

# Food Delivery Online System

---

Wentao Li; login: wentao1

Xi Chen; login: chenx5

## Overview

The proposed software system “food delivery online system” is an enterprise system in Australia. This system allows customers to order food online and helps restaurants process orders. Customers can search their favourite food and generate orders online, and then restaurants will receive the orders and process them. This system is helpful for restaurants to get order online and to handle the food delivery, thereby increasing profits.

Two main roles of the system: restaurant and customer

This system follows the definition of an enterprise system. It is involved in multiple users and features to the enterprise. It is used by food delivery business and would be helpful to manage the food delivery processes. It takes charge of critical resources, dishes and orders. Apparently, this system is able to support business goal that it makes profits and provide service to customers.

## Feature A – Menu Management

Restaurants can post new dishes to their menu on the system. Dishes in the menu can be view by both restaurants and customers. Restaurants can update the attributes of the dishes such as price and description. Restaurants can also delete their dishes.

Use Case ID	UC01
Actor	Restaurant user
Description	The restaurant user would like to post a new dish to the menu.
Procedures	<ol style="list-style-type: none"><li>1. The user logs into the system.</li><li>2. The user selects “New Dish”.</li><li>3. The user input the name, price, type, number in stock and description of the dish.</li><li>4. The user selects “Submit”.</li></ol>

Use Case ID	UC02
Actor	Restaurants
Description	The restaurant user would like to view the information of the dishes in their menu.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "My Menu"</li> <li>3. The user views the dishes list and select one dish.</li> <li>4. The user views the detailed information of the selected dish.</li> </ol>

Use Case ID	UC03
Actor	Restaurant user
Description	The restaurant user would like to update a dish in their menu.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "My Menu"</li> <li>3. The user views the dishes list and select one dish.</li> <li>4. The user selects "Modify".</li> <li>5. The user inputs the new value of attributes of the selected dish.</li> <li>6. The user selects "Submit".</li> </ol>

Use Case ID	UC04
Actor	Restaurant user
Description	The restaurant user would like to delete a dish in their menu.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "My Menu".</li> <li>3. The user views the dishes list and select one dish.</li> <li>4. The user selects "Delete".</li> </ol>

Use Case ID	UC05
Actor	Customer
Description	The customer would like to search for dishes.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "Find Food".</li> <li>3. The user optionally selects keywords, location, price or category.</li> <li>4. The user selects "Search".</li> <li>5. The user views the list of dishes.</li> </ol>

## Feature B – Order Management

Customers can create orders. Orders can be viewed and tracked by the customer and the restaurants. Restaurants can update the state of the order (new->admitted->delivered). Orders can be deleted by restaurants after the food is delivered. Orders can also be deleted by customers before admitted.

Use Case ID	UC06
Actor	Customer
Description	The customer would like to create an order.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects a dish.</li> <li>3. The user selects "Create Order".</li> <li>4. The user inputs address, phone number, amount of dishes and payment method.</li> <li>5. The user selects "Submit".</li> </ol>

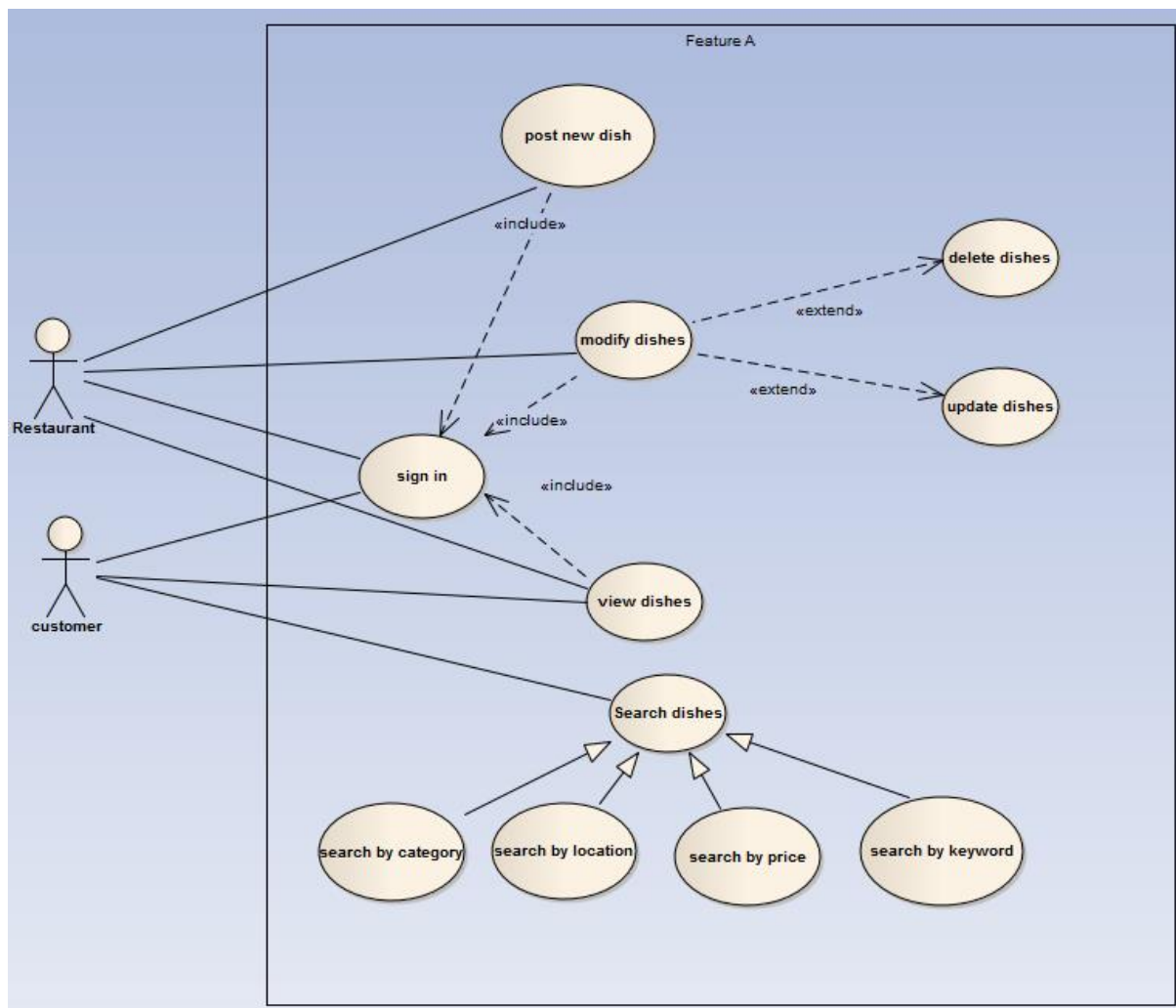
Use Case ID	UC07
Actor	Restaurant user/Customer
Description	The user would like to view their orders.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "My Order".</li> <li>3. The user views the list of their orders.</li> <li>4. The user selects one of the orders.</li> <li>5. The user views the detail of the selected order.</li> </ol>

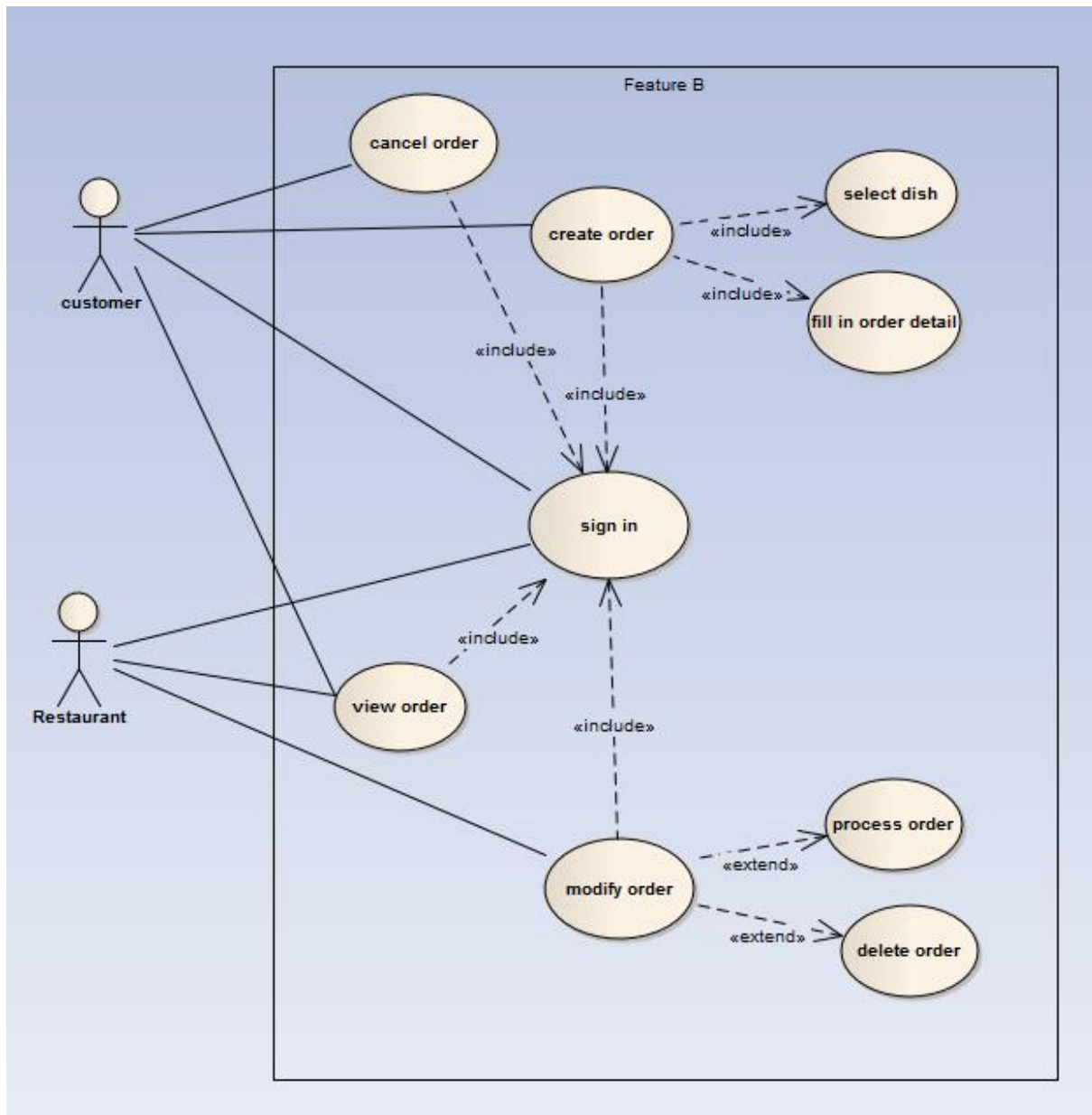
Use Case ID	UC08
Actor	Customer
Description	The customer would like to cancel an order.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "My Order".</li> <li>3. The user views the list of their orders.</li> <li>4. The user selects one of the orders of which state is "new".</li> <li>5. The user selects "Cancel Order".</li> </ol>

Use Case ID	UC09
Actor	Restaurant user
Description	The restaurant user would like to update the state of an order.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects "My Order".</li> <li>3. The user views the list of their orders.</li> <li>4. The user selects one of the orders.</li> <li>5. The user selects "Update Order"</li> <li>6. The user selects the state of the Order.</li> <li>7. The user selects "Submit".</li> </ol>

Use Case ID	UC10
Actor	Restaurant user
Description	The restaurant user would like to delete a completed order.
Procedures	<ol style="list-style-type: none"> <li>1. The user logs into the system.</li> <li>2. The user selects “My Order”.</li> <li>3. The user views the list of their orders.</li> <li>4. The user selects one of the orders of which state is “delivered”.</li> <li>5. The user selects “Delete Order”.</li> </ol>

Use case diagram:





## Properties of enterprise systems

### Persistent data:

Data in this system should be remain persistent. The data might be used to calculate long term profit. Although the price can be modified, the price on the order should not be changed before it is completed.

### A lot of Data:

This system can be used by many restaurant and customers. Therefore the amount of data can be very large. The system should use a database to manage the data.

### Concurrency access:

1. The restaurant is able to modify the in stock number of a dish. The number will be also modified when a customer submit an order. There would be a concurrency issue when a customer is trying to submit an order while the restaurant is modifying the in stock number or trying to delete the dish.
2. An order can be deleted by customer only when the state is "new". And the state can be changed by the restaurant. There would be a concurrency issue when a customer is trying to cancel an order while the restaurant is trying to change the state from "new" to "admitted".

### Many user interface:

Currently there are two user types with different use cases. The user interface would be different. There might be other user types for the system to expand such as 'system admin' or "express company".

### Build on business logic:

The process of the order is one kind of business logic. For example, a user can only cancel the order before it is admitted by restaurant. This process should be handled by the system.

### Integrate with other enterprise systems:

This system can be integrated with accounting systems (to calculation profit), or payroll system (to record workload of delivers).