

Restaurant Management System

1. Introduction

1.1 Purpose

This document defines the functional and non-functional requirements of the Online Reservation and Order Management System and guides the development process.

1.2 Scope

The system will provide a web and mobile-based platform for a growing restaurant chain. Customers will be able to make table reservations, place orders, and leave feedback. Waiters will manage customer interactions, while the kitchen staff will receive order details. Managers will oversee system configurations, user accounts, and performance metrics.

1.3 Case (Scenario)

A group of entrepreneurs decided to open a specialty restaurant focused on fresh seafood and meat dishes. Initially a small business, the restaurant quickly gained a reputation for its high-quality meals, strict hygiene standards, and commitment to customer satisfaction. Word-of-mouth spread as customers raved about the freshness and flavor of the food, increasing the restaurant's popularity. Eventually, its name became widely known, and the business rapidly expanded into a restaurant chain.

However, growth brought new challenges. As the number of customers increased, so did the workload—waitstaff struggled to keep up with orders, and operational costs rose rapidly. The partners gathered to discuss how they could optimize business processes. They sought a solution that would both enhance the customer experience and ease the workload for employees. Ultimately, they decided to develop a user-friendly application and website that could keep pace with the restaurant's expansion.

This platform would allow customers to place orders easily, complete their transactions with minimal interaction with the waitstaff, and make restaurant operations more efficient. Moreover, it was designed to be accessible to people of all ages and income levels, rather than catering to a specific audience. For instance, special menus were designed for students, offering budget-friendly options tailored to their needs.

The partners decided to consult an experienced web and app developer to bring this project to life. However, instead of a basic system, they wanted a platform with a robust database, an elegant and user-friendly interface, and the ability to accommodate a large customer base. Thus, the restaurant's digital transformation journey began.

The system would work as follows: Customers will log in to the application using their password and email, fill in their information, make a reservation, and select their preferred table. Once a customer selects a table, it will be temporarily blocked for other customers for 10 minutes, and only the customer who made the selection can complete the reservation within this time. If the customer does not complete the reservation within the specified time, the system will automatically release the table, making it available for others to reserve. This

will prevent multiple people from attempting to access the same table simultaneously, avoiding potential confusion. Once the reservation has been completed, the customer will receive an SMS and email notification with the table number and reservation time. This way, when they arrive at the restaurant, their table will be ready, and they won't have to wait.

Customers can place orders either during the reservation process or after arriving at the restaurant. However, for orders placed before arriving at the restaurant, payment will need to be made in advance to allow the kitchen to prepare the meals. This can be done through the system. Orders placed at the restaurant can be paid either through the system or by cash or card, depending on the customer's preference. Additionally, customers wishing to reserve a table in advance will need to pay a fee. This system will streamline the restaurant's operations and improve the customer experience, making it more convenient. At the same time, it will prevent potential issues for the restaurant owners. Reservations will also be renewed weekly, preventing long-term conflicts and ensuring that the restaurant's capacity is utilized in the most efficient way.

Waitstaff would also be able to log into the system using their email and password, where they could see which customers were scheduled to arrive and at what time. However, reservations could only be canceled at least 2 hours before the scheduled arrival time; otherwise, the payment would not be refunded. In case of a cancellation, both the managers and the waitstaff would be immediately notified. Customers could call for a waiter through the system, and the notification would reach the waiter via the application. If a waiter is unavailable (for example, if they are on leave or not scheduled for a shift), they would not receive notifications from the system.

The system would provide great convenience not only for customers and waiters but also for the chefs in the kitchen. Ingredients that customers might be allergic to or dislike would be directly communicated to the chefs, allowing the kitchen team to prepare dishes that suit each customer's taste. This feature perfectly aligned with the restaurant's goal of catering to customers from all age groups and with diverse dietary habits.

Additionally, a powerful management panel was needed, not only for customers and staff but also for the managers. Therefore, a comprehensive management module was developed to enable control of the system. Managers would be able to add and update menu items and categories, as well as manage tables. They would also be able to monitor reservation statistics and audit employee accounts by managing system settings. To optimize the restaurant's operations, they would be able to analyze peak hours, the most preferred dishes, and customer behaviors. This would allow them to make strategic decisions to both increase customer satisfaction and enhance the efficiency of business processes.

As the restaurant expanded rapidly, the management team decided to design a new system to manage all operations more efficiently and securely. Firstly, it was recognized that each user would access the system with different permissions. To tackle this, a role management mechanism was developed, allowing everyone to perform actions suitable for their tasks.

Additionally, the application would feature a menu containing information such as the prices of dishes and drinks, allergen details, and content descriptions. Customers would also be able to track the preparation process of their ordered dishes through the system.

One of the most important features was the customer feedback system. After dining, customers would be able to rate the staff service, the quality of the dishes, and the overall atmosphere of the restaurant. However, in order to leave feedback, the customer's reservation or order must be verified. These feedbacks would be regularly reviewed by the restaurant owners and contribute to the continuous improvement of the service.

The restaurant's digital transformation would not be limited to establishing a secure and fast system, but also aimed to provide an accessible and user-friendly experience. For this reason, the platform would be optimized to work seamlessly on both desktop and mobile browsers. Users would be able to easily make reservations, browse the menu, and place orders whether they are at home or at the restaurant.

Given the growing number of customers and the expanding chain network, the platform was designed to effectively manage high-volume traffic. The system would support at least 500 active users daily, and to ensure high performance, reservation processes would be optimized to respond within a maximum of 5 seconds.

Additionally, the system needed to be accessible to everyone. The user interface would be designed according to the WCAG 2.1 accessibility standards, making it easy to use for disabled users as well. Features such as large text options, voice command support, and a design compatible with screen readers would ensure that everyone could easily navigate the system.

To prevent data loss, the system would carry out automatic daily backups, and any changes to reservations would undergo an approval process before being finalized. This would prevent accidental changes or malicious interventions, ensuring that the restaurant's operations continued securely and without disruption.

The system would also have a robust security infrastructure. Managers would manage data security protocols to protect customer and employee information. They would be able to monitor the system's daily activities and take necessary actions when suspicious logins or unusual activities were detected. User privacy would be prioritized, and all personal data would be carefully protected. Customer and staff information would be securely stored, and unauthorized access to personal data would be blocked.

During the digital transformation process, the system needed to be not only fast and user-friendly but also to maintain the highest level of data security. For this reason, passwords in the developed system would not be stored directly. Instead, they would be securely protected by hashing them using the Argon2 algorithm. This would ensure that even in the case of a cyber attack, user passwords could not be compromised, minimizing the security risk.

Sensitive data such as customer reservations, payment information, and contact details would be protected using the AES-256 encryption method. With this system, only authorized individuals would have access to this data, ensuring maximum security against external threats.

Additionally, in order for the system to operate securely over the internet, only the HTTPS (TLS 1.3) protocol would be supported. This would ensure that all data flow between

customers and the restaurant server would be encrypted and protected, providing security against cyber attacks.

Thanks to all these measures, customers would be able to make reservations with peace of mind, waitstaff and managers would use the system securely, and restaurant owners could focus on their work, knowing that operations were both efficient and secure.

At the heart of the system would be the PostgreSQL relational database. This powerful infrastructure would securely store user accounts, reservations, waitstaff and chef information, customer reviews, and many other important data in a structured manner. The main data tables would include customer authentication, reservation scheduling, restaurant and table availability, and customer feedback records, ensuring that each component of the system functions efficiently and orderly.

The development team will follow an agile methodology to carry out all stages of the system in an organized manner, with the goal of launching the application within a year. This process will include requirements analysis, UI/UX design, database design, backend development, testing, and deployment. Each phase will be carefully designed to ensure that the system is user-friendly, secure, and efficient.

To ensure the system can function seamlessly under high traffic, load balancing mechanisms will be implemented. This will prevent system failures during sudden increases in users or peak demand periods, ensuring uninterrupted service for users at all times. Additionally, secure session management will be integrated to keep user data safe. This secure management will protect user data and prevent unauthorized access.

Data security will be prioritized at every stage of the system. Regular backups will prevent data loss, ensuring that potential data loss issues are avoided. Strict access control policies will be enforced so that only authorized individuals can access the system, and unauthorized changes will be prevented.

As a result, the developed system will become a reliable tool for both service providers and restaurants, offering seamless planning, effective customer management, and secure data processing. A highly efficient, secure, and smooth experience will be provided for both users and restaurant owners.

1.4 Target Audience

- Software development team
- Test engineers
- Product managers
- Stakeholders(Customers, Waitstaff, Kitchen Team, Restaurant Owners)

2. General Overview

The system will be offered as both a web and mobile application, providing an intuitive interface that makes it easy for customers to make table reservations, place food orders, and leave feedback. Waitstaff will monitor customer arrivals, while the kitchen team will receive order details. Managers will oversee the system and analyze performance data.

3. User Roles and Permissions

User Type	Permission
Customer	Creating, canceling, and viewing table reservations; placing orders, viewing tables, making payments, tracking orders, and leaving feedback
Waitstaff	Viewing assigned tables, viewing orders and payments, requesting time off, and requesting approval for extra working hours.
Kitchen Team	Viewing customer preferences/allergies and order notes.
Admin	Managing users, system settings, tables, and menus; analyzing feedback and performance, and managing orders and payments

4. Functional Requirements (FR)

Account and Authentication Process

No	Functional Requirements
FR-01	Users must be able to log in using their email and password.
FR-02	Users must be able to create a new account.
FR-03	The system must use a separate Roles table to assign each user a specific role (waitstaff, chef, manager, etc.).

Reservation Process

No	Functional Requirements
FR-04	Customers must be able to view the status of tables and select a table. They should also be able to make a reservation
FR-05	The system must lock the selected table for 10 minutes to complete the reservation.
FR-06	If the reservation is not completed within 10 minutes, the system must automatically release the table.
FR-07	Customers must receive the table number and time information via SMS/email with the reservation confirmation
FR-08	Customers must be able to cancel their reservations, and waitstaff/managers must be notified via SMS or email.

FR-09	Reservation cancellations must be made at least 2 hours before the restaurant arrival time; otherwise, the prepayment will not be refunded.
--------------	---

Order Process

No	Functional Requirements
FR-10	Customers must be able to view the menu, which is divided into categories (food, drinks, desserts). The menu will also include details such as allergen information and ingredients.
FR-11	Customers must be able to place an order either before arriving at the restaurant or after they arrive.
FR-12	For orders placed before arriving at the restaurant, prepayment must be mandatory. For orders placed at the restaurant, payment can be made via cash/card during or after the order.
FR-13	Customers must be able to specify any allergies or preferences in their orders.
FR-14	The kitchen team must receive the order details, including allergies and preferences.
FR-15	Customers must be able to track the preparation process of their orders.

Waiter Management Process

No	Functional Requirements
FR-16	The customer must be able to call a waiter through the system. The notification will reach the waiter via the application.
FR-17	When a waiter is unavailable (e.g., on leave or not on shift), the system must detect this and the waiter will not receive notifications.
FR-18	Waiters must be able to log in to the system using their email and password.

Feedback and Evaluation Process

No	Functional Requirements
FR-19	Customers must be able to rate the service, food quality, and overall atmosphere of the restaurant, as well as leave comments.
FR-20	To leave feedback, it must be verified that the customer has made a reservation or placed an order.
FR-21	Managers must be able to view and analyze customer feedback.

Manager Management Process

No	Functional Requirements
FR-22	Managers must be able to add, update, and delete menu items and categories, as well as manage tables in the system.
FR-23	Managers must be able to track statistics such as daily/weekly reservation counts, order volume, cancellation rates, and customer satisfaction scores.
FR-24	The system must send notifications to waiters/kitchen staff for any order updates or cancellations
FR-25	Managers must be able to manage system settings, track reservation statistics (such as peak times, most preferred dishes, and customer behavior), and audit employee accounts.

5. Non-Functional Requirements(NFR)

No	Non-Functional Requirements
NFR-01	The system must support at least 500 active users per day for each restaurant location and must be scalable as the chain grows.
NFR-02	The system should be optimized to respond to reservation and order operations within a maximum of 5 seconds under ideal conditions.
NFR-03	User data must be protected using the AES-256 encryption standard.
NFR-04	The platform must be compatible with both mobile and desktop browsers.
NFR-05	The user interface must be designed according to WCAG 2.1 accessibility standards.
NFR-06	To prevent data loss, the system must perform daily backups.
NFR-07	Passwords in the system must be hashed and stored using the Argon2 algorithm.

6. Constraints

No	Constraints
C-01	Customers can reserve tables up to 7 days in advance.
C-02	Each table can only be reserved by one customer at a time.
C-03	Orders placed before arriving at the restaurant will not be processed by the kitchen unless prepayment is made.
C-04	The system must operate exclusively over HTTPS.

C-05	Reservation cancellations must be immediately notified to the waitstaff/managers.
-------------	---

7. Data Management

Database: PostgreSQL

Main Tables:

- **customer** (cid, name, surname, email, phone number, password)
- **roles** (id, role_name)
- **reservations** (id, customer_id, timestamp, state, note, order_state, call_waiter, table_id)
- **tables** (id, capacity, status)
- **menu** (id, name, price, category_id, allergenic_info)
- **category** (id, name)
- **order** (id, item_name, reservation_id, quantity)
- **payment** (id, payment_amount, reservation_id, payment_method, payment_status, payment_date)
- **reviews** (id,comment, reservation_id, food rate, staff rate, rest. rate)
- **notifications**(id, not_type, customer_id, content)
- **staff**(id, name, Surname, phone number, email, password)

8. Risks

Risk	Solution/ Mitigation Approach
System failure due to high user traffic	Load balancing will be implemented for scalability.
Data security breaches	AES-256 encryption and secure session management will be used.
Data loss	This risk will be mitigated through daily backups.
Reservation/order conflicts	Table locking and prepayment requirements will prevent conflicts.

9. Deliverables and Timeline

Deliverables	Timeline
Requirement Analysis	2 Weeks

UI/UX Design	3 Weeks
Database Design	2 Weeks
Backend and API Development	5 Weeks
Testing Process	3 Weeks
Deployment & Maintenance	Always