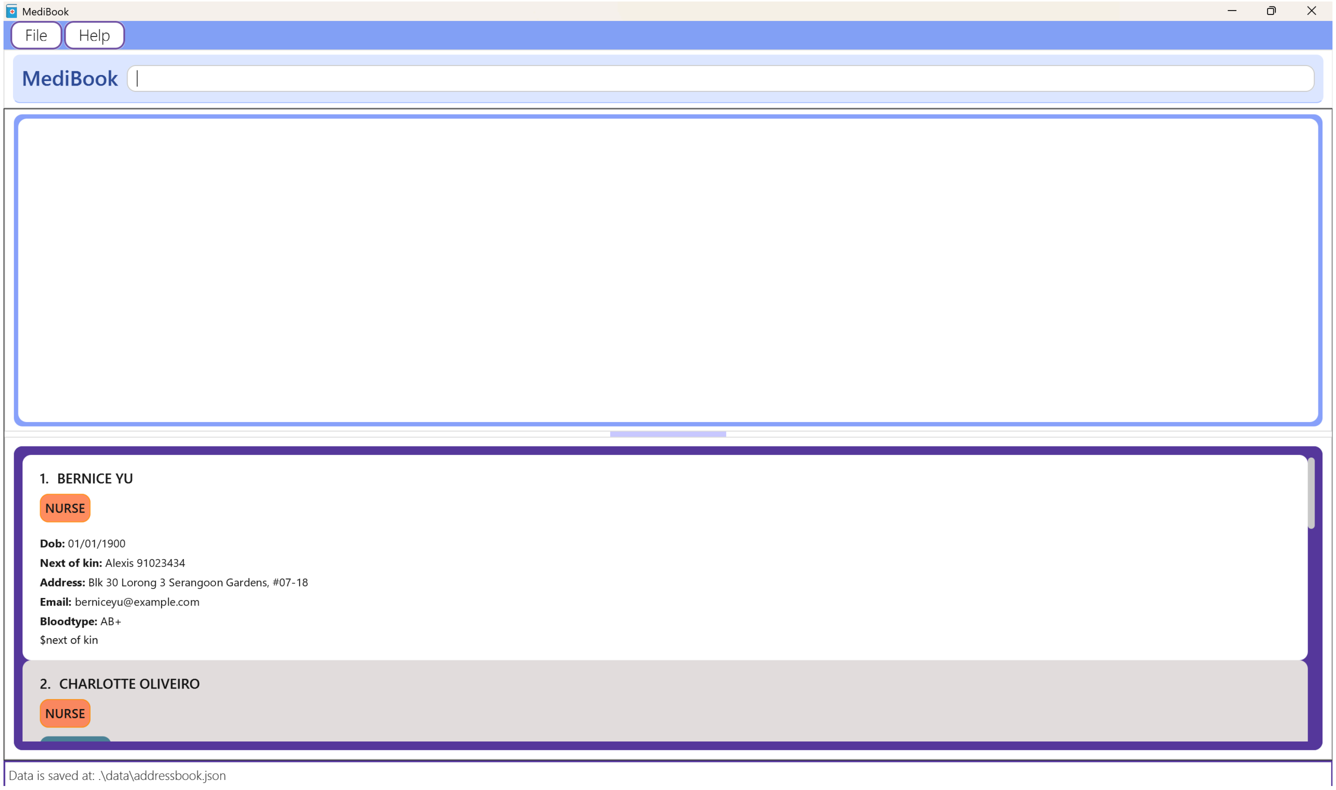
MediBook is a desktop app for managing patient and nurse records, optimized for use via a Command Line Interface (CLI), while still offering the benefits of a Graphical User Interface (GUI). Designed for speed and efficiency, MediBook empowers private nurse centres to assign staff, retrieve patient information, and manage appointments faster than traditional pen-and-paper or GUI-based systems.

Table of Contents

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 MediBook.
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.



1. BERNICE YU

NURSE

Dob: 01/01/1900
Next of kin: Alexis 91023434
Address: Blk 30 Lorong 3 Serangoon Gardens, #07-18
Email: berniceyu@example.com
Bloodtype: AB+
$next of kin

2. CHARLOTTE OLIVEIRO

NURSE

Data is saved at: .\data\addressbook.json

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 dob/01/01/2025 p/98765432 e/johnd@example.com a/John street, block 123, #01-01 b/AB+ ap/Patient` : Adds a contact named `John Doe` to the Address Book.
   - `edit 2 b/AB-` : Edits the blood type of the 2nd contact shown in the current list.
   - `delete 3` : Deletes the 3rd contact shown in the current list.
   - `assign 1 2` : Assigns the patient at index 1 to the nurse at index 2 shown in the current list.
   - `assign delete john doe 3` : Removes assigned nurse John Doe from the patient at index 3.
   - `find j` : Finds all contacts whose names start contains the letter 'j'.
   - `find patient of nurse 1` : Finds the patients assigned to the nurse at index 1 of the currently shown list.
   - `find nurse of patient 2` : Finds the nurses assigned to the patient at index 2 on the currently shown list.
   - `schedule add for patient 2 25/05/2025 1100` : Schedules a check-up for the patient shown at the 2nd index of the currently shown list at the given date and time.
   - `schedule delete for patient 2 25/05/2025 1100` : Deletes a check-up for the patient shown at the 2nd index of the list at the given date and time.
   - `view 4` : Displays the medical history of the person at the index 4 in the current list. If the person is a patient with medical history, the medical history will also be shown.
   - `clear` : Deletes all contacts.
   - `exit` : Exits the app.
6. Refer to the Features below for details of each command.

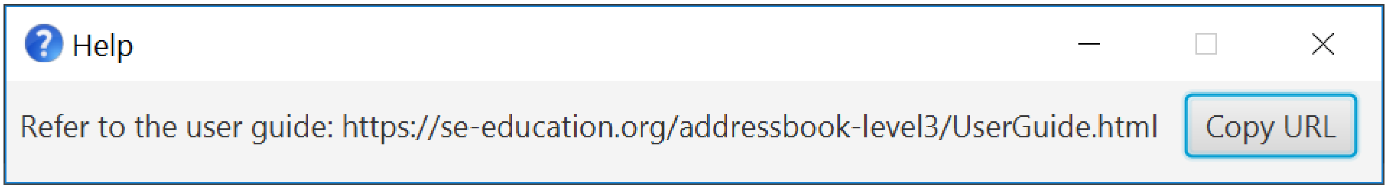
Features

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.
- Tag and Medical History allows only alphanumeric characters and spaces.
- 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` , `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

Format: `help`

Adding a person: `add`

Adds a person to the address book.

Format: `add n/NAME dob/DOB p/PHONE_NUMBER a/ADDRESS b/BLOOD_TYPE ap/APPOINTMENT [e/EMAIL] [nok/NEXT_OF_KIN_NAME_PHONE] [t/TAG]… [mh/MEDICAL_HISTORY]…`

Note: A person can have any number of tags and medical history (including 0)

- Medical history is only for patients, an error will occur if you try to add a nurse with medical history.

Examples:

- `add n/John Doe dob/01/01/2001 p/98765432 e/johnd@example.com a/John street, block 123, #01-01 b/AB+ ap/Nurse t/friend`
- `add n/Betsy Crowe dob/01/01/2001 t/friend e/betsycrowe@example.com a/Newgate Prison p/1234567 t/No Family mh/Insane, but not dangerous b/B+ ap/Patient`

Listing persons: `list`

Displays a list of persons in the address book. You can choose to list all persons, only patients, only nurses or all patients with checkups.

Formats:

`list` — Lists all persons.

`list patient` — Lists only patients.

`list nurse` — Lists only nurses.

`list checkup` - Lists all patients with checkups.

Examples:

`list` : Shows every entry in the address book.

`list patient` : Shows only persons with the appointment role Patient.

`list nurse` : Shows only persons with the appointment role Nurse.

`list checkup` : Shows only the patients with scheduled checkups, sorted from earliest to latest checkup.

Editing a person: `edit`

Edits an existing person in the address book.

Format: `edit INDEX [n/NAME] [dob/DOB] [p/PHONE] [a/ADDRESS] [b/BLOOD_TYPE] [ap/APPOINTMENT] [e/EMAIL] [nok/NEXT_OF_KIN_NAME_PHONE] [t/TAG]… [mh/MEDICAL_HISTORY]…`

- 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 or medical history, the existing tags or medical history of the person will be removed i.e adding of tags or medical history is not cumulative.
- You can remove all the person’s tags by typing `t/` without specifying any tags after it.
- You can remove all the person’s medical history by typing `mh/` without specifying any medical history after it.
- Editing of medical history is only for patients, an error will occur if you try to edit a nurse’s medical history.
- If a change of appointment is required, patient to nurse, do ensure medical history is cleared before changing appointment. Else if nurse to patient, edit the appointment as required.

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/ mh/` Edits the name of the 2nd person to be `Betsy Crower` and clears all existing tags and medical history.
- `edit 3 t/one t/two mh/one mh/two` Edits the tags and medical history of the person at the third index.

Locating persons by name: `find`

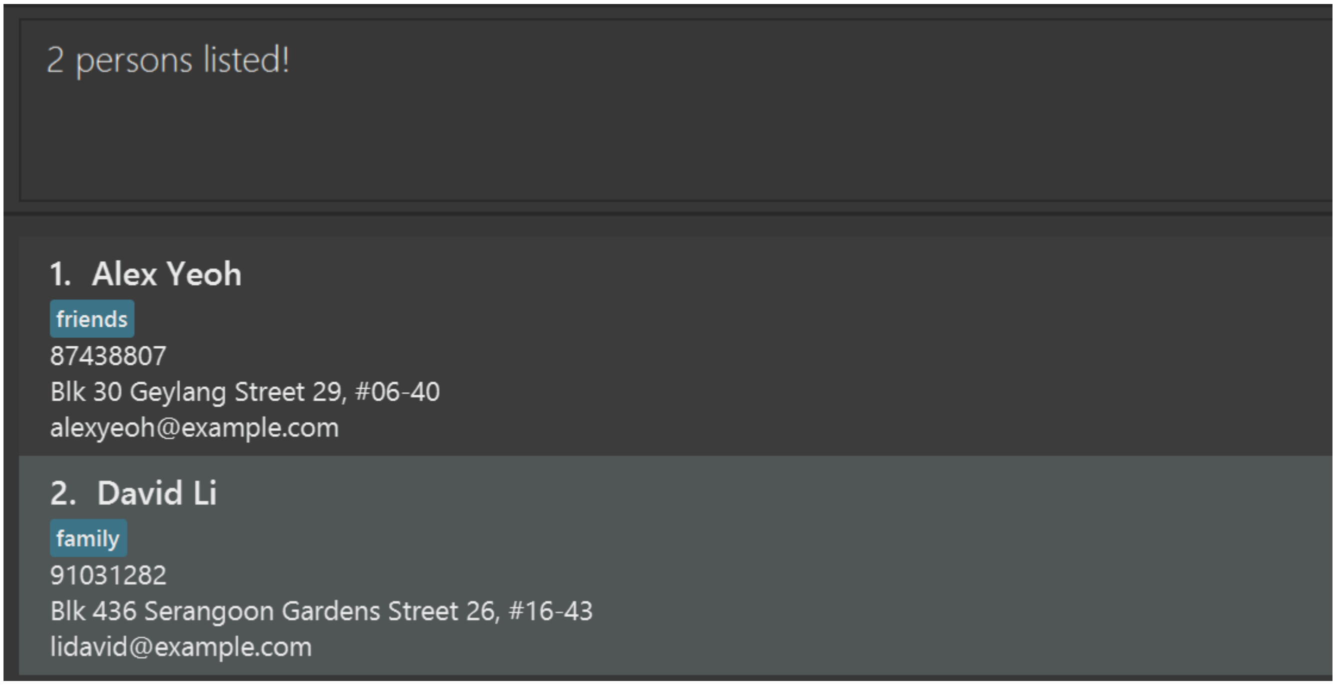
Finds persons whose names contain any of the given keywords or prefixes.

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.
- Any name containing the prefix will be matched e.g. `Han` and `ns` will 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 al` returns `Alex Yeoh` and `Sally`
- `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

Finding patient `find patient of nurse`

Finds patients assigned under a specified nurse.

Format: `find patient of nurse INDEX`

- Finds patients who have the nurse at `INDEX` assigned to them.
- The index refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, …

Examples:

- `find patient of nurse 1`

Finding nurse `find nurse of patient`

Finds nurse(s) assigned to a specified patient.

Format: `find nurse of patient INDEX`

- Finds the nurse(s) assigned to the patient at `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:

- `find nurse of patient 2`

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`

Assign a nurse to a patient : `assign`

Assigns a specified nurse to a specified patient.

Format: `assign PATIENT_INDEX NURSE_INDEX`

- Assigns the nurse at `NURSE_INDEX` to the patient at `PATIENT_INDEX` .
- `NURSE_INDEX` and `PATIENT_INDEX` both refer to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, …

Examples:

- `assign 2 1` assigns the nurse at index 1 to the patient at index 2.

Delete nurse assignment from a patient : `assign delete`

Removes a specified assigned nurse from a specified patient.

Format: `assign delete NURSE_NAME PATIENT_INDEX`

- Removes the assigned nurse with name `NURSE_NAME` from the patient at `PATIENT_INDEX` .
- `NURSE_NAME` needs to match the full name shown on the patient's assigned nurse tag, but is case-insensitive.
- `PATIENT_INDEX` refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, …

Examples:

- `assign delete john doe 2` removes the assignment of Nurse JOHN DOE from the patient at index 2.

Schedule checkup appointments : `schedule add for patient` / `schedule delete for patient`

Schedules a checkup appointment for the patient, or deletes an existing checkup appointment.

Note

- Checkups cannot be created before the current time.
- Checkups can only be created during working hours (9am to 5pm).
- Date Time follows the format of DD/MM/YYYY HHMM.
- A warning will be given if the target patient does not have a assigned nurse.

Actions:

- Add a checkup.
- Delete a checkup (that exists).

Format:

- `schedule add for patient INDEX DATE TIME`
- `schedule delete for patient INDEX DATE TIME`
- `INDEX` refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, …

Examples:

- `schedule add for patient 1 12/12/2025 1200` schedules a checkup for patient at index 1 on 12/12/2025 at 12:00pm
- `schedule delete for patient 1 12/12/2025 1200` deletes a checkup for patient at index 1 on 12/12/2025 at 12:00pm

Viewing a nurse or patient : `view`

Displays details of specified person. If specified person is a patient with medical history, the medical history will be shown in the result display box.

Format: `view INDEX`

- Displays the details of the person at `INDEX` .
- If the person is a patient, then the patient's medical history is shown (if any).
- The index refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, …

Saving the data

MediBook 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

MediBook 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, MediBook 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 MediBook to behave in unexpected ways (e.g., if a value entered is outside of the acceptable range). Therefore, edit the data file only if you are confident that you can update it correctly.

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.

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.

Command summary

Action	Format, Examples
Add	`add n/NAME dob/DOB p/PHONE_NUMBER a/ADDRESS b/BLOODT_TYPE ap/APPOINTMENT [e/EMAIL] [nok/NEXT_OF_KIN_NAME_PHONE] [t/TAG]… [mh/MEDICAL_HISTORY]…` e.g., `add n/John Doe dob/01/01/2001 p/98765432 e/johnd@example.com a/John street, block 123, #01-01 b/AB+ ap/Nurse`
Clear	`clear`
Delete	`delete INDEX` e.g., `delete 3`
Edit	`edit INDEX [n/NAME] [dob/DOB] [p/PHONE] [a/ADDRESS] [b/BLOOD_TYPE] [ap/APPOINTMENT] [e/EMAIL] [nok/NEXT_OF_KIN_NAME_PHONE] [t/TAG]… [mh/MEDICAL_HISTORY]…` e.g., `edit 2 n/James Lee e/jameslee@example.com`
Find	`find KEYWORD [MORE_KEYWORDS]` e.g., `find James Jake`
Find Patient	`find patient of nurse INDEX` e.g., `find patient of nurse 1`
Find Nurse	`find nurse of patient INDEX` e.g., `find nurse of patient 2`
List	`list` `list nurse` `list patient` `list checkup`
Help	`help`
Assign	`assign PATIENT_INDEX NURSE_INDEX` e.g., `assign 2 1`
Assign Delete	`assign delete NURSE_NAME PATIENT_INDEX` e.g., `assign delete john doe 2`

Schedule	`schedule add for patient PATIENT_INDEX DATE_TIME`
	e.g., `schedule add for patient 2 01/01/2025 1100`
	`schedule delete for patient PATIENT_INDEX DATE_TIME`
	e.g., `schedule delete for patient 2 01/01/2025 1100`

View	`view INDEX` e.g., `view 2`

Colour Legend

Field	Colour
Patient Appointment	Light Blue
Nurse Appointment	Orange
Tag	Light Purple
Checkup	Light brown
Assigned Nurse	Green