Software Requirements Specification

for

TABLEDIN: SMART RESERVATIONS AND SEAT COORDINATION SYSTEM

Version 1.0 approved

Prepared by

Nares, Jerry Reivrick C.

Reyes, John Edwin C.

Verzosa, Zrone Jinrx Jbryl F.

Far Eastern University Institute of Technology

September 27, 2025

Table of Contents

1. Introduction **1**
1.1. Purpose 1
1.2. Document Conventions 1
1.3. Intended Audience and Reading Suggestions 1
1.4. Product Scope 1
1.5. References 2
2. Overall Description 2
2.1. Product Perspective 2
2.2. Product Functions 2
2.3. User Classes and Characteristics 3
2.4. Operating Environment 3
2.5. Design and Implementation Constraints 3
2.6. User Documentation 4
2.7. Assumptions and Dependencies 4
3. External Interface Requirements 4
3.1. User Interfaces 4
3.2. Hardware Interfaces 4
3.3. Software Interfaces 5
3.4. Communications Interfaces 5
4. System Features 5
4.1 Customer Account Management 5
4.2 Reservation and Scheduling 6
4.3 Seating Management (Staff/Admin) 6
4.4 Queue Management (Walk-ins) 7
4.5 Reporting 7
5. Other Nonfunctional Requirements **8**
5.1. Performance Requirements 8
5.2. Safety Requirements 8
5.3. Security Requirements 8
5.4. Software Quality Attributes 8
5.5. Business Rules 9
6. Other Requirements **9**
6.1. Data Privacy Compliance 9
6.2. Database Backup and Recovery 9
6.3. Email Notification Provider 9
6.4. System Availability 9
6.5. User Access and Roles 9
6.6. Browser and Platform Support 10
6.7. Audit and Logging 10

6.8. Reporting Standards 10
Appendix A: Glossary **10**
Appendix B: Analysis Models **10**
Appendix C: To Be Determined List **11**

Revision History

Name	Date	Reason For Changes	Version

1.Introduction

1.1.Purpose

This document outlines the requirements for the development of **TabledIn: Smart Reservation and Seat Coordination System,** version 1.0. The system is designed to help small to medium-sized restaurants in the Philippines manage customer reservations and seating arrangements more efficiently. It provides a platform where customers can book tables online, and restaurant staff can monitor and organize seating in real-time. This SRS covers the entire system, including both the customer-facing interface and the admin dashboard used by the restaurant staff. The goal is to improve the dining experience operations to be smoother and more organized.

1.2.Document Conventions

To make this document easy to read and understand, the following conventions are used:

- **Bold text** is used for buttons and interface labels (e.g., **Book Now, Cancel Reservations)**.
- *Italic text* is used for technical terms and system components (e.g., *database, API, user session*).
- Dates are written in the format **DD/MM/YYYY.**
- Currency is shown in **Philippine Peso (₱)**.
- Each requirement is numbered and categorized by module (e.g., 1.1 for Customer Profile Management Module, 1.2 for Appointment Scheduling Module)
- Priorities are indicated for each requirement, and are not assumed to be inherited.

1.3.Intended Audience and Reading Suggestions

This document is intended for:

- **Developers and system designers** who will build and implement the system.
- **Project managers and academic advisers** who will oversee the development and ensure it meets its goals.
- **Restaurant owners and staff** who will use the system in their daily operations.
- **Testers and QA personnel** who will validate the system's functionality.

Readers who are new to the project should start with the Product Scope to understand what the system is about. Developers and testers should focus on the Functional Requirements and System Features, while restaurant staff and end-users may be more interested in the User Interface Design and Use Case Scenarios.

1.4.Product Scope

TabledIn is a smart, web-based reservation and seat coordination system built for restaurants in the Philippines. It allows customers to book a table online, view available time slots, and receive confirmation messages. On the restaurant side, staff can manage seating arrangements, monitor table occupancy, and adjust bookings in real-time. The system is designed to be accessible on desktops.

1.5.References

Sykimte, C. (2023). Revolutionizing the Dining Experience: Exploring Digital Innovations in Food Ordering for Enhanced Customer Satisfaction. Surigao del Norte State University.
Briones, R. M., Galang, I. M. R., & Latigar, J. S. (2023). Transforming Philippine Agri-Food Systems with Digital Technology. Philippine Institute for Development Studies.
Bahillo, E. L. E., & Nasser, T. T. (2022). Documented Experiences among Food Service Businesses towards Future-Proofing the Industry. Philippine EJournals.
Borbon, N. M. D. (2025). Kiosk Revolution: The Transformation of Fast Food in the Philippines. National University Philippines.
Mordor Intelligence. (2025). Philippines Foodservice Market Report.
Clemente, R., et al. (2024). Importance-Performance Analysis of a Restaurant Mobile Booking App in the Philippines.
National Tax Research Center. (2023). Profile of the Philippine Food Service Industry.
Greatclicks. (2024). Digital Solutions for Restaurants in the Philippines.
Parameswaran, A. (2025). Analyzing the Impact of Reservation Systems on Restaurant Operations.
Botti, L., et al. (2022). Smart Service Systems in Restaurant Management.

2.Overall Description

2.1.Product Perspective

TabledIn is a brand-new web-based system created to help small and medium-sized restaurants in the Philippines manage reservations and seating more smoothly. It's not a follow-up to any existing software; it's built from the ground up to solve common challenges in restaurant operations, especially around table bookings and real-time seat coordination.

The system has two main parts:

- A customer-facing portal where people can book tables online and get confirmation messages.
- An admin dashboard for restaurant staff to view, manage, and adjust seating arrangements as needed.

While TabledIn is designed to work independently, it can be expanded in the future to connect with other tools like POS systems or customer loyalty programs. It may also use third-party services (like SMS or email APIs) to send notifications to users.

2.2.Product Functions

- For Customers:
    - Browse available tables and time slots
    - Make, change, or cancel reservations
    - Receive booking confirmations and reminders

- Restaurant Staff:
    - Monitor which tables are occupied or available
    - Organize seating and adjust bookings in real-time
    - View customer details and reservation information
    - Generate reports for daily operations

- Behind the Scenes:
    - Store customer profiles and reservation data
    - Handle secure logins and user sessions
    - Send notifications via email

2.3.User Classes and Characteristics

TabledIn will be used by different types of users, each with their own needs and skill levels:

- Customers - People who book tables online. They don't need technical skills and will use the system occasionally.
- Restaurant Staff/Admin - Hosts, servers, or managers who use the dashboard daily to manage seating and reservations. They'll need basic computer skills.
- System Developers/Admins - Developers or IT personnel who maintain the system. They'll have advanced technical knowledge.
- Testers and QA - People who test the system during development to make sure everything works properly

2.4.Operating Environment

TabledIn will run on desktop computers with internet access. It's designed to work on:

- Operating Systems - Windows, macOS, or Linux
- Web Browser: Chrome, Firefox, Safari, and Edge
- Database: MySQL or PostgreSQL
- Optional Services: Email APIs for notifications

2.5.Design and Implementation Constraints

A few things will shape how TabledIn is built:

- It must follow Philippine data privacy laws (like the Data Privacy Act of 2012)
- Open-source tools are preferred to keep costs low
- The system should support real-time updates for seating coordination
- The interface should be responsive and optimized for desktop use
- Technologies used may include HTML/CSS, JavaScript, Laravel Framework, and SQL
- Security features like HTTPS, user authentication, and role-based access are required.

2.6.User Documentation

To help users get started and use the system effectively, the following materials will be provided:

- A user manual for customers and restaurant staff
- An admin guide for system setup and maintenance
- Online help is built into the dashboard

2.7.Assumptions and Dependencies

Here are a few things we're assuming as we build TabledIn:

- Restaurants will have reliable internet and basic desktop hardware
- The customer will use modern web browsers
- The system may rely on third-party APIs for sending notifications
- Future versions might connect with POS system or mobile apps
- Development will use open-source libraries and frameworks
- Success depends on the restaurant staff being trained and willing to use the system

3.External Interface Requirements

3.1.User Interfaces

TabledIn will have two main user interfaces: one for customers and one for restaurant staff. Both are designed to be simple, clean, and easy to use, even for people who aren't tech-savvy.

- Customer Interface: This is the part of the system that customers will see when they visit the restaurant's reservation page. It will let them choose a date and time, pick a table (if available), and confirm their booking. Buttons like Book Now, Edit Reservations, and Cancel will be clearly labeled. After each action, the system will show a confirmation or a friendly error message if something goes wrong (e.g., "That time slot is already full. Please choose another.").
- Admin Dashboard: This is for restaurant staff. It will show a live view of all reservations and table statuses. Staff can drag and drop bookings, mark tables as occupied or available, and make quick changes to the schedule. The layout will be consistent across all screens, with standard buttons like Save, Refresh, and Information always visible. The design will follow a simple, modern style guide to keep things intuitive.

Both interfaces are built for desktop use and will adjust to different screen sizes. A separate document will include mockups and more detailed US designs.

3.2.Hardware Interfaces

Since TabledIn is a web-based system, it doesn't need any special hardware. It's designed to work on the devices restaurants already have.

- Devices Supported: Desktop computers and laptops
- Input: Standard keyboard and mouse
- Output: Regular monitors or screens
- Internet: A stable internet connection is required for real-time updates

There's no need for barcode scanners, tablets, or other specialized equipment, though those could be considered for future versions.

3.3.Software Interfaces

TabledIn will connect with a few key software components to function properly:

- Database: The system will use a relational database like MySQL or PostgreSQL to store all the important data, customer info, reservations, table layouts, and so on.
- Web Technologies: The frontend will be built using HTML, CSS, and JavaScript, while the backend will use PHP.
- Operating Systems: It will run smoothly on Windows, macOS, and Linux.
- Third-party APIs: For sending reservation confirmations and reminders, the system may use services like Mailtrap.

3.4.Communications Interfaces

TabledIn will rely on standard internet communication protocols to work:

- HTTPS: All data between the user and the server will be encrypted using HTTPS.
- Emails: The system will send out booking confirmations and reminders using email.
- Browser Support: It will work on all major browsers, including Chrome, Firefox, Safari, and Edge.

Security is a top priority. All user data will be protected with encryption, and only authorized users will be able to access the admin dashboard. No special communication tools like VPN are need for this version.

4.System Features

4.1 Customer Account Management

4.1.1 Description and Priority

This feature allows customers to register, log in, and manage their profiles. It ensures secure access to the system.
Priority: High

4.1.2 Stimulus/Response Sequences

- **Stimulus:** User clicks **Sign Up** → enters required details.
- **Response:** System validates inputs → creates account → sends confirmation.
- **Stimulus:** User logs in with valid credentials.
- **Response:** System verifies credentials → grants access to dashboard.

4.1.3 Functional Requirements

- **REQ-1:** The system shall allow customers to create an account with unique credentials
- **REQ-2:** The system shall validate login credentials and grant access if valid
- **REQ-3:** The system shall allow customers to update profile details and change passwords
- **REQ-4:** The system shall reject duplicate or invalid account details (e.g., existing email, weak password).

4.2 Reservation and Scheduling

4.2.1 Description and Priority

Customers can book tables by selecting date, time, number of guests, and seating preferences.
Priority: High

4.2.2 Stimulus/Response Sequences

- **Stimulus:** User selects **Reserve Table**, chooses date, time, number of guests, and seat.
- **Response:** System checks availability → confirms reservation → sends email confirmation.
- **Stimulus:** User edits or cancels booking.
- **Response:** System updates records and sends a notification.

4.2.3 Functional Requirements

- **REQ-5:** The system shall allow customers to book tables with date, time, guests, and seating selection
- **REQ-6:** The system shall prevent double-booking of the same table and time slot
- **REQ-7:** The system shall allow customers to edit or cancel reservations
- **REQ-8:** The system shall send notifications (confirmation, updates, cancellations, reminders) via email/SMS

4.3 Seating Management (Staff/Admin)

4.3.1 Description and Priority

Staff can view, assign, or reassign seating arrangements using a real-time map of the dining area.
Priority: High

4.3.2 Stimulus/Response Sequences

- **Stimulus:** Staff logs into the admin dashboard.
- **Response:** System displays seating layout and reservation statuses.
- **Stimulus:** Staff reassigns a table.
- **Response:** System updates availability and notifies customers if affected.

4.3.3 Functional Requirements

- **REQ-9:** The system shall allow staff to view real-time seating arrangements
- **REQ-10:** The system shall allow staff to assign and reassign tables
- **REQ-11:** The system shall update seating availability in real-time

4.4 Queue Management (Walk-ins)

4.4.1 Description and Priority

Manages queues for walk-in customers by assigning token numbers and showing wait times.
Priority: Medium

4.4.2 Stimulus/Response Sequences

- **Stimulus:** Walk-in customer requests seating.
- **Response:** System issues queue token, displays estimated wait time.
- **Stimulus:** Table becomes available.
- **Response:** System updates queue, notifies next customer.

4.4.3 Functional Requirements

- **REQ-12:** The system shall provide a queue management system for walk-in customers
- **REQ-13:** The system shall display the current token being served and the estimated wait time

4.5 Reporting

4.5.1 Description and Priority

Generates daily, weekly, and monthly reports for reservation trends and peak hours.
Priority: Medium

4.5.2 Stimulus/Response Sequences

- **Stimulus:** Staff selects **Generate Report**.
- **Response:** System retrieves data and generates a formatted report (PDF/Excel).

4.5.3 Functional Requirements

- **REQ-14:** The system shall generate daily, weekly, and monthly reservation reports
- **REQ-15:** The system shall allow admins to export reports in a standard format (e.g., PDF/Excel).
- **REQ-16:** The system shall log report generation activities for auditing.

5.Other Nonfunctional Requirements

5.1.Performance Requirements

TabledIn should be fast and responsive, especially during peak hours when lots of customers are booking tables and staff are managing reservations. Ideally, pages should load in under 5 seconds, and actions like booking or updating a reservation should happen almost instantly.

The admin dashboard, in particular, needs to be updated in real time so staff can see changes as they happen, like when a table becomes available or a customer cancels. This helps avoid confusion and keeps everything running smoothly during busy shifts.

5.2.Safety Requirements

While the system doesn't deal with physical safety, it does handle personal data, so protecting that information is a priority. We'll make sure there are regular backups, secure logins, and clear access controls so only authorized staff can view or change sensitive data.

We're also following the Philippine Data Privacy Act of 2012, which means we'll be careful about how customer information is stored and used. If something goes wrong, like a failed booking or system eros, the system should let the user know right away and log the issue for review.

5.3.Security Requirements

Security is built into every part of TabledIn. All data sent between users and the server will be encrypted using HTTPS. Users will log in with secure credentials, and staff will have different access levels depending on their role.

Passwords will be stored safely using hashing, and sessions will automatically log out after a period of inactivity. We'll also protect the system from common threats like SQL injection and cross-site scripting (XSS), and follow best practices for web security.

5.4.Software Quality Attributes

Here are the qualities we're aiming for with TabledIn:

Easy to use: The system should be simple and intuitive for both customers and staff.
Reliable: It should work consistently without crashing or losing data.
Scalable: As restaurants grow, the system should be able to handle more users, tables, and features.
Maintainable: Developers should be able to update or fix things without breaking the system.
Portable: It should run smoothly on different operating systems and browsers.
Secure: Protecting user data is a must.
We're prioritizing usability and reliability, since the system will be used daily by people who may not be tech experts.

5.5.Business Rules

TabledIn will follow a few basic rules to keep things organized:

Only registered users can make or change reservations.
Staff can cancel or adjust bookings, but those changes should be logged.
Each table can only be booked once per time slot to avoid double bookings.
Admins can manage seating layouts, while regular staff can only view and update reservations.
Confirmation messages (via email or SMS) will be sent automatically after a booking is made or changed.
These rules help make sure the system stays fair, efficient, and easy to manage.

6.Other Requirements

6.1. Data Privacy Compliance

- The system must comply with the Philippine Data Privacy Act of 2012.
- All personal data collected (customer names, contact details, and reservation history) shall only be used for reservation and reporting purposes.

6.2. Database Backup and Recovery

- The system must support automated daily backups of reservation and seating data.
- Backup files must be stored securely with restricted access.
- Recovery procedures should ensure restoration within 24 hours in case of data loss.

6.3.Email Notification Provider

- Gmail SMTP shall be used as the official email provider for sending booking confirmations, updates, and reminders.
- The system must support configuration for Gmail's authentication and security protocols.

6.4.System Availability

- The system should be available 24/7 to allow customers to make reservations at any time.
- Planned maintenance windows must be communicated to restaurant staff at least 24 hours in advance.

6.5.User Access and Roles

The system must clearly define role-based access permissions:

- **Admin:** Full access, including seating layout customization, report generation, and user management.

- **Staff:** Limited access to reservation handling, seating management, and queue updates.
- **Customer:** Reservation creation, modification, and cancellation only.

6.6.Browser and Platform Support

- The system must support the latest stable versions of Chrome, Firefox, Safari, and Microsoft Edge.

6.7.Audit and Logging

- All reservation activities, staff changes, and system errors must be logged for auditing purposes.

6.8.Reporting Standards

- Reports should be exportable in at least two formats: PDF and Excel.
- Each generated report must include the restaurant's name, date range, and timestamp.

Appendix A: Glossary

- Admin – User role with the highest level of access, responsible for managing reservations, staff accounts, reports, and system settings.
- Customer – End-user who creates, modifies, or cancels reservations using the online platform.
- Staff – Restaurant personnel who manage reservations and seating through the admin dashboard.
- Queue Management – The process of handling walk-in customers using token numbers and estimated wait times.
- Seating Layout – The digital representation of the restaurant's table arrangement for monitoring availability.
- SMTP (Simple Mail Transfer Protocol) – Protocol used to send emails through Gmail for booking confirmations and notifications.
- Reservation – A booking made by a customer for a specific table, date, and time.

Appendix B: Analysis Models

- Use Case Diagrams – To illustrate interactions between customers, staff, and admin with the system.
- Data Flow Diagrams (DFD) – To represent the flow of information, including reservation creation, confirmation, and seating updates.
- Entity-Relationship Diagram (ERD) – To show database structure, including customer, reservation, and table entities.
- Context Diagram – To describe system boundaries and external entities such as customers and email providers.

Appendix C: To Be Determined List

TBD-1: Final selection of database system.
TBD-2: Configuration details for Gmail SMTP as the official email notification provider.
TBD-3: Final design of admin seating layout (static grid vs drag-and-drop).
TBD-4: Reporting format confirmation (Excel, PDF, or both).
TBD-5: Definition of staff access permissions.