

# +Work - User Guide

|   |    |
|---|----|
| 1. Introduction .....   | 2  |
| 2. Quick Start .....  | 2  |
| 3. Features .....   | 3  |
| 3.1. Universal Commands .....                                 | 5  |
| 3.2. Member Commands .....                                    | 8  |
| 3.3. Task Commands .....                                      | 17 |
| 3.4. Calendar Commands .....                                  | 24 |
| 3.5. Inventory Commands .....                                 | 29 |
| 3.6. Statistics Commands .....                                | 34 |
| 3.7. Multiline Commands .....                                 | 36 |
| 3.8. Command Line Autocomplete .....                          | 39 |
| 3.9. Settings Commands .....                                  | 40 |
| 3.10. Saving Data .....                                       | 43 |
| 4. FAQ .....  | 44 |
| 5. Command Summary .....                                      | 44 |
| 6. Upcoming Features .....                                    | 45 |
| 6.1. Cloud Integration [coming in v2.0] .....                 | 46 |
| 6.2. Customisable User Profiles [coming in v2.0] .....        | 46 |
| 6.3. AutoComplete Support for prefixes [coming in v2.0] ..... | 46 |
| 6.4. Instant Messaging [coming in v2.0] .....                 | 46 |

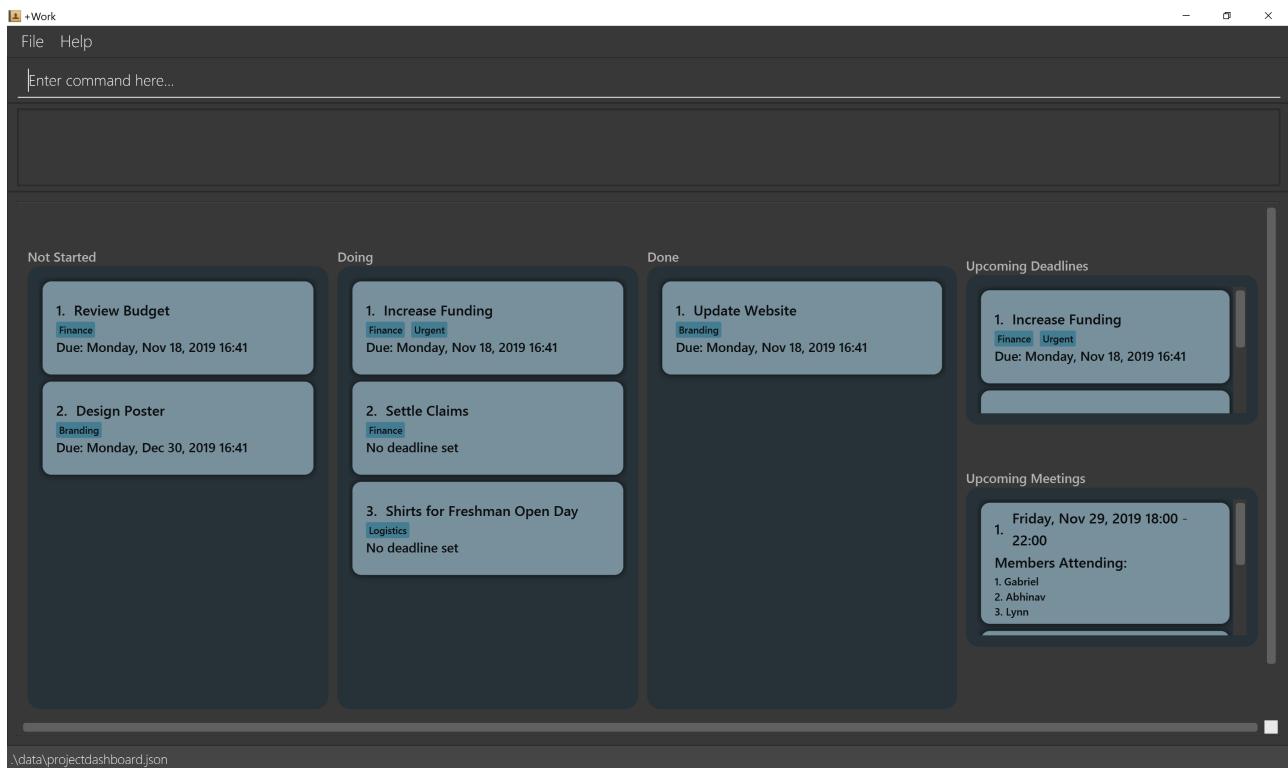
By: CS2103T-T12-1 Since: Sept 2019 Licence: MIT

# 1. Introduction

+Work is a project management tool that allows students to successfully achieve your objectives for their project. It is aimed at NUS project leaders who are in charge of a small project group of less than 20 people. +Work is a one-stop application to help you manage your members, meeting times, task allocations as well as equipment purchased. Take full advantage of our Command Line Interface (CLI) while still having the all the benefits of a Graphical User Interface (GUI). Happy working!

## 2. Quick Start

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



5. Type the command in the command box and press **Enter** to execute it.  
e.g. typing **help** and pressing **Enter** will open the help window.
6. Here are some commands you can try to get you started:
  - **add-member mn/Adam Smith mi/AS** : Adds a member with name "Adam Smith" to the project.
  - **add-task tn/Finish up milestone setting** : Adds a task with name "Finish up milestone" to the project.
  - **assign ti/1 mi/JD** : Assigns the project member "John Doe" (with member id "JD"), to the task "Finish up milestone setting".
  - **doing-task ti/1** : Changes status of task "Finish up milestone setting" to **doing**:

### **3. Features**

## Command Format

- The command begins with the type of command to be executed
  - Example: in `remove-member mi/MEMBER_ID`, remove member is the type of command
- The words in UPPER\_CASE corresponds to the parameters given by the user
  - Example: In `add-member mn/ MEMBER_NAME`, MEMBER\_NAME is the input specified by the user, such as: `add-member mn/Gabriel Seow`
- You can specify the type of parameter given through a prefix. The list of allowed prefixes can be found in the table below:
  - Example: In `add-task tn/TASK_NAME s/TASK_STATUS`, the user inputs the name of the task after `tn/` and inputs the status of the task after `s/`, such as: `add task t/Finish Project Draft s/undone`

*Table 1. List of Prefixes*

| Command Type       | Prefix   | Description<br>(Parameters can be given in any order) |
|--------------------|----------|---|
| <b>Calendars</b>   | fp/      | File path   |
| <b>Inventories</b> | i/       | Inventory name  |
|                    | p/       | Price   |
|                    | ii/      | Inventory tag   |
|                    | ty/      | PDF type  |
| <b>Meetings</b>    | meeting/ | Meeting index   |
|                    | start/   | Start period  |
|                    | end/     | End period  |
|                    | hours/   | Duration of meeting                                   |
| <b>Members</b>     | mn/      | Member name   |
|                    | mi/      | Member ID (set by you)                                |
|                    | mt/      | Member tag  |
|                    | im/      | Image path  |
| <b>Tasks</b>       | tn/      | Task name   |
|                    | ti/      | Task index  |
|                    | at/      | Task deadline   |
|                    | s/       | Task status   |
|                    | tt/      | Task tag  |

- **Tags**

- Tags are the only non-compulsory parameter mentioned in commands (unless

otherwise stated).

- When tags are mentioned as a parameter, it is taken for granted that you can input multiple tags using the format `mt/… mt/… mt/…`

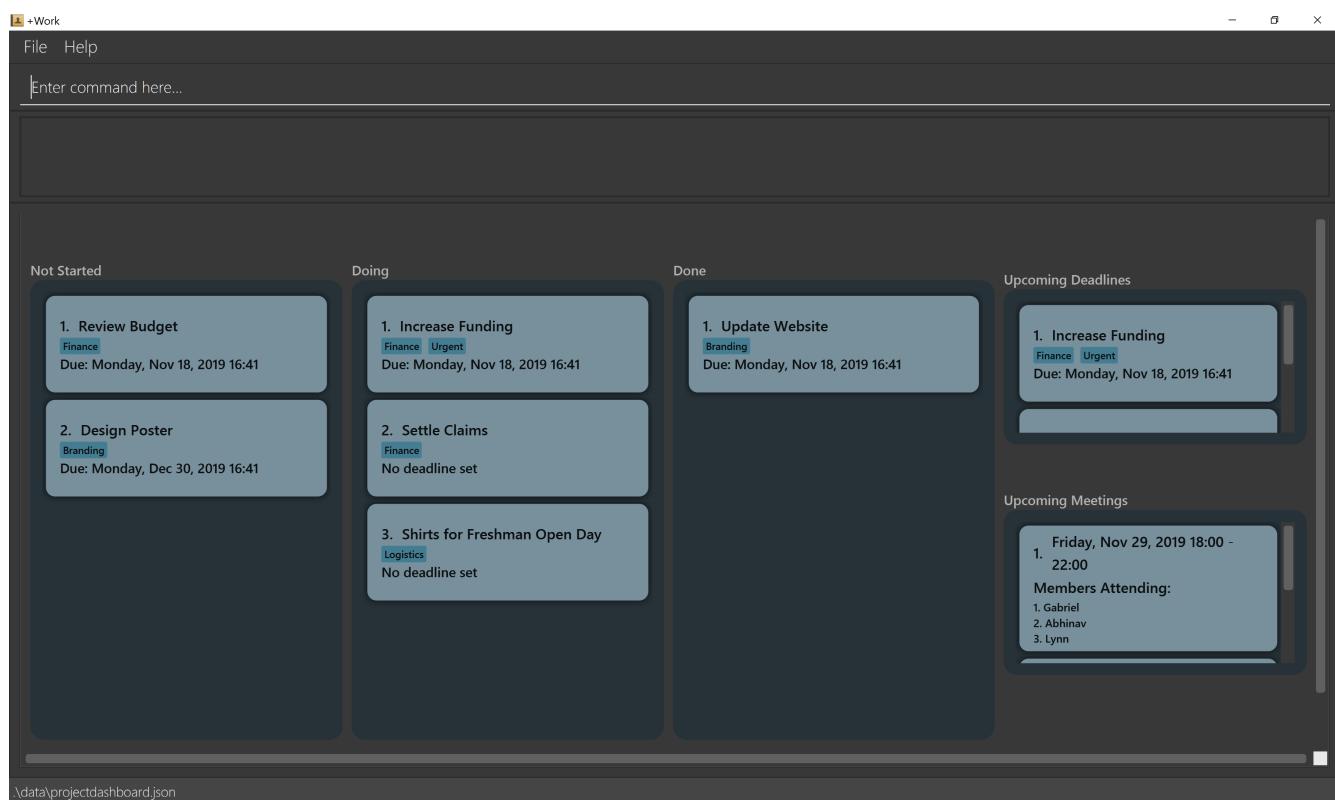
## 3.1. Universal Commands

### 3.1.1. Accessing the project dashboard page: `home`

This command brings you to the project dashboard page, where tasks are displayed.

Format: `home`

Calling the `home` command will bring you to the following page:



### 3.1.2. Accessing the time management page: `calendar`

This command brings you to the time management page where calendar and meeting times are displayed

Format: `calendar`

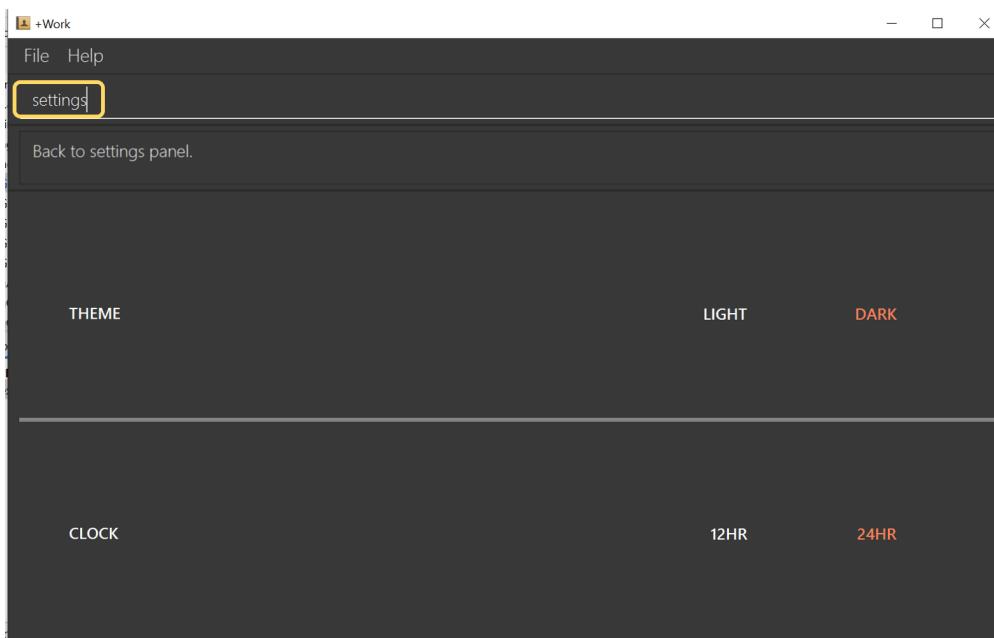
Calling the `calendar` command will bring you to the following page:

### 3.1.3. Accessing the settings page: `settings`

This command brings you to the settings page

Format: **settings**

Entering the **settings** command will bring you to the following page:



**:INFORMATION\_SOURCE:**

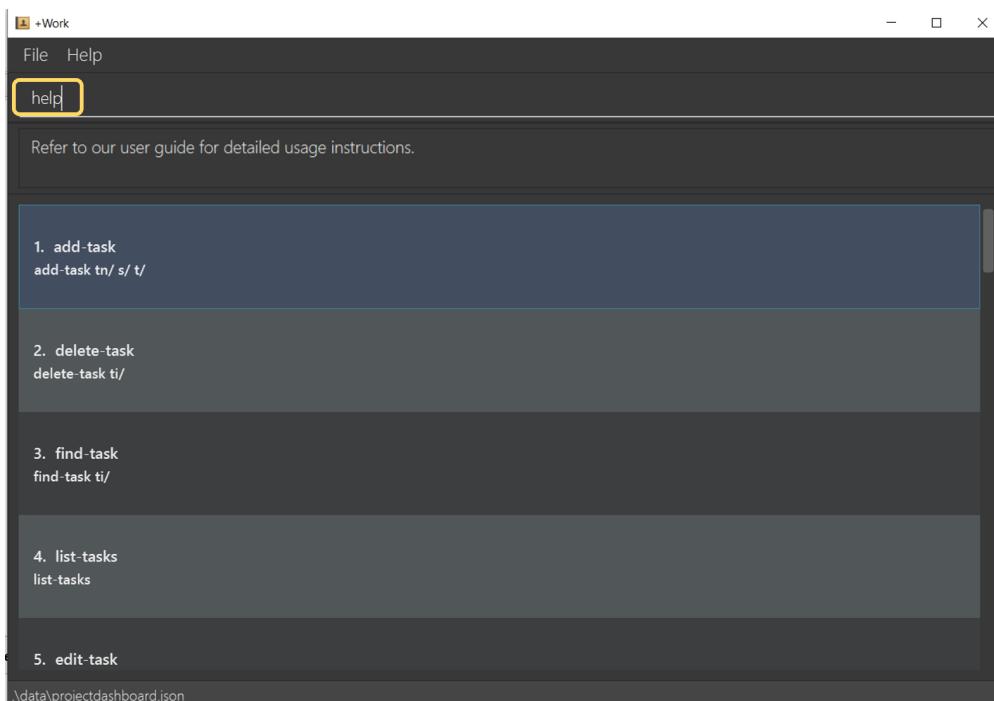
You can refer to section 3.7 on specific settings-related commands to apply.

### 3.1.4. Viewing help: **help**

Displays a list of possible commands for the user

You can toggle through the command list (either through up down keys or mouse) and it will paste the correct syntax into the command line.

Format: **help**



### 3.1.5. Undo a command: **undo**

You can undo your recent commands by using the **undo** command

Format: **undo**

Example:

- Suppose you accidentally deleted task #6 using the **delete-task** command

The screenshot shows a dark-themed application window with a menu bar at the top. Below the menu is a search bar labeled 'Enter command here...'. A large text area below is titled 'Listed all tasks'. The tasks are listed as follows:

- 1. Review Budget  
Finance  
Due on: Monday, Nov 18, 2019 01:47 AM  
List of Members Allocated  
1. AB
- 2. Increase Funding  
Finance Urgent  
Due on: Monday, Nov 18, 2019 01:47 AM  
List of Members Allocated  
1. AB
- 3. Settle Claims  
Finance  
No deadline set  
List of Members Allocated  
1. AB
- 4. Update Website  
Branding  
Due on: Monday, Nov 18, 2019 01:47 AM  
List of Members Allocated  
1. EK
- 5. Shirts for Freshman Open Day  
Logistics  
No deadline set  
List of Members Allocated  
1. SL
- 6. Design Poster  
Branding  
Due on: Monday, Dec 30, 2019 01:47 AM  
List of Members Allocated  
1. JD

- Entering the **undo** command will bring back the deleted task

The screenshot shows the same application window after the **undo** command was executed. The task list now includes task 6 again:

- 1. Review Budget  
Finance  
Due on: Monday, Nov 18, 2019 01:47 AM  
List of Members Allocated  
1. AB
- 2. Increase Funding  
Finance Urgent  
Due on: Monday, Nov 18, 2019 01:47 AM  
List of Members Allocated  
1. AB
- 3. Settle Claims  
Finance  
No deadline set  
List of Members Allocated  
1. AB
- 4. Update Website  
Branding  
Due on: Monday, Nov 18, 2019 01:47 AM  
List of Members Allocated  
1. EK
- 5. Shirts for Freshman Open Day  
Logistics  
No deadline set  
List of Members Allocated  
1. SL
- 6. Design Poster  
Branding  
Due on: Monday, Dec 30, 2019 01:47 AM  
List of Members Allocated  
1. JD

## WARNING

Once you restart +Work, you won't be able to **undo** commands from the previous session!

### 3.1.6. Redo a command: **redo**

You can redo a previously **undone** command by typing **redo**.

Format: **redo**

Example:

- Let's say you deleted a task and you **undo** the command. You can simply use the **redo** command to delete the task again

## 3.2. Member Commands

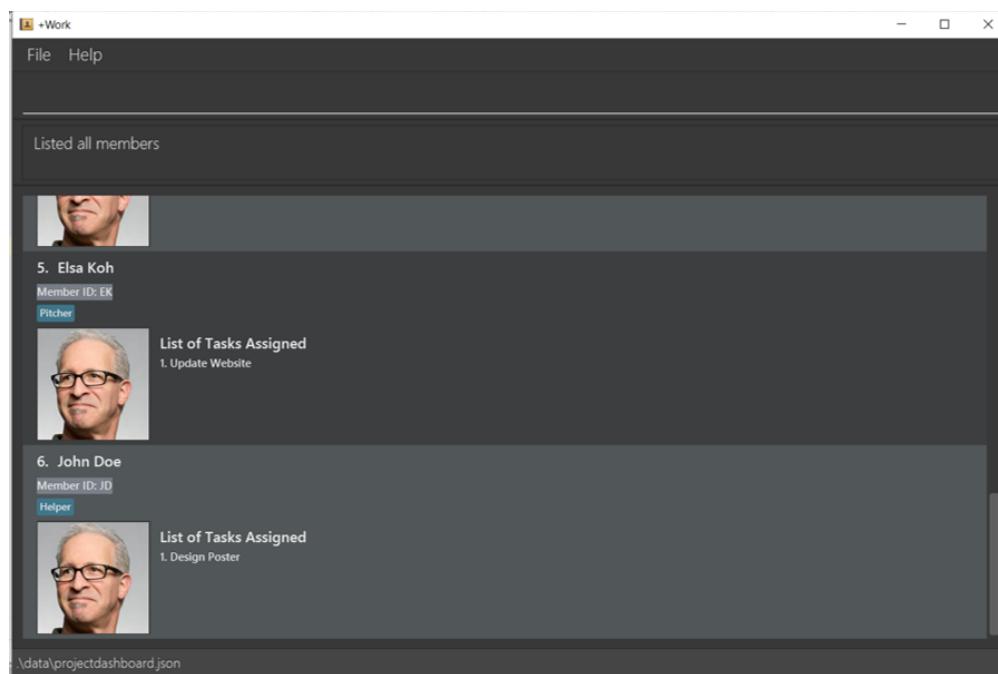
### 3.2.1. Adding a member: **add-member**

To add a member to the list of team members in +Work, use the command **add-member** following the format below.

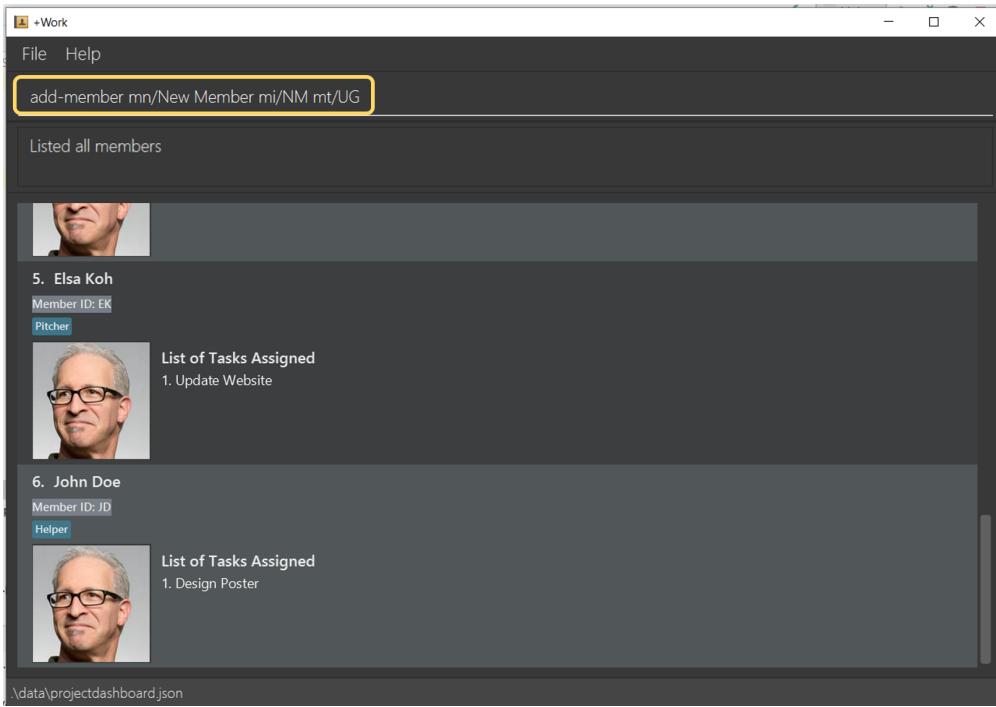
Format: **add-member mn/MEMBER\_NAME mi/MEMBER\_ID mt/TAGS**

Example: **add-member mn/New Member mi/NM mt/UG** can be executed as follows:

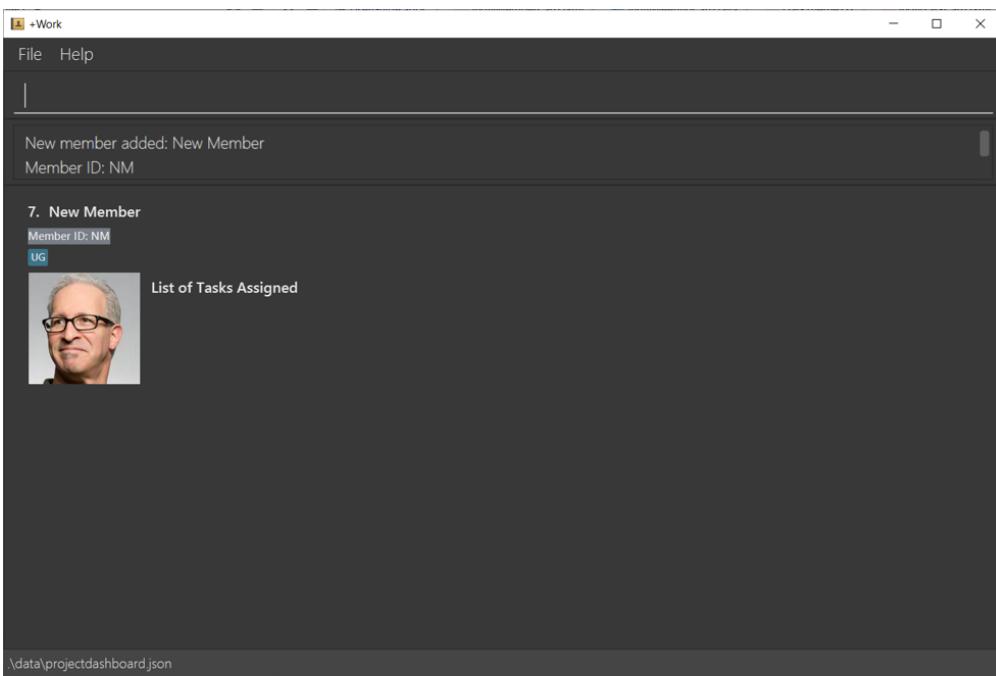
**Step 1:** +Work initially contains a list of 6 project members, as shown below.



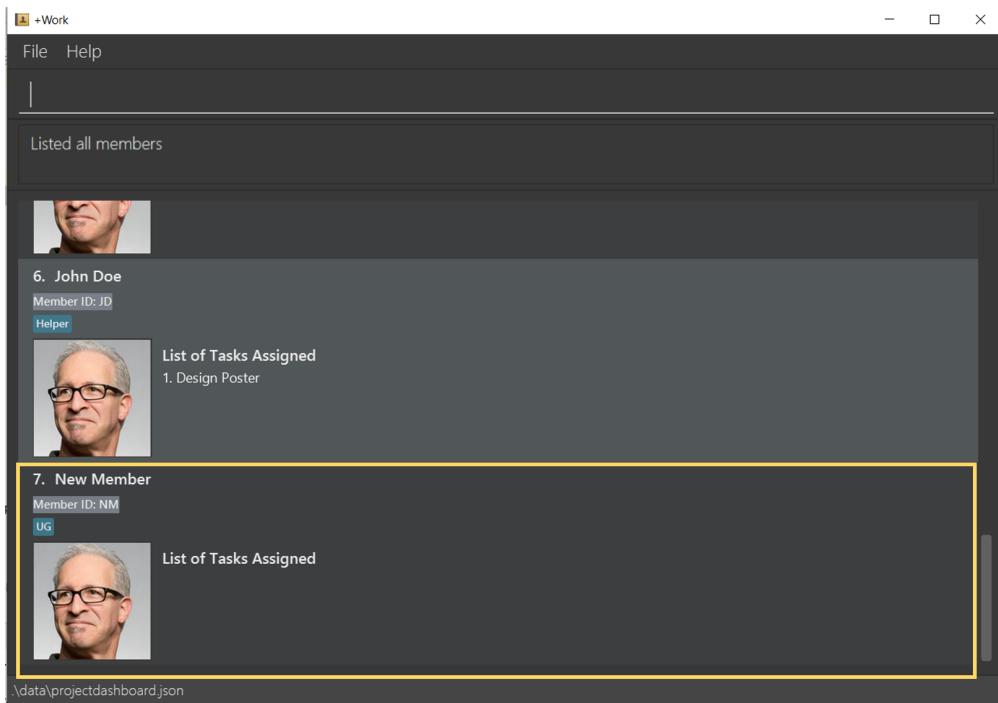
**Step 2:** To add a new project member into +Work, you enter the command **add-member mn/New Member mi/NM mt/UG** into the command prompt box.



**Step 3:** After you hit **Enter**, the result box will display the message "New member added", and a new member with name 'New Member', member ID 'NM' and tag 'UG' is added to +Work.



The addition of a new member can also be seen from the list of members as shown below:



## IMPORTANT

Member ID is an alphanumeric ID set by you, and cannot be changed once the member is created.

### :INFORMATION\_SOURCE:

- Adding a member tag is optional in the adding of a new member.
- It is possible to add a member with multiple tags following this format:  
`add-member mn/New Member mi/NM mt/UG mt/DG mt/...`

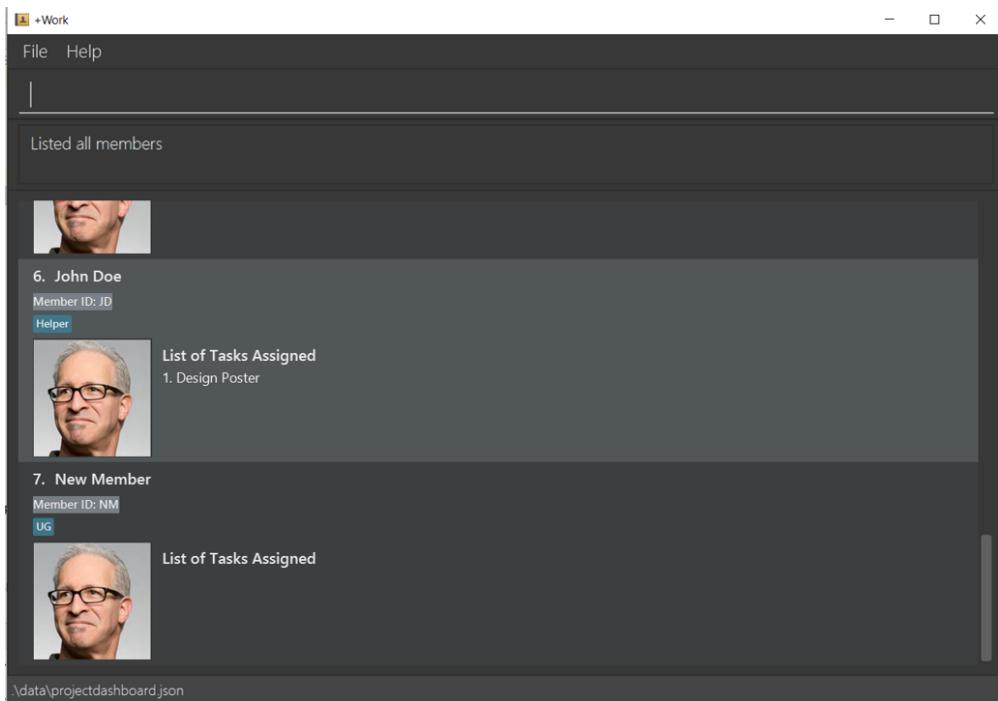
### 3.2.2. Set image for member: `set-image`

To set a profile picture for a member in +Work, use the command `set-image` following the format below.

Format: `set-image mi/MEMBER_ID im/IMAGE_PATH`

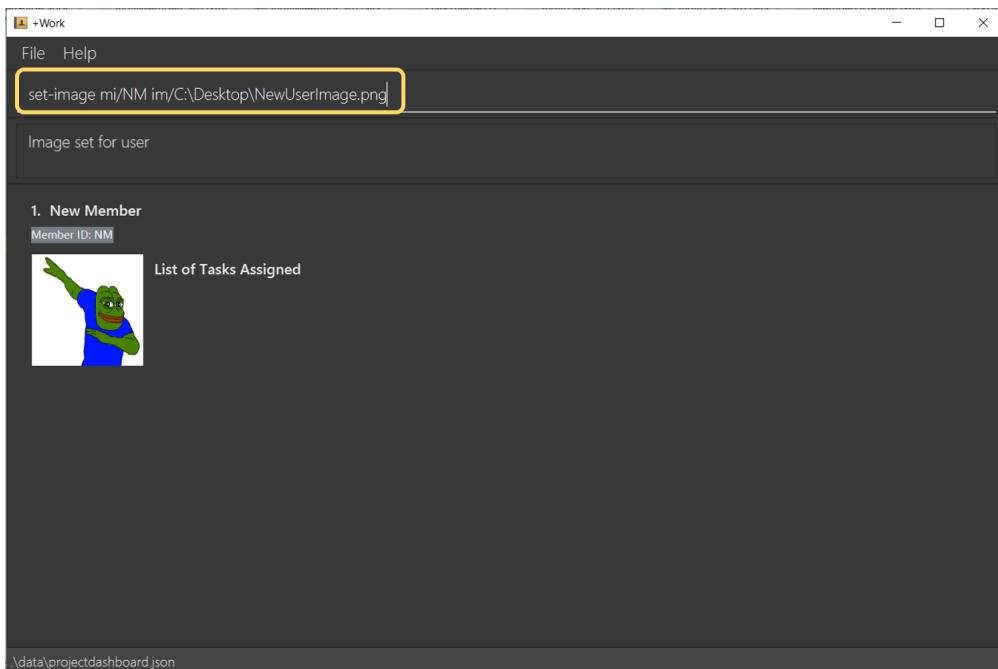
Example: `set-image mi/NM im/C:\Desktop\NewUserImage.png` can be executed as follows:

**Step 1:** +Work initially contains a list of project members with default profile pictures, as shown below.



**Step 2:** To update the profile picture of the project member with member ID 'NM' in +Work to a specified image, you enter the command `set-image mi/NM im/C:\Desktop\NewUserImage.png` into the command prompt box.

**Step 3:** After you **Enter** the command, the member 'New Member' with member ID 'NM' has a new profile picture, specified by the image path you entered.



#### :INFORMATION\_SOURCE:

Image Path refers to the folder path of the image stored in your computer, and should end with .png

## IMPORTANT

If you shift the image's location in your computer, +Work will be unable to find the image to display. A warning message will be displayed, before displaying the default profile picture. It is recommended that you store all the images in a central folder to prevent this from happening.

### 3.2.3. Editing a member : edit-member

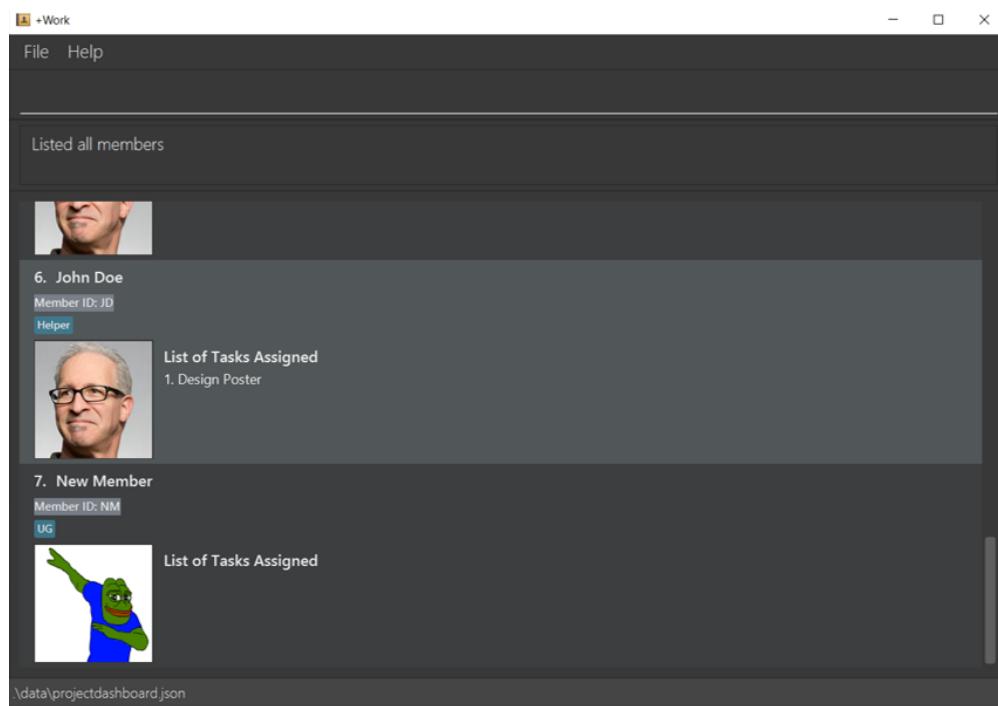
To edit a member in +Work, using the command `edit-member` following the format below.

Format: `edit-member mi/MEMBER_ID mn/ MEMBER_NAME mt/MEMBER_TAG`

- Edits the member at the specified `mi/MEMBER_ID`.
- In this command, all the fields apart from `mi/MEMBER_ID` are optional. However, at least one of the optional fields must be provided.
- Existing values will be updated to the input values.
- When editing tags, the existing tags of the member will be removed i.e adding of tags is not cumulative.

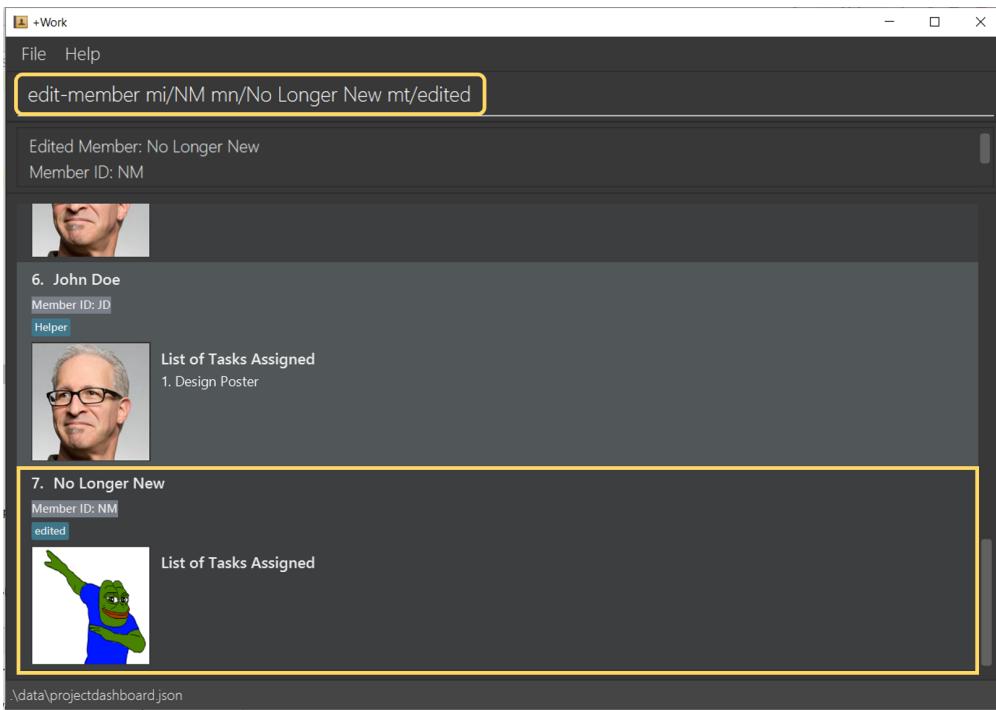
Example: `edit-member mi/NM mn/No Longer New mt/edited` can be executed as follows:

**Step 1:** +Work initially contains a list of 7 project members, as shown below.



**Step 2:** To edit the member name and tag of 'New Member' with member ID 'NM', you enter the command `edit-member mi/NM mn/No Longer New mt/edited` into the command prompt box.

**Step 3:** After you `Enter` the command, the member with member ID 'NM' and tag 'UG' is edited, with a new member name 'No Longer New' and new tag 'edited'.

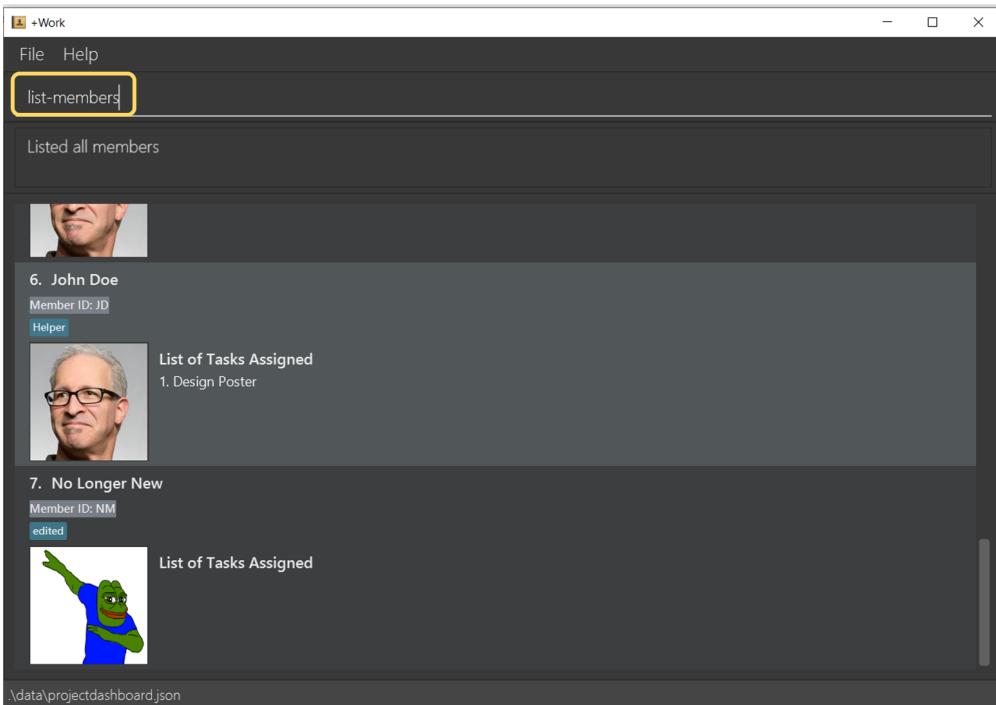


### 3.2.4. List existing members: `list-members`

To get a list of all members added to +Work, used the command `list-members` following the format below.

Format: `list-members`

Example: Entering `list-members` into the command prompt will result in the following:



As seen from the above, all existing project members in +Work will be listed.

### 3.2.5. Removing a member: `remove-member`

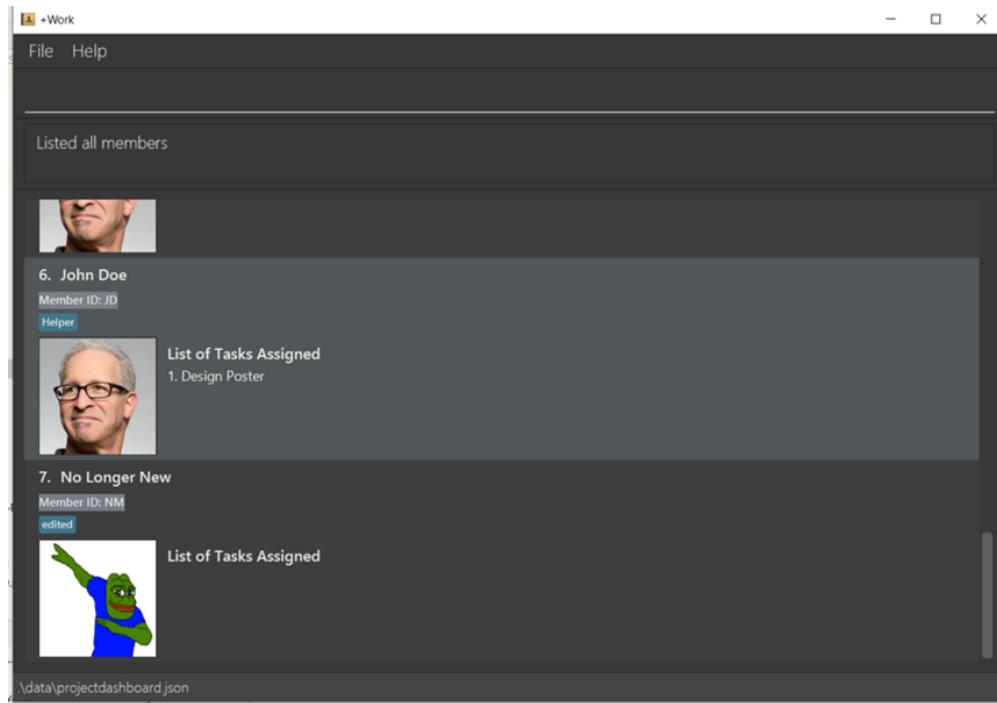
To remove a member from the project, and subsequently remove him from associated tasks, use

the `remove-member` command in the format below.

Format: `remove-member [mi/MEMBER_ID]`

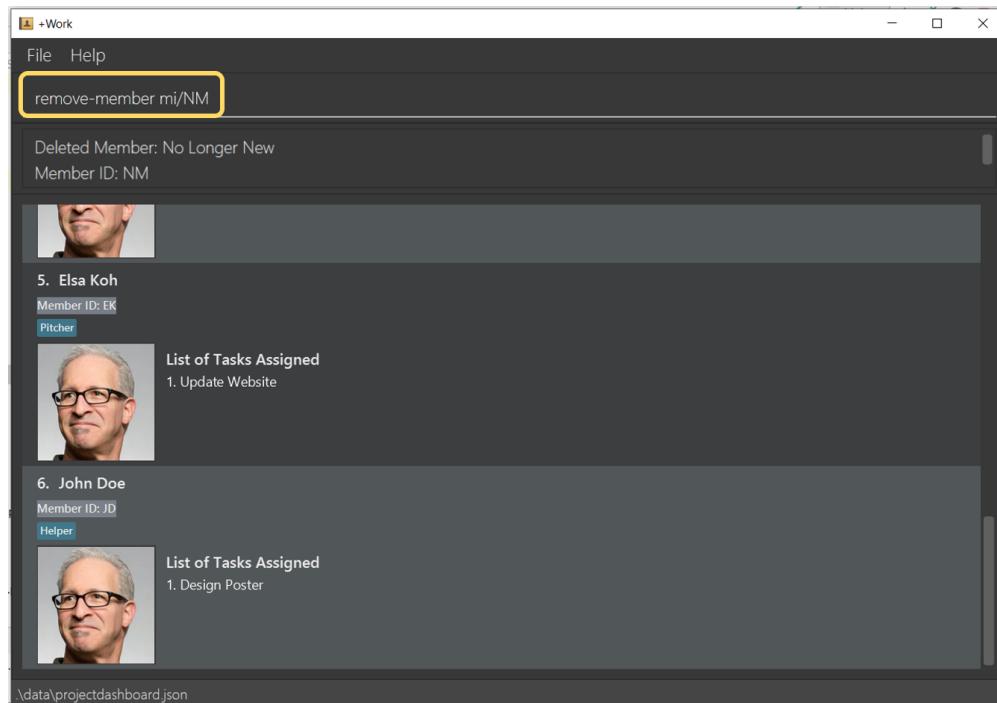
Example: `remove-member mi/GS` can be executed as follows:

**Step 1:** +Work now contains a list of 7 project members, as shown below.



**Step 2:** To remove project member 'No Longer New', with member ID 'NM' from +Work, you enter the command `remove-member mi/NM` into the command prompt box.

**Step 3:** After you `Enter` the command, the member 'No Longer New' is no longer a project member in +Work, as seen from the list of members below.



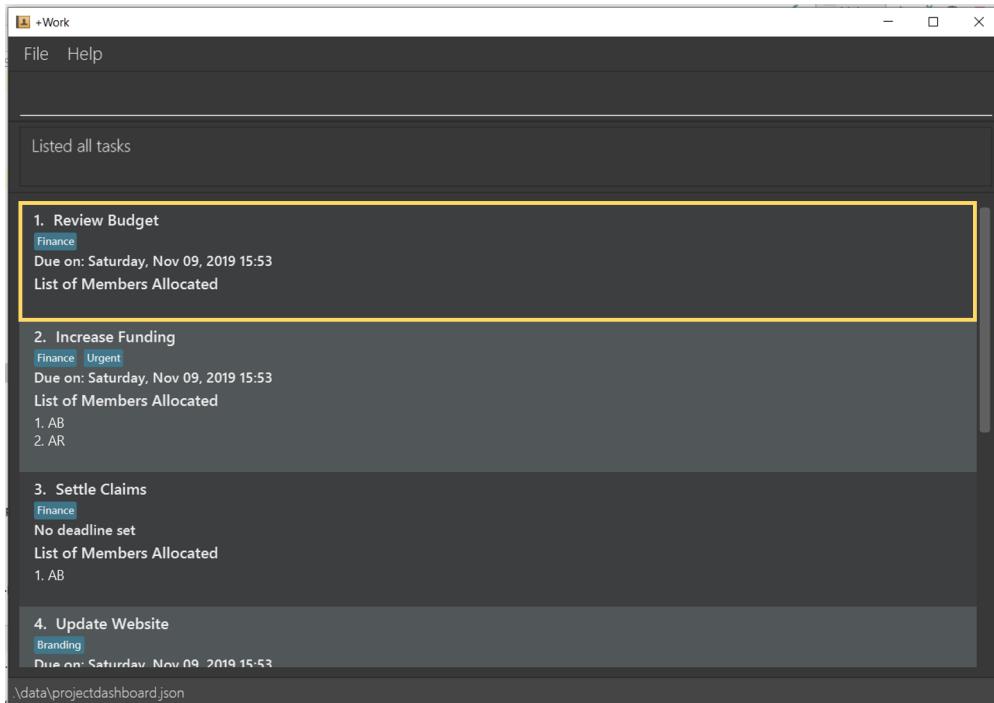
### 3.2.6. Assign a task to a member: `assign`

To assign a task to a specific team member, use the `assign` command in the format below.

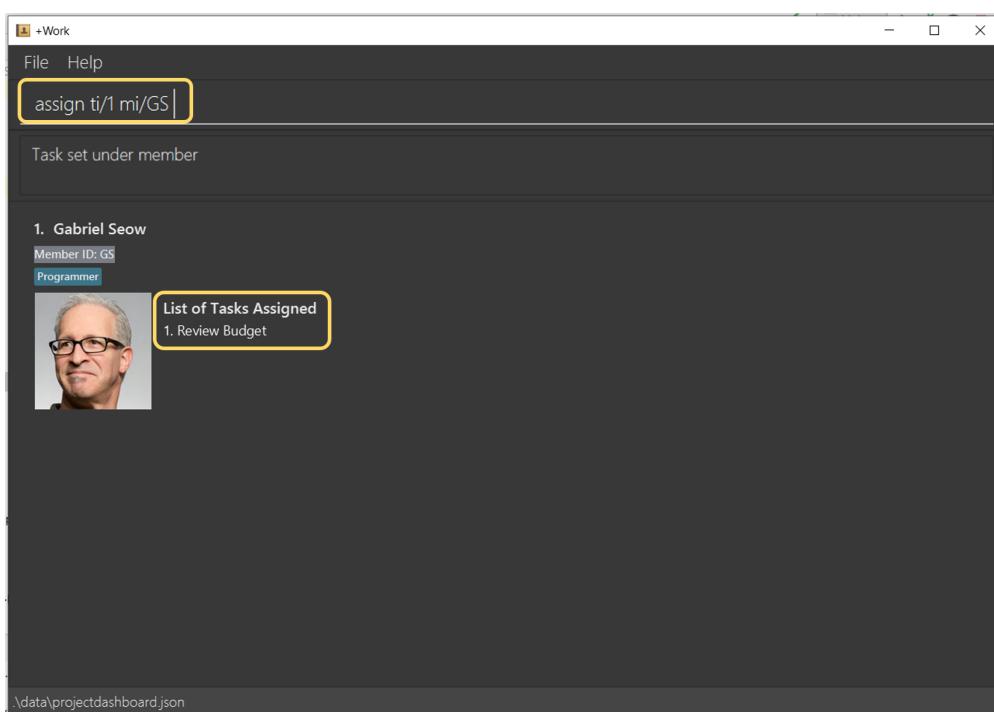
Format: `assign ti/TASK_ID mi/MEMBER_ID`

Example: `assign ti/1 mi/GS` can be executed as follows:

**Step 1:** From the list of tasks shown below, you decide to assign the task 'Review Budget' to project member 'Gabriel Seow' with member ID 'GS'. The task 'Review Budget' has task ID 1, prompting you to enter the command `assign ti/1 mi/GS`.



**Step 2:** After you `Enter` the command, the task 'Review Budget' with task ID '1' is added under member 'Gabriel Seow' with member id 'GS', as seen from the image below.



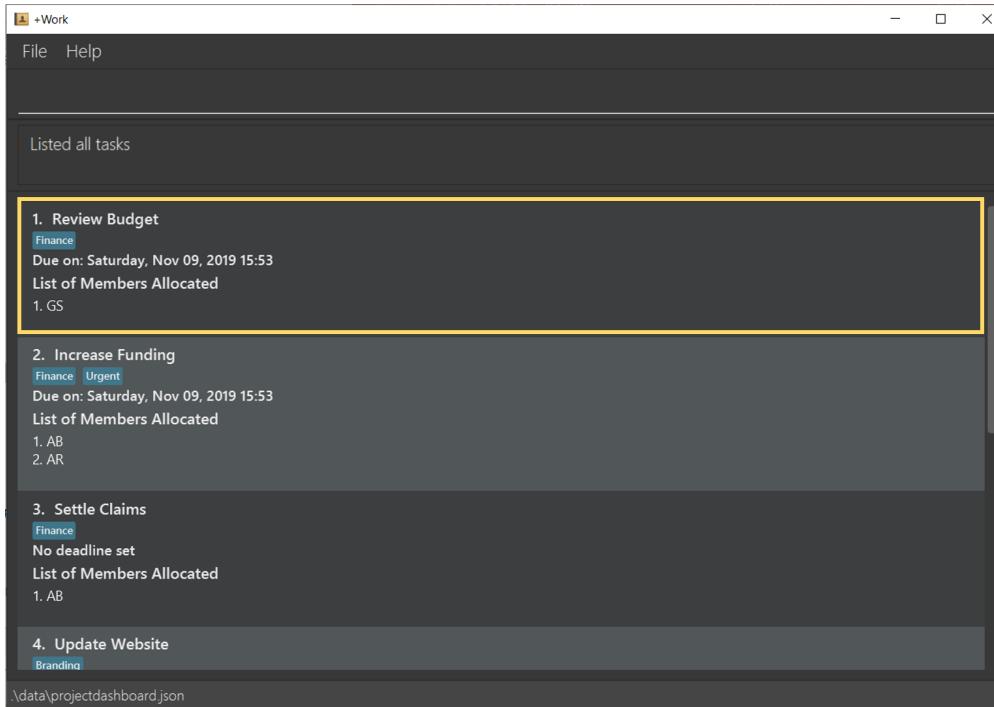
### 3.2.7. Removing a task from a member: fire

To remove a task from a specific team member, use the **fire** command in the format below.

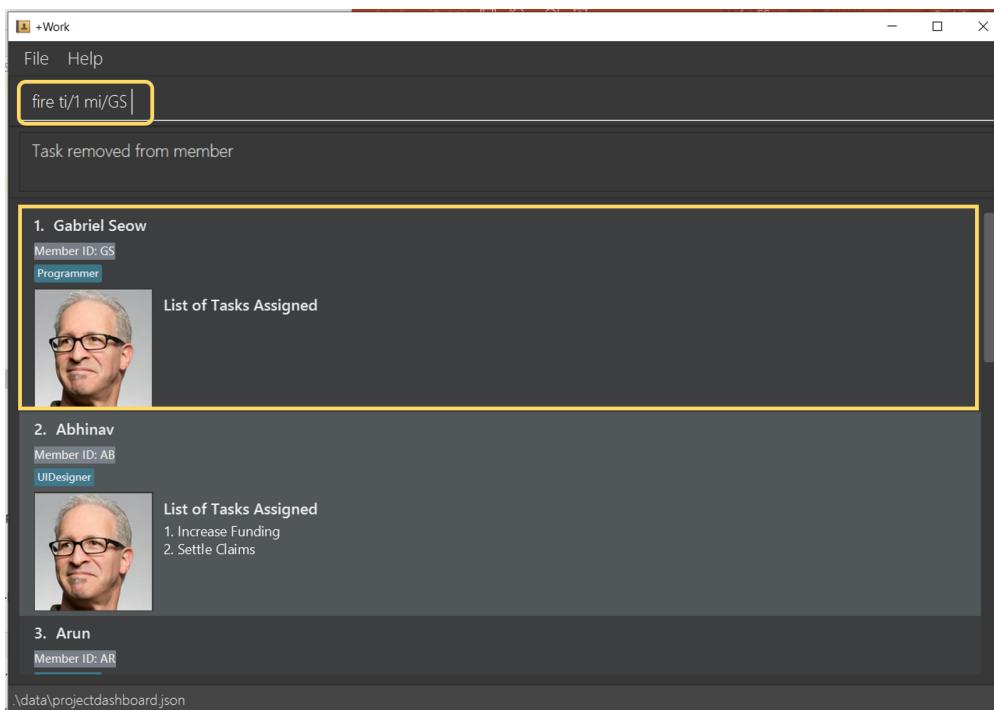
Format: **fire ti/TASK\_ID mi/MEMBER\_ID**

Example: **fire ti/1 mi/GS** can be executed as follows:

**Step 1:** From the list of tasks shown below, you decide to remove project member 'Gabriel Seow' with member ID 'GS' from being assigned to task 'Review Budget'. The task 'Review Budget' has task ID 1, prompting you to enter the command **fire ti/1 mi/GS**.



**Step 2:** After you **Enter** the command, the task 'Review Budget' with task id 1 is removed from member 'Gabriel Seow' with member id 'GS' as seen from the image below.



## 3.3. Task Commands

The following commands will help you manage the tasks in your project.

### 3.3.1. Adding a task: `add-task`

To add a task to the project, use the `add-task` command in the format below.

Format: `add-task [tn/TASK_NAME]`

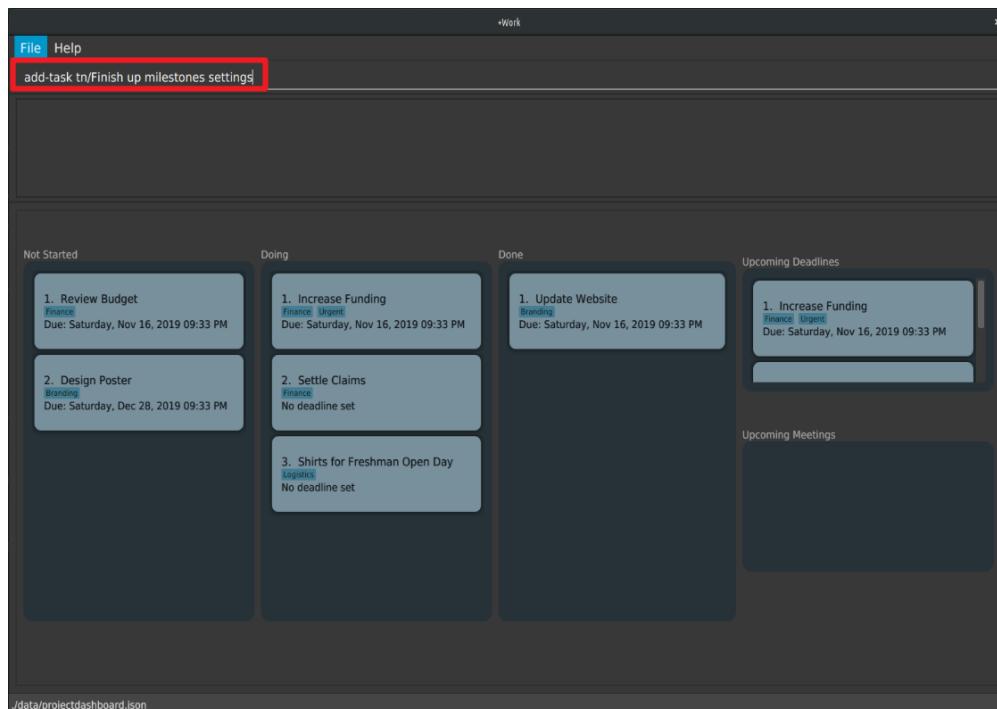
Optional parameters: `[s/STATUS] [tt/TAG]`

**:INFORMATION\_SOURCE:**

If a status is not given for the task, +Work will assign it as `unbegun` or "Not Started" by default.

Example:

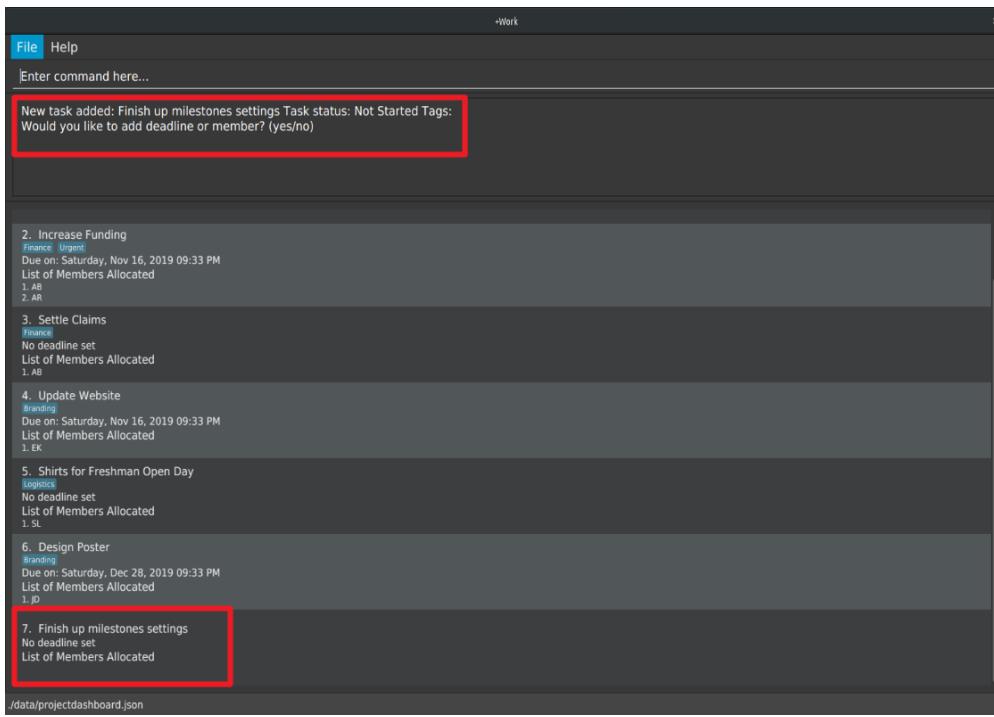
- Suppose you want to add a task called "Finish up milestones settings" to your project. First type `add-task tn/Finish up milestones settings` as show below.



- Hit the `Enter` key and you will see that the task is added to your project!

**IMPORTANT**

The prompt displayed as a result of performing `add-task` is further explained in [Section 3.7, “Multiline Commands”](#).



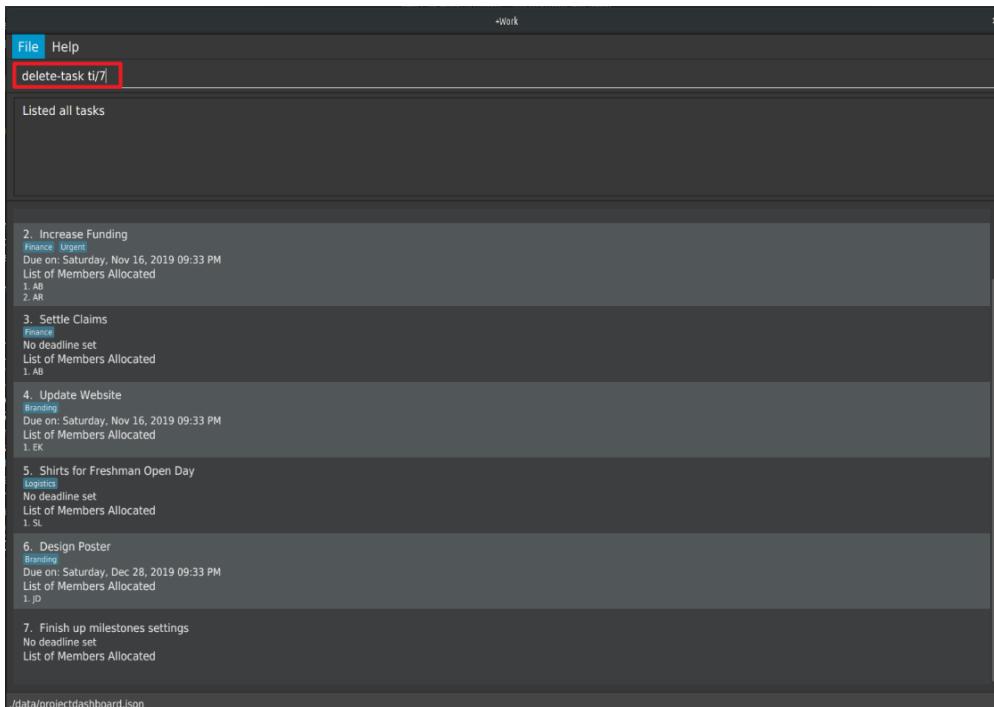
### 3.3.2. Deleting a task: `delete-task`

To delete a task from your project, use the `delete-task` command in the format below.

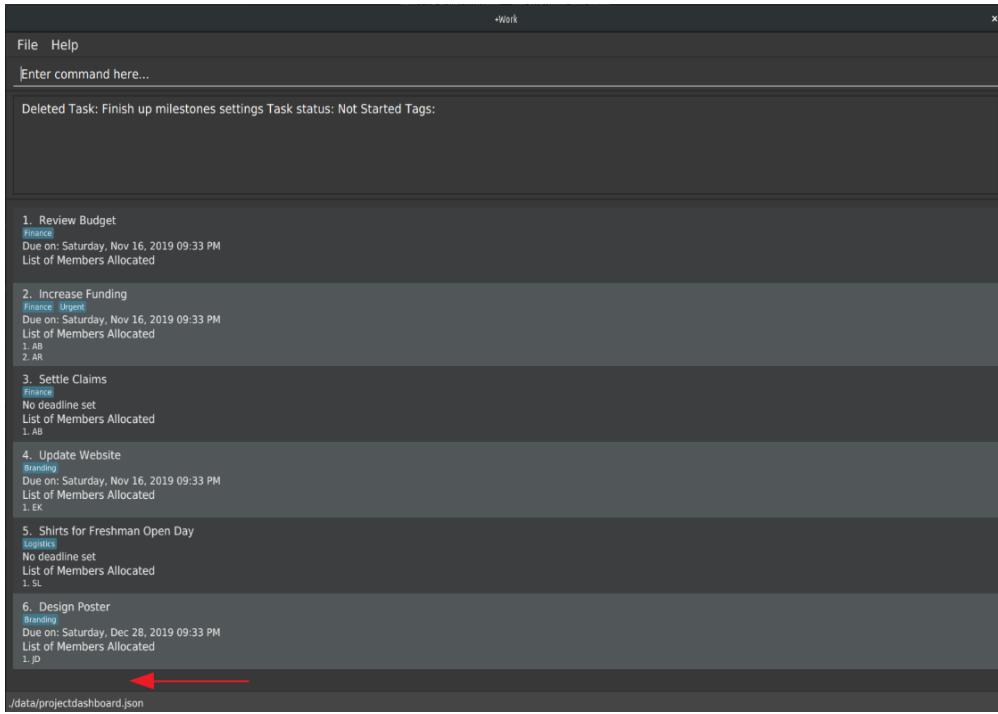
Format: `delete-task [ti/TASK_ID]`

Example:

- Suppose you want to remove the second task in your list. First type `delete-task ti/2` into the command box as shown below.



- Hit the `Enter` key and you will see that the task is removed from your project!



### 3.3.3. Editing a task: edit-task

To edit a task currently in your project, use the **edit-task** command in the format below.

Format: **edit-task [ti/TASK\_ID] [FIELD(S)\_TO\_EDIT]**

Optional parameters: **[tn/TASK\_NAME] [s/STATUS] [tt/TAG] [at/dd-mm-yyyy hh:mm]**

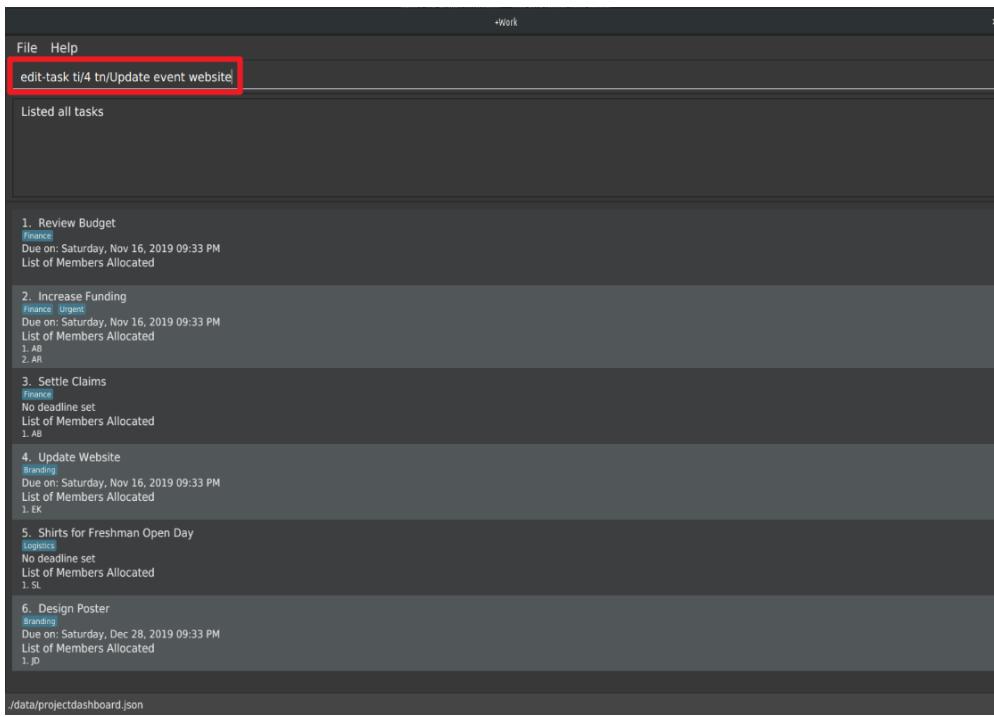
#### IMPORTANT

At least one of the optional parameters must be provided in the command for a task to be edited successfully.

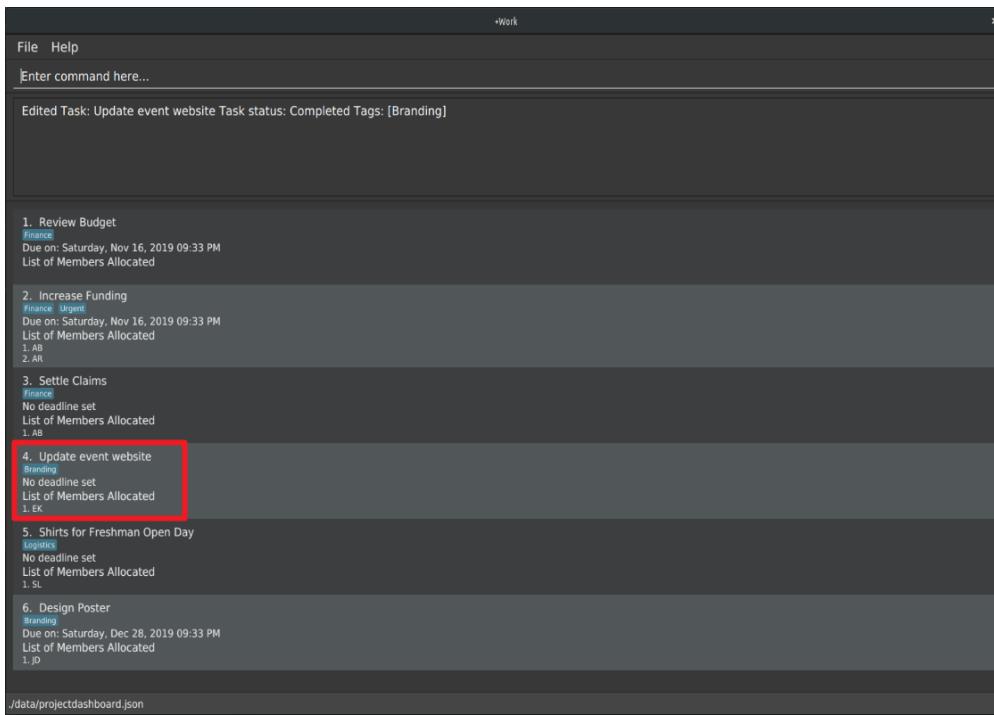
Example:

- Suppose you wanted to update the task name of the fourth task in the diagram below to become "Update event website".

Type **edit-task ti/4 tn/Update event website** into the command box as shown below.



- Hit the `Enter` key and you will see that the task name has been edited!



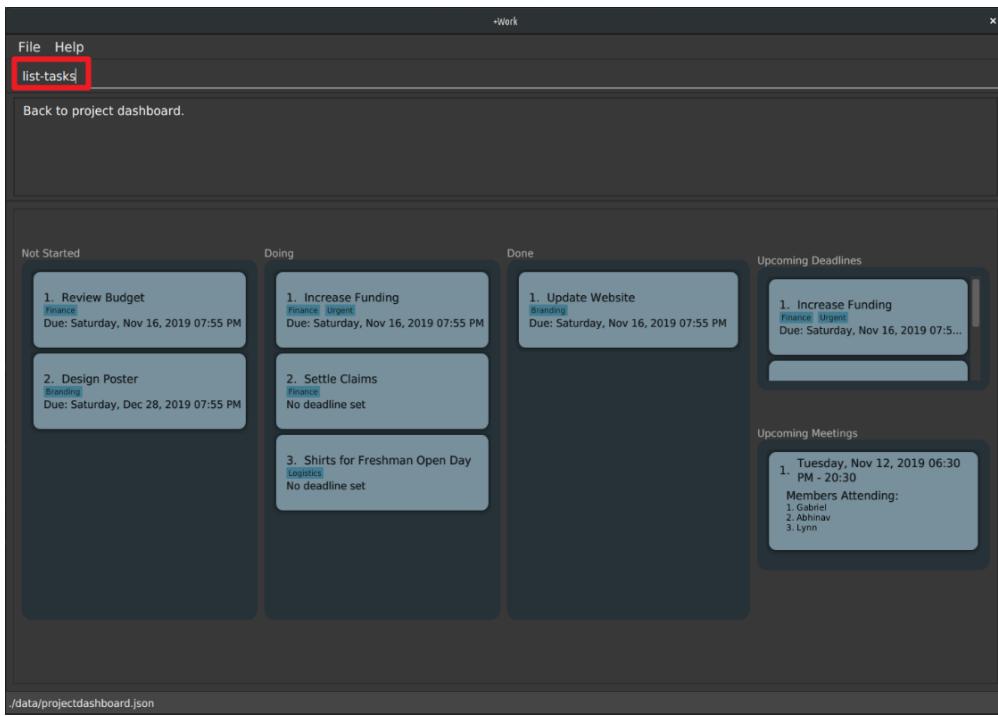
### 3.3.4. Listing all existing tasks: `list-tasks`

To list all the tasks created for the project, use the `list-tasks` command in the format below from any view you are in.

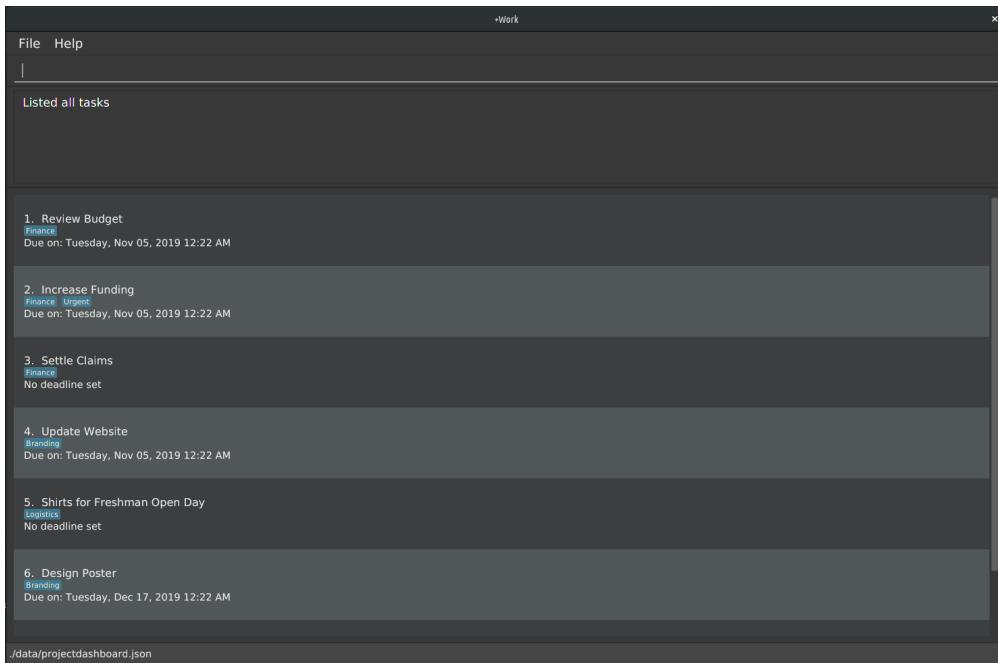
Format: `list-tasks`

Example:

- Type `list-tasks` in the command box as shown below.



- Hit the `Enter` key and you will see all your tasks! You should see a window similar to the one below.

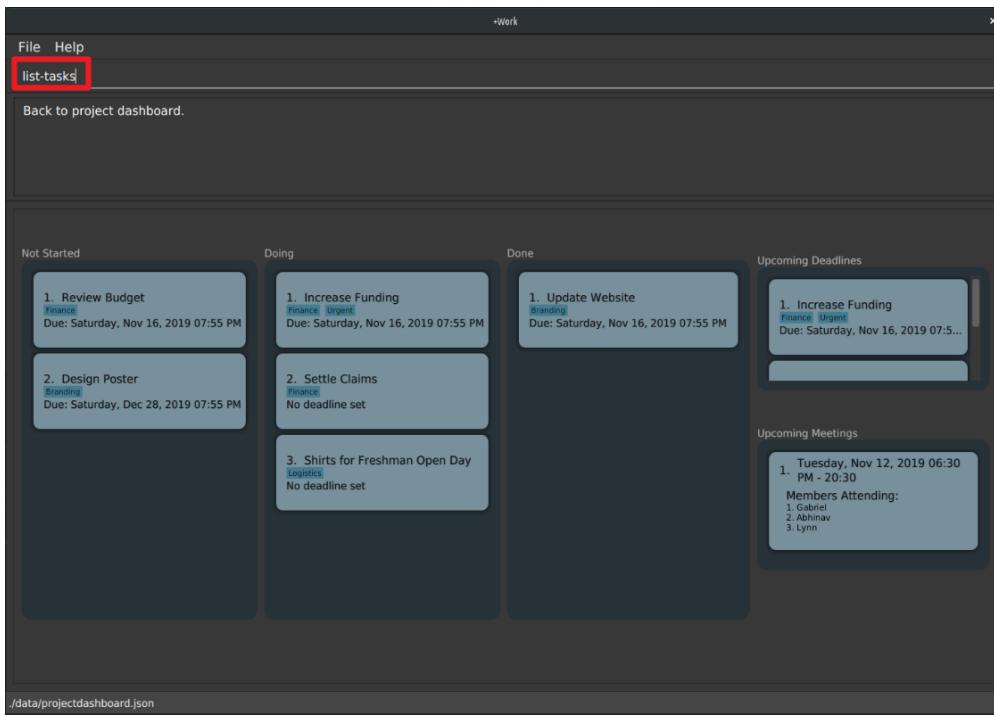


### 3.3.5. Setting a task's status to `done`: done-task

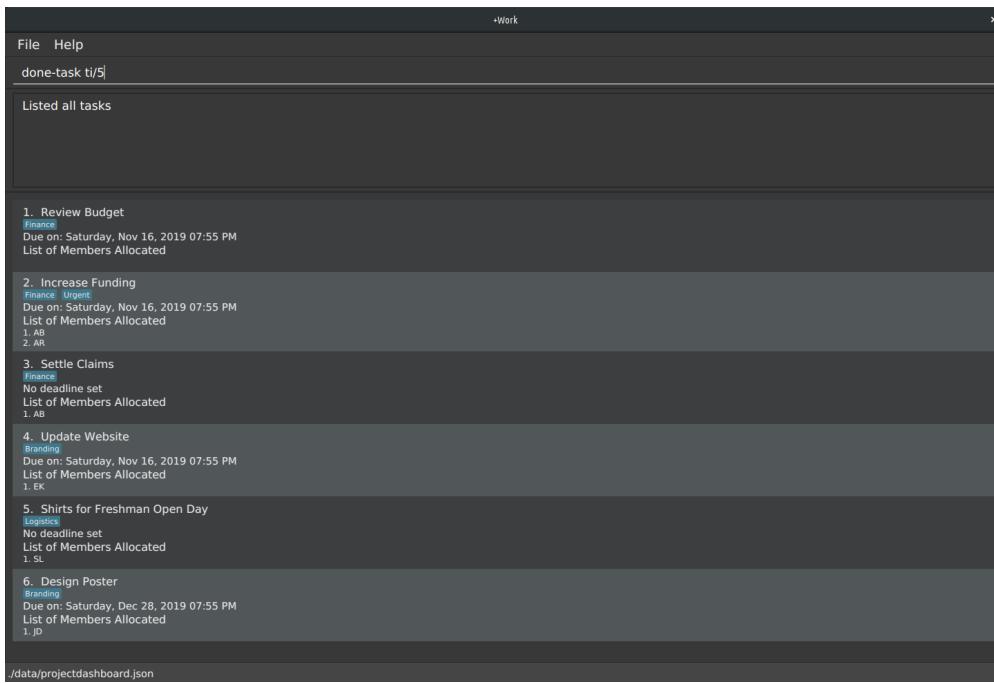
To update the task status to `done`, use the `done-task` command in the format below.  
Format: `done-task [ti/TASK_ID]`

Example:

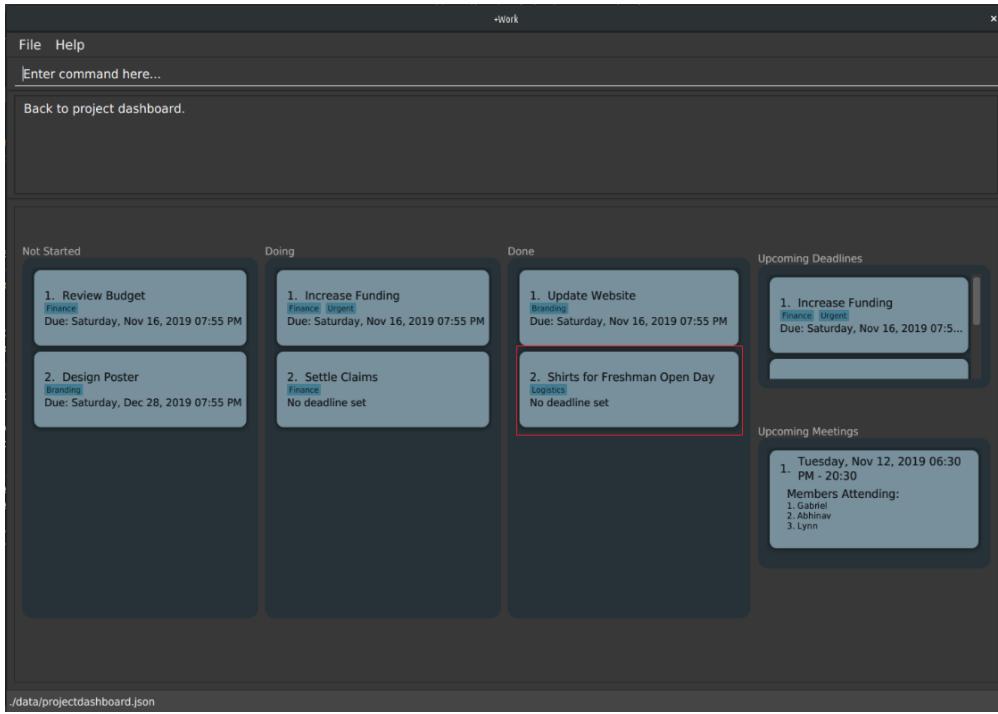
- After finishing the task "Shirts for Freshman Open Day" shown below, you would want to mark it as completed. To do so, first navigate to task list view using `list-tasks`.



- Type `done-task ti/5` into the command box as shown below.



- Hit the `Enter` key and the task will be marked as **done**! You should see a window similar to the one below when you navigate back to `home`. Notice that "Shirts for freshman open day" has been moved to **done**.



:INFORMATION\_SOURCE:

If a task previously had a deadline, it will be removed once it is marked as **done**

### 3.3.6. Setting a task's status to **doing**: doing-task

To update the task status to **doing**, use the **doing-task** command in the format below.

Format: **doing-task [ti/TASK\_ID]**

Example:

- **doing-task ti/3**

This sets the status of the third task to **doing**.

:BULB:

Usage of this command is very similar to setting a task status to **doing** as explained above.

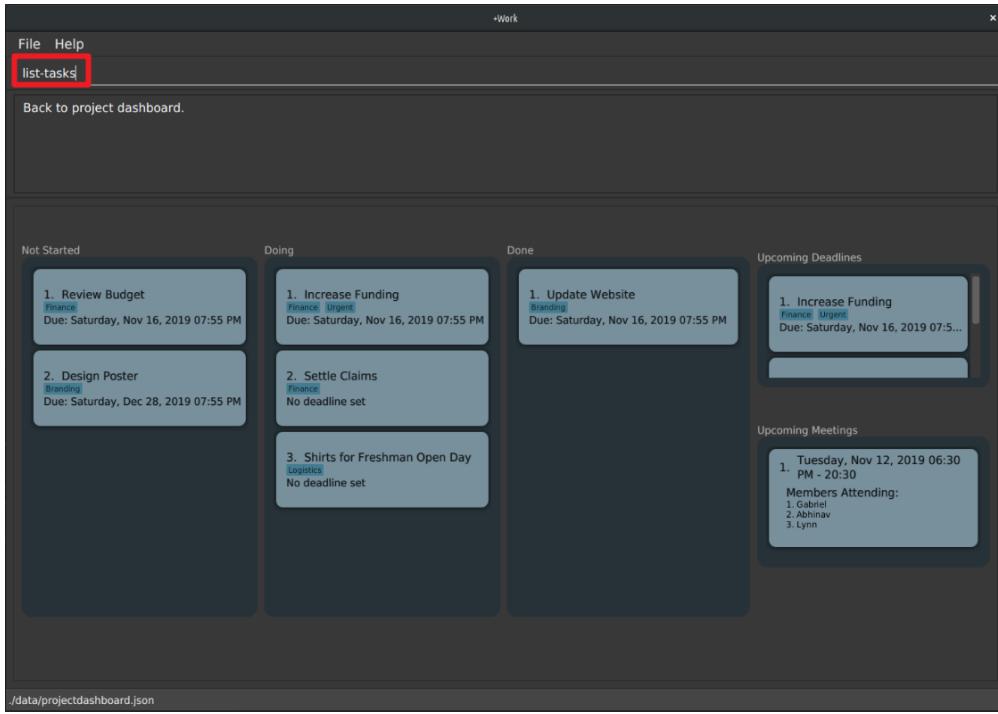
### 3.3.7. Setting a deadline for a task: set-deadline

This sets a deadline for an existing task in your project.

Format: **set-deadline [ti/TASK\_ID] [at/DEADLINE] [at/dd-mm-yyyy hh:mm]**

Example:

- To set a deadline for the task "Shirts for Freshman Open Day" shown below, first navigate to the task list view using **list-tasks**.



- Suppose the deadline for your task is on the 20th of November 2019 at 6pm, enter `set-deadline ti/5 at/20-11-2019 18:00` into the command box as shown below.

[set deadline list preview] | *set-deadline-list-preview.png*

- Hit the `Enter` key and you will see that a deadline has been set!.

[set deadline res list] | *set-deadline-res-list.png*

Now, navigate to the dashboard by entering `home`. Hit the `Enter` key and you will see that the deadline can be seen! You should see a window similar to the one below.

[set deadline res dashboard] | *set-deadline-res-dashboard.png*

**:BULB:**

The "Upcoming deadlines" sidebar (right side of above picture) helps you keep track of tasks which are due in **two weeks** or less.

## 3.4. Calendar Commands

This section contains the commands for managing your team member's calendars and for scheduling a project meeting time.

## Before you start using +Work's calendar commands, make sure

1. Your team members have exported their calendars as an **.ics** file
- :BULB:** Not sure how to export a calendar? Refer to the short guide below for instructions
2. You have collected the calendar files from your team members
  3. You take note of the file path where the files are stored, it will be used in the commands

### How to export a calendar as an **.ics** file?

1. On the NUSmods page, click on the **Download** icon and select download as an iCalendar File(**.ics**)
2. If you wish to include other commitments, simply open 'Google Calendar', click on **Settings**, select **Import & Export** and import the file from Step 1
3. Add any additional commitments through 'Google Calendar'
4. Export the calendar again by clicking **Settings**, followed by **Import & Export** and lastly **Export**
5. The **.ics** file will be downloaded, containing your timetable from NUSmods and 'Google Calendar'

### 3.4.1. Adding a team member's calendar: **add-calendar**

You can add a calendar by inputting the file path of the team member's calendar as well as the name of the team member.

Format: **add-calendar [mn/MEMBER\_NAME] [fp/PATH\_TO\_ICS\_FILE]**

Example:

Adding *John Doe*'s calendar to +Work

- **add-calendar mn/John Doe fp/C:\Users\gabriel\TeamCalendars\john\_calendar.ics**

### 3.4.2. Removing a team member's calendar: **delete-calendar**

You can also remove a calendar by specifying the team member's name.

Format: **delete-calendar [mn/MEMBER\_NAME]**

Example:

Removing *John Doe*'s calendar from +Work

- `delete-calendar mn/John Doe`

### 3.4.3. Find possible meeting times: `find-meeting-time`

You can schedule a meeting by specifying the **duration** of the meeting in hour(s) and the **time period** to search for.

+Work will show a list of suitable meeting time between `START_DATE` and `END_DATE`.

Format: `find-meeting-time [start/START_DATE] [end/END_DATE] [hours/DURATION]`

:INFORMATION\_SOURCE:

+Work recognises date and time in the format 'dd-mm-yyyy  
hh:mm'

:BULB:

+Work shows you the meeting times where the **most** number of people are available

Example:

- Let's say you want to schedule a 2 hour meeting in the upcoming week, between *11th Nov 8 a.m* and *15th Nov 5 p.m*. After entering the details in the correct format, as such

The screenshot shows the +Work project dashboard with a search bar at the top containing the command `find-meeting-time start/11-11-2019 08:00 end/15-11-2019 17:00 hours/2`. Below the search bar, there are four main sections: Not Started, Doing, Done, and Upcoming Deadlines. In the Doing section, there are three tasks: '1. Increase Funding' (Finance, Urgent), '2. Settle Claims' (Finance, No deadline set), and '3. Shirts for Freshman Open Day' (Logistics, No deadline set). In the Upcoming Deadlines section, there is one task: '1. Increase Funding' (Finance, Urgent). On the right side, under 'Upcoming Meetings', there is a box showing a meeting scheduled for Friday, Nov 29, 2019, from 06:00 PM - 10:00 PM, with members attending: Gabriel, Abhinav, and Lynn.

- Hit `Enter` and +Work will display a list of suitable timings as well as the team members that are available for that timing

```

File Help
Enter command here...
Found a meeting time between Monday, Nov 11, 2019 08:00 AM - Friday, Nov 15, 2019 05:00 PM

Meeting Time

1. Monday, Nov 11, 2019 08:00 AM - 10:00 AM
Members Attending:
1. Gabriel
2. Abhinav
3. Lynn

2. Monday, Nov 11, 2019 08:30 AM - 10:30 AM
Members Attending:
1. Gabriel
2. Abhinav
3. Lynn

3. Monday, Nov 11, 2019 09:00 AM - 11:00 AM
Members Attending:
1. Gabriel
2. Abhinav
3. Lynn

4. Monday, Nov 11, 2019 09:30 AM - 11:30 AM
Members Attending:
1. Gabriel
2. Abhinav
3. Lynn

5. Monday, Nov 11, 2019 10:00 AM - 12:00 PM
Members Attending:
1. Gabriel
2. Abhinav
3. Lynn

6. Monday, Nov 11, 2019 10:30 AM - 12:30 PM
Members Attending:
1. Gabriel
2. Abhinav
3. Lynn

7. Monday, Nov 11, 2019 11:00 AM - 01:00 PM
Members Attending:

```

#### :INFORMATION\_SOURCE:

#### IMPORTANT

If there are no suitable timings, +Work will notify you as well

Because showing **ALL** possible meeting timings may not be appropriate, +Work helps by restricting the meeting timings to be between 8 a.m and 10 p.m

### 3.4.4. Schedule a team meeting: add-meeting

After using the command **find-meeting-time**, you can schedule a meeting from the list of possible timings by referring to the **INDEX** of the meeting in the list.

Format: **add-meeting [meeting/INDEX]**

Example:

- Suppose you are looking to schedule a 2 hour meeting between *11th Nov 8 a.m and 15th Nov 5 p.m.*
- After using the **find-meeting-time** command, you are given the following timings

File Help

Enter command here...

Found a meeting time between Monday, Nov 11, 2019 08:00 AM - Friday, Nov 15, 2019 05:00 PM

3. Monday, Nov 11, 2019 09:00 AM - 11:00 AM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

4. Monday, Nov 11, 2019 09:30 AM - 11:30 AM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

5. Monday, Nov 11, 2019 10:00 AM - 12:00 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

Meeting Times

6. Monday, Nov 11, 2019 10:30 AM - 12:30 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

7. Monday, Nov 11, 2019 11:00 AM - 01:00 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

8. Monday, Nov 11, 2019 11:30 AM - 01:30 PM  
Members Attending:

• After looking through the suitable timings, you choose meeting #5 as your preferred timing

4. Monday, Nov 11, 2019 09:30 AM - 11:30 AM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

5. Monday, Nov 11, 2019 10:00 AM - 12:00 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

6. Monday, Nov 11, 2019 10:30 AM - 12:30 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

7. Monday, Nov 11, 2019 11:00 AM - 01:00 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

8. Monday, Nov 11, 2019 11:30 AM - 01:30 PM  
Members Attending:  
1. Gabriel  
2. Abhinav  
3. Lynn

9. Monday, Nov 11, 2019 04:00 PM - 06:00 PM  
Members Attending:

- Using the index of meeting #5, enter the command **add-meeting meeting/5**
- You can then view the recently added meeting at the **home** page

The screenshot shows a project management interface with the following details:

- Not Started:**
  - 1. Review Budget (Finance, Due: Monday, Nov 18, 2019 01:47 AM)
  - 2. Design Poster (Branding, Due: Monday, Dec 30, 2019 01:47 AM)
- Doing:**
  - 1. Increase Funding (Finance, Urgent, Due: Monday, Nov 18, 2019 01:47 AM)
  - 2. Settle Claims (Finance, No deadline set)
  - 3. Shirts for Freshman Open Day (Branding, URGENT, No deadline set)
- Done:**
  - 1. Update Website (Branding, Pending, Due: Monday, Nov 18, 2019 01:47 AM)
- Upcoming Deadlines:**
  - 1. Increase Funding (Finance, Urgent, Due: Monday, Nov 18, 2019 01:47 AM)
  - 2. Settle Claims (Finance, No deadline set)
  - 3. Shirts for Freshman Open Day (Branding, URGENT, No deadline set)
- Upcoming Meetings:**
  - 3. Shirts for Freshman Open Day (Meeting #3, Due: Monday, Nov 11, 2019 10:00 AM - 12:00 PM)
    - Members Attending: L. Gabriel, S. Akhilesh, S. Lynn

### 3.4.5. Remove a team meeting: `delete-meeting`

You can remove a meeting by simply referring to the [INDEX](#) of the meeting in the 'Upcoming Meetings' list.

**:BULB:** You can view your list of meetings by going to the [home](#) page

Format: `delete-meeting [meeting/INDEX]`

Example:

- To remove meeting #3, simply enter the command `delete-meeting meeting/3` and the meeting will be removed

## 3.5. Inventory Commands

### 3.5.1. Adding an inventory: `add-inv`

This command allows you to add an inventory bought or retrieved for a specific task by a specific member.

Format: `add-inv [i/ITEM_NAME] [p/PRICE(optional)] [ti/TASK_ID] [mi/MEMBER_ID]`

#### IMPORTANT

The item name, task id and member id are compulsory inputs. An input without any price value will automatically set the price to \$0. Ensure that an existing task id and member id (as displayed by [list-tasks](#) and [list-members](#) respectively) is being typed into the command box.

Examples:

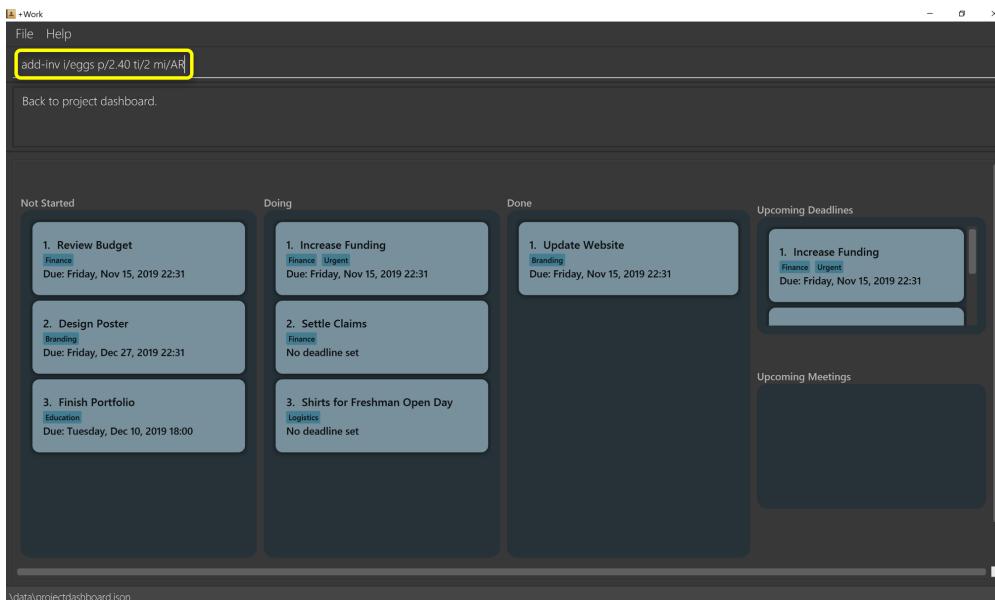
- `add-inv i/scissors ti/4 mi/GS`

When you enter this command, it adds the inventory "scissors" to the inventory list. The item is tagged to task with id as 4 (id is shown by [list-tasks](#)) and is provided by member with the member id "GS" for a price of \$0.

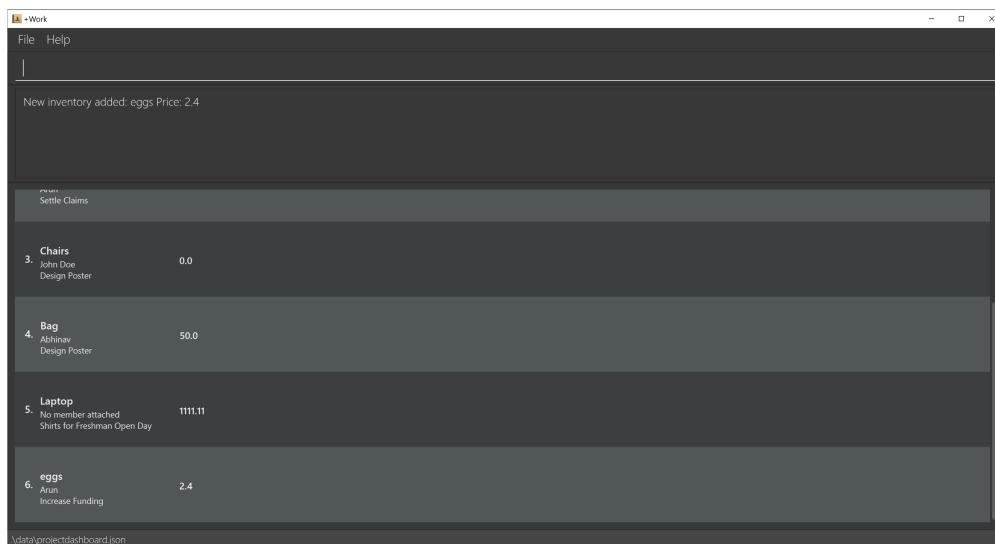
- **add-inv i/eggs p/2.40 ti/2 mi/AR**

When you enter this command, it adds the item “eggs” for \$2.40 to the inventory list. This item is tagged to task tagged to task with id as 2 (id is shown by [list-tasks](#)) and was paid for by the member with member id “AR”. The following pictures show how this command is to be executed.

- First, type **add-inv i/eggs p/2.40 ti/2 mi/AR** into the command box as shown below.



- Then, hit the **Enter** key and you will see that the inventory has been added! You should see a window like the one below.



### 3.5.2. Deleting an inventory: **delete-inv**

This command allows you to delete an inventory.

Format: **delete-inv [ii/ITEM\_ID]**

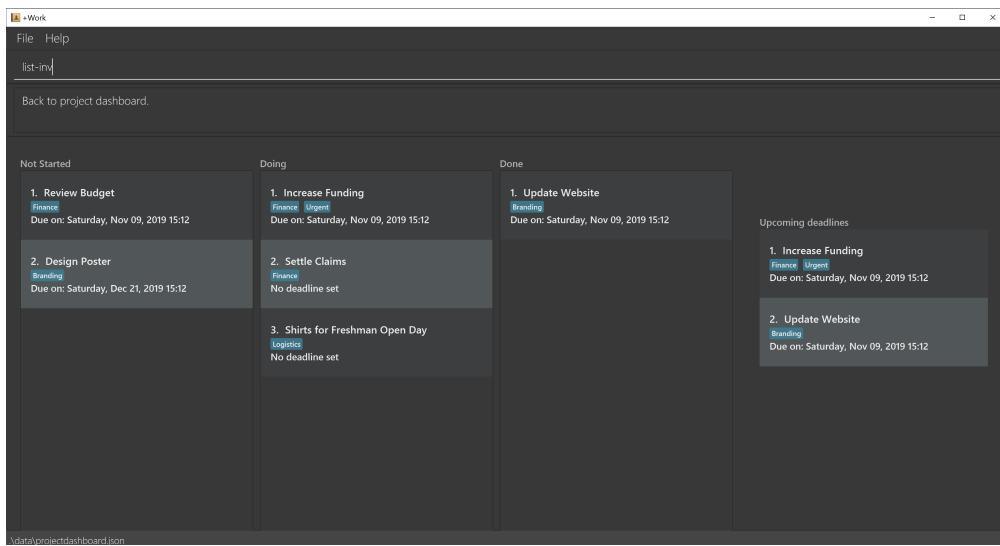
Examples:

- **delete-inv ii/3**

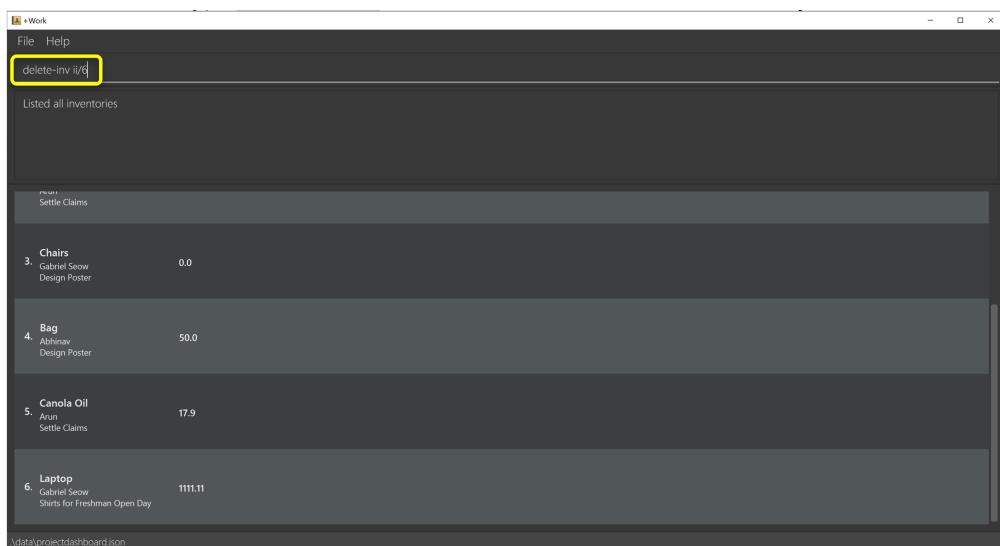
This command deletes the third item from the inventory list. The following pictures show how

this command is to be executed

- First, find the index of inventory that you want to delete, by navigating to inventory list view using **list-inv**.



- Then, type **delete-inv ii/6** into the command box as shown below.



- Hit the **Enter** key and you will see that the inventory has been deleted! You should see a window like the one below.

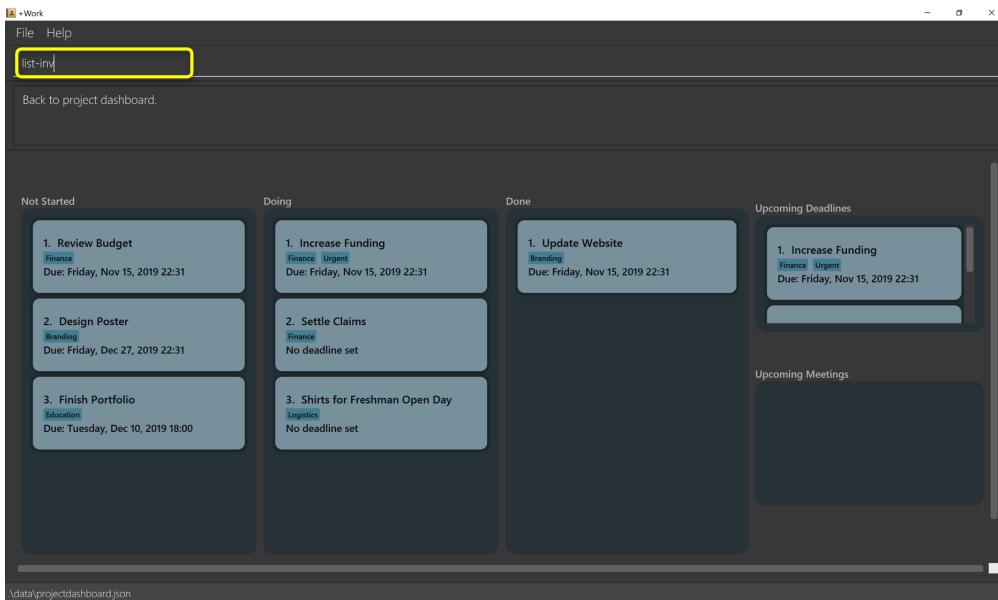


### 3.5.3. List existing inventories: `list-inv`

To get a list of all inventories added to +Work, used the command `list-inv` following the format below.

Format: `list-inv`

Example: Entering `list-inv` into the command prompt will result in the following:



### 3.5.4. Editing a inventory: `edit-inv`

This command allows you to edit the details of an existing inventory.

Format: `edit-inv [ii/ITEM_ID] (i/ITEM_NAME) (p/PRICE) (ti/TASK_ID) (mi/MEMBER_ID)`

:INFORMATION\_SOURCE:

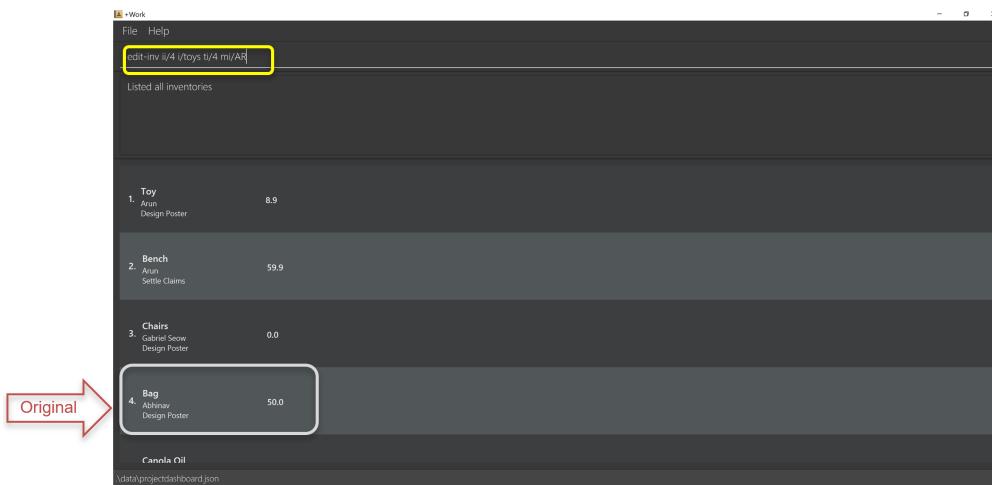
Multiple attributes can be changed at the same time, but at least one of the inventory attributes must be changed. Therefore, this command is invalid: `edit-inv ii/4`. However, this command is valid: `edit-inv ii/4 i/toys ti/4 mi/AR`

Example:

- **edit-inv ii/4 i/toys ti/4 mi/AR**

This command edits the 4th inventory from the inventory list as shown by **list-inv**. It changes the inventory name to toys, the task attached to 4, and the member attached to the member with member ID “AR”. The following pictures show how this command is to be executed.

- First, type **edit-inv ii/4 i/toys ti/4 mi/AR** into the command box as shown below.



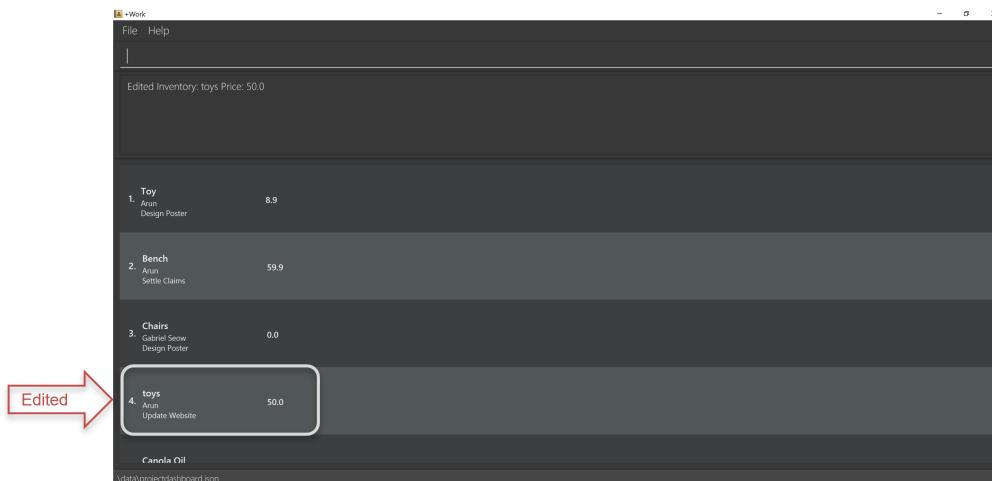
```
edit-inv ii/4 i/toys ti/4 mi/AR
```

Listed all inventories

|           |              |               |      |
|-----------|--------------|---------------|------|
| 1. Toy    | Arun         | Design Poster | 8.9  |
| 2. Bench  | Arun         | Settle Claims | 59.9 |
| 3. Chairs | Gabriel Seow | Design Poster | 0.0  |
| 4. Bag    | Abhinav      | Design Poster | 50.0 |

Canola Oil  
\\data\\project\\dashboard.json

- Hit the **Enter** key and you will see that the 4th inventory has been edited! The name has been changed to toys, task has been changed to the task with index 4, and finally member has been changed to member with index “AR”. You should see a window like the one below.



Edited Inventory: toys Price: 50.0

|           |              |                |      |
|-----------|--------------|----------------|------|
| 1. Toy    | Arun         | Design Poster  | 8.9  |
| 2. Bench  | Arun         | Settle Claims  | 59.9 |
| 3. Chairs | Gabriel Seow | Design Poster  | 0.0  |
| 4. toys   | Arun         | Update Website | 50.0 |

Canola Oil  
\\data\\project\\dashboard.json

### 3.5.5. Creating a report of inventories: pdf

This command allows you to create a PDF report of the existing inventories classified either by the member attached or by the task attached.

Format: **pdf [ty/TYPE]**

#### IMPORTANT

The only two attributes for TYPE are **members** and **tasks**. Other inputs will not work. If a PDF created and is currently open, it has to be closed before another PDF can be created.

Example:

- **pdf ty/members**

This command creates and opens a pdf file of inventories that is classified according to the member attached. The following pictures show how this command is to be executed.

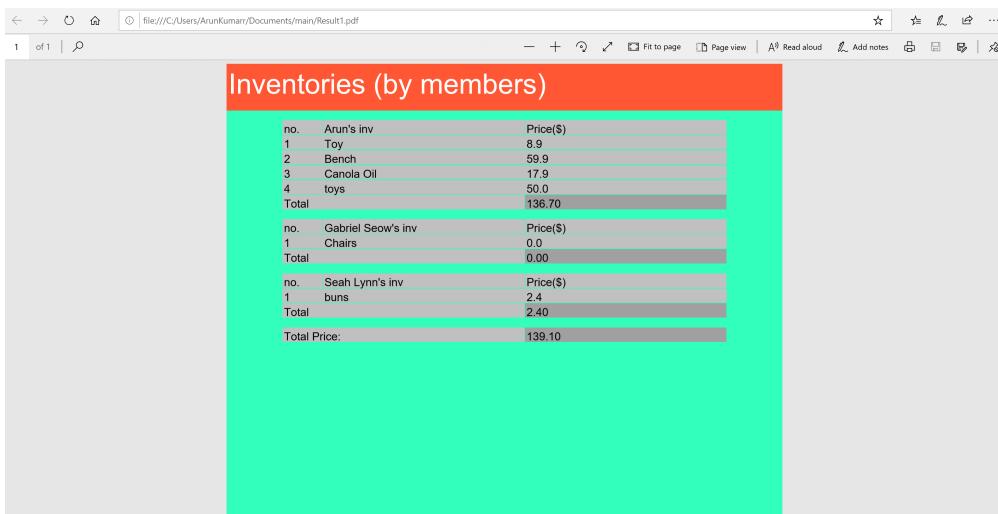
- First, type `pdf ty/members` into the command box as shown below.

The screenshot shows the +Work application interface. At the top, there's a menu bar with 'File' and 'Help'. Below the menu is a search bar containing the command 'pdf ty/members'. Underneath the search bar, the text 'Listed all inventories' is displayed. A list of items is shown, each with a number, name, member name, and price:

- 1. Toy Arun Design Poster 8.9
- 2. Bench Arun Settle Claims 59.9
- 3. Chairs Gabriel Seow Design Poster 0.0
- 4. toys Arun Update Website 50.0

At the bottom left, it says 'Canola Oil' and 'V:\data\project\dashboard.json'.

- Hit the `Enter` key and you will see that the default PDF viewer will open to show the PDF report. You should see a report like the one below.



## 3.6. Statistics Commands

### 3.6.1. Getting statistics of members: `member-stats`

To get statistics relating to the members in +Work, use the statistics command following the format below.

Format: `member-stats`

Calling the `member-stats` command will result in the statistics being displayed as follows:



#### :INFORMATION\_SOURCE:

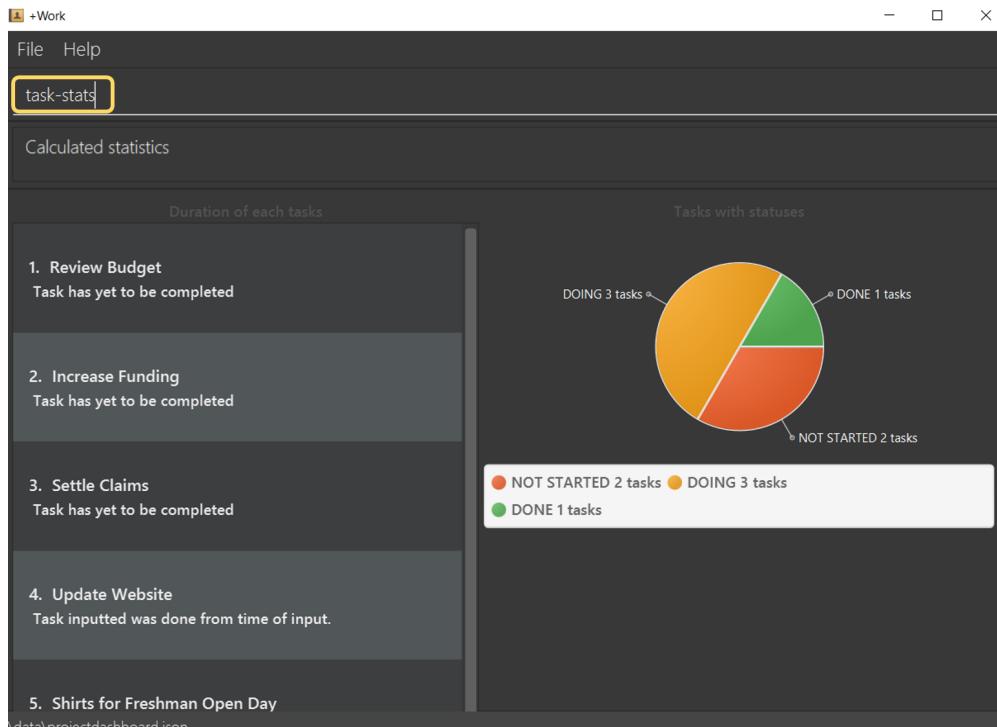
The resultant statistics displayed shows the proportion and number of tasks and inventory items allocated to each project member in +Work.

### 3.6.2. Getting statistics of tasks: **task-stats**

To get statistics relating to the tasks in +Work, use the statistics command following the format below.

Format: **task-stats**

Calling the **task-stats** command will result in the statistics being displayed as follows:



#### :INFORMATION\_SOURCE:

The resultant statistics displayed shows the proportion of tasks that are undone, in progress and completed, as well as the time taken for each task inputted into +Work.

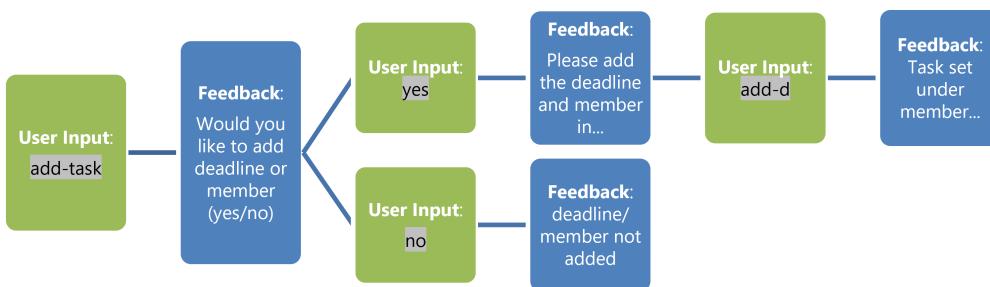
#### :INFORMATION\_SOURCE:

The time spent on each task can only be calculated if the task involved has been marked as **DOING**, and then **DONE**. If you immediately mark an **UNBEGUN** task as **DONE**, the task will be marked as 'done from time of input'.

## 3.7. Multiline Commands

### 3.7.1. Adding a task: add-task

As mentioned in the earlier part of user guide, to add a task to the project, use the **add-task** command in the format below. This will result in a series of questions that allow you to add deadline and a member to the task. The following flowchart shows this series of feedbacks from the application and user inputs.



Format (to add task): **add-task [t/TASK\_NAME]**

Format (for yes): **yes**

Format (for no): **no**

Format (to add deadline and member): **add-d [at/DEADLINE] [mi/MEMBER\_ID]**

**add-d** command does not work unless it is typed after **add-task** and **yes**.

Either deadline or member or both can be added using the **add-d** command. However at least one attribute must be present.

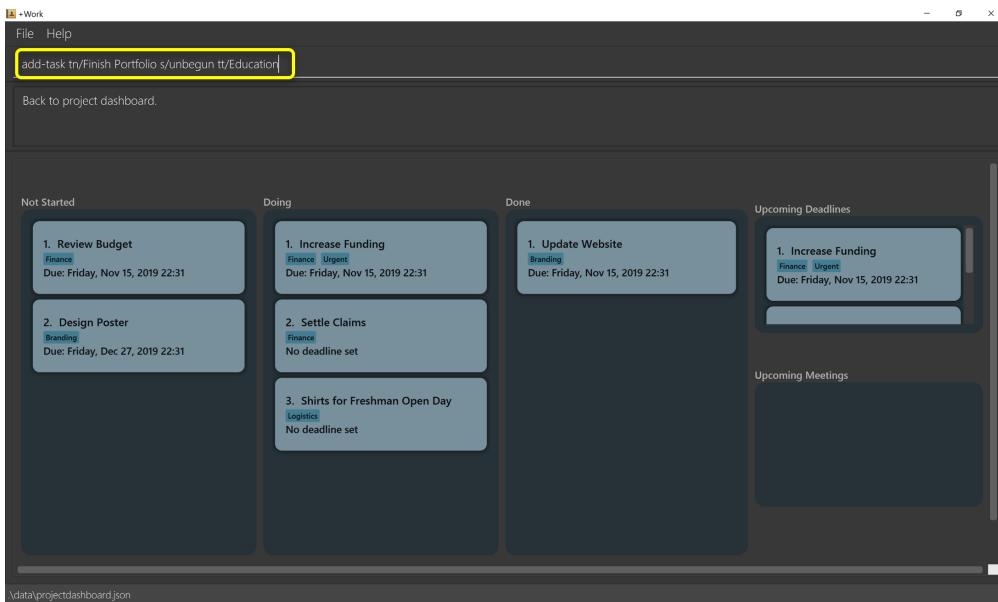
Remember to input deadline using the 24 hour time format of dd-mm-yyyy hh:mm

Example:

- **add-task tn/Finish Portfolio s/unbegun tt/Education**

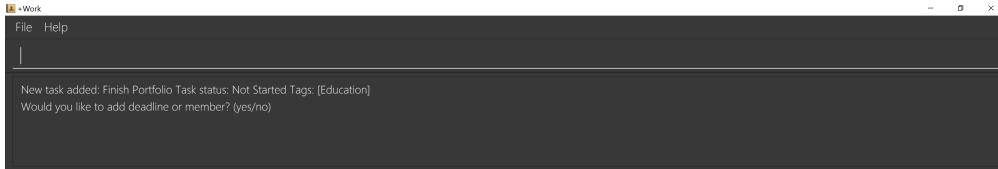
A new task will be added to the project dashboard, which prompts a sequence of questions, as shown in the following pictures

- First, type the **add-task** command

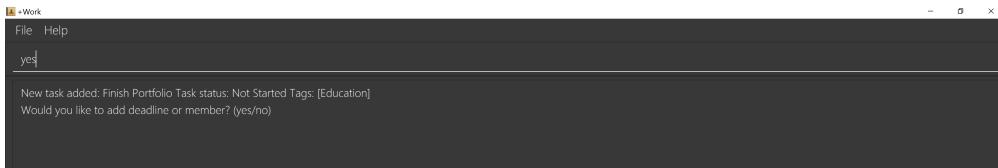


\data\projectdashboard.json

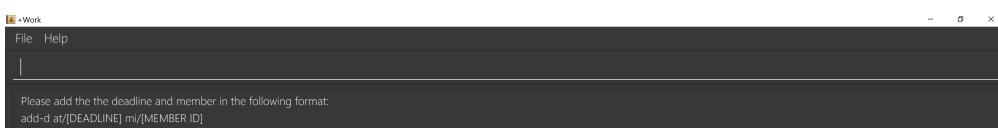
- This prompts the feedback asking for your choice to add deadline and member



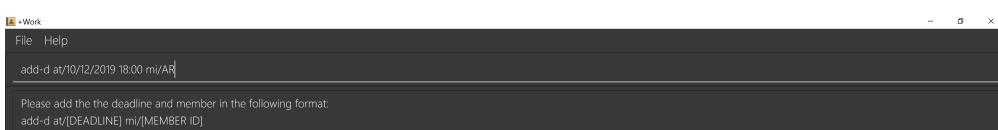
- Type **yes** to add deadline or member



- This prompts to add the details



- Type **add-d at/10-12-2019 18:00 mi/AR** to set deadline at 10-10-2019 18:00 and assign the member with member id “AR” to the task.



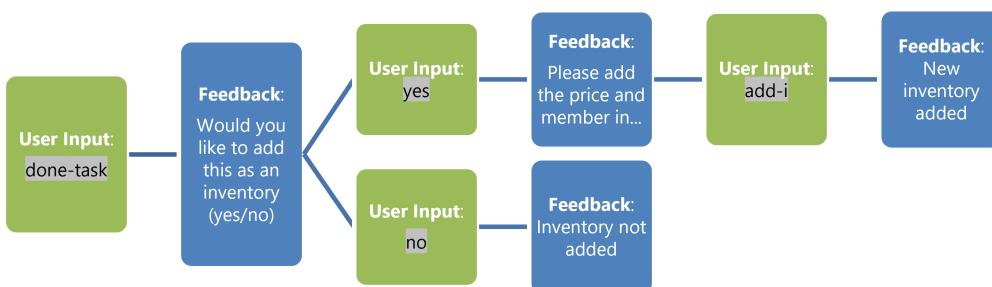
- This sets the deadline and assigns the member successfully! You should see a window like this.



### 3.7.2. Changing a task status to done: **done-task**

**IMPORTANT** only for tasks that are tagged “Inventory”

As mentioned in the earlier part of user guide, to change the task status to “done”, use the **done-task** command in the format below. If the task is tagged as “Inventory”, this will result in a series of questions that allow you to add the task as an inventory. The following flowchart shows this series of feedbacks from the application and user inputs.



Format (for done task): **done-task [ti/TASK\_ID]**

Format (for yes): **yes**

Format (for no): **no**

Format (to add price and member): **add-i [p/PRICE] [mi/MEMBER\_ID]**

**:INFORMATION\_SOURCE:**

**add-i** command does not work unless it is typed after **done-task** and **yes**.

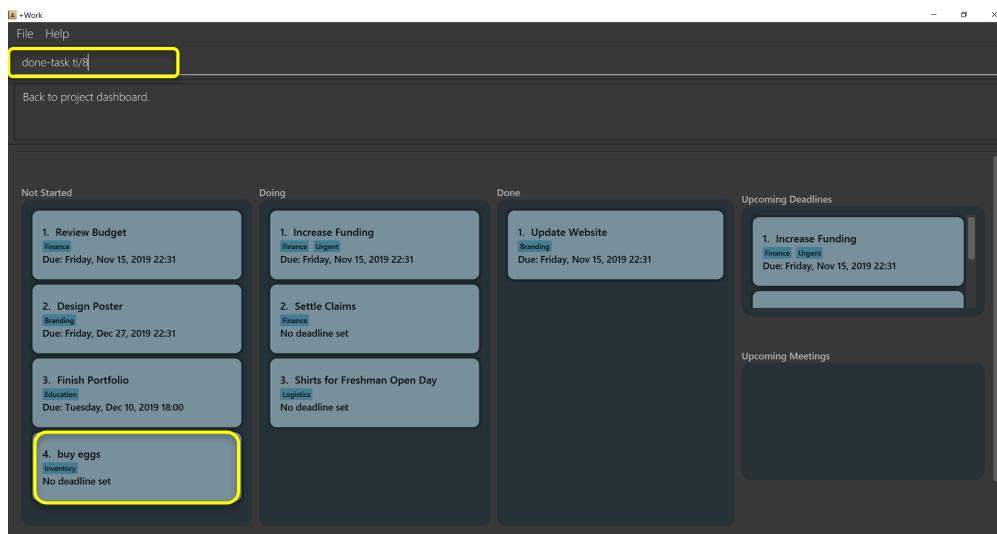
Unlike **add-task**, in this case both attributes price and member id are required.

Example:

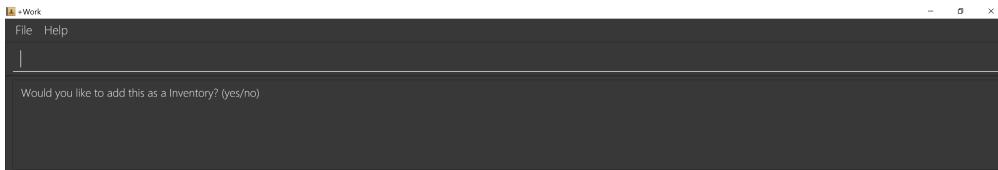
- **done-task ti/8**

The task with index 8 will have its task status changed to done. Since the task is tagged as “Inventory”, this will prompt a series of questions, as shown in the pictures below.

- First, type **done-task ti/8**



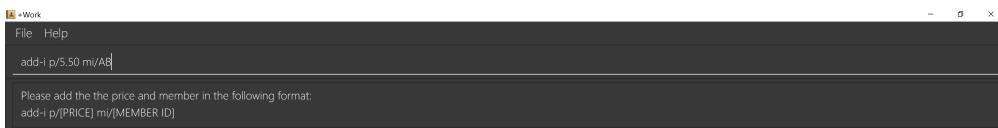
- This prompts the feedback asking for your choice to add it as an inventory.



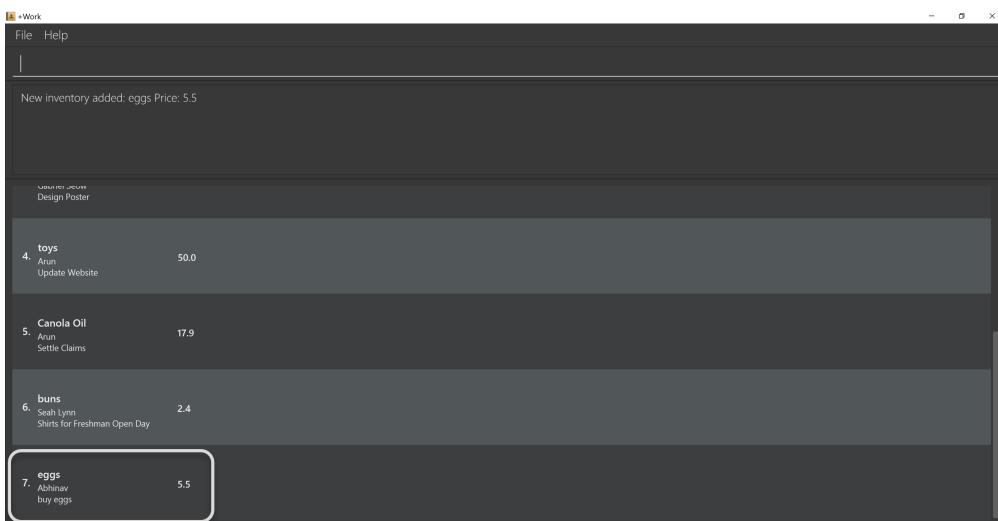
- After typing **yes** as the user input, feedback will prompt you to add the details.



- Type **add-i p/5.50 mi/AB** to set the price at \$5.50 and to assign the member with member id “AB” to the inventory



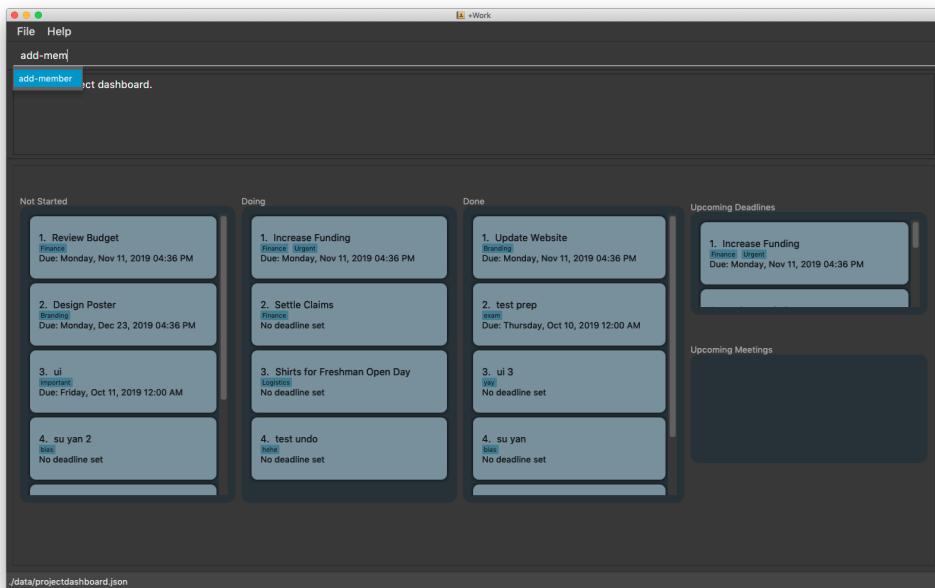
- This sets the price and assigns the member successfully! You should see a window like this.



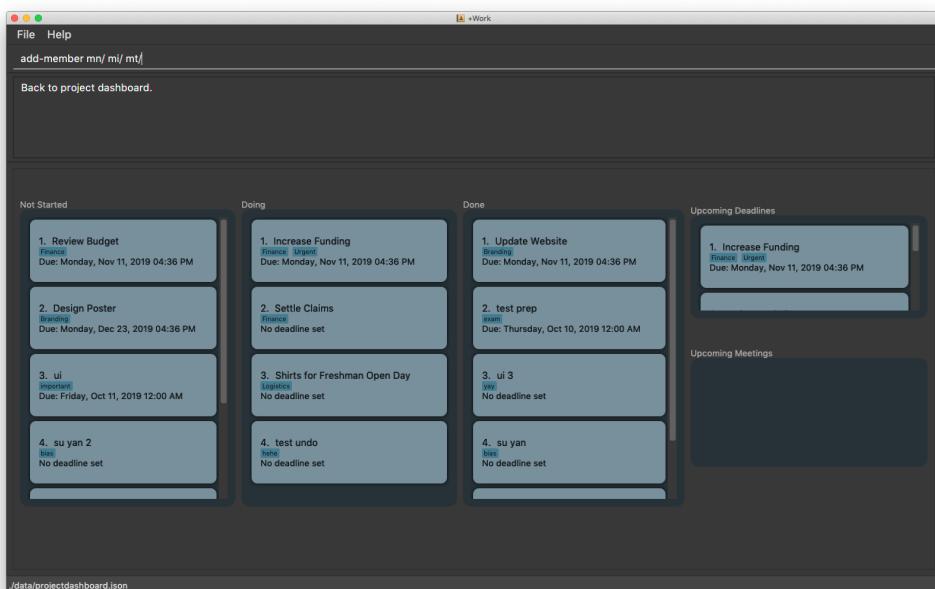
## 3.8. Command Line Autocomplete

+Work will automatically prompt you on the various possible commands based on input and help you paste the correct command format into the command line if chosen.

For example, when you type **add-mem** into the command-line, +Work will prompt you to select **add-member** from the drop down menu.



After selecting `add-member`, +Work will paste `add-member mn/ mi/ mt/` into your command-line automatically!



**:BULB:** You can use `control` to navigate along the various prefixes.

## 3.9. Settings Commands

+Work helps you view your current settings by highlighting your current option! To see your current settings navigate to the settings panel by entering `settings` as described in section 3.1.3.

### 3.9.1. Switching the theme of +Work: `theme`

This command helps you toggle the theme of +Work between `light` and `dark` to suit your viewing

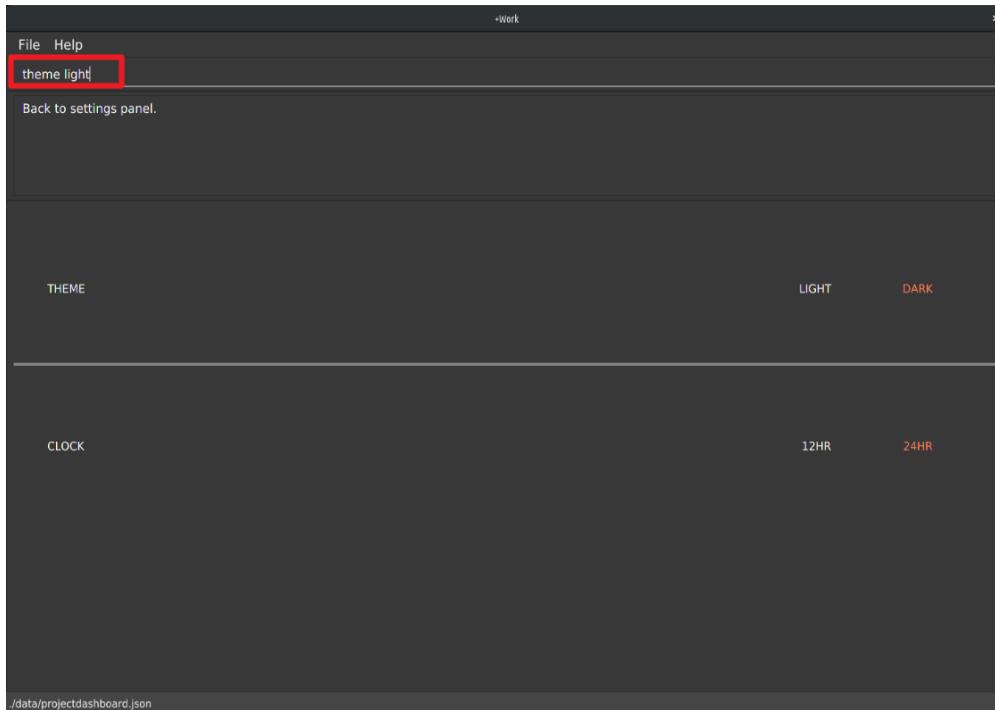
preferences.

**:BULB:** By default, the theme is set to **dark**.

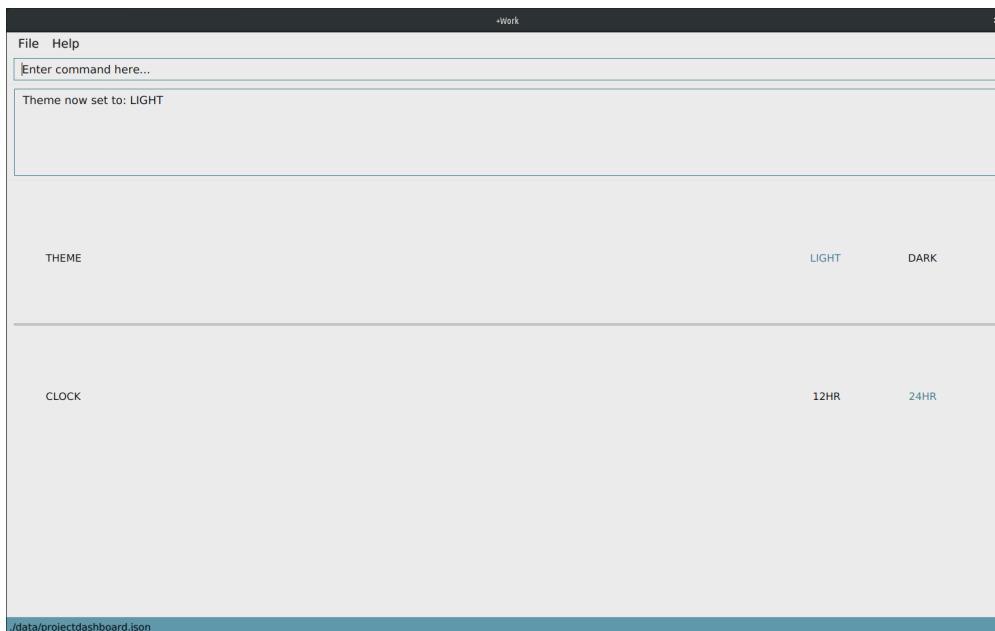
Format: **theme light**

Example:

- Type **theme light** into the command box as shown below.



- Hit the **Enter** key and +Work switches to the **light** theme! As seen below, your choice of **light** is highlighted as well.



- Similarly, **theme dark** switches to the **dark** theme.

### 3.9.2. Switching the time format of +Work: `clock`

This command helps you toggle the time format of +Work between 24 hour and 12 hour clock.

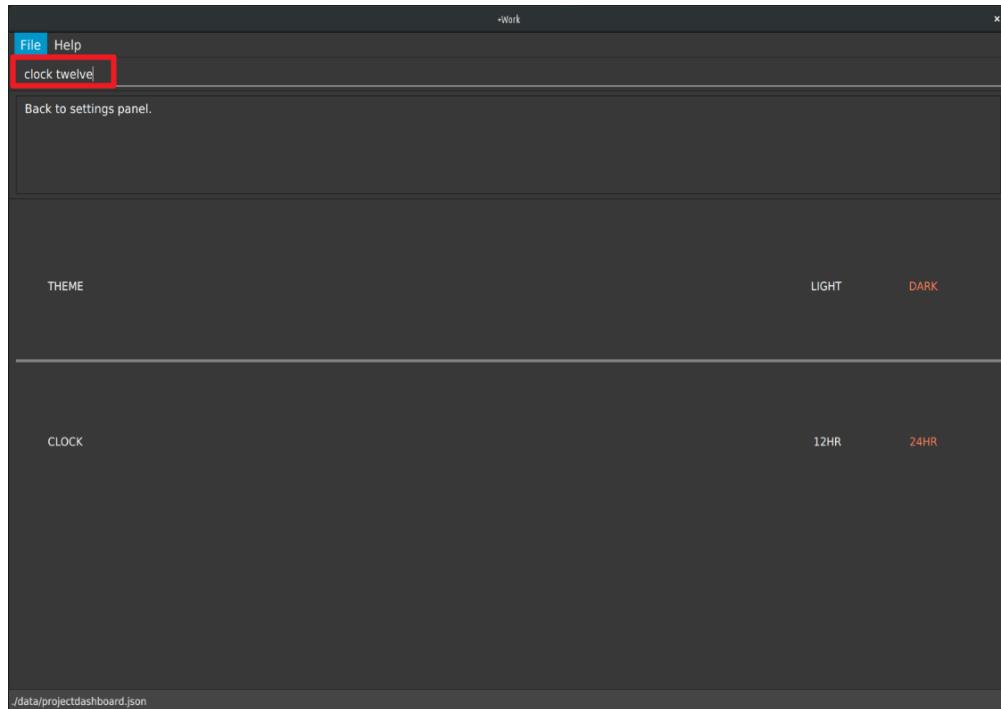
**:BULB:** The time format is set to 24 hour by default.

**:INFORMATION\_SOURCE:** This does not affect the input format of deadlines for tasks, you still need to enter them in the 24 hour format!

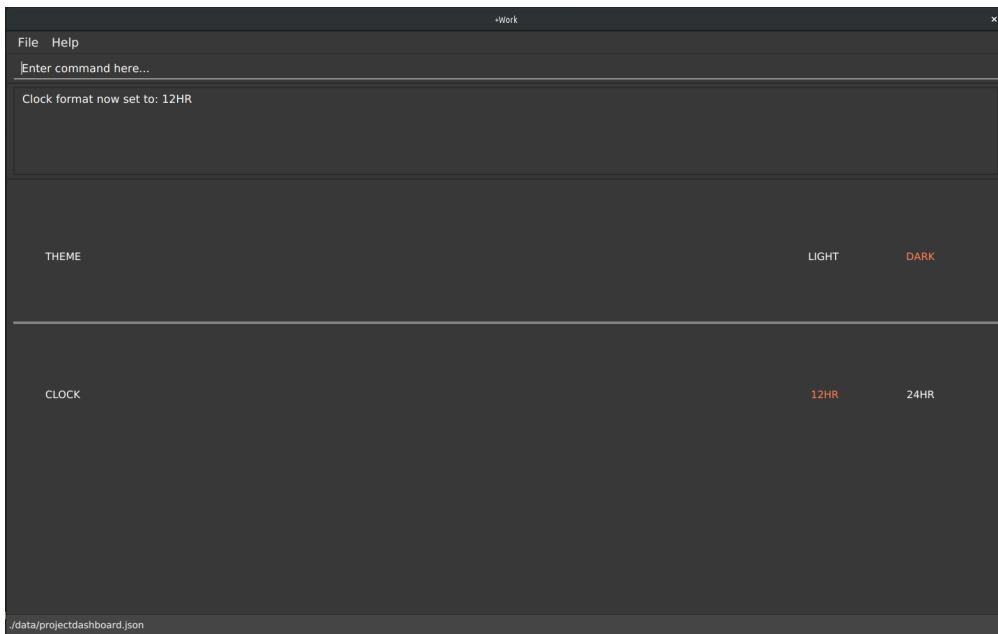
Format: `clock twenty_four`

Example:

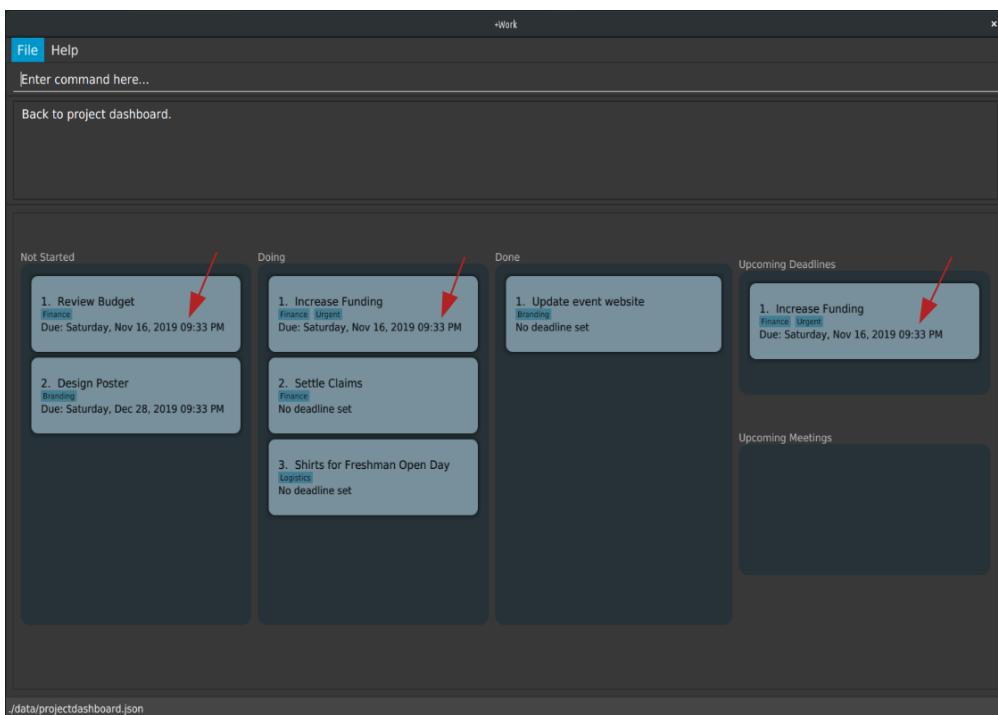
- Type `clock twelve` into the command box as shown below.



- Hit the `Enter` key and +Work switches the time format to the 12 hour clock! As seen below, the new format is highlighted as well.



- Now when you navigate to any view which has time sensitive data, you will see that the format has been switched. For example, if you were to switch back to the dashboard by entering `home` you should see a window similar the one shown below.



- Similarly, `clock twenty_four` switches to the 24 hour clock.

## 3.10. Saving Data

Project data is saved in the hard disk automatically after any command that changes the data. There is no need to save manually.

## 4. FAQ

**Q:** Can I use file formats other than ics for the calendar feature?

**A:** No, the file format has to be in ics folder, downloaded either from [NUSmods](#) or [Google Calendar](#).

**Q:** Can I export the claims report as a word document?

**A:** No, the application only supports exporting of files in PDF format.

**Q:** What if +Work cannot find a timing where everyone is free?

**A:** +Work cannot guarantee to find a meeting time that can suit every team member, but it will provide a list of the best possible timings when the **most** number of members are available.

**Q:** What if the calendar cannot find a timing where everyone is free?

**A:** There will be a list showing the number of available members for the time period you specified, so it will be easier for the user to choose the best possible your meeting.

**Q:** My project has a budget, does the application help me keep track of the projects financial status?

**A:** Using /inventory the application can keep track of current expenses for each task and the member who purchased it, however, there are currently no accounting services available.

**Q:** Some of the tasks for my project require sub tasks to be completed, is there a way to add them?

**A:** Ideally since the user is the project leader, they should only see the main tasks to be accomplished for the project. This would help facilitate their managerial role within the project. Hence, the application does not support subtasks.

**Q:** How can I add multiple members to the same task?

**A:** You can use the **assign** command detailed in in [Section 3.2, “Member Commands”](#).

**Q:** What happens when a task has been completed?

**A:** You can mark the task as **done**, which will move the task to the **done** portion of the dashboard. In the case where you do not want to keep the task in the list, using ‘Remove Task’ will delete it.

## 5. Command Summary

The following is a summary of all the commands +Work offers arranged in alphabetical order.

- **add-calendar [mn/MEMBER\_NAME] [fp/PATH\_TO\_ICS]**: Imports a calendar
- **add-inv [i/NAME] [p/PRICE] [ti/TASKID] [mi/MEMBER\_ID]**: Adds an inventory
- **add-meeting [meeting/index]**: Adds a meeting from the generated meeting time list from **find-meeting-time**
- **add-member [mn/MEMBER\_NAME] [mi/MEMBER\_ID] [mt/MEMBER\_TAG]**: Adds a member
- **add-task [tn/TASK\_NAME] [s/TASK\_STATUS] [tt/TASK\_TAG]**: Adds a task
- **assign [ti/TASK\_ID] [mi/MEMBER\_ID]**: Adds a task to a member
- **clear**: Clears all data from dashboard
- **clock twelve**: Switches deadline display to the 12 hour clock format

- **clock twenty\_four**: Switches deadline display to the 24 hour clock format
- **set-deadline [ti/TASK\_ID] [at/dd-mm-yyyy hh:mm]**: Assigns a deadline to a task
- **delete-calendar [mn/MEMBER\_NAME]**: Deletes calendar of member
- **delete-inv [ii/ITEM\_ID]**: Deletes an inventory
- **delete-meeting [meeting/MEETING\_INDEX]**: Deletes a meeting
- **delete-task [ti/TASK\_INDEX]**: Deletes a task
- **doing-task [ti/TASK\_ID]**: Marks a task as **doing**
- **done-task [ti/TASK\_ID]**: Marks a task as **done**
- **edit-inv [ii/ITEM\_ID] [i/ITEM\_NAME] [p/PRICE] [ti/TASK\_ID] [mi/MEMBER\_ID]**: Edits an inventory
- **edit-member [mi/MEMBER\_ID] [mn/MEMBER\_NAME] [mt/MEMBER\_TAG]**: Edits a member
- **edit-task [ti/TASK\_ID] [FIELDS\_TO\_EDIT]**: Edits a task
- **exit**: Exits +Work
- **fire [ti/TASK\_ID] [mi/MEMBER\_ID]**: Removes a task from a member
- **find-meeting-time [start/dd-mm-yyyy hh:mm] [end/dd-mm-yyyy hh:mm] [hours/HOURS]**: Generates free times among members
- **home**: Switches to the project dashboard
- **help**: Navigates to help page
- **list-members**: Lists all members
- **list-tasks**: Lists all tasks
- **list-inv**: Lists all inventories
- **member-stats**: Displays member statistics
- **pdf [ty/TYPE]**: Creates a PDF document of inventory list
- **redo**: Redoes last known command
- **remove-member [mi/MEMBER\_ID]**: Removes a member
- **set-image[mi/MEMBER\_ID] [im/IMAGE\_FILEPATH]**: Switches to the settings configuration page
- **settings**: Switches to the settings configuration page
- **task-stats**: Displays task statistics
- **theme dark**: Switches to the dark theme
- **theme light**: Switches to the light theme
- **undo**: Undoes last known command

## 6. Upcoming Features

Our team is currently working hard on the following exciting features! Make sure to follow our application for any updates in the future.

## **6.1. Cloud Integration [coming in v2.0]**

Collaborate online with your teammates, and store your project details such as members calendars in the cloud.

## **6.2. Customisable User Profiles [coming in v2.0]**

Create more detailed profiles for members and choose who has authority to change details of the project.

## **6.3. AutoComplete Support for prefixes [coming in v2.0]**

AutoComplete to support users in prompting existing entities. For example, member ID and task tags to be prompted to users when prefix is input.

## **6.4. Instant Messaging [coming in v2.0]**

Communicate with teammates on one central application, instead of using multiple messaging applications.