1. Introduction

1.1 Purpose

Our project "FoodieGo" aims to create a user-friendly and highly interactive web platform that connects users with restaurants or caterers, so that customers can order food online and have it delivered right to their door. We are committed to offering consumers an easy and reliable food ordering experience while providing them with a reasonably wide variety of food choices. With real time tracking, smart logistics and recommendations, we optimize the overall customer experience and support local restaurants to reach more customers and expand their services through delivery. The most important impact of this delivery service platform is making everyday food far more accessible and hassle-free.

1.2 Intended Audience

Consumers: The primary audience for this platform is urban consumers who are looking for ease and speed along with choice in eating out.

Local restaurants: The platform also targets easy-delivery restaurants & cafes planning to find new customers, grow their online activity, and raise sales by means of delivery.

Delivery Personnel: The platform will also cater to gig economy workers, including delivery drivers, providing them with flexible earning opportunities.

1.3 Intended Use

The food delivery system is designed to provide users with a simple, user-friendly interface that allows them to effortlessly browse, order, and receive food from local restaurants. The primary users of the system will include customers looking to enjoy restaurant meals at home, restaurant owners managing orders, and delivery personnel fulfilling those orders. The system aims to streamline the food ordering process and improve the user experience with features that are intuitive and accessible.

Key Functions for End Users

User Account Management: Users will have the ability to create a new account, providing essential information such as name, email, phone number, and address. They will be able to securely log in and log out of their accounts.

Restaurant Listings: The system will display a list of nearby restaurants based on the user's location. Users will also be able to search for any specific restaurant of their own choice.

Restaurant Menus: Upon selecting a restaurant, users will be able to view the restaurant's menu, complete with descriptions, and prices of the available food items.

Order Customization: Users can add or remove food items to and from their cart. They will be able to view the cart at any time to modify quantities or remove items before placing the order.

Placing Orders: Once the cart is finalized, users can proceed to place an order by confirming the restaurant, delivery address, and payment method. They can also apply promotional codes or discounts at checkout, if available.

Order Cancellation: Users will have the option to cancel their order within a specific time frame, depending on the restaurant's policies.

Payment Options: The system will support multiple secure payment methods, including online payment or cash on delivery, ensuring convenience and flexibility for users.

1.4 Product Scope

The goal of this food delivery system is to provide a convenient platform that connects customers with local restaurants, enabling easy ordering and delivery of food. The product aims to simplify the ordering process, expand restaurant reach, and create a scalable solution that enhances user satisfaction while driving revenue growth.

Benefits

Customer Convenience: Users can easily order food from their favorite restaurants and have it delivered to their doorsteps.

Increased Restaurant Reach: Restaurants gain access to a wider customer base, increasing their sales potential.

Revenue Growth: The platform generates revenue through restaurant commissions, delivery fees, and premium services.

Objectives

User-Friendly Interface: Offer a simple, intuitive interface for browsing restaurants, placing orders, and tracking deliveries.

Operational Efficiency: Automate key processes to reduce manual efforts for restaurants and delivery personnel.

Scalability: Ensure the system can handle growth in users and transactions as the business expands.

Alignment with Business Goals

Revenue Generation: The platform supports business growth through increased order volumes and partnerships with restaurants.

Customer Retention: Focus on delivering a seamless experience to encourage repeat usage and build brand loyalty.

Market Expansion: Target a broad customer base, helping the business penetrate new markets and grow its footprint.

1.5 Risk Definitions

This section outlines the potential risks associated with the development, deployment, and operation of the food delivery system. Identifying these risks helps mitigate potential issues that could impact the success of the project.

Technical Risks

System Scalability: As the user base grows, there is a risk that the system may face performance issues or downtime if it is not designed to scale effectively. This could result in slow response times, failed transactions, or service outages.

Integration Failures: The system depends on third-party services such as payment gateways and location-based APIs. Any failure in these integrations could prevent users from placing orders, making payments, or tracking deliveries.

Data Security Breaches: Since the platform handles sensitive user information (e.g., payment details, personal addresses), there is a risk of data breaches if proper security measures (e.g., encryption, secure authentication) are not in place.

Operational Risks

Order and Delivery Delays: Inconsistent or delayed deliveries due to restaurant inefficiencies or delivery personnel issues could lead to user dissatisfaction and reduced platform trust.

Restaurant Partnership Management: Miscommunications or discrepancies between the platform and restaurant partners regarding menu updates, pricing, or availability may result in incorrect orders or poor user experience.

Financial Risks

Revenue Model Instability: The platform's revenue model, which relies on commissions and delivery fees, may be negatively impacted by rising competition, restaurants pulling out, or unanticipated operational costs.

High Customer Acquisition Costs: A high cost to acquire new users due to aggressive marketing and promotions may strain the business financially, especially if customer retention rates are lower than expected.

Legal and Compliance Risks

Regulatory Compliance: The platform must adhere to local laws and regulations regarding food safety, delivery logistics, and data protection. Failure to comply may result in legal action, fines, or the suspension of services.

User Adoption Risks

Low User Engagement: If the platform fails to meet user expectations in terms of ease of use, variety of restaurant choices, or timely delivery, user adoption may be lower than anticipated, impacting overall platform growth.

Market Competition: Strong competition from established food delivery platforms could limit the platform's ability to gain a significant market share.

External Risks

Supply Chain Disruptions: Any disruptions in the supply chain, such as restaurant closures or delivery personnel shortages, could impact the platform’s ability to fulfill orders.

2. Overall Description

2.1 User Classes and Characteristics

The platform targets two distinct user groups: customers and administrators. These groups interact with the system in different ways, each requiring specific features and functionality.

Customers: Customers are individuals who use the platform to order food from a variety of restaurants. These users typically belong to diverse age groups, ranging from young adults to older individuals, and have varying levels of digital literacy. Many users are familiar with online shopping and mobile applications and expect a fast and convenient experience. Key characteristics include:

Tech-Savvy: Many customers are comfortable with mobile apps and online platforms, making navigation and order placement a familiar process.

Occasional Users: Some users may only use the platform sporadically, requiring a straightforward interface that allows them to quickly browse, order, and pay without a steep learning curve.

Demand for Customization: Customers often prefer to customize their orders, expecting flexible options for food selection, special instructions, and quantity adjustments.

Security-Conscious: Given that payment involves financial transactions, users are cautious about sharing sensitive information and expect the platform to implement robust security measures.

Administrators: Administrators are the backbone of the system, responsible for overseeing the day-to-day operations of the platform. They need access to advanced management features to handle various tasks efficiently. Characteristics of admins include:

Highly Skilled: Admins are proficient with the system's backend and understand how to manage food inventories, user data, and payments through a more complex administrative interface.

Task-Oriented: Their primary focus is on updating menus, processing customer requests, monitoring orders, and ensuring that deliveries are timely and accurate.

Real-Time Responsiveness: Admins must continuously monitor incoming orders and update the availability of menu items in real-time to ensure customers have accurate options when browsing.

Problem Solvers: In addition to managing the platform, admins handle customer queries, order issues, and payment disputes, requiring good decision-making and problem-solving abilities.

2.2 User Needs

Understanding the needs of both user classes is essential for creating a functional and user-friendly platform. The needs of customers and administrators are distinct but interconnected, and the platform must cater to both effectively.

Customer Needs:

Simple and Quick Ordering: Customers need a streamlined process for browsing food items, adding them to the cart, customizing their orders, and checking out. The system should support easy food item searches, real-time availability, and clear descriptions of items, allowing users to make informed decisions quickly.

Efficient Payment Process: Users need a secure, fast, and reliable payment system with various payment options like debit/credit cards, mobile payments, and third-party payment gateways. Customers also expect immediate confirmation of successful transactions and, in some cases, the ability to cancel or modify orders post-purchase.

Order Tracking and Updates: After placing an order, customers want real-time tracking of their food's preparation and delivery status. Notifications via email or SMS about order progress—such as confirmation, preparation, dispatch, and expected delivery time—are critical to enhancing user satisfaction.

Account Management and Personalization: Customers need access to account management features such as updating personal information, managing delivery addresses, viewing order history, and saving preferences for faster future orders. A smooth registration and login process, including social media or Google authentication, is also vital for an easy onboarding experience.

Administrator Needs:

Real-Time Menu and Order Management: Administrators require tools to update food items, change availability, and set pricing based on stock levels or daily specials. They must also be able to manage customer orders efficiently, processing requests and communicating with kitchen staff and delivery personnel to ensure timely service.

User and Transaction Oversight: Admins need the ability to view and manage customer profiles, including viewing order histories and addressing customer concerns. For payments, they need to monitor financial transactions, issue refunds, and resolve payment disputes, all within a secure and reliable system.

Order Cancellation and Refunds: As customer orders may occasionally require modifications or cancellations, administrators need tools that allow them to swiftly cancel orders or offer refunds when necessary. This functionality is vital to maintain customer satisfaction and smooth operations.

System Monitoring and Performance: Admins must ensure that the platform operates smoothly, particularly during peak order times. Monitoring server performance, ensuring quick load times, and handling any technical issues are critical for delivering a high-quality user experience.

2.3 Operating Environment

The platform will operate as a web-based application accessible from a variety of devices, including desktops, laptops, tablets, and mobile phones. The design of the platform ensures compatibility with modern web browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

Web Technology Stack: The platform is built using PHP as the primary server-side programming language and MySQL for the database. The front end utilizes HTML, CSS, and JavaScript to ensure a responsive and user-friendly interface. This combination allows for scalability and efficient data management, ensuring the system can handle a high volume of traffic and orders.

Database System: The platform's database system is based on MySQL, a widely used relational database management system (RDBMS). It stores all customer, order, and restaurant data securely and supports efficient querying and data retrieval. The use of relational data structures ensures that customer orders, menu items, and payment transactions are interconnected, improving the overall efficiency of operations.

Device and Platform Compatibility: The platform is designed to be responsive, meaning it adapts to different screen sizes, making it equally functional on mobile devices, tablets, and desktops. This is essential as a large portion of users may prefer to place orders via smartphones.

Internet Connectivity: A stable internet connection is necessary for users and administrators to interact with the platform. Given that the system relies on real-time updates for food availability and order tracking, it is important that the platform handles network latency and disconnections gracefully. Admins need continuous access to manage orders, especially during peak hours.

Security Features: The platform integrates SSL encryption for secure data transmission, especially during payment transactions. This ensures that sensitive customer information, such as login credentials and payment details, remains protected. It also complies with industry-standard security protocols, providing an additional layer of trust for users engaging in financial transactions.

Server Environment: The platform will be hosted on a Linux-based web server, supporting PHP and MySQL. This server environment ensures high performance, reliability, and scalability as the system grows in user base and transaction volume. The system should be capable of handling multiple simultaneous user requests without a significant decline in performance.

Backup and Disaster Recovery: Regular backups of the database and transaction data are necessary to ensure that no information is lost in case of a system failure or cyberattack. Admins should have access to backup features that allow them to restore critical data in case of emergencies.

2.4 Constraints

Cross-Platform Compatibility: The system must support Android and iOS smartphones and tablet PCs as well as desktop browser web solutions and mobile browser solutions.

Third-Party Integration: The system must be able to connect with third-party services like payment, mapping, or any other real-time services to make it more user focused.

Scalability: The system must accommodate at least 50 concurrent users and deal with large loads and advanced usage at busy peak hours, weekends and holidays in particular.

Compliance with Local Data Privacy Regulations: The system must implement appropriate measures for protecting a lot of personal information about users, vendors, delivery partners, and orders.

Standards of Payment Compliance: The platform must ensure that any user data and any payments are done within the guidelines of the PCI DSS (Payment Card Industry Data Security Standard).

Real-Time Updates: The system under consideration must endeavor to provide live updates for changes in orders, delivery, and vendor availability – in under 5 seconds.

Vendor Compliance: The platform must comply with the terms and conditions set by participating restaurants, grocery partners, and delivery service operators, including minimum order amounts, delivery time requirements, and applicable commission fees.

Service Availability: The service must be available 24/7 in the main regions, but it must also account for vendor-specific operating hours (e.g some restaurants may not provide service at night).

Inventory Synchronization: The platform must ensure that the inventory and price information is always updated for each partner and restaurant each time there is a change or an update.

Dynamic Commission Rates: The system must incorporate dynamic commission scales for vendors and delivery partners according to the vendor category, for example, high commission rates can be applied to restaurants while low rates for grocery stores.

User-Friendly Interface: The platform must have a user-friendly design such that customers can explore vendors, add products to their carts, and keep track of orders with ease.

24/7 Customer Support: Customer support must be active on all the days including holidays and at any given hour to assist the users on issues related to purchasing products, money transaction and use of the application.

Secure User Authentication: Users should have the option to log in using their email, social media accounts, or phone number, utilizing secure authentication methods such as One-Time Passwords (OTP) to ensure account safety.

Food Safety Compliance: Vendors must comply with local food safety laws and regulations, and the platform will enforce these requirements as a prerequisite for registration.

Dynamic Tax Calculation: The system must perform the calculation of taxes automatically depending on the location of the vendor and the tax statutes in force to ensure compliance with taxation policies.

Quick Response Times: The system must ensure quick responses, aiming for a maximum delay of 1 second for search queries and real-time updates, such as order status and delivery tracking.

Delivery Partner Availability: The platform must guarantee that there are sufficient delivery partners to execute orders regardless of the peak times.

GPS Tracking for Deliveries: The delivery system must offer real-time updates on the status of orders and the location of delivery partners, utilizing GPS technology for accurate tracking.

2.5 Assumptions

- **User Access:**

  Users are expected to have a Wi-Fi or mobile data internet connection with adequate bandwidth to support a good application experience.

  Potential Problem: Users in rural or underserved areas may face connectivity issues which might affect their ability to use the application effectively.

- **Device Compatibility:**

  Users will access the application via minimum hardware and software requirements on smartphone, tablets, and computers.

  Potential Problem: If many users are using old devices that do not meet the new requirements, they may not be able to use the application.

- **Vendor Participation :**

  All participating vendors are assumed to be willing to align their inventory systems with the platform to give constant updates on what products they currently have in stock.

  Potential Problem: If some vendors do not want to participate or are not able to integrate their systems, there will be conflict with product availability resulting in possible user dissatisfaction.

- **User Familiarity:**

  Users are assumed to have basic knowledge of using mobile and web applications for their online orders and payments.

  Potential Problem: If a significant portion of users lack this familiarity, it could hinder their ability to navigate through the application, requiring additional support and educational resources.

- **Payment Method Acceptance:**

  Users will have access to various payment methods, allowing them to complete transactions smoothly.

Potential Problem: Changes in the policies of payment providers or limited options in specific regions of any given payment platform could lead to transaction failures or user frustration.

- **Delivery Partner Availability:**

  The system is expected to have a sufficient number of delivery partners available during busy and non-busy hours to meet customer demand.

  Potential Problem: If the number of delivery requests surpasses the available delivery partners, it may lead to delays and reduce customer satisfaction.

- **Regulatory Compliance:**

  The system requires all vendors to follow local health, safety and food handling regulations.

  Potential Problem: Regulatory changes or not following the rules and regulations may lead to safety risk and affect the platform's reputation.

- **Real-time Updates:**

  The application will deliver real-time updates on order status, vendor availability, and delivery tracking, provided that all integrated systems operate smoothly.

  Potential Problem: Users may not receive accurate updates if there are problems with system integrations or connectivity which might cause confusion and dissatisfaction to customers.

- **Customer Support:**

  Users will have access to customer support for any questions or issues related to their orders, payments, or the app's functionality.

  Potential Problem: If the customer support team is overwhelmed or inadequately trained, response times may increase, resulting in user frustration.

3. Requirements

3.1 Functional Requirements

<table>
<tr><th>Title: Login</th><th>Priority: High</th><th>Estimate: 1 week</th></tr>
<tr><td colspan="3"><b><u>User Story</u></b>
● <b>As an</b> admin,
● <b>I want to</b> log into my account using my Google credentials,
● <b>So that I can</b> access the system.</td></tr>
<tr><td colspan="3"><b><u>Acceptance Criteria</u></b>
● <b>Given</b> that admin has a google account,
● <b>When</b> the admin chooses to log in using Google,
● <b>Then</b> the system should redirect to the Google login page and verify the credentials through the Google API.</td></tr>
</table>

Title: Order Food	**Priority: Medium**	**Estimate: 1 week**
User Story ● **As a** user, ● **I want to** browse and view available food items, ● **So that I can** select and order the food I want.		
Acceptance Criteria ● **Given** that the user is logged in and looking for food to order, ● **When** the user clicks the search button and enters a food name or category, ● **Then** the system should display a list of available food items that match the search criteria.		

Title: Cancel Food	**Priority: Medium**	**Estimate: 1 week**

User Story

- **As a** user,
- **I want to** cancel a food item after ordering,
- **So that I can** remove it from my order if I change my mind.

Acceptance Criteria

- **Given** that the user has placed an order and now wishes to cancel it,
- **When** the user clicks on the cancel button for the specific food item,
- **Then** the system should successfully cancel the order and remove the food item from the user's current orders list, showing a confirmation of the cancellation.

Title: Add food item	**Priority: High**	**Estimate: 1 week**

<u>User Story</u>

- **As an** admin,
- **I want to** add a new food item that is newly available,
- **So that I can** keep the menu up to date with the latest offerings.

<u>Acceptance Criteria</u>

- **Given** that the admin wants to add a new food item to the menu,
- **When** the admin clicks the "Add" button and inputs the necessary details for the new food item,
- **Then** the system should update the food item database, and the new item should be visible on the menu for customers.

Title: Delete food item	**Priority: High**	**Estimate: 1 week**

User Story

- **As an** admin,
- **I want to** delete a food item that is no longer available,
- **So that I can** ensure the menu only displays available options.

Acceptance Criteria

- **Given** that the admin is managing the food menu and wants to remove an unavailable food item,
- **When** the admin clicks the "Delete" button for the specified item,
- **Then** the system should remove the food item from the database, and it should no longer appear on the customer-facing menu.

<table>
<tr><td><b>Title: Check Customer profile</b></td><td><b>Priority: Medium</b></td><td><b>Estimate: 1 week</b></td></tr>
<tr><td colspan="3"><b><u>User Story</u></b>
● <b>As an</b> admin,
● <b>I want to</b> check the status of a customer’s order,
● <b>So that I can</b> monitor whether the customer has received their food or not.</td></tr>
<tr><td colspan="3"><b><u>Acceptance Criteria</u></b>
● <b>Given</b> that the admin is logged into the system and is searching for a specific customer profile,
● <b>When</b> the admin clicks on the customer ID,
● <b>Then</b> the system should display the customer’s profile, including their order details and current order status, to the admin.</td></tr>
</table>

<table>
<tr><td><b>Title: Payment</b></td><td><b>Priority: High</b></td><td><b>Estimate: 1 week</b></td></tr>
<tr><td colspan="3"><b><u>User Story</u></b>
● <b>As a</b> user,
● <b>I want to</b> pay for my order,
● <b>So that I can</b> complete my purchase and receive my food.</td></tr>
<tr><td colspan="3"><b><u>Acceptance Criteria</u></b>
● <b>Given</b> that the user has a valid payment method linked to their account,
● <b>When</b> the user clicks the payment button,
● <b>Then</b> the system should process the payment and confirm the transaction by showing a payment success message.</td></tr>
</table>

Title: Log out	Priority: High	Estimate: 1 week

User Story

- **As a** user,
- **I want to** log out of my account,
- **So that I can** securely end my session and prevent unauthorized access.

Acceptance Criteria

- **Given** that the user is logged into the system,
- **When** the user clicks the logout button,
- **Then** the system should log the user out, end the session, and redirect to the login page with confirmation that the logout was successful.

3.2 Non-Functional Requirements

Performance Requirements:

- The system should handle up to 50 concurrent users without a noticeable decrease in speed.
- Page load times should not exceed 5 seconds under normal conditions.
- The system must process payments within 10 seconds on average.
- Database queries should return results within 5 seconds to maintain acceptable operation.

Safety Requirements:

- The system should include basic data validation checks to prevent incorrect inputs (e.g., empty fields, invalid payment details).
- User inputs (e.g., food orders, payments) should be periodically backed up to minimize data loss.
- In case of a transaction error, the system should provide simple error recovery, such as allowing users to retry without losing previous inputs.

Security Requirements:

- All sensitive data (such as passwords) should be hashed and salted before being stored in the database.
- Basic two-factor authentication (2FA) should be implemented for admin accounts where feasible.
- User sessions should expire after 30 minutes of inactivity to minimize unauthorized access risks.
- Payment data must be handled securely, adhering to basic industry practices for secure payments (encryption in transit).

Quality Requirements:

- The system should maintain at least 95% uptime, ensuring reasonable availability for users.
- User interfaces should be simple and easy to navigate, ensuring a good user experience.
- The platform should be designed for minor future updates, with enough documentation for essential features.
- Basic error messages must be provided to users when something goes wrong (e.g., invalid input or connection issues), guiding them to try again.