Software Requirements Specification

for

Restaurant Management System

Version 1.0

Prepared by:

K S V KARTIKEYA-220953596
H GANAPATHI KAMATH-220953667
DEEPAK H SUVARNA-220953670

29-09-2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
RMS 1.0	29-09-2024	First version's specifications are defined	1.0

Team Members of the project:
GANAPATHI, SAI VASISTA, DEEPAK

1. Introduction

1.1 Purpose

RMS is an application that aims to digitalize the process of various restaurant management operations including ordering and inventory management and POS. This document aims to capture the system requirements and features particularly related to ordering to be implemented in RMS version 1.0.

1.2 Document Conventions

In this complete document, we will mention priority as “low” or “high” throughout the document and throughout this document, the first letter as capital and any significant term which has been described in bold, Font Type is Times New Roman, Font size is 12 for regular text

1.3 Intended Audience and Reading Suggestions

This document is intended for different types of readers such as restaurant owner, system designer, system developer and tester. By reading this document a reader can learn about what the project is implemented for and how it will present its basic ideas.
This document has a sequential overview of the whole project so if a reader reads the document from top to bottom, he will get a clear idea about the project.

1.4 Product Scope

RMS is a restaurant management system developed with the intention of automating the day-to-day tasks in a restaurant like order and inventory management, bill generation and sales report. This release of the software would deal with these tasks only whereas more areas might be automated in the future versions of this software. The main purpose is to improve the performance of the restaurant. the daily paperwork. With this system the tasks would be performed in less amount of time and more efficiently. An additional benefit of this software is that during the rush hours the load can be balanced effectively, and restaurants would perform better than usual. In addition to this, human error that occurs when performing tasks manually is also minimized and presence of queues in the system to assign tasks to chefs can reduce congestion in the kitchen. The system would also result in reduction of labor which would result in the reduction of expenses of the restaurant. Feedback module would help the restaurant check for how well they are performing, and monthly/yearly figures can be checked by the billing module to see the trends in sales reports. These benefits can potentially result in generation of more revenues for the restaurant.

1.5 References

1) *Google*

https://www.google.com/

2) WinForms c#
https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022

2 Overall Description

2.1 Product Perspective

RMS application will attempt to replace the traditional manual ordering process and is a new self-contained software: the application and the other is MS SQL server database. The application will be used for ordering and interacting with the inventory while the MS SQL server database will be used for storing the inventory and ordering related information about the food items like pending and complete order queues.

The application will have five interfaces. Each for Admin, Manager, Chef. Manager can see/edit the status of available/reserved tables. staff interface will consist of a scrollable menu listing available items and their price. When the waiter selects some dishes and place the order, it will be stored in "pending orders" table in MS SQL database. Chef's/kitchen 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 kitchen. After each item/dish in an order is prepared, the order is marked completed through the Chef's interface, the hall manager gets notified through his interface. Waiter interface has an option for requesting the bill. Bill is printed through the Manager's interface. Admin can change and modify the MS 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 RMS application. Moreover, a Data Flow Diagram (DFD) for better understanding of the system is also given in Appendix B.

The system will:
• Allow waiter to scroll through the menu and select the dishes customer wants.
• Allow waiter to request for bill.
• Assign 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 Head Chef to mark orders complete.
• Allow Manager to mark the bill as paid.
• Notify the when a particular order is complete.
• Allow the Manager to see/edit status of tables reserved and available.

2.3 User Classes and Characteristics

There are four types of users that interact with our system. Firstly, there is a manager, then waiter, Chef and Admin. We'll provide an interface for Chefs as well through which they are looking when waiter take the order from customer and the status of their order queues.

2.4 Operating Environment

It is an application running on a tablet and the tablets are present in a restaurant. Firstly, 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 order. 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

2.5 Design and Implementation Constraints

For the design we are using Guna.Winform ,Icons8 and default designs .

2.6 User Documentation

The software is accompanied by the following materials for further help:
- User Manual Version 1.0
- Online support at RMS2024@gmail.com

2.7 Assumptions and Dependencies

One assumption about the software is that it will always be used on tablets that have enough resources to run the application. If the tablet does not have enough hardware resources available for the application, there may be scenarios where the application does not work as intended or not even at all.

The application uses MS SQL database for Offline storage of information like orders and menu items that needs to be in working state. If the backend interface changes the application needs to be adjusted accordingly

3 External Interface Requirements

3.1 User Interfaces

1. User Interface
   The set interface will contain t screens. All three screen will have a consistent layout.

   ### Place Order
   In this screen, system shows a list of cards (UI Elements) of dishes. Each dish will have an image, its price per serving.

   ### Edit/Cancel Order
   After confirming the order. In the screen will be shown “Edit Order” and “Cancel Order” buttons here will also be a button to request for bill.

   ### Feedback
   In feedback screen, a button for “Request Bill” will be shown. Beneath this button We will display a form which will have different multiple-choice questions and a submit feedback button.

2 **Chef Interface**
   In chef interface, system will show all the current orders in detail i.e. all the dishes of a particular order. In each order, there is a button which will be used to mark that dish cooked.

3 **Admin Interface**
   Admin is authorized to perform CRUD operations on Staff Members, Menu Items and Inventory Items. He’ll be having three different screens for Staff Members, Menu Items, and Inventory.

3.2 Hardware Interfaces

Our system can interact with a hardware device directly. We have to connect our system to the bill printer for handing the hard copy of the bill to the customer. For billing module, we may have to use a credit card reader for payment, but the interaction and the results generated by that reader are just entered into our system manually by the user. Moreover, the central screen in kitchen which will be displaying the status of order queues.

3.3 Software Interfaces

● For Database services system shall use to MS SQL latest version released on January 2024.
● System shall use v4 support library Print Helper for connecting to the printer and a driver to connect to the kitchen screen

4 System Features

4.1 Place Order

4.1.1 Description and Priority

The system will give waiter the ability to place customer orders using our product. It will display List of available dishes in the menu where unavailable dishes will be grayed out. waiter will be able to select multiple dishes based on customer order and their quantity for a particular order.

Priority: high

4.1.2 Stimulus/Response Sequences

When user enters the order activity/page, initially system displays a list of available and unavailable dishes along with their prices.

1. **Stimulus:**

   Customer taps on an available dish.

   Response: System shows a popup having name of the dish and price per serving. Also, it contains a text box for the customer to enter the quantity, OK button and a Cancel button.

1.1. Stimulus:

Waiter taps on an unavailable dish.

Response:

Nothing happens.

Stimulus:

Waiter enters the quantity and press OK button.

Response:

System closes the popup, shows a small green tick mark at the side of dish.

Below the tick mark it shows quantity selected and total price of that dish.

2. **Stimulus:**

Waiter taps on confirm order button at the bottom

Response:

System closes the order screen and displays.

4.1.3 Functional Requirements

REQ-1: The system will show a list of (UI element) of dishes. Each will have a picture of the dish. Below the dish it shows the price in Rupees per serving.

REQ-2: The system must show all available dishes to the Customer / Waiter.

REQ-3: Tap on any of the displayed dish will result in a popup for quantity has been selected.

REQ-4: after receiving the order from customer the order will go to kitchen and be processed

4.2 Chef Order Queue

4.2.1 Description and Priority

Whenever a new order is placed by the Waiter given by the customer, the dishes in the orders are classified into categories. The system has the information of specialty of each chef, it will assign each dish to a corresponding chef and place it in the order queue of that chef. There is a centralized screen in the kitchen which displays queues for each chef. Each item in the queue is labeled with the name of the dish.

priority: high

4.2.2 Stimulus/Response sequences

1. **Stimulus**: Waiter taps the "Confirm Order" button in "Place Order screen".
   Response:
   Displays the dishes on kitchen screen in corresponding chef's queue.

4.2.3 Functional Requirements

REQ-1: System will classify the dishes in the order and add this dish on a particular chef's queue in the kitchen screen.

4.3 Mark Dish as Completed

4.3.1 Description and Priority

The head chef can mark the dish of a particular order complete when notified by the chef.

priority: high

4.3.2 Stimulus/Response sequences

The system will show a list of current orders in earliest first order in head chef screen. Each order also shows order no and table no associated with the order. Moreover, it also shows a list of dishes for each order. Alongside of each dish there is a button saying, "Marked Completed".

Stimulus:
Head chef taps on the "Mark Cooked" button on a dish in an order.

Response:
System changes that button to a green tick.

2. **Stimulus:**
   All the dishes of a particular order have been marked "cooked"
   Response:
   System shows "Order of Table No is ready for serving".
   System shows Title "Food Completed". And can "Request Bill".

4.4 Request Bill

4.4.1 Description and Priority

Generate bill option gives the ability to the waiter to ask for receipt waiter will give the bill to customer and pay the bill.

priority: high

4.4.2 Stimulus/Response sequences

1. **Stimulus:**
   Waiter taps on the generate bill
   Response:
   The system prints the bill through a printer. System will add a bill to the manager's view and it shows "paid".

4.4.3 Functional Requirements

REQ-1: The system must display the details generate for a bill
REQ-2: The system must show manager the order no, table no and total payable amount
REQ-3: The system must give ability to the manager to change the status of the bill to paid.

4.5 Add/Edit/Delete Staff Members

4.5.1 Description and Priority

The system gives ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers etc.

priority: high

4.5.2 Stimulus/Response sequences

Admin screen shows a grid of staff members. There is a button at the top of grid which says Add Member. In the grid after every entry there is a “Edit” and “Remove” button.

1. **Stimulus:**
   Admin taps on “Add Staff” button
   Response:
   System opens another screen with a form

2. **Stimulus:**
   Admin fills the information and hit submit **Response:**
   System responds with “<Staff Member> added successfully”

3. **Stimulus:**
   Admin taps on edit button
   Response:
   System opens a screen with a form prefilled with the existing values.

4. **Stimulus:**
   Admin edits the information and hit submit
   Response:
   System responds with “<Staff Member> edited successfully”

5. **Stimulus:**
   Admin taps on remove button on a particular row
   Response:
   responds with a “<Staff Name> removed successfully”

4.5.3 Functional Requirements

REQ-1: Admin should be able to add all necessary information about the staff member

REQ-2: System must give admin the ability to edit information about all staff members

REQ-3: System must give admin the ability to remove staff members.

4.6 Add/Edit/Delete Menu Items

4.6.1 Description and Priority

The system gives ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.

priority: high

4.6.2 Stimulus/response sequences

Admin screen shows all the previously added dishes. It also shows a “Add Dish” button along with “Edit” and “Remove” with all the available dishes

1. **Stimulus:**
   Admin taps on “Add Dish” button
   Response:
   System opens another screen with a form

2. **Stimulus:**
   Admin fills the information and hit submit
   Response:
   System responds with “<Dish> added successfully”

6. **Stimulus:**
   Admin taps on edit button
   Response:
   System opens a screen with a form prefilled with the existing values.

7. **Stimulus:**
   Admin edits the information and hit submit
   Response:
   System responds with “<Dish Member> edited successfully”

8. **Stimulus:**
   Admin taps on remove button on a particular row
   Response:
   responds with a “<Dish> removed successfully”

4.6.3 Functional Requirements

REQ-1: Admin should be able to add all necessary information about the staff member

REQ-2: System must give admin the ability to edit information about all staff members

REQ-3: System must give admin the ability to remove staff members.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The system must be interactive, and the delays involved must be less. So, in every action response of the system, there are no immediate delays. In case of scrolling through the menu there should be a delay of no more than 2 second before the next page of menu items is displayed otherwise our people's dining experience is affected. The order should be placed in pending orders and be visible to the head chef/chefs in less than 1 second to start the preparation.

Order updates must be made with little delay to avoid delivery delay. Also, when connecting to the MS SQL server the delay to make a successful connection should be less for effective real time communication.

5.2 Safety Requirements

The software is completely environmentally friendly and does not cause any safety violations. The menu will have a flexible font that can be zoomed so as to not over constrain the eyes

5.3 Security Requirements

There is a need for a proper and encrypted login authentication for head chef and admin as employee sensitive information as well as inventory should be protected from hacking. Information transmission should be securely transmitted to MS SQL without any changes in information to avoid disturbances in orders and billing

5.4 Software Quality Attributes

5.4.1 Adaptability:

There can be a change in the menu and information stored in the database about employees and inventory.

5.4.2 Availability:

The system is up and running for most of the time and server is not down for more than a few minutes to avoid inconvenience of the customers.

5.4.3 Correctness:

The bill generated by the application must be accurate and the orders placed should exactly be the same which the user has selected.

5.4.4 Flexibility:

If need arises in the future, software can be modified to change the requirements.

5.4.5 Interoperability:

The data is transferred from the customer's end to the kitchen and then chef assigns orders to each chef. This way data is transferred from one part of the system to another.

5.4.6 Maintainability:

Software can be easily repaired if a fault occurs.

5.4.7 Portability:

Software can be easily installed on devices and would run smoothly according to the requirement.

5.4.8 Reliability:

No matter how many orders are placed, system must give the correct results.

5.4.9 Reusability:

Current version can be used in the future versions with more functionality added.

5.4.10 Robustness:

Software must have checks to ensure that the items that are not available in the menu cannot be selected and the emails, phone numbers added are all valid.

5.4.11 Testability:

All the requirements are fulfilled, response time is low, and all functions are working perfectly.

5.4.12 Usability:

Interface of the software must be easy to use. It would not be complex since managers, chefs have a view, so interface should be simple.

5.5 Business Rules

1. Manager's interface contains the view of tables that are free, and manager can just view and doesn't provide any input to the system.
2. Once the bill is paid, the bill is marked paid.
3. Admin has access to perform add, delete, update operations on the database for menu, inventory, employees and no other person can modify the data in the db.
4. Waiter can place customer order from the list of available items and can update order and pay bill.
5. Main chef assigns orders to chefs and can update the queues and has an additional functionality of load balance.
6. Chefs can only view the orders and cannot remove an order from their queue.

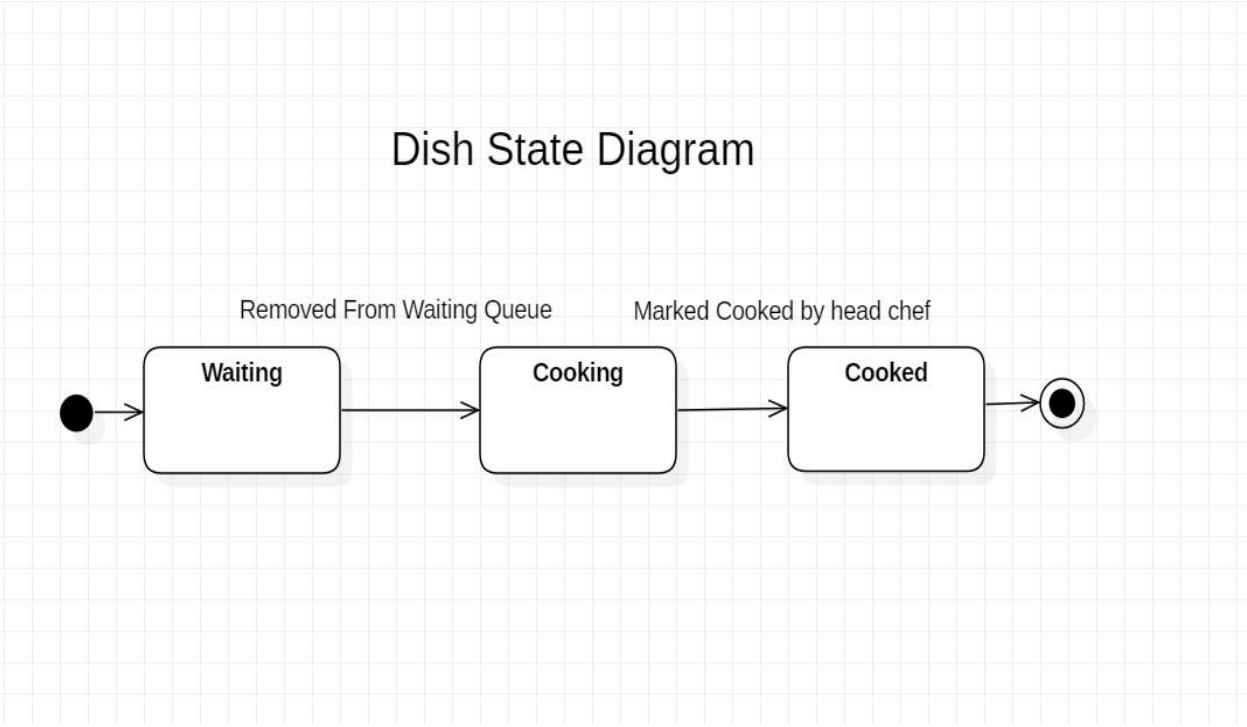
Appendix A: Glossary

CRUD: In computer programming, create, read, update, and delete (CRUD) are the four basic functions of persistent storage. Alternate words are sometimes used when defining the four basic functions of CRUD, such as retrieve instead of reading, modify instead of update, or destroy instead of deleting.

Print Helper: It is an android library that is used to connect to remote printer and send commands to that printer for printing.

Appendix B: Analysis Models

Dish State Diagram



Appendix C: To Be Determined List

Weekly sales report and tracking most ordered dish and prioritizing its inventory stocking feature (restock the items that are most ordered often) is yet to be determined by the client and may need further meetings for elaboration.
Adding POS (point of sale) features to the application is yet to be determined as well.