Software Requirements Specification

Version 1.0

<<Annotated Version>>

June 24, 2020

Inventory control System

Hasan alali Teamleader

Haider kamaria

Submitted in partial fulfillment

Of the requirements of

CS 310 Software Engineering

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made. Refer to the SRS Template for details on the purpose and rules for each section of this document. >>

[Table of Contents i](#_Toc77487619)

[List of Figures 3](#_Toc77487620)

[**1**.**0. Introduction** 4](#_Toc77487621)

[1.1. Purpose 4](#_Toc77487622)

[1.2. Scope of Project 4](#_Toc77487623)

[1.3. Glossary 5](#_Toc77487624)

[1.4. References 5](#_Toc77487625)

[1.5. Overview of Document 5](#_Toc77487626)

[2.**0** **Overall Description** 6](#_Toc77487627)

[2.1 System Environment 6](#_Toc77487628)

[2.2 Functional Requirements Specification 7](#_Toc77487629)

[2.2 Manager Use Case 7](#_Toc77487630)

[Use case: Add Recipe 7](#_Toc77487631)

[Use case: Remove Recipe 8](#_Toc77487633)

[Use case: Update Recipe 9](#_Toc77487635)

[Use case: Update Inventory 10](#_Toc77487637)

[Use case: Add Occasion 11](#_Toc77487638)

[2.3 User Characteristics 12](#_Toc77487648)

[2.4 Non-Functional Requirements 12](#_Toc77487649)

[3.0 **Requirements Specification** 12](#_Toc77487650)

[3.1 External Interface Requirements 12](#_Toc77487651)

[3.2 Functional Requirements 13](#_Toc77487652)

[3.2.1 Add Recipe 13](#_Toc77487653)

[3.2.2 Remove Recipe 14](#_Toc77487654)

[3.2.3 Update Recipe 15](#_Toc77487655)

[3.2.4 Add Occasion 16](#_Toc77487656)

[3.2.5 Update Inventory 17](#_Toc77487657)

[3.3 Detailed Non-Functional Requirements 17](#_Toc77487665)

[3.3.1 Usability 17](#_Toc77487666)

[3.3.2 Relaibility 17](#_Toc77487666)

[3.3.3 performance 17](#_Toc77487666)

[3.3.4 supportability 17](#_Toc77487666)

# List of Figures

Figure 1 : System Environment...............................................6 FIGURE2 : logical structure of inventory manager data………..18

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the inventory control system . It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate. This document is intended for both the stakeholders and the developers.

## 1.2 Scope of the project

This system is designed to maximize the productivity through automation the sale and buy operations, it will convert all paper operations into computer operations

the System aims to provide an efficient interface to the restaurants for managing their grocery inventory based on each item sold. The basic idea involved here is that each item is linked to its atomic ingredients which are stored in a database. At the end of each day, the system analyses the total sale of menu items and proportionately deducts appropriate amount from the resource database. Then it compares the current available resources with the threshold level of each ingredient***.***

## 1.3. Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Manager | The manager of the restaurant/company who handles all the administrative works |
| Recipe | This is the menu item that the restaurant/company provides to its customers |
| Database | Collection of all the information monitored by this system. |
| Vendor | This is the company that provides the restaurant/company with the required ingredients |
| Order | Order is the list of ingredients and the quantities that is or is to be requested from the vendor. |
| Ingredient | This is the entity that the recipe is composed of |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification

Both sections of the document describe the same software product in its entirety, the third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

**2.0. Overall Description**

***2.1 System Environment***

System Interface

Inventory Control System

Manager

**Figure 1 : *System Environment***

The inventory control system has only one active actor(The manager only)

and one cooperating system.

## 2.2 Functional Requirements Specification

***2.2.1******Manager Use case***

**Use case** : Add Recipe

**Diagram**

manager

Add recipe

**Brief description**

The manager opens the System and add the new recipes he wants to the system

**Initial Step-By-Step Description**

1-The manager press “add new recipe “button

2-the manager inter the new recipe ingredients

3-the manager submits the form to the system

4-the system adds the recipe to the database

**Use case :**Remove recipe

**Diagram**

manager

Remove recipe

**Brief description**

The manger opens the system deletes an exist description

**Initial Step-By-Step Description**

1-the manger press Remove recipe button

2- the system shows the list of recipes

3-the manager choose the recipe he wants to delete

4-the manager confirm delete operation

**Use case:** Update Recipe

**Diagram**

manager

Update Recipe

**Brief description:**

the manager updates an exists recipe

**Initial step-by-step**

1-the manager press “update recipe” button

2-the system displays the list of recipes

3- the manager chooses the recipe he wants to update

3-the manager inter the new ingredient

4-the manager press confirm update button

**Use case:** update inventory

**Diagram**

manager

Upadate Inventory

**Brief description**

The manager updates the inventory status

***Initial Step-By-Step Description***

1-the manager press update inventory button

2-the system asks for details of received ingredients

2-the manger enter the new date of received goods

3-the system takes the new ingredients

**Use case:** add occasion

**Diagram**

manager

Add occasion

**Brief description**

The manager adds the occasions

***Initial Step-By-Step Description***

1-the manager presses the add occasion button

2- the system displays a form for more details

3- the manager fills the form with the occasion names and dates

4- manager then press confirm button

**2.3 User characteristics**

The manger is expected to be windows literate and able to use buttons, mouse and similar tools

***2.4 Non-Functional Requirements***

1-The system must be easy to navigate from users with buttons that are easy to understands.

2-The system must offer an easy and simple way of viewing the current inventory status

3-It must provide a password enabled login

4-The software must incorporate a license key authentication process

5-The system must be speed without any lag

6-It must perform all actions quickly and in a successful way ever time

**3.0 Requirements specification**

**3.1** **External Interface Requirements**

The system is not dealing with any external system

**3.2 Functional requirements**

**3.2.1 Add recipe**

|  |  |
| --- | --- |
| Use case name | Add Recipe |
| Participating Actors | Manager(admin) |
| Trigger | The manager presses the add recipe button |
| Precondition | The system is running |
| Basic Path | 1. The Manager activates the “Create New Recipe” function on his/her terminal 2. The System responds by presenting a form to the Manager. The form asks for details associated with the recipe. 3. The Manager completes the form by inserting ingredients to be used in the new recipe. It also adds any new ingredient used in the recipe, after the form has been completed the Manager submits the form to the System. 4. The System acknowledges that the new recipe has been created. It also adds it to the recipe |
| Postcondition | The Manager(admin) is logged on to the System |
| Exception Paths | The manager may close the program at any time |

**3.2.2 Remove Recipe**

|  |  |
| --- | --- |
| Use case name | Remove Recipe |
| Participating Actors | Manager(admin) |
| Trigger | The manager presses the remove button |
| Precondition | The system is running |
| Basic Path | 1. The Manager activates the “Remove Recipe” function on his/her terminal. 2. The System responds by showing the current list of recipes saved on the System. 3. The Manager chooses which recipe(s) to remove and removes them by selecting a delete button through the terminal window. 4. The System confirms with each deletion with the Manager if he/she wants to delete the recipe. 5. The Manager confirms his/her decision with a yes/no. 6. . The System acknowledges the decision by either removing the recipe if responded with “yes” or by cancelling the delete if responded with “no”. It then displays an acknowledgment of the decision by displaying a delete successful or a canceled request. |
| Postcondition | The Manager(admin) is logged on to the System |
| Exception Paths | The manager may close the program at any time |

**3.2.3 Update recipe**

|  |  |
| --- | --- |
| Use case name | Update recipe |
| Participating Actors | Manager(admin) |
| Trigger | The manager presses the update button |
| Precondition | The system is running |
| Basic Path | 1. The Manager activates “Update Recipe” on system. 2. The System responds by bringing up list of recipes. 3. The Manager selects a recipe to change 4. System now shows a update Recipe form with the list of ingredients in the recipe and corresponding amount. 5. The Manager changes the recipe by adding/removing ingredients or updating the amount of ingredients used in the recipe. 6. The Manager then finishes the update by pressing the finish button on the system. 7. The System confirms that the change has been made and updates the databases |
| Postcondition | The Manager(admin) is logged on to the System |
| Exception Paths | The manager may close the program at any time |

**3.2.4 Add Occasion**

|  |  |
| --- | --- |
| Use case name | Add Occasion |
| Participating Actors | Manager(admin) |
| Trigger | The manager presses the add occasionbutton |
| Precondition | The programme is running |
| Basic Path | 1. The Manager activates the “Add Occasion or Event” function on his/her terminal. 2. The System displays a form to be filled out by the manager. 3. The Manager fills out the form by adding a name of the event or occasion and selecting the date(s) the event is to be held. 4. The Manager now fills list of recipes that will be utilized more on the selected day. 5. The System takes the data from the form and calculates the amount of ingredients that may be used up for the given dates based on past data and adds the data to the ingredient database |
| Postcondition | The Manager(admin) is logged on to the System |
| Exception Paths | The manager may close the program at any time |

**3.2.5 Update Inventory**

|  |  |
| --- | --- |
| Use case name | Update Inventory |
| Participating Actors | Manager(admin) |
| Trigger | The manager presses the update inventory button |
| Precondition | The programme is running |
| Basic Path | 1. The Manager activates the “Update Stock inventory” function on his/her terminal. 2. The System now presents a form to the Manager asking for details of the received amount of ingredients. 3. The Manager enters the ingredients and the corresponding quantity received and presses the submit button. 4. The System adds the corresponding amount to the resources database and acknowledges the completion of the process. |
| Postcondition | The Manager(admin) is logged on to the System |
| Exception Paths | The manager may close the program at any time |

**3.3 Detailed non-functional**

• **3.3.1 Usability**

The system must be easy to use by managers such that they do not need to read an extensive amount of manuals.

**• 3.3.2 Reliability**

The System must give accurate inventory status to the user continuously. Any inaccuracies are taken care by the regular confirming of the actual levels with the levels displayed in the system.

o The System must successfully add any recipe, ingredients, vendors or special occasions given by the user

• **3.3.3 Performanc**

The system must not lag, be quickly and perfume without any fault

• **3.3.4 Supportability**

The software is designed such that it works even on systems having the minimum configuration.

**1** Adds/deletes/updates **\***

Manager

Recipe

**1**

**1**

**\***

Ingredients

Vendors

**\***

**FIGURE2 : logical structure of inventory manager data**