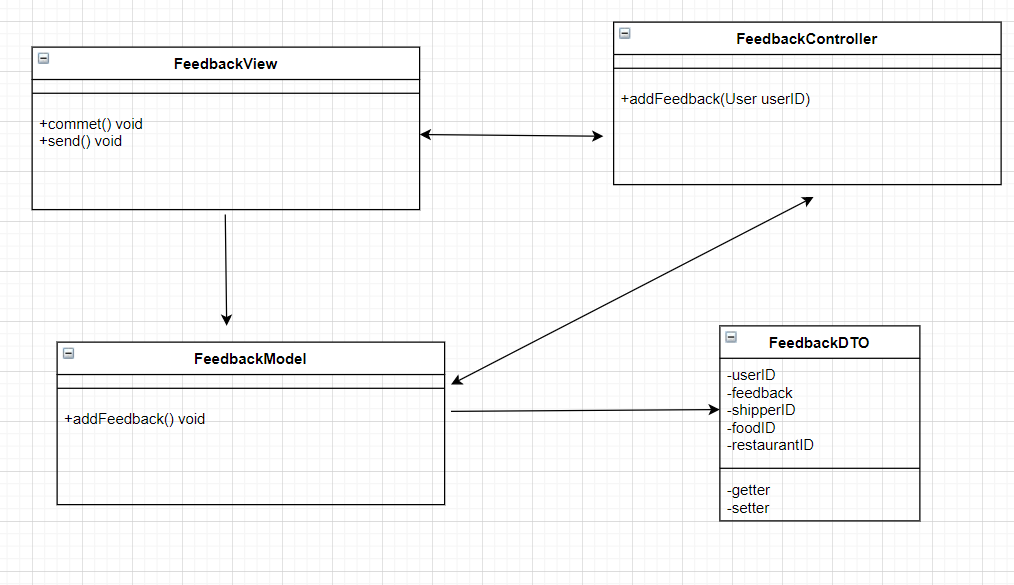
## Use Case Description

UC4: Feedback shipper, food and restaurant

|  |  |
| --- | --- |
| Use case Name | Feedback shipper, shop and restaurant |
| Brief description | This use case describes how the Customer can feedback shipper, shop and restaurant. |
| Actors | Customer |
| Basic Flow | 1. After ordering successful, the user can feedback 2. User enters comment in “Comment” field and rating star about shipper, restaurant or food 3. User clicks “Send feedback” 4. System displays “Thank you your feedback” |
| Pre-conditions | User is logged into the Food Hub system. |
| Post-conditions | The user can see their feedback. |

## Interaction (MVC) style

### Class Diagram



### Sequence Diagram

