Project requirements

Contents

[Introduction 2](#_Toc183553002)

[1. Project Overview 3](#_Toc183553003)

[2. Functional Requirements 3](#_Toc183553004)

[2.1 User Roles and Permissions 3](#_Toc183553005)

[Customers: 3](#_Toc183553006)

[Restaurant Managers: 3](#_Toc183553007)

[Admins: 3](#_Toc183553008)

[2.2 Core Features 4](#_Toc183553009)

[Authentication: 4](#_Toc183553010)

[Restaurant and Menu Management: 4](#_Toc183553011)

[Order Management: 4](#_Toc183553012)

[Payment Integration: 4](#_Toc183553013)

[Search: 4](#_Toc183553014)

[3. Non-Functional Requirements 4](#_Toc183553015)

[Performance: 4](#_Toc183553016)

[Scalability: 4](#_Toc183553017)

[Security: 4](#_Toc183553018)

[Usability: 4](#_Toc183553019)

[Reliability: 5](#_Toc183553020)

[4. Use Case Scenarios 5](#_Toc183553021)

[5. User Stories 5](#_Toc183553022)

[6. Use case diagrams 6](#_Toc183553023)

# Introduction

Mazaq Coffee now has an online ordering website designed for users to get their favorite drinks effortlessly. With an intuitive interface, the website allows customers to browse the menu, place orders, and make payments in an instant. The app also provides managers with tools to manage orders and menu items efficiently while supporting delivery agents handling food delivery.

# Project Overview

* **Project Name:** Mazaq Coffee
* **Purpose:** A web application for users to browse the menu, order online, and make payments online or cash on delivery of food.
* **Scope:**
  + Customers register, browse menus, and place orders.
  + Restaurant owners manage menu items, and view orders.
  + Admins manage users, restaurants, and orders.

# 2. Functional Requirements

## 2.1 User Roles and Permissions

### Customers:

* Register and log in to their account.
* Browse menu.
* Add items to the cart.
* Place orders.
* View order history.

### Restaurant Managers:

* Log in.
* Add, edit, or delete menu items.
* View and manage orders.

### Admins:

* Manage user accounts.
* Oversee orders and payment issues.

## 2.2 Core Features

### Authentication:

* User registration, login, and logout.

### Restaurant and Menu Management:

* Browse menus listings with filters.

### Order Management:

* Add items to the cart and modify cart.
* View total price.
* Place orders and receive order confirmation.

### Payment Integration:

* Support for online payments.
* Display payment success/failure notifications.

### Search:

* Search for restaurants by name or cuisine.

# 3. Non-Functional Requirements

## Performance:

* Average response time for pages should be < 5 seconds.

## Scalability:

* The architecture should support future scaling to more users.

## Security:

* Store passwords securely.

## Usability:

* Friendly design with responsive layouts.
* User takes no more than 10 minutes to learn and get comfortable with using the app.
* Intuitive navigation and clear, easily understood error messages.

## Reliability:

* The app has downtime <10 minutes in a day .

# 4. Use Case Scenarios

* A customer registers, browses a menu, places an order, and pays.
* A restaurant manager updates their menu and confirms orders.
* An admin deletes user account.

# 5. User Stories

* **As a user**, I want to sign up so that I can create an account.
* **As a user**, I want to log in so that I can access my account, and place orders.
* **As a user**, I choose whether I want to pay online or cash on delivery.
* **As a restaurant manager**, I want to edit menu items.
* **As an admin**,I want to delete fraudulent account, or remove closed restaurants.

# 6. Use case diagrams

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a diagram

Description automatically generated