Contents

[Project Specification: 3](#_Toc131143181)

[Functional Requirements: 3](#_Toc131143182)

[Use Case Model 1: 3](#_Toc131143183)

[UML Use Case Diagrams: 5](#_Toc131143184)

[Supplementary Specification: 5](#_Toc131143185)

[Non-functional Requirements: 5](#_Toc131143186)

[Design Constraints: 5](#_Toc131143187)

[Glossary: 5](#_Toc131143188)

# Project Specification:

This document outlines the specifications of a software system that is being developed for a restaurant. The system aims to provide an efficient way for customers to place their orders, and for restaurant staff to manage those orders, update the menu, and process payments.

## Functional Requirements:

* Allow customers to browse the restaurant menu and place orders.
* Allow customers to choose their preferred payment method (cash, credit card, etc.).
* Notify staff of new orders in real-time.
* Allow staff to update the menu, add new items, and remove items.
* Provide staff with an easy-to-use interface for managing orders, including marking them as completed.
* Generate a receipt for the customer once the order is complete.
* Generate sales reports for the restaurant management team.

## Use Case Model 1:

Use Cases Identification:

* Use-Case: Place Order
* Level: Primary
* Primary Actor: Customer
* Main success scenario: Customer selects items from the menu, adds them to the cart, enters payment information, and submits the order.
* Extensions:

1. If the customer selects an item that is not available, they are notified and prompted to remove it from the cart.
2. If the customer enters an invalid payment method or insufficient funds, they are notified and prompted to enter valid information.

* Use-Case: Submit Order
* Level: User Goal
* Primary Actor: Customer
* Main Success Scenario:

1. The customer selects items from the menu.
2. The customer specifies any special instructions or requests for the order.
3. The customer confirms the order and submits payment.
4. The system confirms that the payment was successful.
5. The system sends a notification to the restaurant with the details of the order.
6. The restaurant prepares the order.
7. The system notifies the customer when the order is ready for pickup or delivery.

* Use-Case: Manage Inventory
* Level: Subfunction
* Primary Actor: Restaurant Manager
* Main Success Scenario:

1. The manager selects the inventory management feature from the main menu.
2. The system displays the current inventory levels and quantities.
3. The manager selects an item to update.
4. The manager enters the new quantity for the item.
5. The system updates the inventory record.
6. The system displays the updated inventory levels.
7. The manager saves the changes to the inventory.

* Use-Case: Track Order Status
* Level: User Goal
* Primary Actor: Customer
* Main Success Scenario:

1. The customer selects the "Track Order" option from the main menu.
2. The system prompts the customer to enter their order ID or phone number.
3. The customer enters the requested information.
4. The system displays the status of the customer's order (e.g. "Preparing," "Out for Delivery," "Delivered").
5. The system provides an estimated delivery time, if applicable.
6. The customer closes the tracking window.

# UML Use Case Diagrams:

Diagram

Description automatically generatedThe UML Use Case Diagram will include a Use Case for the primary actor, Customer, to place an order.

# Supplementary Specification:

## Non-functional Requirements:

* Performance: The system must be able to handle a large number of orders and users without any significant delays or downtime.
* Security: The system must be secure and protect customer payment information.
* Usability: The system must be easy to use for both customers and staff, with clear instructions and intuitive interfaces.
* Availability: The system must be available 24/7, with minimal downtime for maintenance or updates.

## Design Constraints:

* The system must be developed using Java and Spring Boot.
* The database management system must be MySQL.
* The front-end must be developed using HTML, CSS, and JavaScript.
* The system must be hosted on AWS.

# Glossary:

* Customer: A user who visits the restaurant and places an order.
* Staff: Employees of the restaurant who manage the orders and the menu.
* Order: A request for food and/or drinks from a customer.
* Cart: A list of items that the customer has selected to order.
* Payment Method: The method that the customer uses to pay for their order (cash, credit card, etc.).
* Sales Report: A report that shows the total sales for a given period of time.