

Ling Zhi Wei - Project Portfolio

PROJECT: Flashmind

Overview

Flashmind is a flashcard program to help users memorise content. FlashMind has a GUI with an easy to use commandline interface(CLI) . It is written in Java, and has about 10 kLoC.

Summary of contributions

- **Major enhancement:** added **the ability to generate statistics about the flashcards along with an accompanying UI**
 - What it does: Allows the user to generate statistics about cards by tag or overall. Displays information in a helpful graphical information as well as in text
 - Justification: This feature improves the product significantly because the user might want to track the progress of his learning experience.
 - Highlights: This enhancement required some architecture changes to allow statistics information to communicated to the UI . It required an in-depth analysis of UI design alternatives. The implementation was challenging due to communication between the GUI to detailed information and creating a separate window with graphics.
- **Major enhancement:** added **the ability to add choices to flashcard**
 - What it does: Allows the user to add a list of choices to the flashcard.
 - Justification: This feature improves the product significantly because the user can now create multiple choice flashcards which greatly improves versatility and .
 - 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.
- **Major enhancement:** added **a timer to the quiz mode**
 - What it does: Allows the user to have timed quizzes.
 - Justification: This feature improves the product significantly because the user can simulate test environments with the quiz system adding a new layer of functionality .
 - Highlights: This enhancement affects existing commands and commands to be added in future. It required an in-depth analysis of UI design. The implementation too was challenging as it required changes to existing commands.
- **Minor enhancement:**
 - Major changes to UI to fix the majority of the bugs and improved usability.

- Added the ability to skip cards in quiz with a skip command
- **Code contributed:** [[Functional code](#)] [[Test code](#)] {give links to collated code files}
- **Other contributions:**
 - Enhancements to existing features:
 - Wrote additional tests for existing features to increase coverage from 62% to 71% in addition to other tests
 - Documentation:
 - Did cosmetic tweaks to existing contents of the User Guide

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.

Skip Flashcards in Quiz

Format: `skip`

*Used to skip a flashcard during a quiz if the user does not know the answer or want to answer

Display current progress

Shows the current progress of a set of flashcards.

Format: `stats [t/TAG]...`

- Statistics shown include number of completed cards and accuracy
- Multiple TAG's can be added to combine statistics of multiple sets
- If no TAG is provided statistics of all the cards will be shown

Examples:

- `stats t/biology` : Displays the progress of a the flashcards tagged with Biology.

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.

Model component

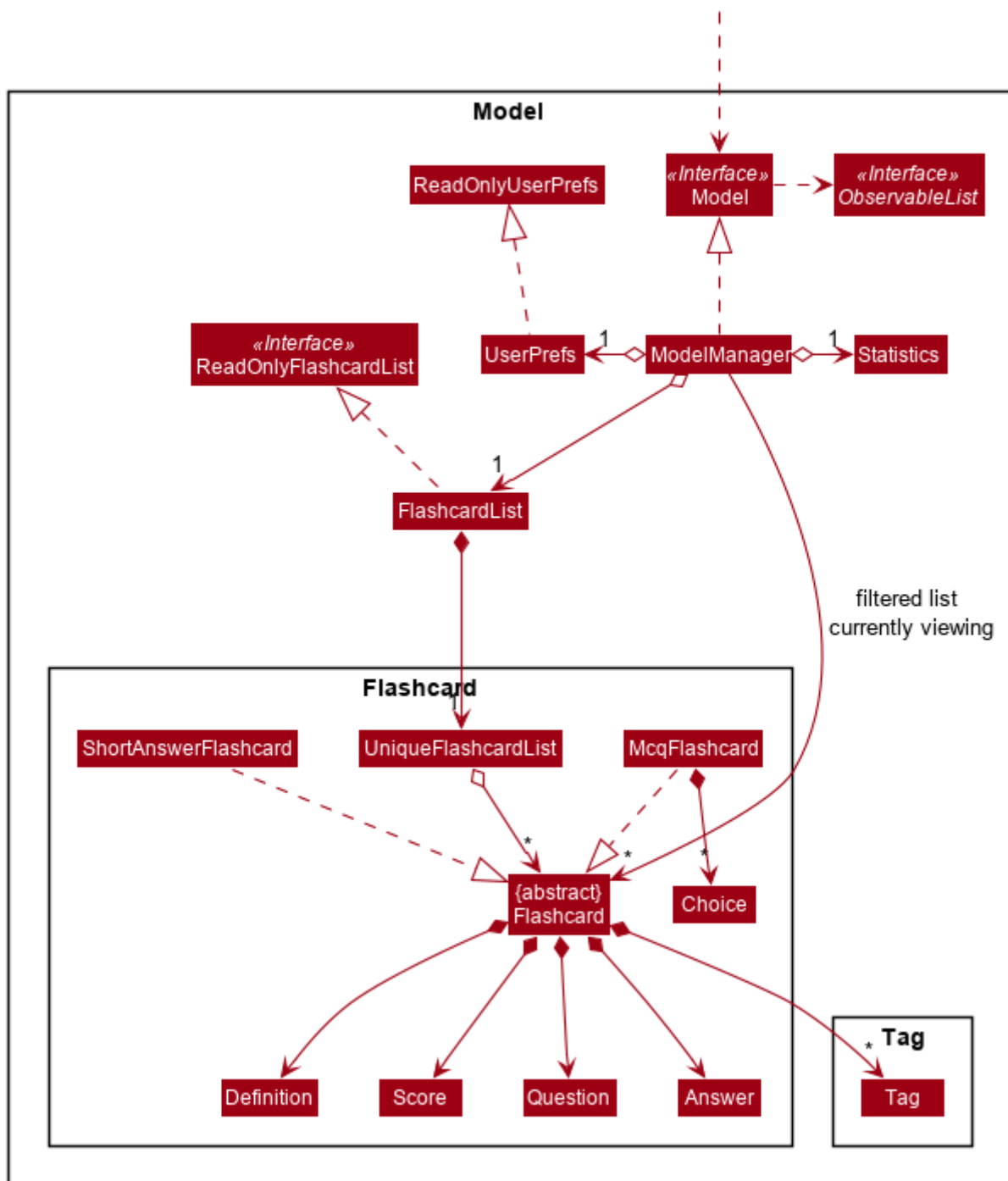


Figure 1. Structure of the Model Component

API : `Model.java`

The `Model`,

- stores a `UserPref` object that represents the user's preferences.
- stores the Flashcard List data.
- exposes an unmodifiable `ObservableList<Flashcard>` that can be 'observed' e.g. the UI can be bound to this list so that the UI automatically updates when the data in the list change.

- also exposes a statistics object to the ui to create the statistic charts .

Use case: UC11 - View the statistics of flashcards

Actor: User

MSS

1. User enters **stats** to ask FlashMind to show stats of all cards
2. System displays the statistics of the selected flashcards in a new window

Use case ends.

Extensions

- 1a. User enters **stats** **[t/Tag]** to ask FlashMind to show stats of cards under specified tags.

Use case resumes at step 2.

- 2a. The tag is empty or does not exist.

2a1. System displays an error message.

user case ends.

Contributions to the README

Given below are sections I contributed to the README. They showcase my ability to write documentation advertising the project.

- Need to cram for an upcoming test? Interested in learning a new language? FlashMind is here for you. This flashcard program helps you memorise more efficiently, reducing study time and increasing the amount of content you can learn
- This application will help anyone looking to remember things faster and for longer, it can help users to:
 - Revise for a test
 - Learn general trivia
 - Pick up a new language
 - And much more!
- FlashMind has a GUI with an easy to use commandline interface(CLI) which is more intuitive than searching through multiple menus for a command.
- Highly customizable: Add as many flashcards as you want to the program and edit them anytime
- Flexible: Easily create flashcard sets from any combination of your existing flashcards

- Multiple modes: Normal flashcard mode for memorizing or a quiz mode to test yourself

PROJECT: Flashmind
