

# TEAM FOODEAZE

## Software Requirement Specification.

Mansi Singh -VU22CSEN0100144

Raj Kunwar- VU22CSEN0100108

N Sai Harini- VU22CSEN0100141

Shikha Yadav-VU22CSEN0101681

### Introduction

- **Purpose:**
- This document describes the software requirements for the FoodEaze online food ordering system. It outlines functional and non-functional requirements, hardware and software specifications, and overall system goals.
- **Scope:**

FoodEaze will allow users to browse food menus, place orders, make payments, track deliveries, and provide feedback. It also includes admin functions for managing restaurants, menus, and orders.

- **Definitions, Acronyms, and Abbreviations:**
  - SRS: Software Requirement Specification
  - UI: User Interface
  - API: Application Programming Interface
- **References:**

Reference any industry standards, books, or prior work.

### System Overview:

The system will support a web and mobile application for customers and an admin portal for restaurant owners. It also integrates with payment gateways and delivery tracking systems.

### Overall Description:

- **Product Perspective:**
- FoodEaze is an independent system but will integrate with third-party systems like payment gateways (e.g., PayPal, Stripe), and delivery services.
- **Product Features:**
  - Browse food items and menus



- Online ordering
- Secure payment
- Real-time delivery tracking
- Admin and vendor control panels
- **User Characteristics:**
  - Customers: Individuals who use the system to order food.
  - Restaurant Admin: Users who manage menus, orders, and deliveries.
  - System Admin: Users who manage the platform, vendors, and users.
- **Dependencies:**

The system is dependent on third-party services like payment and delivery APIs.

## Assumptions and Dependencies:

- Assumes users have access to the internet.
- It depends on third-party services for payments and delivery.



## Software Requirements for FoodEaze

**1.) Functional Requirements:** Functional requirements describe the core functions that the FoodEaze system must perform.

### User Registration & Authentication:

- Users (customers, restaurant owners, delivery partners) must be able to register with email, phone number, or social login.
- Users can log in and out securely with username and password or OAuth (Google, Facebook).
- Password reset functionality for users.

## **2. Browse and Search Menus:**

- a. Customers should be able to browse through available restaurants.
- b. Filters like cuisine type, price range, delivery time, or location must be available for searching.
- c. Restaurants should have searchable menus with images, descriptions, and pricing for each food item.

## **3. Shopping Cart:**

- a. Customers can add items to the cart.
- b. They can modify the order (add/remove items, adjust quantities) before placing it.
- c. A summary of the order must be shown before checkout, including tax, delivery charges, and discounts.

## **4. Order Placement & Payment Processing:**

- a. Customers can proceed to checkout, confirm their delivery address, and make payments.
- b. Multiple payment options (credit/debit cards, PayPal, Google Pay, etc.) should be integrated.
- c. Customers receive order confirmation with an estimated delivery time.

## **5. Delivery Tracking:**

- a. Real-time order status updates (order confirmed, being prepared, out for delivery, delivered).
- b. Integration with Google Maps API or similar for tracking the delivery in real time.

## **6. Review and Ratings:**

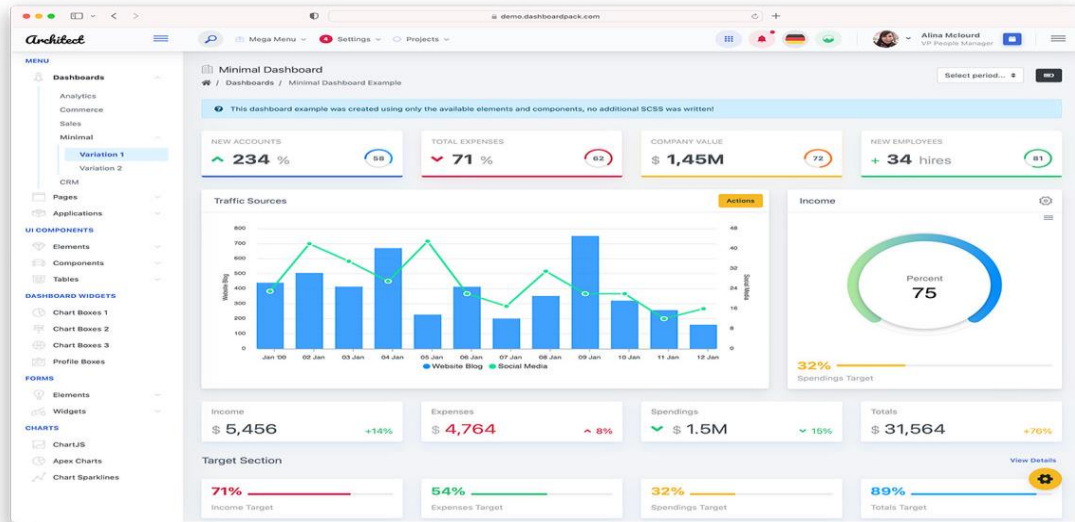
- a. After receiving an order, customers can rate the restaurant and delivery experience.
- b. Users can also provide textual feedback on food quality, delivery time, and overall service.

## **7. Restaurant and Menu Management (for Vendors):**

- a. Restaurant owners can manage their profiles (operating hours, address, contact details).
- b. Menus can be updated regularly (adding new items, setting prices, marking items as unavailable).
- c. Restaurants can view daily, weekly, or monthly sales reports.

## **8. Admin Dashboard:**

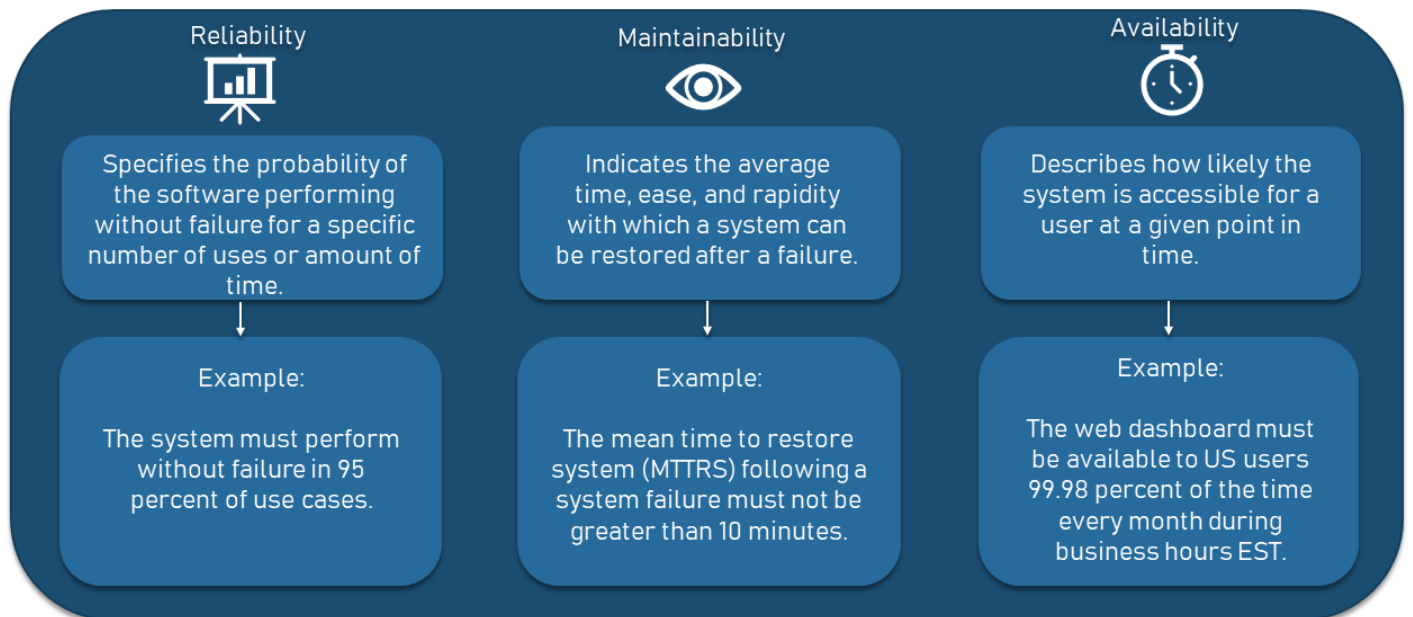
- a. System admins can manage restaurants, delivery personnel, and customer accounts.
- b. Admins can view analytics, resolve customer complaints, and handle disputes between restaurants and customers.
- c. Admins monitor system performance, ensuring uptime and resolving technical issues.



d.

## 2.) Non-Functional Requirements:

### RELIABILITY, MAINTAINABILITY, AND AVAILABILITY NON-FUNCTIONAL REQUIREMENTS



Non-functional requirements focus on the quality and constraints of the system rather than its features.

#### 1. Performance:

- The system should be capable of handling up to 1000 concurrent users without noticeable performance degradation.
- Pages (restaurant search, order placement) should load within 3 seconds.

#### 2. Security:

- All user data (login credentials, personal information, payment details) must be encrypted using SSL/TLS.

- b. Adhere to GDPR standards for user data protection and privacy.
- c. Role-based access control: Customers, vendors, delivery personnel, and admins should have different access levels.

### **3. Scalability:**

- a. The system should scale horizontally to accommodate increased users and restaurants (by adding more servers or resources).
- b. Support for multiple cities and regions as the platform grows.

### **4. Availability:**

- a. The system should be available 24/7 with 99.9% uptime.
- b. Automatic failover systems should be in place to prevent downtime during high traffic or system failures.

### **5. Maintainability:**

- a. The codebase should be modular and well-documented to allow easy maintenance and updates.
- b. System logs and error reporting tools should be integrated for monitoring and troubleshooting.

### **6. Usability:**

- a. The UI should be intuitive, easy to navigate for all user types (customers, restaurant owners, admins).
- b. The interface should be responsive, optimized for both desktop and mobile (adaptive design).
- c. Multilingual support to accommodate users from different regions.

### **7. Reliability:**

- a. The system should recover from failures within 5 minutes in case of crashes or downtime.
- b. Data should be backed up daily to prevent loss in case of server issues.

## **3.) Hardware Requirements:**



### **Client-Side (User).**

#### **1. Device Types:**

- a. Smartphones (Android/iOS) for customers and delivery personnel.
- b. Desktop/Laptop for admin and restaurant portals (vendors and admins).

## 2. Hardware Specifications (Minimum):

- a. **RAM:** 2 GB (mobile), 4 GB (desktop)
- b. **Storage:** 500 MB (for mobile apps)
- c. **Screen Resolution:** 720p minimum for responsive design (mobile and web)



## Server-Side

### 1. Web Server:

- a. Quad-core processor or better.
- b. 8 GB RAM or higher.
- c. 500 GB SSD storage or higher.
- d. Minimum 1 Gbps network connection.

### 2. Database Server:

- a. MySQL/PostgreSQL for relational data.
- b. MongoDB for handling semi-structured data (optional).

### 3. Cloud Infrastructure:

- a. Use scalable cloud services like AWS, Azure, or Google Cloud for scalability and performance.

## 4.) User Requirements

### 1. Customers:

- a. Must have easy access to browse menus, place orders, and track deliveries.
- b. Ability to save favorite restaurants and repeat previous orders.
- c. The option to receive notifications via email, SMS, or in-app alerts.

### 2. Restaurant Owners:

- a. Need a comprehensive dashboard to update menus, manage orders, and analyze sales data.
- b. Receive real-time notifications for new orders.
- c. View customer feedback and ratings.

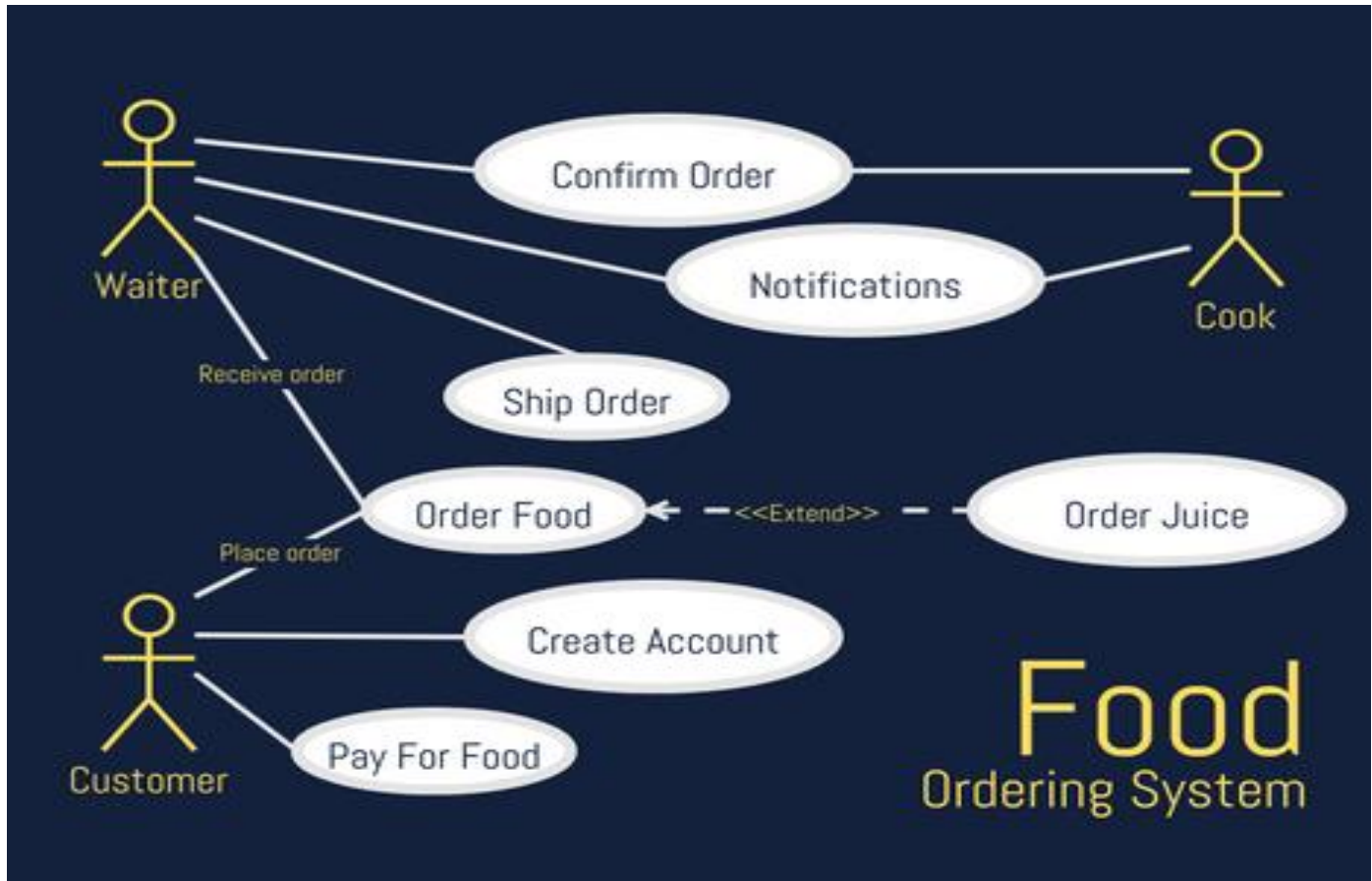
### 3. Delivery Personnel:

- a. Must have an interface to view and accept delivery jobs.
- b. Ability to update delivery status and navigate using real-time maps.

#### 4. Admin Users:

- a. Full control over managing restaurants, users, and delivery partners.
- b. Access to platform performance metrics and the ability to resolve disputes.

Using UML Diagram Use Case Diagram representing Users or Roles associated in the Ordering System:



## Business Domain Requirements

**Business Domain Requirements** define the context and environment in which FoodEaze operates. These requirements are tied to the overall goals of the business and help shape the software's features to meet real-world needs.

### 1. Business Goals & Objectives

FoodEaze's primary business objective is to provide an online platform for users to order food from local restaurants and have it delivered or available for pickup. Key business goals include:

- **Market Expansion:**



- Expand the FoodEaze platform to operate in multiple cities.
- Create partnerships with local restaurants to offer a wide variety of food options.
- **Customer Satisfaction:**
  - Ensure fast and reliable delivery to increase customer loyalty.
  - Provide a seamless user experience across mobile and web platforms.
- **Revenue Generation:**
  - Generate revenue through a commission-based model for each order.
  - Explore alternative revenue streams such as delivery charges, advertising, and premium services for restaurants.

## 2. Key Stakeholders

These are the individuals and groups that are directly impacted by or involved in the success of the FoodEaze system.

- **Customers:**

End-users of the FoodEaze platform place orders for food. Their primary requirements are convenience, timely deliveries, and accurate order tracking.

- **Restaurants/Vendors:**

Restaurant owners and managers who list their menus on the platform. They expect to manage orders efficiently and gain insights through sales analytics.

- **Delivery Personnel:**

Delivery workers or partnered delivery services that facilitate the physical delivery of food. They need real-time updates on delivery assignments and navigation tools to optimize routes.

- **Admin/Operations Team:**

System administrators responsible for maintaining the platform, resolving customer/vendor issues, and ensuring compliance with local regulations and privacy laws.

## 3. Business Rules

Business rules govern how the system operates within the business domain. These are specific conditions that must be followed to align with FoodEaze's business objectives.

- **Commission-based Revenue Model:**

FoodEaze earns a percentage of each completed order paid by the restaurants.

- **Promotions and Discounts:**



Restaurants can offer promotions (discount codes, loyalty programs), and FoodEaze can provide platform-wide deals (e.g., first-time user discount).

- **Cancellation Policy:**

Customers can cancel an order before it's prepared. Refunds may be partial or full, depending on the restaurant's policy and FoodEaze's terms.

- **Vendor Subscription Plans (Optional):**

Restaurants may subscribe to premium packages for enhanced visibility and features like priority listing or targeted advertising on the platform.

## 4. Compliance and Legal Requirements

- **GDPR Compliance:**

FoodEaze must comply with data privacy laws such as the **General Data Protection Regulation (GDPR)** for handling user data within the European Union.

- **PCI-DSS Compliance:**

The platform must follow **Payment Card Industry Data Security Standard (PCI-DSS)** requirements to ensure secure payment processing and storage of credit card information.

- **Local Taxation and Billing Rules:**

The system should support regional taxation systems, ensuring that the appropriate taxes (e.g., VAT, GST) are applied to orders depending on the location.

## 5. Market Analysis & Competitor Overview

- **Competitor Platforms:**

FoodEaze operates in a competitive market with other food delivery services like Uber Eats, DoorDash, and Grubhub. To stand out, FoodEaze focuses on offering:

- Better delivery times.
- Exclusive restaurant partnerships.
- Personalized recommendations based on user preferences.

## 6. Revenue Streams

- **Commission on Orders:**

FoodEaze charges a commission on each order, typically ranging from 10-20% of the total value, depending on the agreement with the restaurant.

- **Delivery Fees:**

Customers pay a delivery fee, which may vary depending on the distance, time, or special offers.

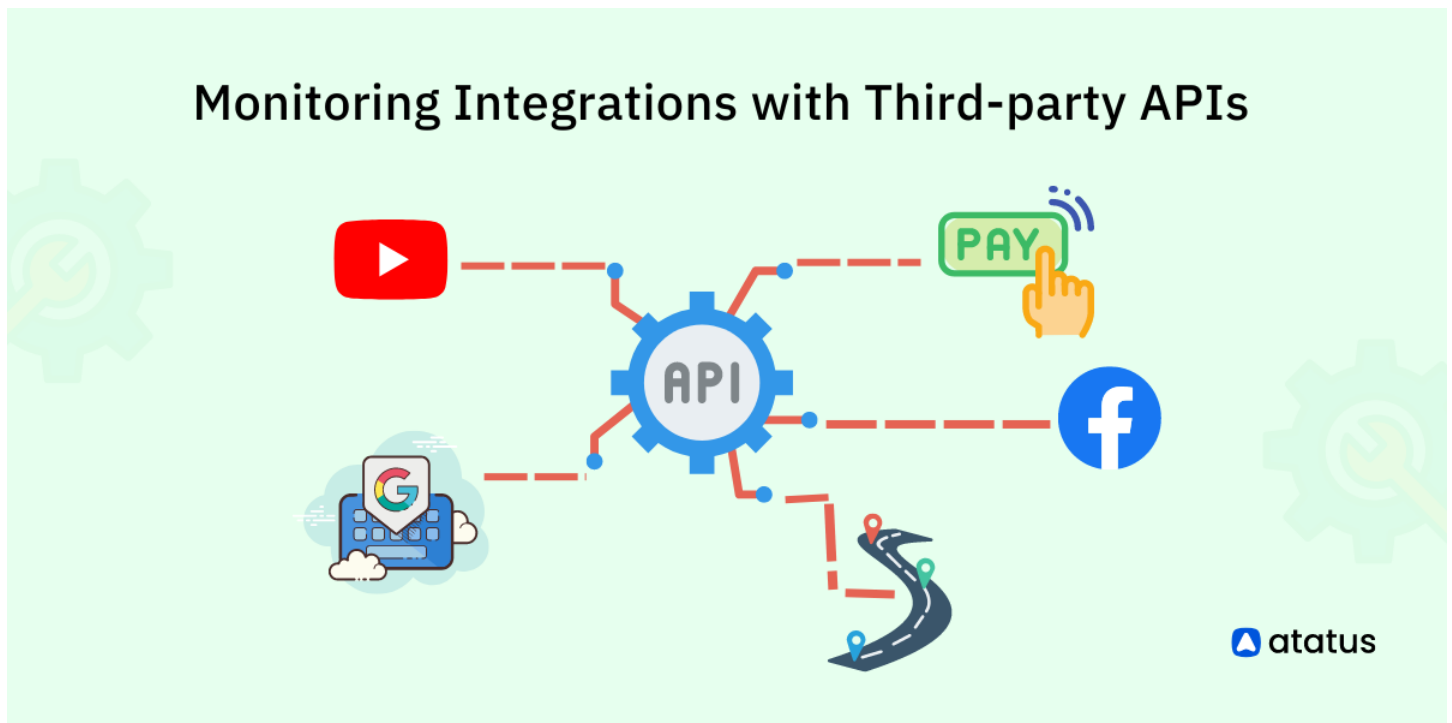
- **Advertising & Featured Listings:**

Restaurants can pay to have their listings promoted on the platform (for increased visibility).

- **Subscription Plans:**

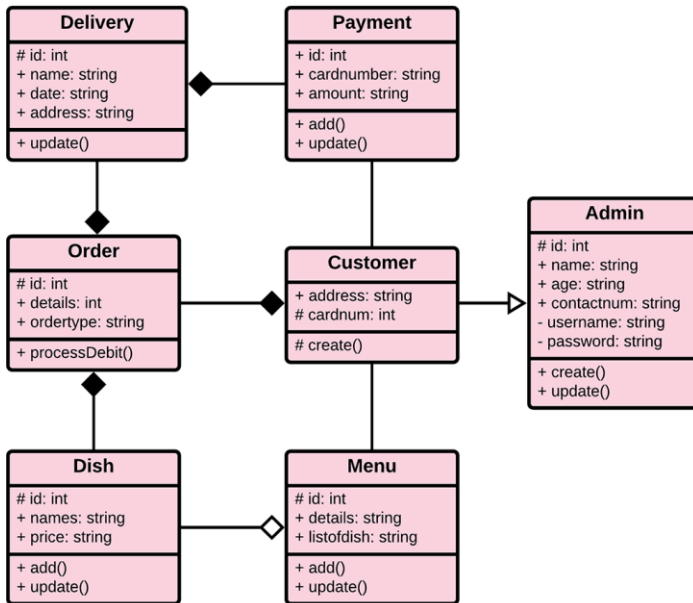
Offering premium features to restaurants, such as enhanced analytics, priority listing, or dedicated customer support.

Some other Aspects to be kept in mind during SRS:



## Other UML DIAGRAMS:

### 1.) Class diagrams:



### 2.) Activity Diagrams:

