**Recipe Recommendations SRS**

1. Introduction

1.1 Purpose

This document describes the requirements for the design of a recipe recommendation system. These requirements are described in terms of agile-style use cases.

1.2 Background

Most people are not born with the ability to cook. Indeed, many people in the United States are brought up with the luxury of having their parents make their dinners for them. While this is lovely in itself, it also means many people enter college with questionable cooking skills. Unfortunately, many come to realize that the habit of eating out may not be justifiable in light of the cost of doing so. So instead of eating out, many eat in.

But what’s to be made, that isn’t ramen (again)? This is the question that motivated this project.

2. System Requirements

2.1 Actor(s)

College student(s) who want to make their own meals, but who don’t quite know how to combine what food they have into a tasty meal.

2.2 High Level Flow

When the user first visits the webpage, the user is asked to login to their account - or create one if they don’t have one already. After this, the user is redirected to a homepage which is prepopulated with recipes. The recipes displayed are tailed to the user, if the user has a history of liking\disliking recipes and\or categories of recipes. If not, then the homepage is prepopulated with the most popular recipes of all time. From the homepage, the user can filter the recipes by ingredients (by exact matches and contains), and can sort the recipes (in a variety of ways). Alternatively, the user is free to simply browse at their leisure.

2.2 Basic User Cases

2.2.1 User interactivity

2.2.1.1 User connects to website

2.2.1.2 User enters which ingredients they have

2.2.1.3.1 System assists the user by offering completion suggestions

2.2.1.3 User selects various options such as time, difficulty, allergies, disliked foods, etc

2.2.1.4 User submits query (the list of ingredients and the options selected in use case 2.2.1.3)

2.2.1.5 System searches database for recipes matching query

2.2.1.6 System returns results from database

2.2.1.7 User is then able to view a list of recipes that match their criteria and click on the recipe they’d like to view

2.2.1.8 If the user would like to modify the search or start over they are able to click a button to do so

2.2.1.9 User is able to copy, paste, and save recipes after clicking on desired recipe

2.3.3 Database Use Cases

2.3.3.1 Database is accessed via system backend.

2.3.3.2 The database is keyed by all of the ingredients in each recipe in addition to the name of the recipe

2.3.3.3 The database receives user input ingredients and searches for asociated recipies, one key at a time

2.3.3.4 A search is conducted throughout the database for all user specified ingredients.

2.3.3.5 A list of recipies is accumulated that correspond to the user input.

2.3.3.6 The database returns list of recipies to user.

Backend

-The frontend is able to communicate with the backend via routes

-In reverse, the backend is able to communicate with the frontend via routes

-The backend serves as an access layer to the database. In other words, the frontend first requests content from the backend, which in turn forwards the request to the database. The backend receives the results from the database and then sends them to the frontend.

-The backend performs some verification of information entered by the user.

2.3.2 Recommendation use cases

2.3.2.1 The system recommends recipes on the basis of the user’s history of recipes the user has indicated they like.

The system can also recommend recipes if the user is missing a (non-key) ingredient. (Ex: a salad recipe requires a certain type of dressing, but the user might have another type of dressing already).

2.3.2.1.1 The system considers these features: category (ie: vegan, paleo), time to make, and the base ingredients

2.3.2.1.2 The system filters away results containing ingredients which the user has indicated they are allergic to or simply dislike

2.3.2.1.3 The system requests the full set of recipes from the database at app launch`

2.3.2.1.3 The system requests “You might also like/be able to cook these recipes” and returns recipes that are only missing a couple of non-key ingredients.

[Possible replacement for the use cases above]

-The user connects to and loads the website

-The user is asked to sign into their account

* If the user does not have an account, the user is asked to create an account.
* 3

-The system redirects the user to the main recipe page.

Recipe population

-If the user has a history of liking recipes and\or categories of food, the recipes shown in the recipe page will be tailored to the user based on this history.

-If the user does not have such a history, the user will instead be presented with the overall most popular recipes.

Browsing

-The user is free to browse around the website as they wish at this point. There are no further mandatory dialogs\information that the user need provide.

Filtering

-The user can filter the list of recipes to include only those recipes that **contain** a user-specified list of ingredients. If any of the ingredients that the user provides are not contained in at least one recipe, a warning message is displayed to the user to alert the user and that ingredient is ignored.

-The user can also filter the list of recipes to include only those which have **exactly** a user-specified list of ingredients.

-The user can filter the list of recipes to include only those which contain a user-specified list of ingredients in addition to 1 or 2 missing ingredients.

Searching

-The user can search for recipes which contain a user-specified list of ingredients.

-The user can search for recipes which contain exactly a user-specified list of ingredients. Ie: they contain the given ingredients and no other ingredients.

-The user can search for recipes which contain a user-specified list of ingredients in addition to 1 or 2 additional ingredients.

Sorting

-The user can sort the recipes by the time required to make them

-The user can sort the recipes by rating

-The user can sort the recipes by estimated cost

Random recipe populations

-The user can replace the existing list of recipes with a user-specified number of random recipes.

Liking

The user can press a button to indicate they like a recipe. A record of their liking this recipe is saved in the user’s profile.

-The user can like a whole category of food (ie: they can indicate they like vegan food in general)

User Profile

-The user can change the information the system has stored about that user, including their email address, username, and password, and profile picture.

-The user can delete their account

-The user can optionally choose that the system retain its records of the user. Otherwise, by default, all user records are deleted from the system.

-The user can delete their system history. This means that the user can delete all system records of which recipes they liked, which categories of foods they liked, and any other information that the system has stored which pertains to that user except their username, password, and profile image. To remove the last three entirely, the user must delete their account. [nice to have feature]

Social media [this whole category is a nice to have]

-The user can like a recipe, and have a message along the lines “I liked reciepe x” appear on their facebook page.