FoodieBot - User Guide

By: `Team SE-EDU` Since: `Feb 2020` Licence: `MIT`

1. Introduction

FoodieBot is a campus food locator application which can help students look for food they want, by recommending the canteens nearest to them. Our target users are students, staff and tourists, in general anyone who comes to or visits NUS. In particular, for the indecisive user, this application can give a random food suggestion tailored to each user based on their budget and/ or past food selections etcetera.

2. About This document

This document shows you how to use the features of Foodiebot.

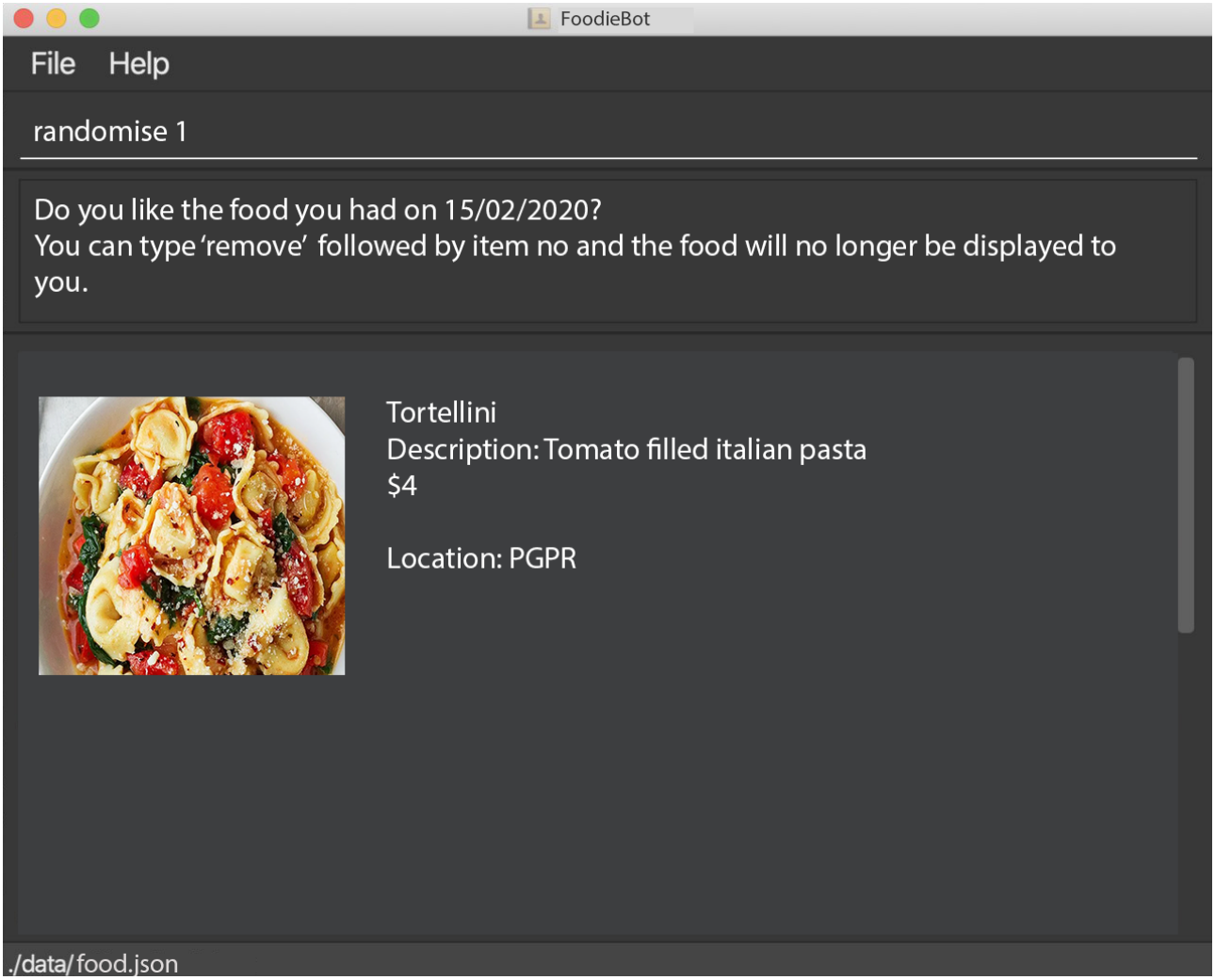
Note the following symbols and formatting used in this document.

NOTE	This symbol indicate important information.
TIP	This symbol indicate tips for the particular feature.

`list` - A grey highlight (called a mark-up) indicates that this is a command that can be typed into the command line and executed by the application.

3. Quick Start

1. Ensure you have Java 11 or above installed in your Computer.
2. Download the latest `foodiebot.jar` here.
3. Copy the file to the folder you want to use as the home folder for your Foodiebot.
4. Double-click the file to start the app. The GUI should appear in a few seconds.



5. Type the command in the command box and press `Enter` to execute it.
   e.g. typing **`help`** and pressing `Enter` will open the help window.
6. Some example commands you can try:
   - **`list`** : Lists all canteens.
   - **`findnearest COM1`** : Locate the nearest canteen from COM1.
   - **`gotodeck f/ COM1`** : Display direction to go to deck from COM1.
   - **`exit`** : Exits the app

4. Features

Command Format

- Words in `UPPER_CASE` are the parameters to be supplied by the user e.g. in `budget set w/ AMOUNT`, `AMOUNT` is a parameter which can be used as `budget set w/ 9.50`
- Items in square brackets are optional e.g `report [w/DATE]` can be used as `report [w/ 12-02-2020]` or as `report`.

4.1. List All Canteens: `list`

Display a list of all available canteens on campus.

Format: `list`

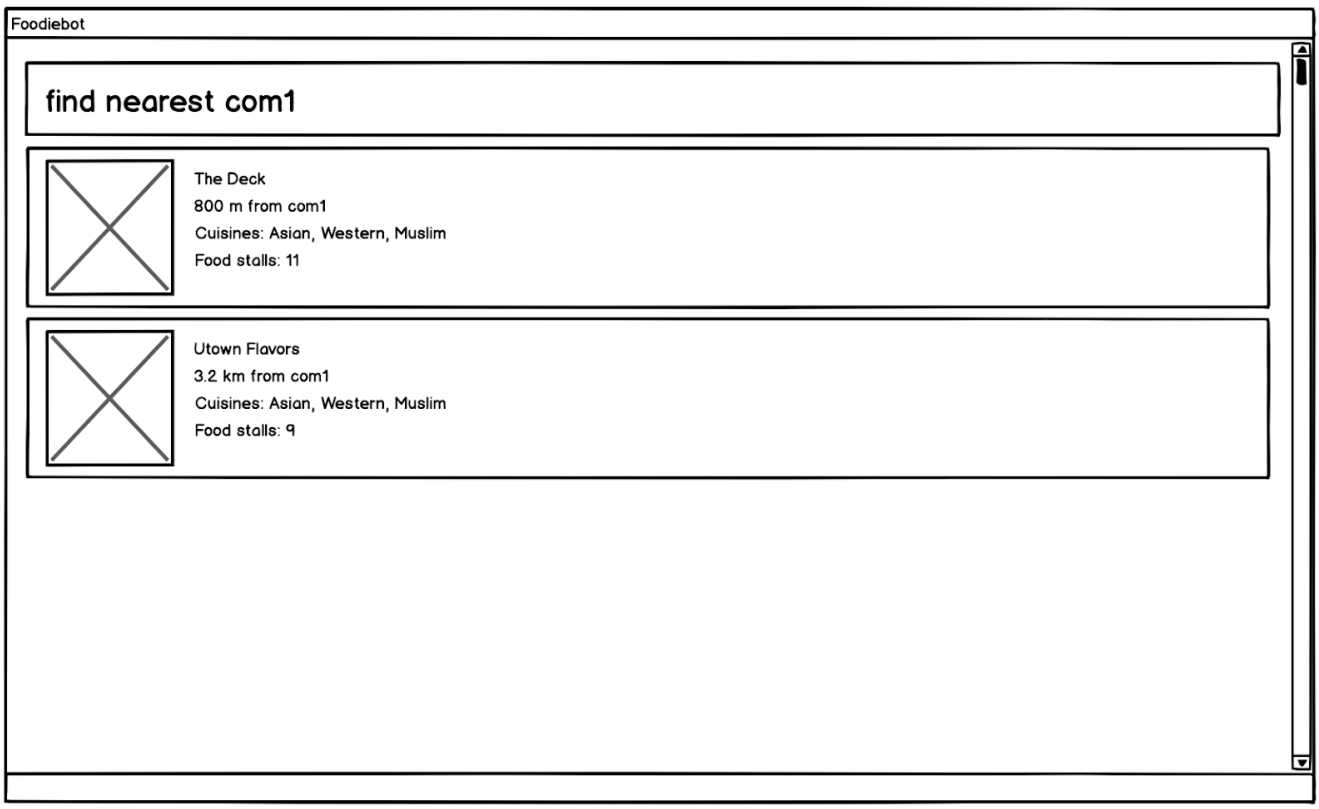
4.2. Find Nearest Canteens: `find nearest`

Display a list of all canteens ordered by increasing distance from current location.

Format: `find nearest BLOCK_NAME`

- The `BLOCK_NAME` will be populated with suggestions as the user types.

NOTE	The `BLOCK_NAME` has to be one of the suggestions. Otherwise an error message will be displayed that requires correct location to be provided.



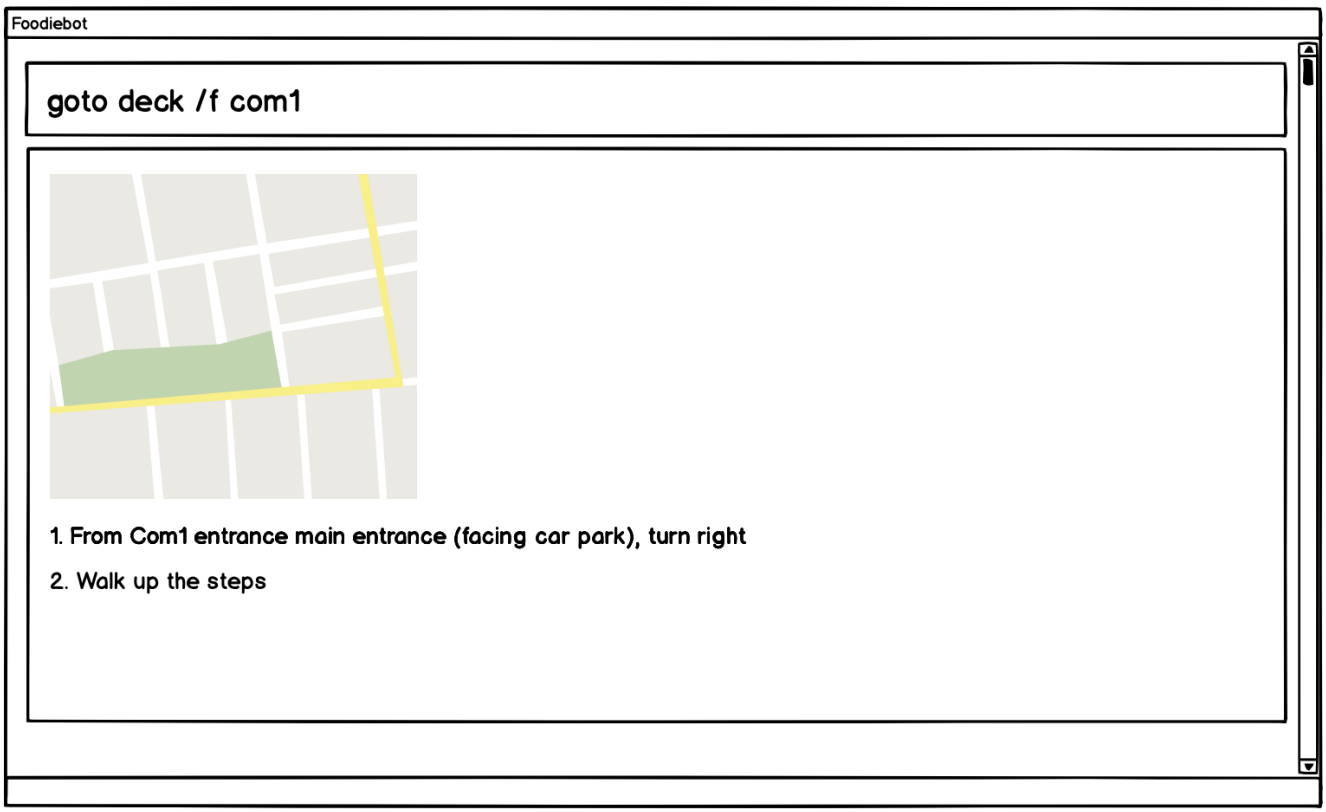
4.3. Go to Nearest Canteen: `goto`

Display a map with the route between start location and destination. Includes the travel instructions and bus services that go to the canteen.

Format: `goto CANTEEN_NAME f/ CURRENT_LOCATION`

- `CANTEEN_NAME` and `CURRENT_LOCATION` field will be populated with suggestions as the user types.

NOTE	`CANTEEN_NAME` and `CURRENT_LOCATION` field has to be one of the suggestions. Otherwise an error message will be displayed that requires the correct location to be provided.



4.4. View All Stores in Canteen: `enter`

Display the stores available at the canteen based on user input.

Format: `enter CANTEEN_NAME`

- The display of the store rating is determined from the user past experience on the food items which were selected.

4.5. View Menu of Store : `menu`

Display the menu of the store based on the user input.

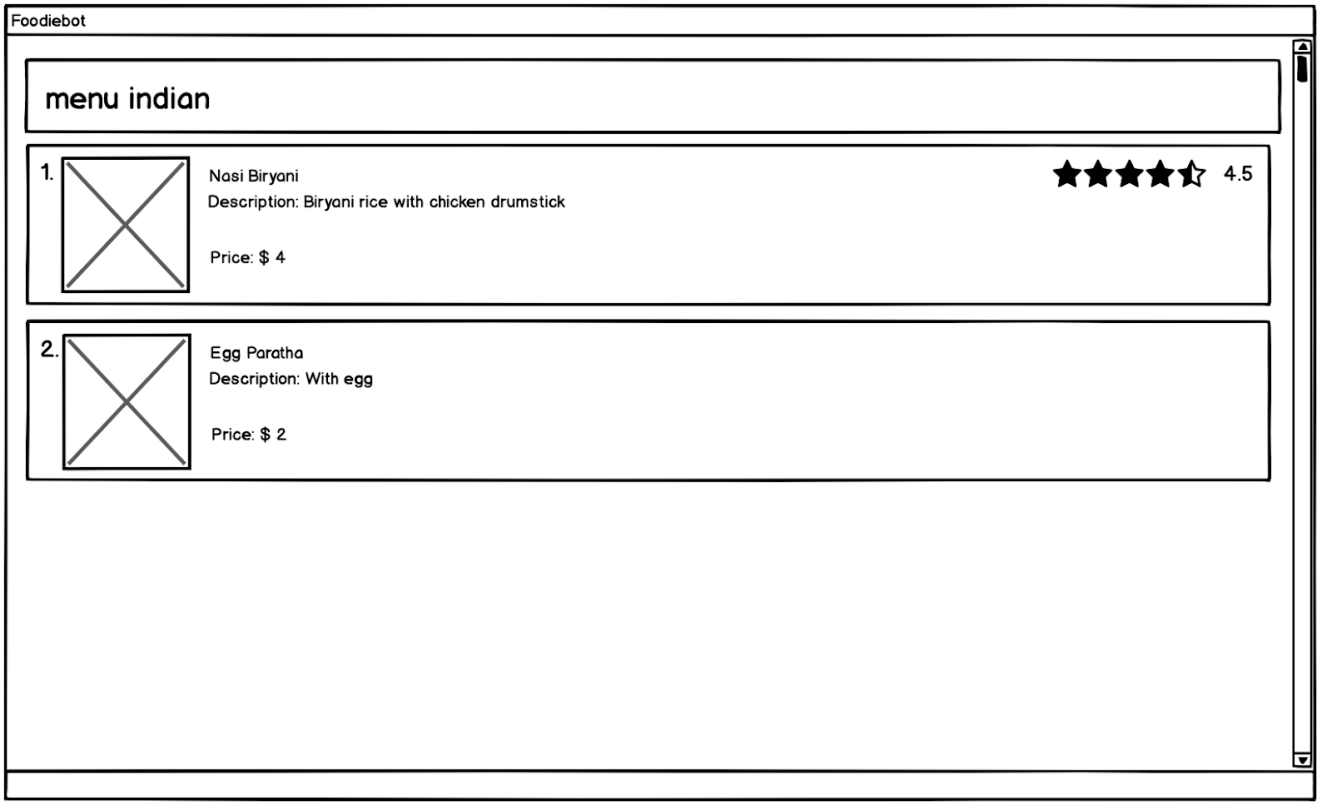
Format: `menu KEYWORD`

NOTE	This function is only available after the user has selected a canteen and store.

- User ratings will be displayed if they are available
  E.g. after the user has selected a food item previously.
- `KEYWORD` includes:
  - `[by price/ name]`: Sorts the menu accordingly.
  - `[tag]`: Displays foods available in the store with the corresponding tag.

Examples:

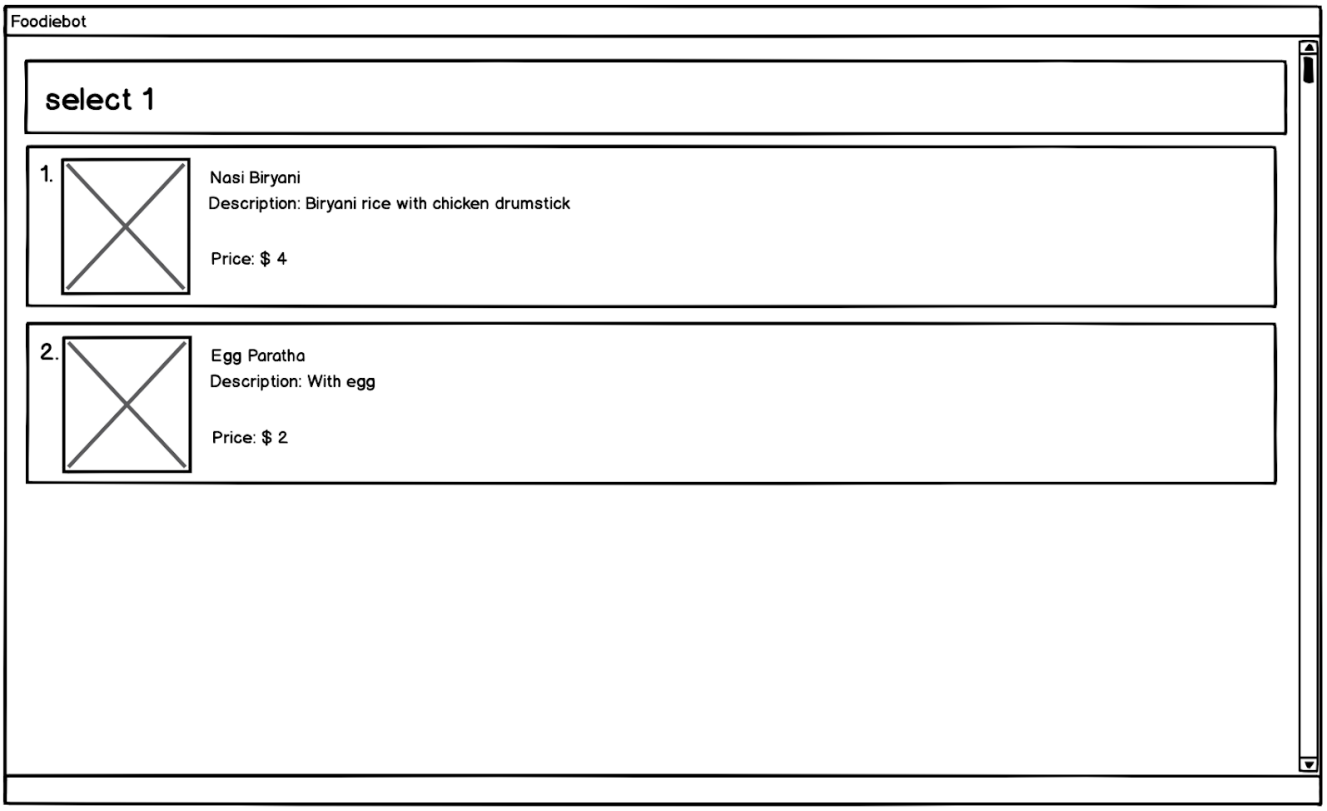
- `menu western`
  Display the food that has been tag with western.



4.6. Select the Food : `select`

This command stores the selected food in the database.

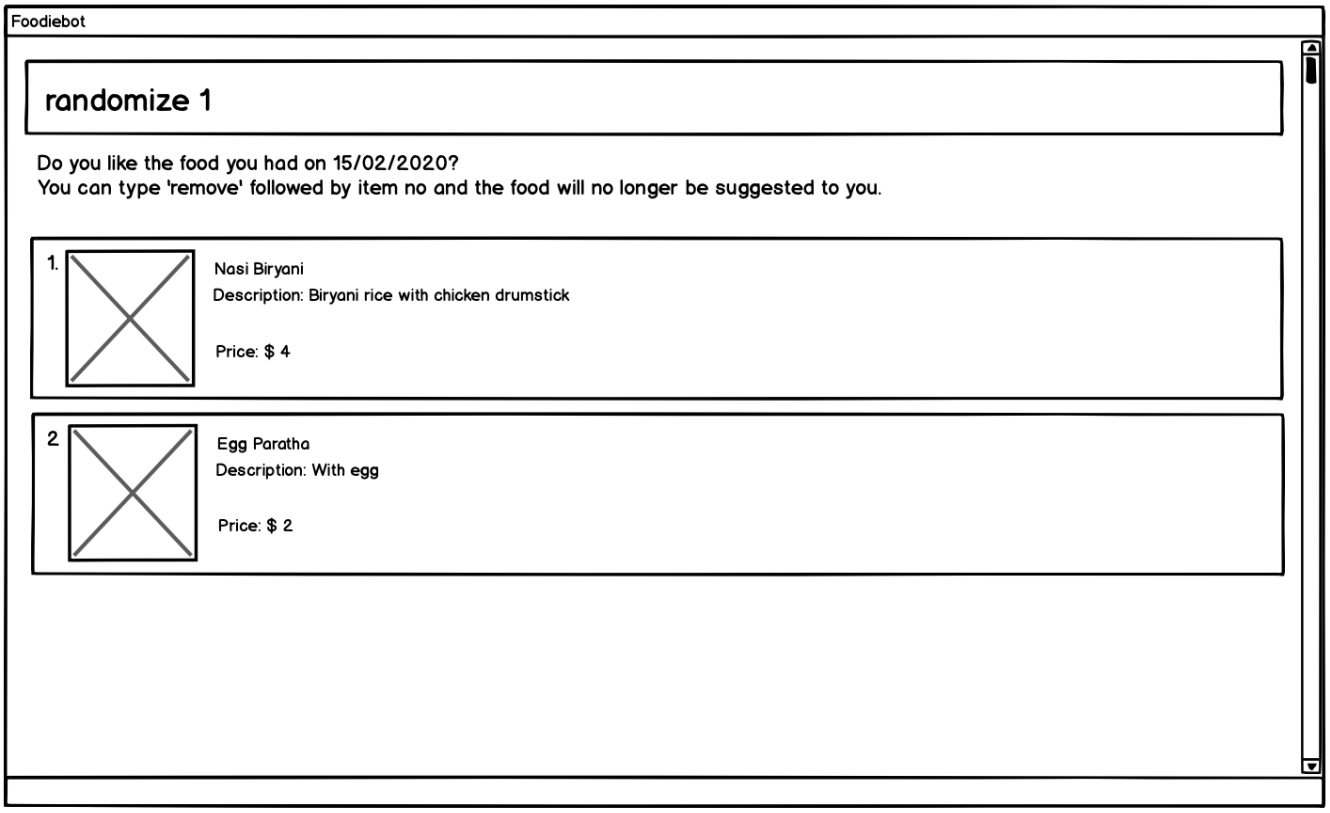
Format: `select INDEX`



4.7. Randomize Food Selection : `randomize`

Display a list of suggestions of food.

Format: `randomize`



4.8. Set a Budget : `budget set`

This command set a daily, weekly or monthly budget.
This budget can be changed, however, this will reset the budget overview for the current budget cycle.

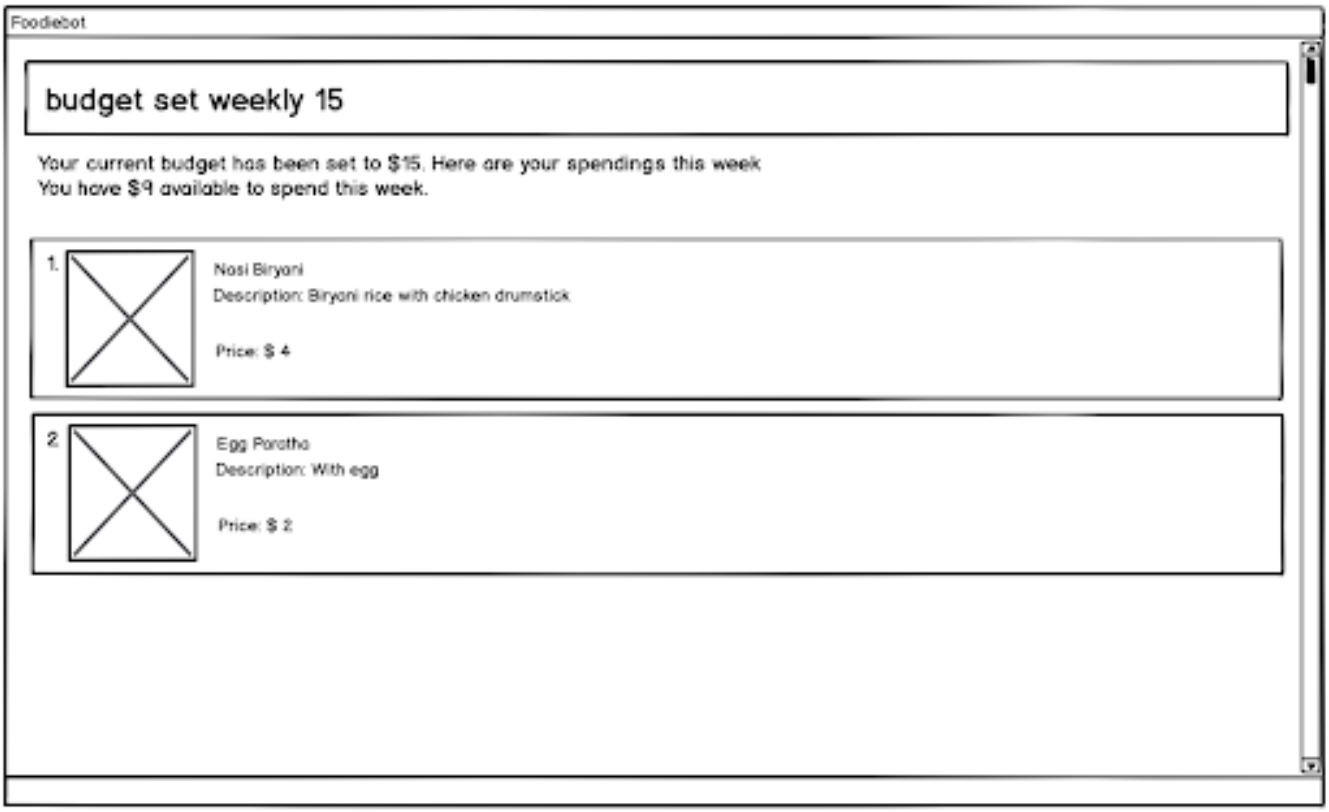
Format: `budget set [PERIOD] [AMOUNT]`

- `AMOUNT`: Defines the limit of your budget.
- `PERIOD`: Defines the length of each period that the budget is effective for.
- List of `PERIOD` inputs includes:
  - `[d/]` - Daily
  - `[w/]` - Weekly
  - `[m/]` - Monthly

NOTE	`PERIOD` field has to be one of the above suggestions. `AMOUNT` field has to be numeric (with or without decimal places). Otherwise an error message will be displayed requesting a correct type to be provided.

Examples:

- `budget set w/ 9.50`
  - Sets your weekly budget to $9.50.
- `budget set m/ 100`
  - Sets your monthly budget to $100.



4.9. View Budget: `budget view`

This command view the current budget, spendings made for the week and the remaining available budget to spend.

Format: `budget view`

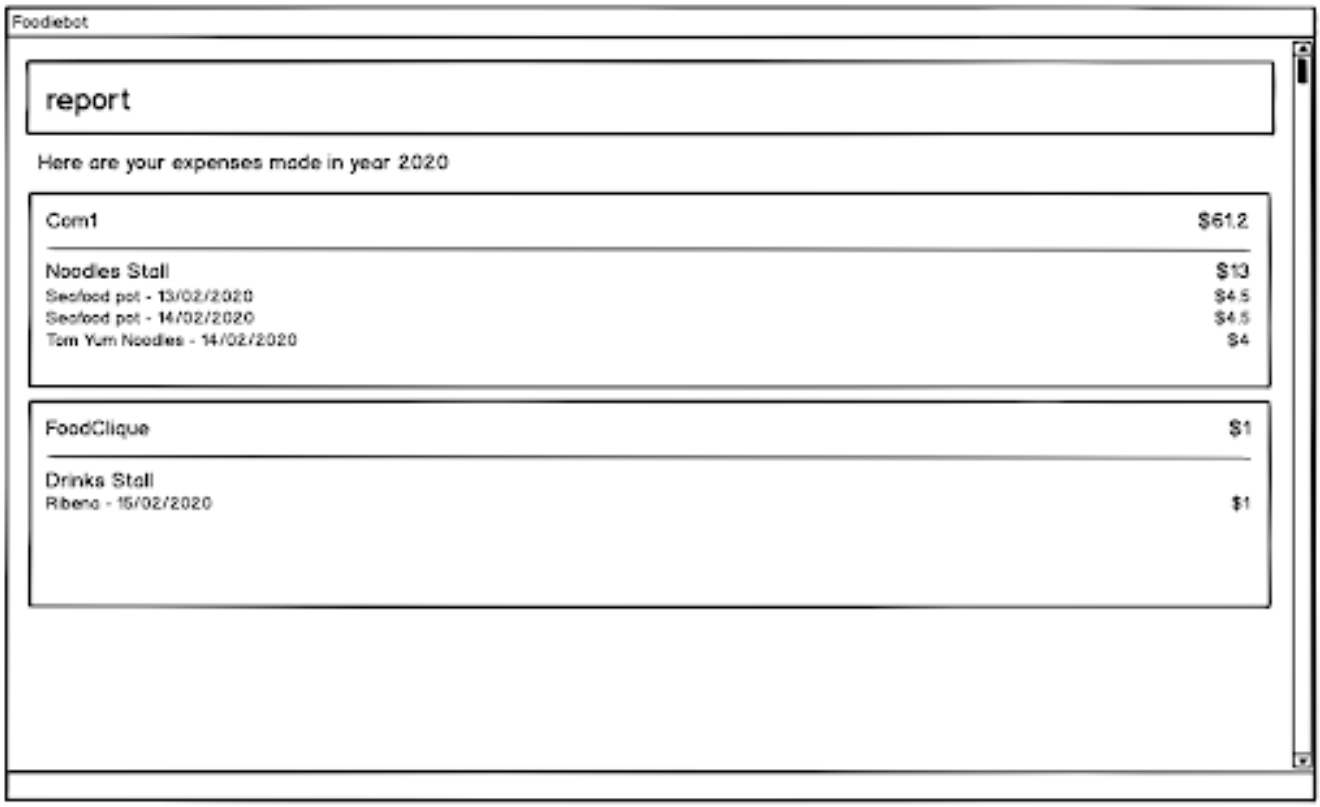
4.10. View Expense Report: `report`

This command generate a report for the spending and food purchases for any period specified.

Formats: `report`

- `[f/FROM_DATE] [t/TO_DATE]` - Generate report from and till the given dates.
  - Example: `report [f/ 12-02-2020] [t/ 30-04-2020]`
- `[w/DATE]` - Generate report for the week of the input date.
  - Example: `report [w/ 12-02-2020]`
- `[m/MONTH]` - Generate report of the input month.
  - Example: `report [m/ jan]`
- `[y/YEAR]` - Generate report of the input year.
  - Example: `report [y/ 2020]`
- `[f/]`, `[t/]`, `[m/]`, `[w/]` and `[y/]` fields are optional.

NOTE	`FROM_DATE` cannot be a future date. `TILL_DATE` cannot be before the FROM_DATE, or the earliest possible date if the [f/] field is empty.



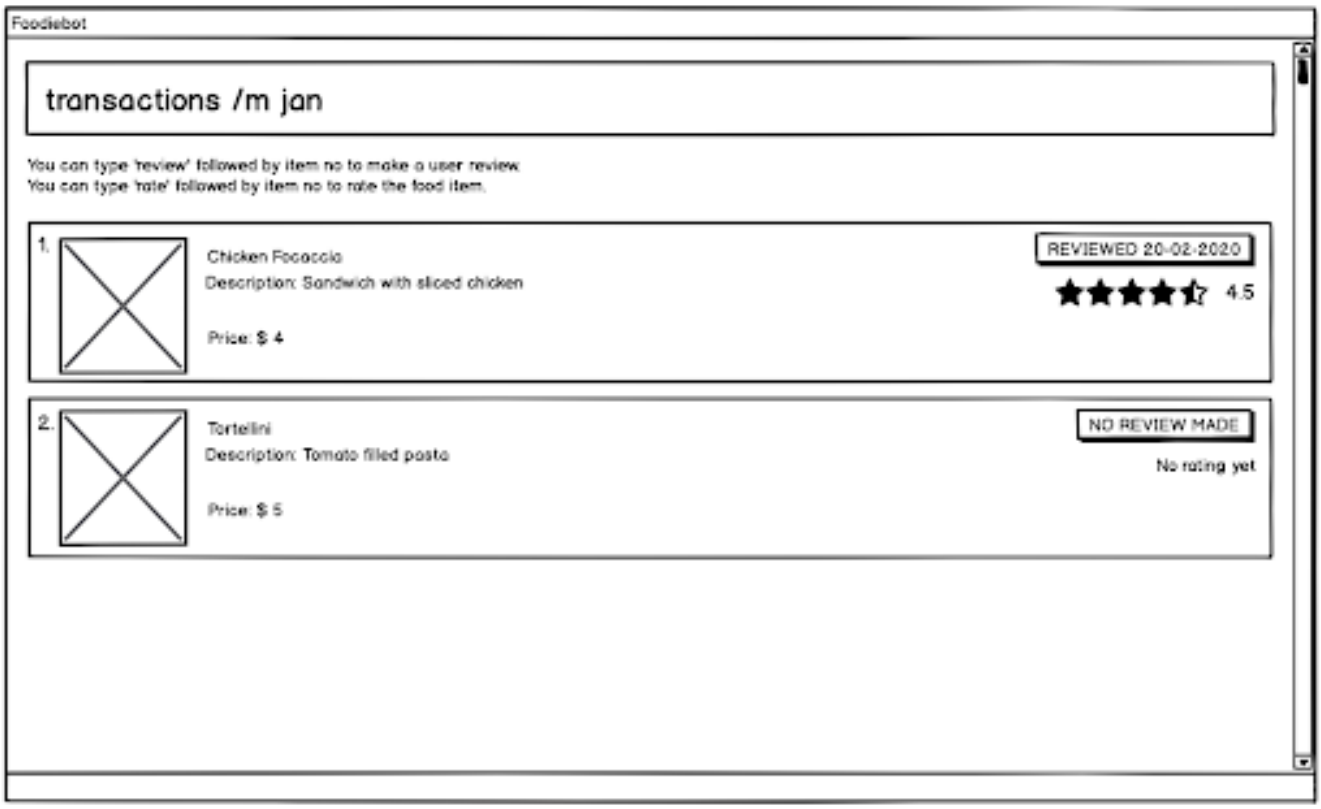
4.11. View Past Transactions: `transaction`

Displays the past transactions using.

Formats: `transaction`

- `[f/FROM_DATE] [t/TO_DATE]` - Generate report from and till the given dates.
  - Example: `report [f/ 12-02-2020] [t/ 30-04-2020]`
- `[w/DATE]` - Display transactions for the week of the input date.
  - Example: transactions [w/ 12-02-2020]
- `[m/MONTH]` - Display transactions of the input month.
  - Example: transactions [m/ jan]
- `[y/YEAR]` - Display transactions of the input year.
  - Example: transactions [y/ 2020]
- `[f/]`, `[t/]`, `[m/]`, `[w/]` and `[y/]` fields are optional.

NOTE	`FROM_DATE` cannot be a future date. `TILL_DATE` cannot be before the FROM_DATE, or the earliest possible date if the [f/] field is empty.

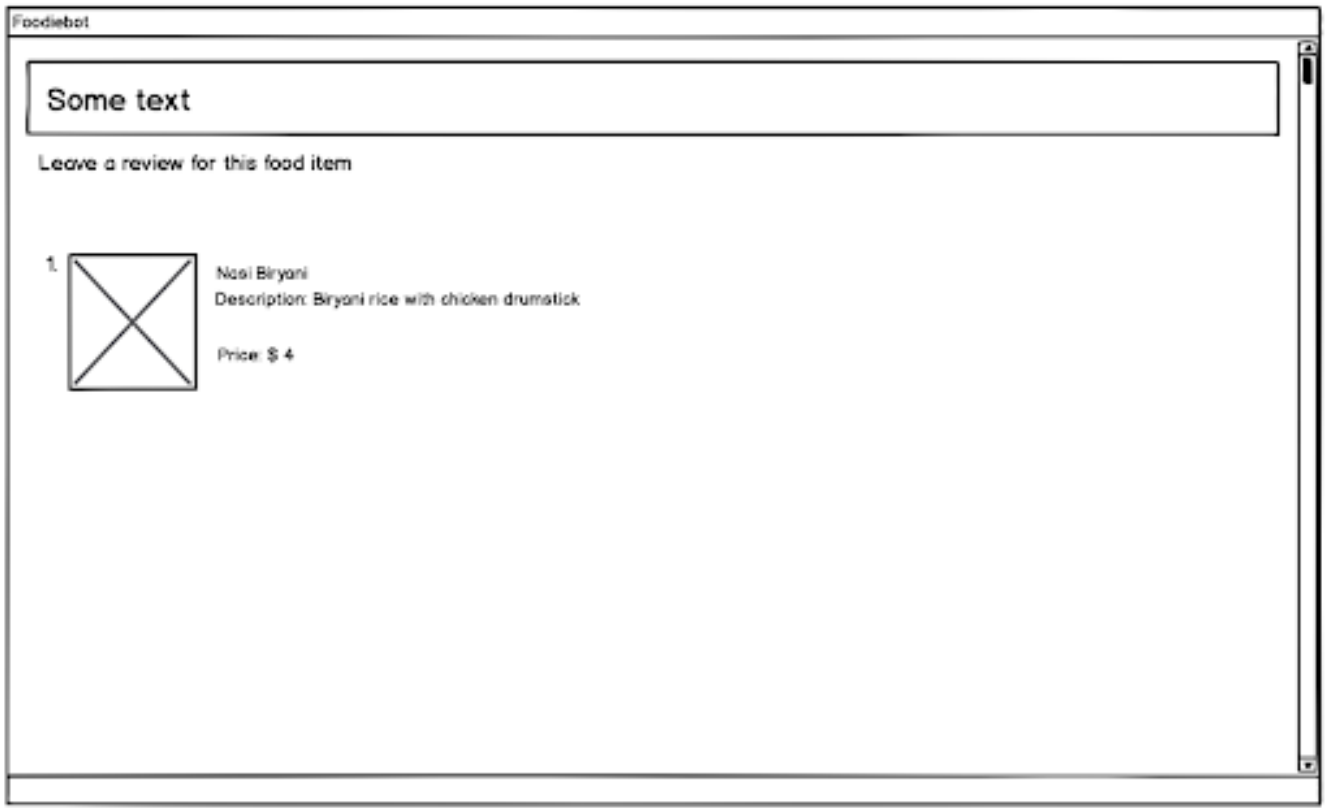


4.12. Review Food Items: `review`

This command allows the user to review food items from the transactions screen as shown in 3.12.

Format: `review INDEX`

> **TIP** User can update existing reviews by using the same command.



4.13. Rate Food Items: `rate`

This command allows user to rate food items from the transactions screen as shown in 3.12.

Format: `rate INDEX`

> **TIP** User can also update existing ratings by using the same command.

5. FAQ

Q: How do I transfer my data to another Computer?
A: Install the app in the other computer and overwrite the empty data file it creates with the file that contains the data of your previous Address Book folder.

Q: Can I write my personal review in other languages?
A:

Q: How can I edit the list of canteen if one canteen close down?
A:

6. Command Summary

Function	Usage	Example
budget set	budget set PERIOD AMOUNT	budget set w/ 9.50
budget view	budget view	
enter	enter CANTEEN_NAME	enter deck
find nearest	find nearest BLOCK_NAME	find nearest COM1
goto	goto CANTEEN_NAME f/ CURRENT_LOCATION	goto The Deck f/ COM1
list	list	
menu	menu KEYWORD	menu chicken
rate	rate INDEX	
randomize	randomize	
report	report	report f/ 12-02-2020 t/ 30-04-2020
review	review INDEX	
select	select INDEX	
transaction	transaction	transaction w/ 12-02-2020