User Guide

Team F14-3: RoomShare

Authors:

Teh Zi Huai

Tran Minh Duong, Harry

Tay Yee En, Ryan

Tay Tze-wei, Caleb

Table of Contents

1. Introduction

1.1 What is RoomShare?

RoomShare is a personal assistant which can help you keep track of your mundane tasks while focusing on your other attention-seeking tasks. RoomShare can also share your tasks and dedicate tasks to others making it the perfect shared personal assistant. RoomShare being a simple Command Line Interface (CLI) application ensures its usability and suitability for almost any one. Get started now by referring to chapter 5 for more details on how to use the application.

This is a mockup of our UI:

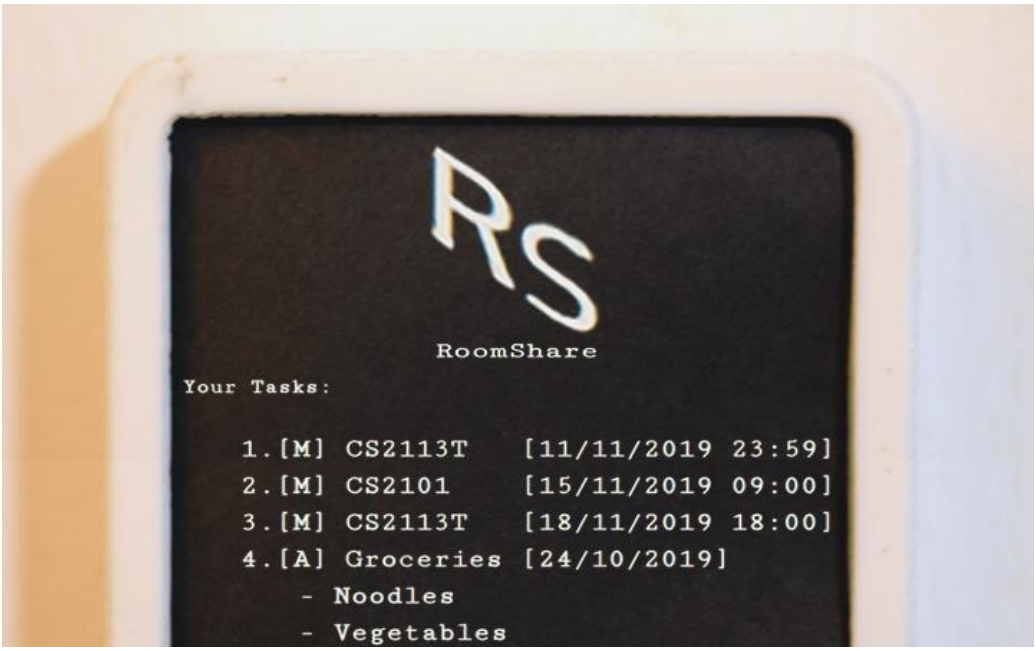

Fig 1.1 The graphical user interface of RoomShare inside hardware

2. About this document

This document is intended for RoomShare users and will explain how to setup RoomShare

Note the following symbols and formatting used in this document:

This symbol indicates important information.

`add`

A grey highlight (called a mark-up) indicates that this is a command that can be typed into the command line and executed by the application.

on your computer, its basic features and how to use them.

3. UI

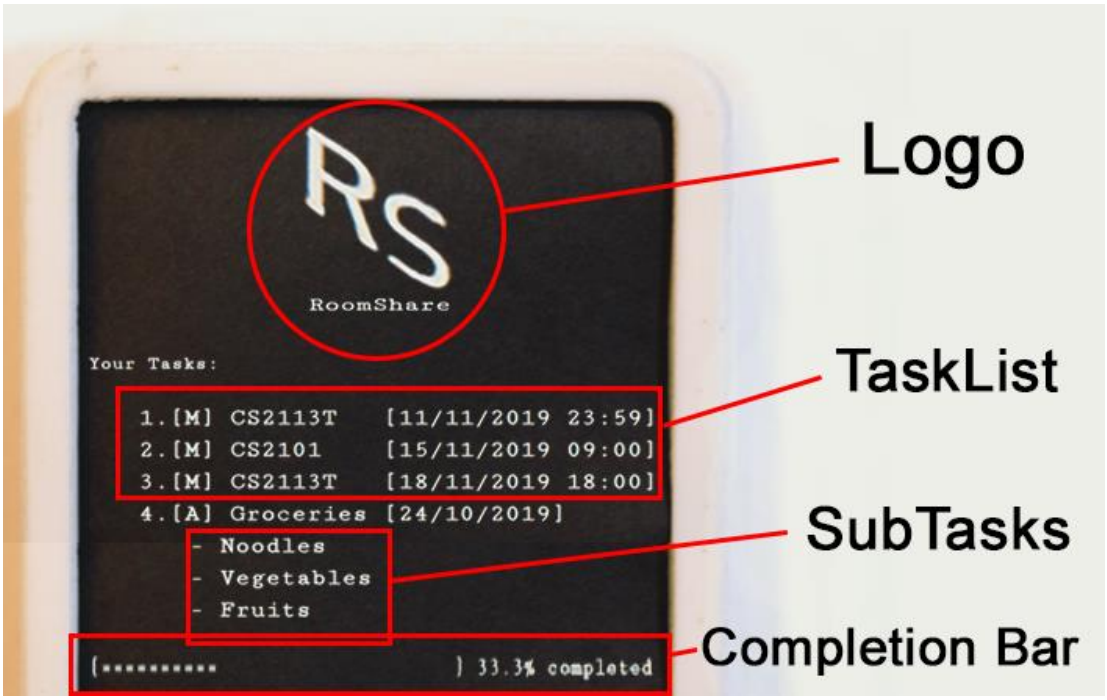

Fig 3.1 User Interface Layout

Above is a picture of the user interface, the descriptions of the different UI features can be found below.

- **Tasklist** - Displays the uncompleted tasks
- **Subtasks** - Assignments can be broken down into different subtasks. Subtasks will be displayed below the
- **Completion Bar** - Displays the percentage of tasks completed

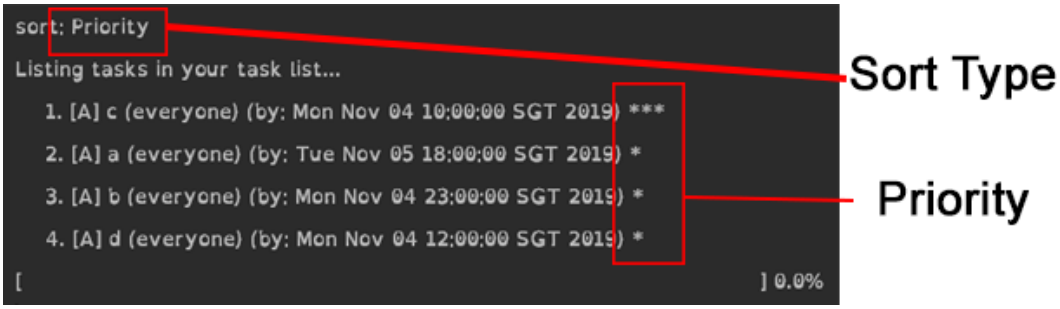

Fig 3.2 Sort type and Priority

- **Sort Type** - Indicates sort type. Priority, Deadline, Alphabetical or Type order.
- **Priority** - Priority of the task.

4. Setting up RoomShare

4.1 Using Jar

Fig 4.1.1 Clicking on Jar file icon

1. Make sure you have Java 11 or above installed in your computer and have set up your JRE appropriately

2. Run the packaged JAR file of RoomShare by double clicking it and a window should appear in a few seconds

3. Type in the command when prompted and press enter to execute it.
   Eg. typing help and pressing Enter will show you a command list and you can learn more about each command by typing it.

4.2 Using Command Line

1. Open CommandPrompt on your computer.

2. Navigate to the folder RoomShare.jar is in. You can do this by typing

"cd" followed by the file path into the terminal as shown below

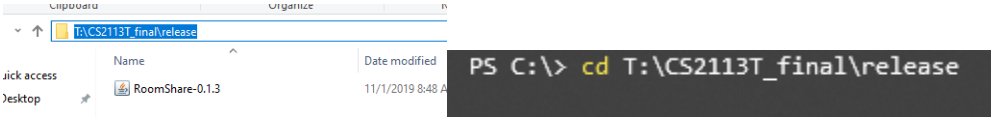

Fig 4.2.1 How to navigate to the correct folder in the command line

3. Type "java -jar" and the name of the jar file and press **Enter**

Fig 4.2.2: How to run the jar file

5. Quick Start

This quick start guide is intended for advanced users who already have the application's environment setup on their computer and wish to jump straight into using the application. For a greater in depth look at how to use the commands, refer to chapter 6.

1. Ensure that you have Java 11 installed

2. Download the latest RoomShare.jar here

3. Copy the file to the folder you wish to have RoomShare in, and double click the file. RoomShare should be brought up in a few seconds

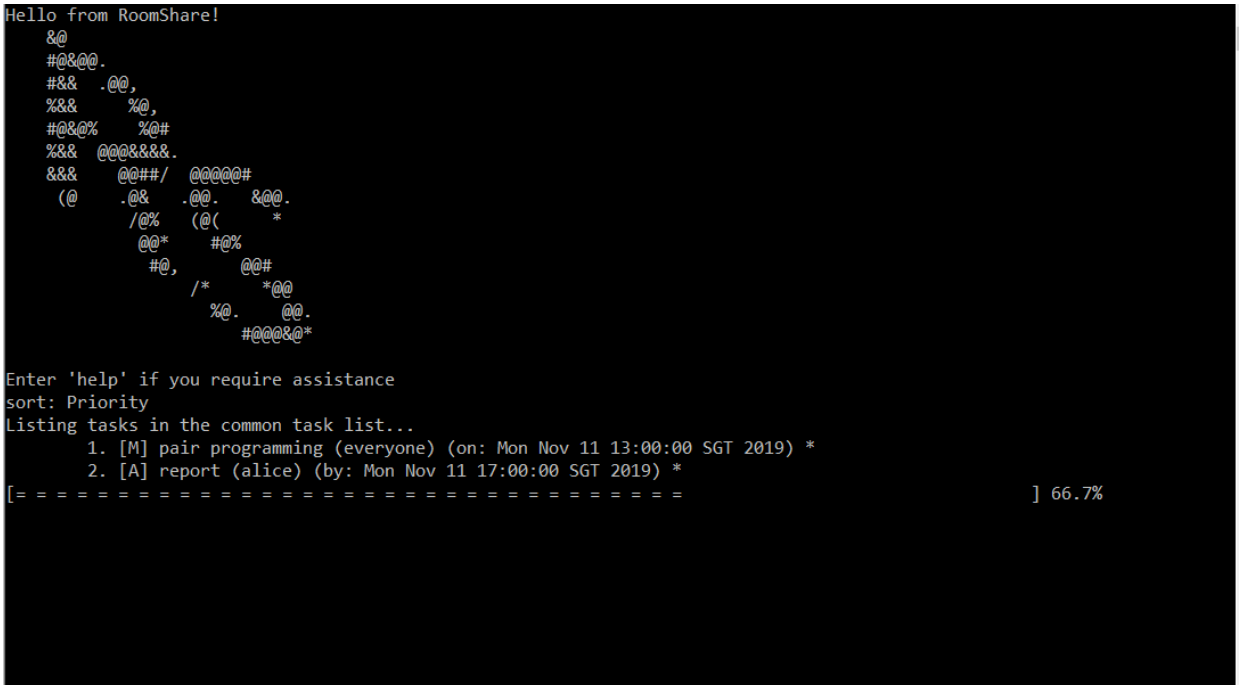

Fig 5.1: RoomShare

4. Type in '`help`' and hit `enter` to bring up the list of commands that can be used

5. Some sample commands you can try:
    - `list`: Refreshes the screen and brings up the list of tasks
    - `add #assignment# (report) &tmr 12:00&`: Adds an assignment titled report which is due tomorrow at 12:00 into the task list
    - `update 1 @me@ *high*`: Updates the assignment in the task list to be tagged to 'me' and sets its priority as high
    - `bye`: Exits the app.

6. Features

6.1 Help [help]

Display the format instructions of a command you need help with.

Format: `help` [COMMAND]
Sample: `help add`
Example:

1. if you need help with any command, for example, command `sort`, type this onto your terminal:

```
help sort
```

Fig 6.1.1: How to use the help command

2. Press **Enter** and you should see the description of how to use `sort` command like this:

```
Sorts the tasks in the task list based on, deadline, priority and alphabetical order
        eg. sort deadline
        This will sort the tasks in the task list by their deadlines
```

Fig 6.1.2: What you would see

6.2 List [list]

Display the current tasks within your task list. This command is mainly used as a way to refresh your screen.

Format: `list`

6.3 Show overdue task [overdue]

Display a separate list of your overdue tasks.

Note: Every time the deadline of a task bypass the current time, it will be automatically removed from the current task list and sent to a separate list of overdue tasks.

Format: `overdue`

Example:

1. Suppose we have a task that is overdue soon, as shown in the figure below. The task that will be overdue soon is task 3.

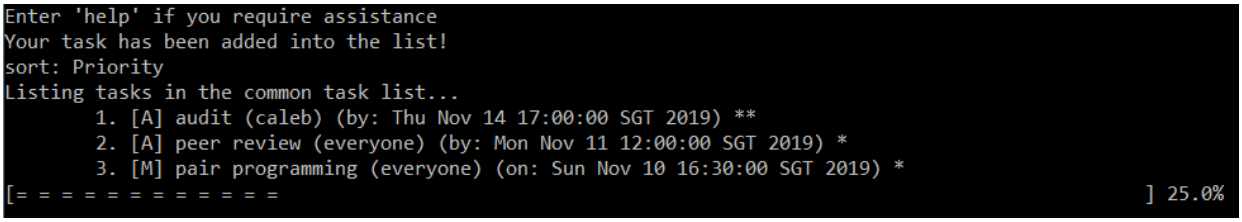

Fig 6.2.1: Sample task list

2. Upon using the `list` command, we find that the task has now been removed from the common task list, as shown in the figure below.

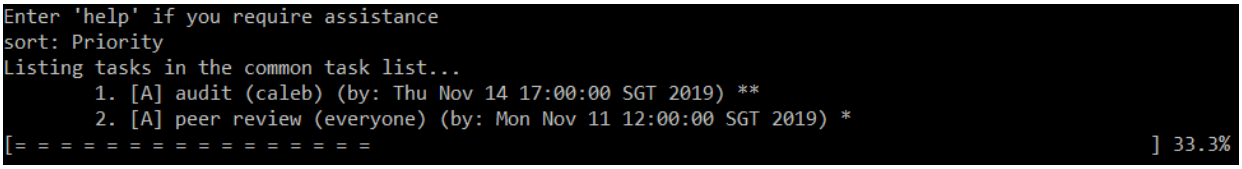

Fig 6.2.2: Overdue task has been removed

3. By using the `overdue` command, we can access the overdue list.

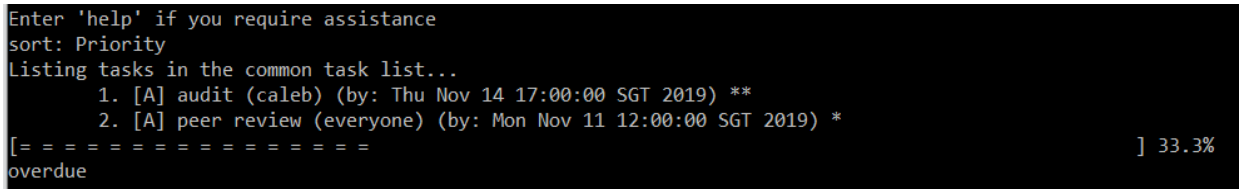

Fig 6.2.3: Using the overdue command

4. We can now see the `overdue` task has been added into the `overdue` list.

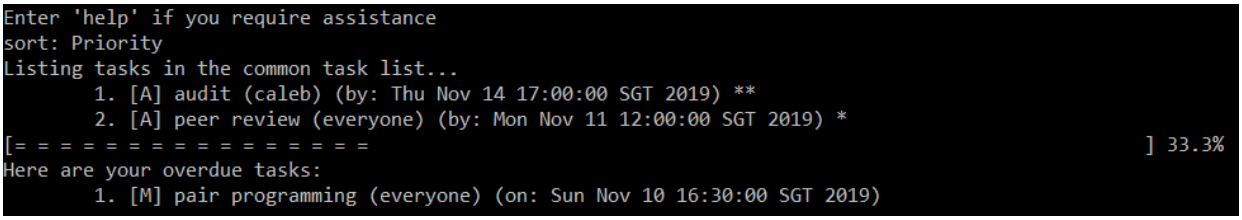

Fig 6.2.4: Overdue list is shown

To know more about how to manipulate `overdue` tasks, see sections **6.17** and **6.18** on rescheduling and removing overdue tasks.

6.4 Show completed task [completed]

Display a list of your completed tasks. This list continues on from the common task list.

Note: Every time a task is set as done, it will be automatically sent to the bottom of the common task list and will only be revealed when you type the `completed` command.

Format: `completed`

Example:

1. To see the completed list, enter the command completed.

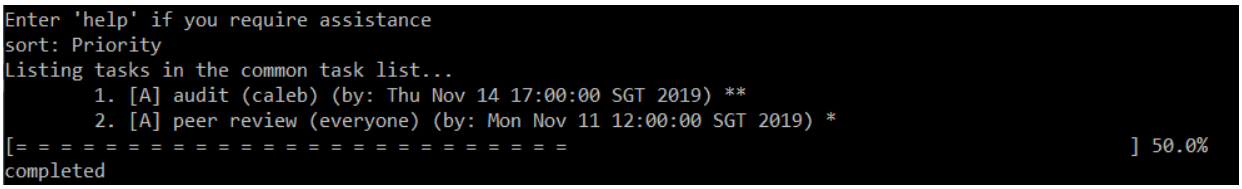

Fig 6.4.1: Using the completed command

2. The completed task list will be shown.

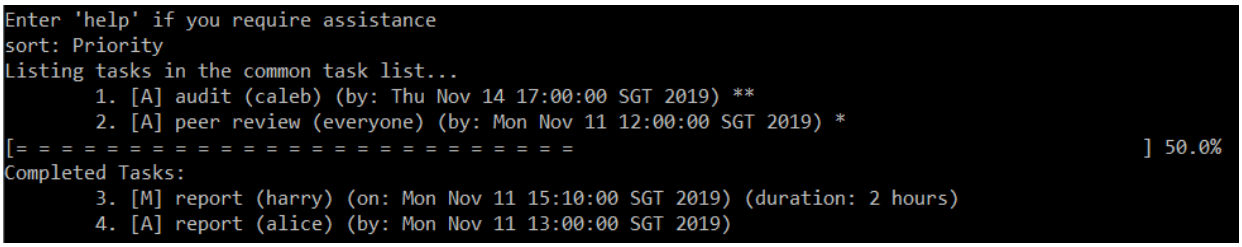

Fig 6.4.2: Output of the command

Take note of how the indices are running on from the incomplete tasks to the completed tasks, as they actually share the same task list.

6.5 Add [add]

Add a new task into your schedule and you can use different tag symbols to represent different information. These tags includes:

- *#TASK_TYPE#*: the type of task the user wants to enter
- *(DESCRIPTION)*: the description of the task
- *&DATE AND TIME&*: the date and time of the task
- **PRIORITY**: the priority of the task
- *^DURATION^*: the duration of the task
- *@ASSIGNEE@*: the person the task is assigned to
- *%RECURRENCE%*: the repeating schedule of the task

All of this information do not have to come in any specific order, other than the `add` command coming first.

Notes: take extra care not to use any of the mentioned special characters in your input for the description and assignee.

Format: `add #TASK_TYPE# &DATE_AND_TIME& (DESCRIPTION) *PRIORITY* ^DURATION^ @ASSIGNEE@ %RECURRENCE%`

#TASK_TYPE# - Assignment, Meeting, Leave
(DESCRIPTION) - description of the task
&DATE_AND_TIME& - DD/MM/YYYY HH:MM or next [DAY]/today/tommorow/ HH:MM
PRIORITY - can only be the following values: high, medium, (default) low
*^DURATION^* - duration of the task in hours or minutes
@ASSIGNEE@ - the person the task is assigned to
%RECURRENCE% whether the task is repeated daily, weekly, or monthly

Sample: `add #assignment# (task report) &tmr 17:00& *high* @ryan@`

Example 1:

1. Type this command into your terminal:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
_____________________________________________________________________________
        List is empty
_____________________________________________________________________________

[                                                                          ] 0.0%
add #meeting# (project brainstorming) &tmr 14:00& ^2 hours^
```

Fig 6.5.1: Sample input for Meeting

➔ The #meeting# tag means the new task added is a meeting.
➔ The (project brainstorming) tag is the description of the meeting.
➔ The &tmr 14:00& tag is the time of the meeting, which is tomorrow14:00.
➔ The ^2 hours^ tag is the duration of the meeting, which is 2 hours.

2. After typing the command and press **Enter**, the task described above will appear in your task list like this:

```
Enter 'help' if you require assistance
Your task has been added into the list!
sort: Priority
Listing tasks in your task list...
        1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
[                                                                                                    ] 0.0%
```

Fig 6.5.2: What you would see

Example 2:

1. Type this command into your terminal:

```
Enter 'help' if you require assistance
Your task has been added into the list!
sort: Priority
Listing tasks in your task list...
        1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
[                                                                                                    ] 0.0%
add #assignment# (submit report) &this fri 16:00& @Harry@
```

Fig 6.5.3: Sample input for Assignment

- ➔ The #assignment# tag means the new task added is an assignment.
- ➔ The (submit report) tag is the description of the assignment.
- ➔ The &this fri 16:00& tag is the deadline of the assignment, which is this friday 16:00.
- ➔ The @Harry@ tag means the assignment is assigned to a person named Harry.

2. Press **Enter** and the task described above will appear in your task list like this:

```
Enter 'help' if you require assistance
Your task has been added into the list!
sort: Priority
Listing tasks in your task list...
        1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
        2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
[                                                                                                    ] 0.0%
```

Fig 6.5.4: What you would see

Example 3:

1. Type this command into your terminal:

```
Enter 'help' if you require assistance
Your task has been added into the list!
sort: Priority
Listing tasks in your task list...
        1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
        2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
[                                                                                                    ] 0.0%
add #leave# (vacation) &21/11/2019 08:00&27/11/2019 18:00& @kelly@
```

Fig 6.5.5: Sample input for Leave

- ➔ The #leave# tag means the task is of the leave task type.
- ➔ (vacation) tag is the description of the leave period.
- ➔ &21/11/2019 08:00&27/11/2019 18:00& tag is the start and end dates of the leave period.
- ➔ @kelly@ tag means the task is assigned to a person named kelly.

Note: For Leave task type you must specify an assignee.

2. Press **Enter** and the task described above will appear in your task list:

```
Enter 'help' if you require assistance
Your task has been added into the list!
sort: Priority
Listing tasks in your task list...
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
       2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
       3. [L] vacation (kelly) (From: Thu Nov 21 08:00:00 SGT 2019 To: Wed Nov 27 18:00:00 SGT 2019) *
[                                                                                                  ] 0.0%
```

Fig 6.5.6: What you would see

6.6 Update [update]

Update a task based on the new corresponding fields from your input. These fields are in the same "tag" format as described in command **6.1 Add [add]**, except the field *#TASK_TYPE#* won't be recognized for this command as you cannot change the type of a task.

Format: `update` [INDEX] *&DATE_AND_TIME& (DESCRIPTION) *PRIORITY* ^DURATION^ @ASSIGNEE@ %RECURRENCE%*

Sample: update 1 @Alice@ (update report) *high*

Example:

1. If you want to change the description, date, or assignee of a task, for example task number 3 in this list:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
       2. [L] vacation (kelly) (From: Thu Nov 21 08:00:00 SGT 2019 To: Wed Nov 27 18:00:00 SGT 2019) *
       3. [A] submit report (harry) (by: Fri Nov 15 18:00:00 SGT 2019) *
[                                                                                                  ] 0.0%
```

Fig 6.6.1: Sample task to be updated

2. Type this command onto your terminal:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) *
       2. [L] vacation (kelly) (From: Thu Nov 21 08:00:00 SGT 2019 To: Wed Nov 27 18:00:00 SGT 2019) *
       3. [A] submit report (harry) (by: Fri Nov 15 18:00:00 SGT 2019) *
[                                                                                                  ] 0.0%
update 3 (update report) @Alive@
```

Fig 6.6.2: Updating task number 3

➔ The number `3` is the index of the task inside your task list that you want to update.

➔ The description of the task will be changed from `submit report` to `update report`.

➔ The assignee will be changed from `Harry` to `Alive`.

3. Press **Enter** and the updated task number 3 will look like this:

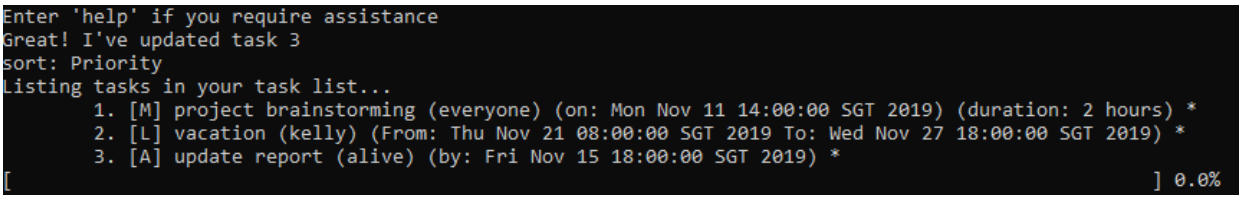

Fig 6.6.3: Task number 3 has been updated

Note: **updating other fields of a task is also similar, but you need to follow the tag format as described above.**

6.7 Snooze [snooze]

Snooze the task you want to select and you must specify how long you want to snooze.

Format: `snooze` [INDEX] [AMOUNT] [UNIT OF TIME]
Sample: snooze 1 2 hours
Example:

1. if you want to `snooze` a task by 2 hours, for example task number 3 in this list:

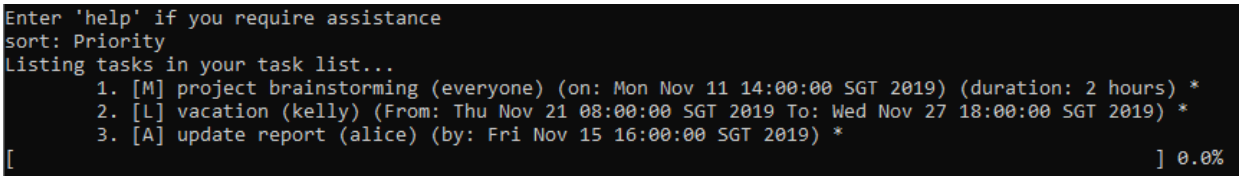

Fig 6.7.1: Sample task to snooze

2. Type this command onto your terminal:

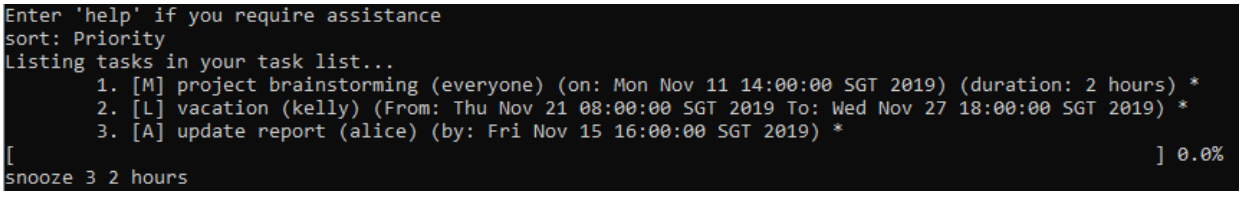

Fig 6.7.2: Using the snooze command

➔ The number `3` is the index of the task inside your task list that you want to snooze.
➔ The number `2` is the amount of time you want to snooze your task.
➔ The word `hours` is the time unit you want to snooze your task.

3. Press **Enter** and the snoozed task should look like this:

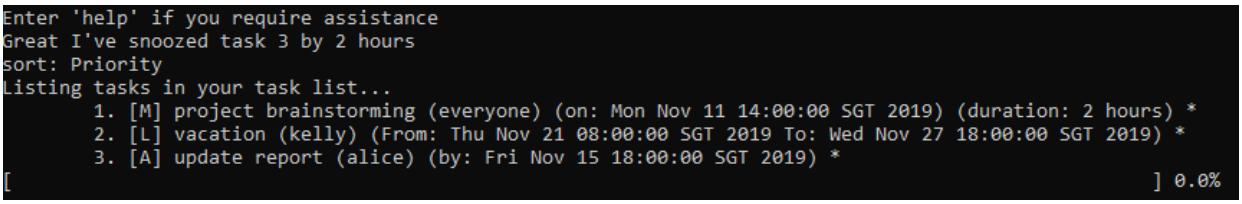

Fig 6.7.3: snoozed task

Notice the deadline of task number 3 has been changed from `(by: Fri Nov 15 16:00:00 SGT 2019)` to `(by: Fri Nov 15 18:00:00 SGT 2019)`.

6.8 Done [done]

Marks the selected task as completed inside your task list. Done tasks will be visually removed from the task list, but it will still be part of the task list.

This command supports ranged based operations.

Format: `done` [INDEX] or `done` [INDEX] - [INDEX]

Sample:

- done 1
- done 1 - 3

Example:

1. If you want to mark a task as done, for example task **number 2** in this list:

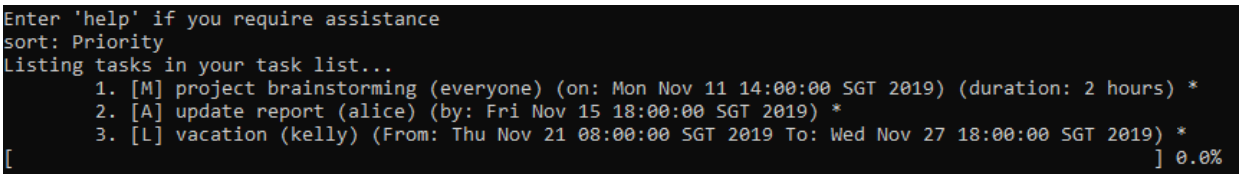

Fig 6.8.1: Sample task to be marked as done

2. Type this command onto your terminal:

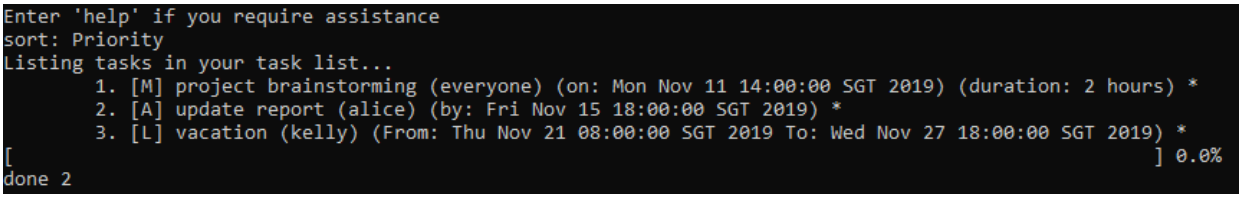

Fig 6.8.2: Marking task number 2 as done

3. Press **Enter** and the task will be removed from the current task list and sent to the list of the completed tasks (as mentioned in **6.4 Show completed task [completed]**).

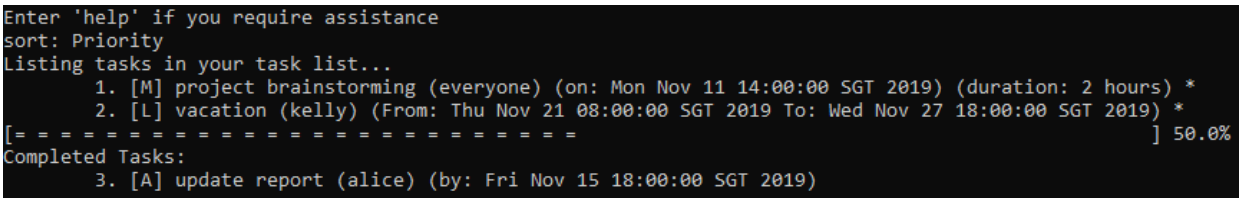

Fig 6.8.3: Previous task number 2 sent to Completed Tasks

6.9 Delete [delete]

Removes the task at the specific index you enter.

Format: `delete` [INDEX]

Or `delete` [INDEX - INDEX]

Sample:

- Delete 1
- Delete 2 - 3

Example:

1. if you want to delete a task inside your task list, for example task number 1 inside this task list:

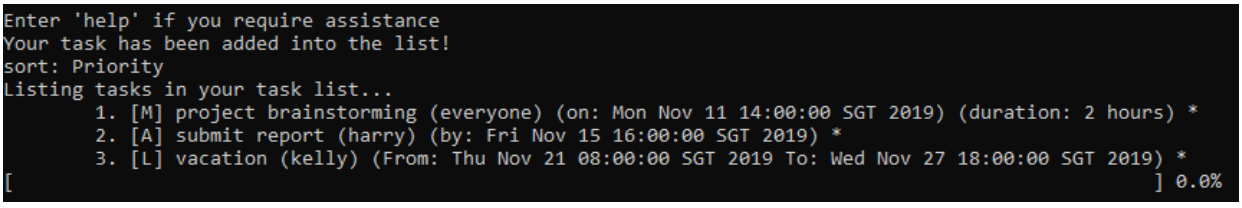

Fig 6.9.1: Sample task list

2. Type this command onto your terminal:

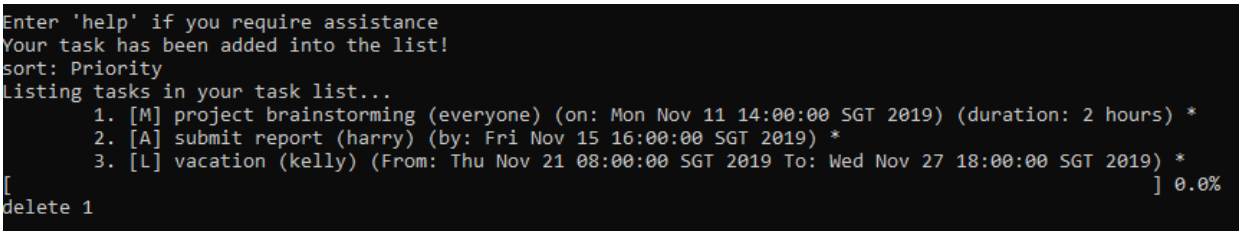

Fig 6.9.2: Delete task number 1

3. Press **Enter** and the resulting list should look like this:

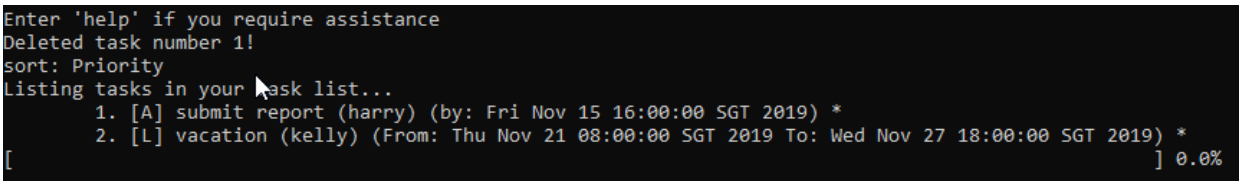

Fig 6.9.3: Task has been deleted

To restore a deleted task, refer to section **6.19 Restore [restore].**

6.10 Find [find]

Looks through the task list to find tasks containing the words you enter. You must enter at least 3 letters into the [KEYWORD] field.

Format: `find` [KEYWORD]
Sample: find report
Example:

1. if you want to find any task containing the word `"report"`, type this onto your terminal:

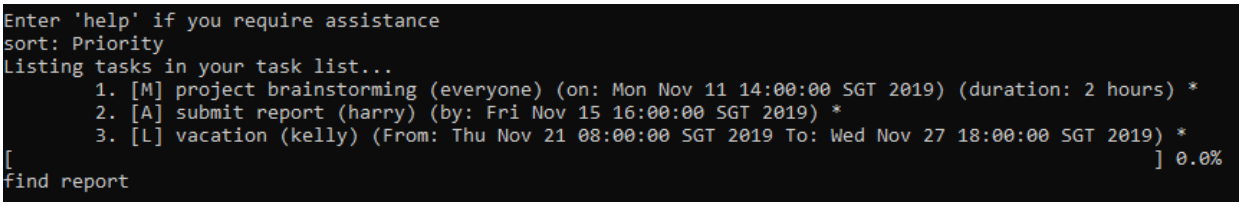

Fig 6.10.1: using the find command

2. Press **Enter** and the matching tasks should appear like this:

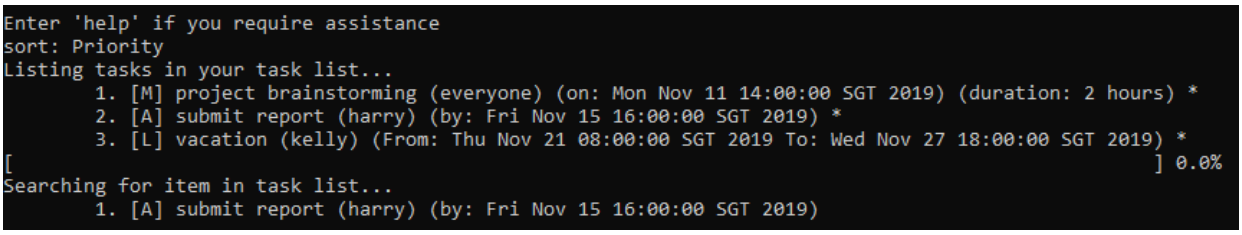

Fig 6.10.1: Results of finding

6.11 Sort [sort]

Sort the data based on the style of order that you enter: alphabetical, priority, deadline, type.
Format: `sort` [ORDER_TO_SORT]
Sample: sort alphabetical
Example:

1. if you want to sort your task list in alphabetical order with an initial list like this:

```
Enter 'help' if you require assistance
Reordering the task list...
sort: Priority
Listing tasks in your task list...
        1. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019) *
        2. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019) *
        3. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019) *
[                                                                                ] 0.0%
```

Fig 6.11.1: Sample task list

2. Type this command onto your terminal:

```
Enter 'help' if you require assistance
Reordering the task list...
sort: Priority
Listing tasks in your task list...
        1. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019) *
        2. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019) *
        3. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019) *
[                                                                                ] 0.0%
sort alphabetical
```

Fig 6.11.2: changing the sort order

3. Press **Enter** and the resulting task list should be sorted alphabetically like this:

```
Enter 'help' if you require assistance
Your sorting preferences have been set to alphabetical
sort: Alphabetical
Listing tasks in your task list...
        1. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019) *
        2. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019) *
        3. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019) *
[                                                                                ] 0.0%
```

Fig 6.11.3: new sorting order

6.12 Adding Subtasks [subtask]

Adds subtasks into the specified assignment.

This command allows you to add a list of subtasks to an `Assignment` in your task list. Only `Assignment type` tasks can have a subtask, so do make sure that you are adding a subtask to an `Assignment`. RoomShare will prevent you from adding subtasks to `Meetings` and `Leaves`.

Notes: Similar to the `add` function, do not use any special characters in the naming of the subtasks.

Format: `subtask` [INDEX_OF_ASSIGNMENT] [LIST_OF_SUBTASKS]

Multiple subtasks can be added in one command to a single `Assignment`. These subtasks must be separated by a comma (“,”) so RoomShare knows that they are different subtasks.

Sample: subtask 1 task1, task2, task3
Example:

1. If you have a big assignment and you want to break it down into smaller subtasks, for example this assignment number 2 in this list:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
        1. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019) *
        2. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019) *
        3. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019) *
[                                                                                ] 0.0%
```

Fig 6.12.1: Sample Assignment

2. Type this command onto your terminal:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
        1. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019) *
        2. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019) *
        3. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019) *
[                                                                                ] 0.0%
subtask 2 buy battery, buy 3D printer, buy laser cutter
```

Fig 6.12.2: adding a subtask

➔ The number `2` is the index of the assignment that you want to break down into smaller subtasks.
➔ `Buy battery, buy 3D printer, buy laser cutter` are the subtasks that you want to add.

3. Press **Enter** and you should see the subtasks listed below the main task like this:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
        1. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019) *
        2. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019) *
                - buy battery
                - buy 3d printer
                - buy laser cutter
        3. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019) *
[                                                                                ] 0.0%
```

Fig 6.12.3: output of the command

6.13 Set Priority [priority]

This command allows the user to change the priority of a task.

Format: `priority` [INDEX_OF_TASK] [NEW_PRIORITY]
Sample: priority 1 high

Example:

1. The priority of a task is always set to low unless specified:

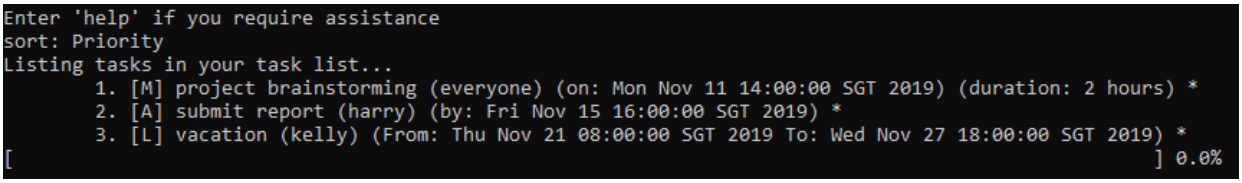

Fig 6.13.1: default priority

2. To change the priority of task number 1 to high, type the following command onto your terminal:

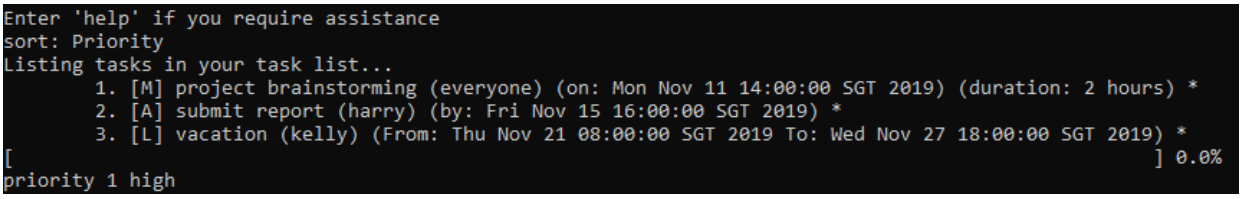

Fig 6.13.2: setting a new priority to task number 1

3. Press **Enter** and you should see the priority of task number 1 updated.

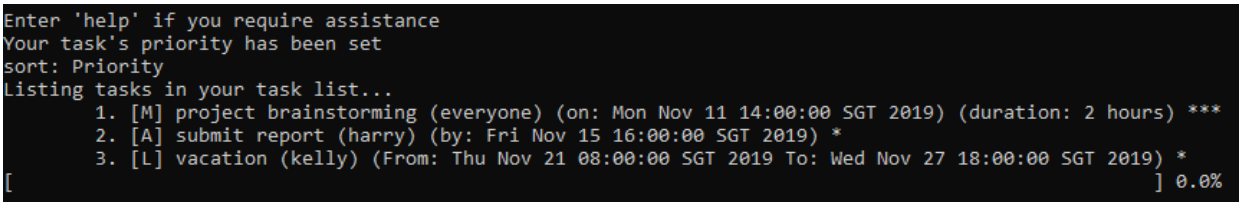

Fig 6.13.3: new priority of the task

Notes: Changing the priority of a task may also change the order of the task list if the current sorting preference is priority.

6.14 Log [log]

This command helps you save a text file containing your current task list.

Format: `log`

Example:

1. Type `log` onto your terminal:

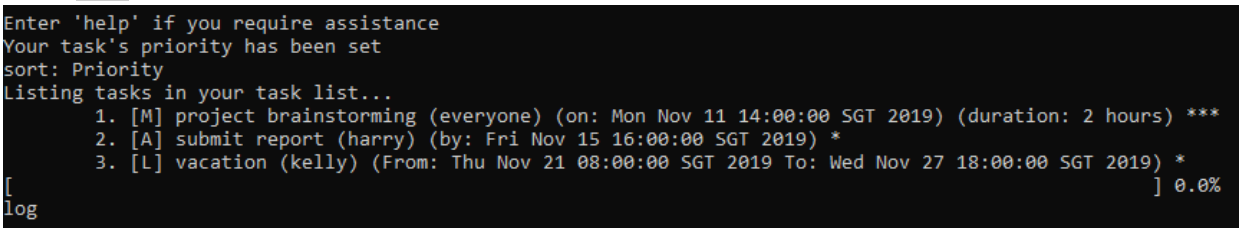

Fig 6.14.1: typing the log command

2. Press **Enter**, the terminal should show you a message like this:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) ***
       2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
       3. [L] vacation (kelly) (From: Thu Nov 21 08:00:00 SGT 2019 To: Wed Nov 27 18:00:00 SGT 2019) *
[                                                                                                    ] 0.0%
Log has been successfully written to logs\log_Sun_Nov_10_20_49_15_SGT_2019.txt
```

Fig 6.14.2: output of the log command

3. Inside the ***logs*** folder, which is located in your ***RoomShare*** program folder, you should find the corresponding text data file when browsing with Windows File Explorer:

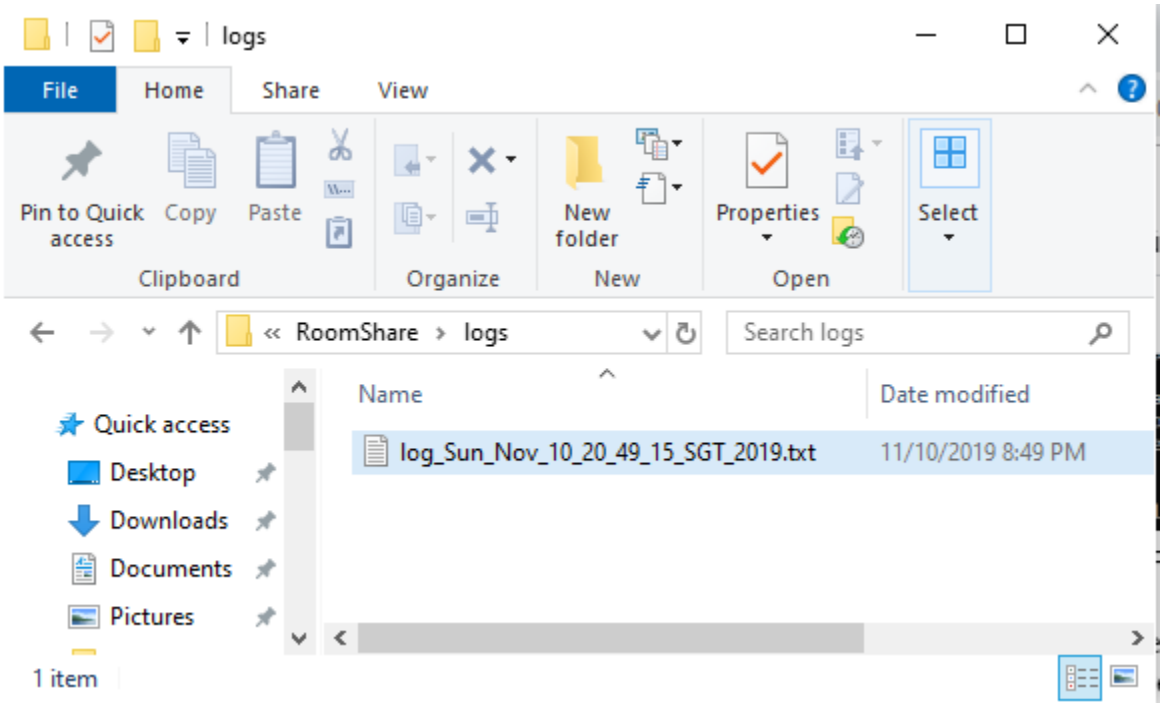

Fig 6.14.2: location of the log file

6.15 Save Data

Any tasks that are in the task list will automatically be saved in the hard disk as a text file. There is no need to save each task manually.

Notes: you **MUST NOT manually edit the "data.txt" file and the "overdue.txt" file** as RoomShare cannot run properly if these files are corrupted. You will risk losing all your previously stored data as a result.

6.16 Showing by Assignee [show]

RoomShare will allow you to see what tasks are assigned to each individual. To see the assigned task, use the command `show`, followed by the user's name. By default, tasks that are assigned to everyone will also show up as tagged to the specified user. This command will also show the completion status of the tagged person's tasks.

On using this command, a list of tasks that are tagged to the specified user will be shown

Format: `show` [NAME _OF_USER]
Sample: show Alice
Example:

1. Suppose you have a task list as shown in the figure below, and you wish to see the tasks tagged to Harry

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) ***
       2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
       3. [L] vacation (kelly) (From: Thu Nov 21 08:00:00 SGT 2019 To: Wed Nov 27 18:00:00 SGT 2019) *
[                                                                                                    ] 0.0%
```

Fig 6.16.1: Sample task list

2. Type this command onto your terminal:

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in your task list...
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) ***
       2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
       3. [L] vacation (kelly) (From: Thu Nov 21 08:00:00 SGT 2019 To: Wed Nov 27 18:00:00 SGT 2019) *
[                                                                                                    ] 0.0%
show harry
```

Fig 6.16.2: show command

3. Youwill now see the task list will display all the tasks assigned to Harry. Take note that the task tagged everyone is also displayed, as well as the completion status for the tasks assigned to Harry:

```
Enter 'help' if you require assistance
These are the tasks assigned to harry:
       1. [M] project brainstorming (everyone) (on: Mon Nov 11 14:00:00 SGT 2019) (duration: 2 hours) ***
       2. [A] submit report (harry) (by: Fri Nov 15 16:00:00 SGT 2019) *
The completion status for 'harry' is:
[                                                                                                    ] 0.0%
```

Fig 6.16.3: Output of the show command

Additionally, you can show all the deleted tasks by specifying the second field as `deleted` instead of a name. This will result in the output as shown below.

```
Here are the tasks that you have deleted and are in temporary storage
       1. [A] attach documents to report (hannah) (by: Mon Nov 11 14:00:00 SGT 2019)
       2. [A] buy equipment (ryan) (by: Wed Nov 13 15:00:00 SGT 2019)
              -buy battery
              -buy 3d printer
              -buy laser cutter
       3. [M] op2 (everyone) (on: Fri Nov 15 13:00:00 SGT 2019)
```

Fig 6.16.4: Output of show deleted

6.17 Reschedule [reschedule]

This command allows you to reschedule tasks that have been set as `overdue`. By using this command, you can specify the index of the overdue task to be rescheduled and give the task a new deadline. The overdue task will then be added into the common task list.

Format: `reschedule` [INDEX_OF_OVERDUE_TASK] [DATE]
Sample: reschedule 1 &tmr 17:00&
Example:

1. Suppose you have an overdue task we wish to reschedule

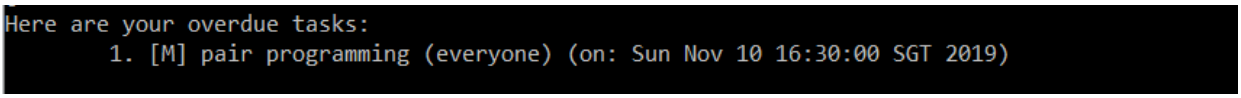

Fig 6.17: overdue task to be rescheduled

2. Use the reschedule command to reschedule the task to a later date

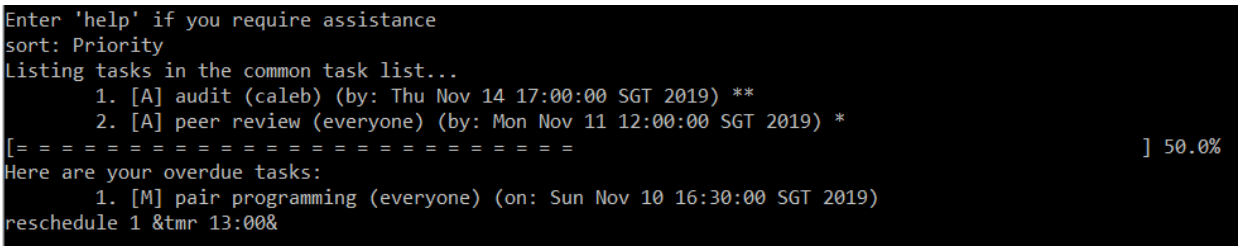

Fig 6.17.1: Using the reschedule command

3. On hitting enter, we see the common task list has the rescheduled task added.

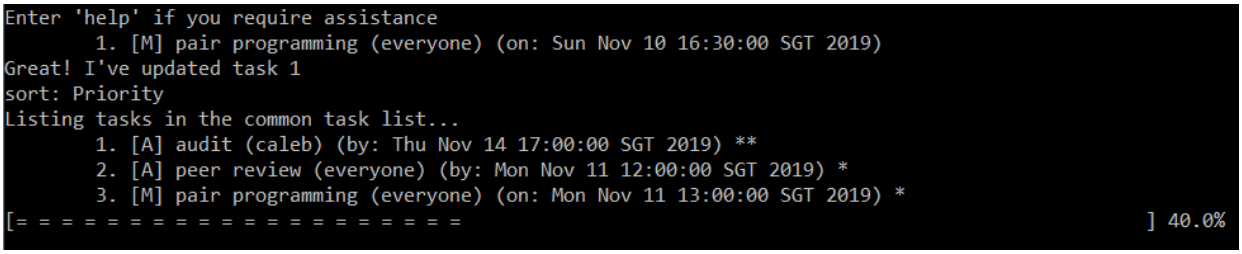

Fig 6.17.8: Rescheduled task has been added back into the task list

6.18 Removing overdue tasks [removeoverdue]

This command allows you to delete an overdue task.

Take note that deleting an overdue task is permanent. Do consider carefully if you wish to delete the overdue task before actually committing to the act.

Format: `removeoverdue` [INDEX_OF_OVERDUE_TASK]
Sample: removeoverdue 1
Example:

1. We use the command removeoverdue on the overdue task list as shown below.

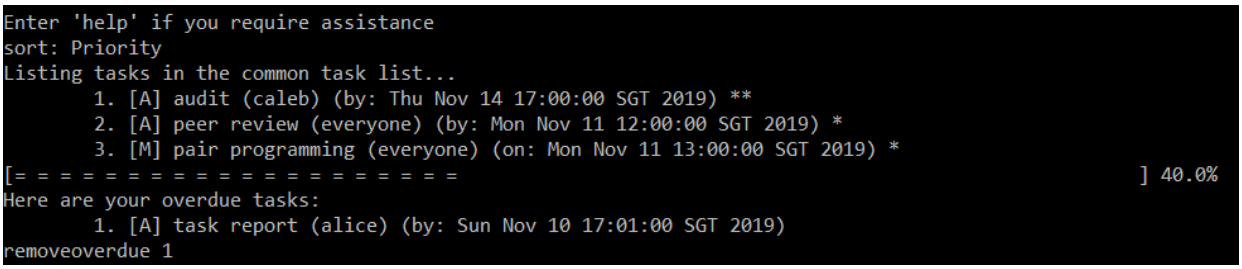

Fig 6.18.1: using the removeoverdue command

2. On hitting enter, you will see that the overdue task has been deleted from the overdue task list

Fig 6.18.2: overdue task has been removed

6.19 Restore [restore]

This command allows you to restore a previously deleted task back into the common task list. Take not that this command only works for the tasks that were deleted from the common task list, and will not work on tasks which were deleted from the overdue list.
This command does not support range based operations.

Format: `restore` [INDEX_OF_TASK]
Sample: restore 1
Example:

1. We have 2 deleted tasks we wish to restore, tasks 3 and 4

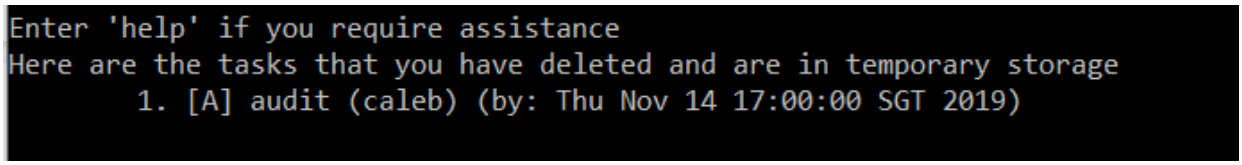

Fig 6.19.1: Sample task to be restored

2. We use the restore command as shown below

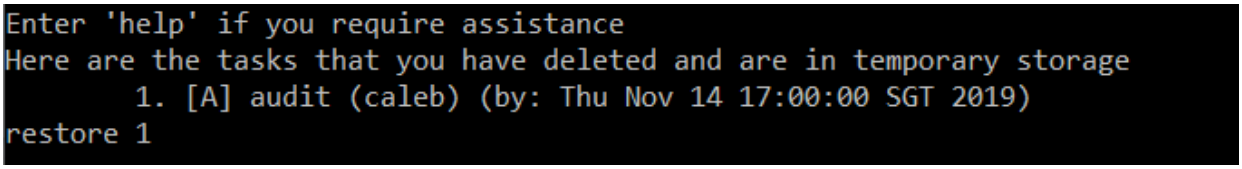

Fig 6.19.2: using the restore command

3. We now see that the task has been restored into the common task list

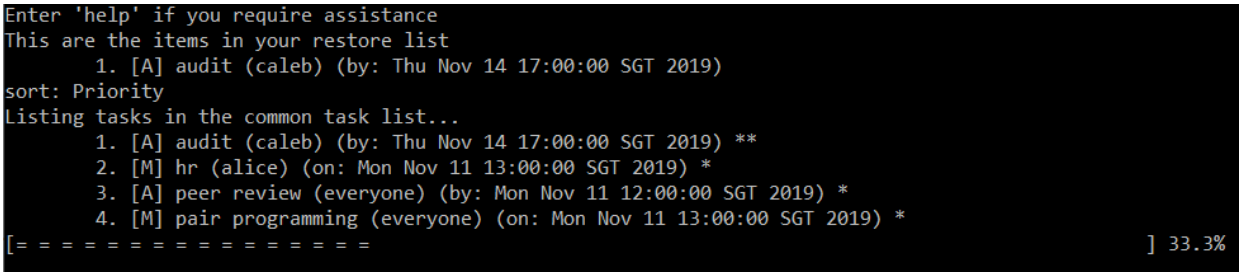

Fig 6.19.3: task has been restored

6.20 Reorder [reorder]

This command allows you to reorder your tasks in the common task list. Type in the command reorder, followed by the indices of the tasks you wish to swap ordering. This command gives you some flexibility in how you might want your tasks list to be shown

Take note that any reordering will not overwrite the sorting method of RoomShare.

Format: `reorder` [INDEX_OF_TASK_1] [INDEX_OF_TASK_2]
Sample: reorder 1 2
Example:

1. We wish to reorder tasks 2 and 3, so we enter the command as shown

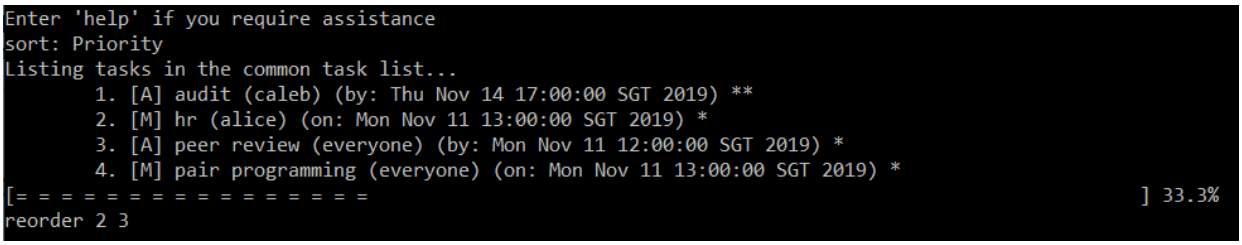

Fig 6.20.1: sample reordering of the tasks

2. The tasks will then be reordered as shown

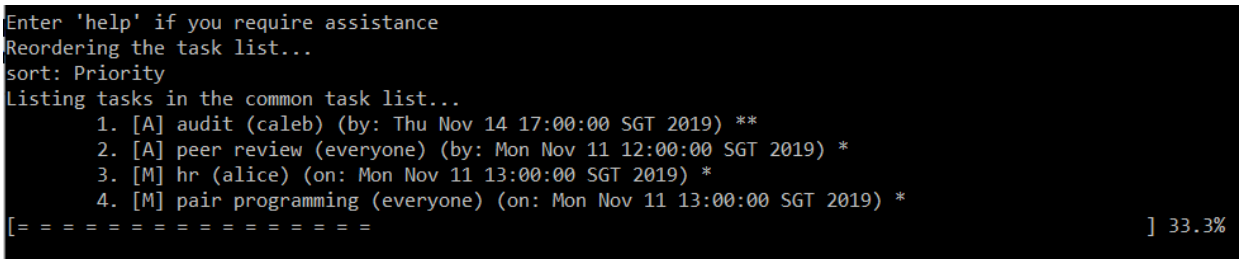

Fig 6.20.2: new ordering of the tasks

6.21 Reopen [reopen]

This command allows you to set a `done` task to an incomplete state. You will need to specify when the deadline of the reopened task is at.

Format: `reopen` [INDEX_OF_TASK] [DATE]

Sample: reopen 1 &12/12/2019 18:00&
Example:

1. Suppose we want to reopen task 6 in the completed list

```
Completed Tasks:
       5. [M] report (harry) (on: Mon Nov 11 15:10:00 SGT 2019) (duration: 2 hours)
       6. [A] report (alice) (by: Mon Nov 11 15:00:00 SGT 2019)
```

Fig 6.21.1: Sample tasks to be reopened

2. We use the reopen command as shown

```
Completed Tasks:
       5. [M] report (harry) (on: Mon Nov 11 15:10:00 SGT 2019) (duration: 2 hours)
       6. [A] report (alice) (by: Mon Nov 11 15:00:00 SGT 2019)
reopen 6 &tmr 17:00&
```

Fig 6.21.1: reopening a completed task

3. We now see that the reopened task has been set to not complete and now appears in the task list

```
Enter 'help' if you require assistance
sort: Priority
Listing tasks in the common task list...
       1. [A] audit (caleb) (by: Thu Nov 14 17:00:00 SGT 2019) **
       2. [A] peer review (everyone) (by: Mon Nov 11 12:00:00 SGT 2019) *
       3. [M] hr (alice) (on: Mon Nov 11 13:00:00 SGT 2019) *
       4. [M] pair programming (everyone) (on: Mon Nov 11 13:00:00 SGT 2019) *
       5. [A] report (alice) (by: Mon Nov 11 17:00:00 SGT 2019) *
[= = = = = = = =                                                          ] 16.7%
These are the tasks that you have already done:
Completed Tasks:
       6. [M] report (harry) (on: Mon Nov 11 15:10:00 SGT 2019) (duration: 2 hours)
```

Fig 6.21.3: reopened task

Take note that `reopen` does not support range based commands as well.

6.22 Bye [bye]

This command allows you to exit RoomShare. On exit, RoomShare will save your data into text files.

Format: `bye`

6.23 Location matching [v2.0]

You would be able to tell the location of other users to add specific tasks for them or to them.

- If you are from work, you might ask another user at the supermarket to help complete certain tasks for example, buying a chocolate bar.

6.24 Real time functionality [v2.0]

Everything added to the app will be real time and available on the internet.

- Database and server functionality for better authentication and handling of tasks
- Instantaneous adding and removal of tasks

7. FAQs

Q: How do I transfer RoomShare to another computer?
Install RoomShare on the new computer, and transfer your text files of data.txt and overdue.txt to the same folder.

Q: Can RoomShare be used in a home setting?
RoomShare is optimised for an office setting, with the tasks being set as Assignment, Meetings and Leave. However, it can by extension be used at home to handle tracking of small tasks using only the Assignment task type

8. Command Summary

- **Help:** `help` [COMMAND]
  E.g
  ```
  help add
  help sort
  help delete
  ```

- **Refresh List:** `list`

- **Add:** `add` *#TASK_TYPE# &DATE_AND_TIME& (DESCRIPTION) *PRIORITY* ^DURATION^ @ASSIGNEE@ %RECURRENCE%*
  E.g.
  ```
  add #meeting# &30/10/2019 11:00& (OP2 meeting) ^2 hours^
  add #assignment# &tomorrow 15:00& (finish report) *high*
  add #leave# (vacation) &today 16:00&tomorrow 18:00& @Cindy@
  ```

- **Update:** `update` [INDEX] *&DATE_AND_TIME& (DESCRIPTION) *PRIORITY* ^DURATION^ @ASSIGNEE@ %RECURRENCE%*
  E.g.
  ```
  update 1 &next sunday 11:00& (video conference)
  update 1 ^2 hours^
  update 1 @Harry and Caleb@ *high*
  ```

- **Done :** `done` [INDEX]
  E.g.
  ```
  done 1
  ```

- **Delete** : `delete` [INDEX] or `delete` [INDEX] - [INDEX]
  E.g.
  ```
  delete 1
  delete 1-3
  ```

- **Find :** `find` [KEYWORD] [MORE_KEYWORD]
  E.g.
  ```
  find report
  ```

- **Snooze :** `snooze` [INDEX][AMOUNT][UNIT_OF_TIME]
  E.g. `snooze 1 2 hours`

- **Sort:** `sort` [ORDER_TO_SORT]
  E.g. `sort alphabetical`
  `sort deadline`
  `sort priority`
  `Sort type`

- **Add subtasks**: `subtask` [INDEX] [LIST_OF_SUBTASK]
  E.g. `subtask 1 buy paper, buy ink, buy pens`

- **Restore task**: `restore` [DELETED_LIST_INDEX]
  E.g. `restore 1`

- **Change Priority**: `priority` [INDEX] [PRIORITY]
  E.g. `priority 1 high`

- **Reorder tasks**: `reorder` [INDEX_1] [INDEX_2]
  E.g. `reorder 1 2`

- **Show completed tasks:** `completed`

- **Show overdue tasks:** `overdue`

- **Reschedule overdue task:** `reschedule` [INDEX_OF_OVERDUE_TASK] [DATE]
  E.g `reschedule 1 &tmr 17:00&`

- **Show by assignee:** `show` [NAME _OF_USER]
  E.g `show Alice`

- **Remove overdue task:** `removeoverdue` [INDEX_OF_OVERDUE_TASK]
  E.g `removeoverdue 1`

- **Log:** `log`

- **Bye :** `bye`