**Teacher’s Notebook - User Guide**

By: Team CS2103-T14-1

1. Introduction

2. Quick Start

3. Features

3.1 Adding: add

3.1.1 Classes

3.1.2 Students

3.1.3 Assignments

3.2 Editing: edit

3.2.1 Students’ information

3.2.2 Assignment list

3.2.3 Students’ grades for individual assignments

3.3 Listing: list

3.3.1 Class list

3.3.2 Student list

3.3.3 Assignment list

3.4 Sending reminders to teachers

3.5 Locating: find

3.5.1 Students by name

3.5.2 Assignments by name

3.6 Deleting: delete

3.7 Undo: undo

3.8 History

3.9 Generate file: generate

3.10 Saving data

3.11 Exiting the programme: exit

4. FAQ

5 Command Summary

**1. Introduction**

Teacher’s Notebook is for teachers who prefer to make use of an application to keep track of the deliverables, tasks and students’ information. It is optimized for teachers who prefer to work with a Command Line Interface (CLI) while still having the benefits of a Graphical User Interface (GUI). If you are a teacher who wants to manage your students and your tasks efficiently, and can type fast, Teacher’s Notebook can get your work done much faster than traditional applications! Jump to Section 2, “Quick Start” to get started! Enjoy!

**2. Quick Start**

1. Ensure you have Java 11 or above installed in your Computer.

2. Download the latest teachersnotebook.jar here.

3. Copy the file to the folder you want to use as the home folder for your Teacher’s Notebook.

4. Double-click the file to start the app. The GUI should appear in a few seconds.

5. Type in the commands and press enter to execute them. Refer to Section 3: Features for more details on the commands.

**3. Features**

**Command Format**

● *Words in UPPER\_CASE are the parameters supplied by the user e.g. in add c/CLASS, CLASS is a parameter which can be used as add c/4C.*

● *Items in square brackets are optional e.g. n/NAME [a/ADDRESS] can be used as n/John Doe a/National University of Singapore or n/John Doe.*

● *Items with … after them can be used multiple times including zero times e.g. [m/MEDICAL\_CONDITIONS]... can be used as (i.e. 0 times), m/Sinus, m/Sinus m/Asthma etc.*

● *Parameters can be in any order e.g. if the command specifies n/NAME e/EMAIL, n/NAME, e/EMAIL is also acceptable*

**3.1 Adding: add (coming in V2.0)**

**3.1.1 Classes**

Adds a class to the class list.

Format: add c/CLASS

Examples:

● add c/4C

● add c/3E5

**3.1.2 Students**

Adds a student to the student list.

Format: add n/NAME p/PHONE\_NUMBER e/EMAIL a/ADDRESS [m/MEDICAL\_CONDITIONS] [t/TAG]...

Examples:

● add n/John Doe p/91234567 e/john.doe@gmail.com a/John Street, Block 123, #01-01

● add n/Jane Doe p/98765432 e/jane.doe@gmail.com a/Jane Street, Block 123, #10-10

**3.1.3 Assignments**

Adds an assignment to the assignment list

Format: add as/ASSIGNMENT d/DEADLINE

Examples:

● add as/CS2103 Team Project d/31/10/2019

● add as/CS2103 Individual Project d/30/09/2019

**3.2 Editing: edit (coming in V2.0)**

**3.2.1 Students’ information**

Edits an existing student in the student list.

Format: edit c/CLASS INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [m/MEDICAL\_CONDITIONS], [t/TAG]...

Examples:

● edit c/4C 2 p/91111111

● edit c/5E2 4 e/new\_email@gmail.com

**3.2.2 Assignment List**

Edits an existing assignment in the assignment list.

Format: edit as/ASSIGNMENT [as/ASSIGNMENT] [d/DEADLINE]

Examples:

● edit as/CS2103 Assignment 2 [as/CS2103 Assignment 3]

● edit as/CS2103 Examinations [d/19/09/2019]

**3.2.3 Students’ grades for individual assignments**

Edits the grades for students in the individual assignments.

Format: edit as/ASSIGNMENT GRADE\_ONE/GRADE\_TWO/GRADE\_THREE/...

Examples:

● edit as/CS2103 70/30/50

● edit as/CS2103 Assignment 1 1/2/3/4/5/6/7/8/9/10

**3.3 Listing: list**

**3.3.1 Class list**

Shows a list of all classes taken by the user.

Format: list class

**3.3.2 Student list**

Shows a list of all students in a particular class of the user.

Format: list [c/CLASS] students

Examples:

● list c/4E3 students

● list students

**3.3.3 Assignment list**

Shows a list of all assignments of classes of the user.

Format: list c/CLASS assignments

Examples:

● list c/4C assignments

● list c/5E3 assignments

**3.4 Sending reminders to teachers (coming in V2.0)**

Sends system notification to user 10 minutes before event (assumes user keeps program running continuously).

Format: none, reminder should come automatically

**3.5 Locating: find**

**3.5.1 Students by name**

Find students whose name matches a given keyword.

Format: find s/STUDENT\_NAME

Examples:

● find s/John Doe

● find s/Jane Doe

**3.5.2 Assignments by name**

Find assignments which name matches given keyword.

Format: find [c/CLASS] as/ASSIGNMENT

Examples:

● find c/4E3 as/CS2103

● find as/CS2103

**3.6 Deleting an assignment: delete (coming in V2.0)**

Deletes the specified assignment from the assignment hashmap. The name refers to the key of the hashmap of assignments.

Format: delete ASSIGNMENT

Examples:

● delete CS2103 midterm

● delete CS2103 project

**3.7 Undo: undo (coming in V2.0)**

Restores application to the state before previous before previous command was executed.

Format: undo

**3.8 History (coming in V2.0)**

Generates user’s last typed command.

Format: up arrow key brings up last typed command into user input box.

**3.9 Generate file: generate (coming in V2.0)**

Generate test score of class for every test into csv file. This will be in a more readable format for users to see.

Format: generate CLASS\_NAME

Examples:

● generate 4C

● generate 5E3

**3.10 Saving data (coming in V2.0)**

The programme automatically saves the data in a txt file whenever a command is issued.

Format: none, saving data will be done automatically

**3.11 Exiting the programme: exit (coming in V2.0)**

Exits the program.

Format: exit

**4. FAQ**

Coming up soon!

**5. Command Summary**

● **Add:** add n/NAME p/PHONE\_NUMBER e/EMAIL a/ADDRESS [m/MEDICAL\_CONDITIONS] [t/TAG] …

e.g. add n/John Doe p/91234567 e/john\_doe@gmail.com

● **Edit:** edit c/CLASS INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [m/MEDICAL\_CONDITIONS], [t/TAG]

e.g. edit c/4C 2 p/91111111

● **List:** list class

● **Find:** find s/STUDENT\_NAME

e.g. find s/John Doe

● **Delete:** delete ASSIGNMENT

e.g. delete midterms

● **Undo:** undo

● **Generate:** generate CLASSNAME

e.g. generate 4C

● **Exit:** exit