

E.T. (Employees Tracker) - User Guide

1. Introduction	1
2. Quick Start	1
2.1. Setting up	1
3. Features	2
3.1. The User Interface	2
3.2. Data storage	3
4. Commands	3
4.1. Logging in : unlock	4
4.2. Viewing help : help	4
4.3. Adding an employee: add	4
4.4. Listing all persons : list	5
4.5. Editing a person : edit	7
4.6. Rating a person : rate	7
4.7. Reviewing an employee : review	8
4.8. Searching for employees: find	8
4.9. Deleting a person : delete	9
4.10. Selecting a person : select	10
4.11. Listing entered commands : history	10
4.12. Undoing previous command : undo	10
4.13. Redoing the previously undone command : redo	11
4.14. Locking the EmployeeTracker temporary: lock	11
4.15. Setting the application unlock password: setPassword	12
4.16. Sorting the employees by any existing field: sort	12
4.17. Adding an event on one employee's timetable: addEvent	12
4.18. Clearing all entries : clear	13
4.19. Exiting the program : exit	13
4.20. Hiding a notification : hide 1	13
4.21. Dismissing a notification : dismiss	13
4.22. Toggling the notification panel : noti	13
4.23. Reminding an employee through email : email	14
4.24. Reminding an employee through WhatsApp : whatsapp	14
4.25. editing a event [coming in v2.0]	14
4.26. deleting a event [coming in v2.0]	14
4.27. Adding an event in mass to many employees' timetable at once [coming in v2.0]	14
4.28. Having my own timetable [coming in v2.0]	14
4.29. Jumping to email composing with a command [coming in v2.0]	14
4.30. Viewing my to-do-list [coming in v2.0]	14
4.31. Changing the window scheme/theme/skin [coming in v2.0]	14
4.32. Encrypting data files [coming in v2.0]	15
5. FAQ	15
6. Command Summary	15

1. Introduction

Employees Tracker (E.T.) is a **Human Resource Manager** application for managers who **prefer to use a desktop app for managing his/her subordinates**. More importantly, E.T. is **optimized for those who prefer to work with a Command Line Interface (CLI)** while still having the benefits of a Graphical User Interface (GUI). If you can type fast, E.T. can get your employees management tasks done faster than traditional GUI apps.

Interested? Jump to the [Section 2, “Quick Start”](#) to get started. Enjoy!

2. Quick Start

2.1. Setting up

1. Ensure you have Java version **1.8.0_60** or later installed in your Computer.

NOTE

Having any Java 8 version is not enough.
This app will not work with earlier versions of Java 8.

2. Download the latest **et-employeetracker.jar** [here](#).
3. Copy the file to the folder you want to use as the home folder for your Address Book.
4. Double-click the file to start the app. The GUI should appear in a few seconds, as shown in *Figure 1*.

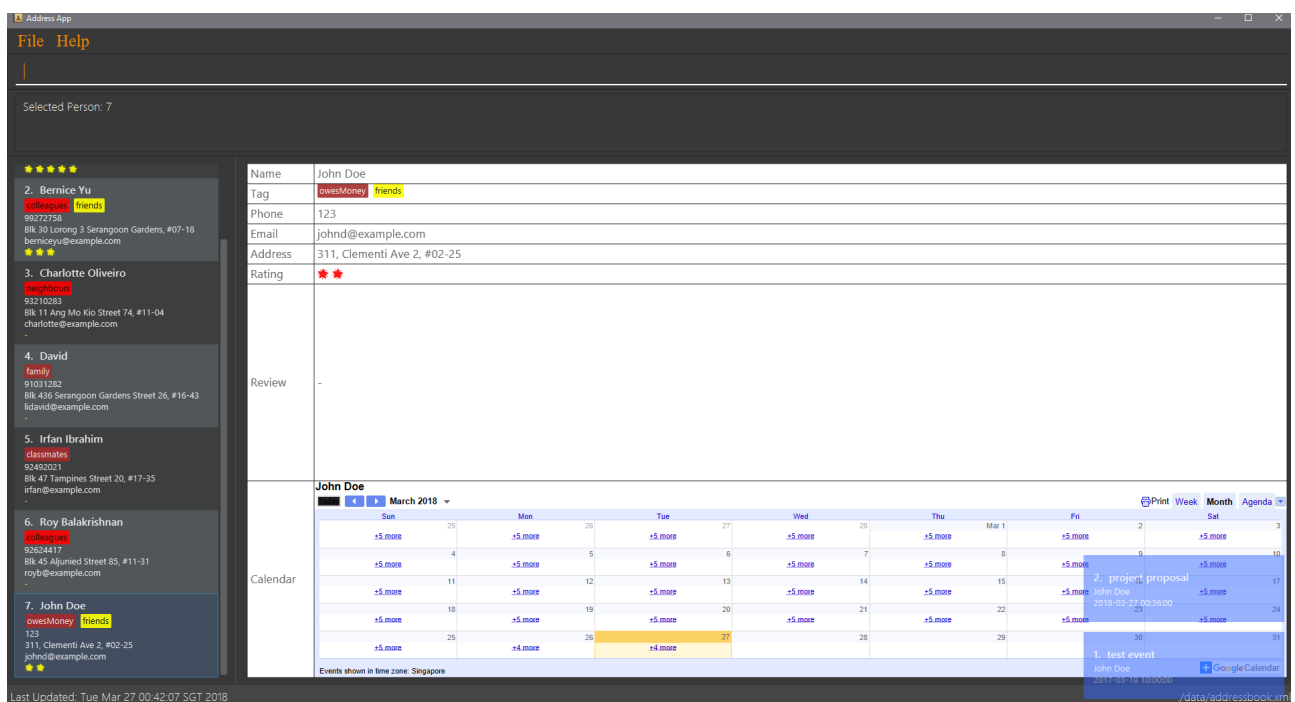


Figure 1. Graphical User Interface (GUI)

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.

Some example commands you can try:

- **list**
lists all employees.
- **add n/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01**
adds an employee named **John Doe** to the Employees Tracker.
- **delete 3**
deletes the 3rd contact shown in the current list.
- **exit**
exits the app.

Refer to [Section 3, “Features”](#) for details of each command.

3. Features

3.1. The User Interface

3.1.1. Knowing the components

Figure 2 shows the 6 major components of the user interface of Employees Tracker:

1. Command Box: where you type in commands
2. Result Display Box: where Employees Tracker shows text result to you
3. Employees List Panel: where the list of all employees is shown
4. Employee’s Profile Panel: where the details of the selected employee is shown
5. Notification cards: where Employees Tracker reminds you about ended events
6. Notification panel: where hidden notification cards are stored

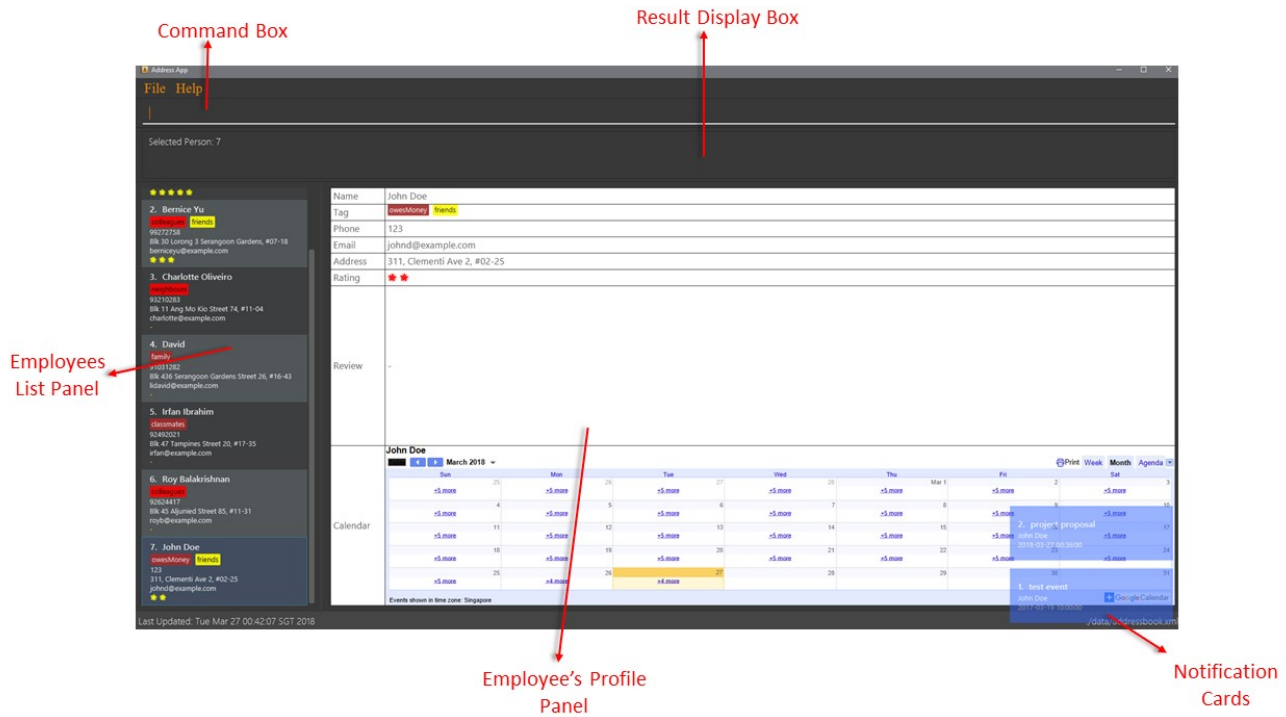


Figure 2. User Interface

3.1.2. Reading the result display box

The result display box shows 3 types of information:

- Error messages, if the command you entered does not exist or does not match the pre-defined format.
- Hints about the format and usage of the command, if you entered a command in the wrong format.
- Statuses of the Employee Tracker, if your command is processed successfully.

NOTE

If the result display box shows text in red, it means that you have entered a non-existing command or a command that does not follow its pre-defined format.

Different tags will perform different colours for clarification

3.2. Data storage

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

There is no need to save manually.

4. Commands

Command Format

- Words in **UPPER_CASE** are the parameters to be supplied by the user e.g. in **add n/NAME**, **NAME** is a parameter which can be used as **add n/John Doe**.
- Items in square brackets are optional e.g **n/NAME [t/TAG]** can be used as **n/John Doe t/friend** or as **n/John Doe**.
- Items with **...** after them can be used multiple times including zero times e.g. **[t/TAG]...** can be used as (i.e. 0 times), **t/friend**, **t/friend t/family** etc.
- Parameters can be in any order e.g. if the command specifies **n/NAME p/PHONE_NUMBER**, **p/PHONE_NUMBER n/NAME** is also acceptable.

4.1. Logging in : **unlock**

Unlocks the Employee Tracker.

Format: **unlock PASSWORD**

NOTE

- The Employees Tracker is locked when Employee Tracker is opened.
- The initial default password is 'admin', refer to [\[Setting the application unlock password: setPassword \[Since v1.3\]\]](#) if you want to change your password.

- Unlock the EmployeeTracker with the PASSWORD which match the application's unlock password.
- PASSWORD is required.
- The PASSWORD is a string that can contain any character.
- The default PASSWORD is **admin**.

Examples:

- **unlock 123**

Unlock the address book with password: 123 (which is also the application's unlock password)

4.2. Viewing help : **help**

Shows the user guide.

Format: **help**

4.3. Adding an employee: **add**

Adds a person to the address book

Format: **add n/NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS [t/TAG]...**

TIP | A person can have any number of tags (including 0)

Examples:

- `add n/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01`
- `add n/Betsy Crowe t/friend e/betsycrowe@example.com a/Newgate Prison p/1234567 t/criminal`

4.4. Listing all persons : `list`

Shows a list of all persons in the address book.

Format: `list`

- List command will shows you a list of persons, as shown in Figure 3.3.
- Each entry of the list contains information of a person.
- In each entry, from top to down, is the name, tags, phone number, address, email address and rating of that person.

NOTE | Rating will be shown as - if the person is yet to be rated.

2. Bernice Yu

colleagues friends

99272758

Blk 30 Lorong 3 Serangoon Gardens, #07-18

berniceyu@example.com



3. Charlotte Oliveira

neighbours

93210283

Blk 11 Ang Mo Kio Street 74, #11-04

charlotte@example.com

-

4. David

family

91031282

Blk 436 Serangoon Gardens Street 26, #16-43

lidavid@example.com

-

5. Irfan Ibrahim

classmates

92492021

Blk 47 Tampines Street 20, #17-35

irfan@example.com

-

6. Roy Balakrishnan

colleagues

92624417

Blk 45 Aljunied Street 85, #11-31

royb@example.com

-

7. John Doe

owesMoney friends

123

311, Clementi Ave 2, #02-25

johnd@example.com



4.5. Editing a person : **edit**

Edits an existing person in the address book.

Format: **edit** INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [r/RATING] [t/TAG]...

- Edits the person at the specified **INDEX**. The index refers to the index number shown in the last person listing. The index **must be a positive integer** 1, 2, 3, ...
- At least one of the optional fields must be provided.
- Existing values will be updated to the input values.
- When editing tags, the existing tags of the person will be removed i.e adding of tags is not cumulative.
- You can remove all the person's tags by typing **t/** without specifying any tags after it.

Examples:

- **edit 1 p/91234567 e/johndoe@example.com**
Edits the phone number and email address of the 1st person to be **91234567** and **johndoe@example.com** respectively.
- **edit 2 n/Betsy Crower t/**
Edits the name of the 2nd person to be **Betsy Crower** and clears all existing tags.

4.6. Rating a person : **rate**

Updates the rating of an existing person in the address book.

Format: **rate** INDEX RATING

- Rates the person at the specified **INDEX**. The index refers to the index number shown in the last persons listing. The index **must be a positive integer** 1, 2, 3, ...
- Both INDEX and RATING must be provided.
- Existing rating will be updated to the input RATING.
- RATING must be a positive integer between 1 and 5, i.e. 1, 2, 3, 4 or 5.

NOTE

- A person will have a null rating by default upon added. This implies that the person is yet to be rated.
- Null rating is shown as **-**.
- You cannot assign null rating to a person. If you want to revert a person back to null rating, please refer to [Section 4.12, “Undoing previous command : undo”](#)

Examples:

- `list`
`rate 1 5`
Gives the 1st person in the last persons listing a 5-stars rating.
- `find Alex`
`rate 2 2`
Gives the 2nd person in the list of Alex(s) a 2-star rating.

4.7. Reviewing an employee : `review`

Assigns a review to an existing employee in the address book.

Format: `review INDEX`

- Rate the employee at the specified `INDEX`.
The index refers to the index number shown in the last persons listing.
The index **must be a positive integer**, i.e. 1, 2, 3, ...
- A pop-up dialog box will appear after user executes the command.
The dialog box will prompt the user to enter the review.
- The review can contain any character without length limitation.
- A review will be tied to a single reviewer. [Implemented in V1.4]

NOTE

- A employees will have a null review by default upon added. This implies that the employee is yet to be reviewed.
- Null review is shown as `-`.

Examples:

- `list`
`review 1`
Gives the 1st employee in the last persons listing a review.
The review will be based on the input in the pop-up dialog box.
- `find n/Alex`
`rate 2`
Gives the 2nd employee in the list of Alex(s) a review.
The review will be based on the input in the pop-up dialog box.

4.8. Searching for employees: `find`

Finds employees whose name, tags, rating, or all of those contain any of the given keyphrases.

Format: `find (n/NAME_KEYPHRASE | t/TAG_KEYPHRASE | r/RATING_KEYPHRASE) [n/NAME_KEYPHRASE]... [t/TAG_KEYPHRASE]... [r/RATING_KEYPHRASE]...`

- The search is case insensitive, e.g. `n/hANs` will match `Hans`
- The order of the keyphrase does not matter, e.g. `n/Hans Bo` will match `Bo Hans`
- The search does not match exactly all the words in a phrase in sequential order, but only finds any phrase that contains the specified keyphrase's words without sequential order, e.g. `n/Hans Bo` will match `Jonathan Bo Hans`
- The search only matches full word, e.g. `n/Han` will not match `Hans`
- Searching by multiple criteria will yield the result of employees that match all the criterias, e.g.
 - `n/Hans Bo t/Sales` will match `Hans Bo` with tag `Friends` but not `Hans Bo` with tag `Enemy` or `John Doe` with tag `Friends`
 - `n/Hans Bo n/John Doe t/Sales t/Marketing` will match only `Hans Bo | Sales` or `Hans Bo | Marketing` or `John Doe | Sales` or `John Doe | Marketing`

Examples:

- `find n/John`
Returns all employees having name `John`.
e.g. `john` and `John Doe`
- `find n/Betsy Tim John`
Returns all employees with name that contains `Betsy Tim John` without any sequential order.
e.g. `Caroline John Tim Betsy` and `Betsy Tim John`
- `find n/Betsy n/Tim n/John`
Returns all employees with name that contains either `Betsy`, `Tim`, or `John`
e.g. `Betsy`, `Tim`, `John`, `Betsy Toe`, `John Cook`, and `Tim John`
- `find n/John t/Friends Colleagues r/3`
Returns all employees with name `John`, both tags `Friends` and `Colleagues`, and rating of 3
e.g. `John | Friends, Colleagues | 3`
- `find n/John t/Friends t/Colleagues`
Returns all employees having both names `John` and either tags `Friends` or `Colleagues`
e.g. `John Doe | Friends`, `John Dick | Colleagues`, `John | Friends, Colleagues`, and `John Dare | Friends, Colleagues, OweMoneys`

4.9. Deleting a person : `delete`

Deletes the specified person from the address book.

Format: `delete INDEX`

- Deletes the person at the specified `INDEX`.
- The index refers to the index number shown in the most recent listing.
- The index **must be a positive integer** 1, 2, 3, ...

Examples:

- `list`
`delete 2`
Deletes the 2nd person in the address book.
- `find Betsy`
`delete 1`
Deletes the 1st person in the results of the `find` command.

4.10. Selecting a person : `select`

Selects the person identified by the index number used in the last person listing.

Format: `select INDEX`

- Selects the person and loads the Google search page the person at the specified `INDEX`.
- The index refers to the index number shown in the most recent listing.
- The index **must be a positive integer** `1, 2, 3, ...`

Examples:

- `list`
`select 2`
Selects the 2nd person in the address book.
- `find Betsy`
`select 1`
Selects the 1st person in the results of the `find` command.

4.11. Listing entered commands : `history`

Lists all the commands that you have entered in reverse chronological order.

Format: `history`

NOTE

Pressing the up and down arrows will display the previous and next input respectively in the command box.

4.12. Undoing previous command : `undo`

Restores the address book to the state before the previous *undoable* command was executed.

Format: `undo`

NOTE

Undoable commands: those commands that modify the address book's content (`add`, `delete`, `edit` and `clear`).

Examples:

- `delete 1`
`list`
`undo` (reverses the `delete 1` command)
- `select 1`
`list`
`undo`
The `undo` command fails as there are no undoable commands executed previously.
- `delete 1`
`clear`
`undo` (reverses the `clear` command)
`undo` (reverses the `delete 1` command)

4.13. Redoing the previously undone command : `redo`

Reverses the most recent `undo` command.

Format: `redo`

Examples:

- `delete 1`
`undo` (reverses the `delete 1` command)
`redo` (reapplies the `delete 1` command)
- `delete 1`
`redo`
The `redo` command fails as there are no `undo` commands executed previously.
- `delete 1`
`clear`
`undo` (reverses the `clear` command)
`undo` (reverses the `delete 1` command)
`redo` (reapplies the `delete 1` command)
`redo` (reapplies the `clear` command)

4.14. Locking the EmployeeTracker temporary: `lock`

Locks the Employee Tracker, so that the data is protected without the need of closing the program or shutting down your computer.

Format: `lock`

- The `Employees List Panel` and `Employee's Profile Panel` will be emptied when Employees Tracker is locked.
- Employee Tracker will not respond to any command other than `unlock` when it is locked.

Examples:

- `lock`

Lock the address book.

4.15. Setting the application unlock password: `setPassword`

Sets Employees Tracker's password.

Format: `setPassword OLD_PASSWORD NEW_PASSWORD`

- Set Employees Tracker's password from OLD_PASSWORD to NEW_PASSWORD.
- OLD_PASSWORD and NEW_PASSWORD are required, and OLD_PASSWORD should match current application unlock password.
- OLD_PASSWORD and NEW_PASSWORD are of String type that can contain any character.
- The default OLD_PASSWORD is `admin`.

Examples:

- `setPassword 123 qwe`
Set Employees Tracker's password from '123' (current application password) to 'qwe'.

4.16. Sorting the employees by any existing field: `sort`

Sorts existing employees by any field

Format: `sort FIELD`

- Sort the employees by any existing field in alphabetical order, note that rate field will be sorted in descending order.
- The field entered must be one of the following: `name`, `phone`, `email`, `address`, `rate`, `tag`.
- Sorting is case-insensitive.
- The sorted results is stored permanently. If you want to revert to the previous ordering, please refer to [Section 4.12](#), “Undoing previous command : `undo`”

Examples:

- `sort name`
Sort the employees by name in alphabetical order (case-insensitive).
- `lock rate`
Sort the employees by rating in descending order.

4.17. Adding an event on one employee's timetable: `addEvent`

Add an event to the employee identified by the index number used in the last person listing.

Format: `addEvent` `INDEX` `title/TITLE` `loca/LOCATION` `stime/STRATTIME` `etime/ENDTIME`
`descrip/DESCRIPTION`

- `STARTTIME` and `ENDTIME` must be in the format `YYYY-MM-DDTHH-MM-SS`.
- The index refers to the index number shown in the most recent listing.
- The index must be a positive integer 1, 2, 3, ...

Examples:

- `list`
`addEvent 2 title/Test Event loca/NUS, Singapore stime/2017-04-09T19:00:00 etime/2018-04-09T21:00:00 descrip/A Test Event`
Adds the event to 2nd employee's timetable in the list.

4.18. Clearing all entries : `clear`

Clears all entries from the address book.

Format: `clear`

4.19. Exiting the program : `exit`

Exits the program.

Format: `exit`

4.20. Hiding a notification : `hide` 1

Hides the notification card identified by the index number used in notification cards.

Format: `hide` `INDEX`

NOTE	The hidden notification card will be stored in the notification panel.
-------------	--

4.21. Dismissing a notification : `dismiss`

Dismisses the notification identified by the index number used in notification cards.

Format: `dismiss` `INDEX`

4.22. Toggling the notification panel : `noti`

Shows the notification panel if it is hidden; hides the notification panel if it is shown.

Format: `noti`

4.23. Reminding an employee through email : email

Opens up the email app of the computer to send a reminder email about the notification identified by the index number used in notification cards.

Format: email INDEX

NOTE

- You will be brought straight to the email composing interface of your email app.
- The recipient of the email will be set to the email address of the employee who was assigned the event.

4.24. Reminding an employee through WhatsApp : whatsapp

Opens up the WhatsApp webpage to send a reminder message about the notification identified by the index number used in notification cards.

Format: whatsapp INDEX

NOTE

- You will be brought straight to the WhatsApp webpage that sends message to the phone number of the employee who was assigned the event.
- You may be required to scan a QR code from your phone to login to WhatsApp webpage.

4.25. editing a event [coming in v2.0]

4.26. deleting a event [coming in v2.0]

4.27. Adding an event in mass to many employees' timetable at once [coming in v2.0]

4.28. Having my own timetable [coming in v2.0]

4.29. Jumping to email composing with a command [coming in v2.0]

4.30. Viewing my to-do-list [coming in v2.0]

4.31. Changing the window scheme/theme/skin [coming in v2.0]

4.32. Encrypting data files [coming in v2.0]

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.

6. Command Summary

- **Add** : `add n/NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS [t/TAG]...`
e.g. `add n/James Ho p/22224444 e/jamesho@example.com a/123, Clementi Rd, 1234665 t/friend t/colleague`
- **Clear** : `clear`
- **Delete** : `delete INDEX`
e.g. `delete 3`
- **Edit** : `edit INDEX [n/NAME] [p/PHONE_NUMBER] [e/EMAIL] [a/ADDRESS] [t/TAG]...`
e.g. `edit 2 n/James Lee e/jameslee@example.com`
- **Review** : `review INDEX`
e.g. `review 2`
- **Find** : `find (n/NAME_KEYPHRASE | t/TAG_KEYPHRASE | r/RATING_KEYPHRASE) [n/NAME_KEYPHRASE]... [t/TAG_KEYPHRASE]... [r/RATING_KEYPHRASE]...`
e.g. `find n/James n/Jake Black t/Sales t/Marketing Senior r/3`
- **List** : `list`
- **Help** : `help`
- **Select** : `select INDEX`
e.g. `select 2`
- **History** : `history`
- **Undo** : `undo`
- **Redo** : `redo`