**Software Requirements Specification (SRS) for NFT Lunch Menu System**

Table of Contents

[**1. Introduction** 5](#_Toc141438822)

[1.1 Purpose 5](#_Toc141438823)

[1.2 Scope 5](#_Toc141438824)

[1.3 Intended Audience 5](#_Toc141438825)

[**2. Overall Description** 6](#_Toc141438826)

[2.1 Product Perspective 6](#_Toc141438827)

[2.1.1 Menu Management: 6](#_Toc141438828)

[2.1.2. Order Placement: 6](#_Toc141438829)

[2.1.3. Inventory Management: 6](#_Toc141438830)

[2.1.4. Real-time Updates: 6](#_Toc141438831)

[2.1.5. Customer Loyalty and Feedback: 6](#_Toc141438832)

[2.1.6. Data Analytics: 7](#_Toc141438833)

[**2.2 Product Features** 7](#_Toc141438834)

[2.2.1. User Registration and Authentication: 7](#_Toc141438835)

[2.2.2. Food Item Management: 7](#_Toc141438836)

[2.2.3. Order Placement and Customization: 7](#_Toc141438837)

[2.2.4. Real-time Availability and Updates: 7](#_Toc141438838)

[2.2.5. Order Management and Processing: 8](#_Toc141438839)

[2.2.6. Reporting and Analytics: 8](#_Toc141438840)

[**2.3 User Classes and Characteristics** 8](#_Toc141438841)

[2.3.1. Pantry Staff: 8](#_Toc141438842)

[2.3.2. Company Managers and Overall Staff: 9](#_Toc141438843)

[**2.4 Operating Environment** 9](#_Toc141438844)

[2.4.1 Hardware Requirements: 9](#_Toc141438845)

[2.4.2 Software Requirements: 10](#_Toc141438846)

[2.4.3 Network Environment: 10](#_Toc141438847)

[**2.5 Design and Implementation Constraints** 10](#_Toc141438848)

[2.5.1. Time Constraints: 10](#_Toc141438849)

[2.5.2. Technical Expertise: 11](#_Toc141438850)

[2.5.3 Compatibility: 11](#_Toc141438851)

[2.5.4. Data Security: 11](#_Toc141438852)

[2.5.5. Scalability: 11](#_Toc141438853)

[2.5.6. Network Reliability 11](#_Toc141438854)

[2.5.7. Integration with Existing Systems: 11](#_Toc141438855)

[2.5.8. Maintenance and Support: 11](#_Toc141438856)

[2.5.9. Legal and Regulatory Compliance: 11](#_Toc141438857)

[2.5.10. Performance: 11](#_Toc141438858)

[2.5.11. User Interface and User Experience: 12](#_Toc141438859)

[**2.6 Assumptions and Dependencies** 12](#_Toc141438860)

[2.6.1 Assumptions: 12](#_Toc141438861)

[2.6.2 External Dependencies: 12](#_Toc141438862)

[**3. System Features and Requirements** 13](#_Toc141438863)

[**3.1 Functional Requirements** 13](#_Toc141438864)

[3.1.1. User Registration: 13](#_Toc141438865)

[3.1.2. User Login: 14](#_Toc141438866)

[2.1.3. Food Item Management: 14](#_Toc141438867)

[3.1.4. Order Placement: 15](#_Toc141438868)

[3.1.5. Order Management: 15](#_Toc141438869)

[**3.2 Non-Functional Requirements** 15](#_Toc141438870)

[3.2.1. Performance: 15](#_Toc141438871)

[3.2.2. Security: 16](#_Toc141438872)

[3.3.3. Usability: 16](#_Toc141438873)

[3.3.4. Scalability: 16](#_Toc141438874)

[3.3.5. Reliability: 16](#_Toc141438875)

[3.3.6. Interoperability: 17](#_Toc141438876)

[3.3.7. Maintainability: 17](#_Toc141438877)

[**4. User Interface Design** 17](#_Toc141438878)

[4.1 Overview of User Interface 17](#_Toc141438879)

[4.2 User Interface Components 19](#_Toc141438880)

[**5. Database Design** 21](#_Toc141438881)

[5.1 Database Overview 21](#_Toc141438882)

[5.2 Database Schema 22](#_Toc141438883)

[**6. System Architecture** 24](#_Toc141438884)

[6.1 High-Level Architecture 24](#_Toc141438885)

[6.2 Detailed Components 26](#_Toc141438886)

[**7. Use Case Diagram** 28](#_Toc141438887)

[7.1 Use Case Descriptions 28](#_Toc141438888)

[7.2 Use Case Diagrams 30](#_Toc141438889)

[**8. Test Plan** 30](#_Toc141438890)

[8.1 Test Scope 30](#_Toc141438891)

[8.2 Test Approach 32](#_Toc141438892)

[8.3 Test Cases 34](#_Toc141438893)

[**9. Deployment** 36](#_Toc141438894)

[9.1 System Requirements 36](#_Toc141438895)

[9.2 Installation and Configuration Instructions 38](#_Toc141438896)

[**10. Maintenance and Support** 38](#_Toc141438897)

[10.1 Maintenance Plan 38](#_Toc141438898)

# **1. Introduction**

## 1.1 Purpose

The purpose of the Lunch Menu System is to provide a web-based platform that is mobile responsive for managing lunch orders for NFT Staff.

## 1.2 Scope

The system will allow the staff to view available food items, choose items of choice, manage their orders, and manage their account details. Additionally, administrators will have access to manage food items and view order reports.

## 1.3 Intended Audience

This Software Requirements Specification (SRS) document is intended for the following groups of people;

1. Developers: The document provides detailed technical specifications and functional requirements that developers will use as a guide to design and implement the Lunch Menu System.

2. Testers: Testers will refer to the SRS document to understand the expected behavior of the system and use it as a basis for creating test cases to validate the system's functionality.

3. Project Managers: Project managers will use the SRS to gain insights into the scope of the project, the requirements, and the estimated effort required for development and testing.

4. Stakeholders: Stakeholders will review the SRS to understand the features and capabilities of the Lunch Menu System and ensure that it aligns with their business needs.

5. Designers: UI/UX designers will refer to the SRS to understand the user interface requirements and design the visual aspects of the system accordingly.

6. Quality Assurance: Quality assurance teams may also refer to the SRS to ensure that the system meets quality standards and fulfills all specified requirements.

The SRS document serves as a comprehensive reference for all individuals involved in the development, testing, and management of the Lunch Menu System, ensuring a common understanding of the project's goals and functionalities.

# **2. Overall Description**

## 2.1 Product Perspective

The Lunch Menu System plays a crucial role in the larger context of the pantry environment by providing an efficient and streamlined solution for managing the lunch menu and orders. Here's how the Lunch Menu System fits into the larger context:

### 2.1.1 Menu Management:

The Lunch Menu System allows the pantry and company to easily manage the daily or weekly lunch menu offerings. They can update and customize the menu based on seasonal availability, special occasions, or customer preferences. This ensures that the menu remains diverse and appealing to customers.

### 2.1.2. Order Placement:

The system enables the NFT Staff to place their lunch orders online in advance. This eliminates the need for manual order-taking and reduces unnecessary costs on food that will not be needed.. It streamlines the order processing and minimizes errors in order preparation.

### 2.1.3. Inventory Management:

With the Lunch Menu System, the pantry staff can keep track of the inventory for each menu item. It helps in managing food supplies and prevents overstocking or shortages, optimizing cost and reducing food wastage.

### 2.1.4. Real-time Updates:

The system provides real-time updates to customers about menu changes, availability, and order status. This enhances transparency and allows customers to make informed decisions about their lunch choices.

### 2.1.5. Customer Loyalty and Feedback:

The system can collect customer feedback on the menu items and service, helping the mangers to continuously improve the offerings.

### 2.1.6. Data Analytics:

The Lunch Menu System generates valuable data on customer preferences, and popular menu items. NFT managers can use this data for strategic decision-making, such as menu optimization and marketing strategies.

Overall, the Lunch Menu System enhances operational efficiency, customer satisfaction, and profitability in NFT. It empowers both managers and staff with user-friendly features and data-driven insights, making it an integral part of the dining experience.

## **2.2 Product Features**

The Lunch Menu System is a comprehensive web-based application designed to streamline and enhance the lunch ordering process in NFT. It offers various key features that cater to both administrators and other staff. Below is an overview of the key features of the system:

### 2.2.1. User Registration and Authentication:

- Staff can create an account by registering with their email and password.

- The system ensures user authentication, allowing only registered users to access their accounts and place orders.

- User registration provides personalized experiences, order history tracking.

### 2.2.2. Food Item Management:

- Pantry staff can manage the daily or weekly lunch menu offerings through the system's dashboard.

- The system allows staff to add, edit, or remove food items from the menu based on availability, popularity, and seasonal variations.

- Each food item is associated with attributes such as name, description, price, and image for better presentation.

### 2.2.3. Order Placement and Customization:

- Customers can browse the menu, select their preferred food items, and add them to their order list.

- The system facilitates order customization by allowing customers to specify preferences.

### 2.2.4. Real-time Availability and Updates:

- The system provides real-time updates on the availability of food items to avoid disappointment due to selection of unavailable items.

- Any changes to the menu or food item availability are instantly reflected for both managers and staff.

### 2.2.5. Order Management and Processing:

- The system centralizes order management for staff.

- Staff can view and process incoming orders, ensuring timely and necessary preparation.

#- Order status updates are sent to customers, keeping them informed about the progress of their orders.

### 2.2.6. Reporting and Analytics:

- The Lunch Menu System generates reports and analytics based on stuff orders, menu popularity, and trends.

- These insights aid managers in making data-driven decisions for menu optimization and business growth.

Overall, the Lunch Menu System offers a seamless and user-friendly experience for both managers and staff, optimizing the lunch ordering process and improving customer satisfaction.

## **2.3 User Classes and Characteristics**

Certainly! Based on the users you've mentioned, let's redefine the user types and their specific roles and characteristics in the Lunch Menu System:

### 2.3.1. Pantry Staff:

- Characteristics:

- The pantry staff are responsible for managing the cafeteria's daily operations and food preparation.

- They handle the kitchen tasks, food item availability, and order processing.

- Roles:

- Manage the Lunch Menu: The pantry staff can add, edit, or remove food items from the menu based on availability and changes in the menu plan.

- Update Food Availability: They update the real-time availability status of food items to inform users about available options.

- Process Incoming Orders: When an order is placed, pantry staff view and process the order for preparation and delivery.

- Monitor Inventory: They keep track of food supplies and ingredients to ensure timely restocking.

- Generate Reports: They may generate reports on popular menu items and sales trends for management analysis.

### 2.3.2. Company Managers and Overall Staff:

- Characteristics:

- Company managers and overall staff are the employees or users within the organization where the cafeteria is located.

- They use the Lunch Menu System to place food orders for themselves or on behalf of their team members.

- They may have varying levels of access and privileges based on their positions in the company.

- Roles:

- Place Orders: Company managers and overall staff can browse the lunch menu and place food orders for themselves or their team members.

- Group Orders: Managers may have the ability to place bulk orders for team meetings or events.

- Set Budgets: Depending on the organization's policies, managers might be able to set budgets for team members' food orders.

- Access Reports: Managers may have access to reports showing overall spending on food orders by their team.

- Manage Team Members: In some cases, managers can manage their team members' accounts and place orders on their behalf.

## **2.4 Operating Environment**

To run the Lunch Menu System, you will need the following hardware, software, and network environment:

### 2.4.1 Hardware Requirements:

1. Server: A dedicated server or cloud-based hosting service to host the Lunch Menu System application and database. The server should have sufficient processing power, memory, and storage capacity to handle user requests and store data securely.

2. Client Devices: The system should be accessible from various client devices, such as desktop computers, laptops, tablets, and smartphones. The hardware requirements for client devices will depend on the platform they are running (Windows, macOS, iOS, Android, etc.).

### 2.4.2 Software Requirements:

1. Operating System: The server should run on a stable and secure operating system. Common choices include Linux distributions like Ubuntu, CentOS, or Red Hat, or Windows Server.

2. Web Server: Install a web server software such as Apache, Nginx, or Microsoft Internet Information Services (IIS) to handle HTTP requests from client devices and serve the Lunch Menu System's web pages.

3. Database Management System: The system will require a relational database management system (RDBMS) to store user data, food items, orders, and other relevant information. The system will use ....

4. Programming Language: The Lunch Menu System is developed using Python, along with a web framework, Django, to handle the backend logic and serve dynamic web pages.

5. Frontend Technologies: The frontend of the system can be developed using HTML, CSS, and JavaScript along with libraries or frameworks like Bootstrap and React to create a responsive and user-friendly interface.

### 2.4.3 Network Environment:

1. Local Area Network (LAN): The Lunch Menu System will be accessible within the organization's premises, which requires a local area network connecting the server and client devices.

2. Internet Connectivity: If the system is accessible from outside the organization, a stable and secure internet connection is essential to ensure smooth communication between the server and external client devices.

3. Firewall and Security Measures: Implement proper firewall configurations and security measures to protect the system from unauthorized access and cyber threats.

## **2.5 Design and Implementation Constraints**

### 2.5.1. Time Constraints:

The project may have a strict timeline for development and implementation, which could limit the amount of time available for extensive testing and refinement.

### 2.5.2. Technical Expertise:

The development team's level of expertise and familiarity with the chosen technologies may impact the complexity of the system's design and features.

### 2.5.3 Compatibility:

The Lunch Menu System needs to be compatible with a wide range of devices and web browsers, which can pose challenges in ensuring a consistent user experience across different platforms.

### 2.5.4. Data Security:

As the system may involve handling sensitive user data, ensuring robust security measures and compliance with data protection regulations is essential.

### 2.5.5. Scalability:

The system should be designed to accommodate future growth and increasing user demands. Failure to plan for scalability can lead to performance issues as the user base grows.

2.5.6. Network Reliability:   
The Lunch Menu System relies on internet connectivity and the organization's network infrastructure. Any network outages or instability could impact the system's availability.

### 2.5.7. Integration with Existing Systems:

If the system needs to integrate with other existing systems (e.g., HR databases, payment gateways), compatibility and data exchange mechanisms need to be carefully planned and implemented.

### 2.5.8. Maintenance and Support:

Adequate provisions for ongoing maintenance and technical support need to be in place to address any issues or bugs that may arise after deployment.

### 2.5.9. Legal and Regulatory Compliance:

The system should adhere to relevant industry standards, data privacy regulations, and local laws to avoid legal complications.

### 2.5.10. Performance:

Ensuring the system performs efficiently, especially during peak hours, is crucial to prevent slowdowns or system crashes.

2.5.11. User Interface and User Experience:   
A user-friendly and intuitive interface is essential for staff members to easily navigate and utilize the system's features effectively.

Identifying and addressing these constraints and limitations during the design and implementation phases will help create a more robust and successful Lunch Menu System.

## **2.6 Assumptions and Dependencies**

### 2.6.1 Assumptions:

Availability of Internet: The Lunch Menu System assumes that users will have access to a stable internet connection to use the system. It is designed to be a web-based application.

User Access Levels: The system assumes that different user roles (pantry staff, company managers, and overall staff) will have specific access levels and permissions to perform their respective tasks.

Food Item Management: It is assumed that the pantry staff will be responsible for updating and managing the food items available in the menu.

User Training: The assumption is that users will receive adequate training and documentation to understand how to use the system effectively.

Compatibility: The system assumes that users will access the application from modern web browsers and devices that support the required technologies.

Data Accuracy: It is assumed that the data provided by users, such as user registration details and food orders, will be accurate and reliable.

### 2.6.2 External Dependencies:

Database Server: The Lunch Menu System depends on a reliable and secure database server to store user information, food items, and order data.

Email Services: The system may use external email services to send notifications, password reset links, and other communication to users.

Internet Service Providers: The system's availability relies on the internet service providers used by the users.

Hardware Infrastructure: The system may depend on the hardware infrastructure provided by the organization, including servers, routers, and client devices.

Web Hosting Services: If the Lunch Menu System is hosted externally, it will depend on the reliability and performance of the web hosting service.

Data Privacy Regulations: The system may need to comply with external data privacy regulations, and its design should consider these legal requirements.

Proper consideration of assumptions and external dependencies is crucial during the requirements gathering process to ensure that the Lunch Menu System is well-designed and meets the needs of all stakeholders.

# **3. System Features and Requirements**

## **3.1 Functional Requirements**

### 3.1.1. User Registration:

Functional Requirements:

a. Users should be able to register for an account by providing their email, first name, last name, and a unique username.

b. The system should validate the email to ensure it is in a proper format and not already registered.

c. Upon successful registration, users should receive a confirmation email.

Use Cases:

- A new user visits the registration page and fills in the required details, including email, first name, last name, and username. They click the "Register" button to create an account.

- The system checks if the email is valid and not already in use. If valid, it sends a confirmation email to the user.

### 3.1.2. User Login:

Functional Requirements:

a. Registered users should be able to log in to the system using their username and password.

b. The system should authenticate user credentials and grant access to the appropriate user role (pantry staff, company manager, or overall staff).

Use Cases:

- A registered user visits the login page and enters their username and password. They click the "Login" button to access their account.

- The system verifies the user's credentials and redirects them to the appropriate dashboard based on their user role.

### 2.1.3. Food Item Management:

Functional Requirements:

a. Pantry staff (administrators) should be able to add, edit, and delete food items from the menu.

b. Each food item should have attributes such as name, description, and category.

c. The system should validate the input for food items to ensure data integrity.

d. Company managers can view the list of food items but cannot modify them.

Use Cases:

- A pantry staff member logs in to the system and navigates to the food item management section. They add a new food item with all the required details.

- The system validates the input for the new food item and adds it to the menu.

- A company manager logs in to the system and views the list of available food items but cannot edit or delete them.

### 3.1.4. Order Placement:

Functional Requirements:

a. Registered users should be able to place food orders from the available menu items.

b. Users can select food items and specify quantities for each item in their order.

c. Users should be able to submit their orders and view and edit their orders.

Use Cases:

- A user logs in to the system and visits the food menu. They select the desired food items and specify the quantity for each item. They click the "Submit" button to proceed.

### 3.1.5. Order Management:

Functional Requirements:

a. Pantry staff and company managers should be able to view and manage incoming food orders.

b. The system should maintain a record of all orders.

Use Cases:

- A pantry staff member logs in to the system and accesses the order management section. They view a list of all orders.

These functional requirements and use cases provide a foundation for developing a Lunch Menu System that meets the needs of its users and stakeholders. Further refinement and collaboration with stakeholders will help in ensuring the system's success.

## **3.2 Non-Functional Requirements**

Non-Functional Requirements for Lunch Menu System:

### 3.2.1. Performance:

a. Response Time: The system should provide quick responses to user actions, such as login, order placement, and menu navigation.

b. Load Time: Web pages and menus should load efficiently to ensure a smooth user experience.

c. Scalability: The system should be able to handle an increasing number of users and food items without a significant decrease in performance.

### 3.2.2. Security:

a. User Authentication: The system should employ secure user authentication mechanisms to prevent unauthorized access.

b. Data Privacy: User data, including personal information and order history, should be stored securely and encrypted.

c. Password Protection: User passwords should be stored in a hashed format to enhance security.

d. Access Control: Different user roles (pantry staff, company managers, and overall staff) should have distinct levels of access to system functionalities.

e. Protection against Attacks: The system should implement measures to prevent common web vulnerabilities, such as SQL injection and cross-site scripting (XSS) attacks.

### 3.3.3. Usability:

a. User-Friendly Interface: The system should have an intuitive and user-friendly interface that is easy to navigate and understand.

b. Responsive Design: The user interface should be responsive and adaptable to various devices, including desktops, tablets, and mobile phones.

c. Error Handling: The system should provide clear and informative error messages to guide users in resolving issues.

d. Accessibility: The system should be designed to accommodate users with disabilities, adhering to accessibility standards.

### 3.3.4. Scalability:

a. Database Scalability: The system's database should be capable of accommodating a growing number of food items and user accounts without performance degradation.

b. Concurrent Users: The system should support a significant number of concurrent users without compromising performance.

c. Load Balancing: Load balancing mechanisms should be implemented to distribute user requests evenly across servers and ensure high availability.

### 3.3.5. Reliability:

a. System Uptime: The system should have high availability with minimal downtime for maintenance and upgrades.

b. Data Backup: Regular data backups should be performed to prevent data loss in case of system failures.

c. Error Handling and Recovery: The system should be equipped to handle unexpected errors and recover gracefully without data loss.

### 3.3.6. Interoperability:

a. Integration with External Services: The system should be able to integrate with external services, such as email services for order notifications.

b. Compatibility: The system should work seamlessly with various web browsers and operating systems.

### 3.3.7. Maintainability:

a. Code Maintainability: the systems codebase should be well-structured and documented for ease of maintenance and future enhancements.

b. Modularity: The system should be designed with a modular approach, allowing easy updates or additions of features.

8. Legal and Regulatory Compliance:

a. Data Protection: The system should comply with data protection regulations and user data privacy laws.

b. Food Safety Standards: If applicable, the system should adhere to food safety and hygiene regulations.

These non-functional requirements ensure that the Lunch Menu System is not only feature-rich but also performs efficiently, remains secure, and meets user expectations while adhering to legal and regulatory standards.

# **4. User Interface Design**

## 4.1 Overview of User Interface

The overall user interface design of the Lunch Menu System should be intuitive, user-friendly, and visually appealing. It should provide a seamless experience for all types of users, including pantry staff, company managers, and overall staff. Here are the key aspects of the user interface design:

1. Navigation:

a. Clear and Consistent Navigation: The navigation menu should be easy to find and consistent across all pages of the system. It should use standard navigation conventions to help users easily access different sections and functionalities.

b. Breadcrumbs: Breadcrumbs should be provided to show users their current location within the system and allow them to navigate back to previous pages.

c. Search Functionality: The system should include a search bar to enable users to quickly find specific food items or orders.

2. Layout:

a. Responsive Design: The user interface should be responsive and adapt to different screen sizes, including desktops, tablets, and mobile phones, ensuring a consistent experience across devices.

b. Organized and Structured: Information should be presented in a clear and organized manner, with well-defined sections and headings to avoid clutter.

c. Grids and Cards: Food items and order details should be displayed using grids and cards, allowing users to view multiple items at a glance.

3. Visual Elements:

a. Color Scheme: A visually pleasing color scheme should be used to enhance the overall appeal of the system. Color differentiation can be used to distinguish different sections and elements.

b. Typography: Clear and readable fonts should be used for all text elements to ensure ease of reading.

c. Icons: Icons should be used to represent actions and functionalities, making the interface more intuitive and reducing the need for textual explanations.

d. Imagery: High-quality images of food items can be used to enhance the visual appeal of the menu.

4. Forms and Input Fields:

a. User-Friendly Forms: Forms for user registration, login, and order placement should be designed in a user-friendly manner with clear labels and appropriate input fields.

b. Validation and Error Messages: Form input should be validated, and informative error messages should be displayed if the user enters incorrect or incomplete information.

5. Interactive Elements:

a. Buttons and Call-to-Action: Interactive elements like buttons should be visually distinct and easily clickable to encourage user actions.

b. Hover and Click Effects: Interactive elements should have hover and click effects to provide visual feedback to users.

6. User Roles and Permissions:

a. Differentiated User Interfaces: Depending on the user role (pantry staff, company managers, overall staff), the system's interface may display different options and functionalities.

b. Access Control: Certain actions and functionalities may be restricted based on user roles to ensure data security and privacy.

Overall, the user interface design of the Lunch Menu System should prioritize ease of use, simplicity, and a visually appealing layout to create an enjoyable experience for all users and encourage them to use the system efficiently for food item management and order placement.

## 4.2 User Interface Components

The Lunch Menu System user interface comprises various components that facilitate different functionalities. Here are the individual components with a brief description of each:

1. Login Form:

- Description: The login form is the initial entry point for users to access the system. It allows users to authenticate themselves using their credentials (username/email and password) to gain access to their accounts and perform specific actions based on their roles.

- Key Elements: Input fields for username/email and password, a "Remember Me" checkbox, a "Forgot Password" link, and a "Sign In" button.

2. Registration Form:

- Description: The registration form enables new users to create an account within the system. It collects necessary information from users, such as username/email and password, to register them as regular users.

- Key Elements: Input fields for username/email, password, and password confirmation, along with a "Sign Up" button.

3. Food Item Management Page:

- Description: This page allows pantry staff and company managers to manage food items available in the menu. They can add, edit, and delete food items from the system.

- Key Elements: List of existing food items, buttons for adding new items and editing existing ones, and options for deleting items.

4. Order Placement Page:

- Description: This page allows regular users and overall staff to place lunch orders. Users can select food items from the menu and place orders for a specific date.

- Key Elements: List of available food items with checkboxes for selection, a date picker for choosing the order date, and a "Submit" button.

5. Order History Page:

- Description: This page displays the order history of users, showing their past lunch orders and order details.

- Key Elements: List of previous orders, order date, food items ordered, and the option to view order details.

6. User Profile Page:

- Description: The user profile page allows users to view and update their account details. Regular users and company managers can change their account information and update passwords if needed.

- Key Elements: User details (username/email), options to update account information, and change password.

7. Navigation Bar:

- Description: The navigation bar is present on all pages and provides links to different sections of the system. It enables users to switch between various functionalities easily.

- Key Elements: Links to Home, Order, Food Items, Order History, and User Profile.

8. Error Messages and Notifications:

- Description: The user interface should display error messages and notifications to inform users about any incorrect input or successful operations, such as successful order placement or password change.

9. Logout Button:

- Description: The logout button allows users to log out of their accounts, ensuring data security and privacy.

These components work together to create a seamless and user-friendly experience for all users, enabling them to interact with the Lunch Menu System effectively.

# **5. Database Design**

## 5.1 Database Overview

The purpose of the database in the Lunch Menu System is to store and manage all the essential information required for the system to function effectively. It stores data related to user accounts, food items, lunch orders, and other relevant details. The database structure is designed to maintain data consistency, integrity, and support efficient retrieval of information.

The Lunch Menu System's database consists of several tables that represent different entities within the system. The structure of the database is organized in a way to establish relationships between these tables, ensuring data coherence and enabling the system to handle complex queries.

1. User Table:

- Purpose: This table stores information about all types of users, including pantry staff, company managers, and regular users. It includes fields such as user ID, username, email, password, first name, last name, and user type (admin, staff).

2. FoodItem Table:

- Purpose: The FoodItem table stores details about different food items available in the cafeteria's menu. It includes fields like food item ID, name, and any other relevant information about the food.

3. LunchOrder Table:

- Purpose: This table holds data related to lunch orders placed by users. It establishes a many-to-many relationship with the User and FoodItem tables, as a single order can involve multiple food items and is associated with a specific user.

- Fields: user ID (foreign key), order date, and a boolean field indicating whether the order has been placed.

The relationships between the tables are as follows:

- One-to-Many (User and LunchOrder): Each user can have multiple lunch orders, but each lunch order belongs to a single user. This relationship is established through the foreign key "user ID" in the LunchOrder table, which references the primary key "user ID" in the User table.

- Many-to-Many (LunchOrder and FoodItem): A single lunch order can include multiple food items, and each food item can be a part of multiple lunch orders. This relationship is established through an intermediate table that stores the combinations of order IDs and food item IDs, linking the LunchOrder and FoodItem tables.

The structure and relationships between the database tables ensure that the Lunch Menu System can accurately store and retrieve user information, food items, and lunch orders efficiently. The design promotes data consistency and integrity, allowing the system to operate smoothly and provide a seamless experience to users while managing their lunch orders and preferences.

## 5.2 Database Schema

Below is the detailed schema for the database of the Lunch Menu System, including the fields and data types for each table:

1. User Table:

- Fields:

- user\_id (Primary Key, Integer)

- username (String)

- email (String)

- password (String)

- first\_name (String)

- last\_name (String)

- user\_type (String: choices: "admin", "staff")

2. FoodItem Table:

- Fields:

- food\_item\_id (Primary Key, Integer)

- name (String)

- description (String)

3. LunchOrder Table:

- Fields:

- user\_id (Foreign Key to User table)

- order\_date (Date)

- is\_ordered (Boolean)

4. LunchOrderFoodItem (Intermediate) Table:

- Fields:

- id (Primary Key, Integer)

- order\_id (Foreign Key to LunchOrder table)

- food\_item\_id (Foreign Key to FoodItem table)

Detailed Explanation:

1. User Table:

- The User table stores information about different users of the system.

- user\_id: A unique identifier for each user.

- username: The username of the user, used for login purposes.

- email: The email address of the user, also used for login and communication.

- password: The hashed password of the user for authentication.

- first\_name: The first name of the user.

- last\_name: The last name of the user.

- user\_type: The type of user, whether "admin," or "staff,"

2. FoodItem Table:

- The FoodItem table stores information about different food items available in the cafeteria's menu.

- food\_item\_id: A unique identifier for each food item.

- name: The name of the food item.

- description: A description of the food item.

3. LunchOrder Table:

- The LunchOrder table stores details of lunch orders placed by users.

- user\_id: A foreign key linking each order to the user who placed it.

- order\_date: The date on which the lunch order is placed.

- is\_ordered: A boolean field indicating whether the order has been placed or not.

4. LunchOrderFoodItem (Intermediate) Table:

- The LunchOrderFoodItem table establishes a many-to-many relationship between LunchOrder and FoodItem tables.

- id: A unique identifier for each entry in this table.

- order\_id: A foreign key linking each entry to the lunch order it is associated with.

- food\_item\_id: A foreign key linking each entry to the food item it represents in the order.

This database schema provides a comprehensive structure to store and manage data related to users, food items, and lunch orders in the Lunch Menu System. The relationships between the tables enable efficient retrieval of information and maintain data consistency and integrity throughout the system.

# **6. System Architecture**

## 6.1 High-Level Architecture

The high-level architecture of the Lunch Menu System consists of several main components that interact to facilitate the functioning of the system. Here is the overview of the main components and their interactions:

1. User Interface (UI):

- The User Interface is the front-end component of the system that allows users to interact with the system.

- It includes various pages such as login, registration, food item selection, order placement, and user profile.

- Users can access the UI through web browsers or mobile devices.

2. Views (Controller):

- Views act as the controller component in the architecture, handling user requests and directing them to appropriate actions.

- They process the data received from the user interface and interact with the database and other components as needed.

3. Models:

- The Models represent the data and business logic of the Lunch Menu System.

- They define the structure of the database and handle data manipulation and validation.

- Models interact with the database to retrieve and store data related to users, food items, and lunch orders.

4. Database:

- The Database stores all the data related to users, food items, and lunch orders.

- It is responsible for data persistence and retrieval.

5. Authentication and Authorization:

- The system implements authentication and authorization mechanisms to control user access and actions.

- Users must authenticate themselves before accessing certain features, and authorization checks ensure that users have the required permissions for specific actions.

6. Business Logic:

- The Business Logic component contains the core logic of the Lunch Menu System.

- It handles various processes, such as order placement, order tracking, and user authentication.

7. External Services:

- The Lunch Menu System may interact with external services, such as email services for user notifications.

Interactions:

- User interacts with the User Interface to perform actions such as login, registration, selecting food items, and placing orders.

- The User Interface sends requests to the Views (Controller) to process user actions.

- The Views interact with the Models to retrieve data from the database or perform data manipulation based on user requests.

- The Models interact with the Database for data storage and retrieval.

- During the order placement process, the Business Logic ensures that the user's selected food items are valid and available for ordering.

- The Authentication and Authorization components verify the user's credentials and permissions for accessing certain features.

- If needed, the system interacts with external services, such as sending order confirmation emails or processing payments.

Overall, the high-level architecture of the Lunch Menu System ensures a seamless flow of information and data between its main components, providing users with an efficient and user-friendly experience.

## 6.2 Detailed Components

The Lunch Menu System consists of multiple components, including the frontend, backend, and database layers. Here is a detailed description of each component:

1. Frontend:

- The Frontend is the user-facing part of the system responsible for presenting the user interface and enabling user interactions.

- It is typically implemented using web technologies like HTML, CSS, and JavaScript.

- The frontend renders web pages that allow users to perform actions such as login, registration, food item selection, and order placement.

- It communicates with the backend through HTTP requests to send user input and receive data and updates from the server.

2. Backend:

- The Backend is the server-side component of the Lunch Menu System responsible for processing user requests, business logic, and data management.

- It is implemented using a server-side programming language such as Python (using Django or Flask framework).

- The backend handles user authentication, authorization, and user-related actions such as login and registration.

- It manages the logic for order placement, order tracking, and other business operations related to the lunch menu.

- The backend communicates with the database layer to retrieve and store data and responds to frontend requests with appropriate data and status codes.

3. Database Layer:

- The Database Layer is responsible for managing the storage and retrieval of data required for the Lunch Menu System.

- It can be implemented using a relational database management system (RDBMS) like PostgreSQL, MySQL, or SQLite.

- The database stores information about users, food items, lunch orders, and any other necessary data.

- The backend communicates with the database layer through database queries to perform CRUD operations (Create, Read, Update, Delete) on the data.

The flow of data and interactions between these components is as follows:

1. User interacts with the Frontend by accessing the Lunch Menu System through a web browser or a mobile application.

2. The Frontend sends HTTP requests to the Backend, containing user input and actions.

3. The Backend processes the requests, performs the required business logic, and interacts with the Database Layer to store or retrieve data.

4. The Backend sends a response back to the Frontend, containing data and status codes, based on the request processing.

5. The Frontend displays the received data to the user and updates the user interface accordingly.

In summary, the Frontend, Backend, and Database Layers work together to provide a seamless and interactive experience for users, allowing them to access and manage the lunch menu system efficiently.

# **7. Use Case Diagram**

## 7.1 Use Case Descriptions

1. Use Case: User Registration

- Actor: Regular User

- Preconditions: The user is accessing the Lunch Menu System for the first time and has not registered before.

- Postconditions: The user's account is created in the system, and they can log in with their registered credentials.

2. Use Case: User Login

- Actor: Regular User

- Preconditions: The user is already registered in the system with a valid account.

- Postconditions: The user is successfully authenticated and granted access to their dashboard.

3. Use Case: View Food Menu

- Actor: Regular User, Company Managers

- Preconditions: The user is logged into the system and navigates to the food menu page.

- Postconditions: The user can view the list of available food items and their details.

4. Use Case: Place an Order

- Actor: Regular User and Company Managers

- Preconditions: The user is logged in, and the lunch menu ordering window is open.

- Postconditions: The user successfully places an order for selected food items on a specified date.

5. Use Case: View Order History

- Actor: Regular User, Company Managers

- Preconditions: The user is logged in, and the order history page is accessible.

- Postconditions: The user can view their previous lunch orders, including dates and food items.

6. Use Case: Manage Food Items

- Actor: Company Managers

- Preconditions: The manager is logged in with admin privileges and navigates to the food item management page.

- Postconditions: The manager can add, edit, or delete food items from the lunch menu.

7. Use Case: View Daily Orders

- Actor: Pantry Staff

- Preconditions: The pantry staff is logged in and has access to the daily order page.

- Postconditions: The pantry staff can view the list of orders placed for the current day.

8. Use Case: Manage Users

- Actor: Company Managers

- Preconditions: The manager is logged in with admin privileges and has access to the user management page.

- Postconditions: The manager can view and manage user accounts, including adding or removing users.

9. Use Case: Generate Reports

- Actor: Company Managers

- Preconditions: The manager is logged in with admin privileges and has access to the reporting feature.

- Postconditions: The manager can generate reports related to lunch orders, user activities, or food item popularity.

10. Use Case: Update Profile

- Actor: Regular User, Company Managers

- Preconditions: The user is logged in and navigates to the profile page.

- Postconditions: The user can update their profile information, such as name, email, or password.

11. Use Case: Logout

- Actor: Regular User, Company Managers

- Preconditions: The user is logged in and decides to log out of the system.

- Postconditions: The user is successfully logged out and redirected to the login page.

Note: The use cases described above are representative of typical interactions with the Lunch Menu System. The actual system may have additional use cases and variations based on specific business requirements and user roles.

## 7.2 Use Case Diagrams

Add the diagram

# **8. Test Plan**

## 8.1 Test Scope

The scope of testing for the Lunch Menu System will cover both functional and non-functional aspects of the application. The goal of testing is to ensure that the system meets the specified requirements and functions correctly for all user roles. The following areas and functionalities will be tested:

1. User Registration and Login:

- Test user registration process, including validation of input fields.

- Verify that users can log in successfully with valid credentials.

- Check for proper error handling for incorrect login attempts.

2. Food Item Management:

- Test the ability of managers to add, edit, and delete food items.

- Verify that the food items are displayed correctly on the menu page.

3. Order Placement:

- Test the user's ability to place an order for selected food items.

- Ensure that the ordering window opens and closes based on specified timings.

4. Order History:

- Verify that users can view their past lunch orders in the order history page.

- Check the accuracy of order dates, food items, and quantities.

5. User Management:

- Test the manager's ability to add and remove users from the system.

- Verify that user roles and permissions are applied correctly.

6. Daily Orders:

- Check that pantry staff can view the list of orders placed for the current day.

- Test the accuracy of order details displayed to the pantry staff.

7. Reporting:

- Test the generation of various reports, such as lunch order reports and user activity reports.

- Verify that the generated reports are accurate and reflect the requested data.

8. Performance:

- Test the system's response time under different user loads to ensure scalability.

- Check for any bottlenecks or performance issues that may affect user experience.

9. Security:

- Test the system for potential vulnerabilities, such as SQL injection or cross-site scripting.

- Verify that user data is securely stored and protected.

10. Usability:

- Conduct usability testing to evaluate the user interface and user experience.

- Check if the system is intuitive and easy to use for all types of users.

11. Compatibility:

- Test the system on different browsers and devices to ensure cross-browser compatibility.

- Check for responsive design and proper rendering on various screen sizes.

12. Accessibility:

- Verify that the system is accessible to users with disabilities, following WCAG guidelines.

13. Integration:

- Test the integration of different system components to ensure smooth data flow.

- Check for data consistency and synchronization.

The testing process will involve unit testing, integration testing, system testing, and acceptance testing to cover all aspects of the Lunch Menu System. Test cases will be designed to cover both positive and negative scenarios to ensure comprehensive test coverage. Additionally, regression testing will be performed after each system update to confirm that existing functionalities remain unaffected.

## 8.2 Test Approach

The approach to testing for the Lunch Menu System will involve a combination of manual testing and automated testing.

1. Manual Testing:

Manual testing will be conducted by QA testers who will interact with the system as end-users to validate its functionalities. Testers will follow predefined test cases and scenarios to ensure that all aspects of the system are thoroughly checked. Manual testing will be useful for validating user interfaces, usability, and user interactions.

2. Automated Testing:

Automated testing will be used to improve testing efficiency and consistency. Automated test scripts will be developed using testing frameworks and tools, such as Selenium or Cypress, to perform repetitive and regression tests. Automated testing will be especially beneficial for testing scenarios with a large number of test cases and for checking system responses under different loads.

The testing approach will be as follows:

1. Test Planning: Testers will collaborate with stakeholders to define the testing scope, objectives, and requirements. Test plans and test cases will be prepared based on functional and non-functional requirements.

2. Unit Testing: Developers will conduct unit testing to verify the correctness of individual components or modules of the system.

3. Manual Functional Testing: Testers will perform manual testing of the application's functional aspects, including user registration, login, order placement, and user management.

4. Automated Functional Testing: Automated test scripts will be created to test repetitive scenarios, such as order placement, and to conduct regression testing after system updates.

5. Performance Testing: Load testing and stress testing will be conducted to assess the system's performance under different user loads.

6. Security Testing: Penetration testing and vulnerability scanning will be performed to identify and mitigate security risks.

7. Usability Testing: Usability testing will be carried out to assess the user-friendliness and accessibility of the application.

8. Compatibility Testing: The system will be tested on different browsers, operating systems, and devices to ensure compatibility.

9. Acceptance Testing: User acceptance testing (UAT) will be conducted to obtain feedback from actual end-users and stakeholders before the system's final deployment.

10. Regression Testing: After each system update or bug fix, regression testing will be performed to verify that existing functionalities remain unaffected.

The combination of manual and automated testing will help ensure that the Lunch Menu System is thoroughly tested for its functionality, performance, security, usability, and compatibility. It will help identify and resolve defects early in the development process, leading to a more reliable and robust system.

## 8.3 Test Cases

Sure, here is a list of test cases for various scenarios and functionalities of the Lunch Menu System:

1. User Registration:

- Test case 1: Verify that a new user can successfully register with valid details.

- Test case 2: Verify that registration fails with existing email or username.

- Test case 3: Verify that all required fields are mandatory during registration.

2. User Login:

- Test case 4: Verify that a registered user can log in with valid credentials.

- Test case 5: Verify that login fails with invalid credentials.

- Test case 6: Verify that the system displays appropriate error messages for login failures.

3. User Profile:

- Test case 7: Verify that a user can view and edit their profile details.

- Test case 8: Verify that changes made to the user profile are saved successfully.

4. Food Item Management:

- Test case 9: Verify that an admin can add a new food item to the menu.

- Test case 10: Verify that an admin can update the details of an existing food item.

- Test case 11: Verify that an admin can delete a food item from the menu.

5. Order Placement:

- Test case 12: Verify that a user can view the list of available food items for the day.

- Test case 13: Verify that a user can add food items to their order.

- Test case 14: Verify that a user can remove food items from their order.

- Test case 15: Verify that the total price of the order is calculated correctly.

- Test case 16: Verify that a user can place the order successfully.

6. Order Management:

- Test case 17: Verify that an admin can view all orders placed for a specific day.

- Test case 18: Verify that an admin can mark an order as delivered.

- Test case 19: Verify that an admin can cancel an order if required.

7. Security Testing:

- Test case 20: Verify that user passwords are stored securely using encryption.

- Test case 21: Verify that access to admin functionalities is restricted to authorized users.

8. Performance Testing:

- Test case 22: Verify that the system can handle multiple simultaneous user logins.

- Test case 23: Verify that the system performs well under different user load conditions.

9. Usability Testing:

- Test case 24: Verify that the user interface is intuitive and easy to navigate.

- Test case 25: Verify that the application is responsive and works well on different devices.

10. Compatibility Testing:

- Test case 26: Verify that the application works correctly on different web browsers (Chrome, Firefox, Safari, Edge).

- Test case 27: Verify that the application is compatible with various operating systems (Windows, macOS, Linux).

These are just some examples of test cases, and additional test cases may be required based on the specific requirements of the Lunch Menu System. Each test case should include clear steps, expected outcomes, and any specific data or conditions required for testing.

# **9. Deployment**

## 9.1 System Requirements

System Requirements for Deploying the Lunch Menu System in a Production Environment:

1. Hardware Requirements:

- Server: A dedicated server or cloud-based virtual machine with sufficient CPU and RAM to handle the expected user load.

- Storage: Adequate storage space to store the application files, database, and any media files (such as food item images).

- Network: High-speed internet connection to ensure smooth communication between the server and users.

2. Software Requirements:

- Operating System: A compatible operating system such as Linux (e.g., Ubuntu, CentOS) or Windows Server.

- Web Server: Apache or Nginx to handle HTTP requests and serve the application.

- Database: A relational database management system (RDBMS) like PostgreSQL or MySQL to store application data.

- Python: The latest stable version of Python installed on the server to run the Django application.

- Django: The latest stable version of Django framework installed on the server.

- Additional Python Libraries: Any additional Python libraries required by the application (e.g., Pillow for image processing).

- Version Control System: Git to manage the source code and facilitate version control.

- Secure Socket Layer (SSL) Certificate: To enable secure communication over HTTPS.

3. Network Security:

- Firewall: A properly configured firewall to protect the server from unauthorized access and potential security threats.

- HTTPS: Use HTTPS with a valid SSL certificate to encrypt data transmitted between the server and clients.

4. Database Security:

- Secure Database Credentials: Ensure that database credentials are stored securely and not exposed in the application code.

- Regular Database Backups: Schedule regular database backups to prevent data loss in case of server failure or data corruption.

5. Performance:

- Caching: Implement caching mechanisms to improve application performance and reduce database queries.

- Load Balancer: If required, use a load balancer to distribute incoming traffic across multiple servers.

6. Scalability:

- Horizontal Scaling: Set up the application to be horizontally scalable, allowing it to handle increased traffic by adding more servers.

- Vertical Scaling: Ensure that the server hardware can be upgraded to handle increased resource demands.

7. Monitoring and Logging:

- Implement monitoring tools to track server performance, application health, and user activity.

- Set up logging to record application errors, user actions, and other relevant information for debugging and auditing purposes.

8. User Access and Authentication:

- Implement strong user access controls and authentication mechanisms to protect sensitive user data and ensure secure logins.

9. Maintenance and Updates:

- Plan regular maintenance and updates for the server, operating system, web server, and application to ensure security and performance.

10. Backup and Disaster Recovery:

- Establish a backup and disaster recovery plan to recover from data loss or server failure.

It is essential to review these system requirements regularly and keep them up-to-date to ensure the Lunch Menu System's smooth operation in a production environment. Additionally, it is advisable to consult with experienced IT professionals to ensure proper configuration and deployment of the system.

## 9.2 Installation and Configuration Instructions

Provide step-by-step instructions for installing and configuring the system.

# **10. Maintenance and Support**

## 10.1 Maintenance Plan

Outline the plan for ongoing