


NUSMeet User Guide

NUSMeet is a **Command Line Interface (CLI)** application designed for **NUS students** to **efficiently manage and organize** their contacts. It focuses on helping users **find coursemates** for study discussions, with **tagging** and **filtering** features tailored to student life.

NUSMeet User Guide

 [Quick start](#)

 [Command Overview](#)

 [Features](#)

1. [Viewing help : help](#)
2. [Adding a friend: add](#)
3. [Listing all friends : list](#)
4. [Editing a friend : edit](#)
5. [Finding friends: find](#)
6. [Deleting a friend : delete](#)
7. [Clearing all entries : clear](#)
8. [Exiting the program : exit](#)
9. [Saving your data](#)
10. [Editing the data file](#)
11. [Automatic Courses Archival](#)

 [FAQ](#)

 [Known issues](#)

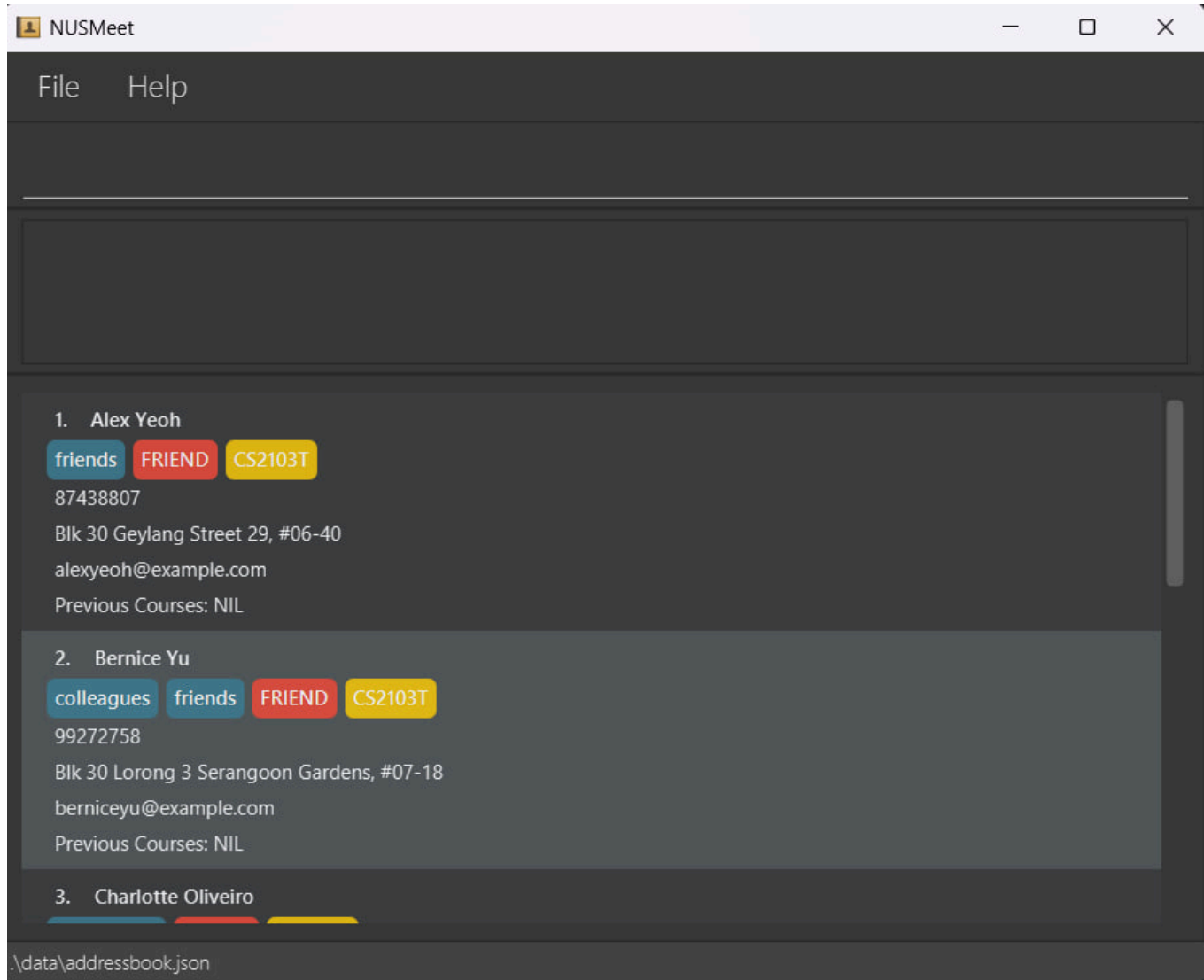
 [Glossary](#)

Quick start

Follow these steps to install and run NUSMeet on your computer!

1. Before running NUSMeet, ensure that you have **Java 17 or above** installed in your Computer.
 - **Windows users:** Install the precise JDK version prescribed [here](#).
 - **Mac users:** Install the precise JDK version prescribed [here](#).
 - **Linux users** Install the precise JDK version prescribed [here](#).
2. Download the latest `.jar` file [here](#).
3. Move the downloaded file to a folder of your choice to be used as the *home folder* for your NUSMeet application.
4. Open a command terminal and navigate to the folder where you placed the `.jar` file using `cd /path/to/your/folder` . Type the `java -jar nusmeet.jar` command to run the application.

A **Graphical User Interface (GUI)** similar to the one shown below 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. Refer to the **Features** below for details of each command.

Command Overview

Now that you have set up NUSMeet, let's explore the key commands you can use!

Action	Format, Examples
Add	<code>add n/NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS [c/COURSE]... [t/TAG]... [f/FRIENDSHIP_LEVEL]</code> e.g., <code>add n/James Ho p/22224444 e/jamesho@example.com a/123, Clementi Rd, 1234665 t/friend t/colleague</code>
Clear	<code>clear</code>
Delete	<code>delete INDEX</code> e.g., <code>delete 3</code>
Edit	<code>edit INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [c/COURSE]... [f/FRIENDSHIP] [t/TAG]...</code> e.g., <code>edit 2 n/James Lee e/jameslee@example.com</code>
Find	<code>find KEYWORD [MORE_KEYWORDS]</code> e.g., <code>find James Jake</code>
List	<code>list</code>
Help	<code>help</code>



Notes about the command format:

1. Upper Case

- 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`.

2. Square Brackets `[]`

- 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`.

3. Triple Dots `...`

- 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.

4. Ordering

- 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.

5. Parameters

- Extraneous parameters for commands that do not take in parameters (such as `help`, `list`, `exit` and `clear`) will be ignored.
e.g. if the command specifies `help 123`, it will be interpreted as `help`.

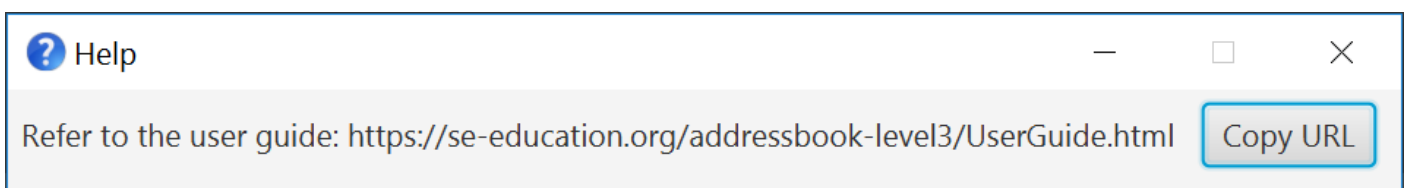
6. PDF

- If you are using a PDF version of this document, be careful when copying and pasting commands that span multiple lines as space characters surrounding line-breaks may be omitted when copied over to the application.

Features

1. Viewing help : `help`

Shows a message explaining how to access the help page.



Format: `help`

2. Adding a friend: `add`

Adds a friend to your contact book.

Format: `add n/NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS [c/COURSE]... [t/TAG]... [f/FRIENDSHIP_LEVEL]`



Note:

- The Friend Tag can only be one of the following `CLOSE_FRIEND`, `FRIEND`, `ACQUAINTANCES`
- A friend can have any number of tags & courses (including 0)

Examples:

- `add n/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01 c/CS2103T c/CS2101 t/friends t/owesMoney f/CLOSE_FRIEND`

- `add n/Betsy Crowe t/friend e/betsycrowe@example.com a/Newgate Prison p/1234567 f/FRIEND`

3. Listing all friends : `list`

Shows a list of all your friends in your contact book.

Format: `list`

4. Editing a friend : `edit`

Edits an existing friend's details in your contact book.

Format: `edit INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [c/COURSE]... [f/FRIENDSHIP] [t/TAG]...`

- Edits the friend at the specified `INDEX`. The index refers to the index number shown in the displayed friend list. 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 friend will be removed i.e adding of tags is not cumulative.
- You can remove all the friend'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 friend to be `91234567` and `johndoe@example.com` respectively.
- `edit 2 n/Betsy Crower t/` Edits the name of the 2nd friend to be `Betsy Crower` and clears all existing tags.

5. Finding friends: `find`

Finds friends who contain any of the given keywords and courses.

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.
- Only full words will be matched e.g. `Han` will not match `Hans`
- Persons matching at least one keyword will be returned (i.e. `OR` search). e.g. `Hans Bo` will return `Hans Gruber` , `Bo Yang`

Examples:

- `find John` returns `john` and `John Doe`
- `find alex david` returns `Alex Yeoh` , `David Li`

2 persons listed!

1. Alex Yeoh

friends

87438807

Blk 30 Geylang Street 29, #06-40

alexyeoh@example.com

2. David Li

family

91031282

Blk 436 Serangoon Gardens Street 26, #16-43

lidavid@example.com

6. Deleting a friend : `delete`

Deletes the specified friend from your contact book.

Format: `delete INDEX`

- Deletes the friend at the specified `INDEX` .
- The index refers to the index number shown in the displayed friend list.
- The index **must be a positive integer** 1, 2, 3, ...

Examples:

- `list` followed by `delete 2` deletes the 2nd friend in the address book.
- `find Betsy` followed by `delete 1` deletes the 1st friend in the results of the `find` command.

7. Clearing all entries : `clear`

Clears all entries from your contact book.

Format: `clear`

8. Exiting the program : `exit`

Exits the program.

Format: `exit`

9. Saving your data

NUSMeet automatically saves all changes to disk. There is no need to manually save your data.

10. Editing the data file

NUSMeet automatically stores data as a JSON file at: `[JAR file location]/data/addressbook.json` . You can manually edit this file.

⚠ Caution:

- If file format is **invalid**, NUSMeet will **discard** all data and start with an empty data file at the next run.
- Certain **incorrect** edits can cause **unexpected** behavior (e.g., invalid values).
- Recommended to **backup** the file before making any changes

11. Automatic Courses Archival

At the end of each semester, NUSMeet will automatically update all entries in your contact book. A friend's current courses will be cleared and moved to the "Previous Courses" section. This saves you time and effort, as you no longer need to manually update each contact.

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 NUSMeet home folder.

! Known issues

1. **When using multiple screens**, if you move the application to a secondary screen, and later switch to using only the primary screen, the GUI will open off-screen. The remedy is to delete the `preferences.json` file created by the application before running the application again.
 2. **If you minimize the Help Window** and then run the `help` command (or use the `Help` menu, or the keyboard shortcut `F1`) again, the original Help Window will remain minimized, and no new Help Window will appear. The remedy is to manually restore the minimized Help Window.
-

Glossary

Word	Meaning
CLI	A text-based interface where users interact with an application
Command Terminal	A console application where users enter commands.
Java	A programming language required to run an application.
JDK	A software package needed to run Java applications.
JSON	JavaScript Object Notation (JSON) is a lightweight data format used to store and exchange data
GUI	A visual interface that allows users to interact with an application.