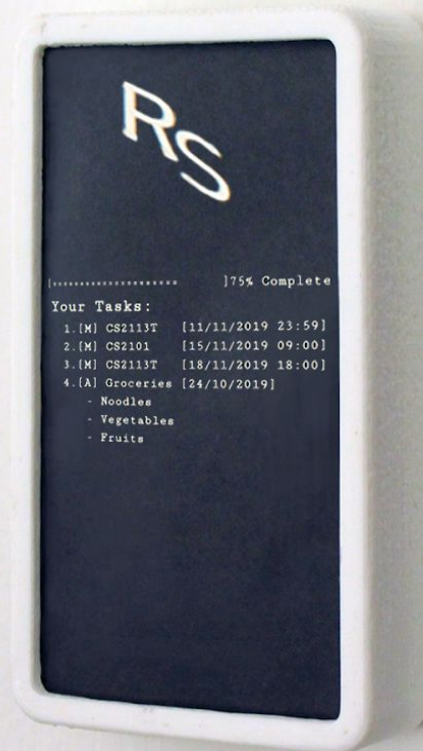# User Guide
Team F14-3: RoomShare

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 section 2 for more details on how to use the application.

This is a mockup of our UI:

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

Note the following symbols and formatting used in this document:

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.
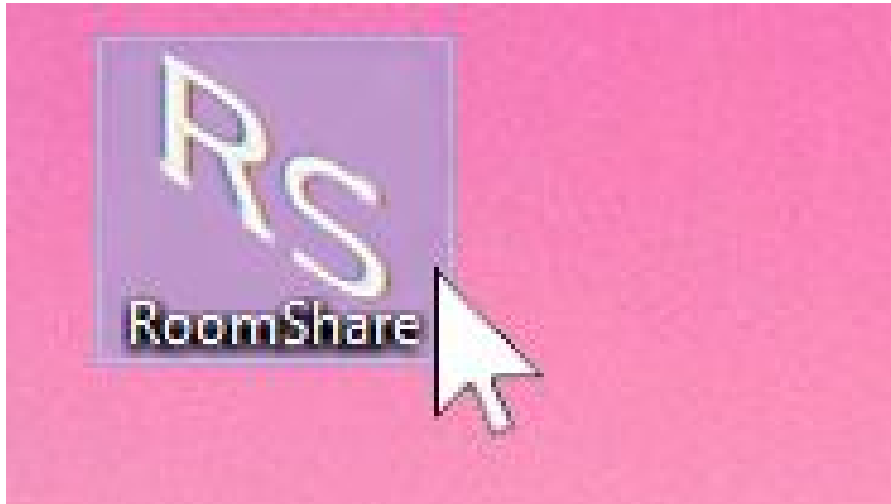
3. Setting up RoomShare

Fig 2 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. Some commands that you can type are:
   - Help : a guide to show what kinds of commands are acceptable
   - List : list all the tasks that are currently in the task list
   - Add : adds the specific type of task into the task list
   - Delete : deletes a task in the task list based on its index
     Eg. delete 3 : deletes the 3rd task in the task list
   - Bye : exits the app

5. Refer to Section 3 (Features for more details of each command)
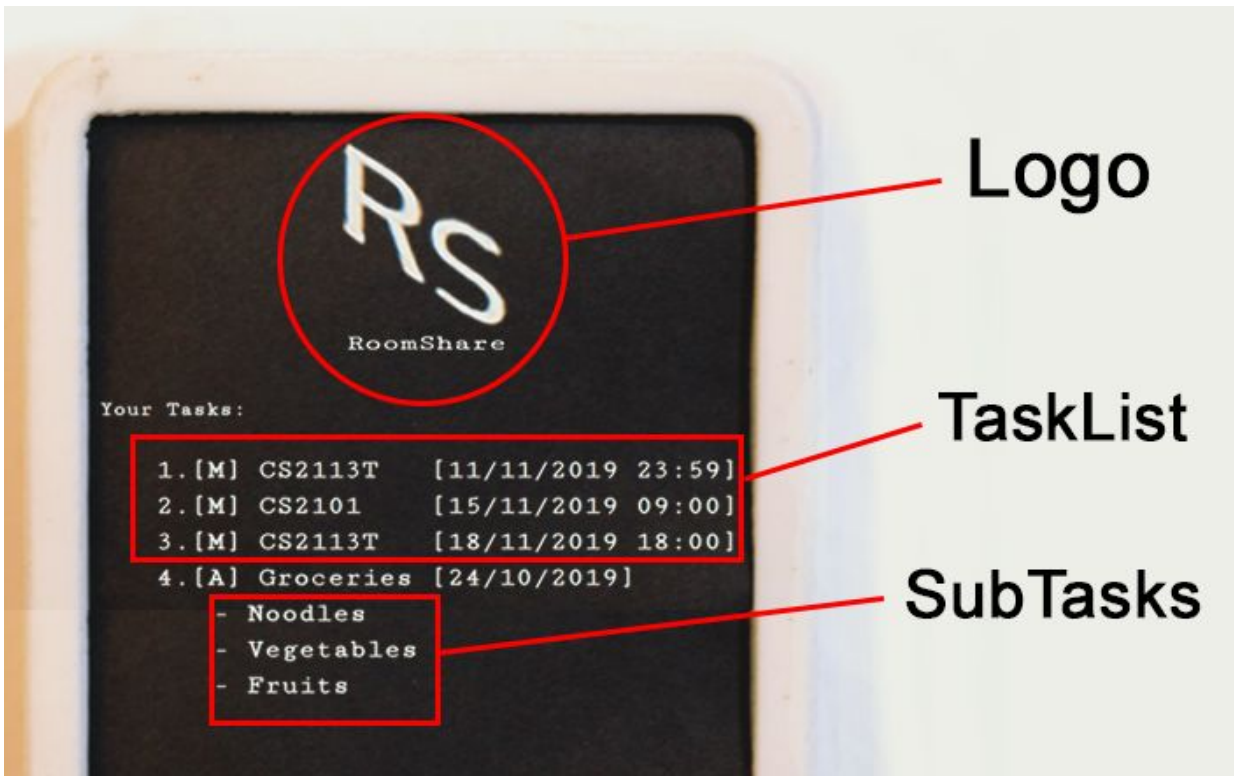
4. UI

4.1 Tour of the UI

Fig 3 User Interface Layout

Above is a picture of the user interface, It includes the Logo at the top with a list of tasks below. Subtasks of a particular task are displayed below the main task.

5.   Features

5.1 Help [help]

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

Format: `help` [COMMAND]

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

```
help sort
```

After typing the command and press **Enter,** 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
```

5.2 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

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

Example 1: Type this command onto your terminal:

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

  - ➔ 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.

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

```
1. [M][x] project brainstorming (everyone) (on: Fri Nov 01 14:00:00 SGT 2019) (duration: 2 hours)
```

Example 2: Type this command onto your terminal:

```
add #assignment# (submit report) &this fri 16:00& @Harry@
```

  - ➔ 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

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

```
2. [A][x] submit report (harry) (by: Fri Nov 01 16:00:00 SGT 2019)
```

5.3 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 **5.1 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%*

Example: If you want to change the description, date, or assignee of a task, for example this task:

```
2. [A][x] submit report (harry) (by: Fri Nov 01 16:00:00 SGT 2019)
```

Type this command onto your terminal:

```
update 2 (update report) @Alice@
```

  - ➔ The number 2 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 Alice.

After typing the command and press **Enter**, the updated task will look like this:

```
2. [A][x] update report (alice) (by: Fri Nov 01 16:00:00 SGT 2019)
```

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

5.4 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]

Example: if you want to snooze a task by 2 hours, for example this task:

```
2. [A][x] update report (alice) (by: Fri Nov 01 16:00:00 SGT 2019)
```

Type this command onto your terminal:

```
snooze 2 2 hours
```

➔ The first number 2 is the index of the task inside your task list that you want to snooze.
➔ The second 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.

After typing the command and press **Enter**, the snoozed task should look like this:

```
2. [A][x] update report (alice) (by: Fri Nov 01 18:00:00 SGT 2019)
```

Notice the deadline has been changed from (by: Fri Nov 01 16:00:00 SGT 2019) to (by: Fri Nov 01 18:00:00 SGT 2019).

5.5 Done [done]

Marks the selected task as completed inside your task list.

Format: `done` [INDEX]

Example: if you want to mark a task as done, for example this task:

```
2. [A][x] update report (alice) (by: Fri Nov 01 18:00:00 SGT 2019)
```

Type this command onto your terminal:

```
done 2
```

➔ The number 2 is the index of the task inside your task list that you want to mark as done.

After typing the command and press **Enter**, the task will be marked as done with a tick like this:

```
2. [A][v] update report (alice) (by: Fri Nov 01 18:00:00 SGT 2019)
```

5.6 Delete [delete]

Removes the task at the specific index you enter.

Format: `delete` [INDEX]
Or `delete` [INDEX - INDEX]

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

```
Listing tasks in your task list...
    1. [A][x] develop graphic feature (caleb) (by: Sun Nov 10 14:15:00 SGT 2019)
    2. [M][x] op2 (everyone) (on: Fri Nov 01 12:00:00 SGT 2019)
    3. [A][v] update report (alice) (by: Fri Nov 01 18:00:00 SGT 2019)
```

Type this command onto your terminal:

```
delete 1
```

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

```
Listing tasks in your task list...
    1. [M][x] op2 (everyone) (on: Fri Nov 01 12:00:00 SGT 2019)
    2. [A][v] update report (alice) (by: Fri Nov 01 18:00:00 SGT 2019)
```

5.7 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]

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

```
find report
```

The matching tasks should appear like this:

```
Searching for item in task list...
       2. [A][x] report updated feature (caleb) (by: Mon Nov 04 15:00:00 SGT 2019)
       3. [A][v] update report (alice) (by: Fri Nov 01 18:00:00 SGT 2019)
```

5.8 Sort [sort]

Sort the data based on the style of order that you enter: alphabetical, priority, deadline.

Format: `sort` [ORDER_TO_SORT]

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

```
Listing tasks in your task list...
       1. [M][x] op2 (everyone) (on: Fri Nov 01 12:00:00 SGT 2019)
       2. [M][x] attach documents to report (hannah) (on: Sat Nov 02 20:00:00 SGT 2019)
       3. [A][x] buy equipment (ryan) (by: Mon Nov 04 14:00:00 SGT 2019)
```

Type this command onto your terminal:

```
sort alphabetical
```

After typing the command and press **Enter**, the resulting task list should be sorted alphabetically like this:

```
Listing tasks in your task list...
       1. [M][x] attach documents to report (hannah) (on: Sat Nov 02 20:00:00 SGT 2019)
       2. [A][x] buy equipment (ryan) (by: Mon Nov 04 14:00:00 SGT 2019)
       3. [M][x] op2 (everyone) (on: Fri Nov 01 12:00:00 SGT 2019)
```

5.9 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.

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.

Example: If you have a big task and you want to break it down into smaller subtasks, for example this task:

```
2. [A][x] buy equipment (ryan) (by: Mon Nov 04 14:00:00 SGT 2019)
```

Type this command onto you terminal:

```
subtask 2 buy battery, buy 3D printer, buy laser cutter
```

- The number 2 is the index of the task 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.

After typing the command and press **Enter**, you should see the subtasks listed below the main task like this:

```
2. [A][x] buy equipment (ryan) (by: Mon Nov 04 14:00:00 SGT 2019)
        -buy battery
        - buy 3d printer
        - buy laser cutter
```

5.10 Log [log]

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

Format: `log`

Example: type `log` onto your terminal and press **Enter**, the terminal should show you a message like this:

```
Log has been successfully written to logs\log_Fri_Nov_01_01_21_01_SGT_2019.txt
```

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:
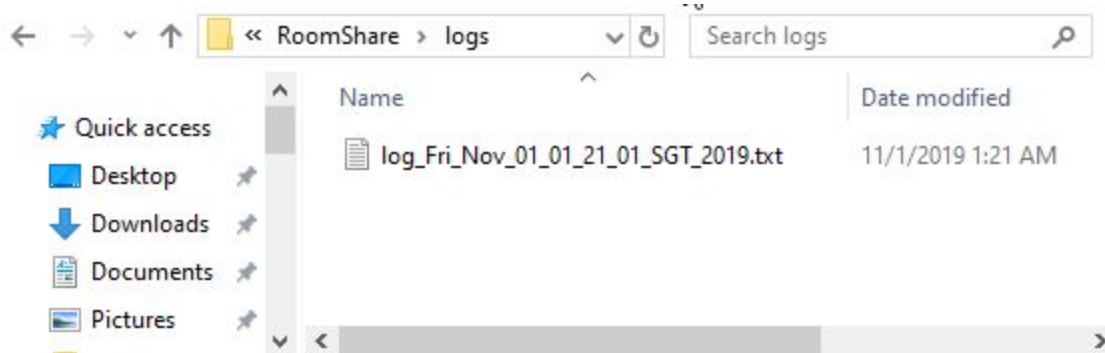
5.11 Save Data

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.

5.12 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.

5.13 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

6.   List of User Commands

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

- **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) &tdy 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

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

- **Log:** `log`

- **Bye :** `bye`