The Bakers Box

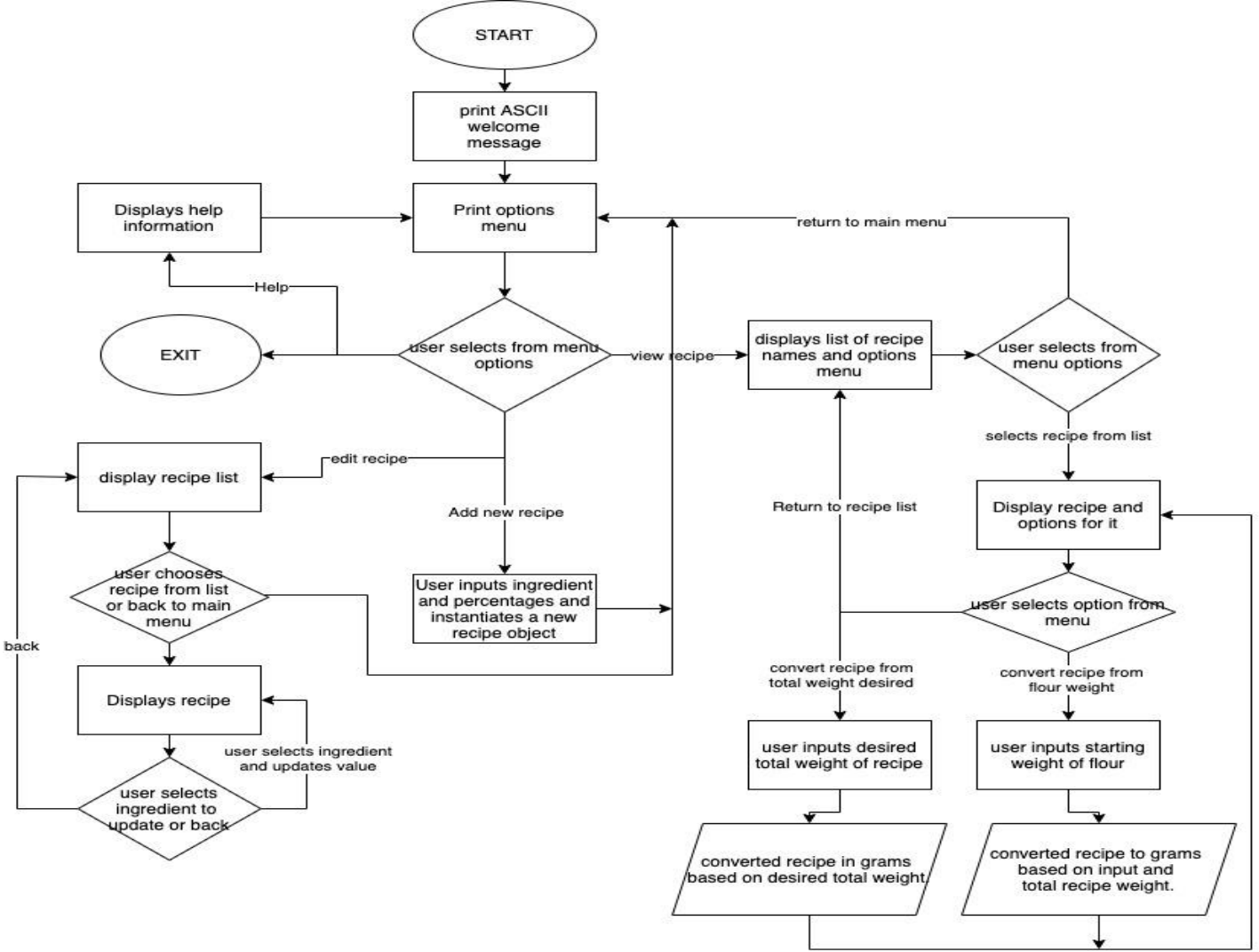
Terminal Application

Nick Saltis

What does this app do?

- Based on a variation of the “Baker’s Percentage” where recipe ingredients are expressed as a percentage of flour weight with flour weight being 100%
- Made more versatile by all ingredients being a percentage of the total weight of a recipe
- Stores percentage based recipes and allows the user to convert all ingredients of a recipe from one input (flour weight/total weight)
- Users can create new recipes, delete existing ones, or edit a recipe per ingredient.
- Recipes are stored in a JSON file so recipes are saved in between sessions
- TTY-prompt is used to create dynamic and interactive menus to navigate the app

Flow Chart


START
print ASCII welcome message
Print options menu
Displays help information
Help
EXIT
user selects from menu options
view recipe
displays list of recipe names and options menu
user selects from menu options
return to main menu
selects recipe from list
Display recipe and options for it
user selects option from menu
Return to recipe list
convert recipe from total weight desired
convert recipe from flour weight
user inputs desired total weight of recipe
user inputs starting weight of flour
converted recipe in grams based on desired total weight
converted recipe to grams based on input and total recipe weight.
edit recipe
display recipe list
Add new recipe
User inputs ingredient and percentages and instantiates a new recipe object
user chooses recipe from list or back to main menu
back
Displays recipe
user selects ingredient and updates value
user selects ingredient to update or back

Main Menu

- Displays ASCII title and welcome message.
- TTY-prompt menu for navigating to different parts of the app.

  Options include:

  - View Recipe Book
  - Add new recipe
  - Edit recipe
  - Help section
  - Exit app

Recipe Book

- A Dynamically created TTY-prompt menu with the names of recipes loaded into the app.
- Return to main menu option
- Options are updated in the same session when recipes are added/edited/deleted.
- Selecting a recipe displays its name and ingredients as well as a new TTY-prompt menu with options to convert recipe or return to recipe selection.
- Selecting a conversion option prompts for an input number (either flour weight or total weight) and displays the recipes with the calculated values.
- Conversion display and options loop indefinitely until user selects return to recipe selection.

Add new recipe

- Prompts user for input of recipe name
- User can select between adding an ingredient or finishing and saving recipe
- When user selects add ingredient they are asked to input ingredient name and then number representing percentage value of ingredient.
- This loops indefinitely until user selects finish and save.
- Returns to main menu when user finishes and saves recipe

Edit recipe

- User selects recipe from recipe book to display and edit
- User selects from TTY-prompt menu with options:
  - Add new ingredient: user inputs ingredient name and percentage value then returns to recipe edit options
  - Re-enter existing ingredient: user selects an ingredient to replace using TTY-prompt then does the same as add new ingredient
  - Delete ingredient: user selects ingredient to delete using TTY-prompt menu and then returns to recipe edit options
  - Delete recipe: entire recipe is deleted from the app.
  - Return to main menu: returns the user back to main menu screen.
- All options loop back to the recipe edit options until user selects return to main menu

Help Section

- Displays a TTY-prompt menu of topics to get help information on to avoid cluttering the window with mountains of text
- Options cover the main features of the app:
  - Recipe book
  - Conversion
  - Adding recipes
  - editing/deleting recipes
- After selecting a topic option user can then choose to return to main menu or select another topic
- Screen clears between each selection to keep terminal window clear

How it Works

- Recipes are stored in a JSON file and are read at app startup. Resulting hash is iterated over and each recipe is initialised in the RecipeBook Class. Each instance is stored within the class itself to be used in class methods.
- TTY-prompt is used to make selections from hardcoded or dynamically created options arrays using class methods. Results of prompts are run through case statements to determine which method should run.
- Conversion takes a user input integer and iterates over ingredient array calculating each recipe's ingredient percentage value with the input printing the results and also in the case of flour weight conversion, returns the total weight of the recipe.

How it Works

- Add new recipe uses user inputs and formats them into a hash that matches the other recipe hashes.
- Newly formatted recipe hash is merged into entire JSON hash and then the file is overwritten with the new value.
- Edit recipe also operates by navigating, updating or removing elements stored and then updating the JSON hash to be written.
- Main_menu functions as the main navigation method with case statement calling the various other methods which in turn call main_menu again to create loops.

Development Review

- Uphill battle trying to figure out how to link together multiple methods and how the flow of the program works once things become interlinked.
- Had to learn how to use JSON once i realised that I was limited to hard coded recipes unless I implemented file read/write functionality.
- Tried relying more on the RecipeBook Class to track and make modifications to its instances however had too much code to restructure to make it work.
- Most of the code involves parsing and formatting information stored in the JSON hash
- Code is dependant on user inputs being formatted to JSON hash correctly.
- Gems should be considered and implemented from the start to avoid major overhauls to code.
- Creating recursive and modifiable code is rewarding.

To be Completed

- Testing and error handling
- Writing logic to filter/modify user inputs to ensure it cooperates with the existing code and is formatted to JSON correctly.
- Ensuring there is at least one recipe in JSON file on app startup and if not creating an instance with hardcoded values of an example recipe.
- Implementing colorize and tty-box gems to further beautify my app.
- Bash scripting.
- DRYing out my code.