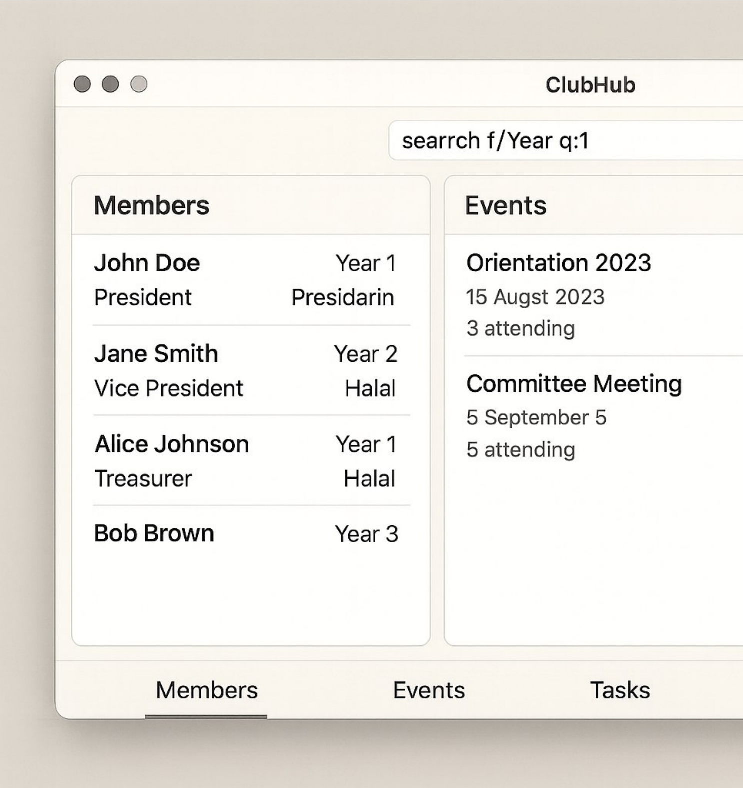
User Guide

ClubHub User Guide

AddressBook Level 3 (AB3) is a **desktop app for managing contacts, optimized for use via a Line Interface** (CLI) while still having the benefits of a Graphical User Interface (GUI). If you can type fast, AB3 can get your contact management tasks done faster than traditional GUI apps.

Quick start

1. Ensure you have Java `17` or above installed in your Computer. **Mac users:** Ensure you have the precise JDK version prescribed here.
2. Download the latest `.jar` file from here.
3. Copy the file to the folder you want to use as the *home folder* for your AddressBook.
4. Open a command terminal, `cd` into the folder you put the jar file in, and use the `java -jar addressbook.jar` command to run the application. A GUI similar to the below should appear in a few seconds. Note how the app contains some sample data.



searrch f/Year q:1

Members

John Doe Year 1
President Presidarin

Jane Smith Year 2
Vice President Halal

Alice Johnson Year 1
Treasurer Halal

Bob Brown Year 3

Events

Orientation 2023
15 Augst 2023
3 attending

Committee Meeting
5 September 5
5 attending

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 contacts.
   - `add n/John Doe y/3 s/A1234567X e/johnd@example.com p/98765432 d/Vegetarian r/President t/leadership` : Adds a club member.
   - `addevent ev/Orientation2025 dt/2025-08-15 desc/NUS Freshmen Orientation` : Adds an event.
   - `clear` : Deletes all contacts.
   - `exit` : Exits the app.

6. Refer to the Features below for details of each command.

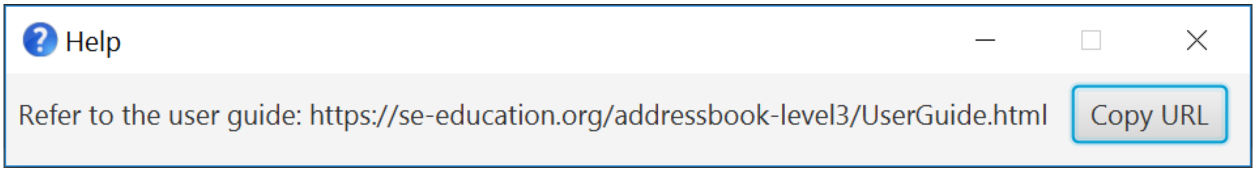
Features

Notes about the 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.
- 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`.
- 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.

Viewing help : `help`

Shows a message explaining how to access the help page.



Refer to the user guide: https://se-education.org/addressbook-level3/UserGuide.html

Figure 1: help message

Format: `help`

Adding a person: `add`

Adds a person to the address book.

Format: `add n/NAME y/YEAR s/STUDENT_NUMBER e/EMAIL p/PHONE d/DIETARY_REQUIREMENTS r/ROLE [t/TAG]…`

Tip: A person can have any number of tags (including 0)

Examples: * `add n/John Doe y/3 s/A1234567X e/johnd@example.com p/98765432 d/Vegetarian r/President` * `add n/Betsy Crowe y/2 s/A7654321A e/betsycrowe@example.com p/1234567 d/Halal r/Member t/cricket`

Listing all persons : `list`

Shows a list of all persons in the address book.

Format: `list`

Editing a person : `edit`

Edits an existing person in the address book.

Format: `edit INDEX [n/NAME] [p/PHONE] [e/EMAIL] [y/YEAR] [s/STUDENT_NUMBER] [d/DIETARY] [r/ROLE] [t/TAG]…`

- Edits the person at the specified `INDEX`. The index refers to the index number shown in the displayed person 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 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.

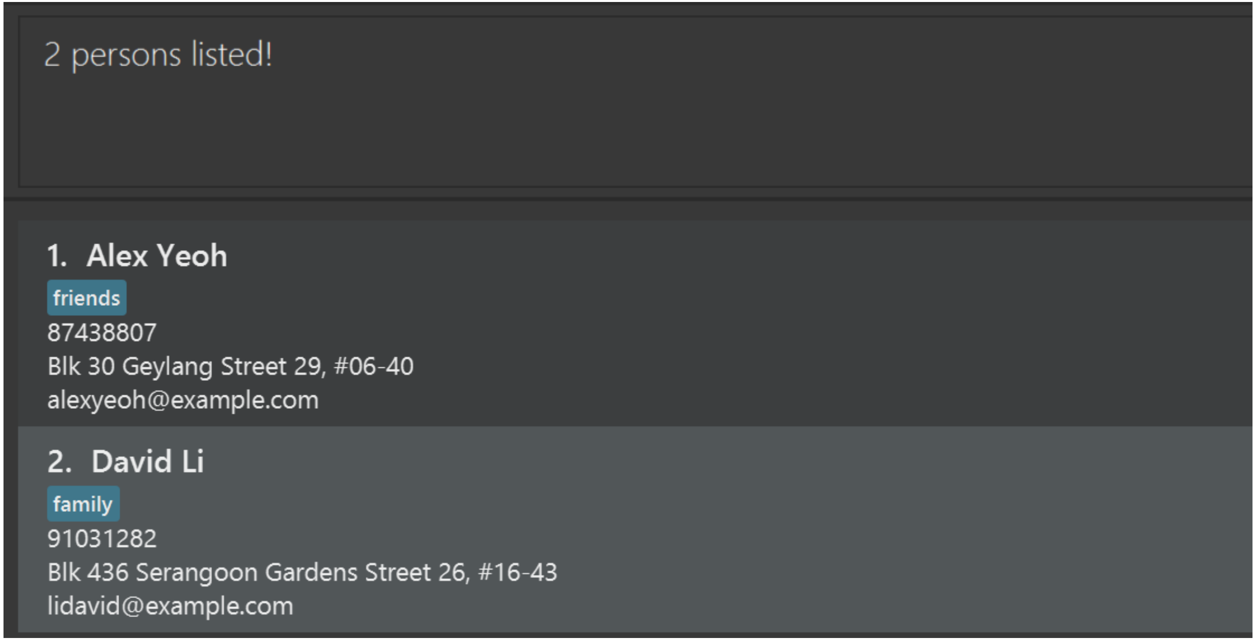
Locating persons by name: `find`

Finds persons whose names contain 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.
- 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

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 displayed person list.
- The index **must be a positive integer** 1, 2, 3, …

Examples: * `list` followed by `delete 2` deletes the 2nd person in the address book. * `find Betsy` followed by `delete 1` deletes the 1st person in the results of the `find` command.

Clearing all entries : `clear`

Clears all entries from the address book.

Format: `clear`

Exiting the program : `exit`

Exits the program.

Format: `exit`

Importing members : `import`

Imports member details from a CSV file into the address book.

Format: `import /from FILEPATH`

Examples: * `import /from members.csv`

Exporting members : `export`

Exports all members in the address book to a CSV file.

Format: `export /to FILEPATH`

Examples: * `export /to members.csv`

Events : `addevent`, `deleteevent`

Manages events in the address book.

Formats: * `addevent ev/EVENTID dt/DATE desc/DESCRIPTION` * `deleteevent e/EVENTID`

Examples: * `addevent e/Orientation2023 dt/2023-08-15 desc/NUS Freshmen Orientation` * `deleteevent e/Orientation2025`

Marking attendance : `markattendance`

Marks a member as attended for an event.

Format: `markattendance e/EVENTID m/MEMBERNAME`

- Marks the specified member as attended for the specified event.
- If the member is already marked as attended, the command will ignore the duplicate and continue.
- Both the event and member must exist in the system.

Examples: * `markattendance e/Orientation2023 m/John Doe` - Marks John Doe as attended for Orientation2023 * `markattendance e/Meeting2023 m/Jane Smith` - Marks Jane Smith as attended for Meeting2023

Adding attendance : `addattendance`

Adds the member to the attendance list.

Format: `addattendance e/EVENTID m/MEMBER[/MEMBER]...`

- Adds the member who is supposed to attend the event to the attendance list.
- If the member has already been added the attendance list, the command will ignore the duplicate and continue.
- Both the member and event must exist in the system.

Examples: * `addattendance e/Orientation2023 m/John Doe` - Adds John Doe to the attending list of Orientation2023 * `addattendance e/Meeting2023 m/Jane Smith` - Adds Jane Smith to the attending list of Meeting2023

Viewing attendance : `viewattendance`

Views the list of members who attended a specific event.

Format: `viewattendance e/EVENTID`

- Shows all members marked as attended for the specified event.
- If no attendance has been recorded for the event, shows "No attendance recorded yet."

Examples: * `viewattendance e/Orientation2023` - Shows all members who attended Orientation2023 * `viewattendance e/Meeting2023` - Shows all members who attended Meeting2023

Showing attendance summary : `showattendance`

Displays a summary of attendance for an event, including counts of attended and absent members.

Format: `showattendance e/EVENTID`

- Shows a summary with separate lists of members who attended and were absent.
- Displays the count for each category.
- The event must exist in the system.

Examples: * `showattendance e/Orientation2023` - Shows attendance summary for Orientation2023 * `showattendance e/Meeting2023` - Shows attendance summary for Meeting2023

Tasks : `addtask`, `deletetask`, `marktask`, `unmarktask`

Manages simple tasks for the club.

Formats: * `addtask TITLE [dl/DEADLINE]` * `deletetask INDEX` * `marktask INDEX` * `unmarktask INDEX`

Examples: * `addtask Submit budget dl/2025-11-01 23:59` * `marktask 1`

Undo/Redo : `undo`, `redo`

Reverts or reapplies the most recent changes.

Formats: * `undo` * `redo`

Budget tracker : `budget`, `setexpense`

Tracks a global budget in SGD with an inclusive date range, and per-event expenses.

Formats: * `budget set a/AMOUNT from/YYYY-MM-DD to/YYYY-MM-DD` * `budget reset` * `budget report` * `setexpense INDEX a/AMOUNT`

Notes: * Currency is SGD with two decimals; negative amounts are not allowed. * The report lists only events whose date is within the budget duration (inclusive). * The report format is:

```
Budget Report

Total budget: 123.45$
Duration: 2025-11-01 to 2025-11-30

EventA: 10.00$
EventB: 25.50$
EventC: 0.00$

Budget remaining: 87.95$
```

Saving the data

AddressBook data are saved in the hard disk automatically after any command that changes the data. There is no need to save manually.

Editing the data file

AddressBook data are saved automatically as a JSON file `[JAR file location]/data/addressbook.json`. Advanced users are welcome to update data directly by editing that data file.

Caution: If your changes to the data file makes its format invalid, AddressBook will discard all data and start with an empty data file at the next run. Hence, it is recommended to take a backup of the file before editing it. Furthermore, certain edits can cause the AddressBook to behave in unexpected ways (e.g., if a value entered is outside the acceptable range). Therefore, edit the data file only if you are confident that you can update it correctly.

Archiving data files `[coming in v2.0]`

Details coming soon …

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 AddressBook home folder.

Command summary

Action	Format, Examples
Add	`add n/NAME y/YEAR s/STUDENT_NUMBER e/EMAIL p/PHONE d/DIETARY r/ROLE [t/TAG]…` e.g., `add n/John Doe y/3 s/A1234567X e/johnd@example.com p/98765432 d/Vegetarian r/President`
Clear	`clear`
Delete	`delete INDEX` e.g., `delete 3`
Edit	`edit INDEX [n/NAME] [p/PHONE_NUMBER] [e/EMAIL] [y/YEAR] [s/STUDENT_NUMBER] [d/DIETARY] [r/ROLE] [t/TAG]…` e.g.,`edit 2 n/Betsy Crower t/`
Find	`find KEYWORD [MORE_KEYWORDS]` e.g., `find James Jake`
List	`list`
Import	`import /from FILEPATH` e.g., `import /from members.csv`
Export	`export /to FILEPATH` e.g., `export /to members.csv`
Events	`addevent ev/EVENTID dt/DATE desc/DESC` `deleteevent e/EVENTID`
Attendance	`markattendance e/EVENTID m/MEMBERNAME` `addattendance e/EVENTID m/MEMBER[/MEMBER]...` `viewattendance e/EVENTID` `showattendance e/EVENTID`
Tasks	`addtask TITLE [dl/DEADLINE]`, `deletetask INDEX`, `marktask INDEX`, `unmarktask INDEX`
Undo/Redo	`undo`, `redo`
Budget	`budget set a/AMOUNT from/START to/END`, `budget reset`, `budget report`, `setexpense INDEX a/AMOUNT`
Help	`help`