CSCE 361 Software Engineering
Software Requirements Specification

Recipe Search Web-Application

Team 14
Marek Kracl
Mac Ozanne
Linlin Chen
Qiming Yuan
Crystal Warta

Version: 1.1 **Date: 9 FEB 2018**

Table of Contents

1. Introduction

This document describes the implementation of a web-based recipe-searching application. This system will be created using a combination of HTML/CSS, node.js, react.js, MySQL and Java.

The application will be implemented in two main phases. In the first phase, a database will be populated with 100 or so sample recipes, curated to allow easier searching, and tags and keywords added. We will also implement a simple GUI and allow for the user to **input a list of their on-hand ingredients**. During this phase, the user will be able to add their on-hand items to a list and search for recipes that contain only those items.

The second phase will focus on fleshing out the GUI elements into a visually appealing web-app, adding the shopping list functionality where a user can select a recipe they do not have all the required ingredients for and the remaining ingredients will be added to their shopping list. We may also add user login functionality and personal data persistence at this phase.

1.1 Purpose

This document is intended to outline the design and implementation of the Recipe Search Web-Application. It provides information on the creation of objects based on their real-world counterparts and the use of object-oriented programming techniques to design the system. Additionally, this document describes the design of the related database and connectivity between the HTML, Javascript front-end, and the database.

1.2 Scope

The objective of this application is to provide users with the ability to easily add on-hand food ingredients to a list, and query a database; returning recipes that include only those ingredients. This will allow users to create a meal using only ingredients they have available. As part of the Agile phase, the application will be extended to include a shopping list and to include functionality for querying recipes that mostly fit the listed ingredients; with missing items added to a shopping list automatically.

1.3 Definitions, Acronyms, and Abbreviations.

Agile - a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
Ant Design - A design language for application, and React.js is used to encapsulate a library of Ant Design component.
ER Diagram - a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system
JSON - JavaScript Object Notation
MySQL - An open source relational database management system (RDBMS) based on Structured Query Language (SQL).
OOP - A programming language model organized around objects rather than "actions and data rather than logic.
React.js - A javascript library for building user interfaces.
XStream - a simple library to serialize objects to XML and back again.

1.4 References

AntDesign - *https://ant.design/docs/spec/introduce*
Bootstrap - *http://getbootstrap.com/docs/4.0/getting-started/introduction/*
ER Diagram - https://www.lucidchart.com/pages/er-diagrams
MySQL - *https://dev.mysql.com/doc/*
Node.js - *https://nodejs.org/en/docs/*
React JS - *https://reactjs.org/docs/hello-world.html*
XStream - *http://x-stream.github.io/*

1.5 Overview

The remainder of this document contains more precise and in-depth information as they pertain to a) potential users and b) developers of this and future systems that may utilize functionality of this system.

Section Two (2) contains information relevant to potential users; high-level descriptions of interfaces and memory requirements for example.

Section Three (3) contains information relevant to developers, both of this system and of systems and applications that may interface with this one.

2. The Overall Description

2.1 Product Perspective

The application is a web-based application which will provide the user with an interface to add any ingredients they may have in their refrigerator or cupboard to a list. When the list is complete, the user will press submit and be presented with a new list of all the possible recipes that include only the ingredients the user listed; complete with a description, the steps involved in preparation, estimated time required, and an image of the completed dish.
During the Agile phase, the user will be able to elect to search for recipes that match some of their on-hand ingredients and the missing ingredients will be added to a shopping list for the user automatically.

2.1.1 System Interfaces

The following sections describe the interfaces required.

2.1.2 Interfaces

The user will access the application through a standard web page built in HTML. The website contain a list of items and a matching checkbox for each. At the bottom of the page, the user will see a 'submit' button which will be used to query the database for the list of checked items.
In the Agile phase, the user will also have the ability to select an option to either match **all** of the items checked, or **some** of the items before submitting.

2.1.3 Hardware Interfaces

Any modern device with a web browser will be supported.

2.1.4 Software Interfaces

Node.js will be used to communicate between the HTML web-page and the MySQL database. Node has this functionality built in.

2.1.5 Communications Interfaces

No communication interfaces are required.

2.1.6 Memory Constraints

This application will be very lightweight and should not have significant or even noticeable effects on memory of the user's system or the server.

2.1.7 Operations

The user will be able to login with their username and password.
The user will interface with the application via checking boxes corresponding to their on-hand ingredients and pressing submit.

2.1.8 Site Adaptation Requirements

A MySQL database must be created for this system to function properly. For this project the database will be populated manually using a script and be accessible on the cse server.

2.2 Product Functions

This app's function is to aid the conveniency of the user in meal and grocery shopping choices.

2.3 User Characteristics

The intended users of this product will be an adult audience who shop and/or cook for themselves and/or their family. The audience does not need to have cooking expertise in order to use the app as ingredients and cooking instructions are included. The use of the app will be fairly straight-forward so there is no need for a tutorial and anyone with internet access can use it.

2.4 Constraints

In order to provide the user with the most accurate results, the input will be selected from a predetermined list of acceptable ingredients; rather than allowing the user to type anything into a search bar, for instance.

2.5 Assumptions and Dependencies

We are assuming the user has a modern computer/tablet or phone and connection to the internet.

2.6 Apportioning of Requirements.

The initial build will include only functionality for the user to select on-hand ingredients, and submit those requirements. The user will then be presented with a list of recipes that only contain those on-hand ingredients.
For the Agile phase, the user will have the option to search for additional recipes that contain ingredients that they do not have on hand. Those recipes will be returned and the missing ingredients will be added to a shopping list.

3. Specific Requirements

3.1 External Interfaces

Ingredient - added by the user via an HTML drop-down menu or checklist (so as to ensure sanitized data) and passed via Javascript to query the database for Recipes that match the given Ingredients.

Recipe - After a recipe or recipes matches from the database, the information will be returned to the user via Javascript and displayed with HTML.

3.2 Functions

Initial Release:

3.2.1. The system shall prompt users to select on-hand ingredients from a drop-down menu or checklist and add it to a list of items to be queried.
3.2.2. The system shall pass the list of items to the node.js script.
3.2.3. The node.js script shall query the MySQL database for all of the listed items in one recipe, populating a list of Recipe information that match the chosen Ingredients and returning them to be displayed in the web page via HTML.

Agile Release:

3.2.5. The functionality described above in 3.2.3. will be expanded to include Recipes that match **some** of the given on-hand ingredients.
3.2.6. The functionality described above in 3.2.4. will be expanded to include:

- The system shall first display any Recipes that match all Ingredients, followed by those Recipes matching some ingredients.
- The system shall then prompt the user to ask if they would like to add the Ingredients **not** in their on-hand list to a shopping list.
- The system will then query the database using the unique identifier of the user-selected Recipe and return a list of non-matching Ingredients.
- The system shall populate a shopping list for the user and display it on screen via HTML.

3.3 Performance Requirements

3.3.1. The system shall handle up to 100 simultaneous users.
3.3.2. The system shall return relevant Recipes to the user and display them in 3 seconds or less.

3.4 Logical Database Requirements

The database shall contain information on approximately 100 Recipes. This information shall contain the following:

- Title
- Description
- Instructions
- List of Ingredients
- Prep Time/Cook Time
- An image of the prepared meal

3.5 Design Constraints

The main design constraints is limiting the length that user may enter into fill in a blank form. An example for this would be request of username, password, search keyword.

3.5.1 Standards Compliance

There are none.

3.6 Software System Attributes

3.6.1 Reliability

The system should be reliable enough that one can use this web application without any errors and the database and their account updates correctly whenever changes are made.

3.6.2 Availability

This system should be able to be accessed by a user at all times using the internet.

3.6.3 Security

The only security features of the system is to protect the user's account by requiring a password during login, so this should successfully keep the account from being accessed by others.
Since the data is curated, it will be impossible for an SQL injection to occur.

3.6.4 Maintainability

Since the system will not have admin access, any administrator must manually enter recipes into the database via scripting.

3.6.5 Portability

Since the application will run on any system with internet access, portability will not be an issue. The application will run natively on a very large cross-section (near 100%) of all available machines.

4. Change Management Process

Changes will be submitted by a member or members of the development team and need to be approved by a simple majority of the development team.

5. Document Approvals

Approved by:	**Name**	**Signature**	**Date**
	Marek Kracl	_Marek Kracl_____	_02/09/2018_____
	Mac Ozanne	_Mac Ozann_____	_02/09/2018_____
	Linlin Chen	_Linlin Chen_____	_02/09/2018_____
	Qiming Yuan	_Qiming Yuan_____	_02/09/2018_____
	Crystal Warta	_Crystal Warta_____	_02/09/2018_____

6. Supporting Information