Sahil S/O Sanjeev Gathani - Project Portfolio

Overview

This document serves as an overview of my contributions to the StudyBuddyPro project. This project was part of an introductory software engineering module known as **CS2103T**. Moreover, the document aims to highlight my technical competence and documentation skills by using the StudyBuddyPro project as an example.

StudyBuddyPro was developed by me, Samuel Lim, Chan Jun Ren, Chen Kai Bin and Jasmine Yeo Jia Min, all sophomore students from National University of Singapore (NUS) studying Computer Science.

Introduction

StudyBuddyPro is a desktop application designed for university students in computing-related fields to supplement their revision. StudyBuddyPro aims to consolidate commonly used studying tools into one integrated platform so the student does not have to fuss about switching between different applications. This prevents disorganisation and improves convenience.

StudyBuddyPro was developed for use through a Command Line Interface (CLI) to take advantage of the typing proficiency of computing students. However, a Graphical User Interface (GUI) is also provided for easier viewing, therefore providing the best of both worlds.

StudyBuddyPro has 3 main features: Flashcard, Notes and Cheatsheet. The purpose of each feature is summarized in the table below.

Table 1. Features table

Feature	Purpose
Flashcard	Create flashcards for quick revision and effective retention
Notes	Create general-purpose notes
Cheatsheet	Create and customize cheatsheets from relevant content found in Flashcard and Note features

Summary of contributions

This section provides an overview of the technical and non-technical contributions I made to the development of StudyBuddyPro. They highlight my ability to:
1) Design, model and implement features that are appropriate for our target users.
2) Work in a software development team setting and handle non-technical tasks such as documentation.
3) Design test cases and write quality code that abides by software engineering principles.
You can find an overall summary of my contributions at my RepoSense.

- **Major Feature Enhancements**
    - Developed the Flashcard `Model`, `Logic` and `Storage` components. This enhancement provides the functionality for one of the three core features in the StudyBuddyPro application. Without it, the application would be missing a key revision tool that students could have used and the application would be less attractive to the target market.
        - What this enhancement included:
            - The `Flashcard` object model and its components such as `Question`, `Answer`, `Title` and `Statistics` objects.
            - Basic commands to use the flashcard model such as `add` and `list`. Advanced features such as the **remind** feature.
            - Saving to and reading from storage files.
        - **Depth**:
            - Implementing this enhancement required a deep technical understanding of the architecture of the software as it modified 3 out of the 4 major components.
        - **Completeness**:
            - The enhancement is fully functional and goes beyond the basic needs of the user.
            - Advanced features like the **remind** feature makes use of scientifically established spaced repetition techniques so the user can effectively revise.
            - Each flashcard’s last viewed date is tracked and our application automatically sets the next date the flashcard should be viewed next which the user can check.
            - If the user tries to exit the application while there are still unrevised flashcards, the application will automatically warn the user about the unrevised flashcards and require a second confirmation before exiting.
        - **Effort**:
            - Understanding the entire existing architecture of the software was extremely challenging.
            - The enhancement modified and improved existing commands, while also linking various commands which was very tricky to implement.
            - The quantity of features added in this enhancement made it very time-consuming.
- **Minor Feature Enhancements**
    - Added a `CommandHistory` class that stores all commands a user inputs while using StudyBuddyPro.

        - This class was used to develop the **remind** feature described earlier.
        - This class was later used by my teammates to improve the existing `delete` commands in all features. This provided an additional layer of safety for the user by preventing accidental deletions.
    - Added a **Flashcard Bank** that provides users with a set of useful flashcards when the application first starts up.
        - The flashcards contained helpful tips for computing students. (Our target demographic)
        - This feature was also used during the product demo of our application for the module.
    - Code quality
        - Refactored `CommandResult` that all other features used. (Pull request #195)
        - Refactored `Storage` that all other features used. (Pull request #340)
        - Added more specific exceptions such as `DuplicateFlashcardQuestionException` and `DuplicateFlashcardTitleException` rather than generic exceptions such as `DuplicateFlashcardException`. This provided better logging for developers.

- **Other contributions**
    - Project Management
        - Set-up the GitHub developers page for our team.
        - Managed milestone v1.0 and v1.1 on GitHub. (2/5 milestones)
        - Managed the issue tracker for our project repository by assigning issues, cross-referencing pull requests with issues and closing issues when completed. (Pull request #340, Issue #301)
    - Testing
        - Wrote all test cases for Flashcard model. (Pull requests #321, #333, #337)
        - Wrote some test cases for Flashcard commands such as `add` and `remind`. (Pull requests #321, TOADD)
        - Wrote all test cases for overall StudyBuddyPro `Storage` component. (Pull request #340)
        - Overall, increased coverage by more than 5%. TOADD
    - Documentation
        - User Guide
            - [To add - Introduction, Quick start?]
        - Developer Guide
            - [To add - Manual Testing?, Use Cases]
        - Tools:
            - Integrated Travis CI, a continuous integration service, into our repository.
    - Community Contributions [ADD IF HAVE SPACE]
        - Contributed to forum discussions (Examples: 1)
        - Reported bugs and suggestions for other teams in the class (Examples: 1)

- PRs reviewed (with non-trivial review comments): #12, #32, #19, #42

Contributions to the User Guide

Given below are sections I contributed to the User Guide. They showcase my ability to write clear and concise documentation that helps the non-technical end user use the application. I also took active effort to ensure my writing was as friendly and inviting as possible. Note that some sections have been omitted which may affect coherence while reading.

Introduction

StudyBuddyPro is a student application that aims to simplify the hassle of revision by providing a suite of tools for effective revision.

StudyBuddyPro is optimized for students who prefer to work with a Command Line Interface (CLI) while still having the benefits of a Graphical User Interface (GUI).

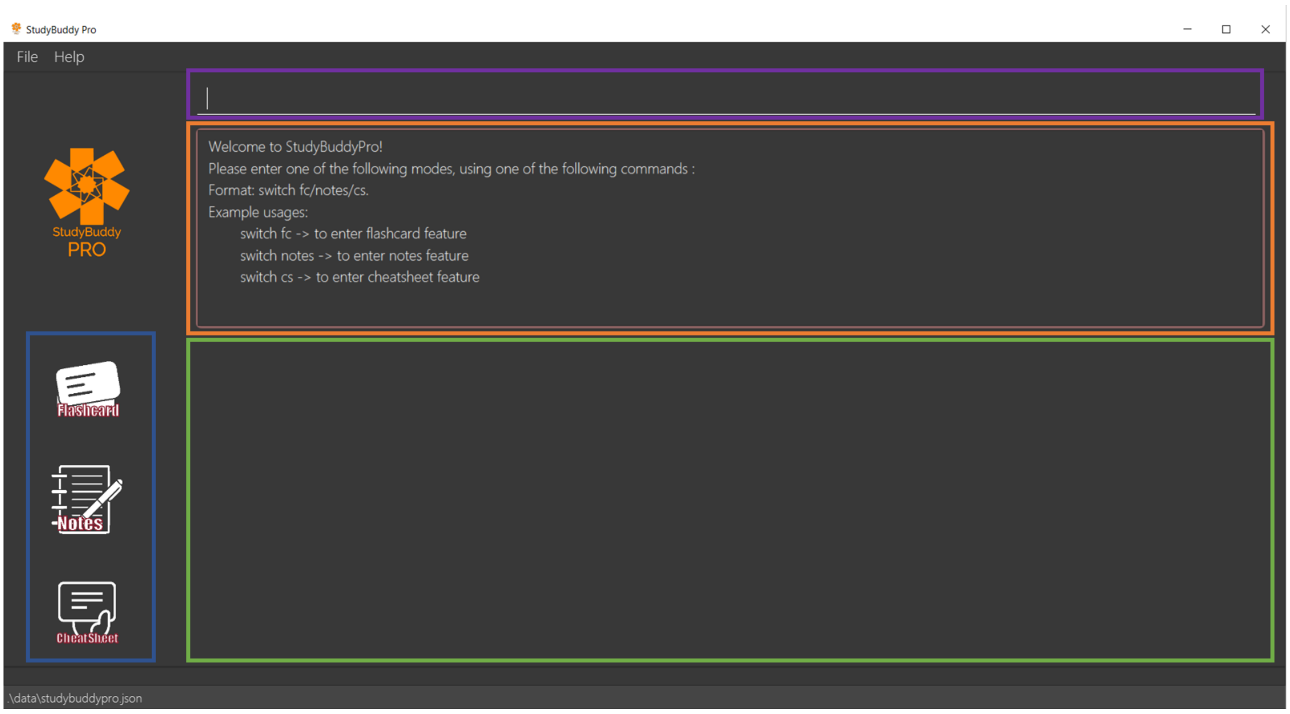
Moreover, StudyBuddyPro comes geared with pre-loaded features specially catered for computing students. So whether you're a computing student getting used to a CLI for the first time or if you're a CLI expert who wants to reap the benefits of a fast typing speed, give StudyBuddyPro a try!

If you're interested, head over to the Quick Start section to get started!

Quick Start

NOTE Please ensure you have Java 11 or above installed before proceeding!

1. Download the latest version of `StudyBuddyPro.jar` here.
2. Place the file in the folder you want to set as the home directory. All data and miscellaneous files associated with StudyBuddyPro will be placed in this folder.
3. Double-click `StudyBuddyPro.jar` to launch the application. The GUI should appear in a few seconds, and should look like the screen shown below. If not, please refer to the first question in the FAQ for help!



Welcome to StudyBuddyPro!
Please enter one of the following modes, using one of the following commands :
Format: switch fc/notes/cs.
Example usages:
switch fc -> to enter flashcard feature
switch notes -> to enter notes feature
switch cs -> to enter cheatsheet feature

.\data\studybuddypro.json

Figure 1. GUI of StudyBuddyPro application displayed on startup

4. Type a command in the command box execute it by pressing kbd:[Enter]. Typed commands will appear in the CLI highlighted in the purple box in the diagram above. Refer to the Command Summary section for a quick overview of all the available commands!

TIP The blue box in the diagram above with the "Flashcards", "Notes" and "Cheatsheets" logo can be used to quickly check which mode you are in! Switching into a mode will highlight the relevant mode’s logo in an orange circle, as shown in this figure.

5. Output from the command will be shown in the boxes highlighted in orange and green in the diagram above. The green box is used to display a flashcard, note or cheatsheet while the orange box outputs feedback from commands.

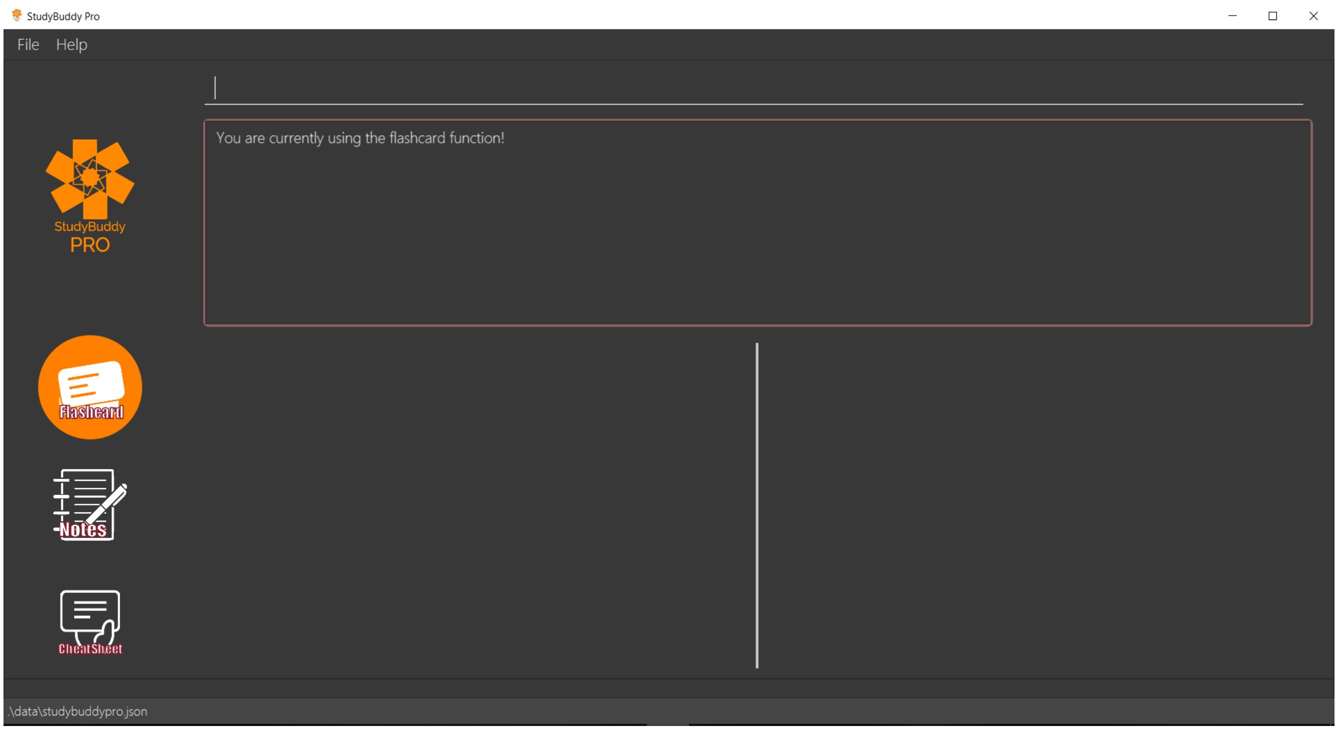
Flashcard Feature

TIP Good news - StudyBuddyPro comes with some preloaded flashcards, specially catered for you as a computing student! Be sure to take a look! Psst - here’s a hint for our more tech-savvy users: You can delete your flashcards.json file in the StudyBuddyPro data folder to restore these default flashcards at any time. Of course, your current flashcards will be deleted as well!

Sick and tired of cramming all your revision at the last minute? Why not give our Flashcards feature a try! This feature can help you create your very own flashcards to help you consistently revise. With our built-in reminder features and timetrial modes to test yourself, use this feature and be on track to better revising habits today!

IMPORTANT

All the operations in this section assume that the user is in the *flashcard* mode. To be sure you are *flashcard* mode, please ensure you used the `switch fc` command before this. Your screen should now look like the one found in the screenshot below.



You are currently using the flashcard function!

.\data\studybuddypro.json

How to create a flashcard: `add`

Adds a flashcard from user input question <QUESTION> and answer <ANSWER>.

```
Format: add q/QUESTION a/ANSWER t/TITLE [tag/TAG]...
```

Example usage:

```
add q/What is 100 Binary in its Decimal form? a/4 t/Binary Stuff tag/CS2100
```

Expected output:

```
New flashcard added:
    New flashcard added: Title: Binary Stuff
    Tags: [cs2100]
```

Editing a flashcard: `edit` (Coming in v2.0)

Edits a flashcard’s question, answer, title, or tags. The flashcard will be referred to by their original title `ORIGINAL_TITLE`.

```
Format: edit ORIGINAL_TITLE [q/NEW_QUESTION] [a/ANSWER] [t/TITLE] [tag/TAG]...
```

- At least one of the optional fields must be provided.

Example usage:

```
edit IntelliJ Question 1 q/What is the meaning of SLAP? a/Single Layer of Abstraction
Principle t/SE Question 1
```

Expected output:

```
Edited flashcard:
    Title: SE Question 1
    Question: What is the meaning of SLAP?
    Tags: [cs2100]
```

- Notice how the fields that are not edited retain their original information. For example, the *tag* field was not changed and so the original *cs2100* tag was retained.
- Multiple fields can be edited at the same time. In the example, the *question*, *answer* and *title* fields were all edited at once.

How to view a list of all available flashcards: `list`

Lists all flashcards.

```
Format: list
```

Expected output:

```
Listed all flashcards:
    Title: Pipelineing Question 1
    Tags: [CS2100]
```

How to delete a flashcard: `delete`

Deletes the flashcard by <FLASHCARD_INDEX>.

The user will be prompted again to confirm their deletion.

```
Format: delete (index)
```

Example usage:

```
delete 6
```

Expected output:

```
Are you sure you would like to delete the following flashcard?
    Title: Binary Question 1
    Tags: [cs2100]
    Please use `delete 6` again to confirm your deletion..
```

Upon keying in `delete 6` again, then the flashcard will be deleted.

Expected output:

```
Deleted Flashcard:  Title: Binary Question 1
    Tags: [cs2100]
```

Find out what flashcards to revise today, or ones you may have missed: `remind`

This feature helps the user check which flashcards are due for revision today and which flashcards overdue for revision. Don't worry, StudyBuddyPro automatically sets the date the flashcard should next be viewed at for optimal learning. These increments also scale with time i.e. newer flashcards will have to be viewed more often.

TIP Be sure to check in everyday to see which flashcards you have due!

NOTE StudyBuddyPro only marks a flashcard as revised and removes it from the due and overdue flashcard list when you see the flashcard's *answer* not just its question! For example, simply using the `view` command without the `show` command to reveal the flashcard's answer will not trick the system. Sorry, it's for your own good!

Example usage:

```
remind
```

If no flashcards due for revision today and no overdue flashcards:

Expected output:

```
Well done - No due or overdue flashcards!
```

If there are flashcards due for revision today but no overdue flashcards:

Expected output:

```
Here are the flashcards due today:
1. Math Question 1 - What is 2 x 2?
```

If there are no flashcards due for revision today but there are overdue flashcards:

Expected output:

```
Here are your overdue flashcards:
1. Math Question 1 - What is 2 x 2? (Was due on 2019-10-30)
```

If there are both flashcards due for revision today and overdue flashcards:

Expected output:

```
Here are the flashcards due today:
1. Math Question 1 - What is 2 x 2?
Here are your overdue flashcards:
1. Math Question 2 - What is 3 x 2? (Was due on 2019-10-30)
```

FAQ

Q: Help! Double-clicking `StudyBuddyPro.jar` does not launch the application - what should I do?
A: Trying running the application from the command line using the following command: `java -jar StudyBuddyPro.jar`. Windows users can use the Command Prompt application to do this while Mac users can use the Terminal application.

Q: The preloaded deck of flashcards is overdue for revision and StudyBuddyPro says the last viewed date for those preloaded flashcards was when I first opened StudyBuddyPro, even though I never even viewed those flashcards! How is this possible?
A: StudyBuddyPro assumes all the default flashcards will be viewed when the application was first opened. Aren't you curious to see what we collated for you?

Contributions to the Developer Guide

Given below are sections I contributed to the Developer Guide. They showcase my advanced technical knowledge through the depth of my contributions. They also showcase my ability to explain complicated technical information in an easy-to-read, digestible manner. This was achieved in a variety of ways, such as the use of examples and industry standard technical diagrams.