Documentation Rev1

**Problem Statement:**

Figuring out what to cook for every single meal is always a hassle. Even if you already know a bunch of recipes. You can compile your recipes in bookmark tabs or sticky notes in cookbooks, but having to sift through everything is difficult especially if you’re trying to plan meals for the week.

Planning and cooking meals can take up a lot of time during the week, even if you already know a bunch of recipes. Recipes can be scattered across cookbooks, blogs, YouTube videos, and phone notes. Planning grocery lists around them can also be time consuming as you have to reference what you already have and if there are any overlaps with the recipes.

**Product:**

The product is a software application that will be a one stop shop for compiling, searching, and filtering recipes. You can upload recipes in a uniform format and search for them based on the name, region, ingredients, etc. You can also store a list of what’s currently in the pantry and based on the recipes that are selected, compile a grocery list.

**Process:**

An Agile process will be used to complete this software design project. Agile values working software over comprehensive documentation.

**Detailed Requirements**

Functional

1. Add and store recipes which include:
   1. Ingredients
   2. Description
   3. Instructions
   4. Tags
   5. Pictures
   6. Source
2. Store currently owned ingredients in a ‘pantry’
3. Edit and removal of recipes and pantry items
4. Recipes are searchable by:
   1. Name
   2. Ingredients
   3. Tags
5. Make a ‘grocery list’ based on recipes selected and current items in ‘pantry’

Nonfunctional

1. Easily navigable UI (tabs, scroll bar, search bar)
2. Quick way to import recipes
3. Blog-like viewing page for recipes

**A diagram of a person

Description automatically generatedUse Case UML Diagram**

**Architecture Design:**

Dish Deck is an interactive system and uses a N-tier architecture style.

A blue file folder with black text

Description automatically generated

**Component Design:**

Recipe

Attributes: [Name, Description, Ingredients, Instructions, Tags, Picture, Source]

Methods:

* \_\_str\_\_()

RecipeBook

Attributes: [recipe, number of recipes]

Methods:

* importRecipe(submethod of loadRecipes)
* loadRecipes() #stores the data in a dictionary
* addRecipes()
* removeRecipes()
* editRecipes()
* search(field, query) (uses dictionary)

Pantry

Attributes: [Item, Quantity]

Methods:

* loadItems()
* addItem(item, quantity, unit)
* removeItems()
* editItems()
* search()

Ingredients

Attributes: [name, quantity, unit]

Methods:

-\_\_str\_\_()

MealPlan

Attributes: [recipeBook, pantry]

Methods:

* selectRecipe()
* deselectRecipe()
* generateGrocery()

**A diagram of a recipe

Description automatically generatedUML Class Diagram**