

Proposal for Online Food Delivery System Project

1. Introduction

1.1 Project Overview

The **Online Food Delivery System** (OFDS) is designed to provide an online platform for ordering food from various restaurants and having it delivered to the customer's doorstep. The system serves both customers and restaurants, ensuring seamless communication and transaction between them. The system will allow users to browse menus, place orders, make payments, and track deliveries in real-time. It will also provide restaurants with an easy-to-use interface for receiving orders, updating menu items, and tracking customer feedback.

The project aims to enhance the convenience of food ordering by leveraging modern technology, such as mobile applications and web portals. Additionally, the system will provide a seamless, user-friendly experience for customers and an efficient platform for restaurants to manage their orders.

1.2 Problem Statement

The current food delivery services often suffer from inefficiencies like poor user interfaces, slow delivery times, limited payment options, and lack of real-time tracking. These issues create a negative experience for customers and difficulties for restaurants in managing orders. Furthermore, existing systems lack integration, and users often find it challenging to navigate through multiple platforms to place an order or track a delivery.

The **Online Food Delivery System** project aims to address these problems by:

- Offering an easy-to-use platform for both customers and restaurants.
- Providing real-time order tracking for customers.
- Integrating multiple payment methods (credit cards, digital wallets, etc.) for convenience.
- Allowing restaurants to manage their menus, inventory, and customer orders effectively.

1.3 Objectives

The main objectives of the **Online Food Delivery System** are:

- To develop a user-friendly platform (web and mobile) for easy food ordering.
- To integrate secure and diverse payment gateways for smooth transactions.
- To enable real-time tracking of orders for customers.
- To provide restaurant owners with an efficient way to manage orders, customer feedback, and menu changes.

- To optimize delivery routes for faster and more accurate deliveries.
-

2. Domain Analysis

2.1 Customer

The **customer** is the primary user of the food delivery system. Their role is to browse through available restaurants, choose food items, place an order, make a payment, and track the delivery. The customer expects a seamless and efficient service with a simple interface, quick order placement, and reliable delivery service.

Key features for customers:

- Easy registration and login process.
- Intuitive restaurant and menu browsing.
- Secure and multiple payment options.
- Real-time order tracking.
- Order history and repeat order functionality.

2.2 Stakeholders

The key stakeholders in this project include:

1. **Customers:** Individuals who place orders through the online platform.
 2. **Restaurant Owners/Managers:** They provide the food menu, receive orders, manage inventory, and handle delivery or preparation.
 3. **Delivery Personnel:** Responsible for delivering the food to customers.
 4. **System Admin:** Maintains the platform, ensuring its smooth functioning and addressing any issues.
 5. **Payment Gateways:** External services that process customer payments.
 6. **Third-Party Service Providers:** For integration with mapping and routing services (e.g., Google Maps API for delivery tracking).
-

3. Prototypes

The **Online Food Delivery System** will have the following primary prototypes:

1. **Customer Interface (Mobile and Web)**
 - **Home Page:** List of restaurants with filtering options (e.g., type of food, price range, customer ratings).
 - **Restaurant Profile:** Menu display, ratings, delivery times.
 - **Order Placement:** Cart management, payment gateway integration.

- **Order Tracking:** Live status updates (order confirmed, being prepared, out for delivery).
 - 2. **Restaurant Interface (Admin Panel)**
 - **Dashboard:** Overview of active orders, delivery status, and customer feedback.
 - **Menu Management:** Add, update, or remove items from the menu.
 - **Order Management:** Accept, process, and track orders.
 - **Analytics:** View trends, customer ratings, and popular items.
 - 3. **Delivery Personnel Interface**
 - **Order Assignment:** Receive delivery instructions with customer details.
 - **Navigation:** GPS integration for route optimization.
 - **Delivery Status Update:** Mark order as “Picked Up,” “Delivered,” or “Failed.”
-

4. Gantt Chart

Below is a simple Gantt chart to help illustrate the timeline of the project:

Task	Start Date	End Date	Duration
Requirement Analysis	2025-01-16	2025-01-23	7 days
System Design	2025-01-24	2025-01-30	6 days
Database Design and Setup	2025-01-31	2025-02-06	7 days
Front-End Development (Customer)	2025-02-07	2025-02-21	14 days
Back-End Development (Restaurant/Admin)	2025-02-22	2025-03-07	14 days
Payment Gateway Integration	2025-03-08	2025-03-12	4 days
API Integration (Maps, etc.)	2025-03-13	2025-03-16	3 days
Testing & Debugging	2025-03-17	2025-03-24	7 days
User Acceptance Testing	2025-03-25	2025-03-28	3 days
Launch & Deployment	2025-03-29	2025-03-30	2 days

1. Requirements Analysis

1.1 Requirements

The requirements for the **Online Food Delivery System** can be divided into two categories:

- **Functional Requirements:** These define what the system will do (e.g., order placement, user authentication, etc.).
- **Non-Functional Requirements:** These describe how the system will perform, focusing on aspects like performance, security, and usability.

Functional Requirements:

1. **User Authentication:**
 - Customers and restaurant owners must be able to sign up, log in, and manage their profiles.
2. **Restaurant Management:**
 - Restaurant owners must be able to register their restaurants, update menus, and manage orders.
3. **Browse and Search:**
 - Customers should be able to search for restaurants by location, cuisine type, or restaurant name.
4. **Menu Display:**
 - Customers can browse the menu, view food items, and see prices, descriptions, and available options (e.g., vegetarian, vegan, etc.).
5. **Order Placement:**
 - Customers can add items to their cart, view the cart, modify quantities, and place an order.
6. **Payment Integration:**
 - Customers should be able to make payments via multiple payment methods (e.g., credit card, PayPal, Stripe).
7. **Order Tracking:**
 - Customers can track the status of their order in real time (e.g., preparing, out for delivery, delivered).
8. **Delivery Management:**
 - Delivery drivers should be able to receive order details, confirm pick-up, and deliver food to the customer's address.
9. **Review & Rating:**
 - After receiving their orders, customers can leave reviews and ratings for restaurants and food items.
10. **Admin Panel:**
 - Admins should be able to manage users (customers, restaurant owners, delivery drivers), monitor activity, and generate reports.

Non-Functional Requirements:

1. **Usability:**
 - The system should be easy to navigate, with a simple and responsive user interface.
 2. **Performance:**
 - The system should load quickly (within 3 seconds) and handle multiple concurrent users without crashes.
 3. **Security:**
 - Secure user authentication (e.g., using JWT or OAuth).
 - Sensitive information (e.g., payment details) should be encrypted.
 4. **Scalability:**
 - The system should be able to handle an increasing number of restaurants, customers, and orders as it grows.
 5. **Availability:**
 - The system should have high availability (e.g., 99% uptime).
-

2. List of Actors

Actors are entities that interact with the system. Here's a list of the primary actors for the **Online Food Delivery System**:

1. **Customer:**
 - A person who places an order for food from a restaurant. Customers can browse the menu, place orders, make payments, track orders, and leave feedback.
 2. **Restaurant Owner:**
 - A business entity that owns a restaurant listed on the platform. They can manage the restaurant's profile, update the menu, accept/reject orders, and track orders.
 3. **Delivery Driver:**
 - A person responsible for delivering food orders from the restaurant to the customer's address. Drivers receive notifications about new orders, pick up the order, and track delivery status.
 4. **Admin:**
 - A system administrator who manages and oversees the overall operation of the platform. Admins can manage users (customers, restaurant owners, drivers), monitor system activity, and generate reports.
-

3. List of Use Cases

Here are the primary use cases for the **Online Food Delivery System**:

1. Customer Use Cases:

- **Sign Up/Log In:** Customers can create an account or log in to access their profile.
- **Browse Restaurants:** Customers can search for restaurants by name, cuisine type, or location.
- **Browse Menu:** Customers can view a restaurant's menu, including details like price, description, and available options.
- **Place Order:** Customers can add food items to their cart, review the order, and place it.
- **Make Payment:** Customers can pay for their order using integrated payment methods.
- **Track Order:** Customers can track the status of their order in real time.
- **Leave Review:** After receiving the food, customers can rate and review the restaurant.

2. Restaurant Owner Use Cases:

- **Sign Up/Log In:** Restaurant owners can create an account or log in to manage their restaurant profile.
- **Manage Menu:** Restaurant owners can add, remove, or update menu items and prices.
- **Manage Orders:** Restaurant owners can accept or reject orders, view the order status, and manage order fulfillment.

3. Delivery Driver Use Cases:

- **Sign Up/Log In:** Delivery drivers can create an account or log in to view orders and delivery details.
- **Receive Order:** Delivery drivers receive notifications about new orders.
- **Pick Up Order:** Delivery drivers pick up the order from the restaurant.
- **Deliver Order:** Delivery drivers deliver the food to the customer's address and update the order status.

4. Admin Use Cases:

- **Manage Users:** Admins can manage users, including customers, restaurant owners, and drivers.
- **Monitor Orders:** Admins can monitor all orders and manage issues like cancellations or refunds.
- **Generate Reports:** Admins can generate reports on orders, payments, and overall system performance.

For **Assignment 2**, which focuses on **Requirements Analysis** for the **Online Food Delivery System**, you will need to break down the functional and non-functional requirements, identify key actors, and describe use cases that capture the system's behavior. Here's a structured guide you can follow:

1. Requirements Analysis

1.1 Requirements

The requirements for the **Online Food Delivery System** can be divided into two categories:

- **Functional Requirements:** These define what the system will do (e.g., order placement, user authentication, etc.).
- **Non-Functional Requirements:** These describe how the system will perform, focusing on aspects like performance, security, and usability.

Functional Requirements:

1. **User Authentication:**
 - Customers and restaurant owners must be able to sign up, log in, and manage their profiles.
2. **Restaurant Management:**
 - Restaurant owners must be able to register their restaurants, update menus, and manage orders.
3. **Browse and Search:**
 - Customers should be able to search for restaurants by location, cuisine type, or restaurant name.
4. **Menu Display:**
 - Customers can browse the menu, view food items, and see prices, descriptions, and available options (e.g., vegetarian, vegan, etc.).
5. **Order Placement:**
 - Customers can add items to their cart, view the cart, modify quantities, and place an order.
6. **Payment Integration:**
 - Customers should be able to make payments via multiple payment methods (e.g., credit card, PayPal, Stripe).
7. **Order Tracking:**
 - Customers can track the status of their order in real time (e.g., preparing, out for delivery, delivered).
8. **Delivery Management:**
 - Delivery drivers should be able to receive order details, confirm pick-up, and deliver food to the customer's address.
9. **Review & Rating:**
 - After receiving their orders, customers can leave reviews and ratings for restaurants and food items.
10. **Admin Panel:**
 - Admins should be able to manage users (customers, restaurant owners, delivery drivers), monitor activity, and generate reports.

Non-Functional Requirements:

1. **Usability:**

- The system should be easy to navigate, with a simple and responsive user interface.
 - 2. **Performance:**
 - The system should load quickly (within 3 seconds) and handle multiple concurrent users without crashes.
 - 3. **Security:**
 - Secure user authentication (e.g., using JWT or OAuth).
 - Sensitive information (e.g., payment details) should be encrypted.
 - 4. **Scalability:**
 - The system should be able to handle an increasing number of restaurants, customers, and orders as it grows.
 - 5. **Availability:**
 - The system should have high availability (e.g., 99% uptime).
-

2. List of Actors

Actors are entities that interact with the system. Here's a list of the primary actors for the **Online Food Delivery System**:

1. **Customer:**
 - A person who places an order for food from a restaurant. Customers can browse the menu, place orders, make payments, track orders, and leave feedback.
 2. **Restaurant Owner:**
 - A business entity that owns a restaurant listed on the platform. They can manage the restaurant's profile, update the menu, accept/reject orders, and track orders.
 3. **Delivery Driver:**
 - A person responsible for delivering food orders from the restaurant to the customer's address. Drivers receive notifications about new orders, pick up the order, and track delivery status.
 4. **Admin:**
 - A system administrator who manages and oversees the overall operation of the platform. Admins can manage users (customers, restaurant owners, drivers), monitor system activity, and generate reports.
-

3. List of Use Cases

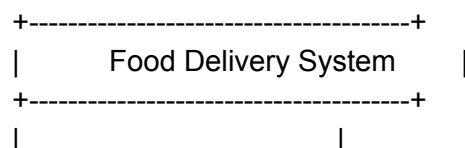
Here are the primary use cases for the **Online Food Delivery System**:

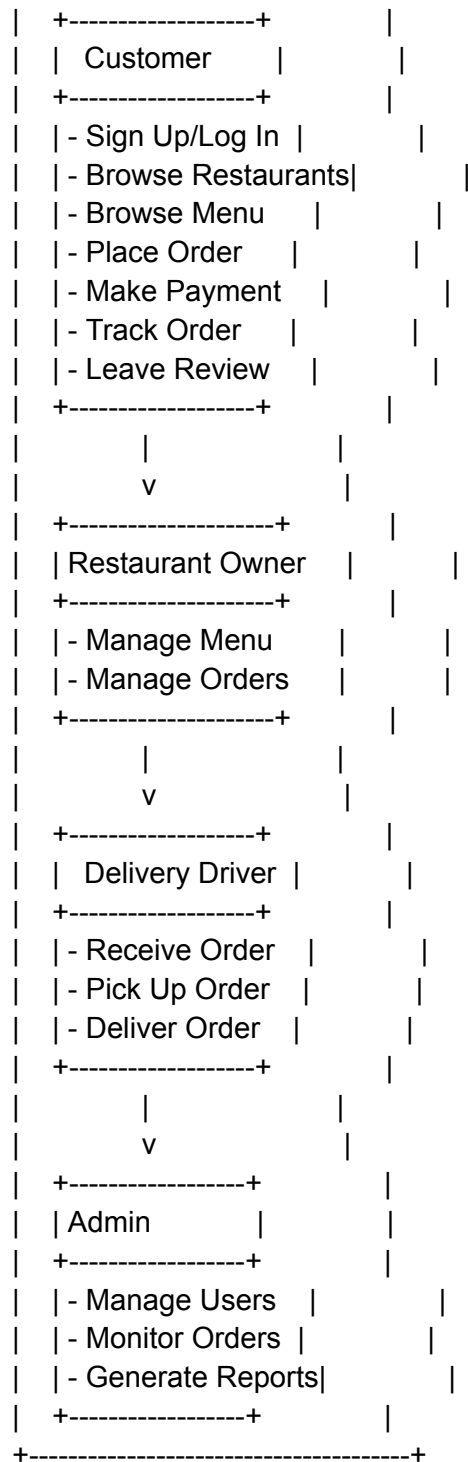
1. **Customer Use Cases:**
 - **Sign Up/Log In:** Customers can create an account or log in to access their profile.

- **Browse Restaurants:** Customers can search for restaurants by name, cuisine type, or location.
 - **Browse Menu:** Customers can view a restaurant's menu, including details like price, description, and available options.
 - **Place Order:** Customers can add food items to their cart, review the order, and place it.
 - **Make Payment:** Customers can pay for their order using integrated payment methods.
 - **Track Order:** Customers can track the status of their order in real time.
 - **Leave Review:** After receiving the food, customers can rate and review the restaurant.
2. **Restaurant Owner Use Cases:**
- **Sign Up/Log In:** Restaurant owners can create an account or log in to manage their restaurant profile.
 - **Manage Menu:** Restaurant owners can add, remove, or update menu items and prices.
 - **Manage Orders:** Restaurant owners can accept or reject orders, view the order status, and manage order fulfillment.
3. **Delivery Driver Use Cases:**
- **Sign Up/Log In:** Delivery drivers can create an account or log in to view orders and delivery details.
 - **Receive Order:** Delivery drivers receive notifications about new orders.
 - **Pick Up Order:** Delivery drivers pick up the order from the restaurant.
 - **Deliver Order:** Delivery drivers deliver the food to the customer's address and update the order status.
4. **Admin Use Cases:**
- **Manage Users:** Admins can manage users, including customers, restaurant owners, and drivers.
 - **Monitor Orders:** Admins can monitor all orders and manage issues like cancellations or refunds.
 - **Generate Reports:** Admins can generate reports on orders, payments, and overall system performance.

4. System Use Case Diagram

A **Use Case Diagram** visually represents the interactions between actors and the system. Here's an outline of how the system use case diagram might look for your **Online Food Delivery System**:





In this diagram, you can see the **actors** (Customer, Restaurant Owner, Delivery Driver, Admin) and the use cases that each actor interacts with in the system.

5. Extended Use Cases

Extended use cases provide more detailed scenarios for each use case. Here's an example for a few use cases:

Use Case: Place Order (Customer)

- **Basic Flow:**
 - The customer logs in or creates a new account.
 - The customer browses available restaurants.
 - The customer selects a restaurant and browses the menu.
 - The customer adds items to their cart.
 - The customer proceeds to checkout.
 - The customer enters delivery details (address, contact information).
 - The customer selects a payment method and completes the payment.
 - The customer receives an order confirmation and estimated delivery time.
- **Alternative Flows:**
 - **A1: Payment Failure:** If the payment fails, the system displays an error message and prompts the customer to retry or choose a different payment method.
 - **A2: Menu Item Unavailable:** If a menu item is unavailable, the system notifies the customer and suggests alternatives.

Use Case: Manage Menu (Restaurant Owner)

- **Basic Flow:**
 - The restaurant owner logs in.
 - The owner navigates to the menu management section.
 - The owner can add new menu items, update existing ones, or remove unavailable items.
 - The system saves the changes and updates the menu visible to customers.
- **Alternative Flows:**
 - **A1: Invalid Menu Data:** If invalid data is entered (e.g., price is missing), the system will prompt the owner to correct the error.