# +Work - User Guide

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

1. Introduction

+Work is a project management tool that allows you to successfully achieve your objectives for your 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.

/data/projectdashboard.json
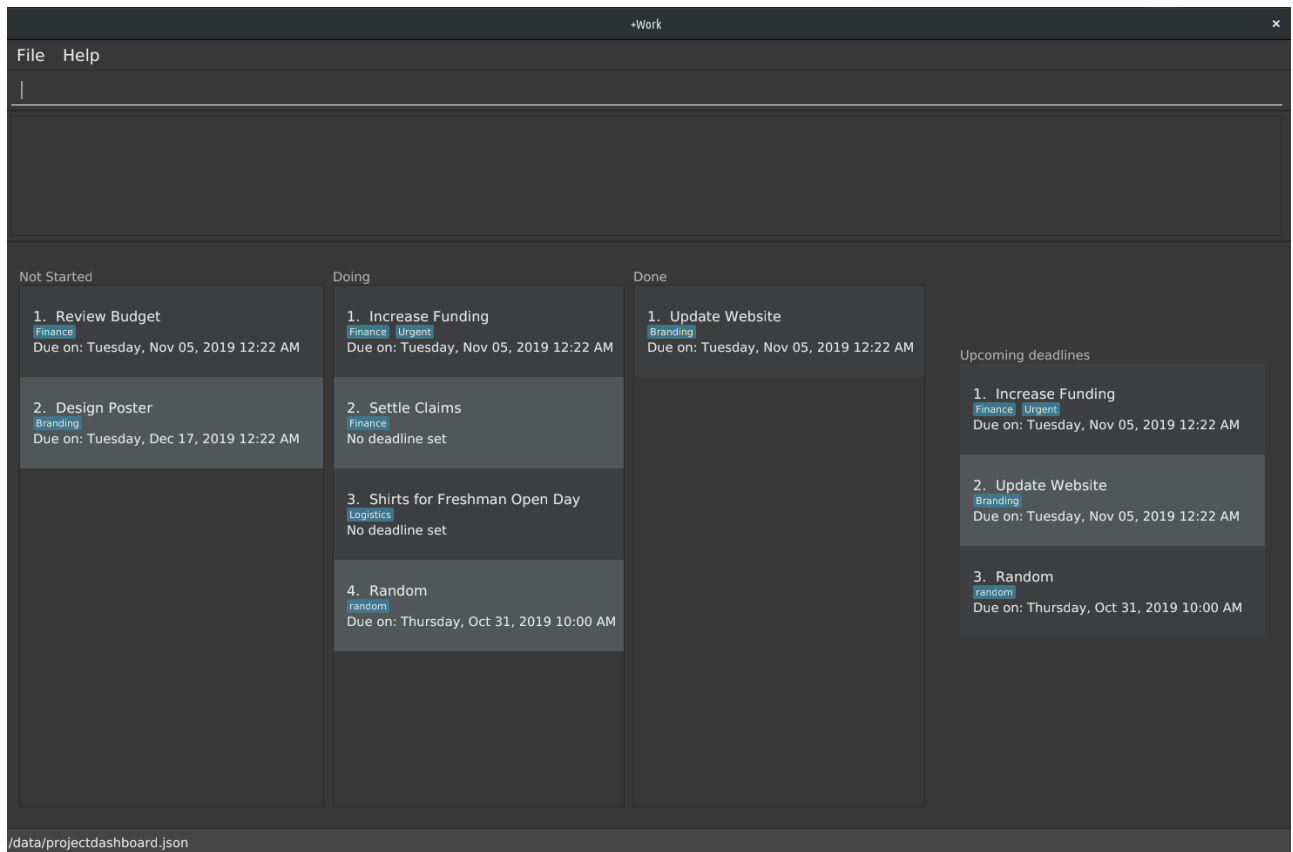
5. Type the command in the command box and press `Enter` to execute it.
   e.g. typing `help` and pressing `Enter` will open the help window.

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

Prefix	Description * Parameters can be given in any order
mn/	Member name
mi/	Member ID (set by you)
mt/	Member tag
im/	Image Path
tn/	Task name
s/	Task status
tt/	Task tag

- **Tags**
  - Tags are the only non-compulsary 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 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:

[Home] | *Home.png*

3.1.2. Accessing 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 settings page: `settings` (Coming in v1.3)

This command brings you to the settings page

Format: `settings`

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

[Settings] | *Settings.png*

> **NOTE**   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`

[Help] | *Help.png*

3.2. Member-related 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.

[BeforeAdd] | *BeforeAdd.png*

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.

[DuringAdd] | *DuringAdd.png*

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.

[DoneAdd] | *DoneAdd.png*

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

[ListAdd] | *ListAdd.png*

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

NOTE	• 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.

[BeforeSet] | *BeforeSet.png*

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.

[SetImage] | *SetImage.png*

NOTE	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, and will display a warning message, 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` is 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.

[BeforeEdit] | *BeforeEdit.png*

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'.

[AfterEdit] | *AfterEdit.png*

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:

[ListMembers] | *ListMembers.png*

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.

[BeforeRemove] | *BeforeRemove.png*

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.

[AfterRemove] | *AfterRemove.png*

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.
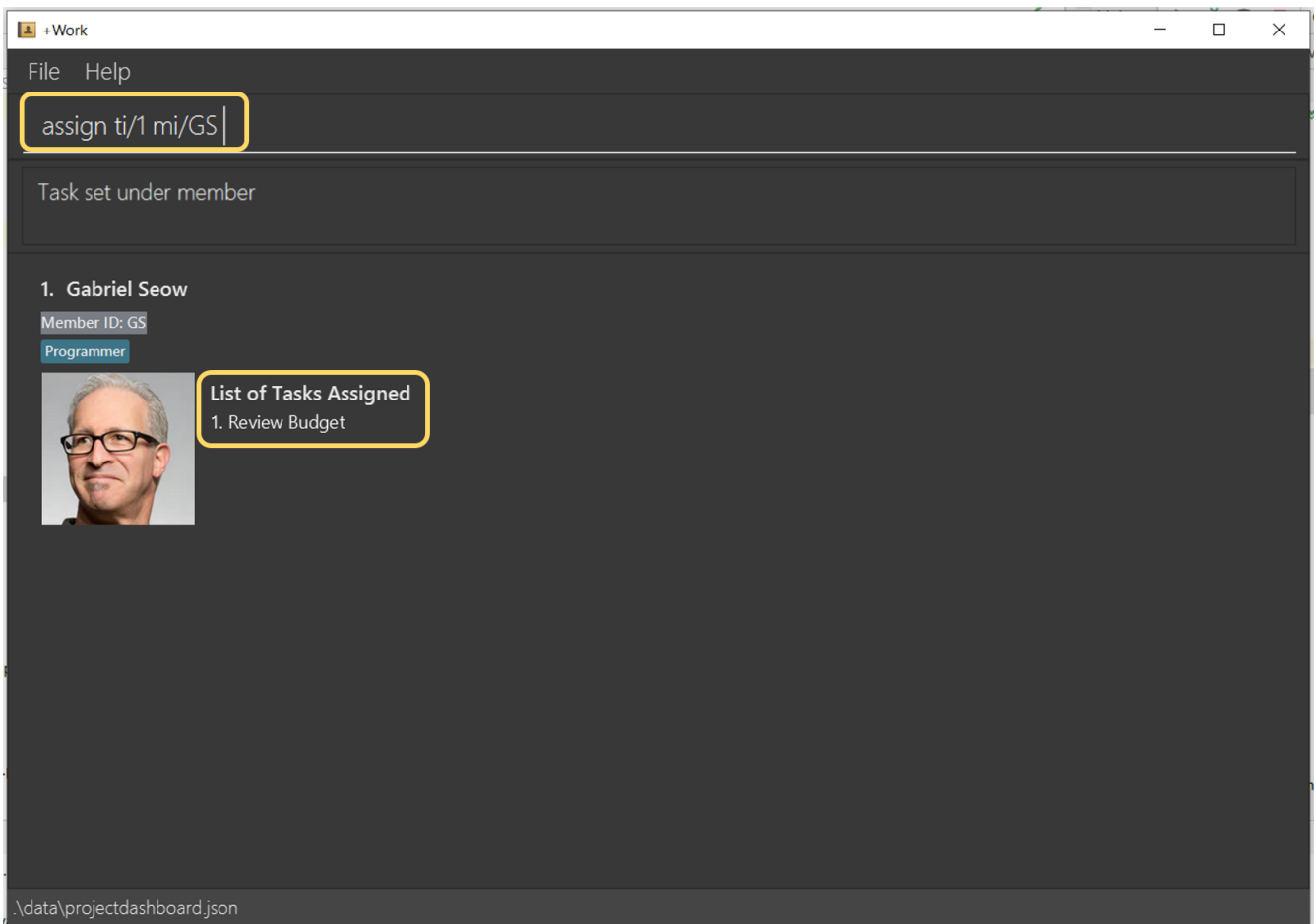
Format: `assign ti/TASK_ID mi/MEMBER_ID`

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`.

[BeforeAssign] | *BeforeAssign.png*

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`.

[BeforeFire] | *BeforeFire.png*

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.

[AfterFireMember] | *AfterFireMember.png*

3.3. Task-related Commands

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 [t/TASK_NAME]`
Optional Parameters: `[s/STATUS] [t/TAG]`

Example:

- `add-task t/Finish up milestone setting`
  A new task will be added to the project dashboard

3.3.2. Removing a task: remove-task

Removes a task from the dashboard

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

Example:

- `remove-task 2`
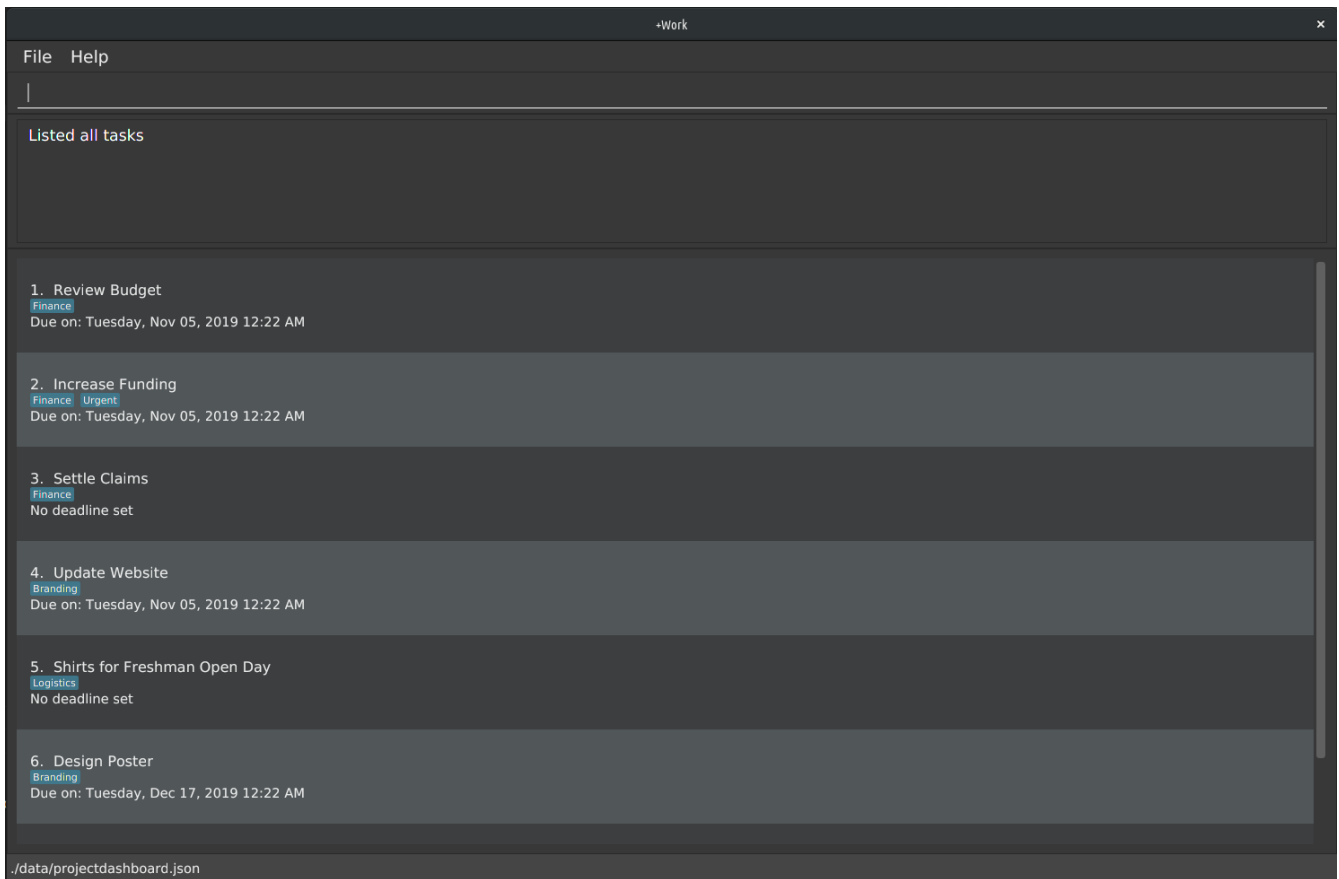 Removes the 2nd task in the dashboard

3.3.3. Listing all existing tasks: `list-tasks`

To list all the tasks created for the project, use the `list-tasks` command in the format below.
Format: 'list-tasks'

Example:

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

```
+Work                                                    ✕

 File   Help
 |
 Listed all tasks




  1.  Review Budget
  Finance
  Due on: Tuesday, Nov 05, 2019 12:22 AM

  2.  Increase Funding
  Finance   Urgent
  Due on: Tuesday, Nov 05, 2019 12:22 AM

  3.  Settle Claims
  Finance
  No deadline set

  4.  Update Website
  Branding
  Due on: Tuesday, Nov 05, 2019 12:22 AM

  5.  Shirts for Freshman Open Day
  Logistics
  No deadline set

  6.  Design Poster
  Branding
  Due on: Tuesday, Dec 17, 2019 12:22 AM


 ./data/projectdashboard.json
```

3.3.4. Setting a task's status to done: done-task

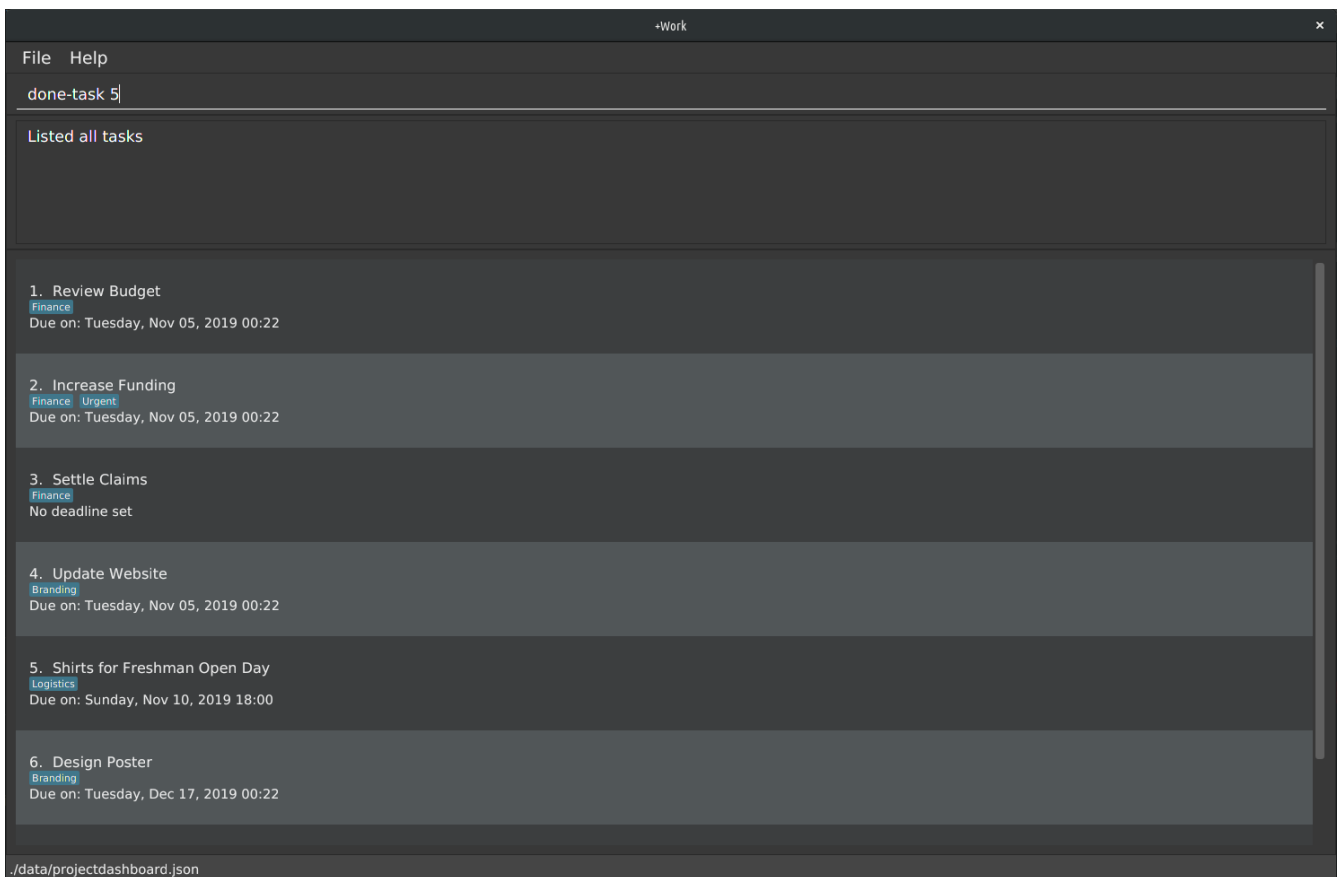To update the task status to done, use the done-task command in the format below.
Format: done-task [ti/TASK_ID]

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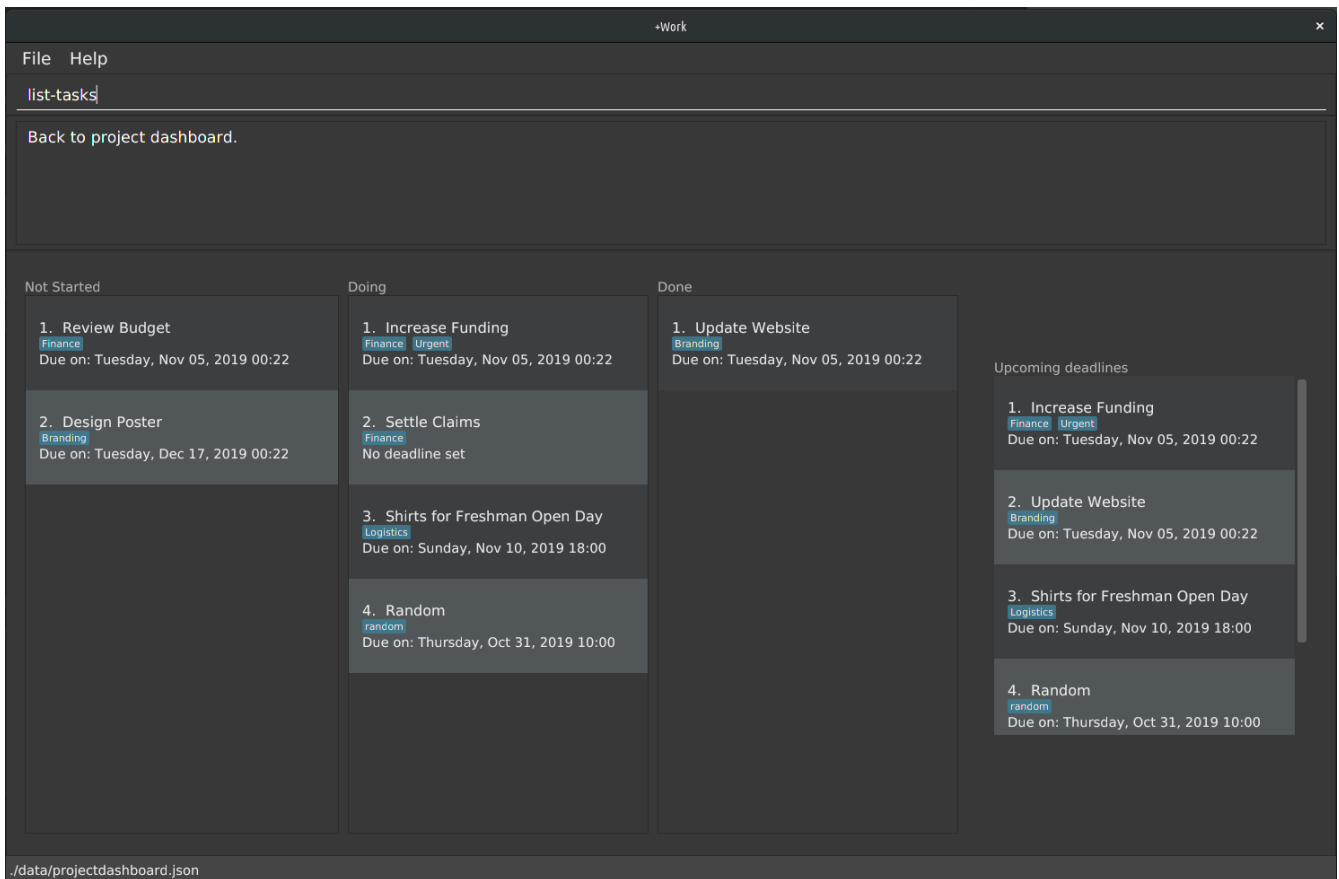
- Type `done-task ti/5` into the command box as shown below.

- Hit the ⌈Enter⌋ key and you will see that the task has been marked as done! You should see a window similar to the one below when you navigate back to home.

[done task result dashboard] | *done-task-result-dashboard.png*

3.3.5. 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/1`

- `doing-task ti/3`
  This sets the status of task 3 to `doing`.

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

3.3.6. Setting a deadline for a task: `set-deadline`
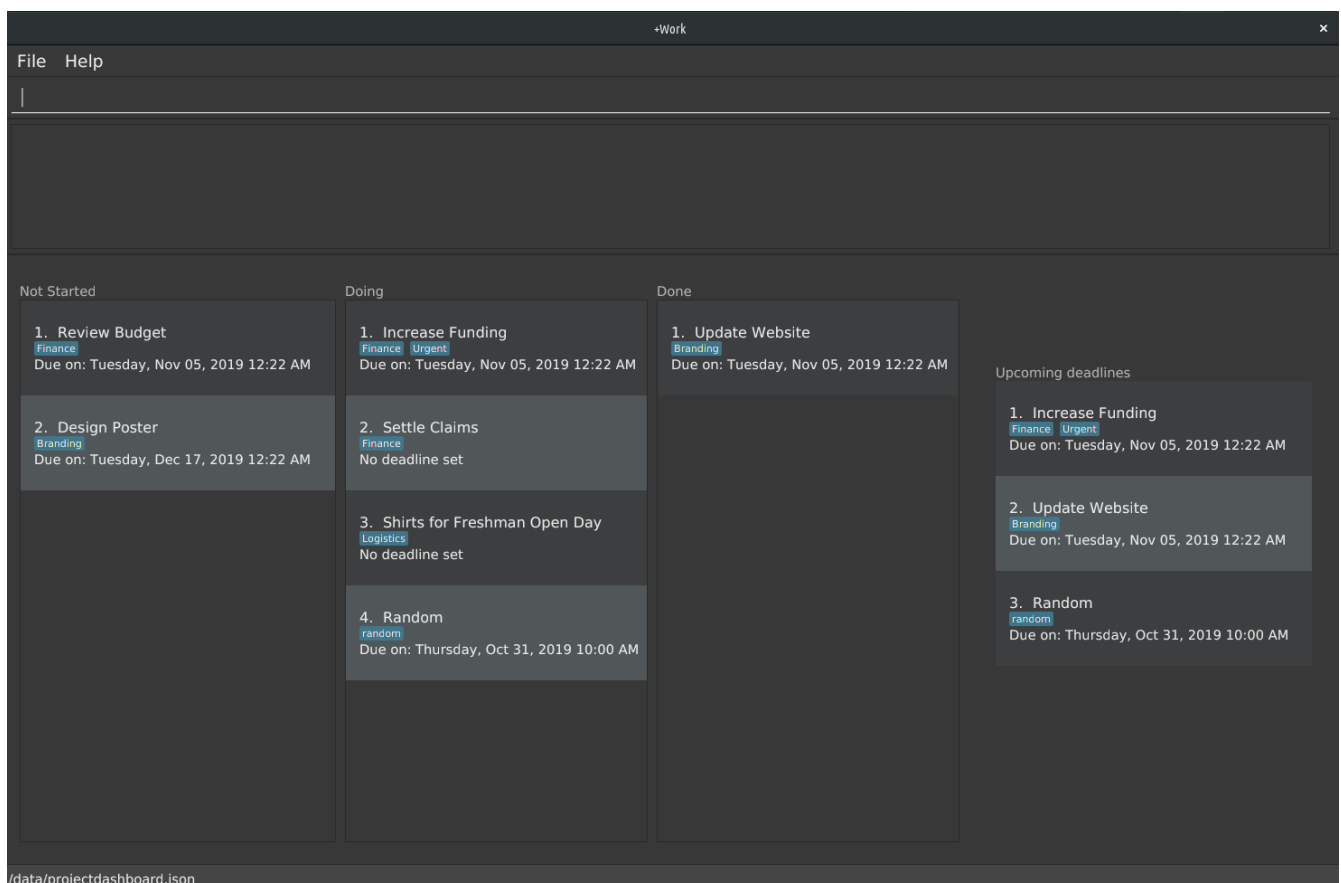
This sets a deadline for an existing task in your project.

Format: `set-deadline [ti/TASK_ID] [at/DEADLINE] [at/dd/mm/yy hh:mm]`

| TIP | Remember to input using the 24 hour time format of dd/mm/yy 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 is on the 10th of November 2019 at 6pm, type `deadline-task ti/5 at/10-11-19 18:00` into the command box.
- Hit the `Enter` key and you will see that a deadline has been set!.
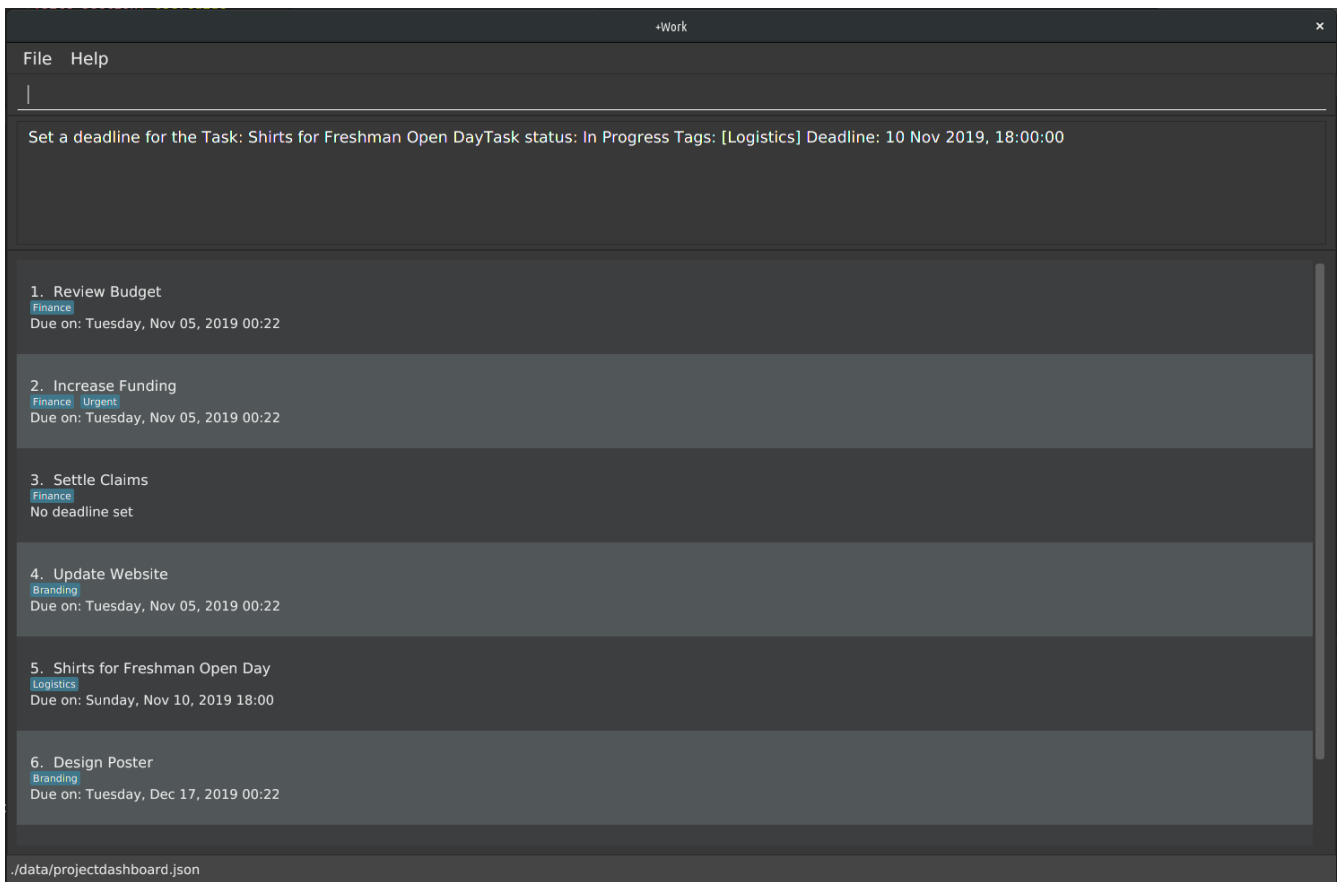
Now, navigate to the dashboard by entering `home`. Hit the `Enter` key and you will see that the deadline can be seen! You should see a window similar to the one below.

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

TIP	The "Upcoming deadlines" sidebar (right side of above picture) helps you keep track of tasks which are due in less than two weeks. end::task[]

3.4. Calendar Commands (Coming in v1.3)

> Suggested steps for you to follow
>
> 1. User first exports their `.ics` file from NUSmods
> 2. Then import it into their google calendar
> 3. Add any weekly commitments to their google calendar
> 4. Export `.ics` file again
> 5. Import all to our application
> 6. enter command `generate-timings`
> 7. Choose the desired timing for the weekly meetings from the grid displayed
> 8. Create a new `Meeting`

3.4.1. Adding a members calendar: `import-calendar`

Format: `import-calendar [mi/MEMBER_ID] [c/PATH_TO_ICS]`

Example:

You can add your project's member John Doe's `ics` file like this,

- `import-calendar mi/JD c/data/john_ics_file`
  Adds a calendar to John's profile

3.4.2. Generating time across all calendars uploaded: `generate-timings`

Format: `generate-timings`

Example:

- `generate-timings`
  Generates a grid showing the number of people who can make it for a particular time slot during the week (mock-up needed)

3.4.3. Adding a meeting: `add meeting`

Format: `add-meeting [at/ dd-mm-yyyy hh:mm] [l/LOCATION]`

to add a new meeting

Example:

- `add-meeting at/10-10-2018 19:00 l/COM2-0204`
  User chooses the meeting time from the grid displayed from generate timings commands (3.4.2)
  . A meeting is added to the internal calendar of the application, which will be displayed on the dashboard.

3.5. Inventory Commands

3.5.1. Adding an inventory: `add-inv`

Adds an inventory item bought or required for a specific project task.

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

Examples:

- `add-inv i/Mahjong Paper p/8.50 ti/2 mi/AR`
  Adds the item "Mahjong paper" for $8.50 to the inventory list. This item is tagged to task 2 and was paid for by member AR.

- `add-inv i/scissors ti/4 mi/GS`
  Adds the item "scissors" to the inventory list. The item is tagged to task 4 and is provided by member GS for no cost.

3.5.2. Deleting an inventory: `delete-inv`

Deletes an inventory item.

Format: `delete-inv [ITEM_ID]`

Examples:

- `delete-inv 3`
  Deletes the third item from the inventory list

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:

[MemberStats] | *MemberStats.png*

> **NOTE** 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:

[TaskStats] | *TaskStats.png*

NOTE	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.

3.7. Settings Commands

3.7.1. Switching the theme of +Work: `theme`

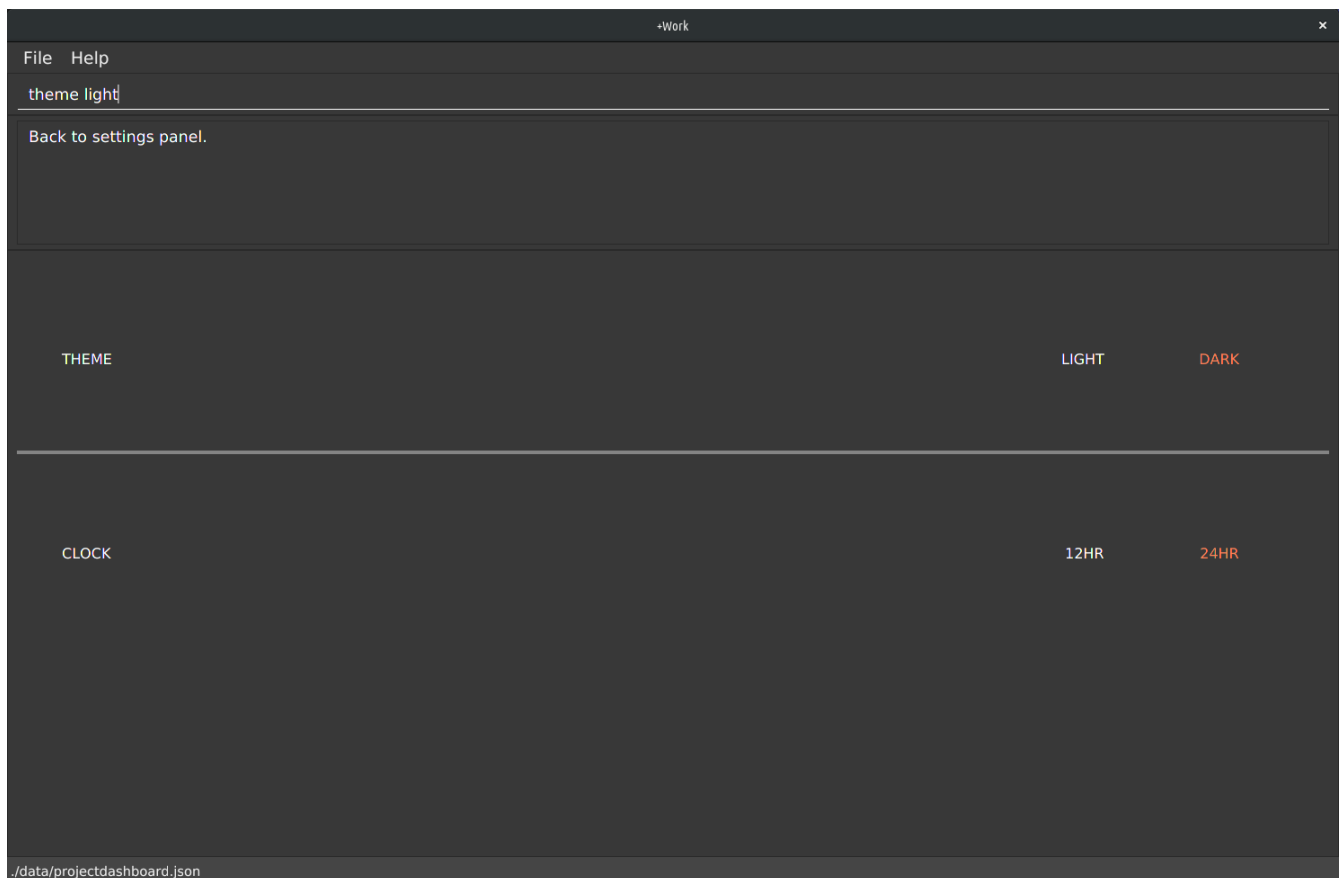This command toggles the theme of +Work between light and dark to suit your viewing preferences.

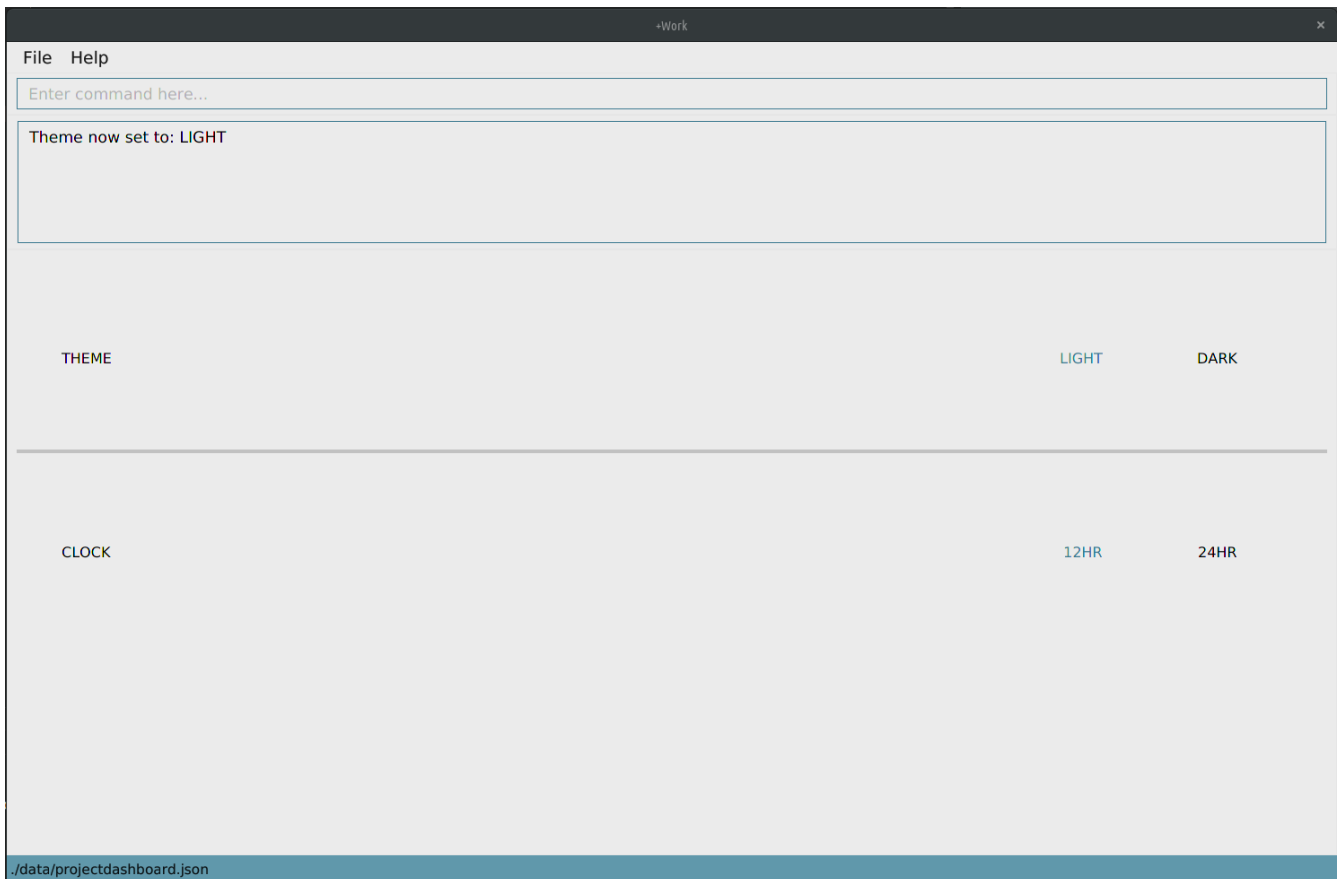|	
TIP	By default, the theme is set to dark.

Format: `theme light`

Examples:

- Type `theme light` as shown below.

- Hit the `Enter` key and +Work switches to the light theme!

- Similarly, `theme dark` switches to the dark theme

3.7.2. Switching the clock format of +Work: `clock`

This command toggles the clock format of +Work between 24 hour and 12 hour format. By default, the clock format is set to 24 hour.

NOTE	This does not affect the input format of deadlines for tasks, they still need to be entered in 24 hour format!.

Format: `clock twenty_four`

Examples:

- `clock twelve`
  Switches to the 12 hour clock

- `clock twenty_four`
  Switches to the 24 hour clock

end::settings[]

3.8. Saving the data

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

3.9. 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 and paste `add-member mn/ mi/ mt/` into your command-line automatically.

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 the calendar cannot find a timing where everyone is free?
A: There will be a grid showing the number of available members by time, so it will be easier for the user to choose the best possible time for the 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 users to the same task?
A: Since +Work is member-oriented, the application displays information according to members. Hence, you have to assign task to all the different members that are working on the task.

Q: What happens when a task has been completed?
A: You can mark the task as 'Done', which will move the task to the bottom of the task list. In the case where you do not want to keep the task in the list, using 'Remove Task' will delete it.

5. Command Summary

- `add-inventory [i/NAME] [p/PRICE] [ti/TASKID] [mi/MEMBER_ID]`: **Adds an inventory**

- `add-meeting [at/ dd/mm/yyyy hh:mm - hh:mm] [l/LOCATION]`: **Adds a meeting**

- `add-member [mn/MEMBER_NAME] [mi/MEMBER_ID] [mt/MEMBER_TAG]`: **Adds a member**

- `add-task [t/TASK_NAME] [m/MEMBER_NAME]`: **Adds a task**

- `assign [ti/TASK_ID] [mi/MEMBER_ID]`: **Adds a task to a member**

- `calendar`: **Switches to the time management page**

- `clock twelve`: **Switches deadline display to the 12 hour clock format**
- `clock twenty_four`: **Switches deadline display to the 24 hour clock format**
- `deadline-task [ti/TASK_ID] [at/dd/mm/yy hh:mm]`: **Assigns a deadline to a task**
- `delete-inventory [ii/ITEM_ID]`: **Deletes an inventory**
- `doing-task [ti/TASK_ID]`: **Marks a task as** `doing`
- `done-task [ti/TASK_ID]`: **Marks a task as** `done`
- `fire-member [ti/TASK_ID] [mi/MEMBER_ID]`: **Removes a task from a member**
- `generate-inventory /task`: **Generates report of inventory by task**
- `generate-inventory /person`: **Generates report of inventory by person**
- `generate-timings`: **Generates free times among members**
- `home`: **Switches to the project dashboard**
- `help`: **Accesses the help window**
- `import-calendar [mi/MEMBER_ID] [c/PATH_TO_ICS]`: (Coming in v1.3) **Imports a calendar**
- `list-members`: **Lists all members**
- `list-tasks`: **Lists all tasks**
- `remove-member [mi/MEMBER_ID]`: **Removes a member**
- `remove-task [ti/TASK_ID]`: **Removes a task**
- `settings`: **Switches to the settings configuration page**
- `theme dark`: **Switches to the dark theme**
- `theme light`: **Switches to the light theme**