

# **Working Title**

***Software Requirements Specification (SRS)***

# CSC.154.0001 - Group 4

*Eric Perry*

*Eric Hodge*

*Oras Alhammadi*

*George Saad*

*Steven Torres*

*February 15, 2019*

*Version 1.0*



**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| 1st Draft | Eric Perry  Eric Hodge  Steven Torres  Oras Alhammadi  George Saad | Initial Draft of SRS Doc for Working Title (Group 4’s Project – Spring 2019) | 02/15/19 |

**Review & Approval**

**Requirements Document Approval History**





**Table of Contents**

Working Title [1](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__667_870855115)

CSC.154.0001 - Group 4 [1](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__669_870855115)

1. General Description ….[4](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__671_870855115)

2. Business Requirements [5](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__673_870855115)

3. User Requirements [5](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__675_870855115)

4. Hardware Requirements……...……………………………………………………………6

5. Functional Requirements [6](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__677_870855115)

6. Non-Functional Requirements [6](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__679_870855115)

7. Preliminary Use Case 6-8

9.1 Use Case Model [9](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__687_870855115)

8. Appendices [8](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__693_870855115)

11.1 Definitions, Acronyms, Abbreviations [6](file:///C:\Users\hodge\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\TempState\Downloads\CSC1540001_Group4_SRS_Draft1%20(1).doc#__RefHeading__695_870855115)

## 1. General Description

**1.1 Introduction**

The document is a compilation of identified requirements for (Working Title) App, an application for tracking and managing an ingredients inventory for food industry clients.

**1.2 Scope of this Document**

The scope of this document will cover various categories of requirements for the (Working Title) app. It will include user stories to highlight user work flow and identify the needs of various forms of users when utilizing the app.

**1.3 Overview**

*(Working Title)* is an inventory management and tracking application that utilizes a relational database. The system stores an inventory of individual ingredients as well as tables of menu items linked to those ingredients. Users will be able to update, in real time, inventory quantities while processing orders from front house to back house. Wait staff and cooks will be notified if an order cannot be completed due to inventory shortage. Managers will be able to view ingredient and menu item history, to include frequency of use/order by period of time, as well as place orders with vendors online or print out an purchase inventory. Market prices for ingredients will also be available through third party integration of APIs/EPIs.

**1.4 Business Context**

Wait staff will be able to assist with accurate inventory management by simply entering orders in the system. Management will be able to accurately track inventory, ordering trends, prices, and be able to place orders with a variety of vendors. Third party vendors will be able to supply API/EPIs so managers will have an up to date market price for selected ingredients.

## 2. Business Requirements

**2.1 Inventory tracking**

Real time inventory updates and requests. Inventory may be viewed or updated individually or as a dish. Dishes may also have alterations made to their base ingredients. Ingredients will also have an expiration tracker.

**2.2** **Inventory history**

Retain records of previous inventory states and orders by date for up to 10 years. Inventory will retain ingredients, dishes, quantities, number of times a dish was ordered, market price of ingredients for that day, what dishes were on special. Records may be annotated with additional information in notes.

**2.3 Trend analysis**

The system will be able to track ordering trends for a period of time, defaulted to a weekly and a monthly view. It will track usage of ingredients and number of times a dish was ordered per time period. It will allow comparison between two periods of time as well. Dishes may also be marked as a special, or as part of a special holiday or event menu.

**2.4 Ordering and Market Price Updates**

The system will allow for the creation of an inventory for order, which may be printed. The system will integrate with third party API/EPIs to allow for real time market pricing of ingredients from selected vendors. The system will allow a filter of vendors for ingredients to be saved as well as notes for each vendor. Ordered items will also have a date ordered for tracking.

## 3. User Requirements

* 1. The user will be able to store menu items with the exact ingredients and cost for each menu item.
  2. The user will be able to add, modify, or delete daily count for a full weeks’ worth of inventory.
  3. The user will be able to track the expiration date of each item and a place for food thrown out.
  4. The user will be able to track the average amount an item is order over a week, month, and year.
  5. The user will receive alarm when certain inventory items are low enough or out.
  6. Incoming inventory will show an ordered-on date with an amount.
  7. Orders from the front of the house will print an order ticket to the kitchen with a timestamp.
  8. Separate user permissions for managers vs servers or cooks.

## 4. Hardware Requirements

The restaurant will require:

- an internet service with a minimum of 300mpbs.

- a desk top computer with 2TB of storage

- POS System with multiple tablets or computers

## 5. Functional Requirements

* 1. Using database to store menu items with ingredients and cost for each menu item as well as cost for each inventory item.
  2. Using a separate section to track the expiration date of each item.
  3. Make calculations using python to update the database with number of items and alarm the user when certain inventory items are low enough or out.
  4. Using python to create an average amount of ingredients used per week, month, and year.
  5. Integrate with third party API/EPIs for market cost of ingredients.
  6. Separate logins to the app with different level permissions

## 6. Non-Functional Requirements

1. The system should be available from 9am to 11pm Monday – Friday.
2. The system should be able to hold 10,000 item records initially.
3. The system should be able to add 10,000 records a year for 10 years.
4. A record should be fully available on the system for at least 7 years.
5. The system should deconflict simultaneous updates from two different input sources
6. The data can be exported to the manager or vendor to make more

portable.

1. System will update in real time.

## 7. Preliminary User Story

### 7.1 Manager User Story

The manager sits down at office computer and uses the mouse to double click to start the program, it loads up to the login page. The manager puts in there given username and password. After less than a minute the home page loads.

The manager decides they want to check yesterday’s inventory used and compare it to the same day the week before. They click the tab labeled ‘Inventory History’ which takes them to a new window with multiple dropdown list where they can choose the year, month, week and day. The manager picks a day, then on the home page repeats the process to pull up another day to compare.

Then the manager decides they need to change the cost of some of the ingredients because their suppliers increased the cost. On the home page they click the tab labeled ‘manual inventory update’ which will open a separate window with a list of categories of the inventory as well as a search box. The manager knows which items they have to change so they type the item in the search box which will show the item with all the associated information in an edit mode. They change the cost and then at the bottom and the top of the page there is a ‘Submit’ button that will confirm the change.

The manager notes that some fish in the inventory has been marked for an expiration coming soon. While searching for fish to order, the manager sees a note that a particular vendor had a high-quality selection. This particular vendor has an EPI that allows the manager to see real time market pricing for the item. He filters the vendors for that item, saving the filter selection, and checks the price that the vendor has for the fish. He adds it to his order inventory.

While looking over the order trends, the manager notices that a particular dish has not been selling well. He marks it as a special, entering a time date for the period of the special. The manager also sees that a particular dish was very popular over the last month, but ingredients often ran out before the end of the week. He adjusts the minimum quantity to keep on hand in the inventory and updates his order inventory.

Finally, the manager prints out his order inventory. He orders what he can through vendor's online services and enters in an ordered-on date for each ingredient.

### 7.2 Wait Staff User Story

A server clocks in for their shift and enters in their user name and password for the app. A menu appears breaking down dishes and beverages by category.

A couple come in to the restaurant and are greeted and sat by the hostess. The server introduces themselves and gives them water. The customers have very complicated orders. The server writes it down and then proceeds to put the order in using the company supplied ordering system that has all the modification buttons for the complicated customer orders. The server first enters the table number for the order, and then selects the dinner menu to bring up a list of dishes currently on the menu. She selects the dishes ordered, checking the modifications box, which opens a menu. She enters in the name of ingredients that the customer wishes to add, adjusts the number of items already in the dishes to the customers preference, and removes ingredients that they did not want.

The ordering system automatically routes the information to the kitchen ticket printer and also to the inventory tracker. The inventory tracker will then make the calculation to update the database with amount of ingredients required for that menu item, also taking into account the modifications. If the menu item requires ingredients that are not available, the system will notify the server before submitting the order for printing. Dishes that have been identified as requiring ingredients that are not available will now be highlighted in red in the user interface for the wait staff. Dishes that were modified will have the changes listed under each dish on the print ticket in bold.

## 8. Appendices

### 8.1 Definitions, Acronyms, Abbreviations

Front House – Service area where customers are served. Wait staff working domain. This is where orders from customers are taken and entered into the system.

Back House – Working domain of kitchen staff. This is where orders from the front house are prepared. Orders entered into the system from the front house route an order ticket to the back house for preparation.

## 9.1 Use Case Model

