

# MyKitchen

## SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

CS 4850 Senior Project

Spring 2024

Professor Perry

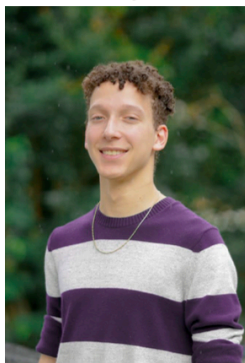
2/9/24

Indy-4 Kitchen

Merrick



Nick



Minchul



## Table of Contents

1. Introduction.....	3
1.1 Overview.....	3
1.2 Project Scope .....	3
1.3 Definitions and Acronyms.....	3
1.4 Assumptions .....	3
2.0 Design Constraints .....	4
2.1 Software/Hardware Environment.....	4
2.2 User Characteristics .....	4
3.0 Functional Requirements.....	4
R.1.0 User Registration and Authentication .....	4
R.2.0 User Profile Management .....	4
R.3.0 Payment Processing.....	4
R.4.0 Navigation and Webpage Flow .....	4
R.5.0 User Location Services.....	4
R.6.0 Kitchen Profile Creation and Management .....	4
4.0 Non-Functional Requirements.....	5
NFR.1.0 Performance.....	5
NFR.2.0 Screen Scalability .....	5
NFR.3.0 Security .....	5
NFR.4.0 Usability .....	5
NFR.5.0 Maintainability .....	5
5.0 External Interface Requirements .....	5
5.1 User Interface Requirements.....	5
5.2 Software Interface Requirements .....	5

# 1. Introduction

## 1.1 Overview

This document is created to give clear and concise software requirements for the MyKitchen project. This document contains requirements gathered from the developers of the project and organized by the project lead.

## 1.2 Project Scope

The MyKitchen project is to turn home kitchens into ghost kitchens through creating a platform developed using a robust and scalable javascript framework React and utilizing PHP for efficient back-end server-side operations. Whereby users data is managed through a MySQL database, ensuring secure and structured storage of Kitchen profiles, user information, menu details, and transaction records. Hosting will be done on AWS guaranteeing reliability and scalability, allowing the platform to handle a growing user demand and expanding service areas. Processing payments will be done using a third party software stripe.

Aligned with broader business objectives, MyKitchen aims to not only offer a platform for culinary exchange but also to cultivate a community around local food culture, promoting sustainability and supporting small-scale kitchen operations. This initiative contributes to the overarching goal of creating more inclusive, diversified, and community-focused food ecosystems.

## 1.3 Definitions and Acronyms

AWS- Amazon Web Service: it is a cloud service that we will use to house our data and website.

## 1.4 Assumptions

For this project many assumptions are being made on the development process and users machines.

Assumptions for users:

1. Accessing the website through search browsers such as Google, Firefox, and Safari.
2. User machines will have at least 8GB of memory and hardware that is at most 20 years old (since 2004)
3. Users have used other food ordering services and have access to a phone number, email, wifi, and have a method of payment whether that be cash or card.

Assumptions for project:

1. Technology being used for the project is current and is at most 10 years old.
2. Password security will be kept using encryption methods.
3. The site will be able to display within a short amount of time (3-10 seconds for example)

## 2.0 Design Constraints

### 2.1 Software/Hardware Environment

An constraint that could occur during the development of the project is the hosting platform AWS, trying to remain on the free tier will be the goal for the project but services that the project would like to explore would constrain our budget. Another issue that could occur with hosting is the probability that AWS will experience outages. This concern is not as great but could result in longer development and future problems.

The project relies on using stripe as a method of payment. Using stripe might present difficulties and paywalls that might effect the development of the project. This concern could cause development to be delayed to figure out methods of payment.

### 2.2 User Characteristics

A concern in the project would be the lack of kitchen sign ups, in other words having a common problem that many platforms suffer from by having too many customers to service providers. As the project is for a senior project this concern is minor, but if it were to expand into a great monetary platform it could result in stunting the growth of the platform. A solution to the problem would be to reach out and find ghost kitchen owners to sign up or to release marketing ads for the platform.

## 3.0 Functional Requirements

### R.1.0 User Registration and Authentication

- **R.1.1** Access to registration page for users without an account.
- **R.1.2** Registration requires username, email, password, and address.
- **R.1.3** Password encryption for user security.
- **R.1.4** Login with email and password for returning users.

### R.2.0 User Profile Management

- **R.2.1** Users can view and access their profile post-login.
- **R.2.2** Users can edit personal information within their profile, including password and address.

### R.3.0 Payment Processing

- **R.3.1** Integration of Stripe API for handling credit/debit card transactions.
- **R.3.2** Secure transaction processing compliant with PCI DSS standards.

### R.4.0 Navigation and Webpage Flow

- **R.4.1** Homepage serves as the entry point for all users.
- **R.4.2** Navigation from the homepage to the login page for authentication.
- **R.4.3** Post-login redirection to user-specific activities and profile.
- **R.4.4** Cart review functionality available prior to the payment process.

### R.5.0 User Location Services

- **R.5.1** Mandatory address input for user profile completion.
- **R.5.2** Utilization of user address to display local kitchens within a 20-30 mile radius.

### R.6.0 Kitchen Profile Creation and Management

- **R.6.1** Users can initiate kitchen profiles through their user profile.

- **R.6.2** Kitchen profiles allow for the addition and editing of dish images, descriptions, pricing, and operational hours.
- **R.6.3** Capability for users to modify the name of their kitchen profile.

## 4.0 Non-Functional Requirements

### NFR.1.0 Performance

- **NFR.1.1** Page Load Time: The website must load within 3 seconds using standard broadband connections.
- **NFR.1.2** Concurrent Users: Support for up to 1,000 simultaneous users without significant performance degradation.

### NFR.2.0 Screen Scalability

- **NFR.2.1** Responsive Design: The website must offer optimal viewing and interaction experience across all devices, adapting layout and resolution to match varying screen sizes.

### NFR.3.0 Security

- **NFR.3.1** Data Encryption: Implement standard security practices, including encryption of sensitive data, to protect user information.
- **NFR.3.2** Security Best Practices: Adherence to industry-standard security protocols to safeguard the platform against unauthorized access and data breaches.

### NFR.4.0 Usability

- **NFR.4.1** Intuitive Interface: The website should feature an intuitive and user-friendly interface that aligns with current UX/UI design best practices.
- **NFR.4.2** Cross-Platform Compatibility: Ensure the website is compatible across various devices and browsers, including desktops, mobile phones, and tablets.

### NFR.5.0 Maintainability

- **NFR.5.1** Code Documentation: The codebase should be clearly documented, allowing for ease of maintenance and the ability for new developers to effectively continue development efforts.

## 5.0 External Interface Requirements

### 5.1 User Interface Requirements

The overall look to the website should accompany three-four colors as a theme. Icons should be favored over words, using the least amount of language as possible to accompany those who do not speak English.

### 5.2 Software Interface Requirements

*The website will use Stripe for processing payments and will use AWS ECO2 for hosting the website and database.*