**Technical Specifications Documentation**

**Group D**

**Table of Contents**

1. Introduction
2. Purpose
3. Scope
4. API’s
5. System Architecture
6. Functional Requirements
7. User Dialogs & Control Flow
8. Background Tasks
9. Database Models
10. Interfaces to Other Systems
11. Nonfunctional Requirements
12. Hardware Requirements
13. Terms
14. Execution Planning

**Introduction**

The purpose of this document is to collect, analyze, and define high-level needs and features of Foodex. It focuses on the capabilities needed by the stakeholders, and the target users, and why these needs exist. It also list the necessary hardware needed to properly utilize this software application. The document will further explain some of the technical side of the inner workings of this program.

**Purpose**

Grocery indexing and suggestive application based on what the users have bought, including packaged and fresh food. This application will suggest easy everyday recipes to help to create an efficient and healthy lifestyle.

**Scope**

Foodex — an application developed and intended for mobile devices — is capable of sending notification and social media sharing. The early development process involves using APIs, algorithms, and databases, which will work together to generate a list of food and recipes based on user input.

**API’s**

1. The Spoonacular API is providing the food item’s nutritional data, price, amounts and conversion, cooking tips, and health information. In addition, it will provide the recipe’s nutrition analysis, cost breakdowns, cook tips, related recipes, scaling and converting, semantic search, and ingredient to product mapping. This API will provide the data our team requires to interpret the searches the user enters into meaningful and valuable data.
   1. Spoonacular API Link — <https://spoonacular.com/food-api>
2. USDA Food Database API has detailed description and expiration day values for food lifespan which will be useful for our sorting algorithm that will base mostly on the current date and the number of day increment for food expiration. This will allow the Foodex users to be notified of their expiring food.
   1. USDA Food Database — <https://ndb.nal.usda.gov/ndb/api/doc>
3. The Free OCR API will be used as a tool to take picture of grocery receipts and convert that into letters in order to use as data for abbreviation algorithm. This will allow the Foodex user to take advantage of the camera feature to convert their receipt into food items within the application.

a. Free OCR API link — <https://ocr.space/OCRAPI>

1. Twitter, Facebook, and Instagram will provide us with the tool to allow Foodex users to share pictures of recipes to social media accounts.
2. Twitter API Link- https://dev.twitter.com/overview/documentation
3. Facebook API Link - https://developers.facebook.com/docs/sharing/android
4. Instagram API Link- https://www.instagram.com/developer/

**System Architecture**

The system will contain APIs to connect to popular social media, algorithms to both using merge sort and binary search food, and MySQL databases to store information on food and recipes. They will all eventually connect and communicate with one another. The program will connect to the database using a JDBC driver to connect to the fully implemented class Driver Manager. Connecting to social media API’s will require us to import The APIs and databases will be accessed first from the user input, and then the algorithms will take place next. The algorithms will convert the text on pictures into text and match them with data that was imported through The Spoonacular API to display the recommended recipes to the users.

**Functional Requirements**

The functional requirements are to decipher text from a picture, take user input, read and pull up relations from a database, recommend relative recipes, read data on food listed from on receipt, send push notifications to the user, and share to social media. The user can then add more inputs to narrow or expand the list of recipes, delete certain keywords from their search in the event of a mistake.

Special needs:

How to make database: internet connection, type, server

* Using the MYSQL environment to create an open source relational database.
* Internet connection will utilize a phone’s sensor for wifi or cellular data
* Have a server that holds the database, information of all the food and recipes, and account information of the users. The user will then use their app to connect to the server and retrieve the information they want.

What language and environment to use,

* Coded in Java and implemented through Android Studios

What are the hardware to implement and OS to run on?

* Hardware that runs this application is a smart phone primarily using one of Android’s OS if user has an Android smart phone.

**User Dialogs & Control Flow**

* 1. Input food by typing keyboard or take picture of receipt and having the text recognition from Free OCR API scan the receipt to get our input.
  2. Food will be sorted using merge sort based on the expiration date received from the USDA Food database API on the order of whichever food expiration date approaches first to which food expires last.
  3. Binary Search algorithms take place to find recipes related within the database using the Spoonacular API to match the food the users input to the recipes.

**Background Tasks**

The background tasks are the databases and APIs being utilized, pictures will be converted and deciphered using OCR text recognition from The Free OCR API, food will be listed by the order of their expiration date receive from USDA Food database API, recipes will be determined by the recipe selection algorithm that matches the recipe to the foods through the Spoonacular API, push notifications will be sent more aggressively when application is infrequently used, and the transition from the picture taken to social media using either Twitter’s, Facebook’s, or Instagram’s API.

**Database Model**

* + Food
    - Keys: Name, nutrition, expiration date
  + Recipe
    - Keys: Ingredients, nutrition
  + Social Media
    - Keys: User account, picture

**Interfaces to Other Systems**

We will implement these interfaces that give the users the option to share to social media once they have chosen a recipe they want. This will allow them to share pictures of the recipe from the app to social media using one of the social API’s listed below.

* + Facebook, Twitter, Instagram, Snapchat
    1. Interfaces to connect to social media to share picture of used recipe

**Nonfunctional Requirements**

* + User interface must look presentable
    1. Straightforward, intuitive, and easily understood
  + Application must run fast, reliable, efficient, and provide a seamless experience
  + Can run on Android
  + Login security should be encrypted for safety using a cipher such as a running key algorithm.
  + Requires internet access

**System Requirements**

* Smartphone from Android.
* Desktop PC

**Terms**

1. Database: Structured set of data
2. API (Application Programming Interface): Set of tools for building software applications, how software applications should interact
3. OS: Operating System
4. Social Media: An Internet-required medium for users to create and share content

**Execution Planning**

Involved APIs in the Process Include:

* 1. 2 weeks
     1. Cam scan OCR for reading receipt images for text conversion.
     2. Food index and expiration from date USDA food for the database.
     3. Nutritional value/ recipes apps such as Food Networks will be used for recipes suggestion.
* The API’s will be executed according to the order listed within 2 weeks, from 01/25- 02/08.

Database: 3 weeks

* Relational database between food name and date will be generated using campus issued server. There will be at least 3 tables: food name, expiration dates, and recipes according to each kind of food. The time span for this part of the project will be 3 weeks, from 02/09-03/03, where tables and attributes will take 2 weeks and 1 week for the relations.

*\*\*Test generation will be integrated with the database during implementation and overall testing will take 2 days from 03/03-03/05\*\**

Algorithm: 5 weeks

*\*\*We will also be implementing many algorithms for the app to be able to run with APIs’ data.\*\**

* 1. Solving abbreviation algorithm for converting receipt phrases into food names for the listing.
  2. Sorting by dates uses the database of food and dates to generate a list of ascending dates of food.
  3. Search by keywords of food items on top of the list for appropriate recipes.
* The goal is to spend 1.5 weeks for each algorithm, from 03/06-04/10.

*\*\*Testing will occur for each algorithm after the end of implementation and overall testing will take ½ week from 04/10-04/14.\*\**

*\*\*Overall assembly and initial bug fixing will take an additional 1.5 weeks from 04/14-04/25.\*\**

*The last two features are add-ons to the apps in with the consideration of timing and prior progress.*

1. **Notification feature: 1 week**
   * To notify users about food expiration for each morning or through customized settings.
2. **Sharing feature: 1 week**
   * To allow sharing through social medias pictures and recipes made for that day’s suggestion.

* The above 2 features will take 2 weeks in total for implementation with testing occurs separately once assembled into the main project. The duration will be from 04/26-05/10.

*\*\*Overall testing and bug fixing will occur during the last week of May, one week before launch date.\*\**