**Software Requirements Specification**

**for**

Restaurant Management System

**Version 1.0 approved**

**Prepared by Aiza, Ali, Haseeb**

**Sphynx Software Solutions**

**November 27**

**Table of Contents**

[**Introduction**](#_heading=h.ejv3q7sqet0t) **3**

[Purpose](#_heading=h.3znysh7) 3

[Document Conventions](#_heading=h.wmwsfj7yy0af) 3

[Intended Audience and Reading Suggestions](#_heading=h.whh1nfcjj0pw) 3

[Product Scope](#_heading=h.x6b2m4aqfndf) **3**

[**Overall Description**](#_heading=h.usjc2kne17tf) **4**

[Product Perspective](#_heading=h.2s8eyo1) 4

[Product Functions](#_heading=h.hi538oljd50o) 4

[User Classes and Characteristics](#_heading=h.ho4dh5pyeaie) 4

[Operating Environment](#_heading=h.16bl2nseizkm) 4

[Design and Implementation Constraints](#_heading=h.qh3eftcn6f9g) 5

[User Documentation](#_heading=h.k20clyx5meod) 5

[Assumptions and Dependencies](#_heading=h.842ztz49wqas) **5**

[**External Interface Requirements**](#_heading=h.u2m44fx1e87e) **6**

[User Interfaces](#_heading=h.2jxsxqh) 6

[Hardware Interfaces](#_heading=h.vk36ryodlowc) 6

[Software Interfaces](#_heading=h.jsxopt408q8l) 6

[Communications Interfaces](#_heading=h.f23bz1h6f9pf) **6**

[**System Features**](#_heading=h.6vyq2todkmbx) **7**

[Automated Food Ordering System](#_heading=h.1jyura6xnf5p) 7

[Table Management System](#_heading=h.frk7vsr7u1da) **7**

[**Other Nonfunctional Requirements**](#_heading=h.3whwml4) **8**

[Performance Requirements](#_heading=h.2bn6wsx) 8

[Safety Requirements](#_heading=h.hw4ujasx6yd9) 8

[Security Requirements](#_heading=h.ecr72sltixtv) 8

[Software Quality Attributes](#_heading=h.8nqeljavpssm) 9

[Business Rules](#_heading=h.9mfb12sjllwv) **9**

[**Other Requirements**](#_heading=h.5biost8f4chs) **9**

# Introduction

## Purpose

This project is planned to enhance the customer and consumer experience in the challenging area of restaurant logistics management. Since we are following the Agile Process Model, there will be several versions of this software, which will help us refine the software according to the dynamic needs of our unique customer. The scope of this project will include food ordering system mainly, and a supply replenishment system as an extra feature added to the main work.

## Document Conventions

We have used the standard IEEE formatting conventions to write this document. Please expect no changes from the official format other than the following:

|  |  |
| --- | --- |
| DB | Database |
| RMS | Restaurant Management System |

## Intended Audience and Reading Suggestions

This SRS document is intended for the use of the developers at Sphynx Software Solutions mainly, and of the customer (the restaurant) contact who follows up with this project.

## Product Scope

The product we are developing is intended to deal with the inefficiencies of the food ordering system at the customer restaurant. We are developing a system that will computerize food ordering at the restaurant and will effectively automate it.

In addition to this main objective of the product, we are also adding another supplemental functionality to our customer as a bonus for working with us and trusting our potential product. We will also provide a module for the restaurant to automate their supplies ordering, eliminating a lot of labor cost for the restaurant.

# Overall Description

## Product Perspective

We are building a restaurant management system. It is a self contained product that the staff of the restaurant will use to facilitate food ordering and table assignment.

## Product Functions

The functionalities of the system would be the following:

* Present the receptionist with an interface to assign table to customer.
* Store demographic information of customers in waiting list.
* Present the menu to customer for ordering food.
* Store an order and send it to the appropriate chef. In case of unavailability of dish,

the system sends an alert message to the customer. Otherwise, display a confirmation message to the customer.

* Upon the confirmation of the order, show an approximate countdown timer for the order.
* Present a customer with their bill and allow them to make payment.
* Notify janitor after a table has been vacated.

## User Classes and Characteristics

* Receptionist should be able to assign a table to the customer when they enter the
* restaurant.
* If there is no table available, receptionist should be able to put the customer on the
* waiting list. Customer should get an SMS with a table number as soon as a table
* becomes available.
* Customer should be able to search for available dishes and drinks.
* Customer should be able to place their order at their table using a tablet.
* Customer should be able to pay their bill.
* Customer should be able to submit feedback.
* The janitorial staff should be notified once a table is vacated.

<Identify the various user classes that you anticipate will use this product.User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

* Distributed Database
* Client/Server system
* Operating system: Windows
* Database: SQL database
* Platform: Python

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

The following are regulatory and corporate policies:

* Code of conduct
* Privacy policies
* Anti-harassment and non-discrimination policies

The following are software design and implementation constraints:

* Access to the web is required.
* Implement the database at least using a centralized database management system.
* Use the latest version of Python.
* Follow Python Naming Conventions.
* Don't use protected inheritance.

## User Documentation

Users will be given an online tutorial to understand the workings of the system. We will also set a quiz for users as a prerequisite before using the system.

## Assumptions and Dependencies

It is assumed that the user is familiar with an internet browser and also familiar with handling the keyboard and mouse.

Since the application is a web based application there is a need for the internet browser. It will be assumed that the users will possess decent internet connectivity.

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

The user interface of our system will be kept as simple as possible to make it as lay-person friendly as possible. Our market survey indicates that the hospitality industry management prefers simplistic graphical interface and layout when it comes to computer-based management systems. We will use TKinter for this. Every page will have a help button, a navigation bar and a sign out button. The front desk page (accessed by the receptionist) will have the following: a map of the seating area with tables color coded according to their availability, a waiting list button that will open a pop-up box containing a list of people on the waiting list, and an assign-waitlist button that will open a pop-up form where the receptionist can enter guests’ details to assign them on the waiting list. The menu page, which will be accessed by the seated customers, will have the following user interface: list of menu items with an add-to-cart button next to them along with their prices, and a button to place the order. This page will have unavailable items greyed out. The orders page will have a list of orders displayed according to which chef has the page open. This will help by only displaying orders that are relevant to each chef in the kitchen. The pay-bill page will contain a card payment system, a cash payment option and a feedback survey.

## Hardware Interfaces

The hardware interface of the system will consist of a database and a server. The communication protocol will be standard.

## Software Interfaces

The software used by this system is Mac and Windows operating systems, SQL database, Python for front-end and back-end, and TKinter for the user interface. The database must be the central storage of data.

## Communications Interfaces

The payment system will utilize the SSL protocol for secure payments and to protect the sensitive information of the customers. The SMS part of the system (required for the waiting list) will follow the MAP part of the SS7 protocol. And, we will utilize the sockets and the database for synchronization across pages and uses.

# System Features

As stated above, our software product will provide two main features, namely, the Automated Food Ordering System and the additional Table Management System will be a part of the RMS.

## Automated Food Ordering System

4.1.1 Description and Priority

This feature is the main component of this project as it will provide frequently used functions that will act as the backbone of the restaurant. This system is of an extremely high priority as it is the main objective of this project.   
Priority Ratings:

|  |  |
| --- | --- |
| Benefit | 9 |
| Cost | 5 |

4.1.2 Stimulus/Response Sequences

The system will provide a simple ordering system on tablets on each table. The restaurant’s customer will order food through the tablet on the system and the chef in the kitchen will be effectively notified of what is to be prepared. This will cut the waiters' work in half and also reduce errors and pedestrian traffic in the restaurant.

4.1.3 Functional Requirements

We will use Android tablets that will be more responsive in their compatibility with the Windows machine that will house our database systems. We will also need an SQL DB.

REQ-1: Android Tablets

REQ-2: Windows DB System

REQ-3: Trained Chefs

## Table Management System

4.2.1 Description and Priority

This feature is another important component of this project as it will help the receptionist manage tables, that is, it will help the receptionist assign tables, plan servicing the tables, and manage reservations more efficiency.   
Priority Ratings:

|  |  |
| --- | --- |
| Benefit | 9 |
| Cost | 3 |

4.2.2 Stimulus/Response Sequences

The system will provide a simple table management system that will be managed by the receptionist at the restaurant. The reservations will show up on the system, and the receptionist will be able to assign appropriate tables based on the time and numbers of the customers. The receptionist will also be able to assign tables in real time to people who show up without reservations. The receptionist will also manage the cleaning schedule of the tables after their use to make better use of the waiters’ manpower at the restaurant.

4.2.3 Functional Requirements

We will use a windows machine that will utilize the SQL DB system mentioned in the previous section. Using this will help us save costs in Database Management..

REQ-1: Windows DB System

REQ-2: Trained Receptionist

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

**EXAMPLE FOR FILLING THIS DOC:**

<https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>