# **Software Requirements Specification (SRS) Document**

## **Team: Direct customers (restaurant owners), salesperson of restaurant.**

## **1. Introduction**

### **1.1 Purpose of this Document**

The Restaurant Point of Sale (POS) is a web application designed to digitise various restaurant management operations, including order processing, inventory management, and intelligent sales reporting.

**1.2 Intended Audience and Reading** **Suggestions**

This document provides a comprehensive description of the requirements for the "POS" software. It outlines the purpose, scope, and detailed specifications for system development. Additionally, it covers external interface, system, and non-functional requirements. This document is intended for customer approval and serves as a reference for future enhancements.

**1.3 Product Scope**

The POS system aims to automate daily restaurant tasks such as order and inventory management, billing, and feedback collection. This initial release focuses on these functionalities, with potential expansion in future versions. The system enhances restaurant efficiency, reduces human errors, optimises resource allocation, and facilitates sales trend analysis. The resulting benefits include improved performance, reduced labour costs, and increased revenue generation.

## **1.4 Definitions and Acronyms**

JWT: JSON Web Token, a secure method for transmitting information between parties as a JSON object.

API: Application Programming Interface, a set of protocols and tools for building software applications.

JPA: Java Persistence API, a Java specification for accessing, managing, and persisting data between Java objects and relational databases.

H2 Database: An in-memory and disk-based Java database.

MySQL: An open-source relational database management system.

React: A JavaScript library for building user interfaces.

Spring Boot: An extension of the Spring framework that simplifies the process of building production-ready applications.

**2. Overall Description**

This section will give an overview of the POS application. The basic functionality of the system as well its context will be explored in detail. It also describes different kinds of stakeholders and user classes associated with the system and what functionality is available for each class. At last, the assumptions and dependencies for the system are presented.

**2.1 Product Perspective**

POS app will attempt to replace the traditional manual ordering process and is a new self contained software system that consists of two parts: one Web application and the other is SQL database. The Web application will be used for ordering and interacting with the inventory while the SQL database will be used for storing the inventory and ordering related information about the food items like pending and complete order queues. The Web application will have five interfaces. Each for Customer, Manager, Head Chef, Admin and Chef. Manager can see/edit the status of available/reserved tables. Customer’s interface will consist of a scrollable menu listing available items and their price. When the customer selects some dishes and places the order, it will be stored in the “pending orders” table in SQL database. The Head Chef’s interface will be such that he is notified of the pending order and he is able to assign it to one the available queues of chefs who are then able to see the new order in their screens or on a central display in the kitchen. After each item/dish in an order is prepared, the order is marked completed through the Head Chef’s interface, the hall manager gets notified through his interface. Customer’s interface has an option for requesting the bill. Bill is printed through the Manager’s interface. Admin can change and modify the SQL database like add new menus or staff, edit current inventory stock etc.

**2.2 Product Functions**

Given below are the major functions that can be performed using a POS app. The system will:-

* Allow Customers to scroll through the menu and select the dishes he/she wants.
* Allow the Customers to cancel/edit the order any time before its prepared.
* Allow Customers to provide feedback regarding the food and overall service of the restaurant.
* Allow Customers to request for bill.
* Allow Customers to ask for help through the system.
* Assign the Head Chef to assign the dishes in an order to chefs according to their specialties.
* Show dish queues and their status, for Chefs.
* Allow admin to perform CRUD (create, retrieve, update and delete) operations on Staff Members, Menu Items and Inventory.
* Allow the Head Chef to mark orders complete.
* Allow the Head Chef to approve cancellation of the dish or order.
* Allow the Hall Manager to mark the bill as paid.
* Notify the Hall Manager when a particular order is complete.
* Allow the Hall Manager to see/edit status of tables reserved and available and their capacities.

**2.3 User Classes and Characteristics**

There are four types of users that interact with our system Firstly, there is a Hall Manager, then Customer, Head Chef and Admin. We’ll provide an interface for Chefs as well through which they are looking at the status of their order queues, but they will not interact with our system.

2.3.1 Customer Class Customers interact with our system directly in order to place orders, modify orders, get bills and give feedback. We do not store any information related to customers in our system. The process of order taking starts from customers placing orders and then the other series of events begin.

2.3.2 Head Chef/Kitchen Manager Head Chef can mark a dish as prepared when a chef tells him to do so. He can approve the cancellation of an order whenever a customer edits or removes a dish from his order. He can also assign a dish to a particular chef based on the specialty of the chef.

2.3.3 Chef Software Requirements Specification for POS Page 5 Chefs don’t interact with the system. They just have to look at the dishes present in their queues and prepare the dishes accordingly. Chef’s name, address and specialty etc. are stored in the database.

2.3.4 Admin Admin’s job is to manage the inventory and other information related to menu and chefs in the system.

## 2.3.5 Hall Manager Hall Managers will provide its input when he marks the bill as paid when customers pay for their order or get the bill printed. Moreover, he gets a notification whenever a particular order is complete, or some customer asks for help through the system. Hall managers can also see tables in the hall and their status i.e. empty or filled.

## **2.4 Operating Environment**

## It is an web application running on a tablet as well as on the desktop are present in a restaurant. Firstly, the manager would be present at the entrance and system in his tab would show the tables that are empty/reserved. There would be a tab present at every table for customers which they will use to give orders. When an order is placed the server would notify the head chef/ kitchen manager who would be in the kitchen. Head chef would use his tab which also would have the system installed and would add the order to the appropriate queues of the chefs. The chefs would be present in the kitchen area and their interface would allow them to check for the dishes they have to prepare. So, the system is running on various tablets but the operating environment and purpose of each is different for each user.

## **3. Functional Requirements**

Any anonymous User will be able register his/her restaurant in the system and can be able to add different products in their account in order to make them customise the menu. Registered shoppers will be able add their staff details.

### **3.1 Menu Building**

#### **Customizable Menu**

● Owners can customise the menu by adding, modifying, or removing items.

● Prices, descriptions, and availability of items can be adjusted.

#### **Menu Item Modifiers**

● Customers can customise orders by adding modifiers to menu items.

● For instance, preferences for cooking steaks or choosing pizza toppings.

#### **Menu Item Categories**

● Menu items are categorised as appetisers, entrees, desserts, and beverages.

#### **Menu Item Pictures**

● Visual representations can be added to menu items using pictures.

#### **Menu Item Descriptions**

● Detailed descriptions can be included for better customer understanding.

### **3.2 Table Management**

#### **Table Layout Management**

● Owners can assign tables graphically on the restaurant layout.

#### **Table Status Tracking**

● Real-time updates of table statuses (occupied, reserved, available) are displayed.

#### **Table Reservation Management**

● Online reservations can be made by customers; staff can manage and confirm reservations.

### **3.3 Sales Reporting**

#### **Restaurant Sales Report**

● Detailed sales report for a specific day, including total bills and applied discounts.

#### **Traditional Sales Report**

● Reports on sold items, prices, sales frequency, and other relevant data.

#### **Restaurant Reporting**

● Tracking of key restaurant operations like ingredient stock levels, popular menu items, and staff performance.

#### **Daily Sales Report**

● Compilation of daily and weekly sales data, categorising items, discounts, and complementary products.

### **3.4 Bill Printing**

#### **Customizable Receipts**

● Owners can personalise receipts with logos and branding.

#### **Item Description**

● Receipts include detailed descriptions of ordered items.

#### **Price**

● Price of each item is displayed on the receipt.

#### **Savings from Sale or Coupon**

● Receipts show savings due to discounts or coupons.

### **3.5 Stock and Inventory Control**

#### **Ingredient Stock Levels**

● Real-time tracking of ingredient stock levels.

#### **Purchase Orders (POs)**

● Generation of purchase orders for ingredients when stock falls below a threshold.

#### **Recipes**

● Management and tracking of ingredient usage in menu item recipes.

#### **Menu Costs**

● Calculation and tracking of menu costs based on ingredients and items.

### **3.6 Customer Relationship Management**

#### **Attach Sale/Transaction to Customer**

● Sales and transactions linked to specific customers for tracking spending.

#### **Purchase History**

● Record of customer purchase history for personalised service.

#### **Capture Customer Information**

● Storage of customer details like name, age, birthday, phone, and email.

#### **Email Marketing**

● Enable email marketing for customer engagement.

#### **Loyalty Program**

● Support for a built-in loyalty program rewarding repeat customers.

### **3.7 Smart Reporting and Analytics**

#### **Sales Reports**

● Detailed reports to analyse sales trends.

#### **Table Turnaround Times**

● Tracking of table occupation and availability times.

#### **Discount Monitoring**

● Monitoring and reporting of applied meal discounts.

## **4. Interface Requirements**

This section defines interfaces with external systems, users, and devices. It covers software interfaces, communication protocols, and data exchange methods for smooth operation and data flow.

## **5. Performance Requirements**

This section specifies expected behaviour under different conditions, including response times, resource usage, error rates, and other performance criteria.

## **6. Design Constraints**

Design constraints encompass hardware/software compatibility, security, algorithmic choices, and other factors influencing system design.

## **7. Non-Functional Attributes**

Attributes such as security, portability, reliability, and scalability are outlined, ensuring the system's quality.

● Performance:

● Response Time: Transactions must be completed within a second to avoid delays.

● Capacity: Initial single system with scalability for future use.

● Supportability:

● IDE and Coding Conventions: Eclipse with Spring Tool Suite (STS) plugin for uniform coding style.

● Design Constraints:

● Software Languages: Java EE 8, HTML, CSS.

● Architecture: Spring MVC framework.

## **8. Preliminary Schedule and Budget**

Provides an initial project plan, estimating development time and cost for the Restaurant POS System.

## **9. Appendices**

Supplementary information including references, definitions, acronyms, and abbreviations for enhanced document clarity.

## **Uses of SRS Document**

● Development: Understand project requirements.

● Testing: Formulate test plans based on specified behaviours.

● Maintenance: Reference for software functionality.

● Project Management: Basis for plans, schedules, and resources.

● Customers: Insight into product features.

● Contractual Agreement: Document serves as an agreement.

● Documentation: Structured description of requirements.