Food hub

### ****Introduction****

The **Food Hub** project is an innovative web-based application designed to provide a seamless and efficient platform for food delivery services. It serves as an intermediary connecting three primary entities: **customers**, **restaurants**, and **administrators**. Customers can browse menus, place orders, and track their deliveries. Restaurants can manage their menus, process incoming orders, and update order statuses. Administrators oversee the entire operation, ensuring smooth interactions and resolving issues when necessary.

The platform is built with a customer-centric approach, prioritizing ease of use, fast order processing, and real-time communication. The application integrates robust functionality such as menu browsing, order placement, payment processing, order tracking, and cancellation management. By leveraging modern technologies and adhering to best practices in software engineering, **Food Hub** aims to revolutionize the food delivery experience for all stakeholders involved.

This document outlines the key use cases of the system, focusing on core functionalities completed in the initial development iteration. These include browsing menus, placing orders, and handling cancellations, ensuring the foundational features of the system are fully operational.

With its intuitive interface and efficient backend, **Food Hub** is poised to become a trusted and reliable food delivery solution.

Use Case 1: Browse Menu

#### ****Description****:

This use case allows customers to explore the food items available for ordering. Customers can view details like name, description, price, and availability. This functionality is essential for helping customers decide what to order.

#### ****Actors****:

* **Customer**
* **System**

#### ****Preconditions****:

1. The system must have restaurants with their menu items added to the database.
2. The customer must be logged in to access the menu.

#### ****Steps****:

1. The customer navigates to the "Browse Menu" page.
2. The system retrieves the list of available restaurants and their corresponding menus.
3. The customer selects a restaurant or a food category to view specific items.
4. The system displays the food items with details such as:
   * Item name
   * Description
   * Price
   * Image (if available)
   * Availability status
5. The customer can:
   * Apply filters (e.g., price range, dietary preferences).
   * Search for specific items.
   * Add selected items to their cart.

#### ****Postconditions****:

1. Selected items are added to the customer’s cart.
2. The customer can proceed to place an order or continue browsing.

### ****Use Case 2: Place Order****

#### ****Description****:

This functionality enables customers to review their cart, confirm the selected items, provide delivery details, and make payments to place an order.

#### ****Actors****:

* **Customer**
* **System**
* **Restaurant**

#### ****Preconditions****:

1. The customer must have items added to their cart.
2. The system must be configured with available payment gateways and valid delivery address formats.

#### ****Steps****:

1. The customer navigates to the "Cart" page and reviews the selected items.
2. The system displays:
   * List of items
   * Quantities and prices
   * Total amount (including taxes and delivery charges)
3. The customer confirms the delivery address or enters a new one.
4. The customer selects a payment method and completes the payment.
5. The system processes the payment and provides a confirmation.
6. The order is assigned to the appropriate restaurant, and notifications are sent to the:
   * Restaurant: to prepare the order.
   * Customer: to confirm the order and provide an estimated delivery time.

#### ****Postconditions****:

1. The order status is set to "Processing."
2. The customer can track the order.

### ****Use Case 3: Cancel Order****

#### ****Description****:

This functionality allows customers to cancel their order under eligible conditions (e.g., before it is dispatched). The admin reviews the cancellation request, and a refund is initiated if applicable.

#### ****Actors****:

* **Customer**
* **Admin**
* **System**
* **Restaurant**

#### ****Preconditions****:

1. The customer must have placed an order.
2. The order must not be in the "Dispatched" or "Delivered" status.

#### ****Steps****:

1. The customer navigates to the "Order History" section.
2. The customer selects the order they want to cancel and initiates a cancellation request.
3. The system checks the order status:
   * If the order is in "Processing" or "Preparation," cancellation is allowed.
   * If the order is "Dispatched" or "Delivered," cancellation is denied.
4. The admin receives the request and reviews it.
5. If the request is approved:
   * The system updates the order status to "Canceled."
   * The restaurant is notified to stop preparing the order (if applicable).
   * Refund processing is initiated.
6. The customer receives a notification of the cancellation outcome.

#### ****Postconditions****:

1. If approved, the order is canceled, and the refund process begins.
2. The customer’s order history reflects the updated status.

Summary of Core Use Cases:

|  |  |  |  |
| --- | --- | --- | --- |
| | **Use Case** | | --- | | Actors | Description |
| Browse Menu | Customer, System | Enables customers to explore restaurant menus and add items to the cart. |
| Place Order | Customer, System, Restaurant | Allows customers to review their cart, provide delivery details, and confirm their order. |
| Cancel Order | Customer, Admin, System, Restaurant | Allows customers to cancel an order if eligible, with admin approval and refund processing. |

