

# CS2103T

## Team Project: CookBuddy

Sharadh Rajaraman  
A0189906L

13<sup>th</sup> April 2020

## 1 Overview

[CookBuddy](#) is a desktop recipe manager for students living in on-campus accommodation, who enjoy cooking, and are also familiar with the command line.

Users may interact with CookBuddy through a Command Line Interface (CLI), and it has a Graphical User Interface (GUI) created with JavaFX.

CookBuddy is written in Java, and has about 10 000 lines of code.

## 2 Summary of contributions

[This RepoSense panel](#) displays this author's contributions to the project.

### 2.1 Enhancements Implemented

#### 2.1.1 Major Enhancements

##### Added image reading, UI display, and processing (PR [#278](#))

- **What it does:** Allows users to add images to their recipes, by specifying a relative or absolute filepath. Images are saved in a data folder local to CookBuddy, and are named with a UID (unique identifier). If an image cannot be found at the specified path, a placeholder is used instead.
- **Justification:** Users should be able to see what the final dish looks like, and an image goes a very long way in showing that. Furthermore, images are a prominent feature of most competing recipe managers, and we feel this is a key feature that allows CookBuddy to compete on a similar standing.
- **Highlights:**
  - Adding reading images required a firm understanding of file I/O, as well as understanding the various Java classes in [java.nio.file](#).  
Appropriate methods for easy extensibility were implemented in [cookbuddy.commons.util.PhotographUtil](#), as well as multiple constructors in [cookbuddy.model.recipe.attribute.Photograph](#).
  - Other file read/write methods were also implemented in [cookbuddy.commons.util.FileUtil](#).
  - PhotographUtil employs the [singleton pattern](#), to prevent initialisation errors.

#### 2.1.2 Minor Contributions

##### Revamped the User Interface (UI) of CookBuddy (PR [#179](#))

- A light theme was preferred by the team members
- [JMetro](#) was used to change the theme
- Appropriate classes and corresponding .fxml files were generated to produce the new UI.
- The panels of the UI were also changed, with a fixed scrolling side panel for the recipe list view, and a selected recipe overview panel.

### **Added Ingredient and Instruction List Attributes (PR #125)**

- Ingredients and instructions were originally stored as un-delimited Strings
- Classes to represent Ingredients, Instructions and their containers were implemented

### **Minor UI and Typeface Changes to Guides (Commit 01a9faf)**

## **2.2 Documentation Contributions**

### **2.2.1 User Guide**

- Added various feature details in § 4
- Added data file information in § 5.

### **2.2.2 Developer Guide**

- § 5, § 6.2 were added, detailing the overall architecture, and the implementation of image processing (see § 2.1.1).
- Future work was briefly detailed in § 6.12;
- Appendices A and B were added; in the latter, the User Stories were placed in a neat table.

## **2.3 Project Management and Teamwork**

### **Project Management**

- Set up the organisation, repository, and initial branches for all group members
- Set up Travis CI and AppVeyor for initial continuous integration testing

### **Teamwork**

- Assisted weaker members in setting up git and with iP tasks
- Set up Zoom meetings to facilitate discussion of project proposals and code reviews, as well as implementation details