

NeoXPro Manager - User Guide

| | |
|--|----|
| 1. Introduction | 1 |
| 2. About | 1 |
| 3. Quick Start | 1 |
| 4. Features | 2 |
| 4.1. Viewing help : help | 2 |
| 4.2. Adding a person: add | 2 |
| 4.3. Listing all persons : list | 3 |
| 4.4. Editing a person : edit | 3 |
| 4.5. Update a tag name on all contacts: updateTag (coming in v2.0) | 3 |
| 4.6. Locating persons by name: find | 4 |
| 4.7. Locating persons by phone number: phone (since v1.1) | 4 |
| 4.8. Locating persons by tags: findTag (since v1.1) | 5 |
| 4.9. Deleting a person : delete | 6 |
| 4.10. Favoriting a person : fav | 6 |
| 4.11. Unfavoriting a person : unfav. | 7 |
| 4.12. Listing all favorited persons : listfav | 7 |
| 4.13. Selecting a person : select | 7 |
| 4.14. Locating a person's address : locate (since v1.3) | 9 |
| 4.15. Adding an event: addEvent (since v1.2) | 9 |
| 4.16. Deleting an event : deleteEvent (since v1.4) | 10 |
| 4.17. Listing entered commands : history | 10 |
| 4.18. Undoing previous command : undo | 10 |
| 4.19. Redoing the previously undone command : redo | 11 |
| 4.20. Clearing all entries : clear | 11 |
| 4.21. Exporting the data: export | 11 |
| 4.22. Special features | 12 |
| 4.23. Exiting the program : exit | 13 |
| 4.24. Saving the data | 13 |
| 5. FAQ | 13 |
| 6. Command Summary | 13 |

1. Introduction

Are you a busy university student having trouble keeping track of your social and academic life? If yes, NeoXPro Manager might just be the app for you.

It is a simple but powerful tool for managing your everyday tasks and keeping track of your friends and acquaintances. With its intuitive interface, user would have no issue navigating the application. NeoXPro also remind you of upcoming events and tasks, so you will never miss those important meetings or assignment deadlines again.

2. About

This guide provide a detailed instruction of how to use NeoXPro and all its features. It also comes with a summary of commands for users are already familiarized with NeoXPro and just need a quick reminder.

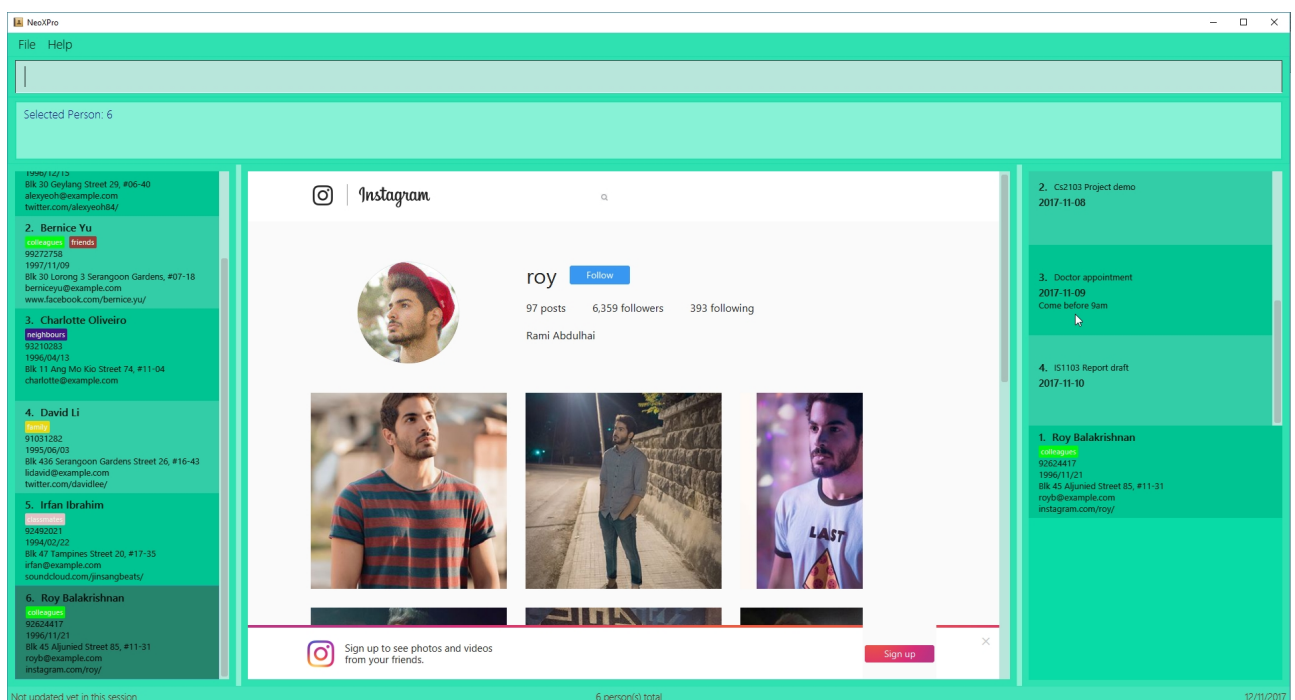
3. Quick Start

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 [addressbook.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.



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 contacts
 - **addn/John Doe p/98765432 e/johnd@example.com b/1993/12/12 a/John street, block 123, #01-01** : adds a contact named **John Doe** to the Address Book.
 - **delete3** : deletes the 3rd contact shown in the current list
 - **exit** : exits the app
7. Refer to the **Features** section below for details of each command.

4. Features

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. Viewing help : **help**

Format: **help**

4.2. Adding a person: **add**

Adds a person to the address book

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

TIP

A person can have any number of tags (including 0). All fields are optional except the person name.

Examples:

- **add n/John Doe p/98765432 e/johnd@example.com b/1995/5/21 a/John street, block 123, #01-01 pr/www.facebook.com/john**
- **add n/Betsy Crowe t/friend e/betsycrowe@example.com a/Newgate Prison p/1234567 b/1999/10/10 t/criminal**

4.3. Listing all persons : **list**

Shows a list of all persons in the address book.

Format: **list**

4.4. Editing a person : **edit**

Edits an existing person in the address book.

Format: **edit** INDEX [n/NAME] [p/PHONE] [e/EMAIL] [b/BIRTHDAY] [a/ADDRESS] [pr/PROFILE] [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.

4.5. Update a tag name on all contacts: **updateTag** (coming in v2.0)

Update a specified tag name with a new tag name on all contacts in addressbook.

Format: 'updateTag [t/oldTAG] [t/newTAG]'

- 'oldTag' must already exist in address book.
- 'newTag' can be a tag that already exists in address book.
- Updates the tag 'oldTAG' with the new tag 'newTAG'.
- After this command is executed, 'oldTAG' no longer exists and the tag 'oldTAG' in any contact will be updated with 'newTAG'.

Examples:

- **'updateTag friend highschool_friend'+** Any contact with tag 'friend' now has tag 'highschool_friend' and no longer has tag 'friend'

4.6. Locating persons by name: **find**

Finds persons whose names contain fully or partially any of the given keywords.

Format: **find** KEYWORD [MORE_KEYWORDS]

- The search is case insensitive. e.g **hans** will match **Hans**
- The order of the keywords does not matter. e.g. **Hans Bo** will match **Bo Hans**
- Only the name is searched.
- Partial words will be matched e.g. **Han** will match **Hans**
- Persons matching at least one keyword will be returned (i.e. **OR** search). e.g. **Hans Be** will return **Hansell Gruber, Robert Lee**

Examples:

- **find John**
Returns **john** and **John Doe**
- **find Betsy Tim John**
Returns any person having names **Betsy, Tim, or John**

4.7. Locating persons by phone number: **phone** (since v1.1)

Finds person whose phone numbers partially match with a number in the specified list

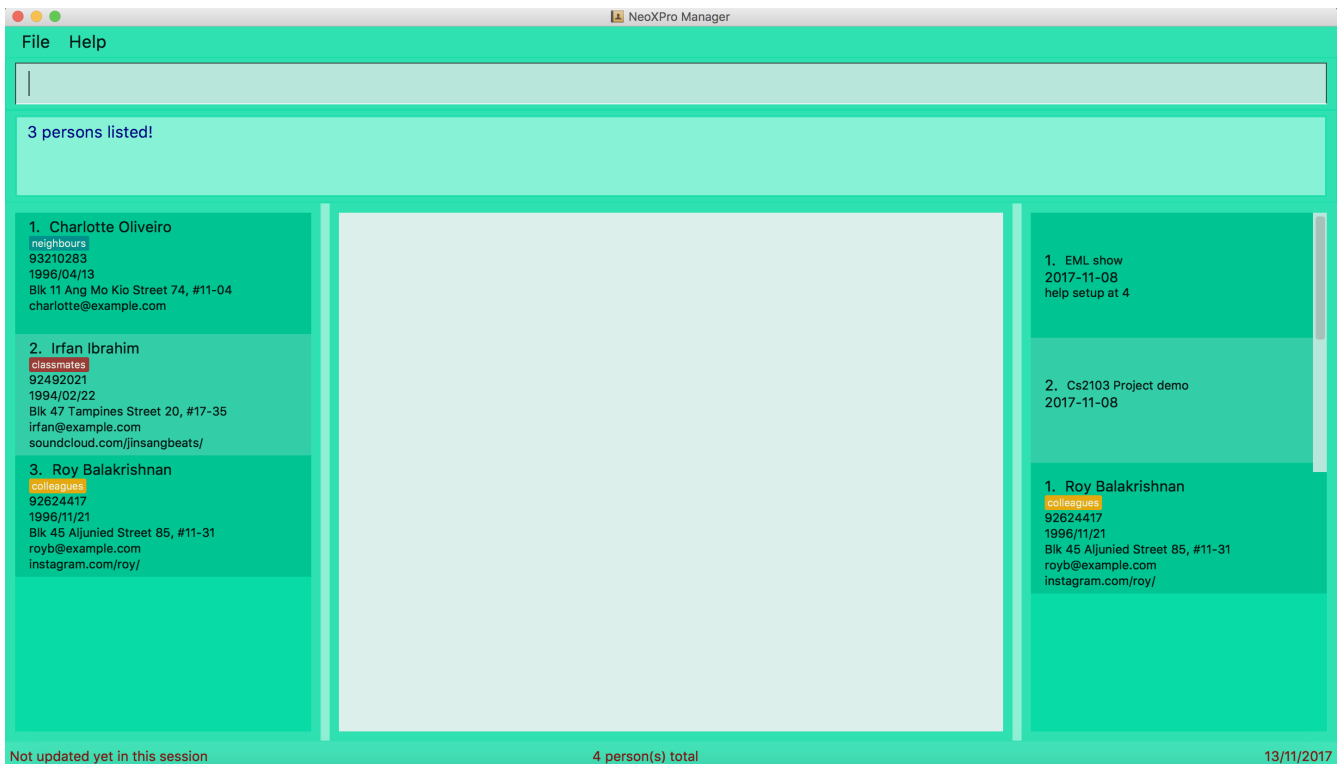
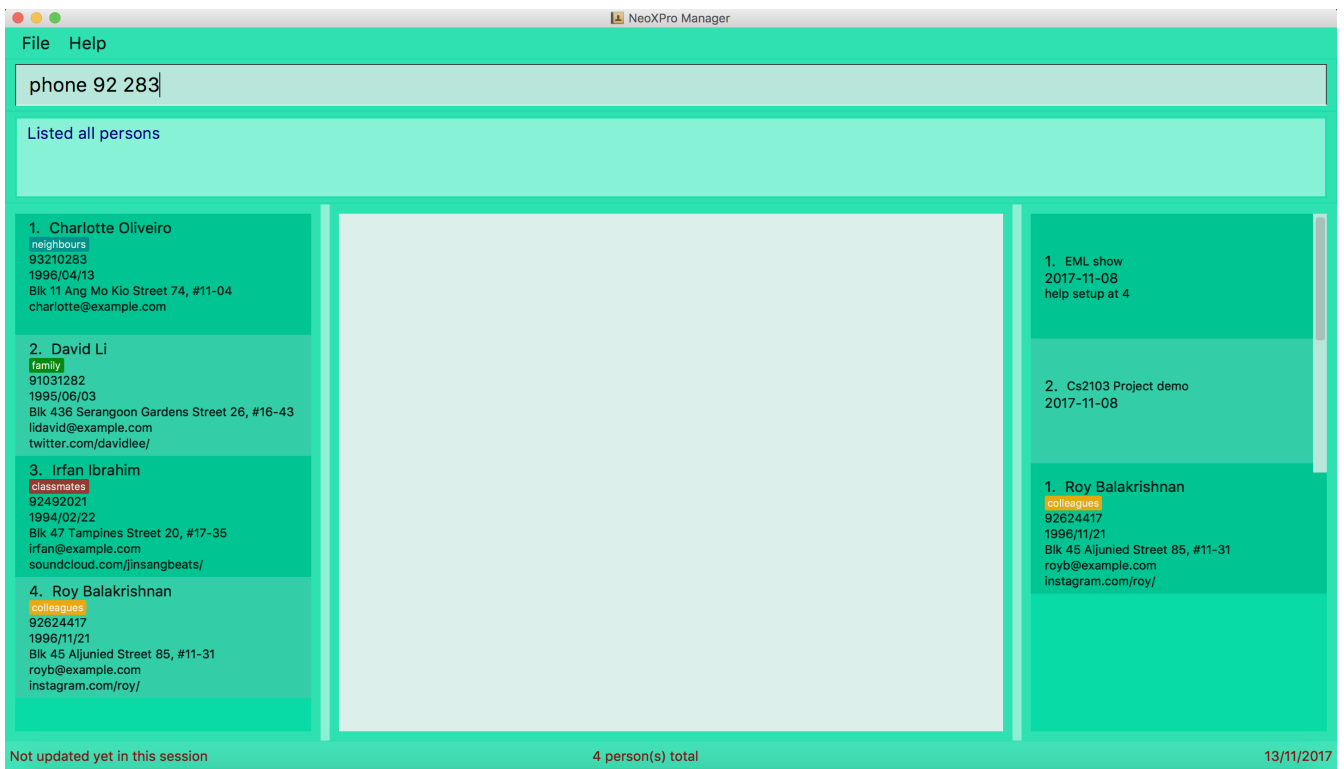
Format: **phone** NUMBER [MORE_NUMBERS]

- The order of numbers being queried does not matter.
- Only the phone number is searched.
- Partial string number will be matched with phones e.g. **12345** will match **123456**

Examples:

- **phone 92**
Returns persons with phone numbers containing 92.
- **phone 92 65**
Returns persons with phone numbers containing **92** or **283**.

The second example is illustrated below:



4.8. Locating persons by tags: findTag (since v1.1)

Finds persons whose tags include all of the given keywords.

Format: findTag KEYWORD [MORE_KEYWORDS]

- The search is case sensitive. e.g `cs2103t` will not match `CS2103T`
- The order of the keywords does not matter.
- Only the tag is searched.
- Only full words will be matched e.g. `cs210` will not match `cs2101`
- Only persons matching at all keywords will be returned (i.e. **AND** search). e.g. e.g. `friend cs2103t` will match all contacts that have both `friend` and `cs2103t` tags

Examples:

- `findTag friend`
Returns any person with a `friend` tag
- `findTag CS2103T friend`
Returns all persons with both `friend` and `CS2103T` tag.

4.9. Deleting a person : `delete`

Deletes a list of specified persons from the address book.

Format: `delete INDEX [MORE_INDICES]`

- Deletes the persons at the specified `INDEX's.
- 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 1`
Deletes the 1st and 2nd person in the address book.
- `find Betsy`
`delete 1`
Deletes the 1st person in the results of the `find` command.

4.10. Favoriting a person : `fav`

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

Format: `fav INDEX`

- Favorites the person at the specified **INDEX**.
- If the person has already been favorited, the system will give you a reminder.
- The index refers to the index number shown in the most recent listing.
- The index **must be a positive integer** 1, 2, 3, ...
- A star will appear next to the person name if he is favorited.

Example:

- **fav 2**
Favorite the 2nd listed person in the address book.

4.11. Unfavoriting a person : **unfav**

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

Format: **Unfav INDEX**

- Unfavorites the person at the specified **INDEX**.
- If the person is not favorited/already unfavorited, the system will give you a reminder.
- The index refers to the index number shown in the most recent listing.
- The index **must be a positive integer** 1, 2, 3, ...
- The star next to the person name will disappear if he is unfavorited.

Example:

- **Unfav 2**
Undo favorite the 2nd listed person in the address book.

4.12. Listing all favorited persons : **listfav**

Shows a list of all favorited persons in the address book.

Format: **listfav**

4.13. Selecting a person : **select**

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

Format: **select INDEX**

- Selects the person at the specified **INDEX**.
- If the person has a valid profile page, address book loads that profile page. Otherwise, it loads the Google search page of the person.
- The index refers to the index number shown in the most recent listing.
- The index **must be a positive integer** 1, 2, 3, ...

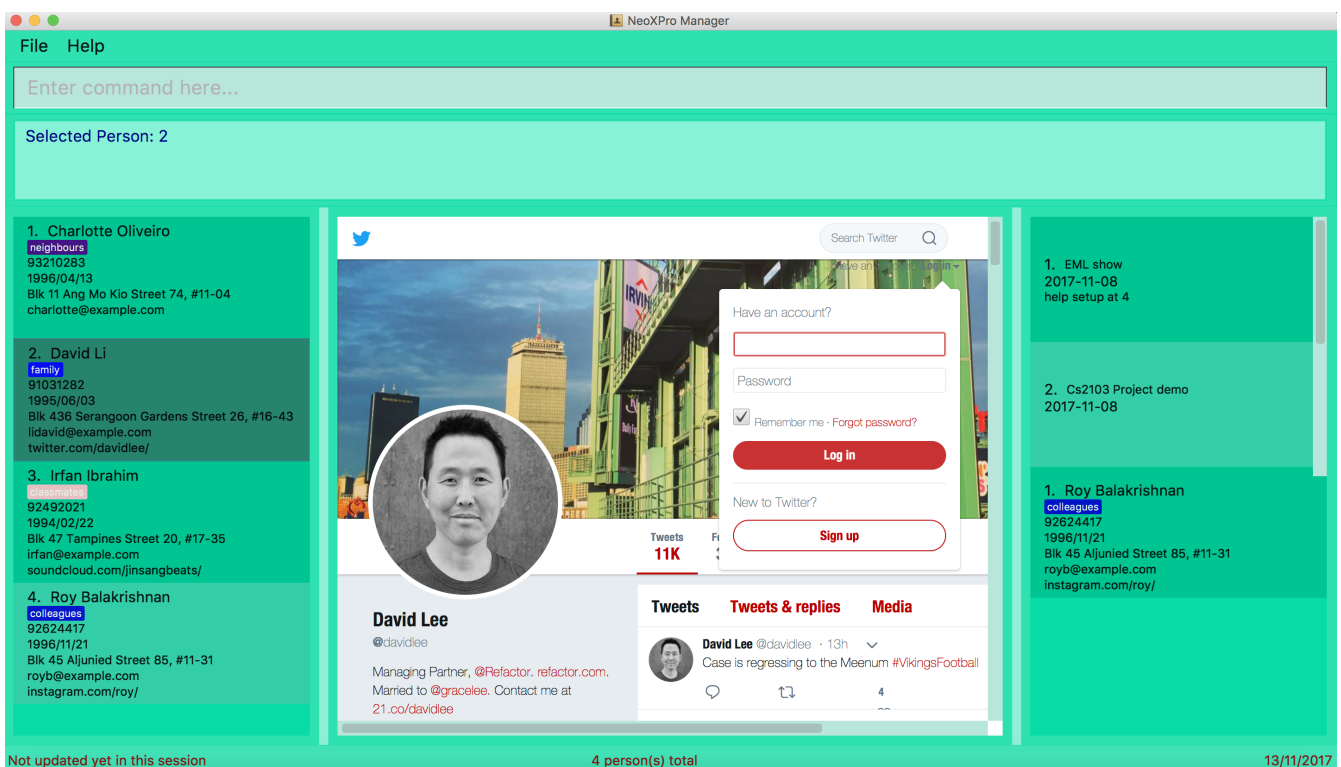
Examples:

• **list**

select 2

Selects the 2nd person in the address book.

The 2nd person in the list has the profile page "twitter.com/davidlee"

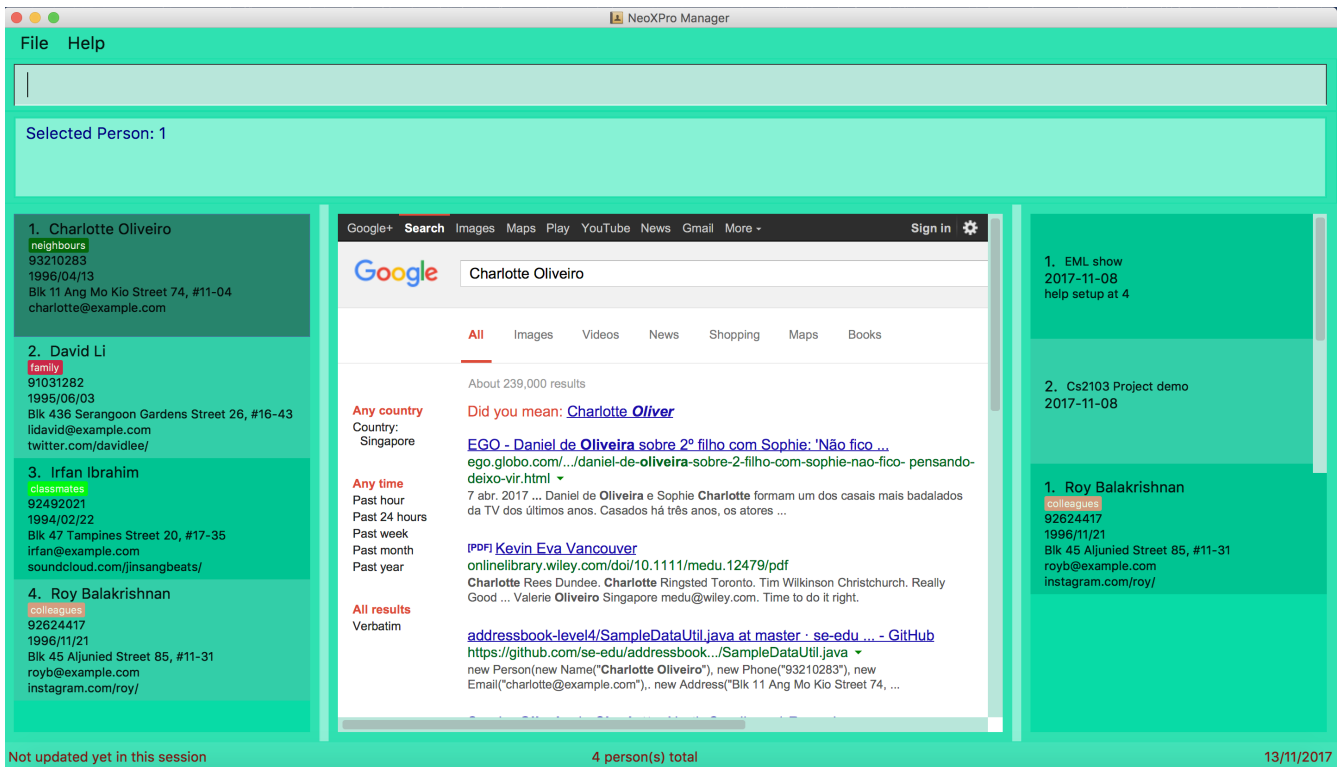


• **find John**

select 1

Selects the 1st person in the results of the **find** command.

The 1st person in the list does not have the profile page in address book.



4.14. Locating a person's address : **locate** (since v1.3)

Locate the person's address identified by the index number used in the last person listing.

Format: **locate** INDEX

- Loads the Google map page of the person's address 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**
locate 2
Locates the 2nd person's address in the address book.
- **find Betsy**
locate 1
Locates the 1st person in the results of the **find** command.

4.15. Adding an event: **addEvent** (since v1.2)

Adds a person to the address book

Format: **addEvent** n/EVENT_NAME d/DD-MM-YY de/[EXTRA]

TIP

The event extra description is optional, can use as a to-do list.
The event date is **yyyy-mm-dd** format.

Examples:

- `addEvent n/Return John 5 bucks d/2017-12-17 de/lunch money`
Add a new event with description.
- `addEvent n/Project Meeting d/2017-11-25 de/`
Add a new event with empty description.

4.16. Deleting an event : `deleteEvent` (since v1.4)

Deletes the specified event from the address book.

Format: `deleteEvent INDEX`

- Deletes the event 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:

- `deleteEvent 2`
Deletes the 2nd listed event in the address book.

4.17. Listing entered commands : `history`

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

Format: `history`

NOTE

Pressing the `↑` and `↓` arrows will display the previous and next input respectively in the command box.

4.18. 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)

4.19. 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)

4.20. Clearing all entries : **clear**

Clears all entries from the address book.

Format: **clear**

4.21. Exporting the data: **export**

Address book data is exported in text form to the input file path. Format: **export [File_Path]**

- The [File_Path] must contain the file name of the exported file. E.g. **export c:\documents and settings\all users\desktop\exportedData**
- The [File_Path] can be just the file name of the exported file instead of the file path. In this case, **export** command will export the file [File_Path] to the current directory.
- The [File_Path] can be blank. In this case, **export** command will export the file with default name "exportFile.txt" to the current directory.

Examples:

- **export c:\documents and settings\all users\desktop\exportedData**

Exports the file "exportedData" to the specified path.

- **export exportedData**

Exports the file "exportedData" to the current directory of NeoXPro.

- **export**

Exports the file "exportFile.txt" to the current directory of NeoXPro.

4.22. Special features

4.22.1. Coming birthday list

The list will show upcoming birthdays of persons in the address book. The list will contains people who are having their birthday in current month. At the end of each month(the last day), the list will show people who are having their birthday in the next month.

Take note: February is a special month which end at 28th or 29th. Only in the cases of 29th of February, people who are having their birthday in March will be shown. Please check the list for upcoming birthday on 1st of March when February is ended at 28th.

Take note: If a new person with relevant birthday is added or a person's birthday is edited to a relevant birthday which supposed to be shown in the list, you should restart the software before seeing changes are made in coming birthday list.



Examples:

- Current date: 2017/06/12
Amy with birthday: 1993/06/21
John with birthday: 1992/06/02
Only Amy is shown in the list

- Current date: 2017/11/30
Amy with birthday: 1998/11/12
John with birthday: 1990/12/01
Jolin with birthday: 1980/12/31
John and Jolin are shown in the list
- Current date: 2017/07/08
Amy with birthday: 1990/07/17
Add a new person, John with birthday: 1990/07/10
Only Amy is shown in the list
Restart the software
Both Amy and John are shown in the list

4.22.2. Auto-reminder (since v1.5)

NeoXPro Manager will remind the user events occur on the day when when starting the application.

4.23. Exiting the program : `exit`

Exits the program.

Format: `exit`

4.24. Saving the data

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

There is no need to save manually.

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 b/BIRTHDAY a/ADDRESS [t/TAG]...`
e.g. `add n/James Ho p/22224444 e/jamesho@example.com b/1993/01/03 a/123, Clementi Rd, 1234665 t/friend t/colleague`
- **Add an event** `addEvent n/EVENT_NAME d/YYYY-MM-DD de/[DESCRIPTION]`
e.g. `addEvent n/Return 5$ d/2017-08-17 de/lunch money`
- **Clear** : `clear`
- **Delete a contact** : `delete INDEX`
e.g. `delete 3`

- **Delete an event** : deleteEvent INDEX
e.g. deleteEvent 2
- **Edit** : edit INDEX [n/NAME] [p/PHONE_NUMBER] [e/EMAIL] [b/BIRTHDAY] [a/ADDRESS] [pr/PROFILE] [t/TAG]...
e.g. edit 2 n/James Lee e/jameslee@example.com
- **Find via name** : find KEYWORD [MORE_KEYWORDS]
e.g. find James Jake
- **Find via tag** : findTag KEYWORD [MORE_KEYWORDS]
e.g. find friend nus
- **List** : list
- **Help** : help
- **Select** : select INDEX
e.g. select 2
- **Locate** : locate INDEX
e.g. select 3
- **History** : history
- **Undo** : undo
- **Redo** : redo