Online Food Ordering System

Emir Sarı

June 4, 2023

1 Chapter 1: Project Definition, Use Cases and Project Management

1.1 Definition of Project

1.1.1 Project Purpose

This project aims to provide stakeholders with an automated online food ordering system that is beneficial to both customers and restaurants. Tangible and intangible benefits of the project are stated in the following table 1:

Tangible Benefits	**Intangible Benefits**
Increased Sales	Improved Customer Service
Increased Number of Customers	Improved Product Quality
Increased Order Accuracy	Improved Competitiveness
Reduced Delivery Delay	Better Delivery Speed
Reduced Labor Costs	Better Insights on Customer Behavior
Better Resource Management	

Table 1: Table of tangible and intangible benefits.

1.1.2 Project Scope

The project required the development team to build an up-to-date food ordering website which lets customers browse through a dynamic menu with plenty of options, organize and review their orders. For each item added to the order, prices will dynamically adjust based on modified options. Customers will be allowed to review their order in full detail before making the payment. After the payment is made, orders will be sent to restaurant staff for preparation and customers will be able to track the status of their order.

1.1.3 Project Constraints

- **Time constraints:** project may have a specific deadline.
- **Budget limitations:** both the development team and the restaurant may have limited financial resources at their disposal.
- **Resource availability:** availability of experienced developers, hardware and software systems.
- **Technological dependencies:** project may depend on certain technologies or platforms such as servers to run the website, new hardware to integrate the system as a real life application, and existing or new software systems.
- **Integration challenges:** developers may need to integrate the system with existing software, databases or other third-party services.
- **Scalability and performance issues:** the system may need to deal with a growing number of users and interactions.

- **Stakeholder expectations:** project needs to align with the expectations of stakeholders such as customers, restaurant owners and project managers.

1.2 Actor Glossary

1. **Customer:** Places food orders using the website.
2. **Restaurant Staff:** Receives and processes orders from customers.
3. **Distributor:** Delivers orders to customers.
4. **Administrator:** Manages the system and updates the menu.

1.3 Use Cases

1. **Browse Menu:** Customer views the menu.
2. **Place Order:** Customer selects items from the menu, creates an order by adding/removing items, customizing their options and finally making the payment.
3. **Review Order:** Customer views and modifies order details before proceeding to payment.
4. **Make Payment:** Customer pays the total cost of the order.
5. **Process Order:** Restaurant staff receives the order and prepares the necessary items.
6. **Deliver Order:** Distributor delivers the order to the appropriate customer.
7. **Manage Menu:** Administrator makes necessary changes to the menu items, prices and options to ensure system reliability.

1.4 Project Effort Estimation

Based on table 2, Unadjusted Actor Weighting Table, for each group of actors, result can be calculated with (result = weighting factor * number of actors).

- Result(SimpleActors) = 1 * 0 = 0
- Result(AverageActors) = 2 * 0 = 0
- Result(ComplexActors) = 3 * 4 = 12

At the end, UAW is obtained by adding each group's result as follows: Unadjusted Actor Weight Total (UAW) = 0 + 0 + 12 = 12.

Actor Type	Description	Weighting Factor	Number	Result
Simple	An external system using well-defined API	1	0	0
Average	An external system which has a protocol-based interface.	2	0	0
Complex	A human	3	4	12

Table 2: Unadjusted Actor Weighting Table

Each use case is classified according to the following transactions:

- Browse menu: load menu, display menu (2 transactions)
- Place order: browse menu, add item, remove item, customize item, review order, make payment, process order (7 transactions)
- Review order: view order details, change order details (2 transactions)
- Make payment: display total order cost, take customer payment info, take customer address info, confirm customer payment, notify system upon payment (5 transactions)
- Process order: send order details to restaurant staff, display order preparation status (2 transactions)
- Deliver order: display customer address info, display order delivery status (2 transactions)
- Manage menu: display menu, modify menu items, save menu modifications (3 transactions)

Based on table 3, Unadjusted Use-Case Weighting Table, for each group of use cases, results can be calculated with (result = weighting factor * number of use cases).

- Result(SimpleUseCases) = 5 * 5 = 25
- Result(AverageUseCases) = 10 * 2 = 20
- Result(ComplexUseCases) = 15 * 0 = 0

At the end, UUCW is obtained by adding each group's result as follows: Unadjusted Use Case Weight Total (UUCW) = 25 + 20 + 0 = 45.

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1 to 3 transactions	5	5	25
Average	4 to 7 transactions	10	2	20
Complex	more than 7 transactions	15	0	0

Table 3: Unadjusted Use-Case Weighting Table

Considering results obtained from tables 2 and 3, Unadjusted Use-Case Points (UUCP) = UAW + UUCW = 12 + 45 = 57.

In tables 4 and 5, for each factor, weighted values can be calculated with (weighted value = weight * assigned value). Assigned values were determined based on the importance of the particular factor in this project.

- Weighted Value(T1) = 2*0 = 0 Weighted Value(T2) = 1*5 = 5
- Weighted Value(T3) = 1*3 = 3 Weighted Value(T4) = 1*1 = 1
- Weighted Value(T5) = 1*1 = 1 Weighted Value(T6) = 0.5*2 = 1
- Weighted Value(T7) = 0.5*4 = 2 Weighted Value(T8) = 2*3 = 6
- Weighted Value(T9) = 1*5 = 5 Weighted Value(T10) = 1*0 = 0
- Weighted Value(T11) = 1*2 = 2 Weighted Value(T12) = 1*0 = 0
- Weighted Value(T13) = 1*1 = 1

By adding weighted values of all technical factors together, TFactor is calculated: Technical Factor Value (TFactor)= 0+5+3+1+1+1+2+6+5+0+2+0+1 = 27, thus
Technical Complexity Factor (TCF)= 0.6 + (0.01*TFactor) = 0.6 + (0.01*27) = 0.87

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
T1	Distributed system	2.0	0	0	
T2	Response time or throughput performance objectives	1.0	5	5	
T3	End-user online efficiency	1.0	3	3	
T4	Complex internal processing	1.0	1	1	
T5	Reusability of code	1.0	1	1	
T6	Ease of installation	0.5	2	1	
T7	Ease of use	0.5	4	2	
T8	Portability	2.0	3	6	
T9	Ease of change	1.0	5	5	
T10	Concurrency	1.0	0	0	
T11	Special security objectives included	1.0	2	2	
T12	Direct access for third parties	1.0	0	0	
T13	Special user training required	1.0	1	1	

Table 4: Technical Complexity Factors

- Weighted Value(E1) = 1.5*4 = 6 Weighted Value(E2) = 0.5*4 = 2
- Weighted Value(E3) = 1*4 = 4 Weighted Value(E4) = 0.5*5 = 2.5
- Weighted Value(E5) = 1*3 = 3 Weighted Value(E6) = 2*5 = 10
- Weighted Value(E7) = -1*0 = 0 Weighted Value(E8) = -1*5 = -5

By adding weighted values of all environmental factors together, EFactor is calculated: Environmental Factor Value (EFactor)= 6+2+4+2.5+3+10+0+(-5) = 22.5, thus
Environmental Factor (EF)= 1.4 + (-0.03*EFactor) = 1.4 + (-0.03*22.5) = 0.725

To conclude, considering results obtained in each section, effort in person-days can be calculated by determining the PHM (person-hours multiplier) and then converting it to a PDM (person-days multiplier):

Factor Number	Description	Weight	Assigned Value (0–5)	Weighted Value	Notes
E1	Familiarity with system development process being used	1.5	4	6	
E2	Application experience	0.5	4	2	
E3	Object-oriented experience	1.0	4	4	
E4	Lead analyst capability	0.5	5	2.5	
E5	Motivation	1.0	3	3	
E6	Requirements stability	2.0	5	10	
E7	Part-time staff	-1.0	0	0	
E8	Difficulty of programming language	-1.0	5	-5	

Table 5: Environmental Factors

Following the rule,
"If $\sum$ (number of Efactors E1 through E6 assigned value < 3) and (number of Efactors E7 and E8 assigned value > 3) ≤ 2, PHM $= 20$", considering values provided in table 5, $0+1 = 1 \leq 2$. Thus, PHM $= 20$.

Average working hours per day is known to be 8 hours, thus PDM can be obtained as follows:
PDM = PHM / hours-per-day = 20 / 8 = 2.5
Adjusted Use Case Points (UCP) = UUCP * TCF * ECF = 57 * 0.87 * 0.725 = 35.9528
Effort in person-days = UCP * PDM = 35.9528 * 2.5 = 89.882

Gantt Chart for the development of Online Food Ordering System is given in figure 1:

ID	Task Name	Duration	Start	Finish	Prede
1	Identify restaurants	2 weeks	Tue 16.05.23	Mon 29.05.23	
2	Compare restaurants	2 weeks	Tue 30.05.23	Mon 12.06.23	1
3	Negotiate with restaurants	3 weeks	Tue 13.06.23	Mon 03.07.23	2
4	Analyze results and choose restaurant	2 weeks	Tue 04.07.23	Mon 17.07.23	3
5	Develop new website	9 weeks	Tue 30.05.23	Mon 31.07.23	1
6	Deploy Website	1 week	Tue 01.08.23	Mon 07.08.23	4,5
7	Train Restaurant Staff	1 week	Tue 01.08.23	Mon 07.08.23	5
8	Test & Improve System Performance	3 weeks	Tue 08.08.23	Mon 28.08.23	6,7
9	Budget Meeting	1 day	Tue 30.05.23	Tue 30.05.23	
10	Hardware & Software Installation	1 day	Tue 01.08.23	Tue 01.08.23	

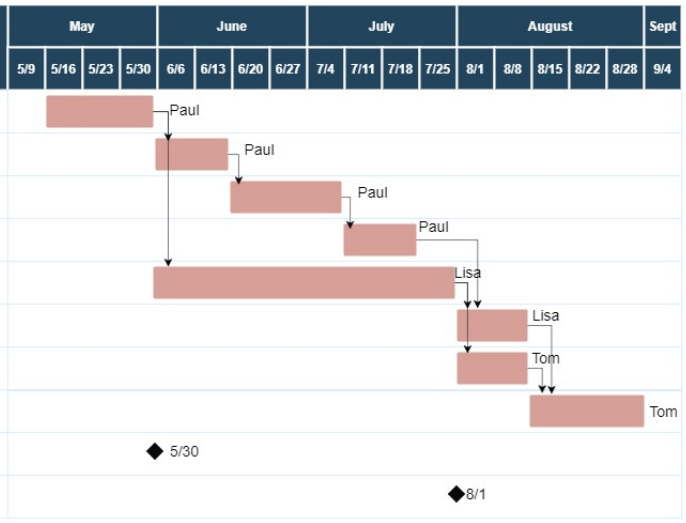

Figure 1: Gantt Chart

2 Chapter 2: Requirements Analysis

2.1 Non-functional Requirements

1. Website Performance Requirements

   1.1 The website will load and display the menu within 2 seconds, providing a responsive and reliable browsing experience for customers.

   1.2 The website will handle at least 400 users at the same time without performance issues, to ensure smooth interactions during busy hours.

2. Website Security Requirements

   2.1 The system will use industry-standard software to protect customers' private data such as payment and address information, during all transactions.

   2.2 Only administrators with valid credentials can modify the contents of the website.

2.2 Functional Requirements

1. Place Order

   1.1 The system should allow customers to select and add, customize or remove items from the order while providing price updates according to customer modifications.

   1.2 The system should save any changes made to the customer's order in real-time and when needed, it should display order details reliably both for the customer and restaurant staff.

2. Process Order

   2.1 The system should automatically notify restaurant staff when a new order arrives. The system should display in-depth order details including customer preferences and customers' special requests if there are any.

   2.2 The system should provide both customers and restaurant staff with means to track order status in real-time.

3 Chapter 3: Functional Modeling

3.1 Use-case Diagram

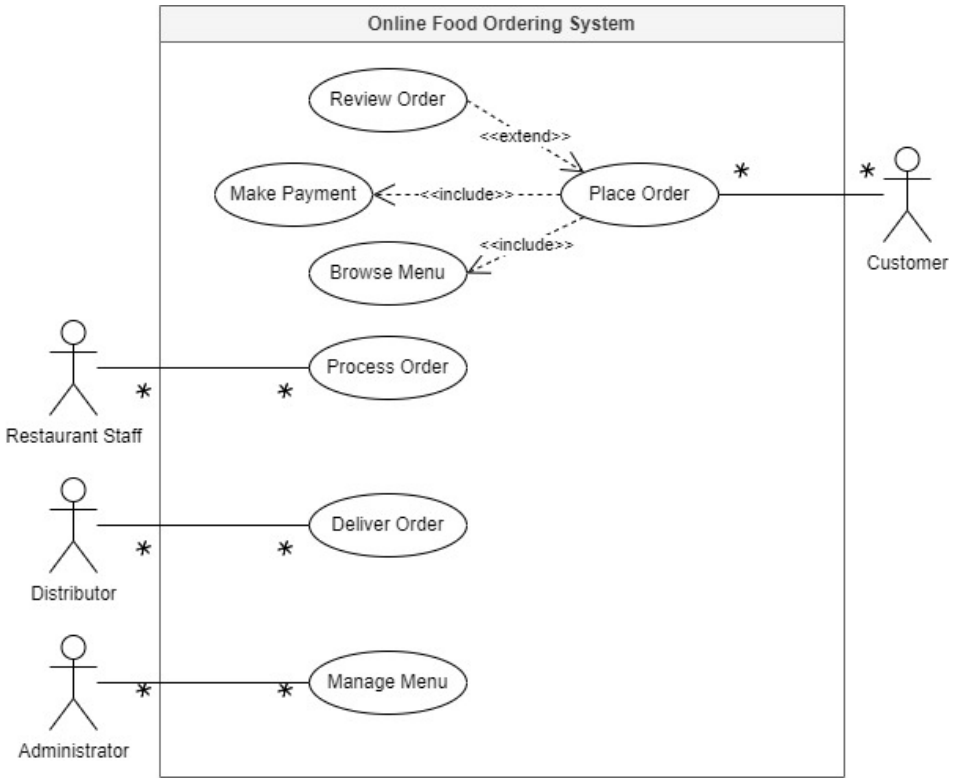

Figure 2: Online Food Ordering System Use-Case Diagram

- The actor Customer is associated with the Place Order use case which extends its functionality with Review Order use case and includes Browse Menu and Make Payment use cases.
- Browse Menu use case provides vital functionality to the system that lets the customer view available items in the menu and visually add or remove items in the order.
- Some functionality of Review Order use case is occasionally needed, when the customer wants to modify or simply view the order in full detail.
- Make Payment use case is another vital part of the system providing functionality to get the payment and address information of the customer and validate the transaction.
- The actor Restaurant Staff is associated with the Process Order use case, which lets the restaurant staff receive and process customer orders accurately.
- The actor Distributor is associated with the Deliver Order use case, which stores customer's address info., keeps track of the distributor's position and notifies the customer accordingly.
- The actor Administrator is associated with the Manage Menu use case, which provides functionality to modify menu items.

3.2 Use-case Descriptors

Tables 6 and 7 show "Place Order" use case's description.

Use Case Name: Place Order	**ID:** 2	**Importance Level:** High
Primary Actor: Customer	**Use Case Type:** Detail, Essential	

Table 6: Use-Case Description for Place Order

3.3 Activity Diagrams

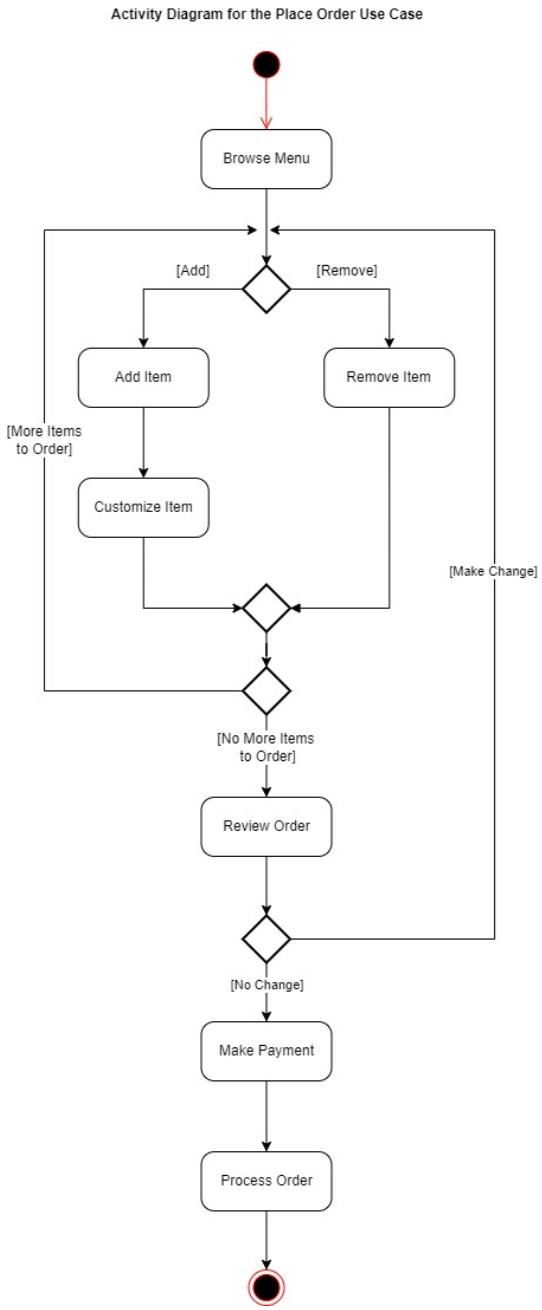

Figure 3: Activity Diagram

- Browse Menu activity represents the customer browsing through the menu and interacting with available items.
- Add Item activity occurs when the customer wants to add an item to the order. It involves selecting an item from the menu and specifying its quantity.
- Customize Item activity occurs for each instance of an item in the order and lets the customer to modify several options of the item.
- Remove item activity occurs when the customer wants to remove an item from the order. It involves selecting an item from the order and removing it.

- Review Order activity occurs when the customer has completed the order. It allows the customer to see order details and make further decisions.
- Make Payment activity involves requesting payment and address information from the customer, as well as checking payment status.
- Process Order activity involves conveying the order to the restaurant staff for preparation.

4 Chapter 4: Behavioral Modeling

4.1 Sequence Diagram 1

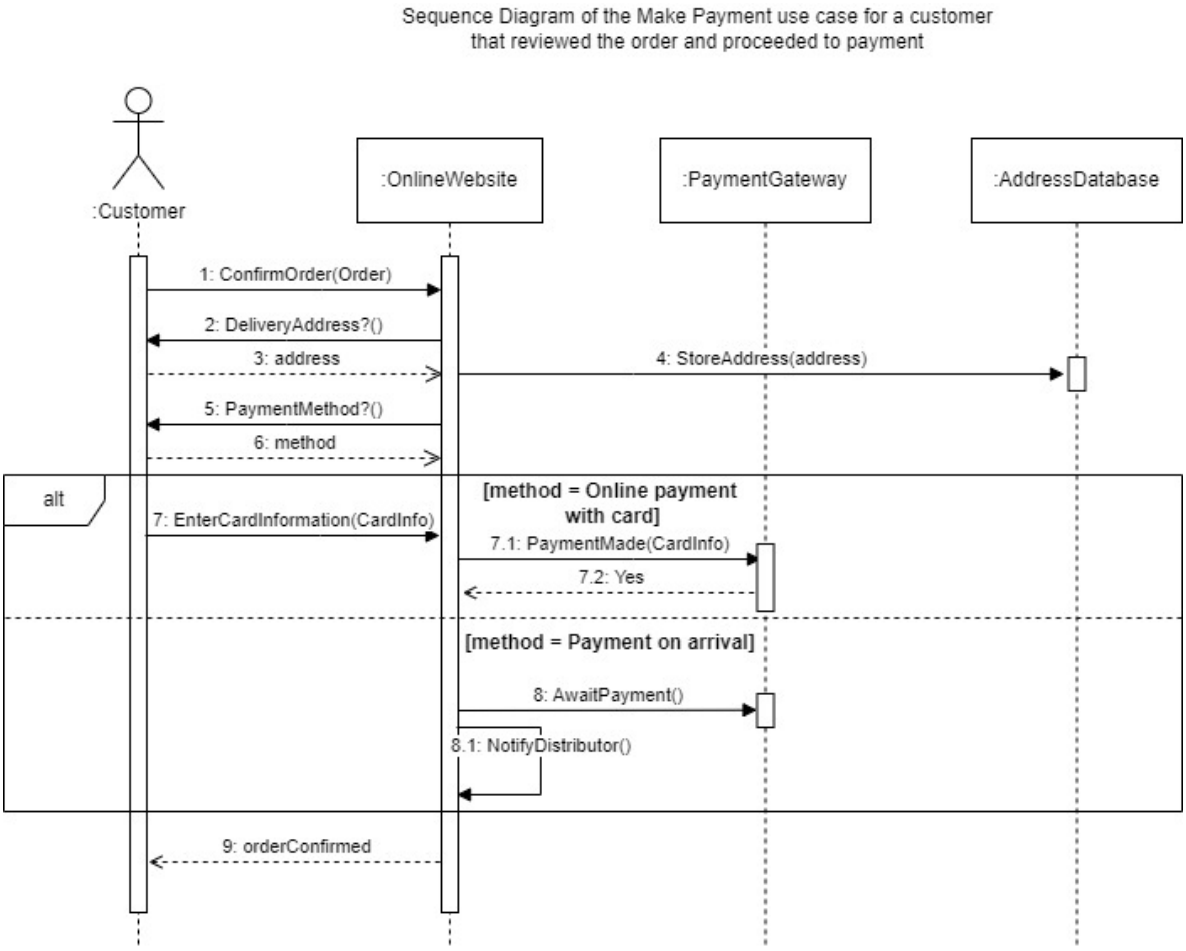

Figure 4: Scenario 1, Sequence Diagram 1

Figure 4 represents the scenario involving Make Payment use case, where the customer has reviewed the order and proceeded to make payment.

When customer proceeds to payment, the system first asks for customer address and stored it in the address database. Then, the online website system asks about customer's preferred method of payment:
If payment will be done online, the system asks for credit card information and makes necessary payment processes;
If payment will be done on arrival, the system marks the order as "awaiting payment" and informs the distributor accordingly. Finally, the system conveys the customer that the order was confirmed.

4.2 Sequence Diagram 2

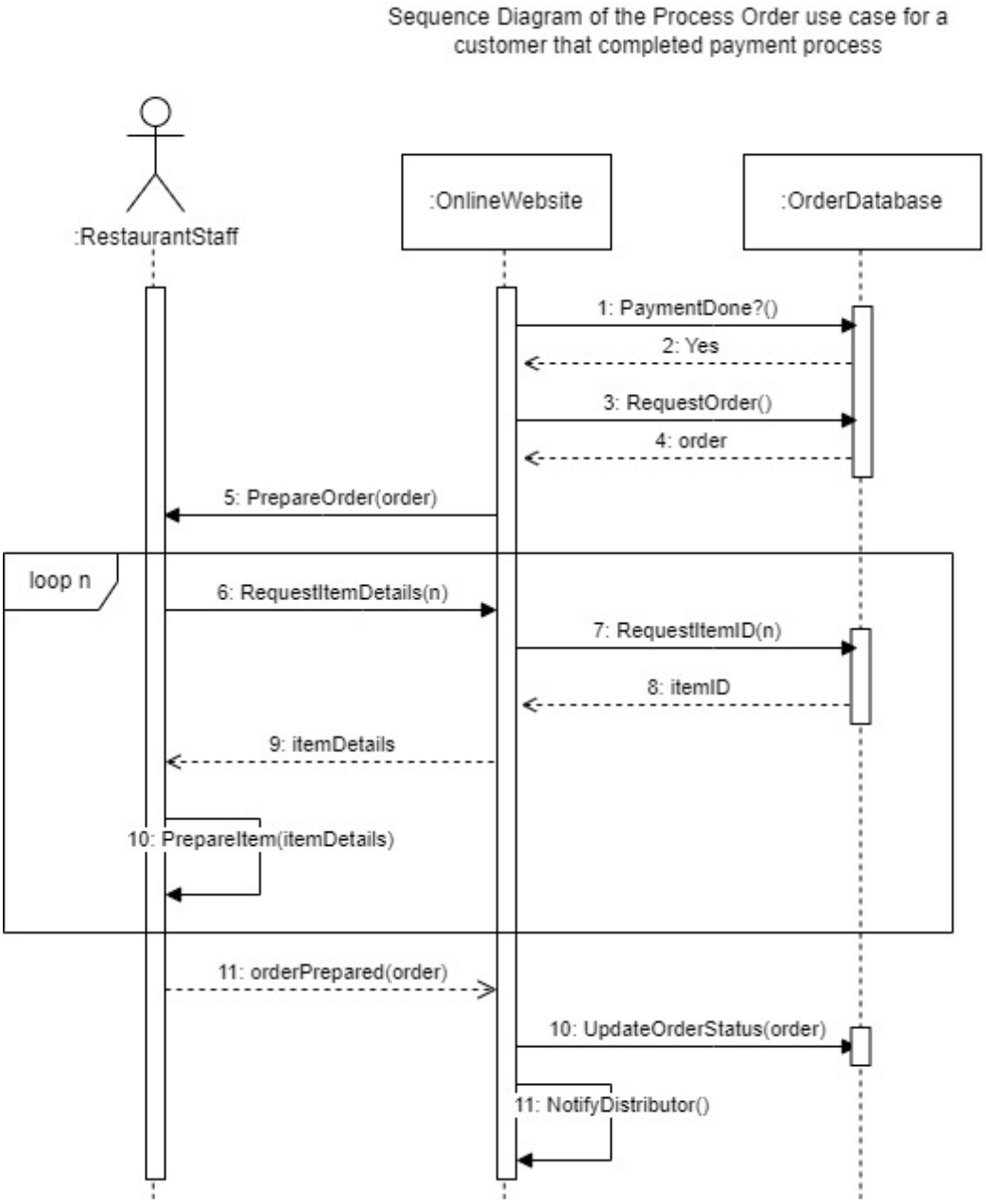

Figure 5: Scenario 2, Sequence Diagram 2

Figure 5 represents the scenario involving Process Order use case, where the customer has completed the payment and the order will be processed.

When Process Order use case is executed, the online website system first gets payment confirmation from order database. In this scenario, since the payment is done, the system requests order information from order database. Order database provides the system with the proper order, then the system redirects the order to restaurant staff for preparation. For each item in the order, restaurant staff

1. Requests item details from the system.
2. The system request item ID from order database to get the matching item.

3. Order database returns item ID to the system.
4. The system returns and displays item details to restaurant staff.
5. Restaurant staff begins preparing the item.

Finally, when the entire order is prepared, restaurant staff notifies the system which will update order status to "on delivery" and will notify available distributors for order delivery.

Stakeholders and Interests:
Administrator - ensures menu items and options are up-to-date, manages the system.
Customer - browses the menu; adds, customizes or removes items; reviews and places an order.
Restaurant Staff - receives and processes the customer's order accurately and swiftly.

Brief Description:
Customer interacts with the website to browse the menu, create an order; add, customize or remove items and make payment.

Trigger: Customer opens the website and initiates the process of placing an order by browsing the menu.
Type: External

Relationships:
Association: Customer
Include: Browse Menu, Make Payment
Extend: Review Order
Generalization: -

Normal Flow of Events:

1. Customer accesses the online food ordering system using the website.

   Browse Menu use case is executed, the customer browses through the menu to view items.

2. Customer begins placing an order while browsing the menu.

3. Using visual guidelines, the system asks the customer if they would like to add or remove an item.

   3.1 If the customer wants to add an item to their order, S-1: Add Item and S-2: Customize Item subflows are performed consecutively.

   3.2 If the customer wants to remove an item from their order, S-3: Remove Item subflow is performed.

   3.3 Step 3 is repeated as long as the customer has more items to order.

4. If the customer has no more items to order

   Review Order use case is executed, the customer reviews order details.

5. If the customer wants to make a change in the order

   Step 3 is executed.

6. If the customer doesn't want to modify the order any further

   Make Payment use case is executed, the customer proceeds to payment.

7. Process Order use case is executed, the order is sent to the restaurant staff for preparation.

SubFlows:

- S-1: Add Item
  1. The system asks the customer to select an item from the menu and specify its quantity.
  2. Customer selects an item and adds it to the order, specifying its quantity.
- S-2: Customize Item
  1. For each instance of an item in the order, the system asks the customer to customize the item according to the customer's preferences.
  2. Customer modifies each instance of the item by selecting flavors, toppings, extras, size and providing any special requests.
- S-3: Remove Item
  1. The system asks the customer to choose an item from the order that the customer wishes to remove.
  2. Customer selects an item from their order and removes it.

Table 7: Use-Case Description for Place Order