

# FitHelper - User Guide

1. Introduction .....	2
2. Quick Start .....	2
3. Common Fields and Keywords .....	3
4. Features .....	4
4.1. Viewing help : <b>help</b> .....	6
4.2. Keep Food and Sports Entries .....	6
4.2.1. Go to dashboard page: <b>home</b> .....	6
4.2.2. Adding an entry: <b>add</b> .....	7
4.2.3. Listing all entries : <b>list</b> .....	9
4.2.4. View Reminder : <b>reminder</b> .....	9
4.2.5. Editing an entry : <b>edit</b> .....	9
4.2.6. Locating entries by name: <b>find</b> .....	10
4.2.7. Deleting an entry : <b>delete</b> .....	11
4.2.8. Sort food/sports entry list: <b>sort</b> .....	12
4.2.9. Check calorie intake/consumption of common food/sports: <b>check</b> .....	13
4.3. Keep Diaries .....	14
4.3.1. Add a new diary: <b>addDiary</b> .....	15
4.3.2. Append new content to an existing diary: <b>appendDiary</b> .....	15
4.3.3. Edit a diary: <b>editDiary</b> .....	15
4.3.4. Find a diary: <b>findDiary</b> .....	15
4.3.5. Delete a diary: <b>deleteDiary</b> .....	16
4.3.6. Clear the diary: <b>clearDairy</b> .....	16
4.4. Clear All Entries And Diaries: <b>clear</b> .....	16
4.5. Undo : <b>undo</b> .....	17
4.6. Redo : <b>redo</b> .....	17
4.7. Calendar View .....	17
4.7.1. Calendar list mode: <b>calendar m/l</b> <b>s</b> [d/DATE] .....	17
4.7.2. Calendar timetable mode: <b>calendar m/tb</b> [d/DATE] .....	18
4.7.3. Calendar display entries from a day: <b>calendar sh</b> /DATE .....	19
4.7.4. Set calendar referenced date: <b>calendar d</b> /DATE .....	19
4.8. Daily Summary .....	19
4.8.1. Today Page: <b>today</b> .....	19
4.8.2. Some Day Page [ <b>coming in v2.0</b> ] .....	20
4.9. Weekly Summary .....	20
4.9.1. Weekly Report [ <b>coming in v2.0</b> ] .....	20
4.9.2. Some Week Report [ <b>coming in v2.0</b> ] .....	21
4.10. Keep User Profile .....	21
4.10.1. Profile Page: <b>profile</b> .....	21

4.10.2. Update Profile Data : <code>update</code> .....	21
4.11. Keep Weight Records .....	22
4.11.1. Weight Page: <code>weight</code> .....	22
4.11.2. Add Weight Records : <code>addWeight</code> .....	22
4.11.3. Edit Weight Records : <code>editWeight</code> .....	23
4.11.4. Delete Weight Records : <code>deleteWeight</code> .....	23
4.11.5. Clear Weight Records : <code>clearWeight</code> .....	24
4.12. Exit the program : <code>exit/bye/quit</code> .....	24
4.13. Save the data .....	24
5. Command Summary .....	25

By: Team AY1920S2-CS2103-T09-4 Since: Feb 2020 Licence: MIT

# 1. Introduction

FitHelper (FH) is for those who **prefer to use a desktop app for managing daily diet and exercises**. More importantly, FH is **optimized for those who prefer to work with a Command Line Interface (CLI)** while still having the benefits of a Graphical User Interface (GUI). If you can type fast, FH can get your weight management tasks done faster than traditional GUI apps. Interested? Jump to the [Section 2, “Quick Start”](#) to get started. Enjoy!

FitHelper mainly serves as a fitness log book where users can **mark down their diet and exercises entries** and **be reminded about their plans**. They can also **visualize their performance** to be encouraged.

# 2. Quick Start

1. Ensure you have Java `11` or above installed in your Computer.
2. Download the latest `FitHelper.jar` [here](#).
3. Copy the file to the folder you want to use as the home folder for your Fitness Helper. Please ensure that writing or reading data inside the home folder that you choose does **not** require administrator permission.
4. Double-click the file to start the app. The GUI should appear in a few seconds.

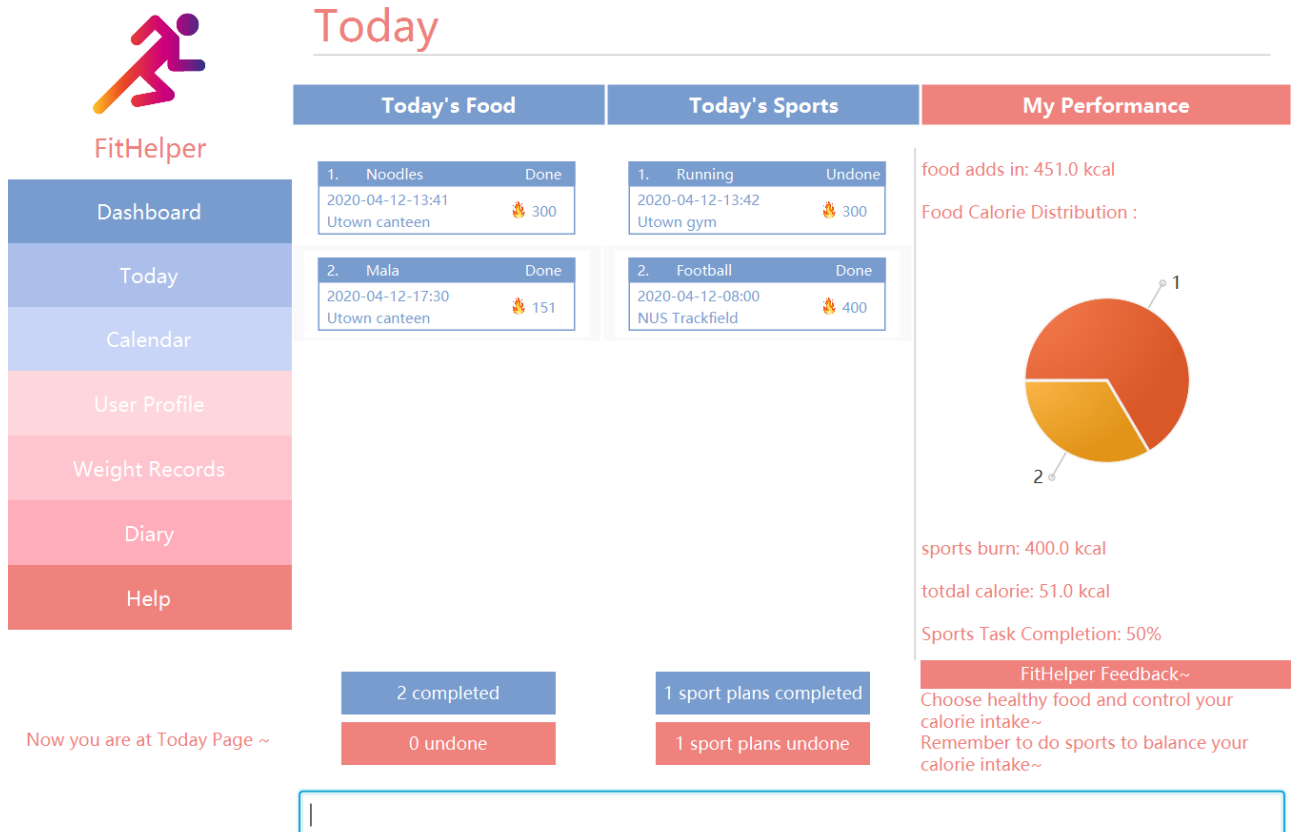


Figure 1. A snapshot of the app

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. Some example commands you can try:
  - `list` : lists all logs
  - `addx/s n/swimming t/Wednesday 18:00 l/utown gym c/500` : adds an entry named `swimming` to the dashboard page.
  - `deletei/3 x/f` : deletes the 3rd entry from food entry list shown in the dashboard
  - `exit` : exits the app

[WARNING] FitHelper does not support expanding the main window. . Refer to [Section 4, “Features”](#) for details of each command.

### 3. Common Fields and Keywords

- `x/` entry type
- `n/` entry name
- `t/` entry time
- `l/` entry location
- `c/` entry calorie
- `s/` entry status

- **r**/ entry remark
- **i**/ entry index
- **d**/ date in format yyyy-MM-dd
- **m**/ mode
- **sh**/ date in format of yyyy-MM-dd in calendar
- **dc**/ diary content
- **dr**/ duration
- **attr**/ profile attribute name
- **v**/ profile attribute value/ weight value
- **by**/ sorting criterion
- **o**/ sorting order
- **k**/ keywords for searching/checking
- **-f** force change flag

## 4. Features

## General rules about command format

- A prefix followed by a word in **UPPER\_CASE** form a parameter to be supplied by the user.

Example: in `add n/NAME`, `n/NAME` is a parameter which can be `n/running`.

- Parameters in square brackets are optional e.g. `n/NAME [r/REMARK]` means both `n/swimming` `r/fun` and `n/swimming` are both valid user inputs.
- Parameters not in square brackets are compulsory e.g. `x/TYPE k/KEYWORDS` means you must input both `TYPE` and `KEYWORDS` parameters.
- Items with `...` after them can be used multiple times including zero times e.g. `[r/remark]...` can be used as (i.e. 0 times), `r/really fun`, `r/really fun r/helps me lose weight` etc.
- Parameters can be in any order e.g. if a command format is `add n/NAME t/TIME`, then `add n/Bob t/2020-04-10-08:00` and `add t/2020-04-10-18:00 n/Bob` are both acceptable.
- The field `TYPE` in this document refers to type of entries. The only valid values are food OR sports, or in acronym f OR s.
- The field `TIME` needs to be in the format of `yyyy-mm-dd-hh:MM`, e.g. `2020-04-10-03:00`
- The field `DATE` should be entered in the format of `yyyy-mm-dd`, e.g. `2020-02-02`.
- Our time parser will auto-correct some invalid date/time entered. More specifically, as long as the month value is valid and the day value is between 1 and 31, the parser will auto-correct an invalid date to the last valid date of the year-month specified by the user.

Example: `2020-02-31-12:34` will be auto-corrected as `2020-02-29-12:34` since `2020-02-29` is the last valid date of Feb 2020. `2020-22-31` and `2020-02-40` will not be auto-corrected since the month value is invalid or the day value is beyond 31.

- The field `INDEX` should be a positive integer representing a one-based index of an entry in an list.
- The field `DURATION` should be a positive double number, representing duration in hours. e.g. `dr/1.5` for 1.5 hours.
- You may input multiple parameters of a same prefix, but only the last input value will be read.

Example: `add n/Alice n/Bob n/Charlie` is equivalent to `add c/Charlie`.

- Please do not input parameters not required by a particular command.

Example: the format of sort command is `sort [x/TYPE] by/SORT_BY [o/ORDER]`, so you should not input other parameters like `n/NAME`, `k/KEYWORDS` or `v/VALUE`.

- The flag `-f` should be added **just after the command word**, e.g. `update -f`.

## 4.1. Viewing help : **help**

Asks the application to go to Help page, which displays a condensed list of available functions with correct format of input. A link to the User Guide is also provided after the list. The user can also go to the Help page by clicking the **Help** button on the left hand side of the application. Note that once the user types **help** or click the **Help** button, the url of the User Guide is copied to the clipboard, so the user can directly access the Guide by pasting the url into the browser.

Format: **help**

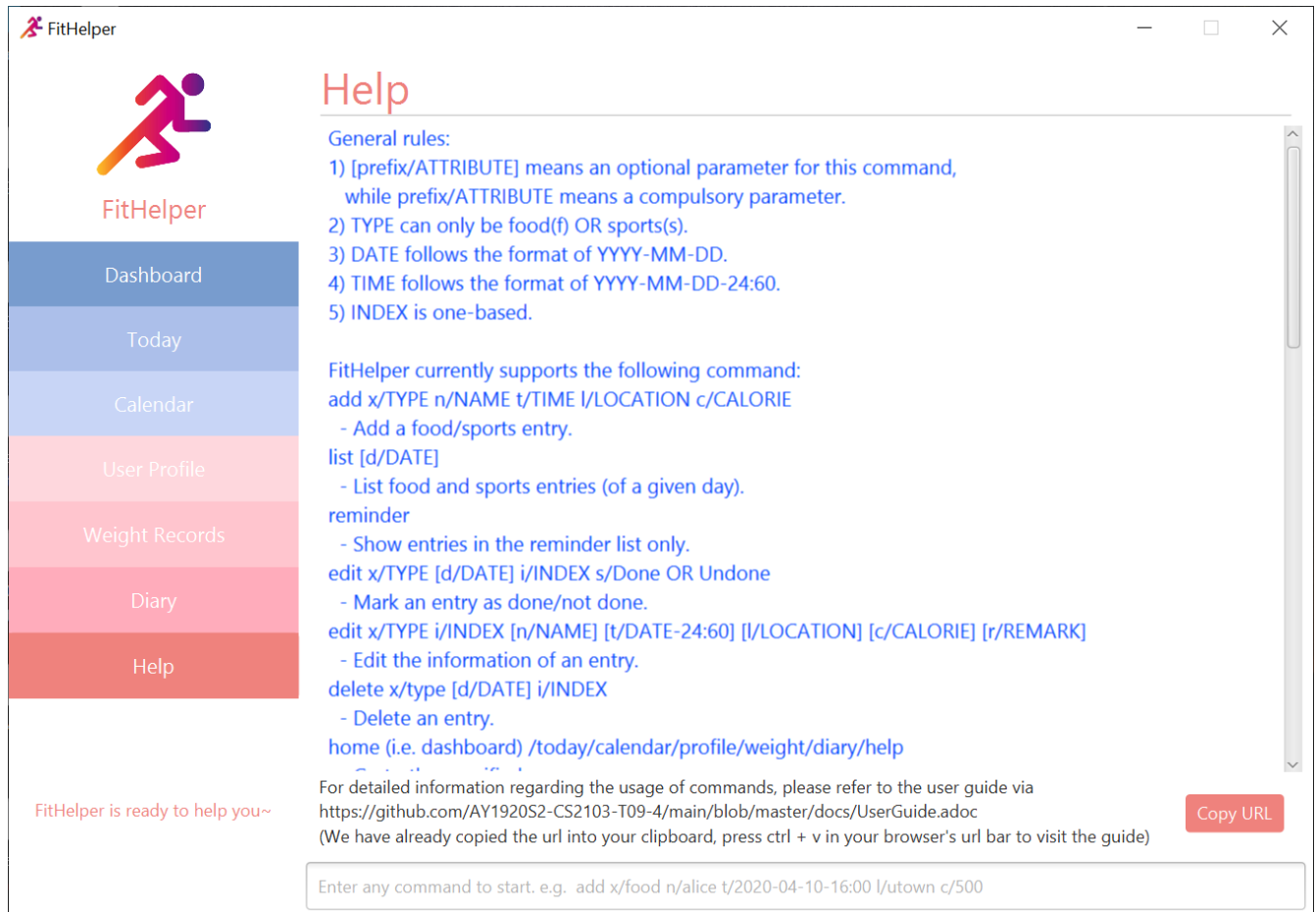


Figure 2. Help page

## 4.2. Keep Food and Sports Entries

### 4.2.1. Go to dashboard page: **home**

Go to the Dashboard page, which displays three entry lists: Food Entry History, Sports Entry History and Reminders.

- **Food Entry History**  
Shows food entries.
- **Sports Entry History**  
Shows sports entries.
- **Reminders**  
Shows entries that are remarked as not done.

Note that you can also go to the Dashboard page by clicking the **DashBoard** button on the left side of the application. However, using **home** command will further **reset the display status** of the Dashboard page to the default setting (showing all entries in all three lists).

Example:

If you input **reminder** command, the DashBoard page will now only show entries in the reminders list (display status changed). If you switch to other page and then try to go back to the Dashboard page by clicking the **DashBoard** button, the Dashboard page will still only show the reminders list. But if you input **home** command, entries in all three lists will be shown.

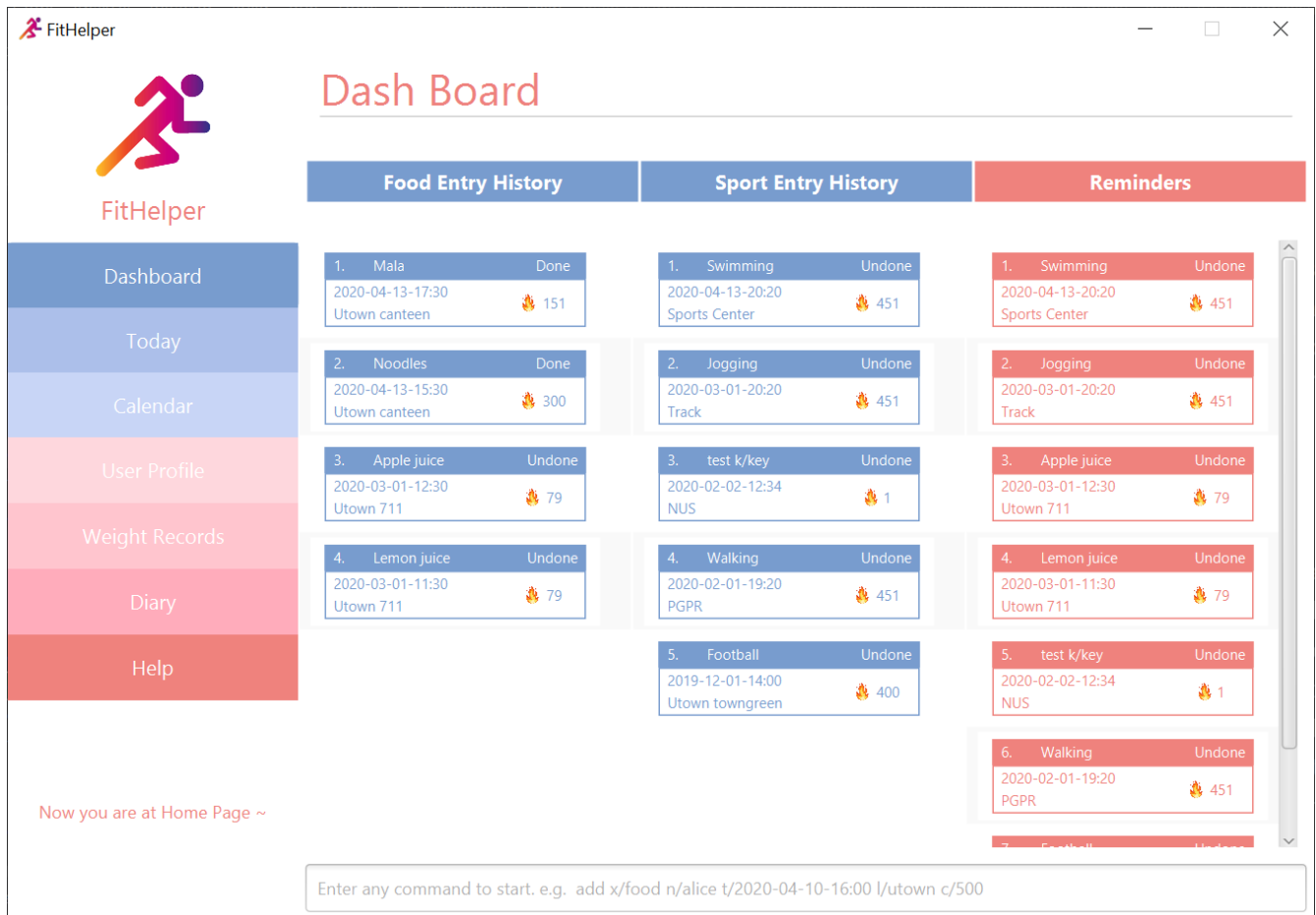


Figure 3. Dashboard page in default display setting

#### 4.2.2. Adding an entry: **add**

Adds an entry to the fitness log book. The entry added will be displayed in DashBoard, Calendar and possibly Today Page (if its date is today) immediately after the command is executed.

Note that due to space constraint, entry card in DashBoard and Today page will only display first 14 characters of its name and first 21 characters of its location. The entry icon in Calendar Page are even more constrained.

- FitHelper accepts duplicate fields, but only the last value specified under this field will be taken.\*
- Default value for unspecified optional field: **status** : Undone; **duration**: 1 hr\*

Format: **add** x/TYPE n/NAME t/DATETIME l/LOCATION c/CALORIE [dr/DURATION]...

Examples:

- add x/s n/running t/2020-04-10-18:00 l/utown gym c/500
- add x/f n/chicken rice t/2020-04-10-11:00 l/Super Snacks c/200

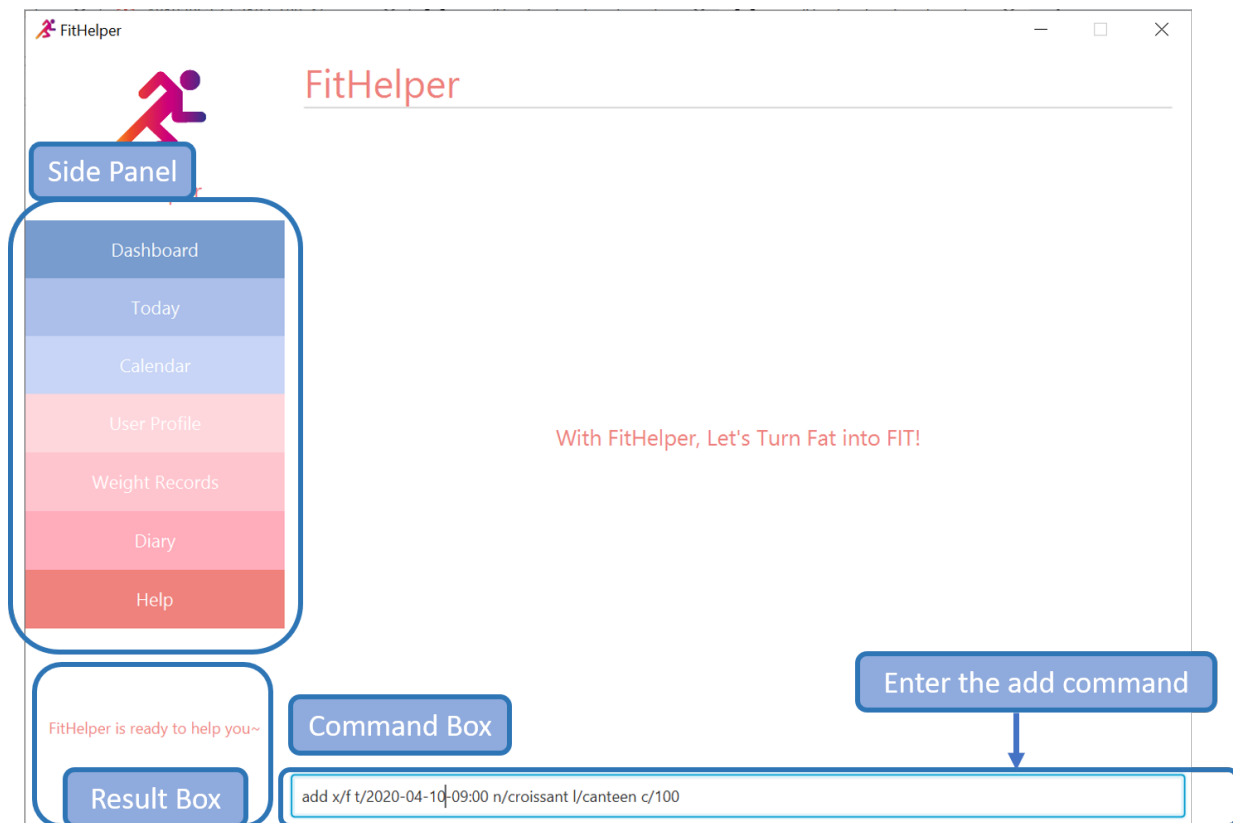


Figure 4. Add Command



Figure 5. Add Command Result



### 4.2.3. Listing all entries : `list`

Shows a list of all entries in the fitness log book.

Format: `list`

### 4.2.4. View Reminder : `reminder`

Shows a list of all undone tasks on the specified date on the home page (Dashboard).

Format: `reminder [d/DATE]`

If no `DATE` is specified, FitHelper displays all undone entries.

Examples:

- `reminder d/2020-03-31`

Switches to Home Page and shows all undone entries on `2020-03-31` in the reminder field.

### 4.2.5. Editing an entry : `edit`

Edits an existing entry in the fitness log book with the specified values.

Format: `edit x/TYPE i/INDEX [s/STATUS] [n/NAME] [t/TIME] [l/LOCATION] [c/CALORIE][dr/DURATION]...`

- At least one of the optional fields must be provided.
- If the user is currently in other pages, a successful execution of `edit` will switch the page to the dashboard.
- To obtain accurate indices, the user should first switch to the home page by calling `home` or `list` and refer to the displayed list in `food entry history` and `sport entry history` fields.
- Additionally, the user can first called `find` command and then refer to the resulting list's indices to `edit` an entry. However, right after a `find`, the user can only refer to entries that are currently displayed.  
i.e. If there are 5 food entries in total, and after `find` 2 food entries are displayed. If the user does not switch back to the home page, he can only refer to the displayed 2 entries (with indices 1 and 2). He can `edit` any entries only when he turns to home page after the `find`.
- The index **must be a positive integer** 1, 2, 3, ...

Examples: (after switching to Home Page) \* `edit x/s i/1 t/2020-04-05-16:00 l/PGP gym`

Edits the time and email location of the 1st entry to be `Friday 4pm` and `PGP gym` respectively.

### Mark an entry as done

Users can mark an entry as done, either a meal or sports, where the calories intake and consumption will be taken in to consideration. Format: `edit x/TYPE i/INDEX s/Done`

## Mark an entry as undone

Similar to the previous command, marking an entry as undone edits the **s/** field and modify it as **Undone**. Format: **edit x/TYPE i/INDEX s/Undone**

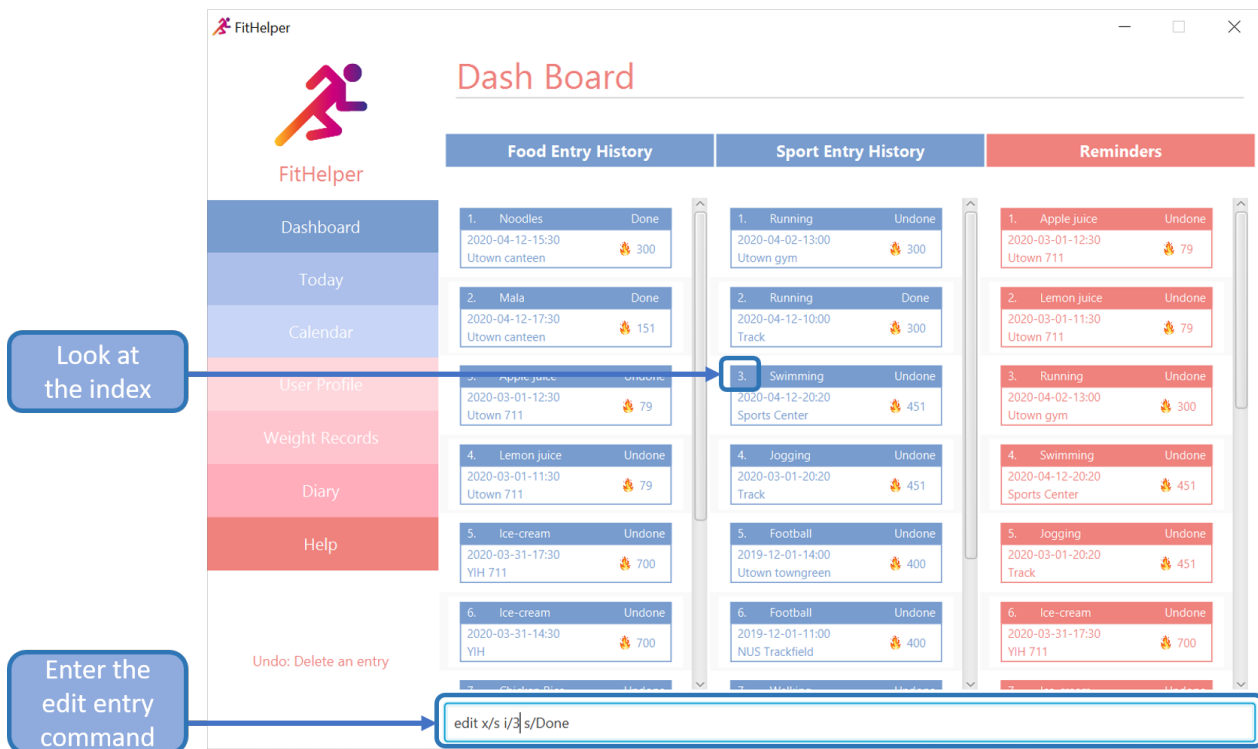


Figure 6. Edit Command

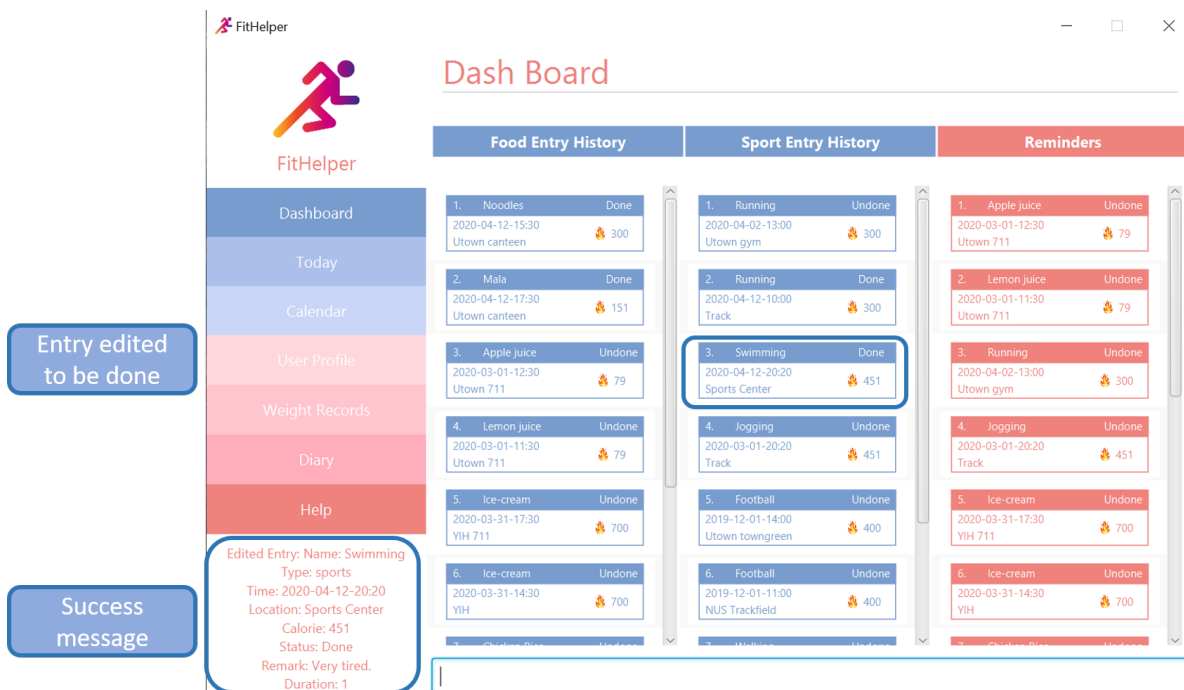


Figure 7. Edit Command Result

## 4.2.6. Locating entries by name: **find**

Finds entries whose names contain any of the given keywords.

Format: **find [x/TYPE] k/ONE\_OR\_MORE\_KEYWORDS** If the **TYPE** field is empty, FitHelper will search from

the whole list.

- The search is case insensitive. e.g **apples** will match **Apples**.
- The order of the keywords does not matter. e.g. **Apple Pie** will match **Pie Apple**.
- Only the name is searched.
- Only full words will be matched e.g. **Straw** will not match **Strawberries**.
- Entries matching at least one keyword will be returned (i.e. **OR** search). e.g. **Apple Banana** will return **Apple Pie**, **Banana Milkshake**.

Examples:

- **find x/s k/running**  
Returns **running** and **slow running**
- **find x/food k/Juice Apple**  
Returns any entry having names **Juice**, or **Apple**

#### 4.2.7. Deleting an entry : **delete**

Deletes the entry at the specified **INDEX** from the fitness log book.

Format: **delete x/TYPE i/INDEX**

- To obtain accurate indices, the user should first switch to the home page by calling **home** or **list** and refer to the displayed list in **food entry history** and **sport entry history** fields.
- Additionally, the user can first called a **find** and then refer to the resulting list's indices to **delete** an entry. However, right after a **find**, the user can only refer to entries that are currently displayed.  
i.e. If there are 5 food entries in total, and after **find** 2 food entries are displayed. If the user does not switch back to the home page, he can only refer to the displayed 2 entries (with indices 1 and 2).
- The index **must be a positive integer** 1, 2, 3, ...

Examples:

- **delete x/sports index/2** Deletes the 2nd sports entry in the sports list from dashboard.

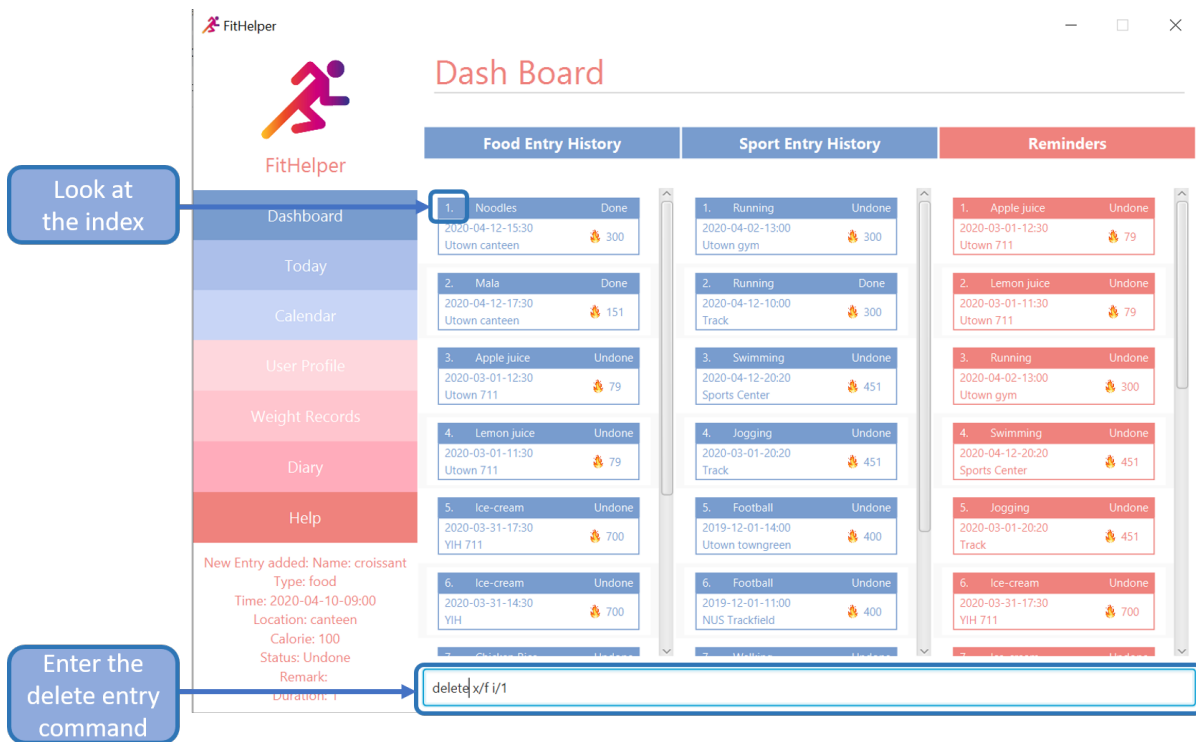


Figure 8. Delete Command

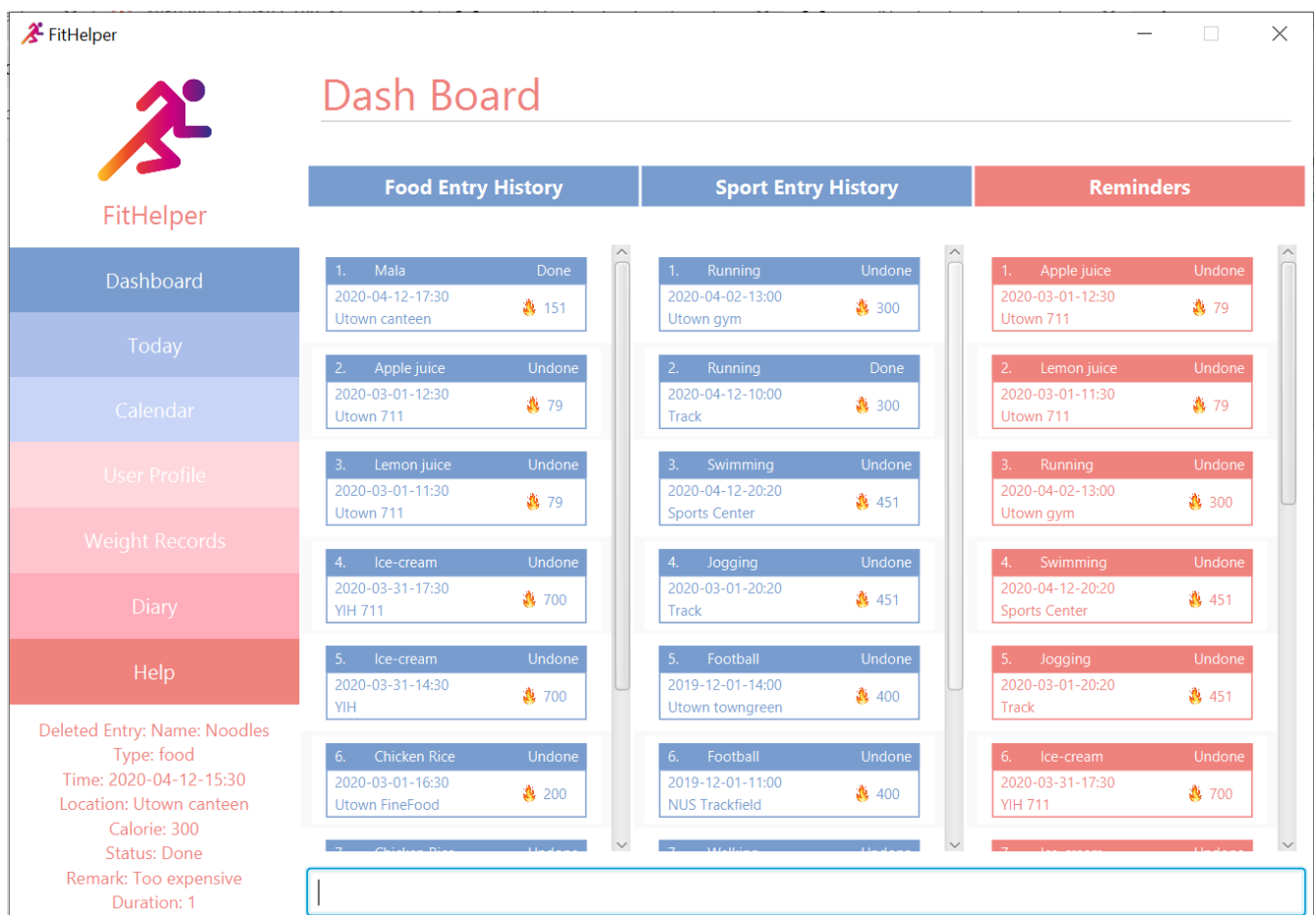


Figure 9. Delete Command Result

#### 4.2.8. Sort food/sports entry list: sort

Sorts food or sports entry list, or both of them, based on starting time or calorie value or name of the entry (case insensitive), in either ascending or descending order. In addition, the reminders

entry list will also be sorted in the same manner specified by the user. The user must specify the sorting criterion **cal/c** OR **time/t** OR **name/n**. If the user does not specify the type of entry list, both lists will be sorted. And if the user does not specify the sorting order **a** OR **d**, the default order is descending (i.e. entry that starts later or entry with higher calorie value or entry whose name starts with later alphabet comes first).

format: **sort** [x/TYPE] **by**/SORT\_BY [o/ORDER]

Examples:

- **sort** x/f **by**/time o/a (sort food entry list and reminders list in ascending order of recording time, i.e. the older entry comes first)
- **sort** **by**/c (sort both food and sports entry list, as well as reminders list, in descending order of calorie value, i.e. entry with higher calorie intake/consumption comes first)

**FitHelper**

## Dash Board

Food Entry History	Sport Entry History	Reminders
1. Mala Done 2020-04-13-17:30 Utown canteen 151	1. Swimming Undone 2020-04-13-20:20 Sports Center 451	1. Swimming Undone 2020-04-13-20:20 Sports Center 451
2. Noodles Done 2020-04-13-15:30 Utown canteen 300	2. Jogging Undone 2020-03-01-20:20 Track 451	2. Jogging Undone 2020-03-01-20:20 Track 451
3. Apple juice Undone 2020-03-01-12:30 Utown 711 79	3. test k/key Undone 2020-02-02-12:34 NUS 1	3. Apple juice Undone 2020-03-01-12:30 Utown 711 79
4. Lemon juice Undone 2020-03-01-11:30 Utown 711 79	4. Walking Undone 2020-02-01-19:20 PGPR 451	4. Lemon juice Undone 2020-03-01-11:30 Utown 711 79
	5. Football Undone 2019-12-01-14:00 Utown towngreen 400	5. test k/key Undone 2020-02-02-12:34 NUS 1
		6. Walking Undone 2020-02-01-19:20 PGPR 451

Both food and sports entry lists, as well as reminder list are sorted by time in descending order.

Figure 10. Dashboard page after all 3 lists are sorted by time in descending order (later entry comes at top)

#### 4.2.9. Check calorie intake/consumption of common food/sports: **check**

Searches through the pre-set database for calorie intake/consumption information about common food/sports (i.e. a datum) whose name matches/contains keywords specified by the user. Note that due to space constraint, at most 3 matching data will be shown at the left-bottom corner of the application. Each datum shows the name of the food/sports and its calorie intake/consumption information: for food, it will be in cal per serving (weight per serving in gram will also be shown); for sports, it will be in cal per hour for a 70kg person. Please note after the command is executed, the application will automatically go to Dashboard page **without resetting the display status**.

The searching works as follows:

- Retrieves food/sports data based on check type specified by the user.
- Checks and adds food/sports datum whose name **equals one of the keywords**. If fewer than 3 food/sports data are added so far, checks and adds food/sports datum whose name **contains the keywords as a whole**. If still fewer than 3 food/sports data are added so far, checks and adds food/sports datum whose name **contains one of the keywords**.
- If no datum is added after all the steps, replies user that search failed. Otherwise, shows user the data.

format: `check x/TYPE k/ONE_OR_MORE_KEYWORDS`

Examples:

- `check x/food k/apple` (searches for calorie intake of food whose name match/contain keyword "apple")

The screenshot shows the FitHelper Dash Board interface. On the left is a sidebar with navigation links: Dashboard, Today, Calendar, User Profile, Weight Records, Diary, and Help. The main area is titled 'Dash Board' and has three tabs: Food Entry History, Sport Entry History, and Reminders. Below the tabs, there's a section titled 'results of checking' which displays matching reference data for 'apple':

- Matching reference data:
- Name: Apple
- Calorie: 95 cal per 1 apple (182 g)
- Name: Applesauce
- Calorie: 167 cal per 1 cup (246 g)
- Note: at most 3 records are shown.

Below this, there's a search bar. To the right, under the 'Reminders' tab, a list of activities is shown with their dates, times, and calorie counts:

Activity	Date	Time	Calories
1. Swimming	2020-04-13	20:20	451
2. Jogging	2020-03-01	20:20	451
3. Apple juice	2020-03-01	12:30	79
4. Lemon juice	2020-03-01	11:30	79
5. test k/key	2020-02-02	12:34	1
6. Walking	2020-02-01	19:20	451
7. Football			

Figure 11. Check command result (matching data found)

## 4.3. Keep Diaries

The user can keep a diary in FitHelper, and command for adding/editing/deleting/searching/clearing diary logs.

- `diary` switches to the **Diary Page** where all diary logs are displayed in chronological order

#### 4.3.1. Add a new diary: `addDiary`

- `addDiary d/DATE dc/DIARYCONTENT`  
adds a new diary on the specified `DATE` with the specified `CONTENT`

If there is previously added diary on the specified `DATE`, calling `addDiary` will rewrite the content.

Examples:

- `addDiary d/2020-03-31 dc/Happy birthday Alice!` Adds a diary log on the date `2020-03-31` with the content `Happy birthday Alice!`.

#### 4.3.2. Append new content to an existing diary: `appendDiary`

- `appendDiary d/DATE dc/APPEDED CONTENT`  
appends new content to the existing diary on the specified `DATE`.

If the specified `DATE` does not have previously added diary log, this command will be discarded with a `DIARY_NOT_FOUND` reminder.

Examples:

- `appendDiary d/2020-03-31 dc/You are one year older now:)` (after `addDiary d/2020-03-31 dc/Happy birthday Alice!`)  
This command appends the new diary content to the existing the diary log on the date `2020-03-31`. The new diary content after this command will be "Happy birthday Alice! You are one year older now:)".

#### 4.3.3. Edit a diary: `editDiary`

- `editDiary d/DATE dc/NEW DIARYCONTENT`  
edits the diary on the specified `DATE` with the new `DIARYCONTENT`.

If the specified `DATE` does not have previously added diary log, this command will be discarded with a `DIARY_NOT_FOUND` reminder. If the existing diary with the specified `DATE` has the same content as the specified content, this command will be discarded with a `DUPLICATE_DIARY` reminder.

Examples:

- `editDiary d/2020-03-31 dc/Happy birthday Alice!`  
This command edits the content of the diary log on the date `2020-03-31` with the content `Happy birthday Alice!`.

#### 4.3.4. Find a diary: `findDiary`

- `findDiary [d/DATE] [k/ONE OR MORE KEYWORDS]`  
finds diaries either on the specified `DATE` or contains the specified `KEYWORDS`.  
Both `DATE` and `KEYWORDS` fields are optional. If neither appear, FitHelper will display all diaries.  
If there is no diary under the specified `DATE`, this command will be discarded with the reminder of `DIARY_NOT_FOUND`.

- The field **DATE** has higher priority than **KEYWORDS** in the search. i.e. If the **DATE** field is non-empty, regardless of the presence and the content of the **KEYWORDS** field, the diary under that date will be displayed. If the specified **DATE** contains no previous diary logs, no diary will be listed.
- If the field **DATE** is left empty, only the **KEYWORDS** field is considered in the search, similar to the case of **find** command for food/sport entries.
- The keyword searching rules are the same as in [Section 4.2.5 Locating entries by name : find](#).

Examples:

- **findDiary k/running**  
Displays diaries with their content containing the keyword **running**, ignoring the letter capitalization.
- **find d/2020-03-31 k/cake**  
Returns the diary on the date of **2020-03-31** regardless of the **KEYWORDS** field.

#### 4.3.5. Delete a diary: **deleteDiary**

- **deleteDiary d/DATE**  
deletes the diary on the specified **DATE**.

If there is no diary under the specified **DATE**, this command will be discarded with the reminder of **DIARY\_NOT\_FOUND**.

Examples:

- **deleteDiary d/2020-03-31**  
This command deletes the diary log on the date **2020-03-3**, if the diary on this date exists, and discards the command if no diary was added previously on this date.

#### 4.3.6. Clear the diary: **clearDairy**

- **clearDairy** clears all diary logs in this FitHelper.

##### **TIP**

The user can always revoke the clearing command by calling **undo**, which will be explained in a later section.

### 4.4. Clear All Entries And Diaries: **clear**

Clears all entries and diaries from the fitness log book.

Format: **clear**



## 4.5. Undo : **undo**

**undo** revokes the last undoable command.

This command back-roll FitHelper to the previous status before the last undoable command was executed.

<b>NOTE</b>	Undoable commands include: <b>add</b> , <b>edit</b> , <b>delete</b> , <b>clear</b> , <b>addDiary</b> , <b>appendDiary</b> , <b>editDiary</b> , <b>deleteDiary</b> , <b>clearDiary</b> . Other commands are not affected by <b>undo</b> command. The same applies for <b>redo</b> .
-------------	--

<b>NOTE</b>	After executing <b>undo</b> or <b>redo</b> , FitHelper switches to <b>Home Page (DashBoard)</b> .
-------------	---

Examples:

- **undo** (after **addDiary d/2020-03-31 dc/I am happy.**)  
This **undo** commands remove the added diary log from FitHelper.

## 4.6. Redo : **redo**

**redo** is the reverse of **undo**.

This command re-execute the last undoable command that has been undone.

<b>WARNING</b>	<b>redo</b> can only be executed after an <b>undo</b> has been previously performed.
----------------	--

Examples:

- **redo** (after **undo**-ing the previous **addDiary d/2020-03-31 dc/I am happy.**)  
This **redo** commands adds the diary log back to FitHelper.

## 4.7. Calendar View

Calendar view will display all the food and sports entries for each day of the referenced week/month. The referenced date is default set to be the current date, but can be changed to a user given date. Calendar can be switched between two modes, list mode or timetable mode, with default set to be in timetable mode.

Format for entering calendar view: **calendar**

### 4.7.1. Calendar list mode: **calendar m/l** [d/DATE]

- **Monthly View**

The lists of entries for food and sports are display by their given dates in the referenced month. Completed entries will be strikethrough. Calorie value is also shown for each date.

- **Calendar of the month**

On the top right corner, the dates of the referenced month displayed, with red-colored date if the given date has more calorie intake than calorie burnt (considering the entries with status done of that particular date), else if calorie burnt is greater than calorie intake, the font color is

green. Default color is blue. For each date, upon clicking, a popup window will show all entries of the date. The ones with strikethrough in text means the status of the entry is done. For this functionality, the corresponding command is `calendar sh/DATE`.

- **Daily calorie status**

On the bottom right corner, calorie status for existing entries will be displayed by their dates, showing the calorie intake from food, calorie burnt from sports as well as total calorie. The calculation only considers entries which are completed.

Format for switching to list mode: `calendar m/l [d/DATE]`



Figure 12. Calendar list mode

#### 4.7.2. Calendar timetable mode: `calendar m/tb [d/DATE]`

- **Weekly View**

The entries of food and sports of the given time period are displayed. Food entries will be in pink, sports entries will be in blue, and entries completed will be in grey regardless of their type. No entries can have time clashes. Entries with long names or locations will not be displayed fully. Only when mouse is over the entry, the details will be shown. Entries upon clicking will have console errors, caused by disabling certain functionality from third party libraries.

- **Calendar of the month**

Same as above.

- **Upcoming list**

It displays all entries of which start time is after the current date and time and within the referenced month. Only entries with status **Undone** will be shown. So if the referenced month is in the past, no entries will be shown.

Format for switching to timetable mode: `calendar m/tb [d/DATE]`

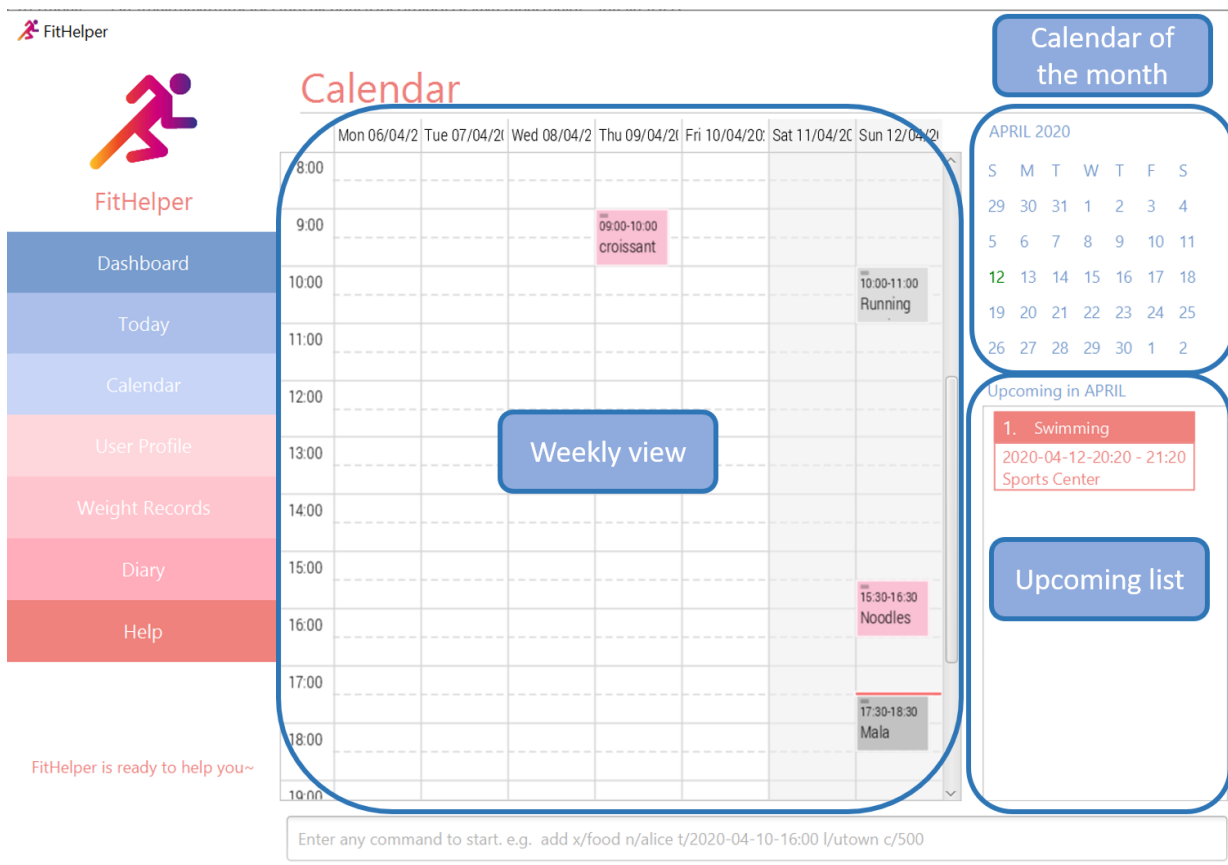


Figure 13. Calendar timetable mode

#### 4.7.3. Calendar display entries from a day: `calendar sh/DATE`

A pop up window will display all entries from the chosen date.

Format for showing entries of a particular date: `calendar sh/DATE`

#### 4.7.4. Set calendar referenced date: `calendar d/DATE`

Sets the referenced date of the current calendar page.

Format for showing entries of a particular date: `calendar d/DATE`

Examples:

- `calendar m/tb d/2020-04-15`
- `calendar m/lis d/2020-04-20`
- `calendar sh/2020-05-01`

## 4.8. Daily Summary

### 4.8.1. Today Page: `today`

Today page serves to be a summary for the daily arrangements.

It shows the daily schedule for the user. Users can see the entries for the day, a recommended lunch place, and their performances. They can also see their diary for the day as well as the rewarding point.

Format: **today**

- **Daily Food/Sports Entries**

The lists of food and sports entries on "today" are displayed in two list view, with indices in chronological order specific for today.

- **Plan Counter**

The 4 counters keep track of the number of done/undone food/sports plans on today.

- **Calorie Report**

Calorie report contains the data of daily calorie intake/consumption from done food/sports entries correspondingly. The food calorie pie chart consists of all food entries on today, regardless of the status. The labels are the corresponding indices of the food entries in **Today's Food** list. From the pie chart, the user can view the component of calorie intake of each food entry, so he/she can adjust the diet plan.

**WARNING**

In cases where some food entries contribute to the great majority of the total food intake, the pie chart only displays labels for food entries that contain relatively high calorie values.

- **Task Completion**

The user's daily task completion is shown in percentage (round to integer).

- **FitHelper Feedback**

Based on the user's intake food calorie and sport task completion, FitHelper provides suggestions and reminders in the **FitHelper Feedback** area.

## 4.8.2. Some Day Page [coming in v2.0]

The **Today Page** for some day in the history can be displayed. Format: **today DATE**

**TIP**

The **DATE** should be equal or prior to **today**. Requiring a **Today Page** for a future **date** will generate a blank page if that particular day does not have any entries.

Examples:

- **today 2020-02-14**

## 4.9. Weekly Summary

### 4.9.1. Weekly Report [coming in v2.0]

A weekly report serves as a summary for the past week. The user can see his performance in the past week. The weekly report also contains his rewarding points, diary logs, and preferred sports and food from the previous week.

Format: **weekreport**

- **Weekly Performance**

The weekly performance is generated based on the ratio of done and undone tasks and the calorie consumption, together with the system feedbacks.

- **Rewarding Points**

Rewarding points gained from the past week and current "fit level" is also shown. The weekly increment of rewarding points from past several weeks can be visualized.

- **Trendy Food and Sports**

Users can see his preferred food and sports, based on his entries for the past week.

#### 4.9.2. Some Week Report [coming in v2.0]

The **Week Report** for some week in the history can be displayed.

Format: **weekreport DATE** where **DATE** specifies the week it is in

**TIP**

The **DATE** should be equal or prior to days in the current week. Requiring a **Week Report** for a future week will generate a blank page if that particular day does not have any entries.

Examples:

- **weekreport**
- **weekreport 2020-02-14**

## 4.10. Keep User Profile

### 4.10.1. Profile Page: **profile**

Profile page serves to be a summary for basic user data.

The profile information includes: Name, Age, Gender, Address, Height, Target Weight, Current Weight and Current BMI.

Format: **profile**

### 4.10.2. Update Profile Data : **update**

Update user data in the profile by attributes. Profile attributes include: Name, Age, Address, Gender, Height and Target Weight.

Every **update** command will lead to the profile page.

Format: **update [-f] attr/ATTRIBUTE v/VALUE**

- If no user profile data is provided by the user, FitHelper will initialize with the sample profile data.
- The updated attribute name is **not** case-sensitive and can include spaces, but the name must match some fields in user profile.
- e.g. Both `attr/target weight` and `attr/TARGETWEIGHT` are acceptable.
- Any updated value should follow its original data type.
- If the chosen updated attribute has already had original value, **flag -f** need to be used to enable **force overwrite**. The flag should be added **just after the command word**, i.e. `update -f attr/ATTRIBUTE v/VALUE`.

Examples:

- `update attr/height v/160`
- `update -f attr/name v/Alice Wang`

## 4.11. Keep Weight Records

### 4.11.1. Weight Page: `weight`

Weight page serves to be a summary for user's weight and BMI changes according to time. It shows user data in graph for easy understanding. By default, it will generate graph from all history data chronologically.

- **Gap notification**

The top notification shows the comparison between user current weight and target weight.

- If current weight is **larger** than target, the gap between the two will be highlighted.
- If current weight is **the same or less** than the target, a succeed notification will be generated.

- **Trend Graph - Weight**

Display a trend graph of user's weight according to time.

- **Trend Graph - BMI**

Display a trend graph of user's BMI according to time. The BMI value is calculated by weight and height value at that date.

Format: `weight`

### 4.11.2. Add Weight Records : `addWeight`

Add a new weight record into the weight records database. A weight record is related to date and weight value, and a auto-computed BMI value will be stored as well.

Every `addWeight` command will lead to the weight page. If a new weight record is added successfully, two new points will be added into the two trend graphs separately.

Format: `addWeight v/WEIGHT_VALUE [d/DATE]`

- If no weight record exists in the database, "Not Available Now" will be shown in Profile Page's Current Weight and Current BMI fields.
- The date of a new weight record can be **omitted** when user inputs the `addWeight` command. By default, it will refer to the date of today.
- The date should be in format of `yyyy-MM-dd`, and should **not after the date of today**.
- **No two weight records should have the same date.** If adding a new weight record with the same date as an existing weight, a warning will be generated, and thus will fail to add.

Examples:

- `addWeight v/50.0 d/2020-02-01`
- `addWeight v/52.30`

### 4.11.3. Edit Weight Records : `editWeight`

Edit an existing weight record in the weight records database. A weight record is identified by its **unique date**, and user can find the date on the x-axis of the Weight Trend Graph.

User are able to edit the weight value, and corresponding BMI value will be auto-computed using the new weight value and user's **current height**.

Every `editWeight` command will lead to the weight page. If a weight record is edited successfully, two new points will change their positions on the two trend graphs separately.

Format: `editWeight [d/DATE] v/NEW_WEIGHT_VALUE`

- The date should be in format of `yyyy-MM-dd`. If no existing weight record is on the input date, the input date is considered as invalid, and thus a warning will be thrown.
- If the date is **omitted** when user inputs the `editWeight` command, by default, it will refer to the date of today.
- If the edited weight record is the **latest weight record** in the database, an update in Profile Page's Current Weight and Current BMI fields can be found as well.
- If the new weight value is the same as original weight value in the weight records, an exception will be thrown.

Examples:

- `editWeight d/2020-02-01 v/51.0`
- `editWeight v/52.40`

### 4.11.4. Delete Weight Records : `deleteWeight`

Delete an existing weight record in the weight records database. Same as `editWeight` command, a weight record is identified by its **unique date**, and user can find the date on the x-axis of the Weight Trend Graph.

Every `deleteWeight` command will lead to the weight page. If a weight record is deleted successfully, two corresponding points will be removed from the two trend graphs separately.

Format: `deleteWeight [d/DATE]`

- The date should be in format of `yyyy-MM-dd`. If no existing weight record is on the input date, the input date is considered as invalid, and thus a warning will be thrown.
- If the date is **omitted** when user inputs the `deleteWeight` command, by default, it will refer to the date of today.
- If the deleted weight record is the **latest weight record** in the database, the second latest weight record will be used to update Profile Page's Current Weight and Current BMI fields.

Examples:

- `deleteWeight d/2020-02-01`
- `deleteWeight`

#### 4.11.5. Clear Weight Records : `clearWeight`

Clear all weight records in the weight records database.

Weight Page's graphs will be empty, and Profile Page's Current Weight and Current BMI fields will be **Not Available Now** after clear all weight records.

Format: `clearWeight`

### 4.12. Exit the program : `exit/bye/quit`

Exits the program.

Format: `exit` or `bye` or `quit`

### 4.13. Save the data

fitness log book data are saved in the hard disk automatically after any command that changes the data.

There is no need to save manually.

Three local database in Json format will exist after running FitHelper:

- **fithelper.json** : data related to entries and diaries.
- **userprofile.json** : data related to user profile.
- **weightrecords.json** : data related to weight records.



## 5. Command Summary

- **Help** - switch to Help Page : `help`
- **Entry** - switch to DashBoard : `home`
- **Entry** - add an entry `add x/TYPE n/NAME t/TIME l/LOCATION c/CALORIE ...`
- **Entry** - list all entries or entries of a given date : `list [d/DATE]`
- **Entry** - view reminders list only : `reminder`
- **Entry** - edit an entry : `edit x/TYPE i/INDEX [n/NAME] [t/TIME] [l/LOCATION] [c/CALORIE] ...`
- **Entry** - find and list entries by keywords : `find k/KEYWORDS`
- **Entry** - delete an entry : `delete x/TYPE i/INDEX`
- **Entry** - sort entry list : `sort [x/TYPE] by/SORT_BY [o/ORDER]`
- **Entry** - check calorie reference : `check x/TYPE k/KEYWORDS`
- **Diary** - switch to Diary Page : `diary`
- **Diary** - add a diary : `addDiary d/DATE dc/CONTENT`
- **Diary** - edit a diary : `editDiary d/DATE dc/CONTENT`
- **Diary** - delete a diary : `deleteDiary d/DATE`
- **Diary** - find a diary : `findDiary [d/DATE] [k/KEYWORDS]`
- **Diary** - clear all diaries : `clearDiary`
- **Clear** - clear all entries and diaries : `clear`
- **Undo** - undo commands related to entries and diaries : `undo`
- **Redo** - redo commands related to entries and diaries : `redo`
- **Calendar** - switch to Calendar Page : `calendar`
- **Calendar** - display from a referenced date : `calendar d/DATE`
- **Calendar** - change to list mode : `calendar m/lS [d/DATE]`
- **Calendar** - change to timetable mode : `calendnar m/tb [d/DATE]`
- **Calendar** - display entries from a particular date : `calendar sh/DATE`
- **Today** - switch to Today Page : `today`
- **Profile** - switch to Profile Page: `profile`
- **Profile** - update profile data: `update [-f] attr/ATTRIBUTE V/VALUE`
- **Weight** - switch to Weight Page: `weight`
- **Weight** - add a weight record `addWeight v/VALUE [d/DATE]`
- **Weight** - edit a weight record `editWeight [d/DATE] v/VALUE`
- **Weight** - delete a weight record `deleteWeight [d/DATE]`

- **Weight - clear all weight records** `clearWeight`
- **Exit the Program:** `exit` or `bye` or `quit`