Software Requirements Specification

for

Reservations Management system for a Restaurant

Version 1.0

Prepared by

Safaa Nahas Haidar Rayya Mohamad Karakit

Khatoon Badria Hiba Altabbal

Mohamad Mostafa Youssef Kurdi Hussein Hamdan

<2025-01-12>

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the detailed functional and non-functional requirements for the development and implementation of the Restaurant Reservation System (RRS). This document serves as a foundational guide for the project team, including developers, project managers, system architects, and stakeholders, ensuring a shared understanding of the system's objectives, scope, and features. The goal is to deliver a robust, reliable, and scalable system for restaurant reservations.

1.2 Scope of this Document

This document focuses on capturing the software requirements for the Restaurant Reservation System, which facilitates the process of booking tables at restaurants for customers and managing reservations for restaurant administrators. While the initial version of the system excludes a front-end interface, the system incorporates email communication for notifications and confirmations. This document covers the system's functionality, performance expectations, and other relevant requirements but excludes hardware or physical infrastructure considerations.

1.3 Overview

The Restaurant Reservation System aims to streamline the reservation process for both customers and restaurant managers. The system will feature capabilities such as table management, customer notifications via email, reservation scheduling, and analytics for restaurant usage. The absence of a front-end interface in this phase is mitigated by providing API-driven interactions and email-based communication channels. The system will be developed with scalability and integration in mind, allowing for future expansion, including web and mobile interfaces.

1.4 Business Context

With the growing demand for digital solutions in the hospitality industry, the Restaurant Reservation System addresses the need for efficient, flexible, and user-friendly table reservation mechanisms. The system benefits customers by offering convenience in booking and ensures restaurants can optimize their seating arrangements and reduce no-shows through effective notifications. The RRS will establish itself as a reliable and adaptable solution, enhancing the overall dining experience for users and operational efficiency for restaurants.

2. Overall Description

2.1 Product Functions

The Restaurant Reservation System (RRS) provides a robust platform for customers to book, modify, and cancel table reservations while ensuring smooth communication with restaurant staff through email notifications. Administrators, managers, and waitstaff are equipped with tools to manage table availability, reservations, and customer interactions. The system integrates Stripe for basic payment needs and ensures scalable, reliable operations suitable for future expansions.

Key functionalities include:

Reservation management for customers, including booking, modifying, and canceling reservations.
Dynamic table status management for waitstaff, enabling real-time updates.
Automated email notifications for reservation confirmations, changes, and cancellations, as well as event announcements and feedback collection.
Daily reservation summary reports for admins and managers.
Emergency notifications for unforeseen closures or disruptions.

2.2 Similar System Information

While there are several existing restaurant reservation systems, the RRS distinguishes itself by emphasizing multi-role functionality and robust email-based communication. Unlike many systems that rely heavily on a front-end interface, RRS focuses on back-end scalability and API-driven interactions, allowing seamless future integrations with web and mobile platforms. The system prioritizes user satisfaction by automating reservation workflows and ensuring streamlined table management for staff.

2.3 User Characteristics

The RRS is designed to cater to a diverse set of users, including:

Customers: Customers can easily make, modify, and cancel reservations through an intuitive interface. Clear notifications and updates ensure they remain informed about their reservation status.
Restaurant Administrators: Admins have a comprehensive view of all operations, including reservations across departments, user management, and reporting tools.
Managers: Managers oversee reservations specific to their departments, handling confirmations, cancellations, and customer communications efficiently.
Waitstaff: Waitstaff use the system to manage table statuses, ensuring real-time updates and smooth service transitions during operations.

2.4 User Problem Statement

Restaurants often face challenges with manual reservation management, including double bookings, miscommunication, and inefficiencies in table utilization. Customers may experience frustration due to unclear reservation statuses or lack of timely updates. The RRS addresses these issues by automating workflows, providing clear and timely email notifications, and allowing real-time table status updates to reduce errors and enhance the dining experience.

2.5 User Objectives

The RRS aims to:

Provide customers with a simple and reliable way to make, modify, or cancel reservations.
Enable restaurant staff to efficiently manage table availability, reservations, and customer interactions.
Ensure timely and clear communication through automated emails.
Offer admins and managers detailed insights into reservation patterns and operational efficiency.
Facilitate event announcements and emergency updates to keep customers informed and engaged.

2.6 General Constraints

The RRS supports both email and Telegram as communication channels for notifications, ensuring timely updates and flexibility for users.
Integration with Stripe's free version limits advanced payment features.
A stable internet connection is required for system functionality, including notifications and API interactions.
The system is designed for a single restaurant, but it can be adapted for multiple restaurants with minimal adjustments.
Compliance with data privacy regulations (e.g., GDPR) ensures secure handling of customer information.

3. System Features

3.1 User Authentication and Authorization

a. Description

The system shall implement secure authentication and authorization for users. This includes:

User Registration: Users (customers, waitstaff, managers, admins) can register in the system.
User Login: Users log in using their credentials.
Password Reset: Users can request a password reset, and the system will send an email with a reset code.
Role-Based Access Control (RBAC): Different user roles (customers, waitstaff, managers, and admins) will have different levels of access and permissions within the system.

b. Criticality : Extreme

c. Technical Issues:

Use secure authentication protocols and token-based session management.
Ensure password recovery process is handled via email with secure reset code. d. Risks:
Data breaches or unauthorized access if security measures are not properly implemented.
Users forgetting passwords may cause frustration if the recovery process fails. e. Dependencies:
Secure backend authentication system and email service provider.

3.2 User Management

a. Description

The system shall provide administrators with tools to manage user accounts efficiently. This includes:

- **User Creation:** Administrators can create new user accounts by entering the username, email address, password, and role.
- **User Management:** Administrators can view, search, and filter a list of users by username, email, or status.
- **User Updates:** Administrators can update user details such as roles, permissions, or account status (e.g., active or inactive).
- **Account Deactivation/Deletion:** Administrators can deactivate or delete user accounts.
- **Password Reset:** Users can request password resets, which will be handled via an email verification process.
- **Profile Updates:** Users can update their personal information (e.g., name and email) through profile settings.

b. **Criticality:** High

c. Technical Issues:

- Ensure secure role-based access control (RBAC) for managing user permissions and roles.
- Implement secure email-based password recovery with verification processes.
- Use strong password encryption and token-based session management.

3.3 Reservation Management

a. Description:

The Reservation Management feature enables customers to reserve tables at the restaurant by specifying their requirements, such as date, time, and the number of seats. Customers can also input a specific table number if desired. The system checks availability based on the provided details and suggests alternatives if the specified table or matching options are unavailable. Reservations are finalized after admin approval.

b. Functional Overview:

- **Reservation Input:**
  Customers can input the following details for a reservation:
  - Start and end dates.
  - Desired time range.
  - Number of seats required.
  - Specific table number (optional).

- **Availability Check:**
  - If a specific table number is entered:
    - The system checks the availability of the specified table.
    - If unavailable, it suggests alternative tables based on the customer's requirements.
  - If no table number is entered, the system directly displays all available options.
- **Table Selection:**
  Customers can view available tables with detailed information such as:
  - Table number.
  - Seating capacity.

- **Order Submission:**
  - Reservations are stored as pending until admin approval.
  - Admins are notified of new reservation requests.
- **Admin Confirmation:**
  - Admins can confirm or reject reservations through a dashboard.
  - Customers are notified of the admin's decision.
- **Table Status Updates:**
  Waitstaff can update the reservation status at different stages, including:
  - "In Service"
  - "Completed"

- **Reservation Payment:**
  - Payments can be made securely via Stripe when the reservation is marked "In Service."
  - The reservation status is updated upon payment completion.

c. Criticality: High

d. Technical Issues:
- Implement a robust availability-checking algorithm to prevent double bookings.
- Ensure table locking during admin confirmation to avoid conflicts.
- Integrate Stripe for secure payment processing.

3.4 Reservation Rating Management

a. Description:
The Reservation Rating System enables customers to provide feedback on their dining experience through a star-based rating and optional text reviews. This feature helps the restaurant gauge customer satisfaction, enhance services, and recognize exceptional staff performance.

b. Functional Overview:
- **Submit a Rating:**
  - Customers can rate their reservation experience on a 1–5 star scale.
  - Optional text reviews (up to 500 characters) can be submitted with the rating.
  - Only customers with completed reservations are eligible to provide ratings.
- **Edit/Delete Ratings:**
  - Customers can edit their ratings and reviews within 48 hours of submission.
  - Customers can delete their ratings if they wish.
- **View Ratings:**
  - Admins can access all ratings and reviews for reservations.
  - The system provides aggregated statistics such as average rating and total number of ratings.
- **Notifications:**
  - Customers receive a notification when their reservation is marked "Completed," prompting them to provide a rating.
- **Integration with Reservations:**
  - Ratings are linked to specific reservations for accurate tracking.
  - The system prevents ratings for reservations that were canceled or marked as "no-show."

3.5 Notifications

a. Description:
The Notification System provides email and Telegram notifications to customers, admins, managers, and waitstaff based on specific triggers and events. The system ensures timely communication for actions such as reservation updates, event changes, and emergencies.

b. Functional Overview:

i. Customer Notifications:
- **Reservation Confirmation:**
  - Email: Sent upon reservation acceptance by the manager.
  - Telegram: Notification sent on confirmation.
- **Reservation Cancellation:**
  - Email: Sent if the reservation is canceled.

    - Telegram: Notification sent on cancellation.
- **Reservation Rejection:**
    - Email: Sent when the reservation is rejected by the manager.
    - Telegram: Notification sent on rejection.
- **Password Reset Request:**
    - Email: Includes a password reset code when requested by the customer.
    - Telegram: Not applicable.

ii. Admin and Manager Notifications:

- **Daily Reservation Reports:**
    - Email: Daily email with reservation details for admins and managers.
    - Telegram: Not applicable.
- **New Event Notifications:**
    - Email: Customers receive email notifications for new or updated events.
    - Telegram: Not applicable.
- **Emergency Notifications:**
    - Email: Sent to customers during emergencies affecting reservations or events.
    - Telegram: Not applicable.
- **Event Rating Request:**
    - Email: Customers are prompted to rate their experience post-reservation.
    - Telegram: Not applicable.
- **Reservation Confirmation/Changes (Manager-specific):**
    - Email: Notifications sent for reservations requiring confirmation or rejection by managers.

iii. Waitstaff Notifications:

- **Reservation Status Updates:**
    - Email: Not applicable.
    - Telegram: Not applicable. Waitstaff update statuses directly in the system.

c. Criticality: High

d. Technical Issues:

- Ensure proper configuration to match notifications with corresponding events.
- Include accurate reservation and customer data in notifications.
- Role-based notifications must function reliably to avoid irrelevant notifications.

e. Risks:

- Notification delays or failures could disrupt operations or lead to customer dissatisfaction.
- Emails marked as spam may reduce their effectiveness.
- Misconfigured Telegram notifications may result in non-delivery or incorrect recipients.

f. Dependencies:

- **Email Services:** SMTP or third-party providers (e.g., SendGrid).
- **Telegram Bot API:** For delivering Telegram notifications.
- **Backend System:** Manages notification triggers, roles, and content delivery.

3.6 Event Management

a. Description:

The Event Management feature enables the restaurant to organize and communicate events effectively. This includes:

- **Event Creation and Updates:**
    - Admins can create or modify events.
    - Notifications with event details are sent to customers and relevant staff upon creation or updates.

- **Event Viewing:**
  - Customers can view event details, such as date, time, location, and type.
  - Customers cannot respond to event notifications.
- **Event Registration:**
  - No ticketing or registration functionality is included in this version.

b. Criticality: Medium

c. Technical Issues:

- Ensure that event details (e.g., date, time, location) are accurate and displayed properly in both the system and notifications.
- Notifications must include comprehensive event details for clarity.

d. Risks:

- Incorrect or incomplete event details could lead to customer or staff confusion and dissatisfaction.

e. Dependencies:

- **Backend System:** To store and manage event data.
- **Notification System:** To deliver updates about events to customers and staff.

3.7 Payment Integration

a. Description:

The system shall allow customers to complete payments for their reservations:

Viewing Reservation Status: Once the reservation is marked as "In Service," the customer will see a "Pay" button.

Entering Payment Amount: The customer will input the cost of the reservation as communicated by the waiter.

Payment Processing: After entering payment details, the customer can complete the payment securely via the Stripe API.

Payment Status Update: The payment status will be updated in the system once the payment is processed.

b. Criticality - High

c. Technical Issues:

Ensure the reservation cost is properly displayed to the customer.

Integrate with Stripe API for secure payment processing.

Update payment status accurately in the system after payment completion.

d. Risks:

Payment failures could lead to delays or errors in processing reservations.

e. Dependencies:

Stripe API for payment processing.

Backend system for managing reservation and payment status updates.

3.8 Emergency Management

a. Description:

The Emergency Management feature enables the restaurant to handle emergency situations efficiently by automating reservation cancellations and notifying affected customers.

b. Functional Requirements:

- **FR1: Emergency Declaration and Reservation Cancellation**
  - The system shall allow administrators to declare an emergency.

    - Upon declaration, the system will automatically cancel all existing reservations.
- **FR2: Customer Notifications**
    - The system shall send notifications to customers to inform them about the emergency and the cancellation of their reservations.
    - Notifications will include details about the emergency (if applicable) and the impact on their reservations.

c. Criticality: High

d. Technical Issues:

- Ensure timely and accurate cancellation of all relevant reservations.
- Ensure that notifications provide clear and precise information to avoid customer confusion.

e. Risks:

- Delays in notifying customers might lead to dissatisfaction.
- Miscommunication in notifications could cause unnecessary panic or confusion.

f. Dependencies:

- **Notification System:** To deliver emergency updates to customers.
- **Reservation System:** To manage cancellations and updates automatically.

3.9 Table Management

a. Description:

The Table Management feature enables the restaurant to efficiently organize and manage tables by categorizing them into different sections, ensuring smooth reservation handling and operational flexibility.

b. Functional Requirements:

- **FR1: Add New Tables**
    - The system shall allow administrators to add new tables.
    - Each table can be assigned to a specific restaurant section (e.g., indoor, outdoor, VIP).
- **FR2: Update Table Details**
    - The system shall allow administrators to update table information, including table number, seating capacity, and section assignment.
- **FR3: Delete Tables**
    - The system shall allow administrators to delete tables that are no longer in use.
    - The system shall prevent the deletion of tables that are currently reserved or in use.

c. Criticality: Medium

d. Technical Issues:

- Ensure proper validation to avoid errors when adding, updating, or deleting tables.
- Maintain consistency by linking tables to their reservations and ensuring no conflicts arise when deleting tables.

e. Risks:

- Accidental deletion or mismanagement of table data could disrupt reservation operations.
- Misconfiguration of table sections may lead to confusion during reservations.

f. Dependencies:

- **Reservation System:** To check and manage table availability and status.
- **Backend System:** For storing and retrieving table details.

3.10 Restaurant Information Management

a. Description:
The Restaurant Information Management feature allows the administrator to maintain and update the restaurant's general details, ensuring accurate and up-to-date information is available to customers.
b. Functional Requirements:
- **FR1: Update General Information**
  - The system shall allow the administrator to update key information, such as:
    - Working hours (e.g., opening and closing times).
    - Restaurant location (e.g., address and map coordinates).
  - Updated information shall be reflected across all customer-facing platforms (e.g., website, notifications).

c. Criticality: Low
d. Technical Issues:
- Ensure seamless synchronization of updated information across all systems and platforms.
- Implement proper validation to prevent entry of incorrect or incomplete data.

e. Risks:
- Outdated or incorrect information could lead to customer inconvenience and dissatisfaction.
- System errors might result in failure to reflect updates in real time.

f. Dependencies:
- **Backend System:** To store and manage restaurant information.
- **Frontend Interfaces:** To display updated details to customers.

3.11 Department Management

a. Description:
The Department Management feature allows the administrator to organize the restaurant into sections (e.g., family area, smoking area) to streamline operations and improve the reservation experience.
b. Functional Requirements:
- **FR1: Create New Sections**
  - The system shall allow the administrator to create new sections for the restaurant, such as "Family Area," "Smoking Area," or "Outdoor Seating."
- **FR2: Edit/Delete Sections**
  - The system shall allow the administrator to edit or delete existing sections.
  - The system shall ensure that sections linked to active reservations cannot be deleted.

c. Criticality: Medium
d. Technical Issues:
Proper linkage between sections and tables to prevent inconsistencies during updates or deletions.
e. Risks:
Accidental deletion of sections could disrupt reservation organization.
f. Dependencies:
- **Table Management System**: To associate tables with specific sections.

3.12 Dish Management

a. Description:
The Dish Management feature enables the administrator to manage the restaurant's menu by adding, updating, and removing dishes to ensure the menu is current and accurate.
b. Functional Requirements:

- **FR1: Add New Dishes**
  - The system shall allow the administrator to add new dishes with details such as:
    - Dish Name
    - Price
    - Category (e.g., Appetizers, Main Course, Desserts)
- **FR2: Update Dish Details**
  - The system shall allow the administrator to update existing dish details, such as price, name, or category.
- **FR3: Delete Dishes**
  - The system shall allow the administrator to delete dishes.
  - The system shall ensure that dishes currently in customer orders cannot be deleted.

c. Criticality: High
d. Technical Issues:

- Proper validation to prevent duplicate dish entries.

e. Risks:

- Errors in dish details could confuse customers or impact sales.

f. Dependencies:

- **Category Management System**: To organize dishes effectively.

3.13 Food Category Management

a. Description:
The Category Management feature allows the administrator to organize dishes into categories, improving menu navigation for customers.
b. Functional Requirements:

- **FR1: Add New Categories**
  - The system shall allow the administrator to add new categories for dishes (e.g., Beverages, Soups, Salads).
- **FR2: Update/Delete Categories**
  - The system shall allow the administrator to update or delete existing categories.
  - The system shall prevent deletion of categories linked to active dishes.

c. Criticality: Medium
d. Technical Issues:

- Ensure seamless updates to categories across the system.

e. Risks:

- Deleting a category linked to dishes could cause inconsistencies in the menu.

f. Dependencies:

- **Dish Management System**: To link dishes with their respective categories.

3.14 Favorites Management

a. Description:
The Favorites Management feature allows customers to save food category or tables to a favorites list for quicker and more convenient access during future reservations.
b. Functional Requirements:

- **FR1: Add Items to Favorites**
  - The system shall allow customers to add food category or tables to their favorites list.
- **FR2: Remove Items from Favorites**
  - The system shall allow customers to remove items from their favorites list.

c. Criticality: Low
d. Technical Issues:

- Maintain a user-specific favorites list to ensure personalized experiences.
  e. Risks:
- Failure to update favorites might frustrate customers.
  f. Dependencies:
- **User Profile System**: To store and retrieve user-specific favorites.
- **Food Category and Table Management Systems**: To fetch details of items added to the favorites list.

3.15 Notification Logs Management

a. Description:
The Notification Log Management feature allows the system to track and store records of all notifications sent to customers via email and Telegram. This is essential for auditing, troubleshooting, and ensuring transparency in communication across different notification methods.
b. Functional Requirements:

FR1: Log Notification Details
The system shall log all notifications sent to customers, capturing the following details:

- User ID (to identify the recipient).
- Notification method (e.g., email, Telegram).
- Status of the notification (e.g., Sent, Failed, Pending).
- Reason for sending the notification (e.g., Reservation ID, Event Trigger).
- Description of the notification content.
- Timestamps for when the notification was created and updated.
- Optional: Soft delete capability to handle temporary or archived logs.

c. Criticality: Medium
d. Technical Issues:

- Ensure secure storage of notification logs to protect customer data.
- Enable efficient querying of logs based on the notification method, recipient, status, or time periods.
- Handle large volumes of logs without causing performance issues or system degradation.

e. Risks:

- Log storage could grow excessively large, increasing database or server costs.
- Mismanagement of logs may expose sensitive information if not handled securely.
- Failure to monitor logs across multiple notification methods may reduce operational efficiency.

3.16 Daily Reservation Reports

- **a. Description:**

The Daily Reservation Reports feature provides the administrator with a summary of reservation activity for the day, allowing them to monitor performance, identify trends, and address operational needs efficiently.

- **b. Functional Requirements:**
  - **FR1: Generate Daily Summary Reports**
    - The system shall allow the administrator to generate a daily summary report that includes the total number of reservations, canceled reservations, and pending reservations.
  - **FR2: View Detailed Reservation Logs**
    - The system shall provide detailed logs of all reservations for the day, including:
      - Reservation ID.
      - Customer name and contact details.
      - Reservation time and duration.
      - Table number and section.
      - Status (pending, approved, canceled).
  - **FR3: Filter and Export Reports**
    - The system shall allow filtering reports by reservation status, sections, or time periods.
    - The system shall enable the administrator to export reports in formats such as PDF or Excel.
- **c. Criticality:**
  - High — Ensures efficient monitoring and operational management.
- **d. Technical Issues:**
  - Ensuring real-time data accuracy in reports.
  - Efficient data retrieval for large datasets during peak days.
- **e. Risks:**
  - Incomplete or inaccurate data due to system errors.
  - Unauthorized access to reservation reports.

4. Nonfunctional Requirements

4.1 Security Requirements

- I implemented robust security measures to protect sensitive data and prevent unauthorized access.
  - **Applied Implementations:**
    - Used secure login mechanisms with password encryption using the bcrypt algorithm.
    - Added protection against common vulnerabilities such as SQL Injection, XSS, and CSRF.
    - Applied a Role-Based Access Control (RBAC) system to restrict sensitive operations based on user roles.
    - Created audit logs for tracking sensitive actions.

4.2 Performance Requirements

- I ensured optimized performance to achieve fast and reliable response times even under high loads.
  - **Applied Implementations:**
    - Used caching mechanisms to reduce database load.
    - Optimized queries to minimize response times for both read and write operations.
    - Implemented lazy loading for heavy data to improve efficiency.

4.3 Maintainability Requirements

- I ensured the system is easy to maintain and update.
  - **Applied Implementations:**
    - Followed a modular design approach to isolate components, simplifying updates.
    - Adhered to coding standards like PSR-12.
    - Documented code, APIs, and system architecture comprehensively.
    - Used version control (Git) to manage changes to the codebase.

4.4 Compatibility Requirement

- I ensured seamless integration of the system with external tools and services.
  - **Applied Implementations:**
    - Used standard communication protocols like RESTful APIs and JSON.
    - Integrated with external services such as Stripe, Telegram API, and email providers.
    - Created clear API documentation to facilitate future extensions or integrations.

4.5 Monitoring and Logging

- I set up a robust monitoring and logging system to detect and resolve issues quickly.
  - **Applied Implementations:**
    - Monitored critical operations and services .
    - Configured automated alerts to detect and address any issues or errors.

5. Operational Scenarios

5.1 User Registration and Login

- **Scenario:**
  A new user (customer) can create an account by providing their email address, password, name, and other required information. Once registered, they can log in using their credentials.
    - Customers can register or log in .
    - Administrators, waiters, and managers have accounts created by the system admin.

5.2 Dish Browsing and Viewing

- **Scenario:**
  Upon logging in, customers can explore the home page to view available dishes and categories (e.g., appetizers, main courses, desserts). This feature is designed to enhance the customer experience.
    - Customers cannot order dishes through the system; this is only for browsing.

5.3 Table Reservation

- **Scenario:**
  A customer reserves a table by entering the following details:
  - **Reservation Date & Time:** Start and end times of the reservation.
  - **Number of Seats:** The number of people attending.
  - **Table Number (Optional):** Customers can input a table number if they wish to reserve a specific table.
    - If the specified table is available, the system reserves it.
    - If unavailable, the system suggests alternative tables.
    - If no table number is entered, the system displays all available tables immediately.
  - **Notification Method:** The customer chooses between Telegram or email notifications.

The system validates the reservation request and searches for available tables that meet the criteria.
The customer can:

- View available tables.
- Select a table and confirm the reservation request.
- Receive a notification once the admin or manager approves/rejects the request.
- Cancel approved reservations if needed.

Pending reservations are automatically canceled after a defined period of inaction.

5.4 Reservation Approval or Rejection

- **Scenario:**
  Admins or managers review incoming reservation requests.
  - **Approval**: The customer receives a notification confirming the reservation.
  - **Rejection**: The customer receives a notification explaining the rejection.

5.5 Reservation Check-In and Service Start

- **Scenario:**
  Upon the customer's arrival at the restaurant:
  - The waiter verifies the reservation and starts the service.
  - The waiter updates the system to indicate the reservation is active.

5.6 Reservation Completion and Payment

- **Scenario:**
  After the customer completes their dining experience:
  - The waiter ends the reservation, marking the service as complete.
  - **Payment Options:**
    - Customers can pay using one of the following methods:
      - **Cash:** Handled directly by the waiter.
      - **Electronic Payment:** Customers can complete the payment using their application, which is synchronized with the system.

5.7 Post-Reservation Review and Favorites

- **Scenario:**
  After the reservation is marked as completed, customers receive a notification prompting them to leave a review and rating for their experience.
  - Customers can evaluate their reservation and service quality.
  - Customers can add tables or dishes to their favorites list for future reference.

5.8 Notifications

- **Scenario:**
  Customers receive notifications for key events based on their selected method (Telegram or email):
  - Reservation approval or rejection.
  - Payment confirmation.
  - Review requests.
  - Pending reservation cancellations.

5.9 Waiter and Manager Roles

- **Scenario:**
  - **Waiters:**
    - Verify and confirm reservations.
    - Start and end services, ensuring smooth operations.
    - Facilitate payments, whether in cash or by guiding customers to complete electronic payments through their application.
  - **Managers:**
    - Approve or reject reservation requests based on availability and policy.
    - Oversee sections and assign tables to optimize service efficiency.

5.10 Reports and Analytics

- **Scenario:**
  Admins and managers can access reports for:
  - Reservations (e.g. days).
  - Table utilization rates.
  - Customer reviews and feedback summaries.

6. System Behavior

6.1 Sequence diagrams: