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 documentation targeting end-users.*

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.*

{Include summary}