

P1 Proposal: Subsgredient

Brief Project Description:

This app enables users to find ingredient substitutions for recipes based on the category and calorie counts of the ingredients. The app makes use of Red-Black Trees to efficiently search through the ingredients data and retrieve it in a sorted manner. The data is loaded from a local data file that the user specifies.

Representative Tasks Performed Using this Application:

1. Find an ingredient that is a good replacement for passion fruit.
 2. Display statistics of the ingredient data set.
-

Backend Developer (BD) Role

The backend developer writes code that reads ingredient data from a CSV file, inserts it into two different Red-Black Trees, and accesses it based on commands from the frontend. To read in the data, the backend developer develops two java classes that define an ingredient object and that can be inserted into a Red-Black Tree. One of those classes will use the name string of the ingredient as the key so that names of ingredients can be looked up efficiently, the other one will use the caloric value of the ingredient. The backend supports data sets formatted in the same way as [this ingredients dataset from kaggle](#) in CSV format. The frontend accesses the backend's functionality through method calls on the backend object.

Interface Design Responsibilities:

- An interface for a class that defines a single ingredient and exposes ingredient properties required: category, name, and calories. This interface can be implemented by two different classes, one that compares the ingredients based on names, the other by calories.
- An interface for the backend that exposes the required functionality to the frontend: read data from a file, get a list of up to three ingredients to replace a specified ingredient, get the number of ingredients and categories as dataset statistics.

Presentation Responsibilities:

After integration, the backend developer demonstrates a search with the app to find an ingredient that is a good replacement for passion fruit.

Frontend Developer (FD) Role

The frontend developer writes code that drives an interactive loop of prompting the user to select a command, then requests any required details about that command from the user, and then displays the results of the command. The commands to include are: a command to specify (and load) a data file, a command that lists up to three replacements (with the same or minimally higher caloric value than the original ingredient) for an ingredient specified by the user, a command to list the number of ingredients and the number of categories, and a command to exit the app. The results are computed with the help of the BD's code. When the user enters invalid input, instructive feedback about what they should enter is displayed.

Interface Design Responsibilities:

- An interface for a class that implements the functionality of the frontend for the app. The class has a constructor that accepts a reference to the backend and a `java.util.Scanner` instance to read user input. It also has a method that starts the main command loop for the user. To allow for easier testing of the frontend, the command loop is broken down into a separate method for the main menu and each of the sub menus.

Presentation Responsibilities:

After integration, the frontend developer will record a video demonstration of the task: Display statistics of the ingredient data set.