**Requirement Gathering and Analysis Phase**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 06-07-2024 |
| Team ID | SWTID1720074953 |
| Project Name | SB Foods-Food Ordering App |
| Maximum Marks |  |

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Gmail  Registration Through Form  Registration through LinkedIn |
| FR-2 | User Authentication | Confirmation via Email  Confirmation via Social Media |
| FR-3 | Menu browsing | View Menu by Categories  Search menu items |
| FR-4 | Place Order | Add items to Cart  Edit Cart Contents  Select Delivery Address  Choose Delivery Time |
| FR-5 | Payment Processing | Payment via Cash-On-Delivery(COD) or payment via Credit/Debit Card Stripe |
| FR-6 | Order Tracking | Track Order Status  Receive Order Updates |
| FR-7 | Account Management | Edit Profile Information  Change Password  View Order History |
| FR-8 | Promotions and Discounts | Apply Promo Code |
| FR-9 | Customer Support | Contact Support Team  Resolve Order Issues |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Usability refers to the ease with which users can interact with and navigate the application to achieve their goals effectively. For a food delivery application, usability involves intuitive user interfaces (UI), clear navigation, minimal steps for placing orders, accessible menus, and responsive design across devices. It also includes features like order tracking, user-friendly checkout processes, and personalized recommendations to enhance user experience. |
| NFR-2 | **Security** | Security measures aim to protect user data, transactions, and application integrity from unauthorized access, breaches, and malicious activities. For the food delivery app, robust security practices include encryption of sensitive data (e.g., user credentials, payment information), secure API endpoints using HTTPS, implementation of authentication and authorization mechanisms (e.g., JWT tokens), regular security audits, and compliance with industry standards (e.g., GDPR, PCI-DSS). |
| NFR-3 | **Reliability** | Reliability ensures the consistent availability and performance of the application under normal and peak loads. For the food delivery app, reliability involves minimizing downtime, ensuring data integrity, and providing accurate order processing and delivery tracking. It includes fault tolerance mechanisms (e.g., redundant servers, database replication), proactive monitoring, automated backups, and quick recovery plans in case of failures to maintain uninterrupted service. |
| NFR-4 | **Performance** | Performance relates to the responsiveness, speed, and efficiency of the application in handling user interactions and processing requests. In the context of a food delivery app, performance considerations include fast loading times for menus and order pages, quick response to user inputs (e.g., adding items to cart, updating order status), optimizing API responses, efficient database queries, and utilization of caching mechanisms to reduce latency. |
| NFR-5 | **Availability** | Availability ensures that the food delivery application is accessible to users whenever they need it, without interruptions or downtime. This involves deploying redundant systems, load balancing to distribute traffic evenly across servers, using content delivery networks (CDNs) for static assets, and implementing disaster recovery plans. High availability is crucial for maintaining user trust and meeting service level agreements (SLAs). |
| NFR-6 | **Scalability** | Scalability refers to the ability of the application to handle increased workload or growth in users, transactions, and data without compromising performance. For a food delivery app, scalability is achieved through horizontal scaling (adding more servers or instances), microservices architecture for modular and independent services, elastic scaling of resources based on demand, and optimizing resource allocation to accommodate peak loads efficiently. The solution can scale by adding more food items, expanding to new geographic locations, integrating more payment gateways, and enhancing the admin panel with more features |