

Grocery List Generator

Problem Statement

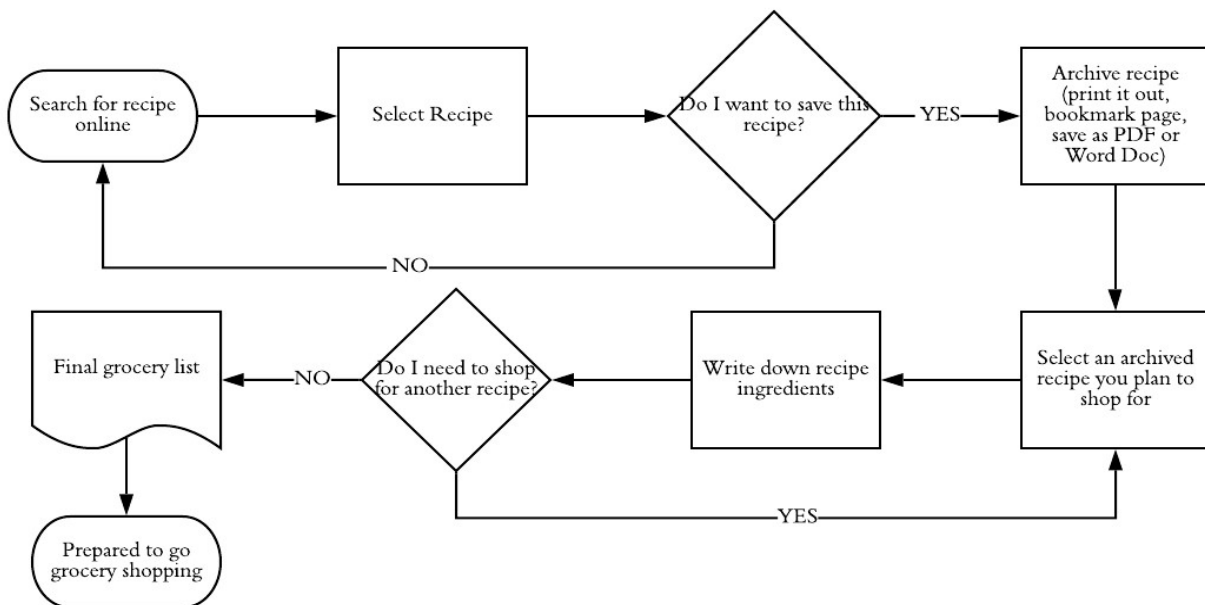
The problem that this system would like to address is the complexity that arises when shopping for multiple recipes. This can be a problem in purchasing an appropriate amount of ingredients for each recipe. The impact this problem can have is either having a surplus of ingredients leading to wastage or a shortage of ingredients either requiring you to not make the recipe you originally planned or make an additional trip to the grocery store.

User Needs

Anybody who cooks on a regular basis is affected by the problem. These people include but are not limited to individuals who cook for themselves and/or friends and family, restaurant staff, and caterers. People who cook a variety of recipes need a simple way to list ingredient amounts across recipes they plan to prepare so that they know the appropriate amount of each ingredient to purchase to minimize ingredient shortages and wastage as well as make their grocery shopping experience more efficient.

Current State Processes

Current State Process Diagram



The **Current State Process Diagram** illustrates the current process people go through in selecting recipes to prepare and then generating the grocery list for these recipes. The main pain points in this process are those related to generating the grocery list such as going through each recipe and writing down the ingredients.

Potential Solutions

One potential solution would be for the user to create a spreadsheet with all their recipes. When selecting recipes to prepare, the user could copy and paste the ingredients onto a separate sheet. This improves the

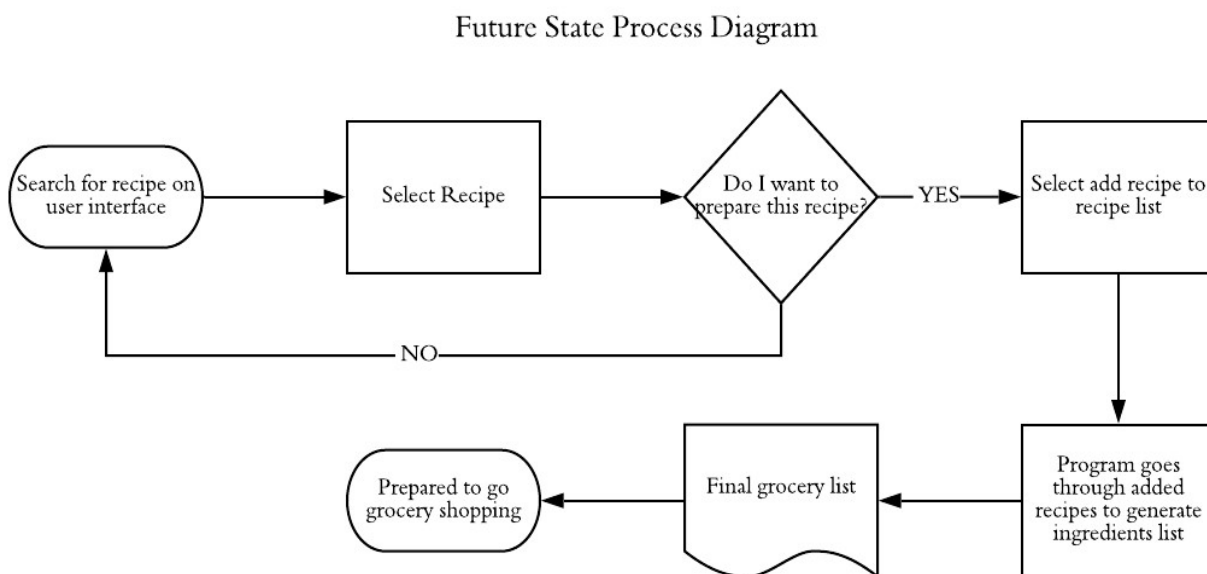
steps of writing and grouping the ingredients. This solution still requires the user to import their recipes to a separate interface, and the user would potentially have to sift through many sheets to find the recipes they wanted to make.

Another potential solution is a user interface that allows the user to search for recipes and add them to a list of recipes to prepare. Once a recipe is added, the ingredients would be added to a grocery list that the user could then access. This option is better than the spreadsheet solution because it requires less work for the user as the searching, saving, and listing capabilities are all in one place and largely done by the program.

Proposed Solution

I am proposing to implement a user interface and system that should allow users to lookup recipes and add them to a list. A list of all the ingredients needed across the recipes from the recipe list should be generated and then made available to the user. This solution simplifies the process of generating a grocery list as the user no longer has to individually go through each recipe and manually record the needed ingredients.

Future State Processes



When compared to the Current State Process Diagram, this future process is much simpler. For the user, it removes the steps in between saving the recipe and having the finalized grocery list. It also reduces room for error in generating the list such as missing an ingredient. Additionally, it provides consistency for the user in that they can reuse the same system prior to each time they go grocery shopping.

System Objectives

The system should generate a grocery list for an end user after they search for and add recipes to a list which is also available to the user on the interface. The search feature and recipe list will be on the user interface, and the grocery list will be presented and then made available to the user in the form of a txt file, making it easy for the user to access and use all parts of the system.

Functionality Requirements

Recipe Search

The first sub-component of the system is the recipe search. The user interface will allow the user to input a recipe name or keyword into an entry box. This input will then be read into the API URL in the code to retrieve the JSON data for the recipes returned from the search. A list of recipe names will be generated and then presented to the user in a ListBox on the interface.

Recipe Selection

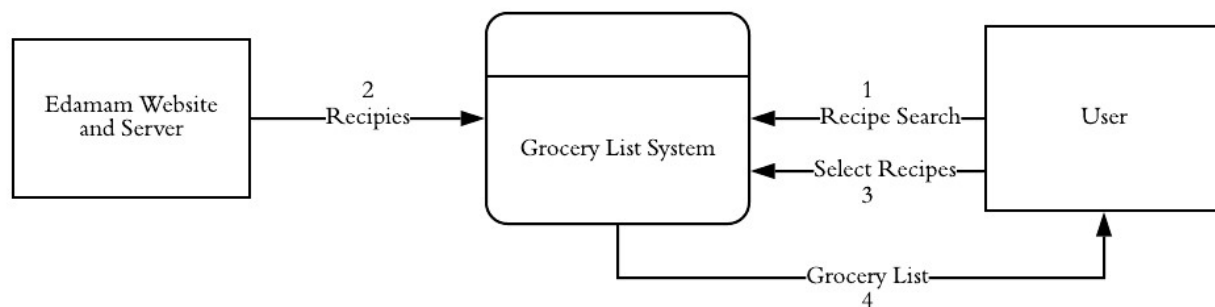
From the recipes in the ListBox, the user will be able to select one or more, and then click the add recipes button (See Interface Requirements for visualization). Once this button is clicked, lists will be appended, one with the selected recipe name and one with its ingredients.

List Generation

From the lists created from the recipe search and selection process, the list of recipe names will be added onto the Recipe List ListBox of the interface. The list of ingredients, names and amounts, will be printed and added to a txt file. These lists will be able to be deleted by selecting the added recipes and clicking the remove button on the interface which will run code to clear the selected item from the list created in the recipe selection and list generation processes.

Information Requirements

Data Flow Diagram



As the above data flow diagram illustrates, there are two inputs required for the proper functioning of the system. The first is the user input for the recipe search which will be collected through an entry box on the user interface. Once this input is received by the system, it will need data input from the Edamam website through the API. This data will be received in a JSON format. After this data is received, the user will be able to select recipes and the grocery list corresponding to these selected recipes will be produced by the system. This is the main output of the system and will be in the format of a list.

Interface Requirements

The interface will be created using the Tkinter package (see Technology Requirements). The interface will have a text entry box for users to search for recipes, a ListBox for users to select which search results they would like to add to their recipe list, and four buttons. The first button is to run the search. The second button will add the recipes selected in the ListBox to the recipe list and selected recipes ListBox.

The last two buttons will be to remove recipes from the selected recipes ListBox and to generate the grocery list. A sample of what the interface will look like is displayed below.

Search for a recipe by entering a recipe name or key term

Search term

Search

ListBox with search results

Add Recipe

Added Recipes

Remove Recipe

Show Grocery List

Technology Requirements

APIs

This system requires the Edamam Recipe Search API. When users search for recipes on the interface, the API retrieves recipe data from Edamam.com's database of recipes. This data is read into the program in a JSON format. This API is necessary to allow the user to search the web for recipes and for the program to parse the data to generate the grocery list.

More information about the Edamam API can be found here: <https://developer.edamam.com/edamam-docs-recipe-api>

Python Packages

There are three third-party Python packages that will need to be installed: dotenv, requests, Tkinter, and pytest. The dotenv package allows the API keys to be hidden from the public for security purposes. The dotenv package can then retrieve these secret values and read them into the program code without their values being revealed. The requests package is necessary to issue HTTP requests and use the capabilities of the Edamam API. The package interacts with the API allowing the program to exchange data with the API. This data exchange is critical in the search functionality of the system. The Tkinter package is necessary to create the graphical user interface of the system. This package will enable the program to have an interface that allows users to input information to search for recipes, select recipes with a ListBox, and then view their grocery list. The pytest package allows the automated tests incorporated into the system to run.