GAZEEEBO - User Guide

By: `Team CS2113T-F10-4` Since: `Sep 2019` Licence: `NUS`

Content Page

- 3.2.24. Create a recurring task based on a keyword: weekly/monthly/yearly
- 3.2.25. See all the tasks that are marked as undone: undone list
- 3.2.26. Tagging tasks: #TAG_NAME
- 3.2.27. Undo commands: undo
- 3.2.28. Note section for a particular period
  - 3.2.28.1. Adding a note to a particular day/week/month: addNote
  - 3.2.28.2. Editing a note for a particular day/week/month: editNote
  - 3.2.28.3. Deleting a note for a particular day/week/month: deleteNote
  - 3.2.28.4. Listing out all notes for a particular day/week/month: listNote

3.3. Places page: places

- 3.3.1. Adding a place: add
- 3.3.2. Finding a place: find
- 3.3.3. List all places: list
- 3.3.4. Delete a place from the list: delete
- 3.3.5. Undo previous places command: undo

3.4. Note page: notes

- 3.4.1. View goal and list of modules: view
- 3.4.2. Edit goal: edit /n
- 3.4.3. Add a module: add /n
- 3.4.4. Delete a module: delete /n
- 3.4.5. Go to the module page: module /n
  - 3.4.5.1. View the module’s notes: view
  - 3.4.5.2. Edit the module’s name: edit mod /n
  - 3.4.5.3. Add an assessment: add assmt /n /a
  - 3.4.5.4. Edit an assessment name: edit assmt /n /a
  - 3.4.5.5. Edit an assessment weightage: edit weight /n /a
  - 3.4.5.6. Delete an assessment: delete assmt /n
  - 3.4.5.7. Add a miscellaneous information: add msc /n
  - 3.4.5.8. Edit a miscellaneous information: edit msc /n /a
  - 3.4.5.9. Delete a miscellaneous information: delete msc /n
  - 3.4.5.10. View the list of possible commands for the module page: commands
  - 3.4.5.11. Go back to the note page: esc
  - 3.4.5.12. Find an assessment: find [coming in v2.0]
  - 3.4.5.13. Find a miscellaneous information: find [coming in v2.0]
- 3.4.6. View the list of possible commands for the notes page: commands
- 3.4.7. Go back to the main page: esc
- 3.4.8. Find a module: find [coming in v2.0]

3.5. Contacts page: contacts

- 3.5.1. Adding a contact: add name,number
- 3.5.2. Finding a contact: find name
- 3.5.3. List all contacts: list
- 3.5.4. Delete a contact from the list: delete name

3.5.5. View commands in Contacts page: commands

3.5.6. Undo a previous Contact command: undo

3.6. CAP page: cap

3.6.1. Adding a new module: add semester number,module’s code,module’s credit,module’s grade

3.6.2. Finding a module: find module

3.6.3. List all modules: list all/semester number

3.6.4. Delete a module from the list: delete module

3.6.5. View commands in CAP page: commands

3.6.6. Undo previous command: undo [coming in v2.0]

3.7. Expenses page: expenses

3.7.1. Add expenses command: add

3.7.2. Find expenses on a certain date: find date

3.7.3. List all expenses: list

3.7.4. Delete a certain expense: delete

3.7.5. Undo previous expenses command: undo

3.8. Specialization page: spec

3.8.1. List all specializations and technical electives: list

3.8.2. Key in completed electives: complete

3.8.3. Undo complete elective command: undo [coming in v2.0]

3.9. Password to enter the planner: jjjry

3.10. Change Password based on keyword: change password

3.11. Module Planner page: moduleplanner

3.11.1. Add module to specific semester column: add

3.11.2. Add module to specific semester column: add

3.11.3. Delete module from specific semester column:delete

3.11.4. Shift module from specific semester column to another column: shift

3.11.5. Show a prerequisite tree of a module : prerequisite

3.11.6. Undo a previous module planner command: undo

3.11.7. Auto adjust the table based on prerequisites: adjust [Coming in v2.0]

3.12. Exiting the program: bye

3.13. Redo function: redo [coming in v2.0]

4. UI

1. Introduction

Gazeeebo is a multifunctional school planner with features that will ease your university life as a Computer Engineering (CEG) student. It is optimized for those who prefer to work with a keyboard rather than a mouse and it does not require internet to operate. Gazeeebo will be the CEG planner you never knew you needed!

2. Quick Start

1. Ensure you have Java 11 or above installed in your computer.
2. Download the latest [CS2113T-F10-4][Gazeeebo].jar here.
3. Copy the file to the folder you want to use as the home folder for your Gazeeebo application.
4. Open up your computer's command line window. It is Command Prompt for Windows; Terminal for Mac and Linux.
5. Change directory to the folder where you have placed your gazeeebo.jar file by using the `cd` command. By doing this, the text files created by [CS2113T-F10-4][Gazeeebo].jar will be stored in this folder.
6. Run the [CS2113T-F10-4][Gazeeebo].jar file on the command line with the format `java -jar FULL_FILE_PATH_OF_JAR_FILE`. (The last dot is a full stop and not meant to be typed into the command line)
    a. To get the file path of the [CS2113T-F10-4][Gazeeebo].jar file, right click on it and select properties as shown in the picture below.

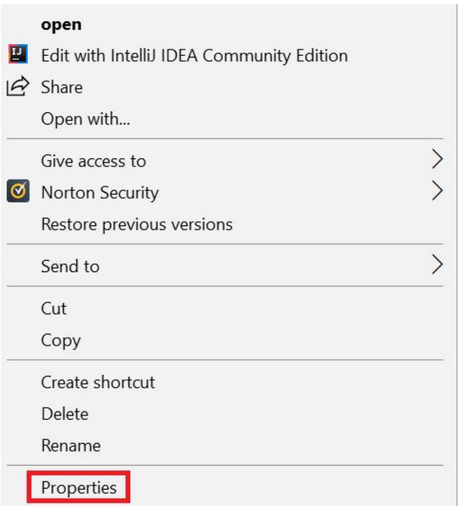
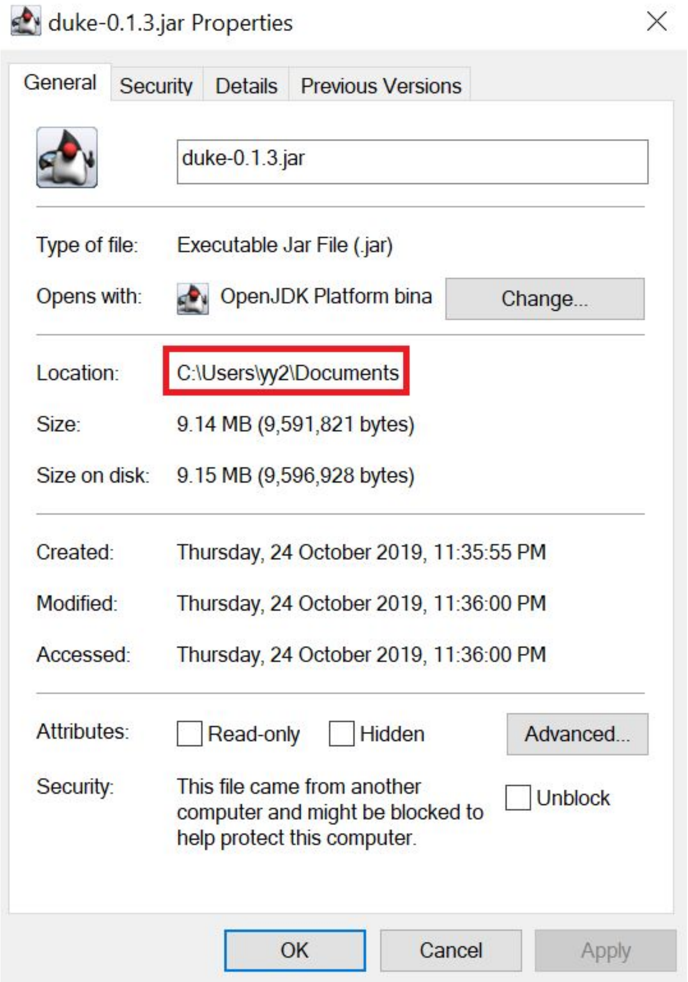


Figure 2.1. Right click the jar file and select properties

b. In the window that pops up, copy the location path as shown by the red box below and paste that into the command line window.



duke-0.1.3.jar Properties

duke-0.1.3.jar

Type of file: Executable Jar File (.jar)

Opens with: OpenJDK Platform bina

Location: C:\Users\yy2\Documents

Size: 9.14 MB (9,591,821 bytes)

Size on disk: 9.15 MB (9,596,928 bytes)

Created: Thursday, 24 October 2019, 11:35:55 PM

Modified: Thursday, 24 October 2019, 11:36:00 PM

Accessed: Thursday, 24 October 2019, 11:36:00 PM

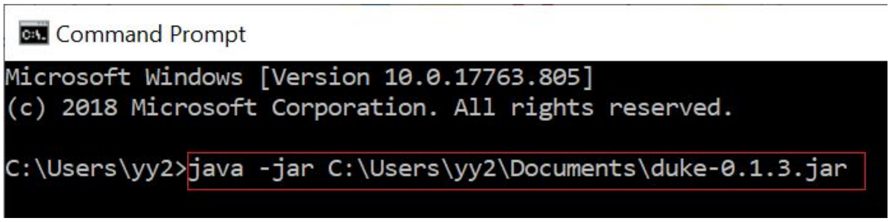
Attributes: □ Read-only □ Hidden

Security: This file came from another computer and might be blocked to help protect this computer. □ Unblock

Figure 2.2. Copy the location of the gazeeebo.jar file

c. Without pressing the spacebar or ENTER key, continue to type in `[CS2113T-F10-4][Gazeeebo].jar`. (the last dot is a full stop and not meant to be typed into the command line)

7. Or you could just drag the [CS2113T-F10-4][Gazeeebo].jar file into the command line window after typing in `java -jar`. (The last dot is a full stop and not meant to be typed into the command line)
8. Finally, press ENTER to execute the [CS2113T-F10-4][Gazeeebo].jar file.
9. An example of running a jar file is shown in the picture below in Figure 2.3. The text in the red box is what you type into the command line window.



```
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\yy2>java -jar C:\Users\yy2\Documents\duke-0.1.3.jar
```

Figure 2.3. Command Prompt command example

10. System will prompt you for a password, default password: `jjjry`
11. Some example commands you can try:
    a. `help`: displays the full help page
    b. `notes`: goes to the note page
12. The next section, section 3 on features, will explain in more detail about the usage of each command.
13.

3. Features

Difference Between the Fonts Used

`Courier New` font is used for commands to be keyed into the command line of the product, while Arial font is used for all else, such as descriptions and notes pertaining to a command.

Command Format

Words in `UPPER_CASE` are the parameters to be supplied by the user e.g. in `todo TASK_DESCRIPTION`, `TASK_DESCRIPTION` is a parameter which can be used as `todo eat`.

3.1. Viewing help: help

The following 2 commands can be used everywhere in the application.

3.1.1. Viewing the full help page: help

Calls out the full help page.
Format: `help`

3.1.2. Viewing help for a specific page: help

Calls out help for a specific page in the application.
Format: `help PAGE_NAME`

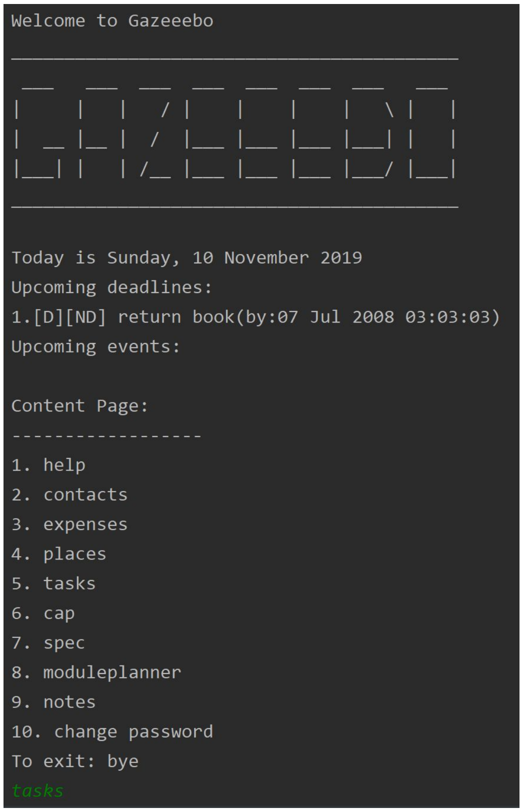
List of possible commands:

- `help tasks`
- `help places`
- `help notes`

- `help module`
- `help contacts`
- `help cap`
- `help expenses`
- `help spec`
- `help password` – Provides help on how to login and the default password.
- `help change password`
- `help moduleplanner`
- `help bye`

3.2. **Tasks Page: tasks**

Brings you to your tasks page from the main menu page.
In your tasks page, you can add different kinds of tasks to your schedule and view your upcoming tasks. The following commands in this section can only be used in the tasks page.



```
Welcome to Gazeeebo
__________________________________________
 ___   ___  ___  ___  ___  ___  ___   ___
|     |   |   / |    |    |    |   \ |   |
|  __ |__ |  /  |___ |___ |___ |___| |   |
|___| |   | /__ |___ |___ |___ |___/ |___|
__________________________________________


Today is Sunday, 10 November 2019
Upcoming deadlines:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
Upcoming events:

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
tasks
```

Figure 3.2.1 How to get to the tasks page

How to get to the tasks page:

- Type in the command `tasks` in the main menu page and press ENTER as shown in Figure 3.2.1 above.

- You can only go to the tasks page from the main menu.

3.2.1. Adding a todo: todo

Adds a todo task to the task list.

Format: `todo TASK_DESCRIPTION`

Examples:

- `todo eat`
- `todo watch TV`

3.2.2. Adding a deadline: deadline

Adds a deadline task to the task list.

Format: `deadline TASK_DESCRIPTION /by YYYY-MM-DD HH:MM:SS`

Examples:

- `deadline assignment /by 2019-12-04 12:07:08`
- `deadline watch TV /by 1988-06-27 08:46:37`

3.2.3. Adding an event: event

Adds an event task to the task list.

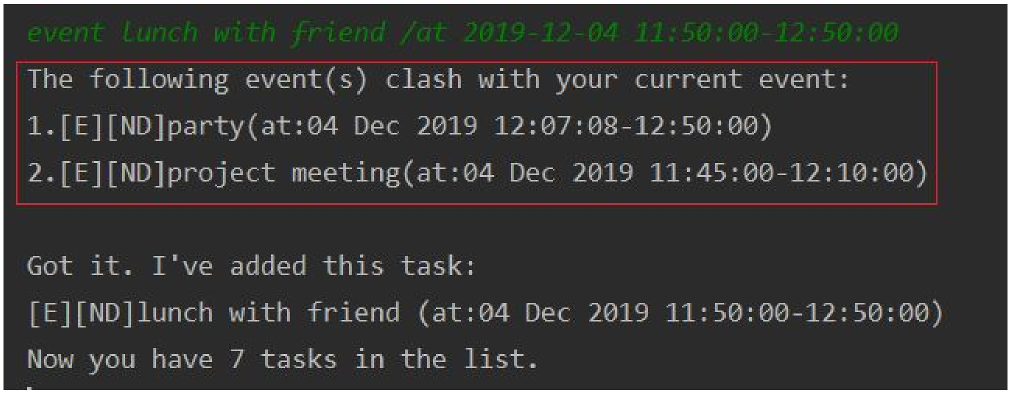
Format: `event TASK_DESCRIPTION /at YYYY-MM-DD HH:MM:SS-HH:SS:MM`

Examples:

- `event party /at 2019-12-04 12:07:08-12:50:00`
- `event project meeting /at 1988-06-27 08:46:37-09:50:10`

3.2.1. Detect scheduling anomalies

If the event you are adding has a conflict in timing with an existing event, Gazeeebo will show you which events clash with your incoming event as shown in the figure below.



```
event lunch with friend /at 2019-12-04 11:50:00-12:50:00
The following event(s) clash with your current event:
1.[E][ND]party(at:04 Dec 2019 12:07:08-12:50:00)
2.[E][ND]project meeting(at:04 Dec 2019 11:45:00-12:10:00)

Got it. I've added this task:
[E][ND]lunch with friend (at:04 Dec 2019 11:50:00-12:50:00)
Now you have 7 tasks in the list.
```

Figure 3.2.1.1 Detect scheduling anomalies

3.2.4. Listing out the task list: list

List out all the tasks in the task list.

Format: `list`

3.2.5. Delete task(s): delete

Deletes n number of tasks at once from the list or delete all tasks at once.

Format: `delete TASK_NUM1 and TASK_NUM2`

Format: `delete all`

Examples:

- `delete 3 and 5`
- `delete 6`
- `delete all`

3.2.6. Set a task as done: done

Marks a task as done.

Format: `done TASK_NUM`

Examples:

- `done 1`
- `done 6`

3.2.7. Find tasks based on a keyword: find

Searches for tasks based on a specified keyword.

Format: `find KEYWORD`

Examples:

- `find read`
- `find eat`

3.2.8. Viewing the schedule for a particular period

3.2.8.1. Day: scheduleDaily

Lists out your schedule and your notes for the specified date.

Format: `scheduleDaily YYYY-MM-DD`

Examples:

- `scheduleDaily 2019-08-09`
- `scheduleDaily 2022-10-26`

3.2.8.2. Week: scheduleWeekly

Lists out your schedule and your notes for the specified week.

Format: `scheduleWeekly YYYY-MM-DD YYYY-MM-DD`

Format details:

- The first date in the format above must be a Monday and the second date must be a Sunday.
- The first date must be before the second date.
- The number of days between the two dates must be 7.

Examples:

- `scheduleWeekly 2019-10-07 2019-10-13`
- `scheduleWeekly 2019-09-02 2019-09-08`

3.2.8.3. Month: scheduleMonthly

Lists out your schedule and your notes for the specified month.
Format: `scheduleMonthly YYYY-MM`

Examples:

- `scheduleMonthly 2020-10`
- `scheduleMonthly 2019-09`

3.2.9. Adding a 'do-within-a-period' task: /between

Adding tasks that need to be done within a certain period to the list.
Format: `TASK_DESCRIPTION /between YYYY-MM-DD and YYYY-MM-DD`

Examples:

- `collect book /between 2019-12-04 and 2019-12-05`
- `order food /between 2017-12-04 and 2017-12-05`

3.2.10. Adding a Tentative Event base on a keyword: tentative

Creating a tentative event with description of the event.
Format: `tentative TASK_DESCRIPTION`

Examples:

- Tentative return book

Key in all possible time slots, ends input with a keyword: /

Format :
```
YYYY-MM-DD HH:MM:SS-HH:MM:SS
YYYY-MM-DD HH:MM:SS-HH:MM:SS
/
```

Examples:

- `2019-08-04 05:04:03-01:01:01 2019-09-18 07:11:44-10:10:10 /`

Format Detail:

1. The date must follow yyyy-mm-dd hh:mm:ss format.

Ensure your creation with a keyword: yes
Example： `yes`

3.2.11. Confirming a Tentative Event base on a keyword: confirm

Confirming one of the timeslots of the exact tentative event with its INDEX.
Format: `confirm INDEX`

Example:

- `confirm`

Choosing which time slot to confirm with its SEQUENCE_NUMBER
Format: SEQUENCE_NUMBER

Example:

- `1`

Ensure your confirmation with a keyword: yes
Example： `yes`

3.2.12. Giving reminders for upcoming tasks

List out the upcoming tasks when the program starts up.

3.2.13. Editing a task: edit

Edit task with task’s INDEX.
Format: `edit INDEX`

Example:

- `edit 1`

Choose time/description/both to edit
Format: `time`

Example:

- `2019-10-10 10:10:10`

3.2.14. Adding a task with a fixed duration: /require

Adding a task takes a fixed amount of time but does not have a fixed start/end time to the list.

Format: `TASK_DESCRIPTION /require FIXED_AMOUNT_OF_TIME`

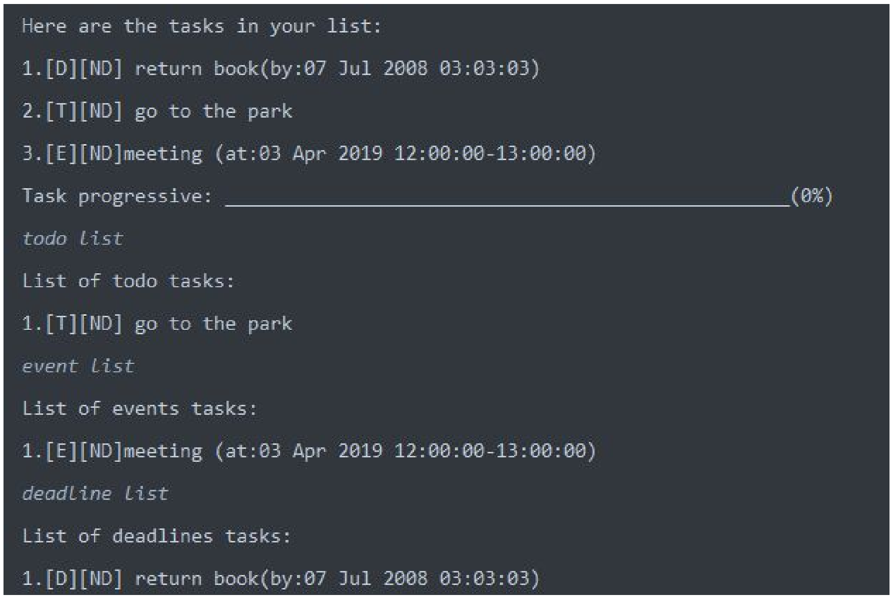
Example:

- `reading the sales report /require 2 hours`
- `project meeting /require 4 hours`

3.2.15. Categorize tasks: CATEGORY_NAME

List out all the categorized tasks in the category task list.

Format: `CATEGORY_NAME list`



```
Here are the tasks in your list:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
2.[T][ND] go to the park
3.[E][ND]meeting (at:03 Apr 2019 12:00:00-13:00:00)
Task progressive: ________________________________________________(0%)
todo list
List of todo tasks:
1.[T][ND] go to the park
event list
List of events tasks:
1.[E][ND]meeting (at:03 Apr 2019 12:00:00-13:00:00)
deadline list
List of deadlines tasks:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
```

Figure 3.2.15.1. Shows different categories of todo, event and deadline lists

Example:

- `event list`
- `deadline list`
- `todo list`

3.2.16. Rearrange my tasks based on the task importance which I have set: rank

List shows my ranked tasks based on priority in descending order.

Format: `TASK_NUM rank RANK_NUM`

Example:

- `4 rank 1`
- `3 rank 2`
- `2 rank 3`
- `1 rank 4`

3.2.17. Calendar View

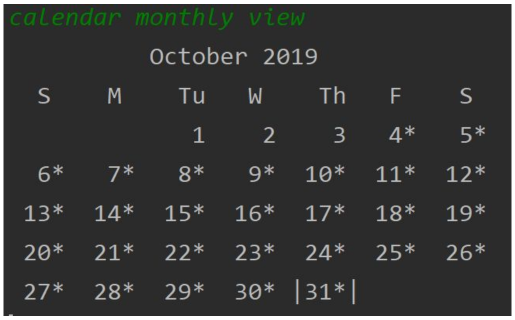
3.2.17.1 View current month in a calendar view: calendar monthly view

Shows the dates and current month in a calendar view.

Dates with tasks will be demarcated with a ‘ * ’.

Current date will be demarcated between ‘ | ’.

Format: `calendar monthly view`



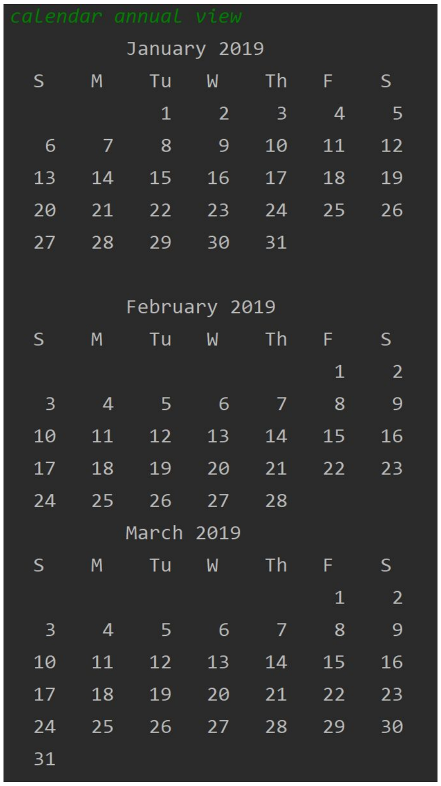
```
calendar monthly view
          October 2019
  S     M     Tu    W     Th    F     S
               1     2     3    4*    5*
 6*    7*     8*    9*   10*   11*   12*
13*   14*    15*   16*   17*   18*   19*
20*   21*    22*   23*   24*   25*   26*
27*   28*    29*   30*  |31*|
```

Figure 3.17.1.1. Calendar monthly view

3.2.17.2 View current year in a calendar view: calendar annual view

- ➔ Shows the dates and months in a calendar view.
- ➔ Dates with tasks will be demarcated with a ‘ * ’.
- ➔ Current date will be demarcated between ‘ | ’.

Format: `calendar annual view`



```
calendar annual view
          January 2019
  S     M     Tu    W     Th    F     S
               1     2     3     4     5
  6     7      8     9    10    11    12
 13    14     15    16    17    18    19
 20    21     22    23    24    25    26
 27    28     29    30    31


          February 2019
  S     M     Tu    W     Th    F     S
                                 1     2
  3     4      5     6     7     8     9
 10    11     12    13    14    15    16
 17    18     19    20    21    22    23
 24    25     26    27    28
          March 2019
  S     M     Tu    W     Th    F     S
                                 1     2
  3     4      5     6     7     8     9
 10    11     12    13    14    15    16
 17    18     19    20    21    22    23
 24    25     26    27    28    29    30
 31
```

Figure 3.2.17.2. First 3 months of the calendar annual view

3.2.18. Set a ‘done’ task as undone: undone

Marks a task as undone.

Format: `undone TASK_NUM`

Examples:

- `undone 1`
- `undone 6`

3.2.19. Save the list to a text file: save list [coming in v2.0]

Allows user to save the list in the text file.

Format: `save list`

3.2.20. See all the tasks that are marked as done: done list

List out all the tasks that are done in a list

Format: `done list`

3.2.21. Adding a do-after task: /after

Add a follow-up task when a task is marked as done.

Format: `FOLLOW_UP_TASK /after COMPLETED_TASK`

Examples:

- `return book /after read book`

3.2.22. Snooze a task based on a keyword: reschedule

➔ Snooze a task’s timeline with task’s `INDEX`

Format: `reschedule INDEX`

Example:

- `reschedule 1`

➔ Key in your new timeline with the data

Format: `YYYY-MM-DD HH:MM:SS`

Example:

- `1998-04-05 08:09:14`

Format Detail:

2. The date must follow yyyy-mm-dd hh:mm:ss format.

Ensure your rescheduling with a keyword: yes

Example：

- `Yes`

3.2.23. Reschedule a task based on a keyword: snooze

Reschedule a task’s timeline with task’s INDEX

Format: `snooze INDEX`

Example:

- `snooze 1`

Key in how much time you want to prolong

Format: `Y M D H`

Example:

- `1 2 3 4`

3.2.24. Create recurring tasks based on a keyword: weekly/monthly/yearly

A recurring task has any of the keywords.

Format: `TASK_DESCRIPTION`

Example:

- `event weekly seminar /at 2019-10-10 10:10:10-11:11:11`

Format: `done INDEX`

Example:

- `done 1`

3.2.25. See all the tasks that are marked as undone: undone list

List out all the tasks that are undone in a list

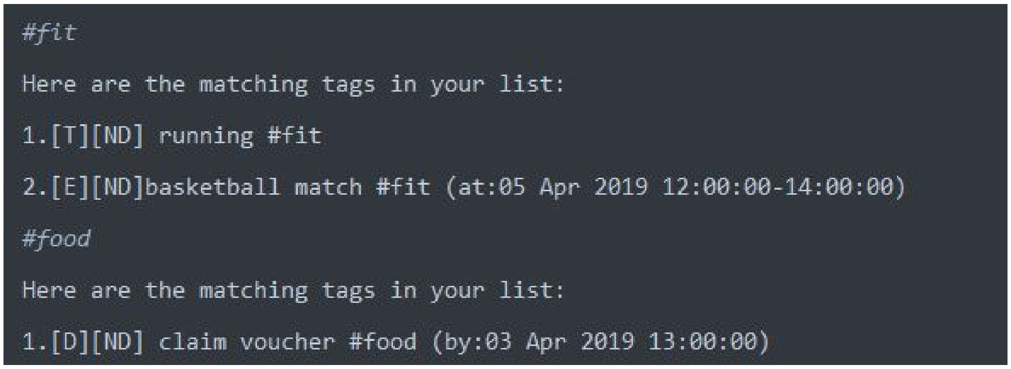
Format: `undone list`

3.2.26. Tagging tasks: #TAG_NAME

Tag tasks using the hashtag, and see the same tagged tasks in a list.

Format: `TASK_DESCRIPTION #TAG_NAME`

Format: `#TAG_NAME`



```
#fit
Here are the matching tags in your list:
1.[T][ND] running #fit
2.[E][ND]basketball match #fit (at:05 Apr 2019 12:00:00-14:00:00)
#food
Here are the matching tags in your list:
1.[D][ND] claim voucher #food (by:03 Apr 2019 13:00:00)
```

Figure 3.2.26.1 Displays different tag lists #fit and #food

Example:

To tag:

- `todo running #fit`
- `event basketball match #fit /at 2019-04-05 12:00:00-14:00:00`
- `deadline claim voucher #food /by 2019-04-03 13:00:00`

To see the tasks with the same tags:

- `#fit`
- `#food`

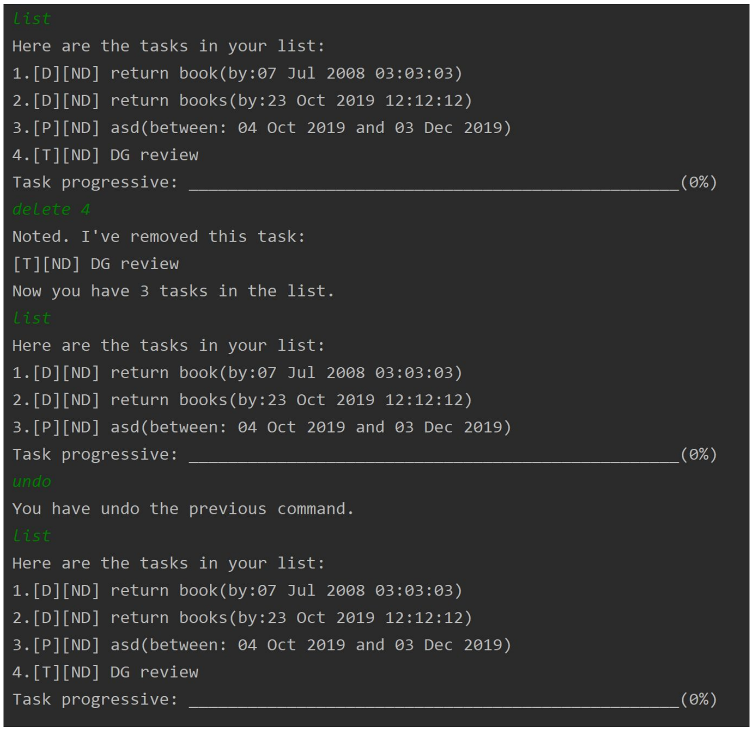
3.2.27. Undo commands: undo

Undo previous tasks commands

Format: `undo`

Previous task commands that you can undo:

- EventCommand
- TodoCommand
- DeadlineCommand
- TentativeEventCommand
- DoneCommand
- DeleteCommand
- DoAfterCommand
- TimeBoundCommand
- FixDurationCommand
- ConfirmTentativeCommand
- UndoneCommand



```
list
Here are the tasks in your list:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
2.[D][ND] return books(by:23 Oct 2019 12:12:12)
3.[P][ND] asd(between: 04 Oct 2019 and 03 Dec 2019)
4.[T][ND] DG review
Task progressive: __________________________________________________(0%)
delete 4
Noted. I've removed this task:
[T][ND] DG review
Now you have 3 tasks in the list.
list
Here are the tasks in your list:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
2.[D][ND] return books(by:23 Oct 2019 12:12:12)
3.[P][ND] asd(between: 04 Oct 2019 and 03 Dec 2019)
Task progressive: __________________________________________________(0%)
undo
You have undo the previous command.
list
Here are the tasks in your list:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
2.[D][ND] return books(by:23 Oct 2019 12:12:12)
3.[P][ND] asd(between: 04 Oct 2019 and 03 Dec 2019)
4.[T][ND] DG review
Task progressive: __________________________________________________(0%)
```

Figure 3.2.28.1 Undo a delete task command

3.2.28. Note section for a particular period

3.2.28.1 Adding a note to a particular day/week/month: addNote

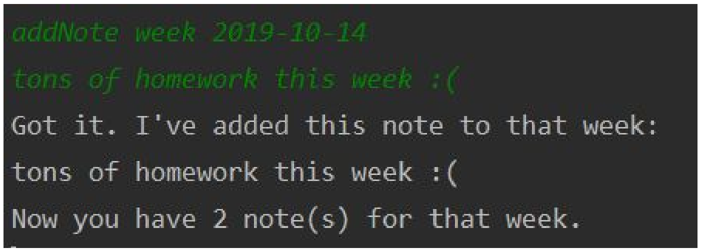
Adds a note to the note section of the specified day, week or month.
Format (day): `addNote day YYYY-MM-DD`
Format (week): `addNote week YYYY-MM-DD`
Format (month): `addNote month YYYY-MM`

Format details:

- The date specified in Format (week) above has to be a Monday.

Steps for adding a note (this example is for a particular week):

1. Type in the command in the format specified above (e.g. `addNote week 2019-10-14`).
2. Press ENTER.
3. Type in the note you want to add on that new line (e.g. `tons of homework for this week :(`).
4. Press ENTER.
5. An example is shown in the figure below.



```
addNote week 2019-10-14
tons of homework this week :(
Got it. I've added this note to that week:
tons of homework this week :(
Now you have 2 note(s) for that week.
```

Figure 3.2.28.1.1 Adding a note to a particular week

Examples for command format:

- Adding a note to a particular day:
  - `addNote day 2020-10-11`
- Adding a note to a particular week:
  - `addNote week 2019-10-14`
- Adding a note to a particular month:
  - `addNote month 2019-01`

3.2.28.2. <u>Editing a note for a particular day/week/month: editNote</u>

Edits an existing note in the note section of the specified day, week or month.

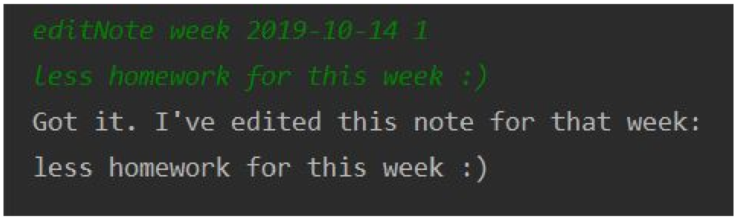
Format (day): `editNote day YYYY-MM-DD NOTE_NUMBER`
Format (week): `editNote week YYYY-MM-DD NOTE_NUMBER`
Format (month): `editNote month YYYY-MM NOTE_NUMBER`

Format details:

- `NOTE_NUMBER` starts from index 1.
- The date specified in Format (week) above has to be a Monday.

Steps for editing a note:

1. Type in the command in the format specified above (e.g. `editNote week 2019-10-14 1`).
2. Press ENTER.
3. Type in the edited version of the note on that new line (e.g. `less homework for this week :)`).
4. Press ENTER.
5. An example is shown in the figure below.



```
editNote week 2019-10-14 1
less homework for this week :)
Got it. I've edited this note for that week:
less homework for this week :)
```

Figure 3.2.28.2.1 Editing a note to a particular week

Examples for command format:

- Editing a note for a particular day:
  - `editNote day 2020-10-11 1`
- Editing a note for a particular week:
  - `editNote week 2019-10-14 3`
- Editing a note for a particular month:
  - `editNote month 2019-01 5`

3.2.28.3. Deleting a note for a particular day/week/month: deleteNote

Deletes an existing note in the note section of the specified day, week or month.

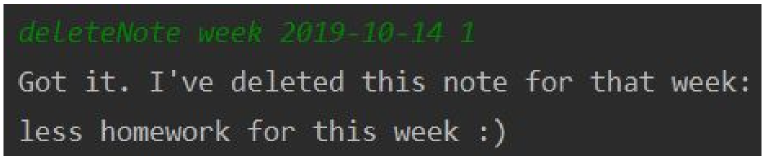
Format (day): `deleteNote day YYYY-MM-DD NOTE_NUMBER`
Format (week): `deleteNote week YYYY-MM-DD NOTE_NUMBER`
Format (month): `deleteNote month YYYY-MM NOTE_NUMBER`

Format details:

- `NOTE_NUMBER` starts from index 1.
- The date specified in Format (week) above has to be a Monday.

Steps for deleting a note (assigned to a day):

1. Type in the command in the format specified above (e.g. `deleteNote week 2019-10-14 1`).
2. Press ENTER.
3. An example is shown in the figure below.



```
deleteNote week 2019-10-14 1
Got it. I've deleted this note for that week:
less homework for this week :)
```

Figure 3.2.28.3.1 Deleting a note to a particular week

Examples for command format:

- Deleting a note that was assigned to a particular day:
  - `deleteNote day 2020-10-11 1`
- Deleting a note that was assigned to a particular week:
  - `deleteNote week 2019-10-14 3`
- Deleting a note that was assigned to a particular month:
  - `deleteNote month 2019-01 5`

3.2.28.4. Listing out all notes for a particular day/week/month: listNote

Lists out all the notes in the note section of the specified day, week or month.

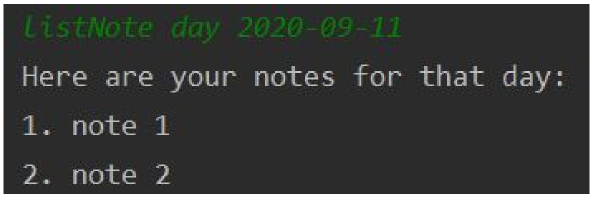
Format (day): `listNote day YYYY-MM-DD`
Format (week): `listNote week YYYY-MM-DD`
Format (month): `listNote month YYYY-MM`

Format details:

- The date specified in Format (week) above has to be a Monday.

Steps for listing out notes (for a particular day):

1. Type in the command in the format specified above (e.g. `listNote day 2020-09-11`).
2. Press ENTER.
3. An example is shown in the figure below.



```
listNote day 2020-09-11
Here are your notes for that day:
1. note 1
2. note 2
```

Figure 3.2.28.4.1 Listing the notes for a particular week

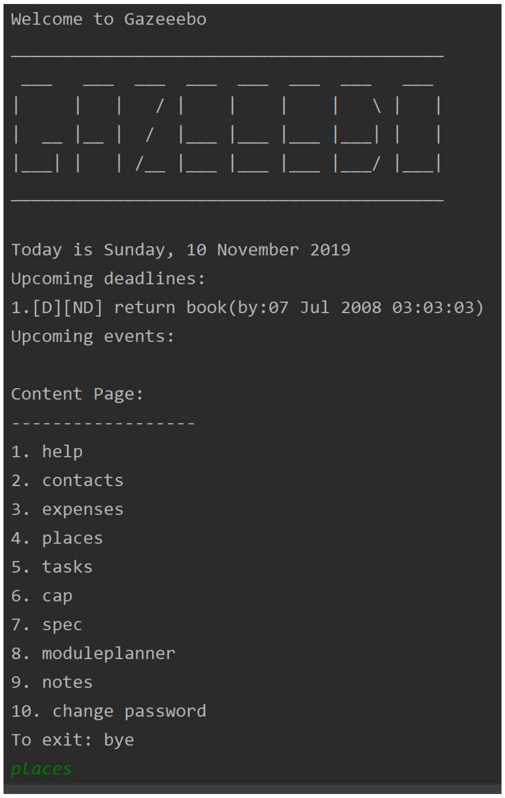
Examples for command format:

- Listing the notes for a particular day:
  - `listNote day 2020-10-23`
- Listing the notes for a particular week:
  - `listNote week 2019-10-14`
- Listing the notes for a particular month:
  - `listNote month 2019-01`

3.3. Places Page: places

A function to locate places and rooms in NUS School of Computing.

In the places page, you can search for a place that you want to find or you can add and delete places and locations to the current list.



```
Welcome to Gazeeebo
__________________________________________

 ___   ___  ___  ___  ___  ___  ___   ___
|     |   |   / |    |    |    |   \ |   |
|  __ |__ |  /  |___ |___ |___ |___| |   |
|___| |   | /__ |___ |___ |___ |___/ |___|
__________________________________________


Today is Sunday, 10 November 2019
Upcoming deadlines:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
Upcoming events:

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
places
```

Figure 3.3.1 How to get to places page

How to get to places page:

- Type in the command `places` in the main menu page and press ENTER as shown in Figure 3.3.1 above.
- Alternatively, you can type in the index of places, `4` and press ENTER.
- You can only go to the places page from the main menu.

3.3.1. Adding a new place and location: add

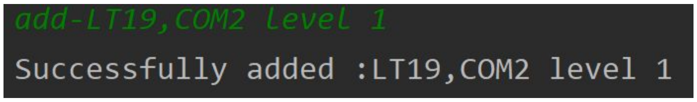
Adds and stores a new place and location.

Format : `add-room,location`

Steps for adding a place:

1. Type in the command `add-room,location` in the above format. Eg. `LT19,COM2 Level 1`
2. Press ENTER. A success message will appear.
3. Alternatively, you can input `add` or the index of the add command, `1`.

4. The system will then prompt you to enter the room and location in this format, `room,location`.



```
add-LT19,COM2 level 1
Successfully added :LT19,COM2 level 1
```

Figure 3.3.1.1. An example of adding a place

3.3.2. Finding a place in NUS School of Computing (SOC): find-place

Gives you the location of a specific place in SOC.

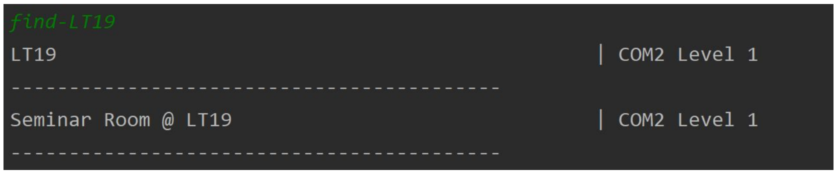
Format : `find-place`

Format details:

- `place` is the name of the place you want to find.

Steps for finding a place:

1. Type in the command in the format specified above. Eg. `find-LT19`
2. Press ENTER.
3. Alternatively, type `find` or the index for find command, `2`.
4. Press ENTER.
5. The system will prompt you to enter the place you want to find.



```
find-LT19
LT19                                               | COM2 Level 1
------------------------------------------
Seminar Room @ LT19                                | COM2 Level 1
------------------------------------------
```

Figure 3.3.2.1. An example of finding a place

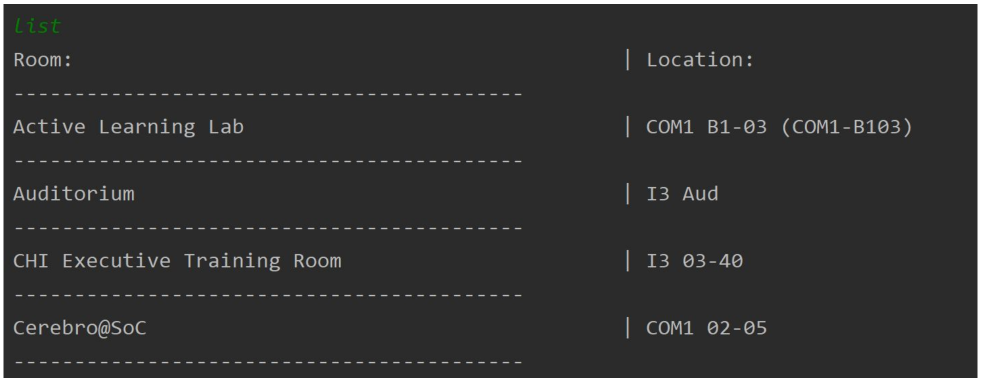
3.3.3. Listing all places in NUS School of Computing (SOC): list

Lists out all places in SOC.

Format : `list`

Steps for list command:

1. Type in `list` and press ENTER.
2. Alternatively, type the index of list command, `4` and press ENTER.



```
list
Room:                                               | Location:
------------------------------------------
Active Learning Lab                                 | COM1 B1-03 (COM1-B103)
------------------------------------------
Auditorium                                          | I3 Aud
------------------------------------------
CHI Executive Training Room                         | I3 03-40
------------------------------------------
Cerebro@SoC                                         | COM1 02-05
------------------------------------------
```

Figure 3.3.3.1. An example of list command

3.3.4. Deleting a place in the list: delete-place

Deletes an existing place list of places.

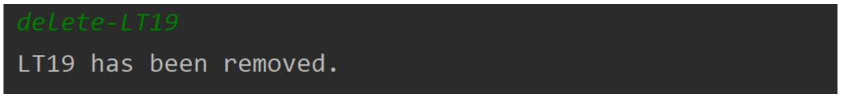
Format : `delete-place`

Format details:

- `place` is the name of the place you want to delete

Steps for deleting a place:

1. Type in the command in the format specified above. Eg. `delete-LT19`
2. Press ENTER.
3. Alternatively, you can input `delete` or the index of the delete command, `3`.
4. Press ENTER.
5. The system will then prompt you to enter the name of the room you wish to delete.



```
delete-LT19
LT19 has been removed.
```

Figure 3.3.3.4.1. An example of deleting a place

3.3.5. Undo previous places command: undo

Undo the previous places command.

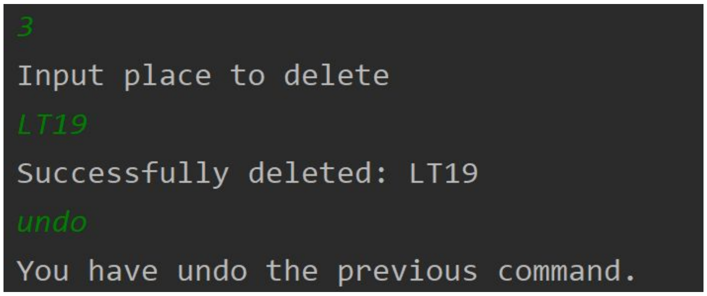
Commands that can be undone:

- add
- delete

Format : `undo`

Steps for undo command:

1. Type in `undo` and press ENTER.
2. Alternatively, type the index of undo command, `5` and press ENTER.



```
3
Input place to delete
LT19
Successfully deleted: LT19
undo
You have undo the previous command.
```

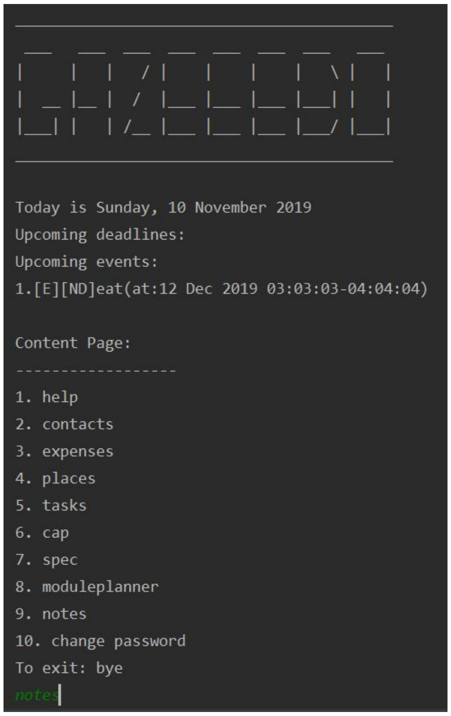
Figure 3.3.5.1. An example of undoing a delete command

3.4. Note Page: notes

Brings you to your note page from the main menu page.

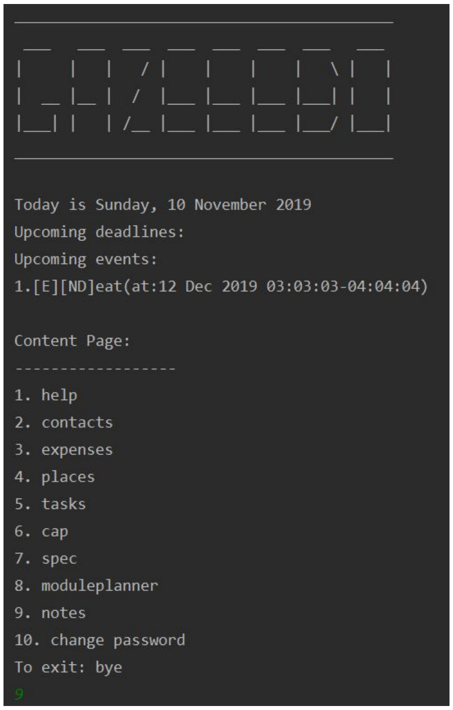
In your note page, you can specify your goal and also record information about your modules. To see what kind of information can be recorded about your modules, see section 3.4.5. below.

Note: The following commands in this section (ie. Section 3.4), except the commands in Section 3.4.5.1 - 3.4.5.10, can only be used in the note page. The commands in Section 3.4.5.1 - 3.4.5.10 can only be used in the module page as explained in Section 3.4.5. below.



```
________________________________________
 ___   ___ ___ ___ ___ ___ ___   ___
|    |  |   /|   |   |   |  \|   |
|  __|__|  / |___|___|___|___|   |
|___||  |/__ |___|___|___|___/|___|
________________________________________

Today is Sunday, 10 November 2019
Upcoming deadlines:
Upcoming events:
1.[E][ND]eat(at:12 Dec 2019 03:03:03-04:04:04)

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
notes
```

Figure 3.4.1 How to go to your note page: method 1



```
__________________________________________
 ___   ___  ___  ___  ___  ___  ___   ___
|    |   |   /|    |    |    |   \|   |
|  _ |_  |  / |___ |___ |___ |___||   |
|___||   |/__ |___ |___ |___ |___/ |___|
__________________________________________

Today is Sunday, 10 November 2019
Upcoming deadlines:
Upcoming events:
1.[E][ND]eat(at:12 Dec 2019 03:03:03-04:04:04)

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
9
```

Figure 3.4.2 How to go to your note page: method 2

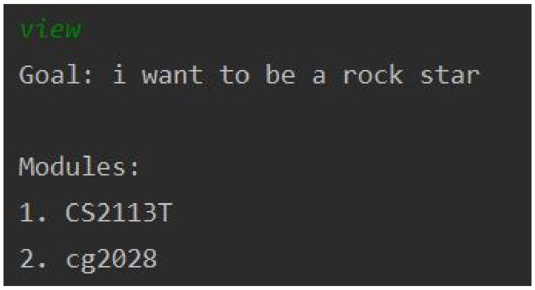
How to go to your note page:

- Type in the command `notes` as shown in Figure 3.4.1 or `9` as shown in Figure 3.4.2 above in the main menu page and press.
- You can only go to the note page from the main menu.

3.4.1. <u>View goal and list of modules: view</u>

Lists out your goal and the list of modules you have added as shown in Figure 3.4.1.1.

Format: `view`



```
view
Goal: i want to be a rock star

Modules:
1. CS2113T
2. cg2028
```

Figure 3.4.1.1 Viewing your goal and your list of modules

3.4.2. Edit goal: edit /n

Edits your existing goal to the new goal you specify as shown in Figure 3.4.2.1.

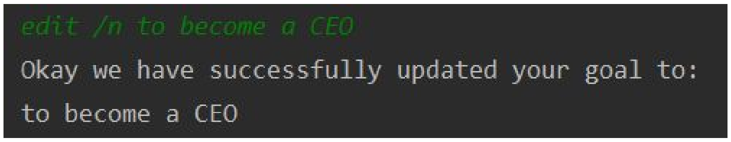
Format: `edit /n NEW_GOAL`

Format details:

- The white space between `/n` and `NEW_GOAL` is optional.

Examples:

- `edit /n i want to be a rock star`
- `edit /nto become a CEO`



```
edit /n to become a CEO
Okay we have successfully updated your goal to:
to become a CEO
```

Figure 3.4.2.1 Editing your goal

3.4.3. Add a module: add /n

Adds a module to your note page as shown in Figure 3.4.3.1.

Format: `add /n MODULE_NAME`

Format details:

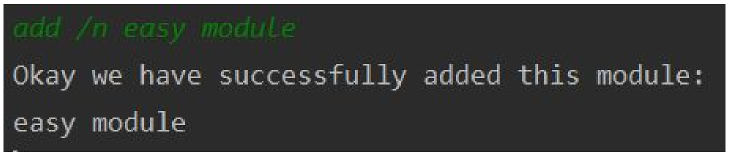
- The white space between `/n` and `MODULE_NAME` is optional.

Constraints:

- You cannot add a module with the same name as an existing module.
- You have the freedom to name your module whatever you want ie. the `MODULE_NAME` does not need to be what it is called by NUS.

Examples:

- `add /n annoying module`
- `add /n cs1231`
- `add /nSoftware Engineering & Object-Oriented Programming`



```
add /n easy module
Okay we have successfully added this module:
easy module
```

Figure 3.4.3.1 Adding a module

3.4.4. Delete a module: delete /n

Deletes a module from your note page as shown in Figure 3.4.4.1.

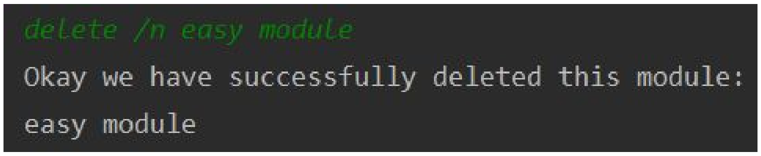
Format: `delete /n MODULE_NAME`

Format details:

- The white space between `/n` and `MODULE_NAME` is optional.

Examples:

- `delete /n annoying module`
- `delete /n cs1231`
- `delete /nSoftware Engineering & Object-Oriented Programming`

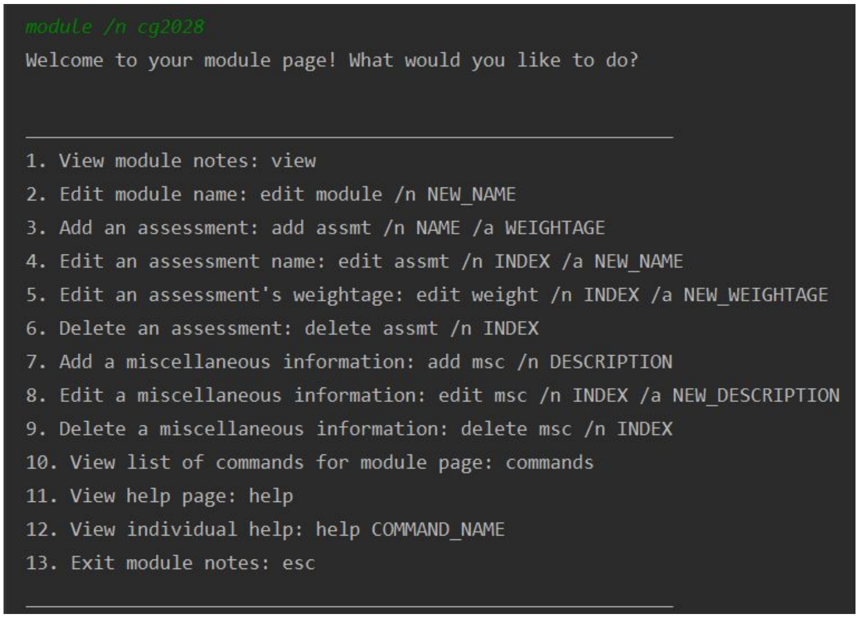


```
delete /n easy module
Okay we have successfully deleted this module:
easy module
```

Figure 3.4.4.1 Deleting a module

3.4.5. Go to a module page: module /n

Brings you to a module page as shown in Figure 3.4.5.1 below, which contains the notes for the module you specify. The notes for the module are grouped into 2 categories, namely assessments and miscellaneous.



```
module /n cg2028
Welcome to your module page! What would you like to do?


__________________________________________________________
1. View module notes: view
2. Edit module name: edit module /n NEW_NAME
3. Add an assessment: add assmt /n NAME /a WEIGHTAGE
4. Edit an assessment name: edit assmt /n INDEX /a NEW_NAME
5. Edit an assessment's weightage: edit weight /n INDEX /a NEW_WEIGHTAGE
6. Delete an assessment: delete assmt /n INDEX
7. Add a miscellaneous information: add msc /n DESCRIPTION
8. Edit a miscellaneous information: edit msc /n INDEX /a NEW_DESCRIPTION
9. Delete a miscellaneous information: delete msc /n INDEX
10. View list of commands for module page: commands
11. View help page: help
12. View individual help: help COMMAND_NAME
13. Exit module notes: esc

__________________________________________________________
```

Figure 3.4.5.1 Going to a module page

Note: The following commands in Section 3.4.5.1 - 3.4.5.10 can only be used after you go to the module page.

Format: `module /n MODULE_NAME`

Format details:

- The white space between `/n` and `MODULE_NAME` is optional.

Constraints:

- You can only go to a module page from the note page.
  - ie. You have to first go to your note page and then type in `module /n MODULE_NAME` in your note page to be able to go to a module page.

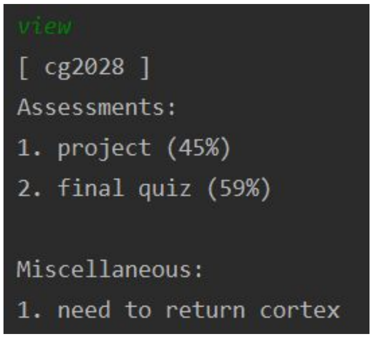
Examples:

- `module /n annoying module`
- `module /n cs1231`
- `module /nSoftware Engineering & Object-Oriented Programming`

3.4.5.1. View the module's notes: view

Shows you the notes you have for the module as shown in Figure 3.4.5.1.1 below.

Format: `view`



```
view
[ cg2028 ]
Assessments:
1. project (45%)
2. final quiz (59%)

Miscellaneous:
1. need to return cortex
```

Figure 3.4.5.1.1

3.4.5.2. Edit the module's name: edit mod /n

Edits the name of the module as shown in Figure 3.4.5.2.1 below.

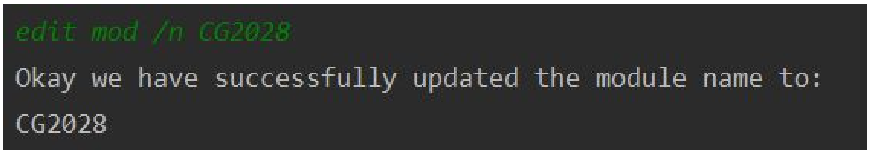
Format: `edit mod /n NEW_NAME`

Format details:

- The white space between `/n` and `NEW_NAME` is optional.

Examples:

- `edit mod /n A+ module`
- `edit mod /n CS1231`
- `edit mod /nSoftware Engineering & Object-Oriented Programming`



```
edit mod /n CG2028
Okay we have successfully updated the module name to:
CG2028
```

Figure 3.4.5.2.1 Editing the module's name

3.4.5.3. Add an assessment: add assmt /n /a

Adds an assessment to the module as shown in Figure 3.4.5.3.1.

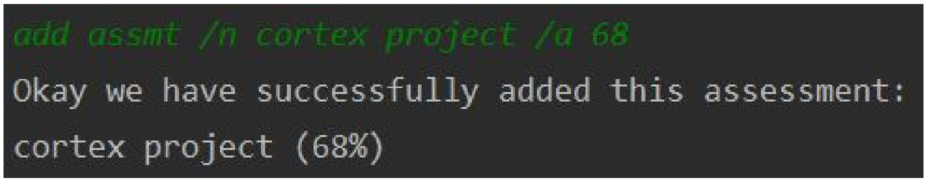
Format: `add assmt /n NAME /a WEIGHTAGE`

Format details:

- The white space between `/n` and `NAME` is optional.
- The white space between `/a` and `WEIGHTAGE` is optional.

Examples:

- `add assmt /n group project /a 47`
- `add assmt /npresentation /a20`



```
add assmt /n cortex project /a 68
Okay we have successfully added this assessment:
cortex project (68%)
```

Figure 3.4.5.3.1 Adding an assessment

3.4.5.4. Edit an assessment's name: edit assmt /n /a

Edits the name of the specified assessment as shown in Figure 3.4.5.4.1 below.

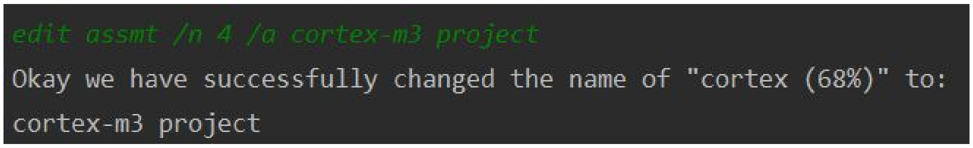
Format: `edit assmt /n INDEX /a NEW_NAME`

Format details:

- The white space between `/n` and `INDEX` is optional.
- The white space between `/a` and `NEW_NAME` is optional.
- `INDEX` refers to the index of the assessment in the list of assessments when you use the `view` command.

Examples:

- `edit assmt /n 4 /a project`
- `edit assmt /n2 /afinal exam`



```
edit assmt /n 4 /a cortex-m3 project
Okay we have successfully changed the name of "cortex (68%)" to:
cortex-m3 project
```

Figure 3.4.5.4.1 Editing an assessment's name

3.4.5.5. Edit an assessment's weightage: edit weight /n /a

Edits the weightage of the specified assessment as shown in Figure 3.4.5.5.1 below.

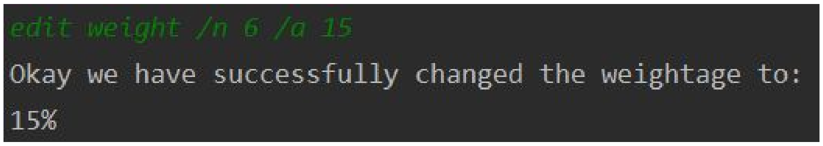
Format: `edit weight /n INDEX /a NEW_WEIGHTAGE`

Format details:

- The white space between `/n` and `INDEX` is optional.
- The white space between `/a` and `WEIGHTAGE` is optional.
- `INDEX` refers to the index of the assessment in the list of assessments when you use the `view` command.
- `WEIGHTAGE` is in percentage eg. if you specify it as 30, it will be 30%

Examples:

- `edit weight /n 4 /a 30`
- `edit weight /n2 /a49`



```
edit weight /n 6 /a 15
Okay we have successfully changed the weightage to:
15%
```

Figure 3.4.5.5.1 Editing an assessment's weightage

3.4.5.6. Delete an assessment: delete assmt /n

Deletes the specified assessment as shown in Figure 3.4.5.6.1.

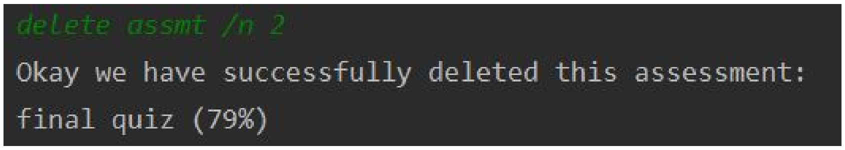
Format: `delete assmt /n INDEX`

Format details:

- The white space between `/n` and `INDEX` is optional.
- `INDEX` refers to the index of the assessment in the list of assessments when you use the `view` command.

Examples:

- `delete assmt /n 4`
- `delete assmt /n2`



```
delete assmt /n 2
Okay we have successfully deleted this assessment:
final quiz (79%)
```

Figure 3.4.5.6.1 Deleting an assessment

3.4.5.7. Add a miscellaneous information: add msc /n

Adds a miscellaneous information to the module as shown in Figure 3.4.5.7.1 below.

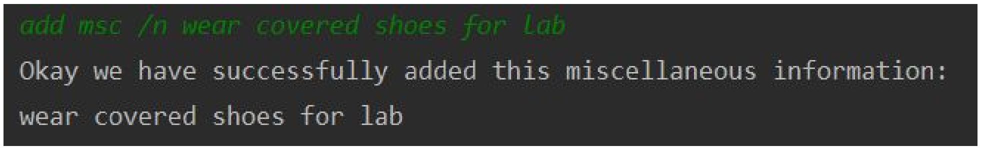
Format: `add msc /n DESCRIPTION`

Format details:

- The white space between `/n` and `DESCRIPTION` is optional.

Examples:

- `add msc /n a heavy module`
- `add msc /nwear covered shoes for lab`



```
add msc /n wear covered shoes for lab
Okay we have successfully added this miscellaneous information:
wear covered shoes for lab
```

Figure 3.4.5.7.1 Adding a miscellaneous information

3.4.5.8. Edit a miscellaneous information: edit msc /n /a

Edits the specified miscellaneous information as shown in Figure 3.4.5.8.1 below.

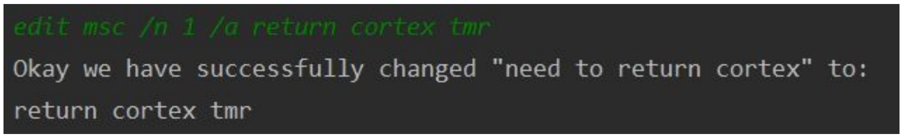
Format: `edit msc /n INDEX /a NEW_DESCRIPTION`

Format details:

- The white space between `/n` and `INDEX` is optional.
- The white space between `/a` and `NEW_DESCRIPTION` is optional.
- `INDEX` refers to the index of the assessment in the list of miscellaneous information when you use the `view` command.

Examples:

- `edit msc /n 4 /a no webcast for this mod`
- `edit msc /n2 /alecturer starts lecture exactly at 6pm`



```
edit msc /n 1 /a return cortex tmr
Okay we have successfully changed "need to return cortex" to:
return cortex tmr
```

Figure 3.4.5.8.1 Editing a miscellaneous information

3.4.5.9. Delete a miscellaneous information: delete msc /n

Deletes the specified miscellaneous information as shown in Figure 3.4.5.9.1 below.

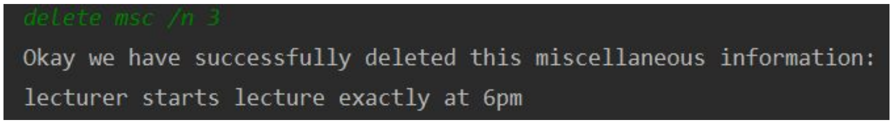
Format: `delete msc /n INDEX`

Format details:

- The white space between `/n` and `INDEX` is optional.
- `INDEX` refers to the index of the assessment in the list of miscellaneous information when you use the `view` command.

Examples:

- `delete msc /n 4`
- `delete msc /n2`



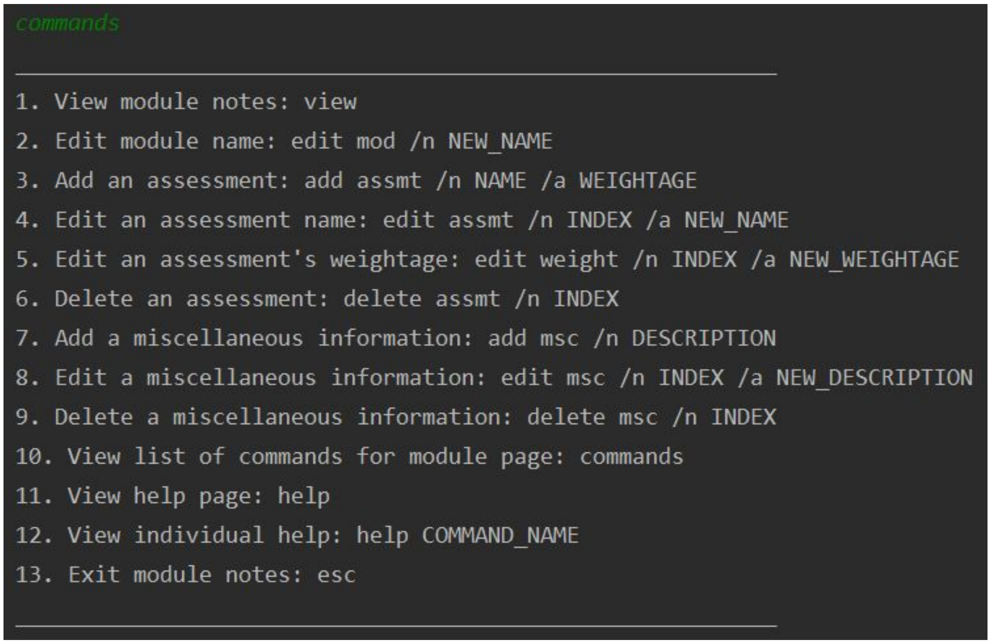
```
delete msc /n 3
Okay we have successfully deleted this miscellaneous information:
lecturer starts lecture exactly at 6pm
```

Figure 3.4.5.9.1 Deleting a miscellaneous information

3.4.5.10. View the list of possible commands for the module page: commands

Shows you the possible things you can do on the module page as shown in Figure 3.4.5.10.1.

Format: `commands`



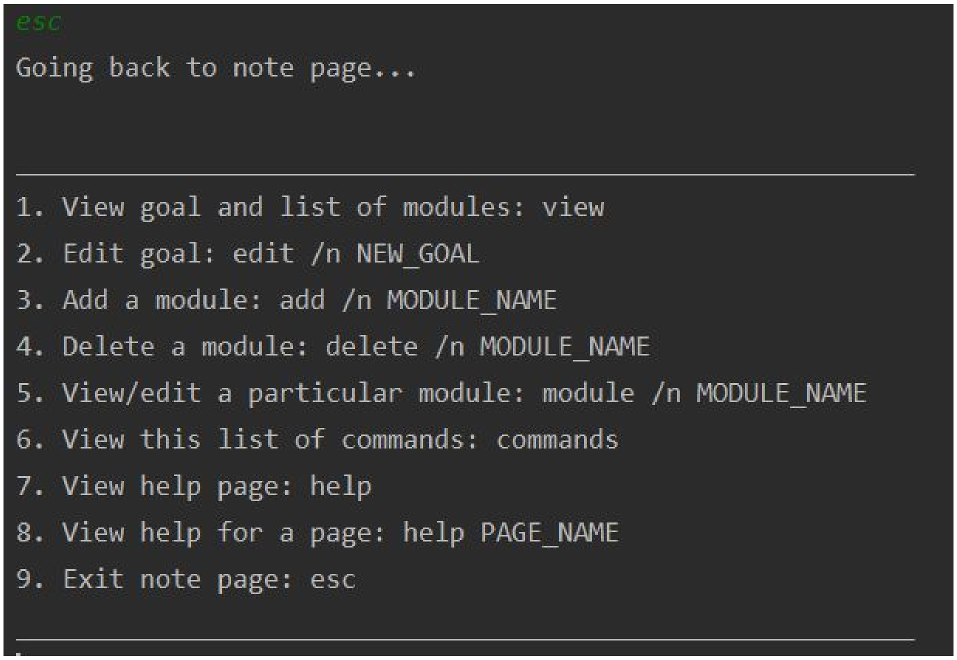
```
commands
__________________________________________________________
1. View module notes: view
2. Edit module name: edit mod /n NEW_NAME
3. Add an assessment: add assmt /n NAME /a WEIGHTAGE
4. Edit an assessment name: edit assmt /n INDEX /a NEW_NAME
5. Edit an assessment's weightage: edit weight /n INDEX /a NEW_WEIGHTAGE
6. Delete an assessment: delete assmt /n INDEX
7. Add a miscellaneous information: add msc /n DESCRIPTION
8. Edit a miscellaneous information: edit msc /n INDEX /a NEW_DESCRIPTION
9. Delete a miscellaneous information: delete msc /n INDEX
10. View list of commands for module page: commands
11. View help page: help
12. View individual help: help COMMAND_NAME
13. Exit module notes: esc

__________________________________________________________
```

Figure 3.4.5.10.1 Viewing list of possible commands in a module page

3.4.5.11. Go back to the note page: esc

Brings you back to the note page from the module page as shown in Figure 3.4.5.11.1 below.

Format: `esc`



```
esc
Going back to note page...

____________________________________________________________
1. View goal and list of modules: view
2. Edit goal: edit /n NEW_GOAL
3. Add a module: add /n MODULE_NAME
4. Delete a module: delete /n MODULE_NAME
5. View/edit a particular module: module /n MODULE_NAME
6. View this list of commands: commands
7. View help page: help
8. View help for a page: help PAGE_NAME
9. Exit note page: esc

____________________________________________________________
```

Figure 3.4.5.11.1 Going back to note page

3.4.5.12. Find an assessment: find [coming in v2.0]

Finds all assessment matching the keyword.

Format: `find assmt /n KEYWORD`

Format details:

- The white space between `/n` and `KEYWORD` is optional.

Examples:

- `find assmt /n project`
- `find assmt /nexam`

3.4.5.13. Find a miscellaneous information: find [coming in v2.0]

Finds all miscellaneous information matching the keyword.

Format: `find msc/n KEYWORD`

Format details:

- The white space between `/n` and `KEYWORD` is optional.

Examples:

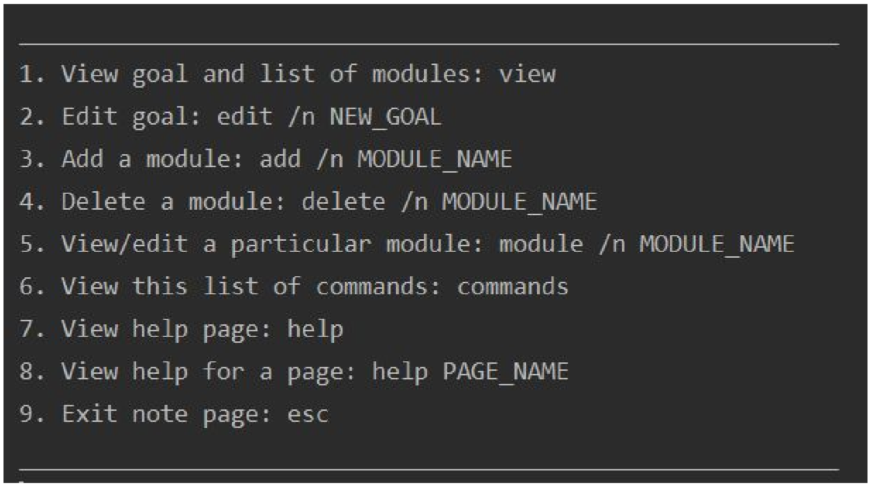
- `find msc /n jacket`

- `find msc /nmosquito`

3.4.6. View the list of possible commands for the notes page: commands

Shows you the possible things you can do in the notes page.

Format: `commands`



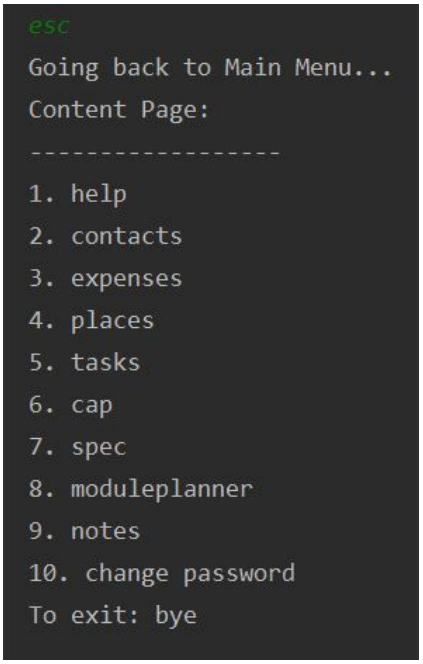
```
____________________________________________________________
1. View goal and list of modules: view
2. Edit goal: edit /n NEW_GOAL
3. Add a module: add /n MODULE_NAME
4. Delete a module: delete /n MODULE_NAME
5. View/edit a particular module: module /n MODULE_NAME
6. View this list of commands: commands
7. View help page: help
8. View help for a page: help PAGE_NAME
9. Exit note page: esc
____________________________________________________________
```

Figure 3.4.6.1 Viewing list of possible commands in the note page

3.4.7. Go back to the main page: esc

Brings you back to the main page from the note page as shown in Figure 3.4.7.1 below.

Format: `esc`



```
esc
Going back to Main Menu...
Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
```

Figure 3.4.7.1 Going back to main page

3.4.8. Find a module: find [coming in v2.0]

Finds all modules matching the keyword.

Format: `find /n KEYWORD`

Format details:

- The white space between `/n` and `KEYWORD` is optional.

Examples:

- `find /n cs`
- `find /ncg2028`

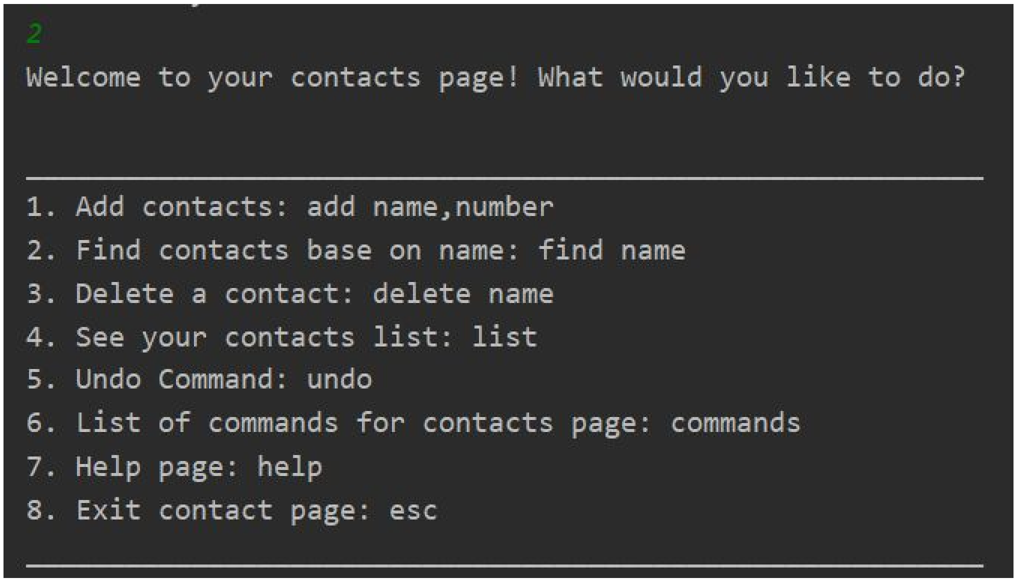
3.5. Contacts Page

Function to store contacts.
Enter the contacts page.

1st way:

Step 1: `2`

Example:



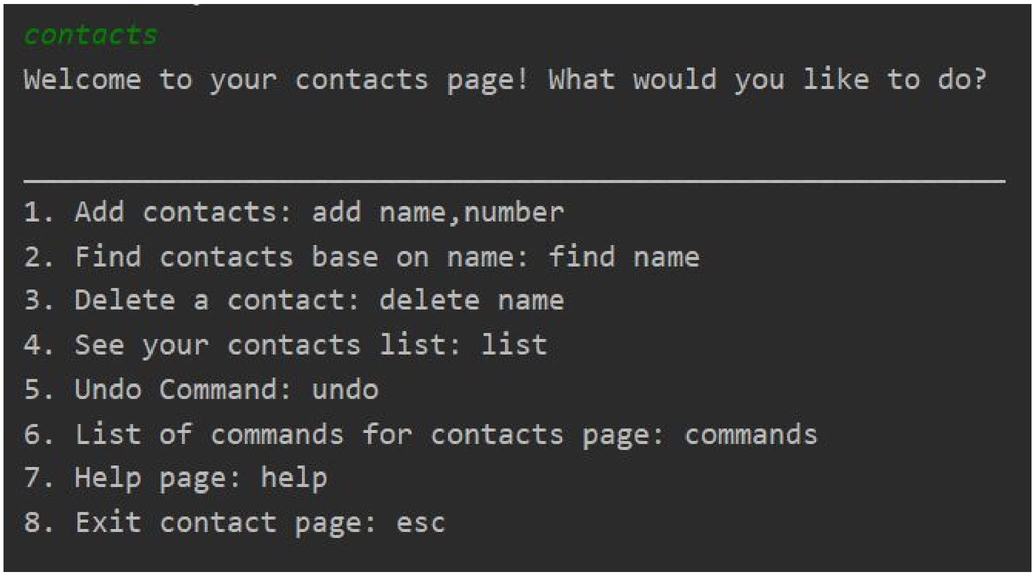
```
2
Welcome to your contacts page! What would you like to do?

__________________________________________________________
1. Add contacts: add name,number
2. Find contacts base on name: find name
3. Delete a contact: delete name
4. See your contacts list: list
5. Undo Command: undo
6. List of commands for contacts page: commands
7. Help page: help
8. Exit contact page: esc
__________________________________________________________
```

Figure 3.5.1. Input index to start up contacts page

2nd way:

Step 1: `contacts`

Example:



```
contacts
Welcome to your contacts page! What would you like to do?

__________________________________________________________
1. Add contacts: add name,number
2. Find contacts base on name: find name
3. Delete a contact: delete name
4. See your contacts list: list
5. Undo Command: undo
6. List of commands for contacts page: commands
7. Help page: help
8. Exit contact page: esc
```

Figure 3.5.2. Input `contacts` to enter contacts page.

3.5.1. Adding a new contact name and number

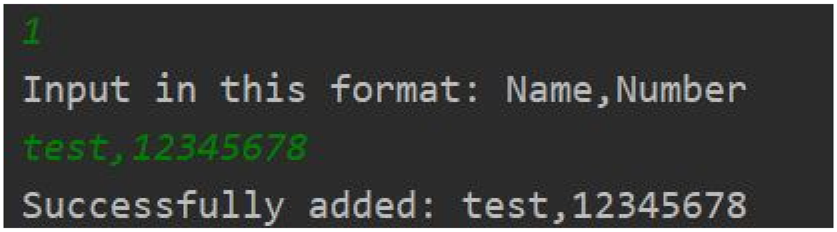
Call out the add feature and input the number. There are 3 ways of calling out add feature:

1st way:

Step 1: `1`

Step 2: `NAME, NUMBER`

Example:



```
1
Input in this format: Name,Number
test,12345678
Successfully added: test,12345678
```

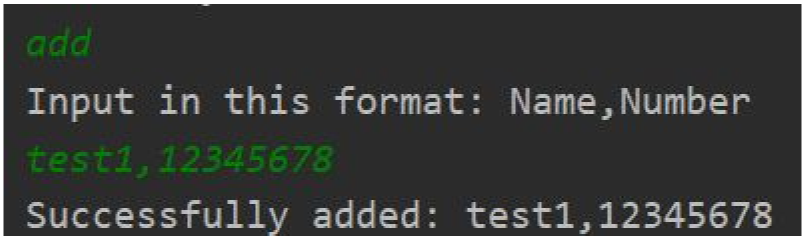
Figure 3.5.1.1. 2-step index process to call add.

2nd way:

Step 1: `add`

Step 2: `NAME, NUMBER`

Example:



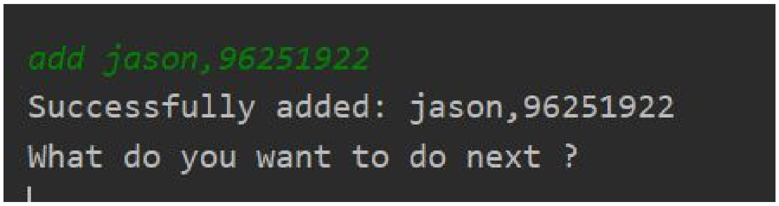
```
add
Input in this format: Name,Number
test1,12345678
Successfully added: test1,12345678
```

Figure 3.5.1.2. 2-step input `add` process to call add.

3rd way:

Step 1: `add NAME,NUMBER`

Example:



```
add jason,96251922
Successfully added: jason,96251922
What do you want to do next ?
```

Figure 3.5.1.3 1- step process to call add.

3.5.2. Finding a contact

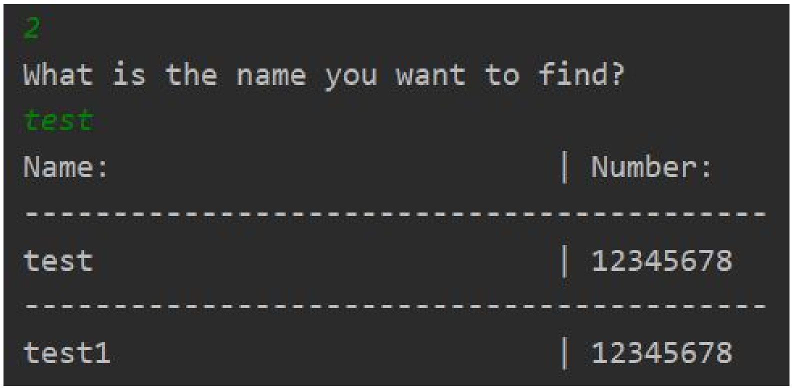
Find contact names that have the keyword. There are 3 ways of calling out find feature:

1st way:

Step 1: `2`

Step 2: `NAME`

Example:



```
2
What is the name you want to find?
test
Name:                           | Number:
------------------------------------------
test                            | 12345678
------------------------------------------
test1                           | 12345678
```

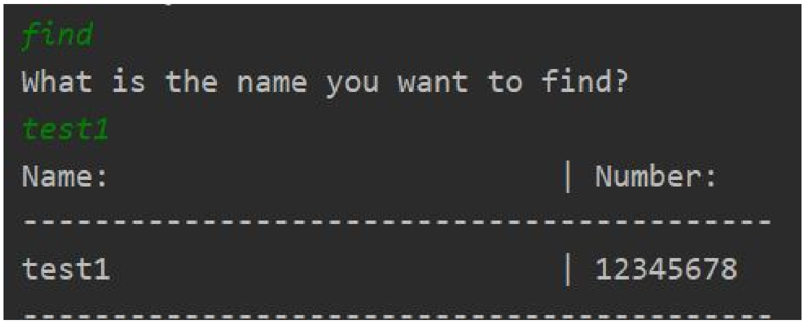
Figure 3.5.2.1. 2-step index process to call find

2nd way:

Step 1: `find`

Step 2: `NAME`

Example:



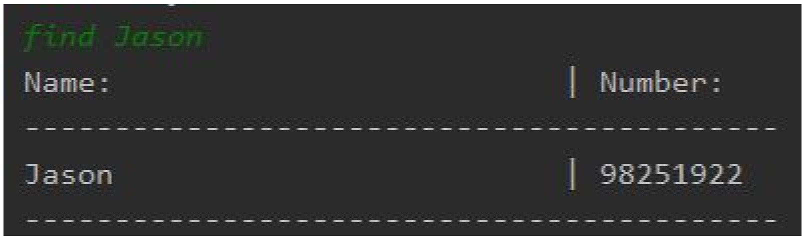
```
find
What is the name you want to find?
test1
Name:                           | Number:
------------------------------------------
test1                           | 12345678
------------------------------------------
```

Figure 3.5.2.2. 2-step input find process to call find.

3rd way:

Step 1: `find NAME`

Example:



```
find Jason
Name:                         | Number:
-----------------------------------------
Jason                         | 98251922
-----------------------------------------
```

Figure 3.5.2.3. 1-step input to call find.

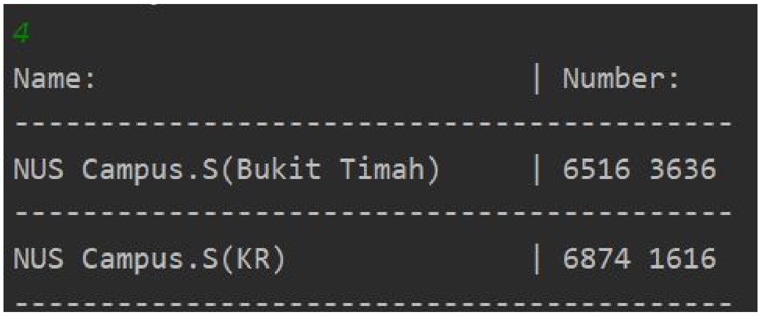
3.5.3. Listing all contacts

Lists all contacts. There are 2 ways of calling out list feature:

1st way:

Step 1: `4`

Example:



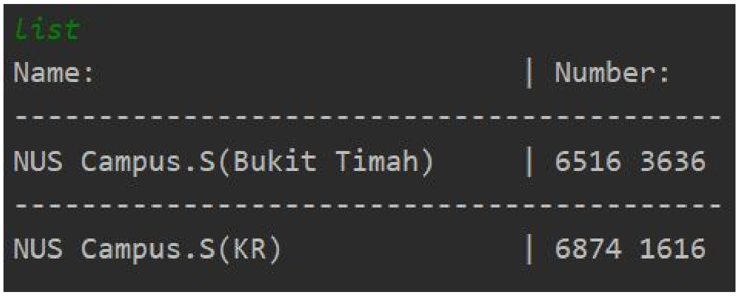
```
4
Name:                         | Number:
-----------------------------------------
NUS Campus.S(Bukit Timah)     | 6516 3636
-----------------------------------------
NUS Campus.S(KR)              | 6874 1616
-----------------------------------------
```

Figure 3.5.3.1.Call list by index

2nd way:

Step 1: `list`

Example:



```
list
Name:                         | Number:
-----------------------------------------
NUS Campus.S(Bukit Timah)     | 6516 3636
-----------------------------------------
NUS Campus.S(KR)              | 6874 1616
```

Figure 3.5.3.2. Call list by input `list`*.*

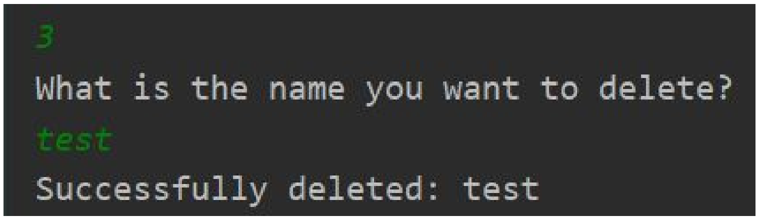
3.5.4. Deleting a contact in the list

Deletes an existing contact in the list. There are 3 ways of calling out delete feature:

1st way:

Step 1: `3`
Step 2: `NAME`

Example:



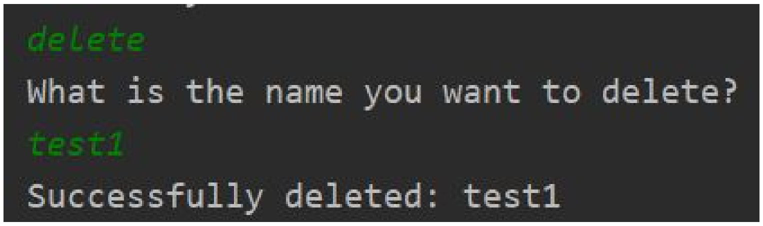
```
3
What is the name you want to delete?
test
Successfully deleted: test
```

Figure 3.5.4.1. 2-step index input to call delete.

2nd way:

Step 1: `delete`
Step 2: `NAME`

Example:



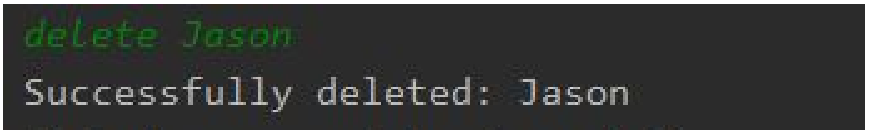
```
delete
What is the name you want to delete?
test1
Successfully deleted: test1
```

Figure 3.5.4.2. 2-step input delete to call delete.

3rd way:

Step 1: `delete NAME`

Example:



```
delete Jason
Successfully deleted: Jason
```

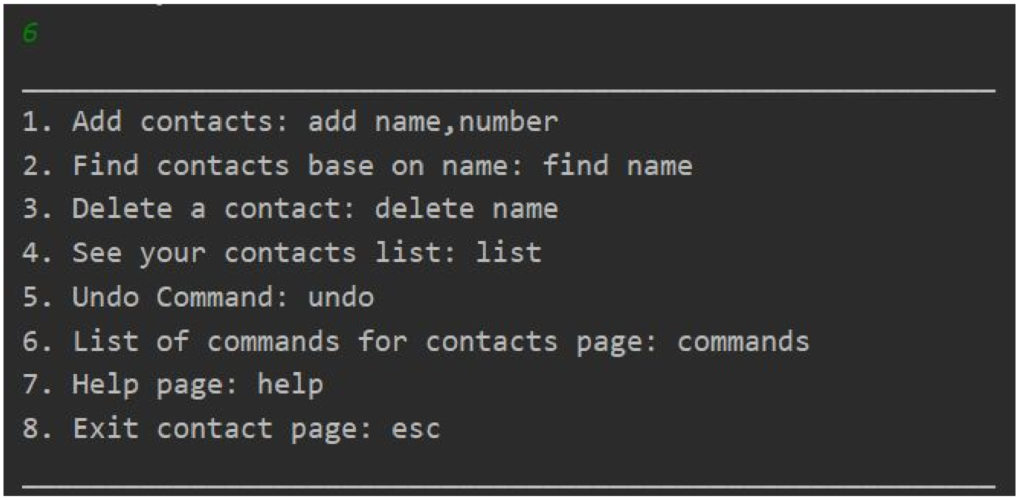
Figure 3.5.4.3 1-step input to call delete.

3.5.5..View commands in Contact page

View the commands in contact page. There are 2 ways of viewing the commands.

1st way:

Step 1: `6`
Example:



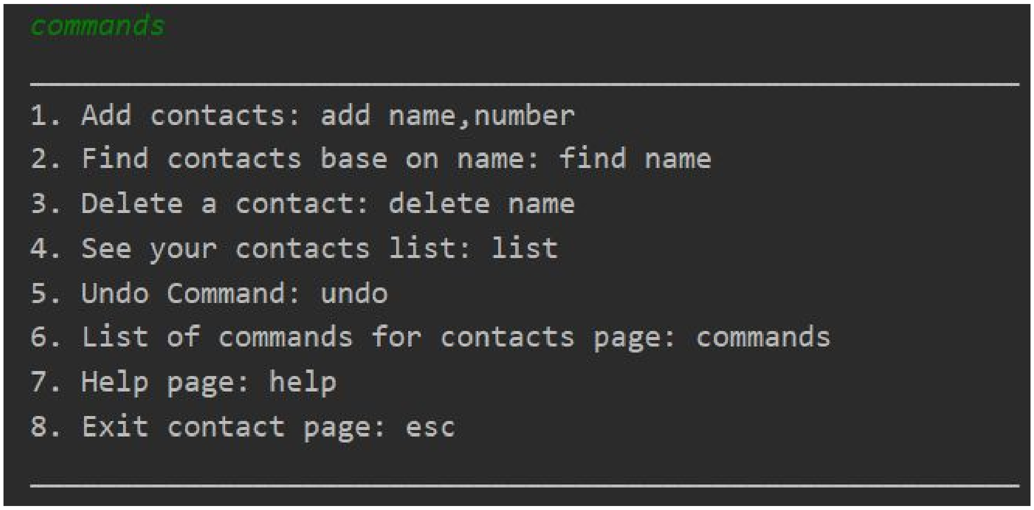
```
6

______________________________________________________________
1. Add contacts: add name,number
2. Find contacts base on name: find name
3. Delete a contact: delete name
4. See your contacts list: list
5. Undo Command: undo
6. List of commands for contacts page: commands
7. Help page: help
8. Exit contact page: esc

______________________________________________________________
```

Figure 3.7.5.1 Input index to view contact's commands.

2nd way:

Step 1: `commands`

Example:



```
commands

______________________________________________________________
1. Add contacts: add name,number
2. Find contacts base on name: find name
3. Delete a contact: delete name
4. See your contacts list: list
5. Undo Command: undo
6. List of commands for contacts page: commands
7. Help page: help
8. Exit contact page: esc

______________________________________________________________
```

Figure 3.7.5.2 Input `commands` to view contact page commands.

3.5.6. Undo previous contact command: undo

Undo the previous contact command. There are 2 ways of calling out add feature:

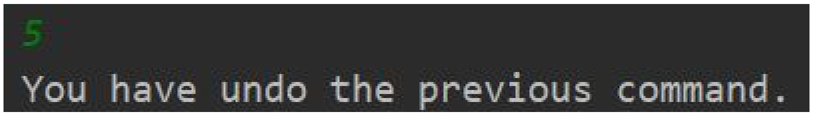
Commands that can be undone:

- add
- delete

1st way:

Step 1: `5`

Example:



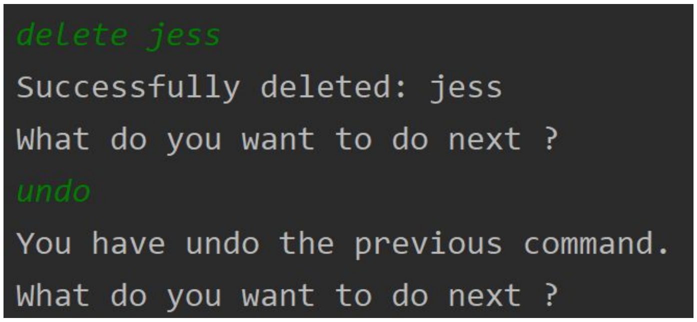
```
5
You have undo the previous command.
```

Figure 3.5.6.1. Input index to undo command.

2nd way:

Step 1: `undo`

Example:



```
delete jess
Successfully deleted: jess
What do you want to do next ?
undo
You have undo the previous command.
What do you want to do next ?
```

Figure 3.5.6.2. An example of undoing a delete command

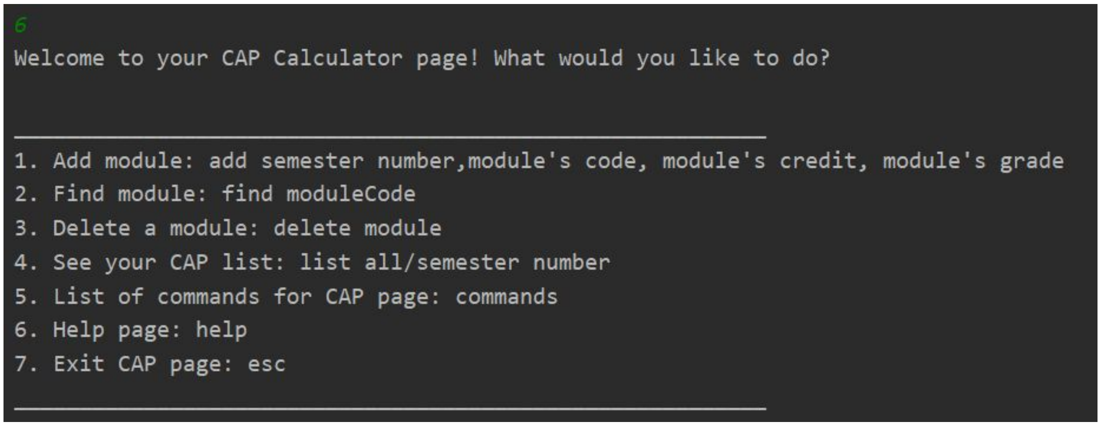
3.6. <u>CAP Page</u>

A function to calculate the CAP of modules and to store module’s sem number, code, credit and grade.

1st way:

Step 1: `6`

Example:



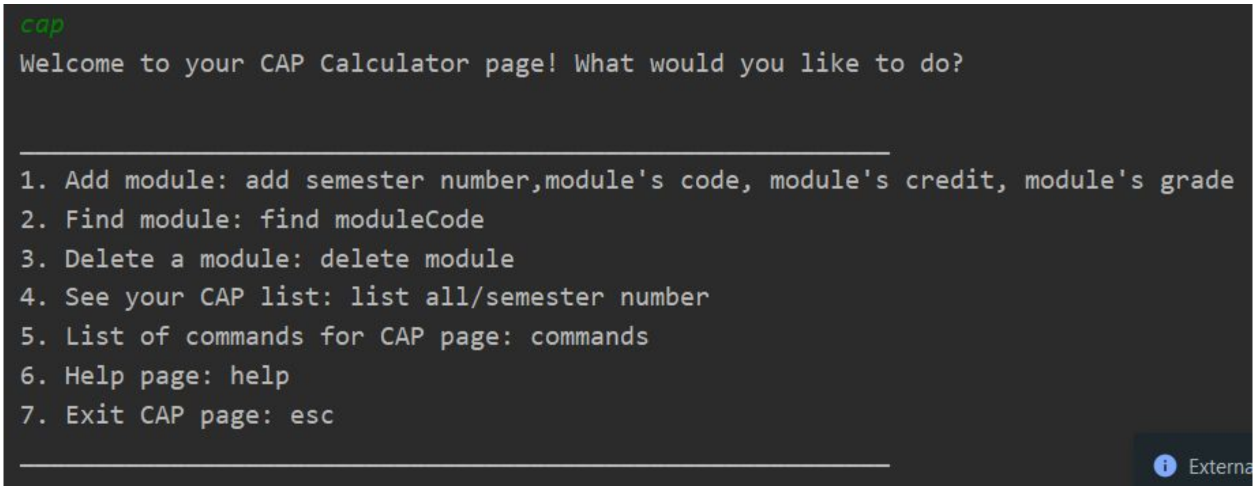
```
6
Welcome to your CAP Calculator page! What would you like to do?


__________________________________________________________
1. Add module: add semester number,module's code, module's credit, module's grade
2. Find module: find moduleCode
3. Delete a module: delete module
4. See your CAP list: list all/semester number
5. List of commands for CAP page: commands
6. Help page: help
7. Exit CAP page: esc
__________________________________________________________
```

Figure 3.6.1. Using index to enter CAP page.

2nd way:

Step 1: `cap`

Example:



```
cap
Welcome to your CAP Calculator page! What would you like to do?


__________________________________________________________
1. Add module: add semester number,module's code, module's credit, module's grade
2. Find module: find moduleCode
3. Delete a module: delete module
4. See your CAP list: list all/semester number
5. List of commands for CAP page: commands
6. Help page: help
7. Exit CAP page: esc
__________________________________________________________
```

Figure 3.6.2. Using command `cap` to enter CAP page.

3.6.1. <u>Adding a new module's sem number, code, credit, and grade</u>

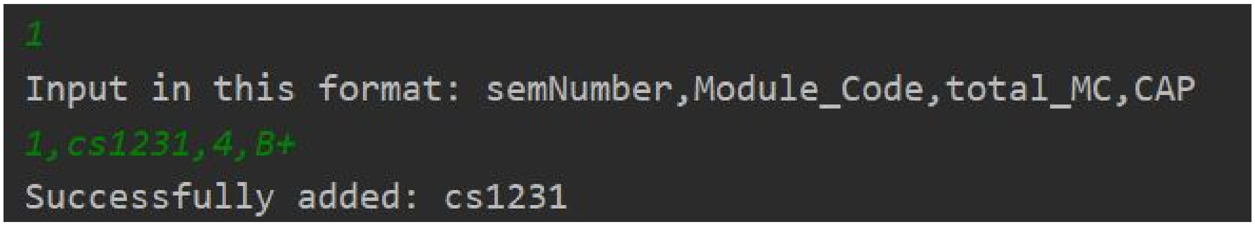
Call out the add feature and input the required module information. There are 3 ways of calling out add feature:

1st way:

Step 1: `1`

Step 2: `SEMESTER_NUMBER,MODULE_CODE,MODULAR_CREDIT, MODULE_GRADE`

Example:



```
1
Input in this format: semNumber,Module_Code,total_MC,CAP
1,cs1231,4,B+
Successfully added: cs1231
```

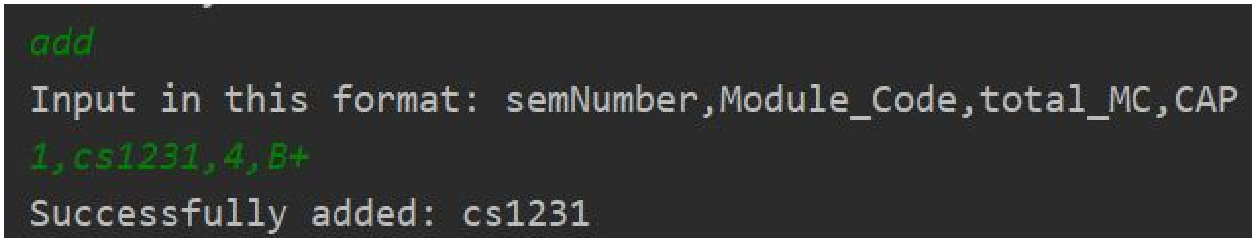
Figure 3.6.1.1. 2-step input index process to call add

2nd way:

Step 1: `add`

Step 2: `SEMESTER_NUMBER,MODULE_CODE,MODULAR_CREDIT, MODULE_GRADE`

Example:



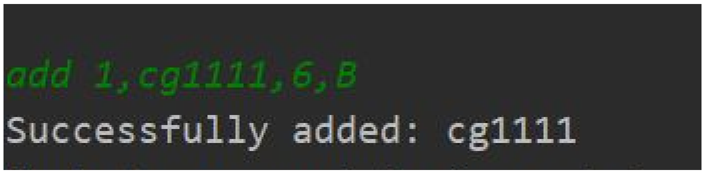
```
add
Input in this format: semNumber,Module_Code,total_MC,CAP
1,cs1231,4,B+
Successfully added: cs1231
```

Figure 3.6.1.2. 2-step input `add` process to call add.

3rd way:

Step 1: `add SEMESTER_NUMBER,MODULE_CODE,MODULAR_CREDIT, MODULE_GRADE`

Example:



```
add 1,cg1111,6,B
Successfully added: cg1111
```

Figure 3.6.1.3. 1-step process to call add

3.6.2. Finding a module(s)

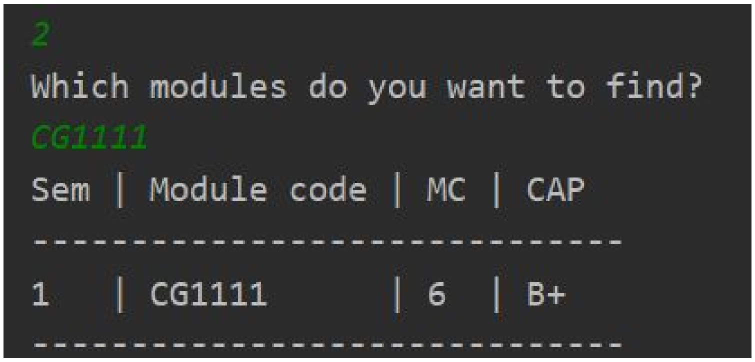
Finds all the modules by module's code. There are 2 ways of calling out the find feature:

1st way:

Step 1: `2`

Step 2: `MODULE_CODE`

Example:



```
2
Which modules do you want to find?
CG1111
Sem | Module code | MC | CAP
------------------------------
1   | CG1111      | 6  | B+
------------------------------
```

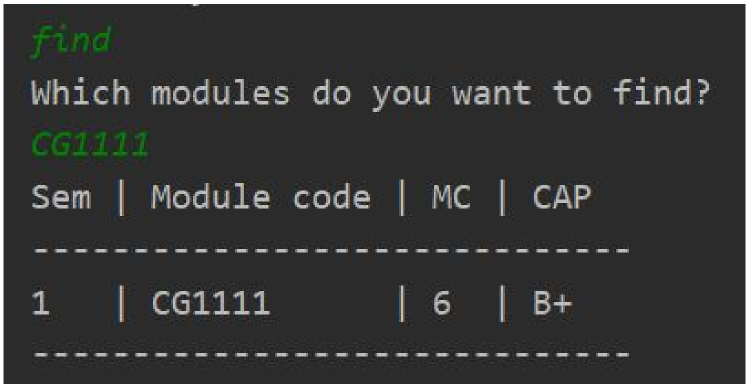
Figure 3.6.2.1. 2-step input index process to call find.

2nd way:

Step 1: `find`

Step 2: `MODULE_CODE`

Example:



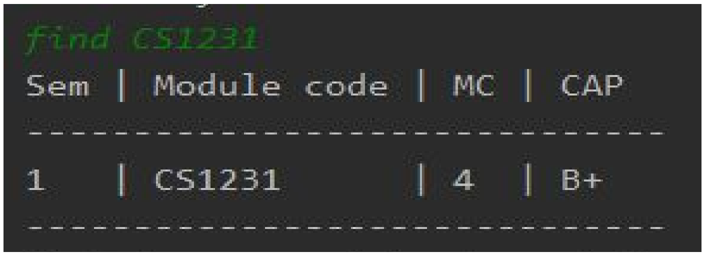
```
find
Which modules do you want to find?
CG1111
Sem | Module code | MC | CAP
--------------------------------
1   | CG1111      | 6  | B+
--------------------------------
```

Figure 3.6.2.2. 2-step input `find` process to call find.

3rd way:

Step 1: `find MODULE_CODE`

Example:



```
find CS1231
Sem | Module code | MC | CAP
--------------------------------
1   | CS1231      | 4  | B+
--------------------------------
```

Figure 3.6.2.3. 1-step input process to call find.

3.6.3. Listing modules

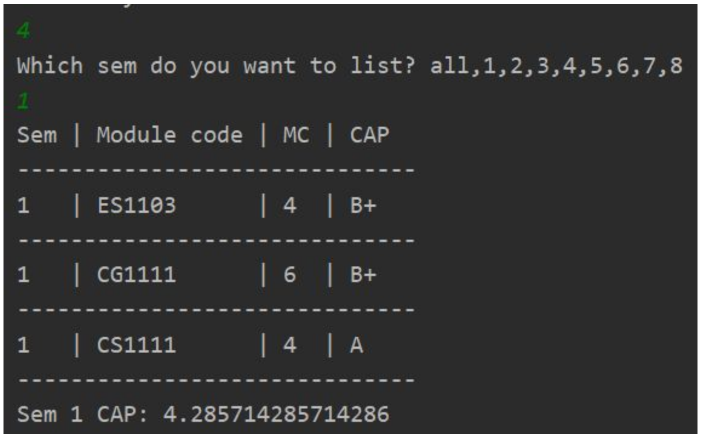
Call out the list feature

1st way:

Step 1: `4`

Step 2: `SEMESTER_NUMBER`

Example:



```
4
Which sem do you want to list? all,1,2,3,4,5,6,7,8
1
Sem | Module code | MC | CAP
------------------------------
1   | ES1103      | 4  | B+
------------------------------
1   | CG1111      | 6  | B+
------------------------------
1   | CS1111      | 4  | A
------------------------------
Sem 1 CAP: 4.285714285714286
```

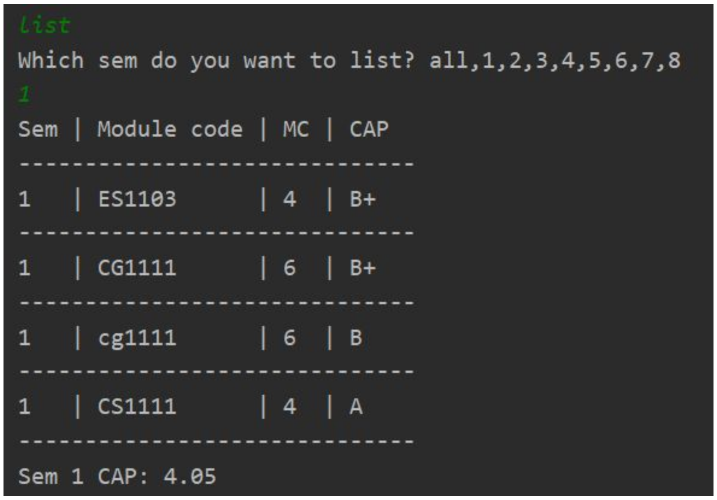
Figure 3.6.3.1. 2-step input index process to call list

2nd way:

Step 1: `list`

Step 2: `SEMESTER_NUMBER`

Example:



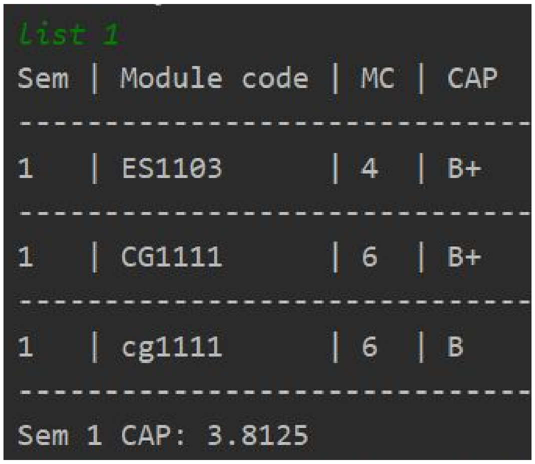
```
list
Which sem do you want to list? all,1,2,3,4,5,6,7,8
1
Sem | Module code | MC | CAP
------------------------------
1   | ES1103      | 4  | B+
------------------------------
1   | CG1111      | 6  | B+
------------------------------
1   | cg1111      | 6  | B
------------------------------
1   | CS1111      | 4  | A
------------------------------
Sem 1 CAP: 4.05
```

Figure 3.6.3.2. 2-step input `list` process to call list.

3rd way:

Step 1: `list SEMESTER_NUMBER`

Example:



```
list 1
Sem | Module code | MC | CAP
------------------------------
1   | ES1103      | 4  | B+
------------------------------
1   | CG1111      | 6  | B+
------------------------------
1   | cg1111      | 6  | B
------------------------------
Sem 1 CAP: 3.8125
```

Figure 3.6.2.3. 1 step input process to call list.

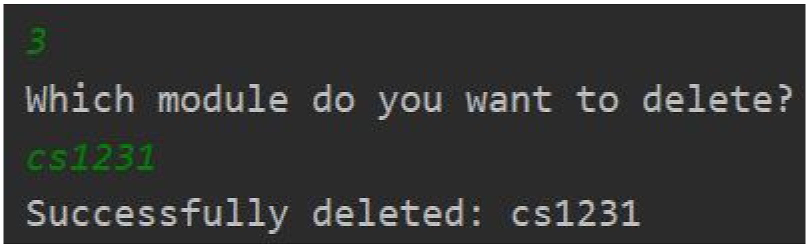
3.6.4. <u>Deleting a module in the list</u>

1st way:

Step 1: `3`

Step 2: `module’s code`

Example:



```
3
Which module do you want to delete?
cs1231
Successfully deleted: cs1231
```

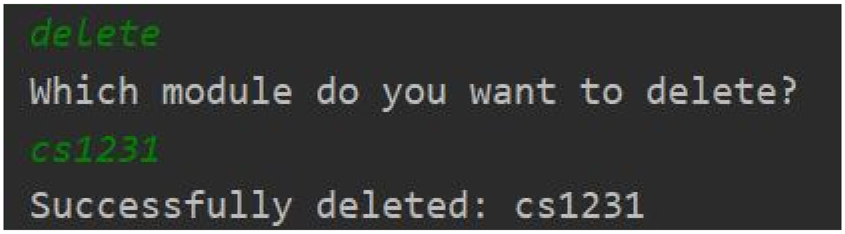
Figure 3.6.4.1. 2 step input index process to call delete.

2nd way:

Step 1: `delete`

Step 2: `module’s code`

Example:



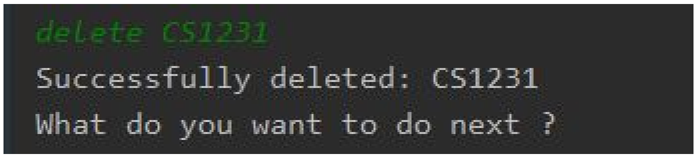
```
delete
Which module do you want to delete?
cs1231
Successfully deleted: cs1231
```

Figure 3.6.4.2. 2 step input `delete` process to call delete.

3rd way:

Step 1: `add semester number,module’s code,module’s credit,module’s grade`

Example:



```
delete CS1231
Successfully deleted: CS1231
What do you want to do next ?
```

Figure 3.6.4.3. 1 step input process to call delete.

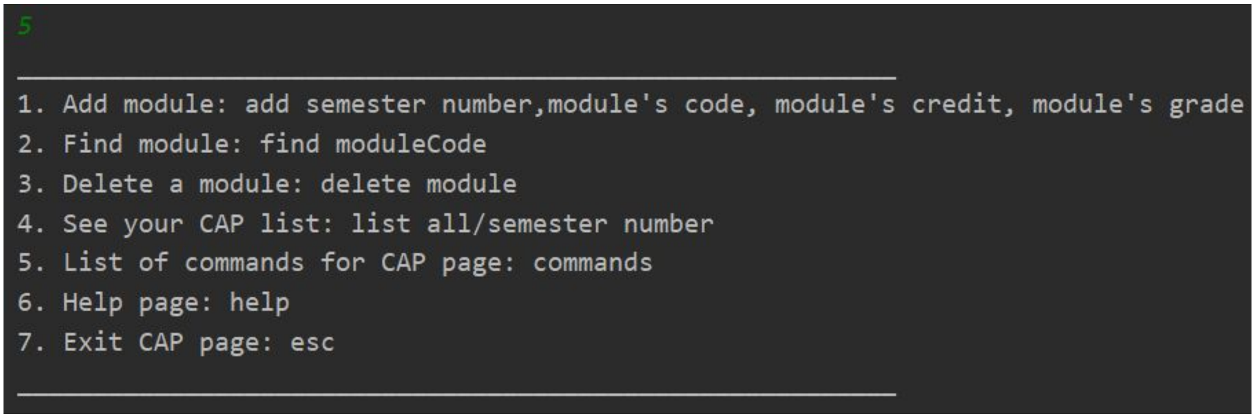
3.6.5. Show commands in CAP Page

Shows what commands can do in CAP page.

1st way:

Step 1: `5`

Example:



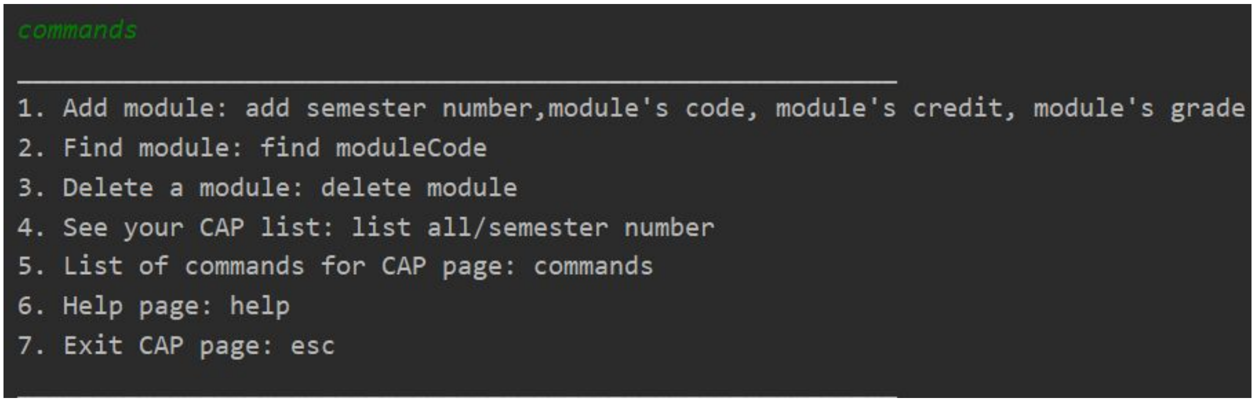
```
5
__________________________________________________________
1. Add module: add semester number,module's code, module's credit, module's grade
2. Find module: find moduleCode
3. Delete a module: delete module
4. See your CAP list: list all/semester number
5. List of commands for CAP page: commands
6. Help page: help
7. Exit CAP page: esc
__________________________________________________________
```

Figure 3.6.5.1. Input index to view CAP page commands.

2nd way:

Step 1: `commands`

Example:



```
commands
____________________________________________________________
1. Add module: add semester number,module's code, module's credit, module's grade
2. Find module: find moduleCode
3. Delete a module: delete module
4. See your CAP list: list all/semester number
5. List of commands for CAP page: commands
6. Help page: help
7. Exit CAP page: esc
____________________________________________________________
```

Figure 3.6.5.2. Input commands to view CAP page commands.

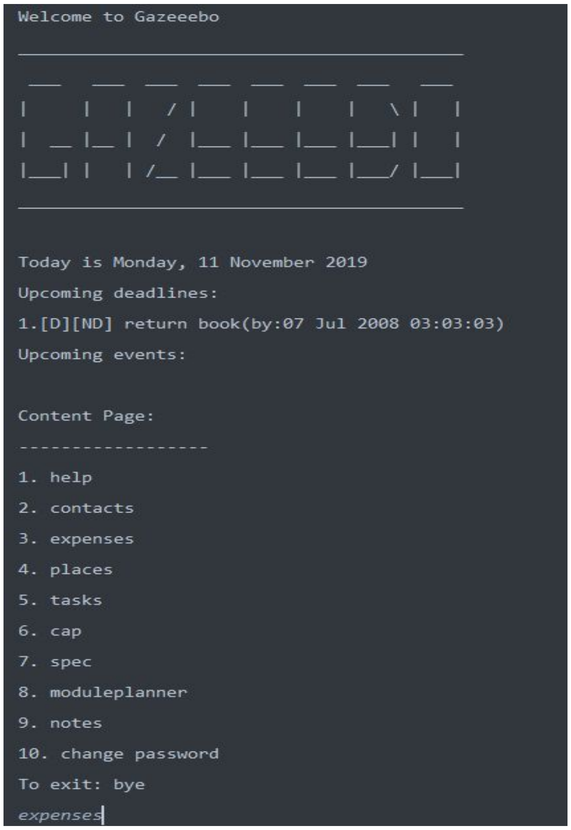
3.6.6. Undo previous command: undo [coming in v2.0]

Undo the previous command for CAP Page.

Format: `undo`

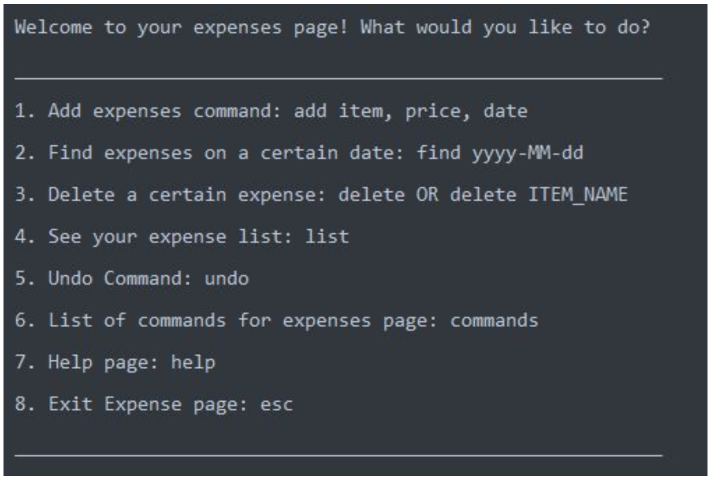
3.7. Expenses Page

Brings you to your expenses page from the main menu page.
In your expenses page, you can record your expenses to help them manage their finances and see what they spend their money on.



```
Welcome to Gazeeebo
__________________________________________

 ___   ___  ___  ___  ___  ___  ___   ___
|     |  |  / |   |   |   |  \ |   |
|  __ |__|  /  |___ |___ |___ |___| |   |
|___| |  | /__ |___ |___ |___ |___/ |___|
__________________________________________


Today is Monday, 11 November 2019
Upcoming deadlines:
1.[D][ND] return book(by:07 Jul 2008 03:03:03)
Upcoming events:

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
expenses
```

Figure 3.7.0.1 How to go to your expenses page



```
Welcome to your expenses page! What would you like to do?

__________________________________________________________

1. Add expenses command: add item, price, date

2. Find expenses on a certain date: find yyyy-MM-dd

3. Delete a certain expense: delete OR delete ITEM_NAME

4. See your expense list: list

5. Undo Command: undo

6. List of commands for expenses page: commands

7. Help page: help

8. Exit Expense page: esc

__________________________________________________________
```

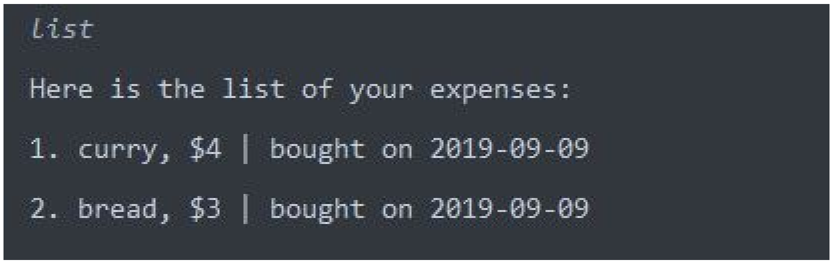
Figure 3.7.0.2 Expenses page

How to go to your expenses page:

- Type in the command `expenses` in the main menu page and press ENTER as shown in Figure 3.7.0.1 and Figure 3.7.0.2 above.
- You can only go to the note page from the main menu.

3.7.1. <u>List all existing expenses: list</u>

Lists out your expenses you have added.



```
list

Here is the list of your expenses:

1. curry, $4 | bought on 2019-09-09

2. bread, $3 | bought on 2019-09-09
```

Figure 3.7.1.1 Expense list

Format: `list`

Example:

- `list`

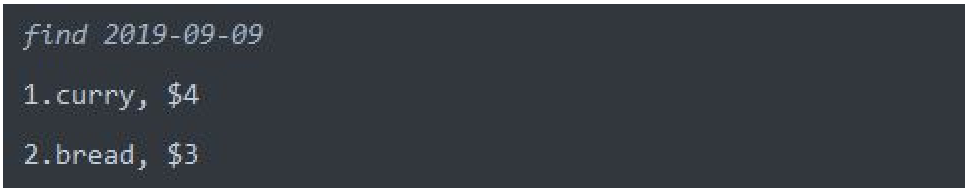
3.7.2. <u>Finding the record of expenses on a certain date: find</u>

Finds all expenses recorded on a specific date.

Format: `find DATE_OF_PURCHASE`

Example:

- `find 2019-09-09`



```
find 2019-09-09
1.curry, $4
2.bread, $3
```

Figure 3.7.2.1 How to find list of expenses on a date

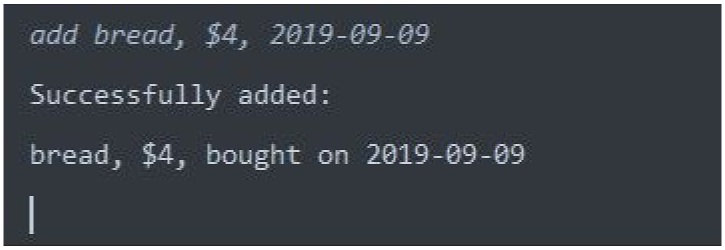
3.7.3. Adding a new expense: add

Adds and store the new expenses to a list.

Format: `add ITEM, PRICE, DATE`

Example:

- `add bread, $4, 2019-09-09`



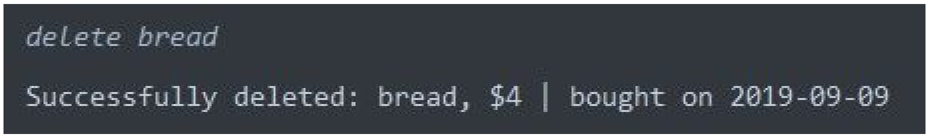
```
add bread, $4, 2019-09-09
Successfully added:
bread, $4, bought on 2019-09-09
|
```

Figure 3.7.3.1 Steps to add a new expense

3.7.4. Delete a certain expense: delete

Deletes an expense from your expenses list .

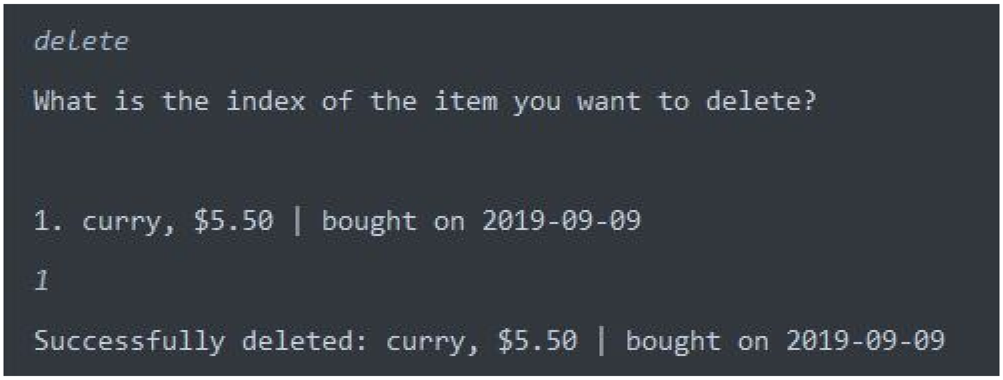
Format: `delete ITEM`



```
delete bread
Successfully deleted: bread, $4 | bought on 2019-09-09
```

Figure 3.7.4.1 Delete expense

Alternative format: `delete`



```
delete
What is the index of the item you want to delete?


1. curry, $5.50 | bought on 2019-09-09
1
Successfully deleted: curry, $5.50 | bought on 2019-09-09
```

Figure 3.7.4.2 Alternative way to delete expense

Steps for deleting an expense:

1. Type the command `delete` and press ENTER.
2. Type the index of the expense you want to delete and press ENTER.

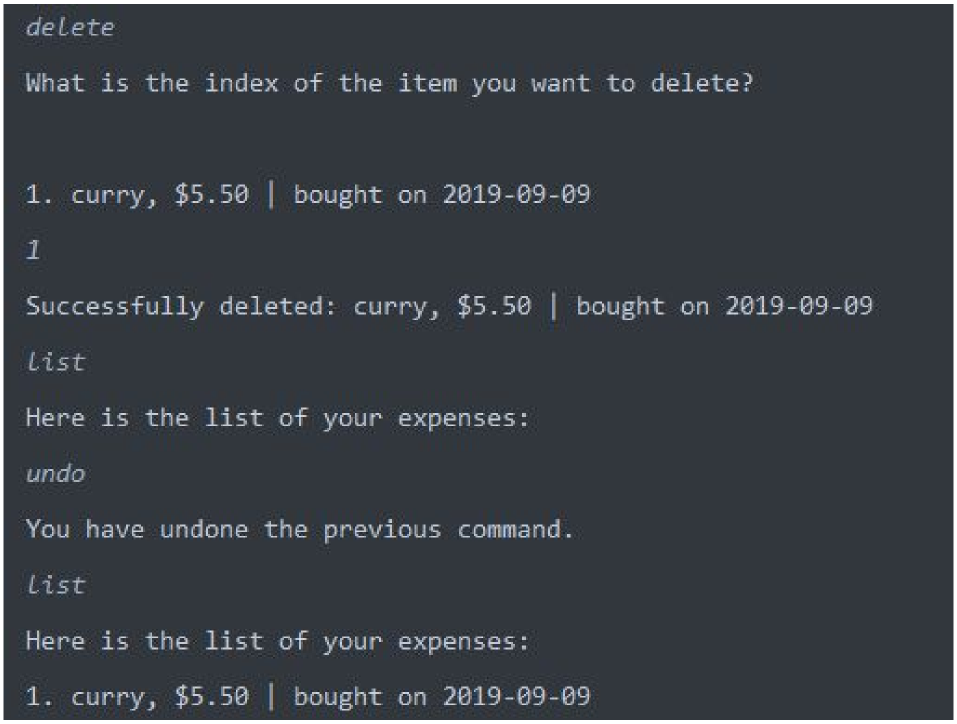
3.7.5. Undo previous expense command: undo

Undo the previous expense command.

Commands that can be undone:

- add
- delete

Format : `undo`



```
delete
What is the index of the item you want to delete?


1. curry, $5.50 | bought on 2019-09-09
1
Successfully deleted: curry, $5.50 | bought on 2019-09-09
list
Here is the list of your expenses:
undo
You have undone the previous command.
list
Here is the list of your expenses:
1. curry, $5.50 | bought on 2019-09-09
```

Figure 3.7.5.1. An example of undoing a delete command

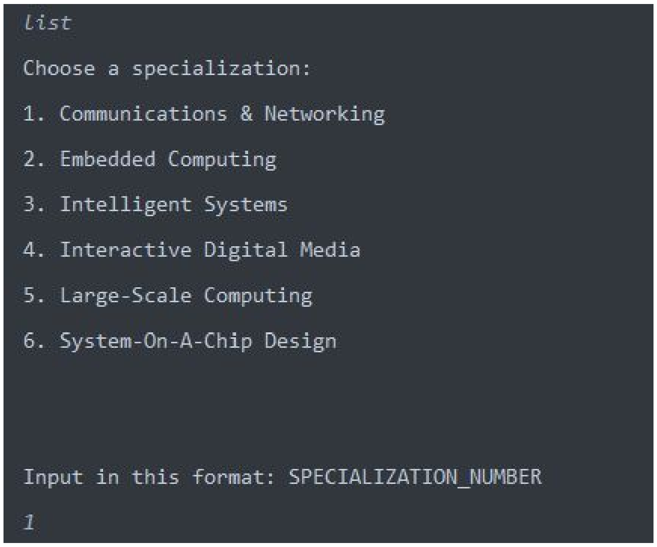
3.8. Specialization Page: spec

Brings you to your specialization page from the main menu page. This specialization page helps you be clearer of what specialization you can take in your course and record the technical elective modules taken to see the progress of achieving that specialization.

3.8.1 List all specialization and Technical Electives

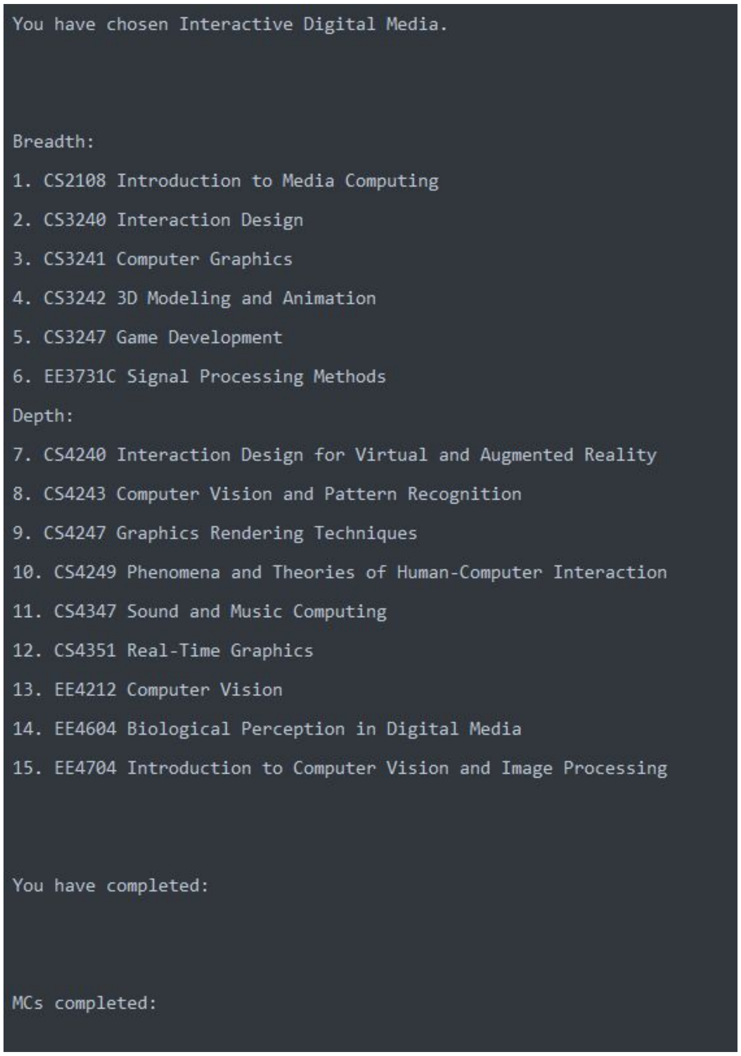
Shows all possible specializations and their respective modules. Number of MCs completed out of 20 will be shown too. (20MCs are needed to complete a specialization)

Format: `list`



```
list
Choose a specialization:
1. Communications & Networking
2. Embedded Computing
3. Intelligent Systems
4. Interactive Digital Media
5. Large-Scale Computing
6. System-On-A-Chip Design



Input in this format: SPECIALIZATION_NUMBER
1
```

Figure 3.8.1.1. When list is called



```
You have chosen Interactive Digital Media.



Breadth:
1. CS2108 Introduction to Media Computing
2. CS3240 Interaction Design
3. CS3241 Computer Graphics
4. CS3242 3D Modeling and Animation
5. CS3247 Game Development
6. EE3731C Signal Processing Methods
Depth:
7. CS4240 Interaction Design for Virtual and Augmented Reality
8. CS4243 Computer Vision and Pattern Recognition
9. CS4247 Graphics Rendering Techniques
10. CS4249 Phenomena and Theories of Human-Computer Interaction
11. CS4347 Sound and Music Computing
12. CS4351 Real-Time Graphics
13. EE4212 Computer Vision
14. EE4604 Biological Perception in Digital Media
15. EE4704 Introduction to Computer Vision and Image Processing



You have completed:



MCs completed:
```

Figure 3.8.8.2. Choose a specialization to see the technical electives

Example:

- `list`

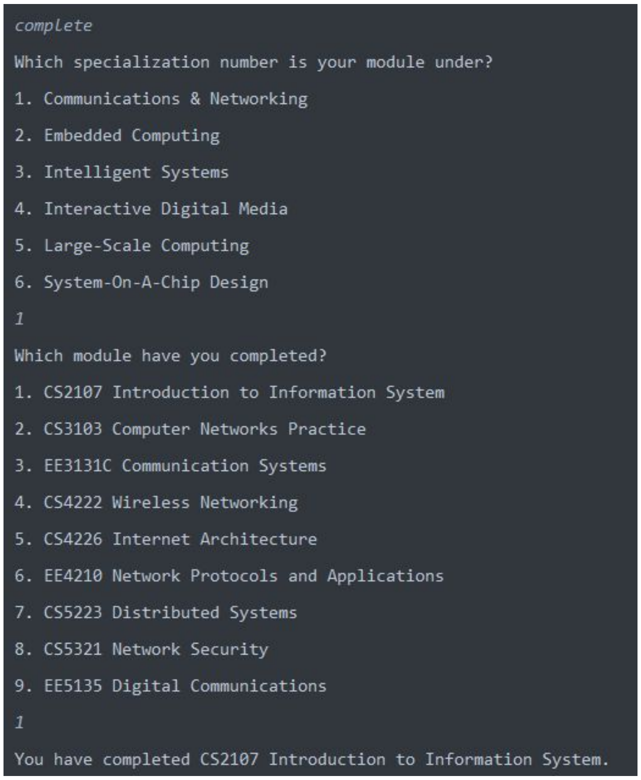
Steps for showing the list of specialization and technical electives:

1. Type the command `list` and press ENTER.
2. You will be prompted with the hint “Input in this format: SPECIALIZATION_NUMBER” to choose your desired specialization
3. Type the index of the specialization and press ENTER.

3.8.2 Key in completed electives

Allow users to mark their chosen elective as completed and record it under the specific specialization.

Format: `complete`



```
complete
Which specialization number is your module under?
1. Communications & Networking
2. Embedded Computing
3. Intelligent Systems
4. Interactive Digital Media
5. Large-Scale Computing
6. System-On-A-Chip Design
1
Which module have you completed?
1. CS2107 Introduction to Information System
2. CS3103 Computer Networks Practice
3. EE3131C Communication Systems
4. CS4222 Wireless Networking
5. CS4226 Internet Architecture
6. EE4210 Network Protocols and Applications
7. CS5223 Distributed Systems
8. CS5321 Network Security
9. EE5135 Digital Communications
1
You have completed CS2107 Introduction to Information System.
```

Figure 3.8.2.1. Marking a technical elective as completed

Steps for recording a completed technical elective module:

1. Type the command `complete` and press ENTER.
2. You will be prompted with the question “Which specialization number is your module under?”
3. Type the index of the specialization and press ENTER.
4. You will be prompted with the question “Which module have you completed?”
5. Type the index of the module completed and press ENTER.

3.8.3 Undo completed elective command: undo [coming in v2.0]

Allow users to undo the previous command.

Format: `undo`

3.8.4 Uncheck a completed elective: uncheck [coming in v2.0]

Allows user to uncheck any completed elective.

Format: `uncheck INDEX`

3.9. Password to enter the planner

The password is a security measure to prevent other users from accessing the user’s Gazeeebo without his permission. Default password is `jjjry`.

Enter the correct password set

Format: `Password_text`

Example:

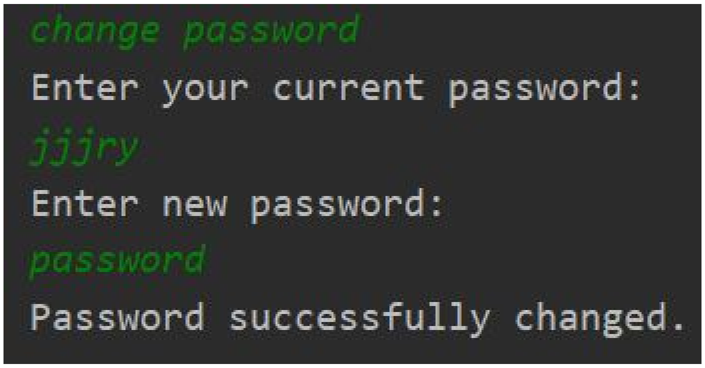
- `jjjry`

3.10. Change Password based on keyword: change password

Change Password. Feature can be called in the main page.

Step 1: Input `change password`
Step 2: Input old password.
Step 2: Input new password.

Example:



```
change password
Enter your current password:
jjjry
Enter new password:
password
Password successfully changed.
```

3.11. Module Planner Page

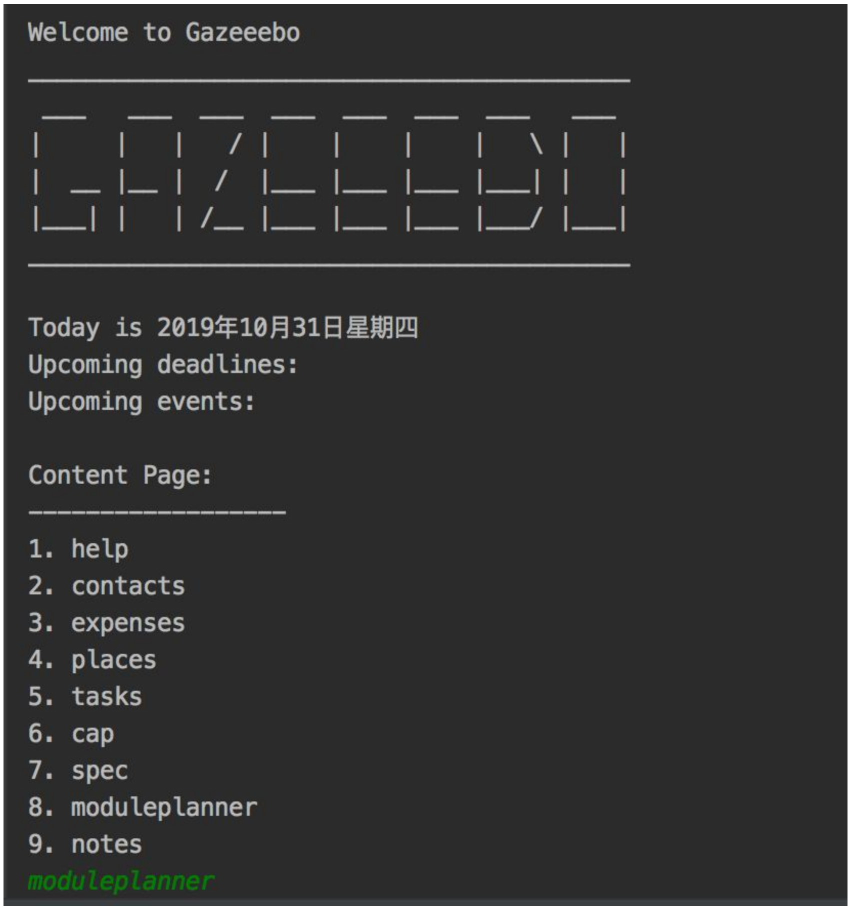
The Module Planner block mainly helps users to create and manage a 4-year module plan for studying life in NUS

How to get to the Module Planner page:

First way:

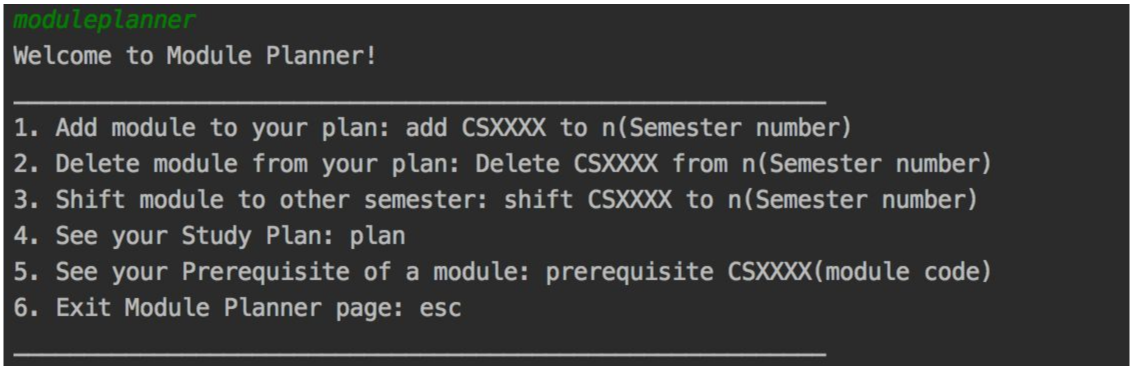
- From the main page, type in the command *`moduleplanner,`* And then system will drops into Module Planner session
- You could only enter the module planner page from the main page

Following image shows the above step and the start page of Module Planner session after the user enters in.



```
Welcome to Gazeeebo
__________________________________________
 ___   ___  ___  ___  ___  ___  ___   ___
|     |   |   / |    |    |    |   \ |   |
|  __ |__ |  /  |___ |___ |___ |___| |   |
|___| |   | /__ |___ |___ |___ |___/ |___|
__________________________________________

Today is 2019年10月31日星期四
Upcoming deadlines:
Upcoming events:

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
moduleplanner
```

Figure 3.11.1. How to get into the Module Planner Page



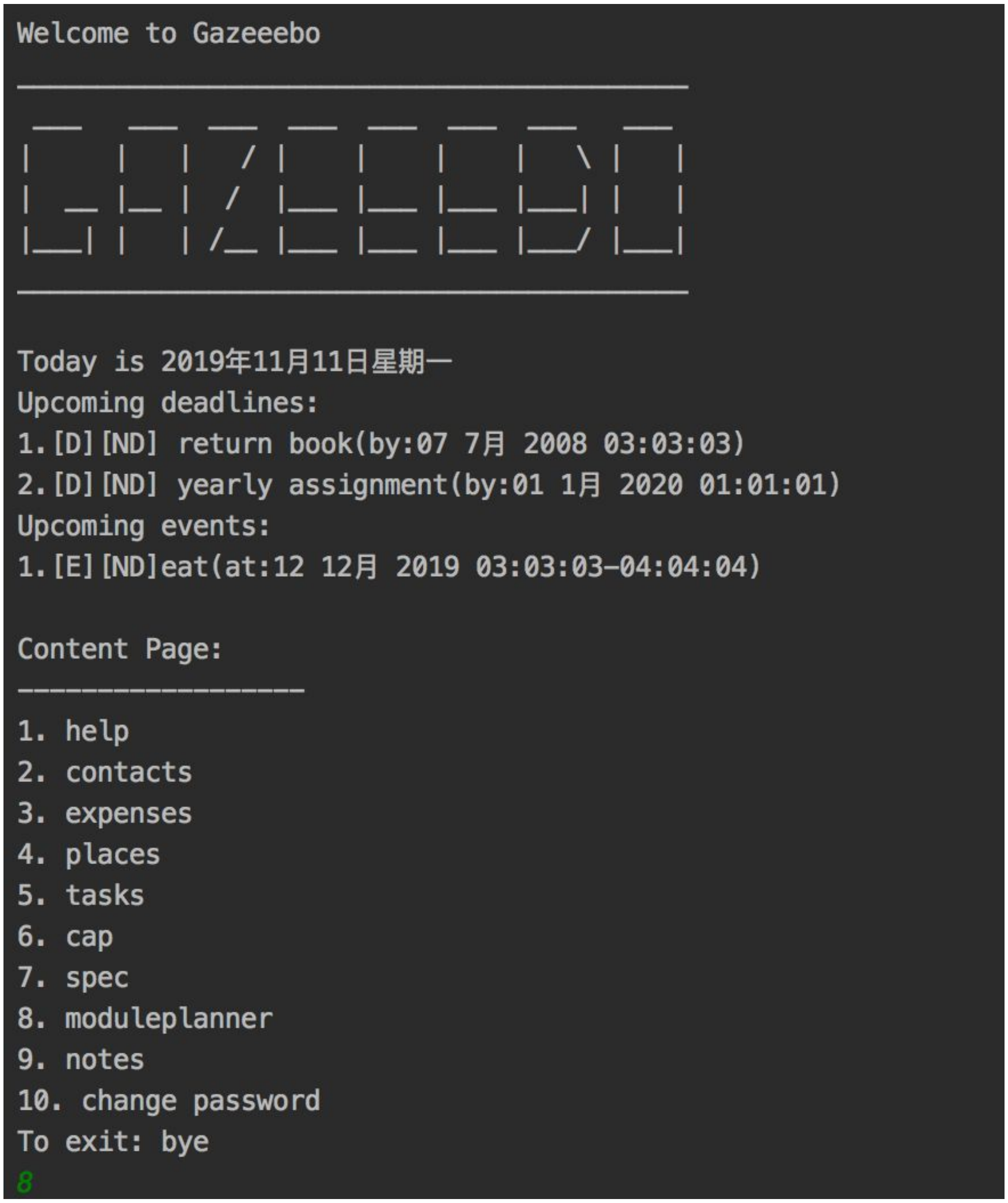
```
moduleplanner
Welcome to Module Planner!
__________________________________________________________
1. Add module to your plan: add CSXXXX to n(Semester number)
2. Delete module from your plan: Delete CSXXXX from n(Semester number)
3. Shift module to other semester: shift CSXXXX to n(Semester number)
4. See your Study Plan: plan
5. See your Prerequisite of a module: prerequisite CSXXXX(module code)
6. Exit Module Planner page: esc
__________________________________________________________
```

Figure 3.11.2 The start page of Module Planner page

Second way:

- From the main page, type in index `8,` And then system will drops into Module Planner session
- You could only enter the module planner page from the main page

Following image shows the above step and the start page of Module Planner session after the user enters in.



```
Welcome to Gazeeebo
__________________________________________
 ___   ___  ___  ___  ___  ___  ___   ___
|     |   |   / |    |    |    |   \ |   |
|  __ |__ |  /  |___ |___ |___ |___| |   |
|___| |   | /__ |___ |___ |___ |___/ |___|
__________________________________________

Today is 2019年11月11日星期一
Upcoming deadlines:
1.[D][ND] return book(by:07 7月 2008 03:03:03)
2.[D][ND] yearly assignment(by:01 1月 2020 01:01:01)
Upcoming events:
1.[E][ND]eat(at:12 12月 2019 03:03:03-04:04:04)

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
8
```

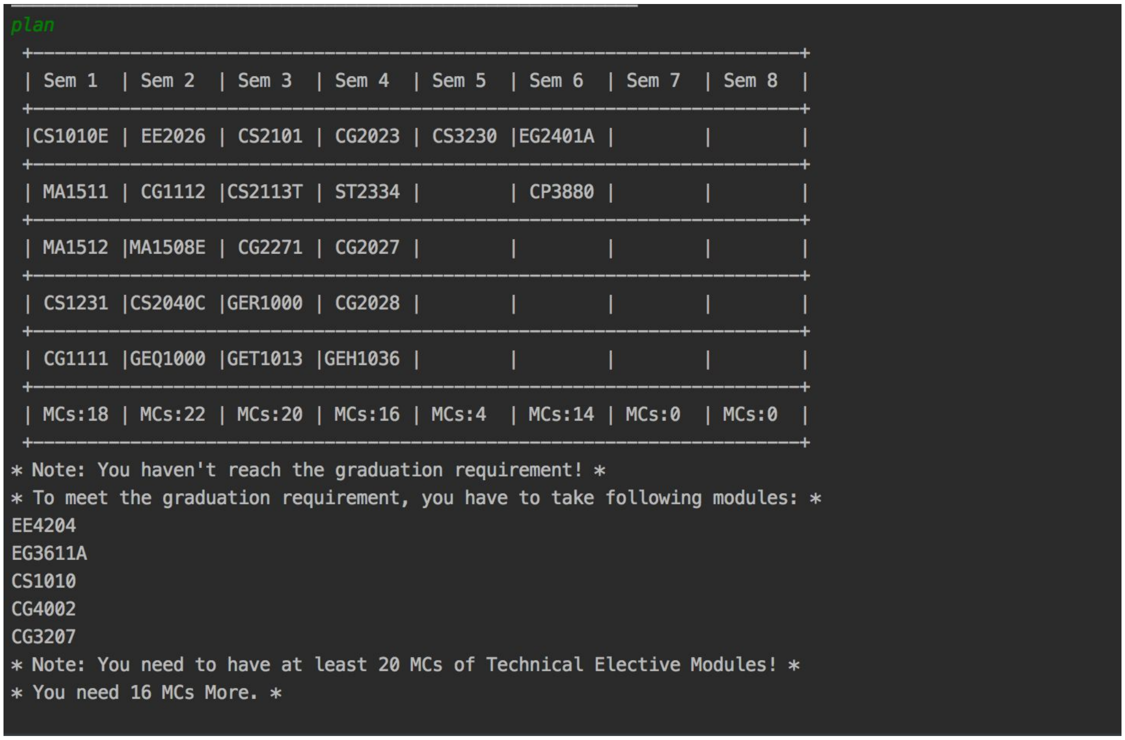
Figure 3.11.3. Another way of entering the module planner page.

3.11.1. <u>Display the module plan table and showcase suggestions: plan</u>

Description: showcase the module plan table

Format: `plan`

- If it is the first time user is using this feature, the system showcase default module plan table which is the CEG recommended module plan for AY18/19 intake.



```
plan
 +-----------------------------------------------------------------------+
 | Sem 1  | Sem 2  | Sem 3  | Sem 4  | Sem 5  | Sem 6  | Sem 7  | Sem 8  |
 +-----------------------------------------------------------------------+
 |CS1010E | EE2026 | CS2101 | CG2023 | CS3230 |EG2401A |        |        |
 +-----------------------------------------------------------------------+
 | MA1511 | CG1112 |CS2113T | ST2334 |        | CP3880 |        |        |
 +-----------------------------------------------------------------------+
 | MA1512 |MA1508E | CG2271 | CG2027 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CS1231 |CS2040C |GER1000 | CG2028 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CG1111 |GEQ1000 |GET1013 |GEH1036 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | MCs:18 | MCs:22 | MCs:20 | MCs:16 | MCs:4  | MCs:14 | MCs:0  | MCs:0  |
 +-----------------------------------------------------------------------+
* Note: You haven't reach the graduation requirement! *
* To meet the graduation requirement, you have to take following modules: *
EE4204
EG3611A
CS1010
CG4002
CG3207
* Note: You need to have at least 20 MCs of Technical Elective Modules! *
* You need 16 MCs More. *
```

Figure 3.11.1.1: Example of displaying module plan using `plan` command

3.11.2. Add module to specific semester column: add

Description: Add a new module to any column of the study plan table

Format: `add module_code to semester_number`

Example: `add CS4223 to 5`

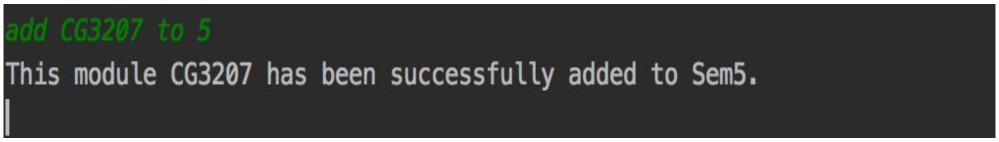
Steps to add a command:

1. Type in command `add module_code to semester_number` and press ENTER
2. System will showcase the module is successfully added.
   - If adding is not successful, an IOException message will be displayed

Format details:

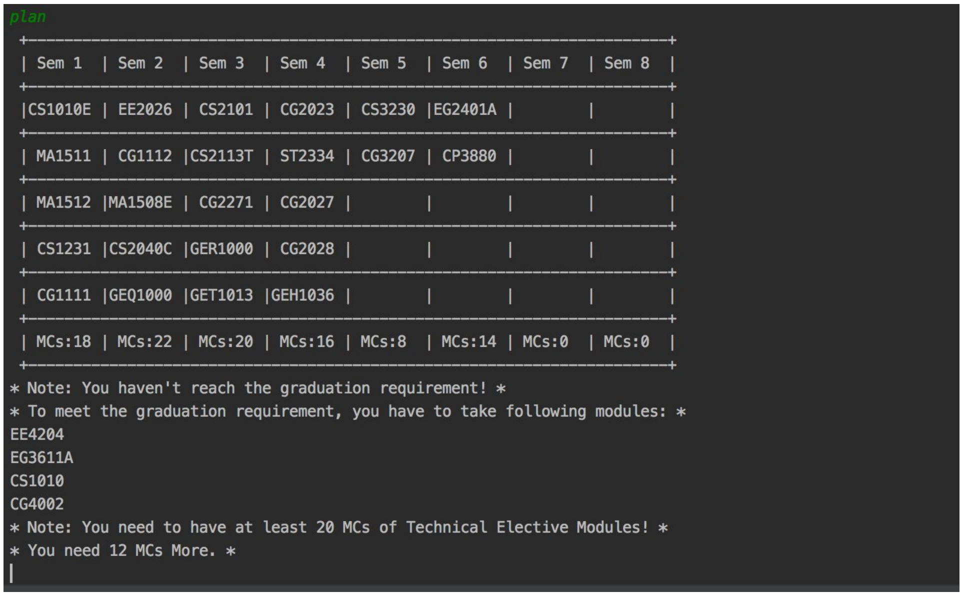
1. `Module_code` add should not been existed inside the study_plan

2. `semester_number` starts from 1 to 8, could not exceed 8 or be smaller than 1.



```
add CG3207 to 5
This module CG3207 has been successfully added to Sem5.
```

Figure 3.11.2.1. Example of adding a module CG3207



```
plan
 +-----------------------------------------------------------------------+
 | Sem 1  | Sem 2  | Sem 3  | Sem 4  | Sem 5  | Sem 6  | Sem 7  | Sem 8  |
 +-----------------------------------------------------------------------+
 |CS1010E | EE2026 | CS2101 | CG2023 | CS3230 |EG2401A |        |        |
 +-----------------------------------------------------------------------+
 | MA1511 | CG1112 |CS2113T | ST2334 | CG3207 | CP3880 |        |        |
 +-----------------------------------------------------------------------+
 | MA1512 |MA1508E | CG2271 | CG2027 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CS1231 |CS2040C |GER1000 | CG2028 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CG1111 |GEQ1000 |GET1013 |GEH1036 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | MCs:18 | MCs:22 | MCs:20 | MCs:16 | MCs:8  | MCs:14 | MCs:0  | MCs:0  |
 +-----------------------------------------------------------------------+
* Note: You haven't reach the graduation requirement! *
* To meet the graduation requirement, you have to take following modules: *
EE4204
EG3611A
CS1010
CG4002
* Note: You need to have at least 20 MCs of Technical Elective Modules! *
* You need 12 MCs More. *
```

Figure 3.11.2.2. Results of adding, you could see module CG3207 inside Sem 5 column

3.11.3. Delete module from specific semester column: delete

Description: delete an existing module from any column of the study plan table

Format: `delete module_code from semester_number`

Example: `delete CS4223 from 5`

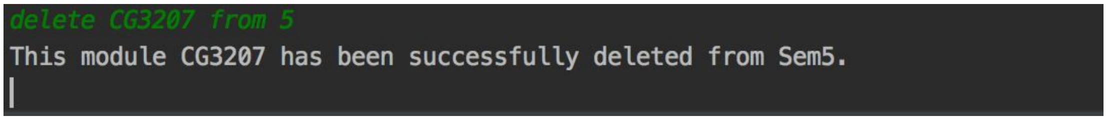
Steps to add a command:

3. Type in command `delete module_code from semester_number`and press ENTER
4. System will showcase the module is successfully deleted.
   - If deleting is not successful, an IOException message will be displayed

Format details:

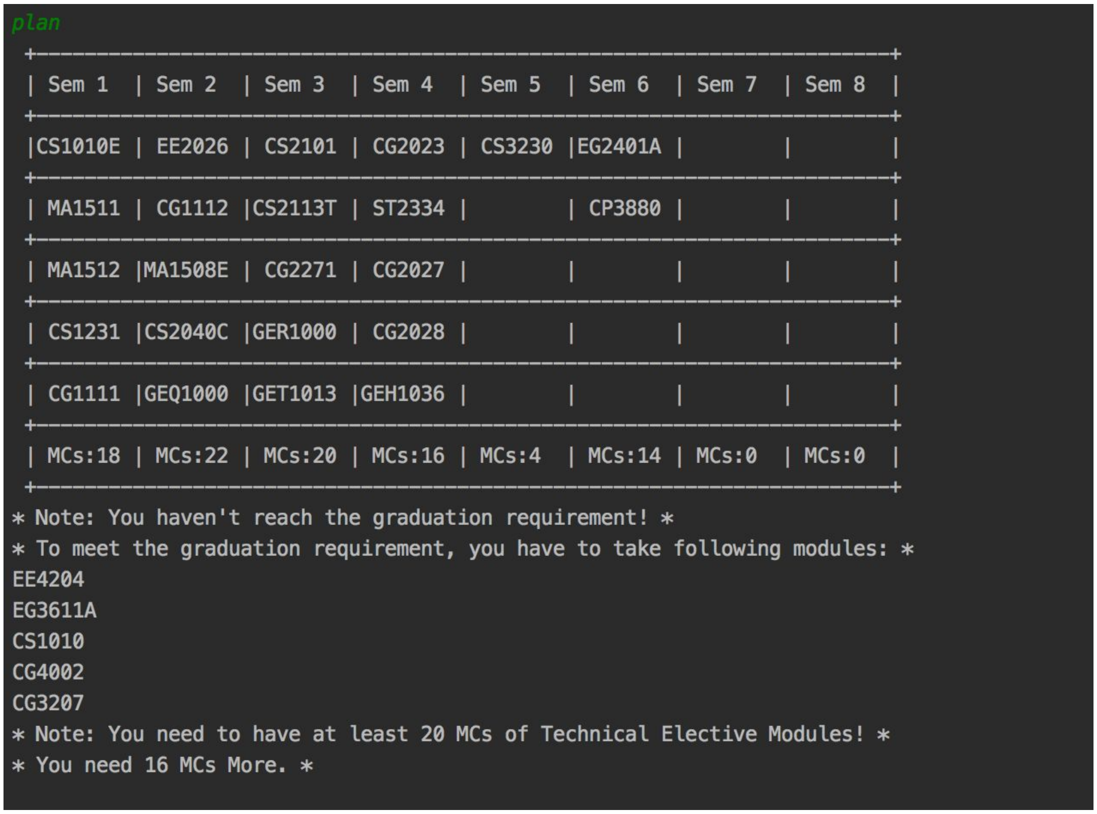
1. `Semester_number` could not be smaller than 1, it neither could exceed 8.

2. `Module_code` should be a module that is already existed inside the study plan



```
delete CG3207 from 5
This module CG3207 has been successfully deleted from Sem5.
```

Figure 3.11.3.1 Example of deleting module CG3207 from semester 5.



```
plan
 +-----------------------------------------------------------------------+
 | Sem 1  | Sem 2  | Sem 3  | Sem 4  | Sem 5  | Sem 6  | Sem 7  | Sem 8  |
 +-----------------------------------------------------------------------+
 |CS1010E | EE2026 | CS2101 | CG2023 | CS3230 |EG2401A |        |        |
 +-----------------------------------------------------------------------+
 | MA1511 | CG1112 |CS2113T | ST2334 |        | CP3880 |        |        |
 +-----------------------------------------------------------------------+
 | MA1512 |MA1508E | CG2271 | CG2027 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CS1231 |CS2040C |GER1000 | CG2028 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CG1111 |GEQ1000 |GET1013 |GEH1036 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | MCs:18 | MCs:22 | MCs:20 | MCs:16 | MCs:4  | MCs:14 | MCs:0  | MCs:0  |
 +-----------------------------------------------------------------------+
* Note: You haven't reach the graduation requirement! *
* To meet the graduation requirement, you have to take following modules: *
EE4204
EG3611A
CS1010
CG4002
CG3207
* Note: You need to have at least 20 MCs of Technical Elective Modules! *
* You need 16 MCs More. *
```

Figure 3.11.3.2. Deleting results, you can see CG3207 is removed from column Sem 5

3.11.4. Shift module from specific semester column to another column: shift

Description: shift an existing module from any column to another column of the study plan table

Format: `shift module_code to semester_number`

Example: `shift CS4223 to 5`

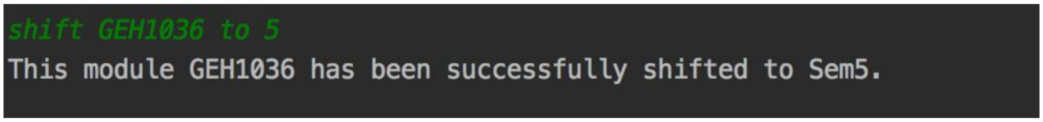
Steps to add a command:

5. Type in command `shift module_code to semester_number`and press ENTER

6. System will showcase the module is successfully shifted.
    - If shifting is not successful, an IOException message will be displayed

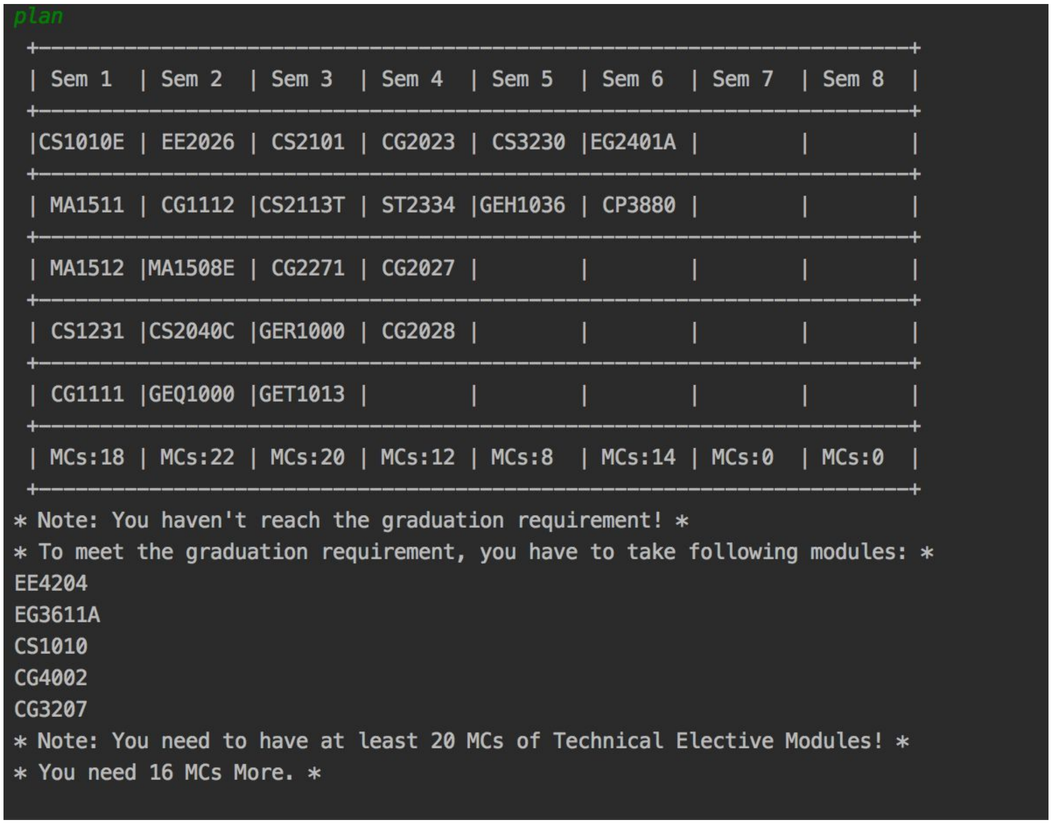
Format details:

1. `Module_code` must be an existing module inside the study plan.
2. `Semester_number` could not be smaller than 1 or greater than 8.



```
shift GEH1036 to 5
This module GEH1036 has been successfully shifted to Sem5.
```

Figure 3.11.4.1 Example of shifting a module GEH1036 to semester 5.



```
plan
 +-----------------------------------------------------------------------+
 | Sem 1  | Sem 2  | Sem 3  | Sem 4  | Sem 5  | Sem 6  | Sem 7  | Sem 8  |
 +-----------------------------------------------------------------------+
 |CS1010E | EE2026 | CS2101 | CG2023 | CS3230 |EG2401A |        |        |
 +-----------------------------------------------------------------------+
 | MA1511 | CG1112 |CS2113T | ST2334 |GEH1036 | CP3880 |        |        |
 +-----------------------------------------------------------------------+
 | MA1512 |MA1508E | CG2271 | CG2027 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CS1231 |CS2040C |GER1000 | CG2028 |        |        |        |        |
 +-----------------------------------------------------------------------+
 | CG1111 |GEQ1000 |GET1013 |        |        |        |        |        |
 +-----------------------------------------------------------------------+
 | MCs:18 | MCs:22 | MCs:20 | MCs:12 | MCs:8  | MCs:14 | MCs:0  | MCs:0  |
 +-----------------------------------------------------------------------+
* Note: You haven't reach the graduation requirement! *
* To meet the graduation requirement, you have to take following modules: *
EE4204
EG3611A
CS1010
CG4002
CG3207
* Note: You need to have at least 20 MCs of Technical Elective Modules! *
* You need 16 MCs More. *
```

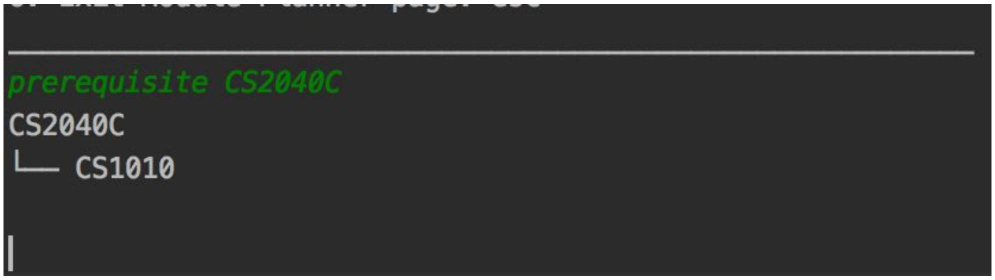
Figure 3.11.4.2. Results of shifting, you could see GEH1036 is now under column Sem 5

3.11.5. Show a prerequisite tree of a module: prerequisite

Description: showcase all prerequisite requirements for a specific module.

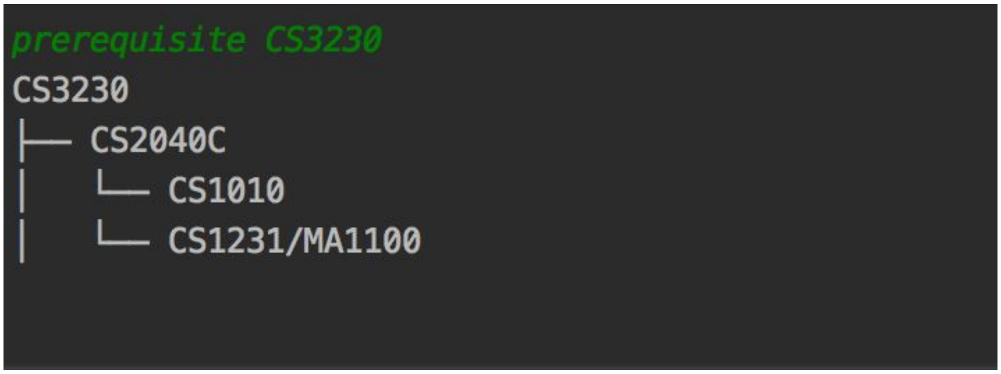
Format: `prerequisite Module_Code`

Example: `prerequisite CS2040C`



```
────────────────────────────────────────────────────────────
prerequisite CS2040C
CS2040C
└── CS1010

|
```

Figure 3.11.5.1. Example outcome of prerequisite command



```
prerequisite CS3230
CS3230
├── CS2040C
│    └── CS1010
│    └── CS1231/MA1100
```

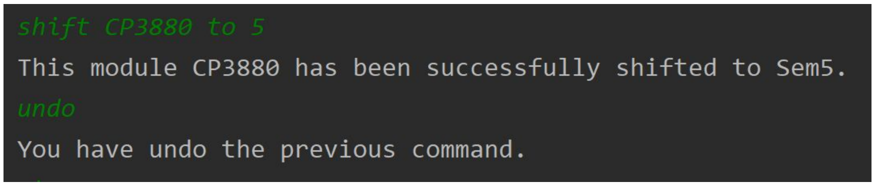
Figure 3.11.5.2. Another example of checking prerequisites of CS3230

3.11.6. Undo previous module planner command: undo

Undo the previous module planner command.
Commands that can be undone:

- add
- delete
- shift

Format : `undo`



```
shift CP3880 to 5
This module CP3880 has been successfully shifted to Sem5.
undo
You have undo the previous command.
```

Figure 3.11.6.1. An example of undoing a shift command

3.11.7. Auto adjust the table base on prerequisites: adjust [Coming in v2.0]

Description: Users could type in adjust command, the system will check whether modules inside the study plan have met their prerequisites. If not, the system will then auto fill in prerequisite module

3.12. Exiting the program: bye

Exits the program.

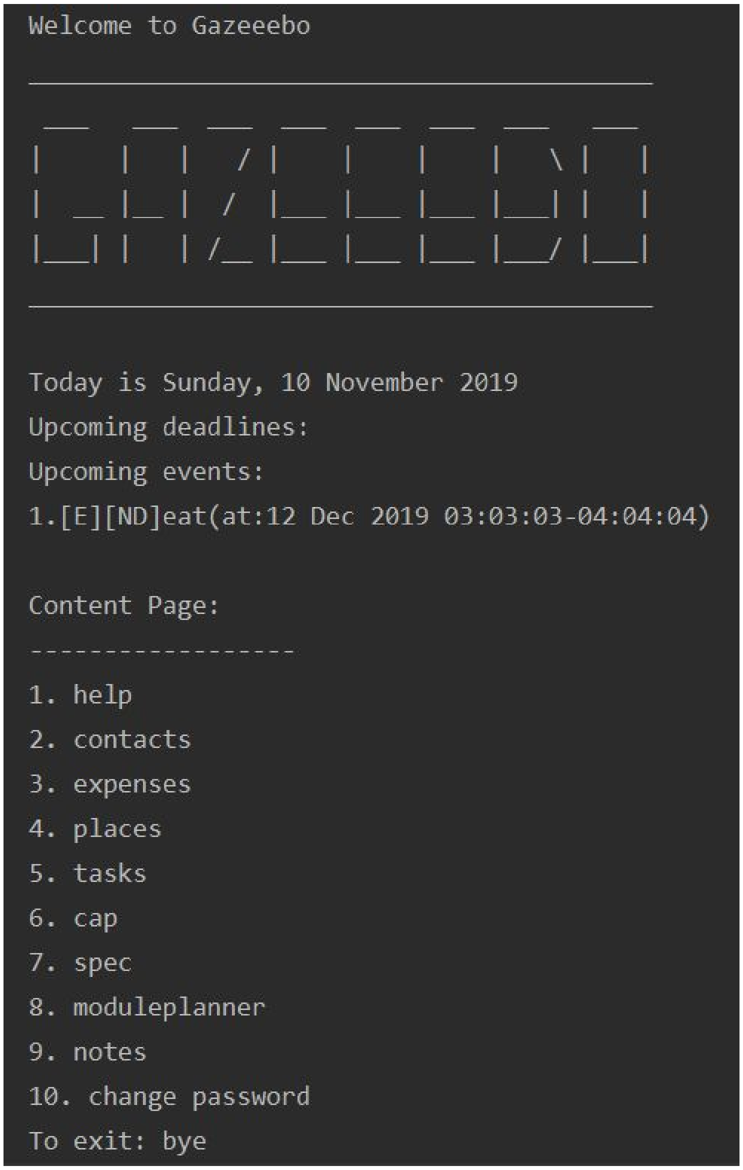
Format: `bye`

3.13. Redo command: redo [coming in v2.0]

Redo the undo command.

Format: `redo`

4. UI



```
Welcome to Gazeeebo
__________________________________________

 ___   ___  ___  ___  ___  ___  ___   ___
|     |   |   / |    |    |    |   \ |   |
|  __ |__ |  /  |___ |___ |___ |___| |   |
|___| |   | /__ |___ |___ |___ |___/ |___|
__________________________________________


Today is Sunday, 10 November 2019
Upcoming deadlines:
Upcoming events:
1.[E][ND]eat(at:12 Dec 2019 03:03:03-04:04:04)

Content Page:
------------------
1. help
2. contacts
3. expenses
4. places
5. tasks
6. cap
7. spec
8. moduleplanner
9. notes
10. change password
To exit: bye
```

Figure 4.1 UI of Gazeeebo