



User Guide

Updated as of 11/11/2019

Prepared by:
CS2113T-F14-4 / CS2101-SC07-GROUP2

Table of Contents

1. Introduction

1.1 What is SpinBox?

2. Quick Start

3. Features

3.1. Exiting the program : bye

3.2. Change to the main tab : view / main

3.3. Viewing the calendar tab : view / calendar

3.4. : Viewing the modules tab : view / modules

3.4.1. : Adding a module : add / module

3.4.2. : Removing a module : remove / module

3.5. : Viewing a specific module : view / modules

3.5.1. Viewing the Tasks section : view / tasks

3.5.1.1. Adding a todo task : add / todo

3.5.1.2. Adding a deadline task : add / deadline

3.5.1.3. Adding an event task : add / event

3.5.1.4. Adding an exam event : add / exam

3.5.1.5. Adding a lab event : add / lab

3.5.1.6. Adding a lecture event : add / lecture

3.5.1.7. Adding a tutorial event : add / tutorial

3.5.1.8. Removing a single task : remove / task

3.5.1.9. Removing multiple tasks : remove-* / task

3.5.1.10. Updating the status of a task : update / task

3.5.1.11. Updating the status of multiple tasks : update-* / task

3.5.1.12. Editing the name of a task : set-name / task

3.5.1.13. Editing the date of a schedulable task : set-date / task

3.5.2. Viewing the Files section : view / files

3.5.2.1. Adding a file to be tracked : add / file

3.5.2.2. Removing a single file : remove / file

3.5.2.3. Removing multiple files : remove-* / file

3.5.2.4. Updating the status of a file : update / file

3.5.2.5. Updating the status of multiple files : update-* / file

3.5.2.6. Editing the name of a file : set-name / file

3.5.3. Viewing the Grades section : view / grades

- 3.5.3.1. Adding a graded component : add / grade
 - 3.5.3.2. Updating the status of a graded component : update / grade
 - 3.5.3.3. Removing a graded component : remove / grade
 - 3.5.3.4. Updating graded component scores with relative values : score / grade
 - 3.5.3.5. Updating graded component scores with absolute percentage : score / grade
 - 3.5.3.6. Entering final grade : result / grade [coming in v2.0]
- 3.5.4. Adding a custom note : add / note
- 3.5.5. Removing a custom note : remove / note
- 3.5.6. Editing a custom note: set-name / note
- 3.6. Viewing the Help page : help
 - 3.6.1. Viewing the Help page on the view command : help / view
 - 3.6.2. Viewing the Help page on the add command : help / add
 - 3.6.3. Viewing the Help page on the remove command : help / remove
 - 3.6.4. Viewing the Help page on the remove-* command : help / remove-*
 - 3.6.5. Viewing the Help page on the set-name command : help / set-name
 - 3.6.6. Viewing the Help page on the set-date command : help / set-date
 - 3.6.7. Viewing the Help page on the update command : help / update
 - 3.6.8. Viewing the Help page on the update-* command : help / update-*
 - 3.6.9. Viewing the Help page on the export command : help / export
 - 3.6.10. Viewing the Help page on the score command : help / score
 - 3.6.11. Viewing the Help page on the populate command : help / populate
 - 3.6.12. Viewing the Help page on the find command : help / find
- 3.7. Locating module component by name : find
- 3.8. Saving the data
- 3.9. Exporting data : export
- 3.10. Calculate CAP : cap [coming in v2.0]
- 3.11. Populate with sample data : populate
- 3.12. Logs

4. FAQ

5. Command Summary

1. Introduction

Welcome to the *SpinBox User Guide*! In this guide, you can find step by step instructions on setting up SpinBox, as well as ways to utilise SpinBox to its maximum potential in order to better manage your university life and achieve your academic goals. Interested? Proceed to the next section, [Quick Start](#) to get started. We wish you a fruitful experience with SpinBox!

1.1 What is SpinBox?

SpinBox is a desktop companion app for NUS students to provide important, self-contained features that current learning platforms fail to provide. Managing a multitude of tasks, deadlines, on top of a busy schedule can be a strenuous and frustrating experience for many students, especially as the semester progresses into the later weeks. Additionally, needing to frequently context switch between 5-6 modules often takes a toll on a student's ability to perform to the best of his/her abilities.

With SpinBox, we aim to provide you with the high level view while still being able to dive right into the details. Through our intuitive and powerful Graphical User Interface (GUI), you can easily manage your tasks, files and grades, classified according to modules. This allows you to focus on what's most important while shielding you from being overwhelmed by context switches and other distractions.

Additionally, you are also able to view your five most urgent tasks across all modules on the main tab, as well as your upcoming exam dates. On the Calendar tab, you will be able to view your tasks for the month, arranged in a chronological fashion.

SpinBox is optimised for students who prefer to work with a Command Line Interface (CLI) while still having the benefits of a GUI. If you can type fast, SpinBox can organise your tasks, grades and files significantly faster than traditional GUI apps. Through SpinBox, we also hope that you can hone your prioritization and time management skills, which are beneficial and transferable to the workplace.

SpinBox currently supports Windows, Linux and macOS users.

2. Quick Start

1. Ensure you have Java 11 or above installed on your computer.
2. Download the latest SpinBox.jar [here](#).
3. Copy the file to the folder you want to use as the home folder for SpinBox.
4. Double-click the file to start the app. The GUI as shown in Figure 1 should appear in a few seconds.

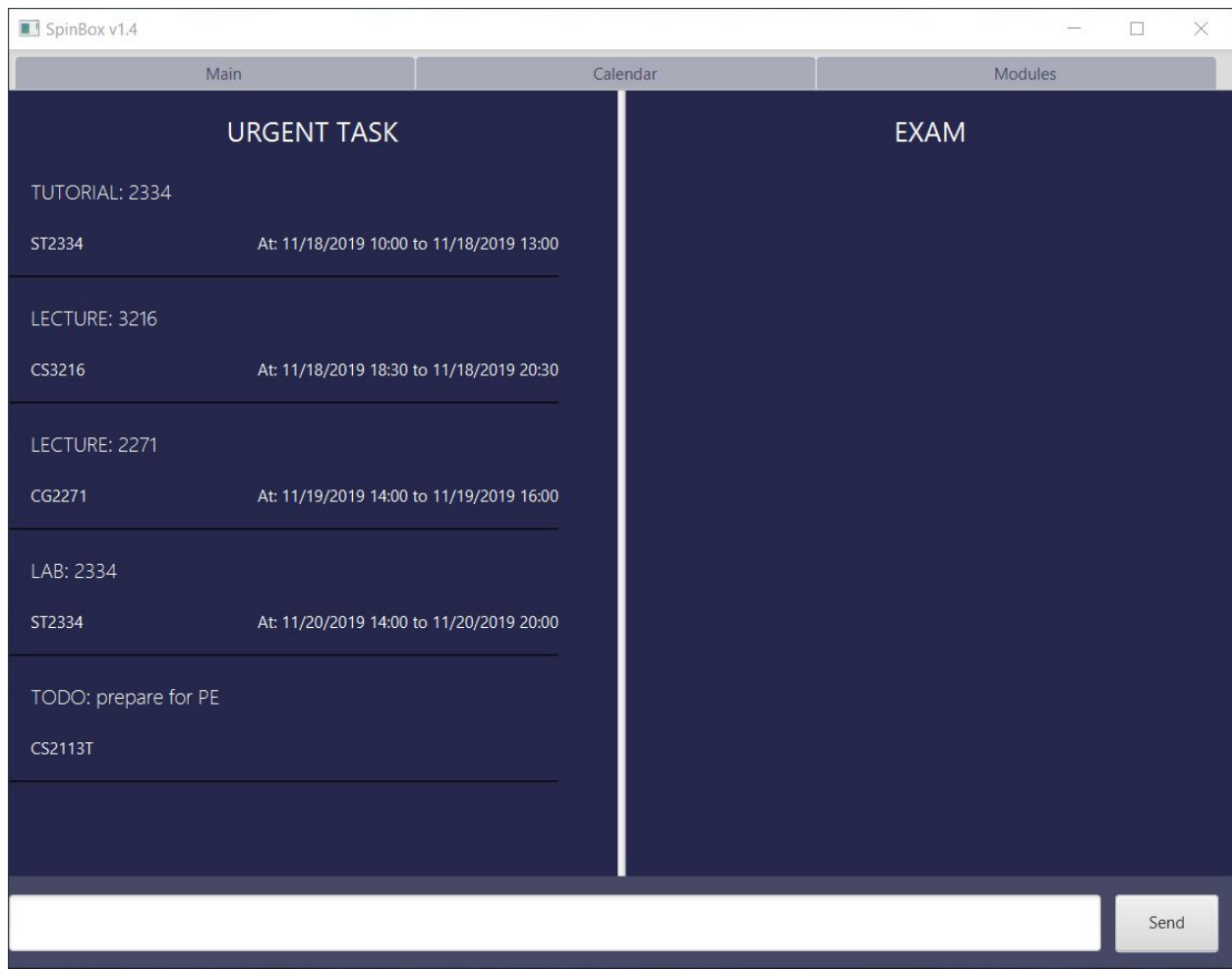


Figure 1. SpinBox's Main Tab, which is shown upon launching SpinBox

5. On startup, you may observe the following text within the input box:

We notice you have no existing data. Type "populate" into this input box to load sample data.



You have the option of using our provided sample data to experiment with or test out the possible commands, by typing in `populate` into the same box.

6. Type a command in the command box and press `Enter` to execute it.
e.g. typing `view / calendar` and pressing `Enter` will open the calendar.
 7. Some example commands you can try (in successive order):
 - `add / module CS2113T Software Engineering & OOP` : creates a module with the module code of CS2113T and a description of Software Engineering & OOP
 - `view / CS2113T` : view the Task subtab of the CS2113T module
 - `add CS2113T / todo myFirstTask` : adds a Todo task named myFirstTask to CS2113T
 - `remove CS2113T / task 1` : deletes the 1st task shown in the current list, which should be the todo named myFirstTask
 - `bye` : exits the app
 8. Refer to Section 3, [Features](#) for the details of each command as well as their shortened versions for power-users.
-


3. Features

SpinBox supports all standard functionalities expected of a good task management application, such as creating, updating and deletion of items. Additionally, SpinBox also boasts several other features that are tailored to a student's needs. Ultimately, we aim to enhance your user experience through a streamlined task, file and grade management process.

Note the following symbols and formatting used in this document:

1. : Used to provide information regarding certain errors that you may face while using SpinBox.
2. : Used to denote an important tip that further enhances your user experience.

Command Format
<ul style="list-style-type: none">• Words in <code>[UPPER_CASE]</code> are the parameters to be supplied by the user e.g. in <code>view / modules CG1111 [TYPE]</code>, <code>[TYPE]</code> is a parameter which can be used as <code>view / modules CG1111 tasks</code>.• <code>[INDEX]</code> refers to the index number shown in the displayed file/grade/note/task list. The index must be a positive integer, e.g. 1,2,3, ...• <code>[DATETIME]</code> is to be entered as <code>MM/DD/YYYY HH:MM</code>, and can be substituted using natural language, e.g. <code>tomorrow 6pm</code>. However, the accuracy of natural language input cannot be guaranteed. We strongly recommend sticking to our specified format for best results.

 Pressing the `↑` and `↓` arrows will display the previous and next input respectively in the command box.

3.1. Exiting the program : `bye`

Exits the application, and can be entered at any point in time by the user.

Format: `bye`

3.2. Change to the main tab : `view / main`

Default active tab when the application is started. Displays a list of the five most urgent pending tasks across all modules on the left side of the screen and a list of modules on the right (Figure 1).

Format: `view / main`

3.3. Viewing the calendar tab : `view / calendar`

Displays the calendar in the calendar tab populated using task data across all modules. The calendar only displays tasks that have a start and end date (a To do or deadline will not show up on the calendar). The current month is displayed by default, unless the full command is used.

Format: `view / calendar` or `view / calendar [MM/YYYY]`

An expected output is shown below.

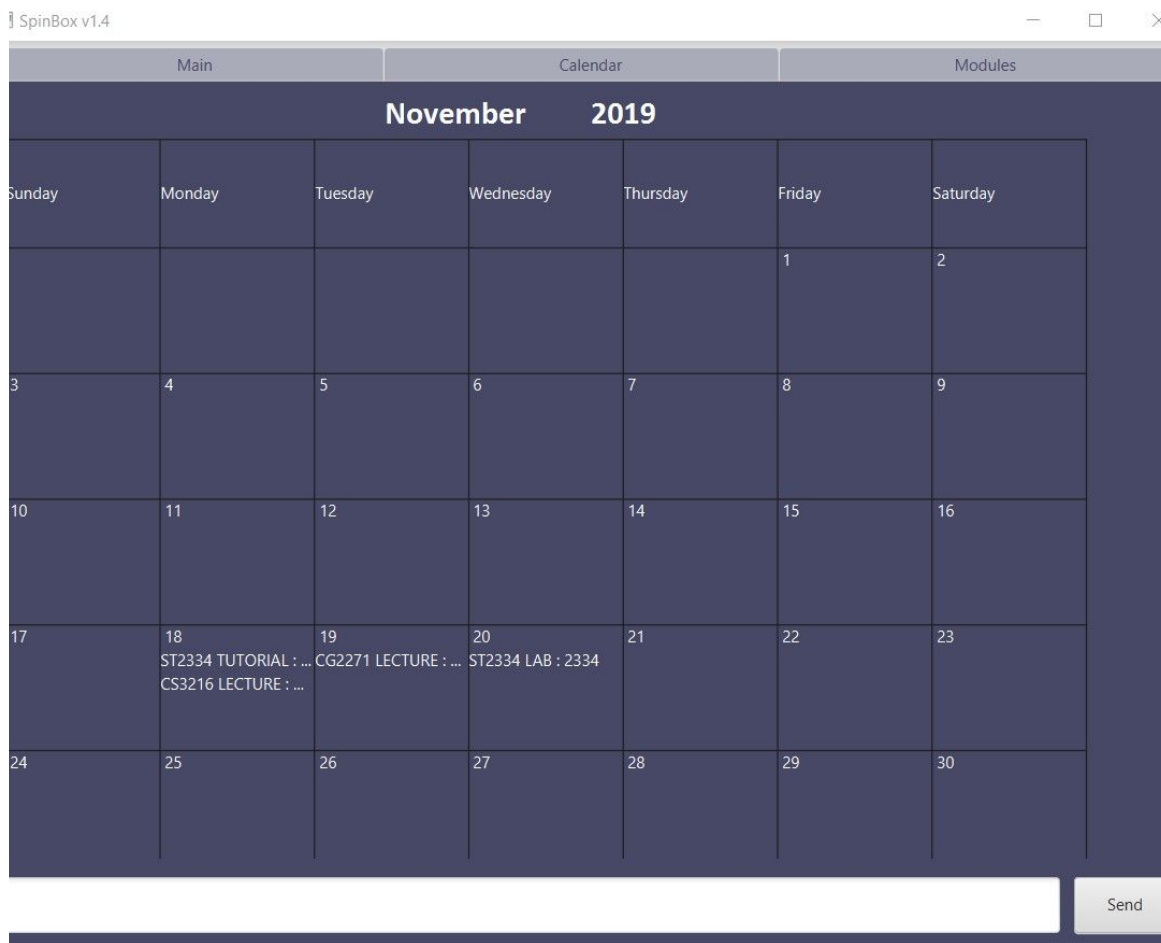


Figure 2: The calendar tab, displaying the schedule for November 2019

3.4. Viewing the modules tab : `view / modules`

Displays the high-level modules tab, which contains a list of the user's added modules. Each module appears as a clickable button, which is one of the ways to access the details of the module.

Format: `view / modules`

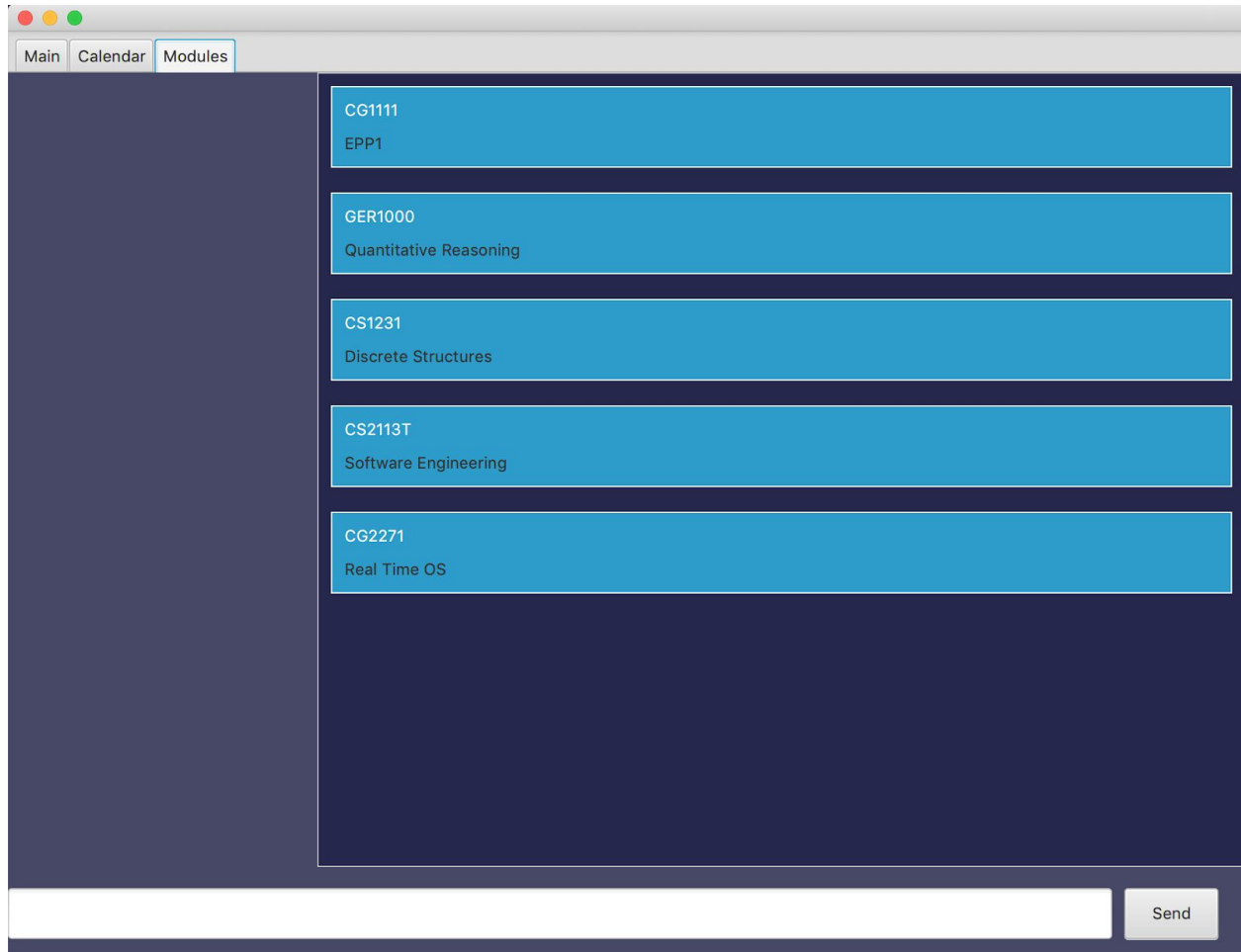


Figure 3: Modules tab, which contains a list of clickable modules

3.4.1. : Adding a module : `add / module`

Add a module to SpinBox.

Note: Module code must be unique, two modules with the same code cannot co-exist within SpinBox.

Format: `add / module [MODULE_CODE] [MODULE_DESCRIPTION]`

Example:

- `add / module CG1111 EPP1`

3.4.2. : Removing a module : `remove / module`

Delete a module from SpinBox.

Format: `remove / module [MODULE_CODE]`

Example:

- `remove / module CG1111`

i Attempting to remove a module that is not existent will result in an error message displayed.

3.5. : Viewing a specific module : `view / modules`

View the overall page for a specific module with the task subtab selected by default.

Format: `view / modules [MODULE_CODE]` or `view / [MODULE_CODE]`

Example:

- `view / modules CG1111` or `view / CG1111`

An example of the expected output is shown in Figure 4.

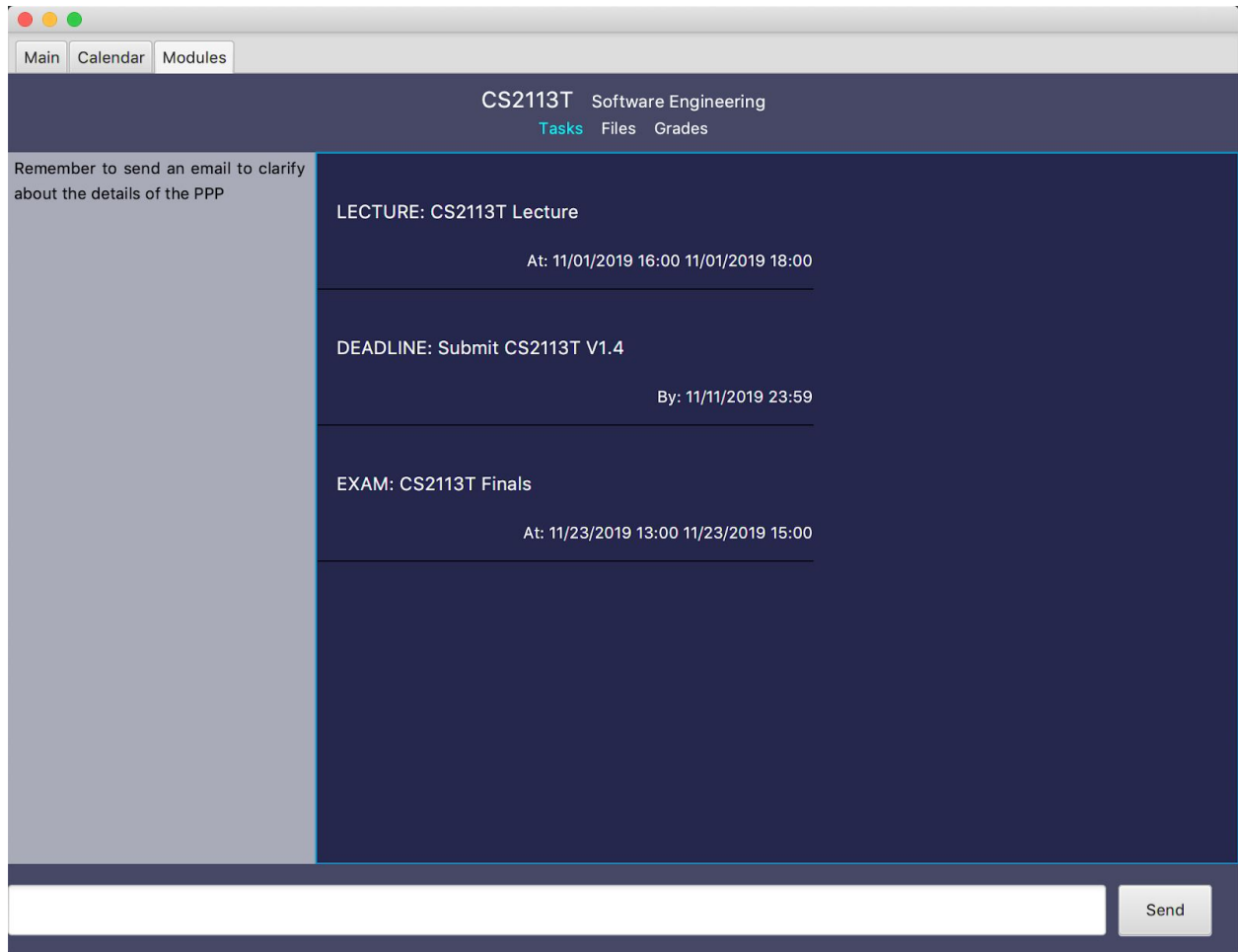


Figure 4: Module-specific view, with the Tasks subtab selected

💡 With the implementation of an intuitive command interpreter in SpinBox, you may provide the input in varying formats and still produce the same intended effect. The formats will be listed out in the command explanations below.

3.5.1. Viewing the Tasks section : `view / tasks`

View the tasks section for the currently selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `view / tasks` or `view / [MODULE_CODE] tasks`

Example:

- `view / tasks` (If CG1111 is currently displayed)

- `view / CG1111 tasks` (If on a different tab / module)

3.5.1.1. Adding a todo task : `add / todo`

Adds a todo type task to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / todo [DESCRIPTION]` or `add [MODULE_CODE] / todo [DESCRIPTION]`

Example:

- `add / todo Return book` (If module is currently displayed)
- `add CG1111 / todo Return book` (If on a different tab / module)

3.5.1.2. Adding a deadline task : `add / deadline`

Adds a deadline type task to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / deadline [DESCRIPTION] by: [DATETIME]` or

`add [MODULE_CODE] / deadline [DESCRIPTION] by: [DATETIME]`

Example:

- `add / deadline Submit assignment by: 10/04/2019 23:59` (If module is currently displayed)
- `add CG1111 / deadline Submit assignment by: 10/04/2019 23:59` (If on a different tab / module)

3.5.1.3. Adding an event task : `add / event`

Adds an event type task to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / event [DESCRIPTION] at: [DATETIME to DATETIME]` or

`add [MODULE_CODE] / event [DESCRIPTION] at: [DATETIME to DATETIME]`

Example:

- `add / event Project Showcase at: 10/05/2019 10:00 to 10/05/2019 12:00` (If module is currently displayed)
- `Add CG1111 / event Project Showcase at: 10/05/2019 10:00 to 10/05/2019 12:00` (If on a different tab / module)

3.5.1.4. Adding an exam event : `add / exam`

Adds an exam event to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / exam [DESCRIPTION] at: [DATETIME to DATETIME]` or

`add [MODULE_CODE] / exam [DESCRIPTION] at: [DATETIME to DATETIME]`

Example:

- `add / exam Finals at: 10/05/2019 10:00 to 10/05/2019 12:00` (If module is currently displayed)
- `add CG1111 / exam Finals at: 10/05/2019 10:00 to 10/05/2019 12:00` (If on a different tab / module)

3.5.1.5. Adding a lab event : `add / lab`

Adds a lab event to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / lab [DESCRIPTION] at: [DATETIME to DATETIME]` or

`add [MODULE_CODE] / lab [DESCRIPTION] at: [DATETIME to DATETIME]`

Example:

- `add / lab Studio 2 at: 10/05/2019 10:00 to 10/05/2019 12:00` (If module is currently displayed)
- `add CG1111 / lab Studio 2 at: 10/05/2019 10:00 to 10/05/2019 12:00` (If on a different tab / module)

3.5.1.6. Adding a lecture event : `add / lecture`

Adds a lecture event to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / lecture [DESCRIPTION] at: [DATETIME to DATETIME]` or

`add [MODULE_CODE] / lecture [DESCRIPTION] at: [DATETIME to DATETIME]`

Example:

- `add / lecture CS2113T Lecture at: 10/05/2019 10:00 to 10/05/2019 12:00` (If module is currently displayed)
- `add CS2113T / lecture CS2113T Lecture at: 10/05/2019 10:00 to 10/05/2019 12:00` (If on a different tab / module)

3.5.1.7. Adding a tutorial event : `add / tutorial`

Adds a tutorial event to the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `add / tutorial [DESCRIPTION] at: [DATETIME to DATETIME]` or

`add [MODULE_CODE] / tutorial [DESCRIPTION] at: [DATETIME to DATETIME]`

Example:

- `add / tutorial CS2113T Tutorial at: 10/05/2019 10:00 to 10/05/2019 12:00` (If module is currently displayed)
- `add CS2113T / tutorial CS2113T Tutorial at: 10/05/2019 10:00 to 10/05/2019 12:00` (If on a different tab / module)

3.5.1.8. Removing a single task : `remove / task`

Removes the specified task from the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `remove / task [INDEX]` or `remove [MODULE_CODE] / task [INDEX]`

- Deletes the task at the specified `[INDEX]`.
- The `[INDEX]` can be found on the GUI, next to each task.

Example:

- `remove / task 2` (Deletes the 2nd task in the list of tasks belonging to the selected module)
- `remove CG1111 / task 2` (Deletes the 2nd task in the list of tasks belonging to the module CG1111)

3.5.1.9. Removing multiple tasks : `remove-* / task`

Removes the specified multiple tasks from the list of tasks under the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `remove-* / task [INDEXES]` or `remove-* [MODULE_CODE] / task [INDEXES]`

- Every `[INDEX]` is separated with a ','
- Deletes the tasks at the specified `[INDEXES]`.
- The `[INDEXES]` can be seen from the GUI, next to each task.

Example:

- `remove-* / task 2,4,6` (deletes the 2nd, 4th and 6th tasks in the list of tasks under the selected module)
- `remove-* CG1111 / task 2,4,6` (deletes the 2nd, 4th and 6th tasks in the list of tasks under CG1111)

3.5.1.10. Updating the status of a task : `update / task`

Update the status of the specific task to the done status value specified of the selected module. Alternatively, use the full version of the command if the module is not currently selected.

Format: `update / task [INDEX] [DONE_STATUS]` or

`update [MODULE CODE] / task [INDEX] [DONE_STATUS]`

- The `[DONE_STATUS]` value is either `done` or `notdone`, where `done` marks the task as done and `notdone` marks the task as not done.
- Update the task at the specified `[INDEX]` to the specified `[DONE_STATUS]` value.
- The `[INDEX]` can be seen from the GUI, next to each task.

Examples:

- `update / task 1 done` (update the 1st task in the currently selected module's list of tasks to the done state)
- `Update CG1111 / task 2 notdone` (updates the 2nd task in the list of tasks under module CG1111 to the not done state)

3.5.1.11. Updating the status of multiple tasks : `update-*` `/ task`

Update the status of the specified tasks under the currently selected module to the done value specified. Alternatively, use the full version if the module is not currently selected.

Format: `update-* / task [INDEXES] [DONE_STATUS]` or

`update-* [MODULE_CODE] / task [INDEXES] [DONE_STATUS]`

- The `[DONE_STATUS]` value is either `done` or `notdone`, where `done` marks all tasks as done and `notdone` marks the task as not done.
- The `[INDEXES]` can be seen from the GUI, next to each task.

Examples:

- `update-* / task 1,3,5 done` (updates the 1st, 3rd and 5th task in the list of tasks of the current module to the done state)
- `update-* CG1111 / task 2,4,6 notdone` (updates the 2nd, 4th and 6th task in the list of tasks under the module CG1111 to the not done state)

3.5.1.12. Editing the name of a task : `set-name / task`

Edit the name of a task under the selected module to a new name. Alternatively, use the full version if the module is not currently selected.

Format: `set-name / task [INDEX] to: [NAME]` or
`set-name [MODULE_CODE] / task [INDEX] to: [NAME]`

- Edits the name of the task at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the command `view / tasks`

Example:

- `set-name / task 1 to: return book` (Name of first task under current module edited to return book)
- `set-name CG1111 / task 2 to: return book` (Name of second task under module CG1111 edited to return book)



This command allows you to edit the description of a task, so that you do not have to manually remove a task and add a new task with the updated description.

3.5.1.13. Editing the date of a schedulable task : `set-date / task`

Edit the date of a schedulable task under the selected module to a new date. Alternatively, use the full version if the module is not currently selected.

Format for single datetime tasks: `set-date / task [INDEX] to: [DATETIME]` or
`Set-date [MODULE_CODE] / task [INDEX] to: [DATETIME]`

Format for double datetime tasks: `set-date / task [INDEX] to: [DATETIME to DATETIME]` or
`set-date [MODULE_CODE] / task [INDEX] to: [DATETIME to DATETIME]`

- Edit the date of the schedulable task at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the GUI, next to each task.

Example:

- `set-date / task 1 to: 01/01/2019 01:00 to 02/01/2019 01:00` (First task under currently selected module set to new datetime)
- `Set-date CG1111 / task 2 to: 01/01/2019 01:00 to 02/01/2019 01:00` (Second task under module CG1111 set to new datetimes)

i Attempting to edit the date of a non-schedulable task (i.e. Todo) will result in an error message.

3.5.2. Viewing the Files section : `view / files`

View the files section for the currently selected module. Alternatively, use the full version if the module is not currently selected. An example of the output is shown below.

Format: `view / files` or `view / [MODULE_CODE] files`

Example:

- `view / files` (View files under currently selected module)
- `view / CG1111 files` (View files under module CG1111)

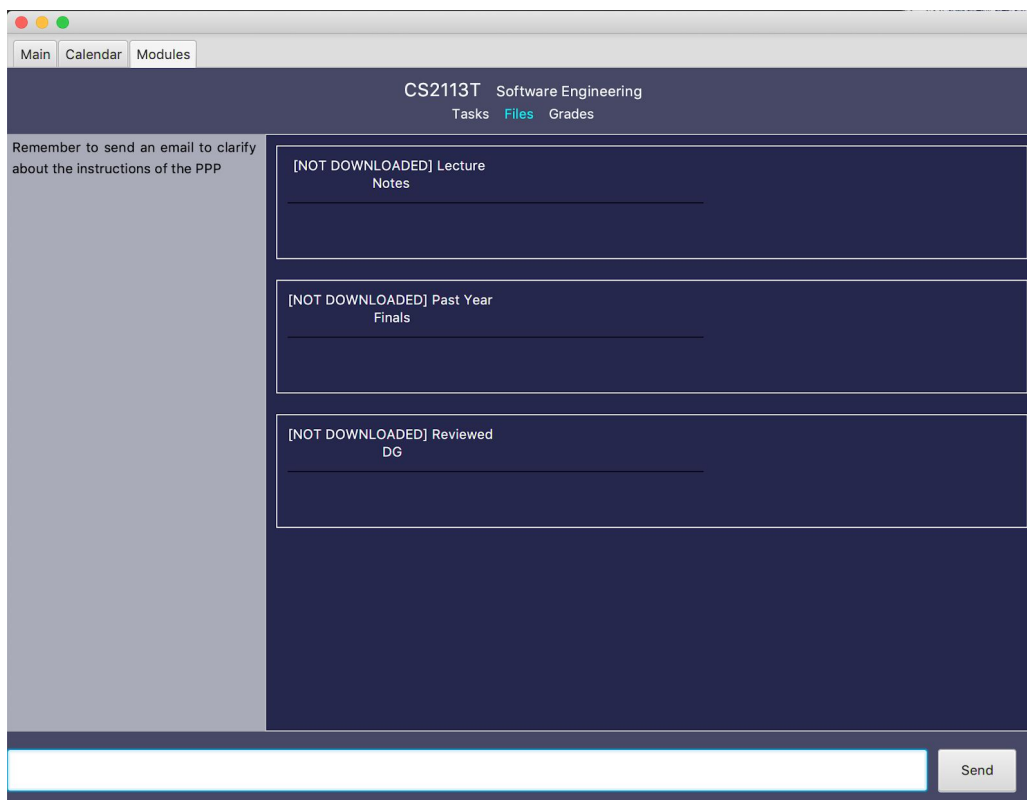


Figure 5: Module-specific view, with the Files subtab selected

3.5.2.1. Adding a file to be tracked : `add / file`

Add an available file to the list of files under the selected module. Alternatively, use the full version if the module is not currently selected.

Format: `add / file [FILENAME]` or `add [MODULE_CODE] / file [FILENAME]`

Example:

- `add / file file1` (Add file1 under currently selected module)
- `add CG1111 / file file1` (Add file1 under module CG1111)

3.5.2.2. Removing a single file : `remove / file`

Removes the specified file from the list of files under the selected module. Alternatively, use the full version if the module is not currently selected.

Format: `remove / file [INDEX]` or `remove [MODULE_CODE] / file [INDEX]`

- Removes the file at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the GUI, next to each file.

Example:

- `remove / file 2` (removes the 2nd file under the currently selected module)
- `Remove CG1111 / file 2` (removes the 2nd file under module CG1111)

3.5.2.3. Removing multiple files : `remove-* / file`

Removes the specified multiple files from the list of files. Alternatively, use the full version if the module is not currently selected.

Format: `remove-* / file [INDEXES]` or `remove-* [MODULE_CODE] / file [INDEXES]`

- Every index is separated with a ','
- Removes the files at the specified `[INDEXES]`.
- The `[INDEXES]` can be seen from the GUI, next to each file.

Examples:

- `remove-* / file 2,4,6` (removes the 2nd, 4th and 6th files from the currently selected module)
- `remove-* CG1111 / file 3,5,7` (removes the 3rd, 5th and 7th files from CG1111)

3.5.2.4. Updating the status of a file : `update / file`

Update the status of the specific file to the done status value specified. Alternatively, use the full version if the module is not currently selected.

Format: `update / file [INDEX] [DONE_STATUS]` or

`update [MODULE_CODE] / file [INDEX] [DONE_STATUS]`

- The `[DONE_STATUS]` value is either `done` or `notdone`, where `done` marks the file as downloaded and `notdone` marks the file as not downloaded.
- Update the file status at the specified `[INDEX]` to the specified `[DONE_STATUS]` value.
- The `[INDEX]` can be seen from GUI, next to each file

Examples:

- `update / file 1 done` (updates the 1st file in the currently selected module to the downloaded state)
- `update CG1111 / file 2 notdone` (Updates the 2nd file in the module CG1111 to the not downloaded state)

3.5.2.5. Updating the status of multiple files : `update-* / file`

Update the status of the specific files to the done status value specified. Alternatively, use the full version if the module is not currently selected.

Format: `update-* / file [INDEXES] [DONE_STATUS]` or

`update-* [MODULE_CODE] / file [INDEXES] [DONE_STATUS]`

- The `[DONE_STATUS]` value is either `done` or `notdone`, where `done` marks the file as downloaded and `notdone` marks the task as not downloaded.
- Updates the file at the specified `[INDEXES]` to the specified `[DONE_STATUS]` value.
- The `[INDEXES]` can be seen from the GUI, next to each file.

Examples:

- `update-* / file 1,3,5 done` (updates the 1st, 3rd and 5th task in the list of tasks to the downloaded state)
- `update-* CG1111 / file 3,5,7 notdone` (updates the 3rd, 5th and 7th task in the list of tasks to the not downloaded state)

3.5.2.6. Editing the name of a file : `set-name / file`

Edit the name of a file under the selected module to a new name. Alternatively, use the full version if the module is not currently selected.

Format: `set-name / file [INDEX] to: [NAME]` or
`set-name [MODULE_CODE] / file [INDEX] to: [NAME]`

- Edits the name of the file at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the command `view / files`

Example:

- `set-name / file 1 to: CS2113T Week 10 Slides` (First file under currently selected module edited)
- `set-name CS2113T / file 1 to: CS2113T Week 10 Slides` (First file under module CS2113T edited)

3.5.3. Viewing the Grades section : `view / grades`

Assessments form the basis for grading student performance in a majority of modules, and SpinBox offers a natural way of keeping track of your progress in these high-stakes assessments - using graded components found under the 'grades' subtab of a module.

Condensed command, if currently viewing a module: `view / grades`

Full command, to specify a module explicitly: `view / modules [MODULE_CODE] grades`

An example of the expected graphical output is shown below.

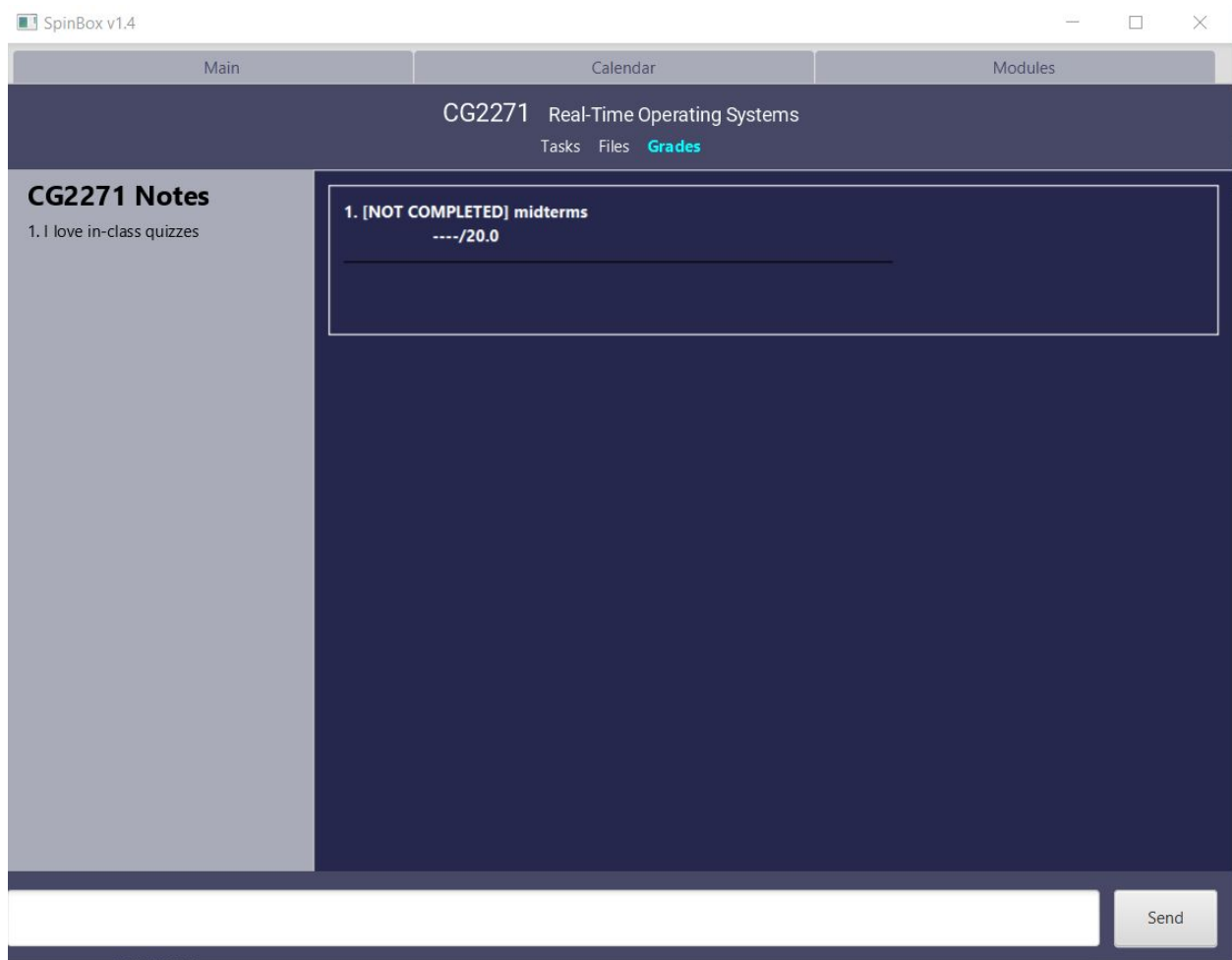



Figure 6: Module-specific view, with the Grades subtab selected

3.5.3.1. Adding a graded component : `add / grade`

Add a new graded component under the selected module. Alternatively, use the full version if the module is not currently selected.

`add / grade [COMPONENT] weightage: [WEIGHTAGE]%` or

`add [MODULE_CODE] / grade [COMPONENT] weightage: [WEIGHTAGE]%`

 : **No restriction on numerical input** for WEIGHTAGE - to accommodate for special case like negative grading, bonus points etc.

Example:

- `add / grade Written Report weightage: 12.5%` (adds a graded component to the currently selected module)
- `add CG1111 / grade Written Report weightage: 12.5%` (adds a graded component to the module CG1111)

3.5.3.2. Updating the status of a graded component : update / grade

This is to allow the user to track his progress throughout the semester. Set a graded component as complete or incomplete under the selected module. Alternatively, use the full version if the module is not currently selected.

Format: `update / grade [INDEX] [DONE_STATUS]` or
`update [MODULE_CODE] / grade [INDEX] [DONE_STATUS]`

- The `[DONE_STATUS]` value is either `done` or `notdone`, where `done` marks the graded component as completed and `notdone` marks the file as not completed.
- Update the grade component at the specified `[INDEX]` to the specified `[DONE_STATUS]` value.
- The `[INDEX]` can be seen from the GUI, next to each grade item.

Example:

- `update / grade 1 done` (updates the status of the 1st graded component under the currently selected module to completed)
- `Update CG1111 / grade 1 notdone` (updates the status of the 1st graded component under the module CG1111 to not completed)

3.5.3.3. Removing a graded component : **remove / grade**

Removes a graded component under the selected module. Alternatively, use the full version if the module is not currently selected.

Format: `remove / grade [INDEX]` or `remove [MODULE_CODE] / grade [INDEX]`


- Removes the grade component at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the GUI, next to each grade item.

Example:

- `remove / grade 1` (removes the 1st graded component under the currently selected module)
- `Remove CG1111 / grade 1` (removes the 1st graded component under the module CG1111)

3.5.3.4. Updating graded component scores with relative values : **score / grade**

You can update a graded component for the currently selected module with the actual scores received. Using this command will mark the graded component as **done**.

 : The marks entered will be automatically be converted to an absolute percentage, using the graded component's weightage as the baseline. E.g. for a Written Report worth 10%, entering 27/36 will be converted into 7.5%.

`score / [INDEX] marks: [XX/YY]` or
`score [MODULE_CODE] / [INDEX] marks: [XX/YY]`

Example:

- `score / 1 marks: 27/36`
(scores the 1st graded component under the currently selected module)
- `score CG1111 / 2 marks: 27/36`
(scores the 2nd graded component under the module CG1111)

3.5.3.5. Updating graded component scores with absolute percentage : score / grade

Update a gradebook component for the currently selected module with the weighted score attained as an overall percentage. Alternatively, use the full version if the module is not currently selected. Additionally, this will mark the graded component as done.

Format: `score / [INDEX] [XX]%` or `score [MODULE_CODE] / [INDEX] [XX]%`

Example:

- `score / 1 marks: 7.5%` (scores the 1st graded component under the currently selected module)
- `score CG1111 / 1 marks: 7.5%` (scores the 1st graded component under the module CG1111)

3.5.3.6. Entering final grade : result / grade [coming in v2.0]

Update the gradebook for the currently selected module with the actual grade received. Also marks all graded components as completed.

Format: `result / grade [INDEX] [X]`

Example:

- `result / grade 1 A+`

3.5.4. Adding a custom note : `add / note`

Add the custom note into the notes section under the selected module. Alternatively, use the full version if the module is not currently selected. An example output is shown below.

Format: `add / note [MESSAGE]` or `add [MODULE_CODE] / note [MESSAGE]`

Example:

- `add / note Remember to send an email to the professor` (adds a note to the currently selected module)
- `add CG1111 / note Remember to send an email to the professor` (adds a note to the module CG1111)

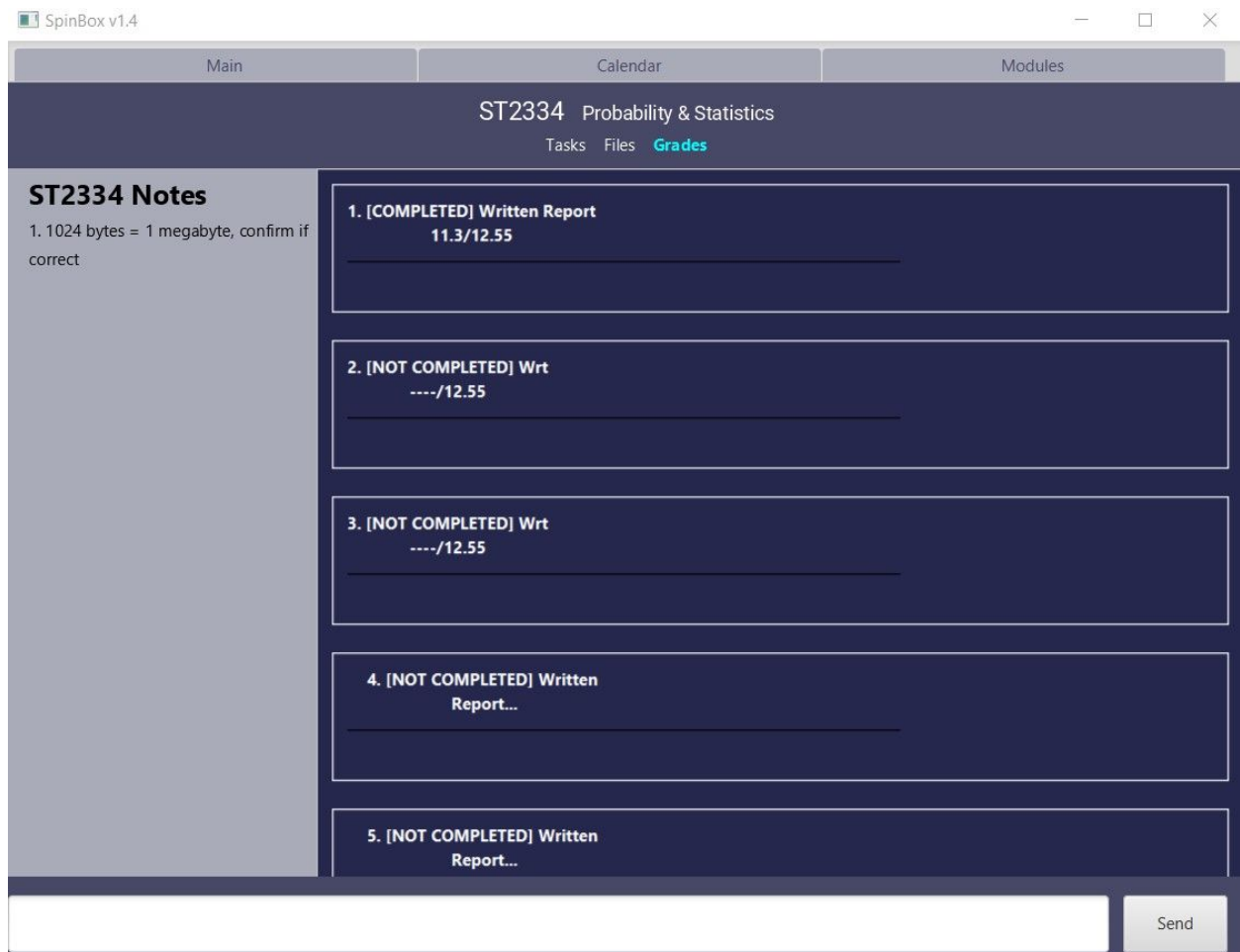


Figure 7: Notes is visible on the left pane, regardless of subtab selected

3.5.5. Removing a custom note : `remove / note`

Remove a custom note under the selected module. Alternatively, use the full version if the module is not currently selected.

Format: `remove / note [INDEX]` or `remove [MODULE_CODE] / note [INDEX]`

- Removes the custom note at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the notes section shown in the application when in any specific modules page.

Example:

- `remove / note 1` (removes the first note from the currently selected module)
- `remove CG1111 / note 1` (removes the first note from the module CG1111)

3.5.6. Editing a custom note : `set-name / note`

Edit a custom note under the selected module. Alternatively, use the full version if the module is not currently selected.

Format: `set-name / note [INDEX] to: [NAME]` or `set-name [MODULE_CODE] / note [INDEX] to: [NAME]`

- Edits the description of the custom note at the specified `[INDEX]`.
- The `[INDEX]` can be seen from the notes section shown in the application when in any specific modules page.

Example:

- `set-name / note 1 to: Send email` (Edits the description of the first note from the currently selected module to Send email)
- `set-name CG1111 / note 1 to: Send email` (Edits the description of the first note from module CG1111 to Send email)

3.6. Viewing the Help page : `help`

Shows the help page, which consists of the brief overview of the available help commands.

Format: `help`

3.6.1. Viewing the Help page on the view command : `help / view`

Shows the view command help page, which details on how the view command can be used.

Format: `help / view`

3.6.2. Viewing the Help page on the add command : `help / add`

Shows the add command help page, which details on how the add command can be used.

Format: `help / add`

3.6.3. Viewing the Help page on the remove command : `help / remove`

Shows the remove command help page, which details on how the remove command can be used.

Format: `help / remove`

3.6.4. Viewing the Help page on the remove-* command :

help / remove-*

Shows the remove-* command help page, which details on how the remove-* command can be used.

Format: `help / remove-*`

3.6.5. Viewing the Help page on the set-name command :

help / set-name

Shows the set-name command help page, which details on how the set-name command can be used.

Format: `help / set-name`

3.6.6. Viewing the Help page on the set-date command :

help / set-date

Shows the set-date command help page, which details on how the set-date command can be used.

Format: `help / set-date`

3.6.7. Viewing the Help page on the update command :

help / update

Shows the update command help page, which details on how the update command can be used.

Format: `help / update`

3.6.8. Viewing the Help page on the update-* command :

help / update-*

Shows the update-* command help page, which details on how the update-* command can be used.

Format: `help / update-*`

3.6.9. Viewing the Help page on the export command :

help / export

Shows the export command help page, which details on how the export command can be used.

Format: `help / export`

3.6.10. Viewing the Help page on the score command :

help / score

Shows the score command help page, which details on how the score command can be used.

Format: `help / score`

3.6.11. Viewing the Help page on the populate command :

help / populate

Shows the populate command help page, which details on how the populate command can be used.

Format: `help / populate`

3.6.12. Viewing the Help page on the find command : `help / find`

Shows the find command help page, which details on how the find command can be used.

Format: `help / find`

3.7. Locating module component by name : `find`

Finds module component with names that contain any of the given keywords within the currently selected module. The output will be a list of items that contains the specified keyword. Alternatively, use the full version if the module is not currently selected.

Format: `find / [ITEM_TYPE] [KEYWORD]` or

`find [MODULE_CODE] / [ITEM_TYPE] [KEYWORD]`

- Possible `[ITEM_TYPE]`: task, file, grade
- The search is case insensitive. e.g `Work` will match `work`
- Only the name is searched.

Examples:

- `find CG1112 / file tutorial` (returns all files in CG1112 containing the word 'tutorial' in its filename)
- `find / task exam` (returns all tasks containing 'exam' in the module that the user is currently on)

3.8. Saving the data

SpinBox data are saved in the hard disk automatically after any command that changes the data.

There is no need to save manually.

3.9. Exporting data : `export`

SpinBox items can be exported at any time for inclusion into a dedicated ecosystem, i.e. an Excel planner. This is achieved via the export command. You can export all tasks, files or grades of a specific module, or deadlines across all modules into a human-readable text file. The exported file will be stored in the SpinBoxData/exports folder.

Format when not within the desired module: `export [MODULE_CODE] / [ITEM_TYPE]`

- Possible `[ITEM_TYPE]`: files/grades/tasks

Example:

- `export CG1111 / grades`

Format when currently viewing desired module: `export / [ITEM_TYPE]`

- Possible `[ITEM_TYPE]`: files/grades/tasks

Example:

- `export / files`

Format to export deadlines across all modules: `export / deadlines`

Example:

3.10. Calculate CAP : `cap`[coming in v2.0]

Calculates a CAP value across modules with an actual grade entered.

- Format: `cap`

3.11. Populate with sample data : `populate`

Populates an empty SpinBox with sample data for the purposes of testing. Has no effect when SpinBox is non empty (has 1 or more modules).

- Format: `populate`

3.12. Logs

SpinBox automatically logs events and user actions for debugging in case things go wrong. The logs can be found within SpinBoxData/spinBox-log.txt, which can be useful for the user to attempt to troubleshoot issues if they occur.

4. FAQ

Q: Where can I find Java version 11?

A: You can download Java version 11 [here](#). We recommend installing Java SE Development Kit 11.0.4 on your computer.

Q: How do I transfer my data to another Computer?

A: Install the app on the other computer and overwrite the empty data file it creates with the file that contains the data of your previous SpinBox folder.

Q: I corrupted the data through manual editing. How do I fix it?

A: Check the logfile to find out more details about where the corrupted data exists. Either delete the respective text file and start over or attempt to fix the error.

5. Command Summary

Category	Command	Command Format and Example
General	bye	Format: bye Example: bye
	view	Format: view / [PAGE] Example: 1. view / main 2. view / calendar
	help	Format: 1. help 2. help / [COMMAND] Example: 1. help 2. help / set-name
	find	Format: 1. find [PAGE_CONTENT] / [ITEM_TYPE] [KEYWORD] Example: 1. find CG1112 / file tutorial 2. find / task exam
	export	Format: 1. export [ITEM_TYPE] 2. export [MODULE_CODE] / [ITEM_TYPE] Example: 1. export / files 2. export CG1111 / grades
	populate	Format: populate Example: populate
	cap [coming in v2.0]	Format: bye Example: bye
Module Section	view	Format: 1. view / modules 2. view / [MODULE_CODE] 3. view / modules [MODULE_CODE] 4. view / tasks

		<ol style="list-style-type: none"> 5. view / [MODULE_CODE] tasks 6. view / files 7. view / [MODULE_CODE] files 8. view / grades 9. view / [MODULE_CODE] grades <p>Example:</p> <ol style="list-style-type: none"> 1. view / modules 2. view / CG1111 3. view / modules CG1111 4. view / tasks 5. view / CG1111 tasks 6. view / files 7. view / CG1111 files 8. view / grades 9. view / CG1111 grades
	add	<p>Format:</p> <p>Adding a module:</p> <ol style="list-style-type: none"> 1. add / module [MODULE_CODE] [MODULE_DESCRIPTION] <p>Adding a task/file/grade/note (include [MODULE_CODE] after add if the module is not currently selected):</p> <ol style="list-style-type: none"> 2. add / todo [DESCRIPTION] 3. add / deadline [DESCRIPTION] by: [DATETIME] 4. add / [TASKTYPE] [DESCRIPTION] at: [DATETIME] <p>* Possible [TASKTYPE]: Event/Exam/Lab/Lecture/Tutorial</p> <ol style="list-style-type: none"> 5. add / file [FILENAME] 6. add / grade [COMPONENT] weightage: [WEIGHTAGE] 7. add / note [MESSAGE] <p>Example:</p> <ol style="list-style-type: none"> 1. add / module CG1111 EPP1 2. add / todo Return book 3. add CG1111 / todo Return book 4. add / deadline Submit assignment by: 10/04/2019 23:59 5. add / event Project Showcase at: 10/05/2019 10:00 to 10/05/2019 12:00 6. add / file file1 7. add / grade Written Report weightage 12.5 8. add / note Send email
	remove	<p>Format:</p> <p>Removing a module:</p>

		<ol style="list-style-type: none"> 1. <code>remove / module [MODULE_CODE]</code> <p>Removing a task/file/grade/note (include [MODULE_CODE] after remove if the module is not currently selected):</p> <ol style="list-style-type: none"> 2. <code>remove / task [INDEX]</code> 3. <code>remove / file [INDEX]</code> 4. <code>remove / grade [INDEX]</code> 5. <code>remove / note [INDEX]</code> <p>Example:</p> <ol style="list-style-type: none"> 1. <code>remove / module CG1111</code> 2. <code>remove / task 1</code> 3. <code>remove CG1111 / task 1</code> 4. <code>remove / file 1</code> 5. <code>remove CG1111 / file 1</code> 6. <code>remove / grade 1</code> 7. <code>remove CG1111 / grade 1</code> 8. <code>remove / note 1</code> 9. <code>remove CG1111 / note 1</code>
	remove-*	<p>Format:</p> <p>Removing multiple tasks/files/grades (include [MODULE_CODE] after remove-* if the module is not currently selected):</p> <ol style="list-style-type: none"> 1. <code>remove-* / task [INDEXES]</code> 2. <code>remove-* / file [INDEXES]</code> <p>Example:</p> <ol style="list-style-type: none"> 1. <code>remove-* / task 1,2</code> 2. <code>remove-* CG1111 / task 1,2</code> 3. <code>remove-* / file 1,2</code> 4. <code>remove-* CG1111 / file 1,2</code>
	update	<p>Format:</p> <p>Updating a task/file/grade (include [MODULE_CODE] after update if the module is not currently selected):</p> <ol style="list-style-type: none"> 1. <code>update / task [INDEX] [DONE_STATUS]</code> 2. <code>update / file [INDEX] [DONE_STATUS]</code> 3. <code>update / grade [INDEX] [DONE_STATUS]</code> <p>Example:</p> <ol style="list-style-type: none"> 1. <code>update / task 1 done</code> 2. <code>update CG1111 / task 1 notdone</code> 3. <code>update / file 1 done</code> 4. <code>update CG1111 / file 1 notdone</code> 5. <code>update / grade 1 done</code>

		6. <code>update CG1111 / grade 1 notdone</code>
	update-*	<p>Format: Updating multiple tasks/files (include <code>[MODULE_CODE]</code> after <code>update-*</code> if the module is not currently selected):</p> <ol style="list-style-type: none"> 1. <code>update-* / task [INDEXES] [DONE_STATUS]</code> 2. <code>update-* / file [INDEXES] [DONE_STATUS]</code> <p>Example:</p> <ol style="list-style-type: none"> 1. <code>update-* / task 1,2 done</code> 2. <code>update-* CG1111 / task 1,2 notdone</code> 3. <code>update-* / file 1,2 done</code> 4. <code>update-* CG1111 / file 1,2 notdone</code>
	set-name	<p>Format: Editing the name of a task/file/note (include <code>[MODULE_CODE]</code> after <code>set-name</code> if the module is not currently selected):</p> <ol style="list-style-type: none"> 1. <code>set-name / task [INDEX] to: [NAME]</code> 2. <code>set-name / file [INDEX] to: [NAME]</code> 3. <code>set-name / note [INDEX] to: [NAME]</code> <p>Example:</p> <ol style="list-style-type: none"> 1. <code>set-name / task 1 to: return book</code> 2. <code>set-name CG1111 / task 2 to: return book</code> 3. <code>set-name / file 1 to: file2</code> 4. <code>set-name CG1111 / file 2 to: file2</code> 5. <code>set-name / note 1 to: note2</code> 6. <code>set-name CG1111 / note 2 to: note2</code>
	set-date	<p>Format: Editing the date of a schedulable task (include <code>[MODULE_CODE]</code> after <code>set-date</code> if the module is not currently selected):</p> <ol style="list-style-type: none"> 1. <code>set-date / task [INDEX] to: [DATETIME]</code> <p>* Possible schedulable tasks: Deadline/ Event/Exam/Lab/Lecture/Tutorial</p> <p>Example:</p> <ol style="list-style-type: none"> 1. <code>set-date / task 1 to: 01/01/2019 01:00 to 02/01/2019 01:00</code> 2. <code>set-date CG1111 / task 1 to: 01/01/2019 01:00 to 02/01/2019 01:00</code> 3. <code>set-date / task 2 to: 01/01/2019 20:00</code> 4. <code>set-date CG1111 / task 2 to: 01/01/2019 20:00</code>

	score	<p>Format: Updating a graded component score with relative values (include [MODULE_CODE] after score if the module is not currently selected):</p> <ol style="list-style-type: none"> 1. score / [INDEX] marks: [XX/YY] <p>Updating a graded component score with absolute percentage (include [MODULE_CODE] after score if the module is not currently selected):</p> <ol style="list-style-type: none"> 2. score / [INDEX] marks: [XX%] <p>Example:</p> <ol style="list-style-type: none"> 1. score / 1 marks: 27/36 2. score CG1111 / 1 marks: 27/36 3. score / 1 marks: 7.5% 4. score CG1111 / 1 marks: 7.5%
	Result [coming in v2.0]	<p>Format: result / grade [INDEX] [X]</p> <p>Example: result / grade 1 A+</p>