Software Requirements Specification

For

DELIGHT KITCHEN

Kitchen Management System

Mentor

Mr. Suleman Sir

Prepared By

230340120015_AHER PRASHANT DADASAHEB
230340120053_CHAVAN PRASAD BHAIRAVNATH
230340120104_LOKESH
230340120111_MANGATE NILESH NARAYAN
230340120173_RAKESH PRABHUDAS CHAUDHARI

Table of Content

1. Introduction

1.1 Purpose

This document presents a detailed explanation of the objectives, features, user interface and application of Kitchen Management System(KMS) in real life. It will also describe how the system will perform and under which it must operate. This document will provide a baseline for design of the database, user interfaces, coding & evaluation of test plans. It will be used as a solid foundation for continued product evaluation.

The aim of this document is to explain the functionality of the project developed for Restaurant Management System(KMS). This document provides the flow about how the services are being managed and provided to a customer in an efficient manner whenever he makes a visit to a restaurant.

1.2 Definitions, Acronyms, and Abbreviations

We will use some acronyms throughout this document. Abbreviations and definition of some useful terms we will use are given below:

Term	Definition
SRS	Software Requirement Specifications
KMS	Kitchen Management System
DBMS	Database Management System
API	Application Program Interface
System Admin	System admin is a person who is responsible for managing the whole system and who has full access to the system
System User	A person who is using or operating the system but with a limited privilege
Database	Collection of all the information monitored by this system.
Field	A cell within a form

1.3 Project Scope

This system will help to manage the restaurant business systematically. In this management system, we will provide a web application that can be used by the Manager along with the staff of that restaurant to manage the food orders given by the customers, so that the manager of that restaurant can monitor and ensure the smooth conduct of the whole system.

This will lead to serving food faster and ultimately a better kitchen place to revisit it again. In addition to this, the system is capable of hiring different job roles required for a restaurant, the required information about staff will be saved in the system which can be only accessed by the system admin.

1.4 References for Requirement Analysis and Designs

- www.slideshare.net :- world's largest professional content sharing community, from this websites we take references regarding the blueprint of srs document.

- www.wikipedia.com :- free online encyclopedia

- IEEE recommended practices for SRS. ANSI / IEEE Std 830 – 1993.

- https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/ :- biggest tech article and details explanation provider

- https://www.w3schools.com/ :- online tutorials website

2. Overall Description

2.1 Product Perspective

The Kitchen Management System helps the Kitchen/restaurant manager to manage the activities more effectively and efficiently by computerizing menu ordering, billing and inventory control.

The system processes transactions and stores the resulting data. Reports will be generated from these data which help the manager to make appropriate business decisions for the Kitchen/restaurant. For example, knowing the number of customers for a particular time interval, the manager can decide whether more waiters and chefs are required. Moreover, easily calculate daily expenditure and profit.

The whole management system is designed for a general Computerized Digital Kitchen or Restaurant. So that any restaurant owner can get it and can start an automated process to his kitchen.

2.2 User Characteristics

In our project users are the staff members of the Kitchen/Restaurant, there is no outer user (customers) can access the application, they will just select the menu and are supposed to just order, and the orders will be handled by staff members who are working under a manager.

Our application is dedicated and mainly focused on making the functionality of a typical Kitchen/Restaurant in which the manager is an admin of our application who can have the access to manage the whole functionality of that web application.

A Customer arrives in a Kitchen/restaurant, any waiter who is allotted a particular table/s by the manager will take the order given by the customer, by looking through the menu section of that application, and will confirm the order which will be further received by the chef who can accept or deny that order. As soon as the confirmation of that order is approved by the chef, the details of that order will be sent to the billing section, which will be handled by cashiers. Finally, the bill of that particular order will be generated based on the quantity and food category selected as ordered by the customer.

Additionally, this web app is designed such that if any user wants to seek a job in the Kitchen, he can register himself by choosing a particular role required in that Kitchen.

2.3 Product Functionality

- ❖ There are various roles in a kitchen/restaurant
  - Manager
  - Chef
  - Cashier
  - Waiter

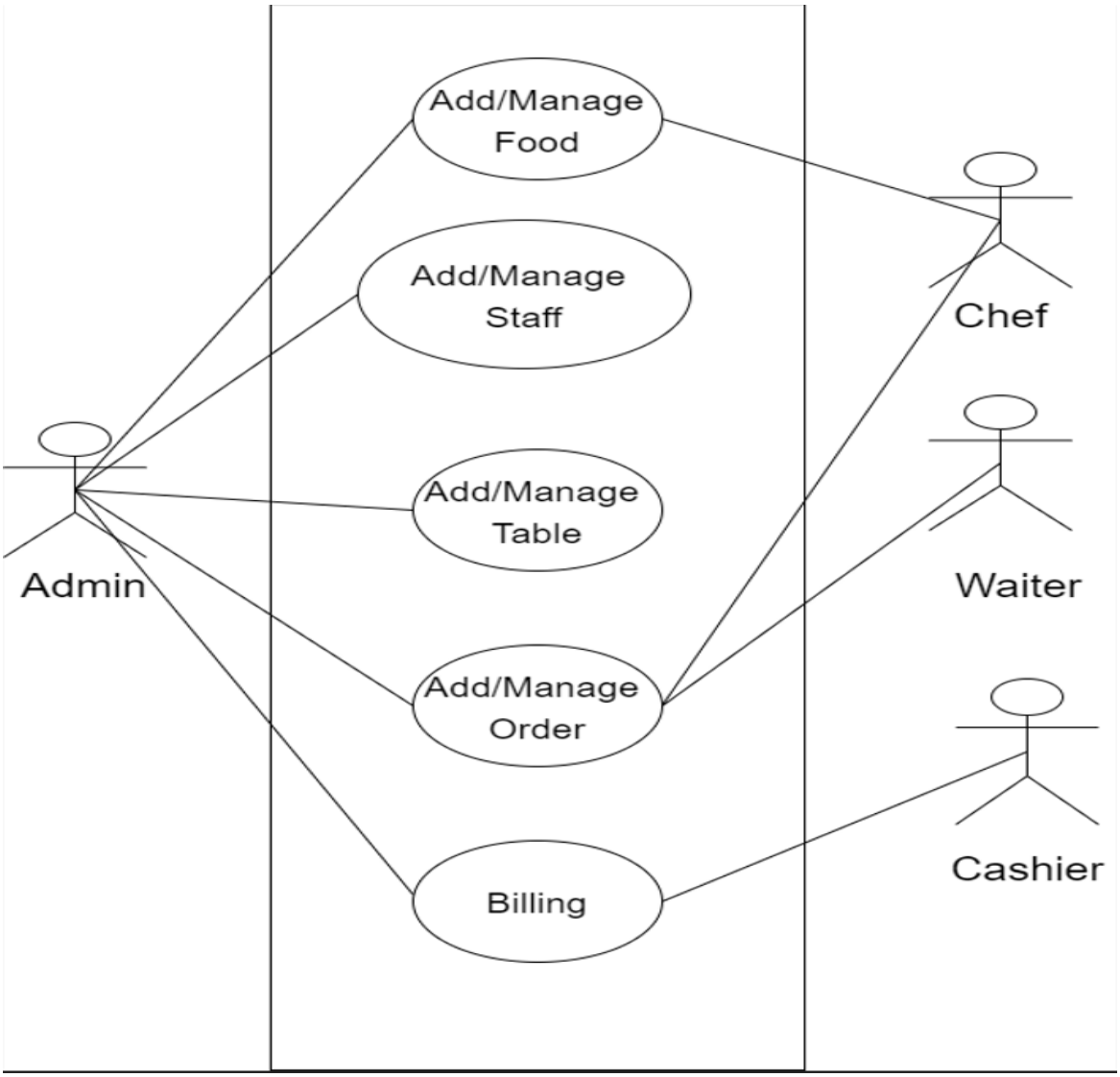
- ❖ Functionalities of different Role:
  - Manager
    - ➢ Add/Manage Food(Availability)
    - ➢ Add/Manage Tables
    - ➢ Add/Manage Staffs
    - ➢ Add/Manage Orders
    - ➢ Billing of Orders

  - Chef
    - ➢ Add/Manage Food (Availability)
    - ➢ Manage Orders

  - Cashier
    - ➢ Billing of Orders

  - Waiter
    - ➢ Add/Manage Orders

2.3.1 Use Cases



The use cases for each of the members are described in this section:

- **Chef Use Case**

Use case: Prepare Food

Description:

The chef can see the orders of customers and check whether this order can be taken or not and then confirms the order and starts preparing the food. When the food is ready the chef alerts the waiter to serve the food. He can also edit what ingredients are available and what ingredients are demanded.

- **Waiter Use Case**

Use case: Take Order and Serve Food

Description:

The waiter can see the food orders and the ready foods in the kitchen to be served. After serving the food the waiter will mark the order as served.

- **Cashier Use Case**

Use case: Receive payment

Description:

The Cashier can only take payment from the customer and save it into the system database with respect to the food item and also check if the customer is eligible for a discount. If yes, then take the payment accordingly.

- **Admin Use Case**

Use case: Maintain System Description (Manager)

Description:

The Admin has full access to the system. He will maintain flow of the whole system to ensure better and secure service and solve any error appearing in the system.

3. Specific Requirements

3.1 External Requirement

➪ Back-end Server Configuration:

- ★ For backend we have used J2EE (Spring Boot)
- ★ Minimum 4 gb ram required
- ★ Minimum intel core i3 required
- ★ Software or Ide we will use STS, Git etc.

➪ Database Server Configuration:

- ★ For database Management we will used RDBMS
- ★ MySql is the database software will use

➪ Frontend **Configuration:**

- ★ For frontend development ReactJS (JavaScript library) have used.
- ★ For that Visual studio environment (IDE) is also required.
- ★ Web Browser – Internet
- ★ Explorer/Netscape
- ★ Other client application software as per requirement

➪ Operating System Configuration:

- ★ For development windows is the perfect
- ★ But Linux OS can be also used

3.2 Performance Requirement

- ➢ The product will be based on a local server.
- ➢ The product will take initial load time.
- ➢ The performance will depend upon hardware components.
- ➢ Payment system will be fully secure through the POS system.
- ➢ Different databases for employees.

3.3 Safety and Security Requirements

- ➢ The source code developed for this system shall be maintained in configuration management tools.
- ➢ The whole system is secured only Admin (Manager) can access all the data.
- ➢ This system will use HTTPS, Because of this protocol this is more secure.
- ➢ Payment system will be fully secure through the POS system.
- ➢ Different databases for employees.
- ➢ It should be as per the industry standard.