Cai Ruicong - Project Portfolio

PROJECT: Calgo

This portfolio page highlights some of my contributions to Calgo - a Software Engineering project developed in my second year of undergraduate studies in the National University of Singapore.

About the Team

We are 5 Year 2 Computer Science undergraduates reading CS2103T: Software Engineering.

About the Project

Our project, Calgo is minimalistic calorie tracker. Calgo was built upon the brownfield project [Address Book Level 3 (AB3) project](#), morphed incrementally over a period of 3 months in accordance to software development principles. Calgo is written in java and totals to around 10 thousand lines of code.

Calgo's primary purpose is to provide fast typists with a tool that they can use to store records on their food consumption. Calgo uses a command line interface (CLI) to interact with the user. Whenever the user interacts with our App, the feedback is immediate and the effects are contained within a single window. This is so that our users can focus on their goals while Calgo do the heavy lifting. With many amazing features that are simple and fast to pick up, I hope our users will find it enjoyable and easy to monitor their health.

Summary of contributions

- **Major enhancement**: I implemented the `Consumption Records` of the App, which serves as a backend for features implemented by me and my teammates.

  ○ What it does: This allows users to reliably stored data in our app.

  ○ Justification: We need a storage that can load and save information of user so that it will be useful in the long run. Features such as report and graphs need data to work.

  ○ Highlights: This enhancement affects existing commands and commands to be added in future. It required an in-depth analysis of design alternatives. The implementation too was challenging as it required changes to existing commands. The future versions of `Calgo` will mainly be working with this feature to implement extensions. The implementation of `Consumption Record` was written to mirror the structure of AB3. However, it was much more because it has been scaled up keep track of many states and lists based on dates, as opposed to 1 list in AB3.

- **Major enhancement**: I implemented the `nom` and `vomit` command

  ○ What it does: These commands allows users to create, read, update and delete entries that represents their food consumption.

  ○ Justification: These feature is the reason our App exists, it addresses what the user wants to

do primarily.

- ◦ Highlights: `Nom` allows user to rate their meal experience, which is used to generate actionable insights based on the user's profile.

- **Minor enhancement**: I implemented the `stomach` command.

  - ◦ What it does: It allows users to browse `Consumption Records` on different dates.

  - ◦ Justification: The `stomach` command is as essential as a remote control to a TV as it is to the user.

  - ◦ Highlights: Although this might sound simple, it actually required a lot of planning and design to implement the structure because the data structure had to maintain a record for each date.

- **Code contributed**: You can view my contributions using reposense

- **Other contributions**:

  - ◦ Documentation:

    - ▪ Wrote sections for User Guide and Developer Guide: #273 #252 #302

    - ▪ Wrote Javadocs: #221 #215

  - ◦ Project and team management & contributions:

    - ▪ Maintained the issue tracker and assigned issues.

    - ▪ Contributed to user stories.

    - ▪ Managed the team repository and set up relevant services.

  - ◦ Beyond the team:

    - ▪ Peer testing & bug reporting: #13 #12 #9 #7 #5

Contributions to the User Guide

Given below are sections I contributed to the User Guide. They showcase my ability to write documentation targeting end-users.

include::../UserGuide.adoc

Contributions to the Developer Guide

Given below are sections I contributed to the Developer Guide. They showcase my ability to write technical documentation and the technical depth of my contributions to the project.

PROJECT: PowerPointLabs