

---

# Software Requirements Specification

for

## Authentic Chef

Version <1.0>

Prepared by

**Group Name:** Team Green

Alexander Her  
Koryale Brooks  
Randy Saetern  
Megan Tamiyasu  
Narendra Mannan

110228436  
109168767  
110436592  
108405160  
300042172

akher@mail.fresnostate.edu  
Koryale16@mail.fresnostate.edu  
randysaetern@mail.fresnostate.edu  
MTamiyasu427@mail.fresnostate.edu  
narenman7@mail.fresnostate.edu

<b>Instructor:</b>	Dr. Shih-Hsi "Alex" Liu
<b>Course:</b>	CSCI 150
<b>Lab Section:</b>	<i>T-TH 3:15 – 4:20</i>
<b>Teaching Assistant:</b>	Ms. Joshly Papiril Johnson
<b>Date:</b>	November 2, 2020

# Contents

<b>CONTENTS.....</b>	<b>II</b>
<b>REVISIONS.....</b>	<b>II</b>
<b>1 INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE .....	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
<b>2 OVERALL DESCRIPTION.....</b>	<b>2</b>
2.0 PRODUCT OVERVIEW .....	2
2.1 PRODUCT FUNCTIONALITY .....	2
2.2 DESIGN AND IMPLEMENTATION CONSTRAINTS .....	2
<b>3 SPECIFIC REQUIREMENTS .....</b>	<b>3</b>
3.0 EXTERNAL INTERFACE REQUIREMENTS .....	3
3.1 REQUIREMENTS .....	3
3.2 USER INTERFACE REQUIREMENTS.....	4
3.3 FUNCTIONAL REQUIREMENTS:.....	5
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>8</b>
4.0 PERFORMANCE REQUIREMENTS.....	8
4.1 SAFETY AND SECURITY REQUIREMENTS .....	8
<b>5 PRIORITIZATION.....</b>	<b>9</b>

# Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Alexander Her Narenda Mannan Megan Tamiyasu Koryale Brooks Randy Saetern	First initial draft of SRS Requirements. Finished all of the required fields.	11/02/2020

## 3 Introduction

*Authentic Chef is a Full-Stack project that utilizes the MERN Stack methodology to create a user-centric food e-commerce application.*

### 3.1 Purpose

Authentic Chef is a Full-Stack e-commerce application that allows users to market their own products as well as buy products from other users. The purpose of this wiki is to showcase the project requirements for the development of the application. The software requirements are an essential part of the project to clearly identify project milestones and priorities.

### 3.2 Product Scope

Authentic Chef is a user-centric focused application. Most food e-commerce applications include restaurants but there did not exist an application for users to market their own food. The authenticity in the food should serve to match the application name as the food will be made by a variety of local chefs. With the rise of COVID-19, our application would also aid in the creation of more jobs as people would be able to market their own cooking skills to others to make an income.

Every user will have the option to register as a buyer or seller. As a buyer, the user should be allowed to browse existing food options and chef profiles to determine if they would like to purchase food from them. As a seller, the user will be allowed to upload pictures onto their own chef profile and market their products on the application. To become a seller, the user will need to be approved by a system administrator.

### 3.3 Intended Audience and Document Overview

The intended audience will be the developers that need to review this document should any questions or issue rise during development and the professor who will be reviewing the quality of the requirements. This document should serve to provide any users a better understanding of the project scope and requirements.

This wiki intends to go in-depth about the project's scope as well as the requirements that need to be followed. This document will go over the basic application functionality. This should give developers a good idea of how the application should work. Next, the document will over the specific functional requirements that will need to be met to ensure that the application is working according to the standards discussed.

## 4 Overall Description

### 4.1 Product Overview

This product will be web based only that can be accessed by a PC. The web page will be connected to a database that will store important information. Admin will be the only individuals can access this database and users will be the ones to provide the information that will be stored in the database. Admin will operate on a separate server from the users.

### 4.2 Product Functionality

Users will be able to sign up and login in once registers. They will also be able to search a plethora of chefs and their profiles. A food ordering system will allow the users to purchase meals from the listed chefs.

- Login
- Sign up
- Edit Profile (add meals, change profile picture)
- Ordering system (cart, card information for purchase)
- Search meals/chefs
- Leave reviews for chefs

### 4.3 Design and Implementation Constraints

Possible limitations

- Interfacing with the user's social media accounts
- Security for user's private information (passwords, banking information)
- Updating software after it is completed

## 5 Specific Requirements

### 5.1 External Interface Requirements

#### 5.1.1 User Interfaces

*The User Interfaces Include Home Page, Food Listing Page, Food Item Page, Profile Page, Dashboard, Checkout Page, Message System Page, etc.*

#### 5.1.2 Hardware Interfaces

The hardware required for this web application is an IOT device that can connect to WiFi and display html.

#### 5.1.3 Software Interfaces

The web application would interface with the MongoDB database to have users submit requests for foods and to have sellers submit their potential food options. The MongoDB database would store user and seller information to thoroughly process submissions and requests.

### 5.2 Requirements

The requirements for the system entail three different types of users interacting with the software with different features for each user.

#### 5.2.1 User Privileges

##### **Seller:**

- Can edit their availability
- Can post/delete pictures of their food
- Can edit food prices
- Can view who leaves reviews on their service
- Can review buyers
- Can interact with buyers through chat system
- Can categorize the type of food being sold

##### **Buyer:**

- Can review chefs and their food
- Can order food from chefs
- Can communicate with chefs via a chat system
- Can cancel their order
- Can search for foods and active sellers available on the website
- Can see the reviews of other customers
- Can select food based on categories (Chinese, healthy, price, etc.)

**Administrator:**

- Can approve/deny sellers application
- Can approve dishes sold by sellers
- Can ban users
- Can add/remove locations

## **5.3 User Interface Requirements**

### **3.1.1 ID: UIR1:**

#### **Home Page (All listings)**

Rationale: The user/seller/admin needs to access the home page

Dependency: N/A

### **3.1.2 ID: UIR2:**

#### **Seller or Buyer Registration**

Description: When a guest enters the website, he can register as a seller or buyer.

Rationale: In order for the seller/buyer to register and create an account on the website.

Dependency: UR1

### **3.1.3 ID: UIR3:**

#### **Seller or Buyer or Admin Login**

Description: Once the seller or buyer has registered, they can login to their account through a login interface. This login interface lists a username and password input field for the user to input. A button to submit the required fields should also be displayed below the input fields.

Rationale: In order for the seller or buyer login to use the website

Dependency: UR2

### **3.1.4 ID: UIR4:**

#### **Seller Profile**

Description: Once the seller profile is registered, the seller should be able to view his or her profile. The seller's profile view consists of data fields and registered data. A button would be displayed for the seller to edit their profile, and another button would be displayed to allow seller to submit their changes.

Rationale: In order for the seller to view their profile dashboard.

Dependency: UR3

### **3.1.5 ID: UIR5:**

#### **Buyer Profile**

Description: Once the buyer profile is registered, the buyer should be able to view his or her profile. This profile would include details such as credit card information, name, address, email address, contact info, and user settings.

Rationale: In order for the buyer to view their profile dashboard.

Dependency: UR3

3.1.6 ID UIR6:

#### **Admin Dashboard**

Description: The administrator of the site should be able to access a dashboard that has full admin privileges for modifying site data. These data include seller profile and buyer profiles.

Rationale: In order for the admin to access his site data editing privileges.

Dependency: UIR3

3.1.7 ID: UIR7:

#### **Seller Dashboard**

Description: The seller account has a dashboard that allows access to key features such as adding dishes, processing orders, and modifying previous dishes.

Rationale: In order for the seller to view and modify their dish listings.

Dependency: UIR 3

3.1.8 ID: UIR8:

#### **Food Search and Listing**

Description: A viewer of the website should be able search and look around for food listings available. The viewer would be able to see the food image, the chef profile, and the food name listed.

Rationale: In order to allow viewers of the website to view what the site offers in terms of foods available.

Dependency: UR1

3.1.9 ID: UIR9:

#### **Food Item Details**

Description: A seller, buyer, or admin can view the food item posted. Depending on the logged in user, different functions are available for the viewing user. A seller or an admin would have the option to modify the food details, while a buyer or viewer would only be able to put the item in their cart.

Rationale: In order to allow users access to functions pertaining the food.

Dependency: UIR1

## **5.4 Functional Requirements:**

3.2.1 ID: FR1:

#### **Home Page (All listings)**

Rationale: The user/seller/admin needs to access the home page

Dependency: N/A

3.2.2 ID: FR2:

#### **Seller or Buyer Registration**

Description: When a guest enters the website, he can register as a seller or buyer.

Rationale: In order for the seller/buyer to register and create an account in the website.

Dependency: FR1

3.2.3 ID: FR3:

**Seller or Buyer or Admin Login**

Description: Once the seller or buyer has registered, their credentials such as username and password are stored in the MongoDB database so they can login to the website subsequently with their registered credentials.

Rationale: In order for the seller or buyer login to use the website

Dependency: FR2

3.2.4: ID: FR4:

**Forgot password**

Description: In case the seller or buyer has lost access to their account and want to reset their login password in the database.

Rationale: In order for the seller or buyer to reset their password to access their account.

Dependency: FR3

3.2.5: ID: FR5:

**Admin Panel**

Description: When an administrator is setup and users are created, the Admin Panel will allow the user to approve and deny dishes, approve or deny sellers applying to be on the website, can ban buyers and sellers, and can add or remove geo locations on the website.

Rationale: In order to maintain the website and have sense of order.

Dependency: FR1

3.2.6: ID: FR6:

**Food Search**

Description: A viewer of the website should be able search and look around for food listings available.

Rationale: In order to allow viewers of the website to view what the site offers in terms of foods available.

Dependency: FR1

3.2.7: ID:FR7:

**Order Food**

Description: A user who is already registered and signed in should be able to order food.

Rationale: In order to allow buyers to request food.

Dependency: FR3

3.2.8: ID: FR8:

**Adding Dishes**

Description: Allows a seller to add dishes to their profile, by mentioning the dish name, uploading a picture, price and location.

Rationale: In order for a seller to list their dishes on the website

Dependency: FR3



3.2.10: ID: FR11:

**Image Upload**

Description: The seller has to upload images of the dishes so the buyers can view the food items before purchasing on our website.

Rationale: In order for the user to sell their dishes, they should be able upload images of their dishes.

Dependency: FR8

3.2.9: ID:FR9:

**Chat System**

Description: Allows the user and seller to interact with each other.

Rationale: To discuss any specific requests regarding orders (e.g., allergies, etc.).

Dependency:

3.2.12: ID: FR12: Food Review

## 6 Other Non-functional Requirements

### 6.1 Performance Requirements

This is a web application, so the requirement is a browser, preferably Google Chrome, because you all testing and front-end development was done on that. However, this is a responsive web-design so it can be used on mobile or desktop environments. The IDE we used is Visual Studio Code version (). Software requirements are

- MongoDB (v 5.10.9)
- Express (v 4.17.1)
- Node js (v 12.18)
- React (v 16.13.1)
- AdminBro (v 3.2.5)
- Bootstrap (v 4.5.2)

We want to make the user experience as smooth as possible and that's why we are compressing our images to .webp in order to make sure the website loads the highest quality images with less time. We are also using the feature of React Router so that saves time of fetching additional files from the server side and instead just loads the component in that page directly. This will not only make the access faster but will also make it look like you're using an app rather than a website. The idea here is to make the user experience as smooth as possible.

### 6.2 Safety and Security Requirements

Since ours is a food app, the sellers must have a business license in order to sell their food and also follow the food safety and quality protocols set by the government and Food and Drug Administration (FDA).

We are planning to integrate login with Google or social media accounts to add an extra layer of security to the user and only that have been verified by these services can sign up/in to our application.

## 7 Prioritization

Week	Front-end Goal	Back-end Goal
1-2	-logo -color theme	-set a price to seller's food
2-3	-Buyer login page -Seller Login page	- Buyer account -personal info - food order system - search engine for food and other users
4-5	-Homepage (Nav bar + Image Carousel) -footer(privacy policy, small description,etc)	-admin account -implement a way to see all users
6-7	-How it works Page -Filter	-admin(cont) -approve dishes -ban users** -add/remove locations -approve applications (login requests)
8-9	- Meals Page (product display cards with price) -Buyer profile page	-Query Data to display.
10-11	- Admin Panel (dashboard that shows how many sellers/users, approval pending, etc.) - review page	-Implement CRUD Features for admin profile -Implement a new data type for the database to handle(Review).
12-13	- Contact us page (Form with fields for name, contact info, etc.) -not found page	-Login authentication -image upload
14-15	- Host Page (shows cards of different sellers with the name, dishes, and info about user, pictures, etc. -privacy policy page	-review system -chat system -payment system