

# MediBook

## User Guide

### About Medibook

MediBook is a **desktop app for private nurse center managers, to manage 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 nurses to patients
- Retrieve patient and nurse information
- Manage checkup sessions and medical history of patients

All within a user-friendly interface that is faster than traditional pen-and-paper or GUI-based systems. Get ready to revolutionize your private nurse centre management with **MediBook**, by getting started on a quick [Installation Guide](#) and exploring the [Features](#) of the app below!

### Using this User Guide

This user guide is designed to help you get started with **MediBook**, and to provide a simple and concise reference if you need any help while using the app. You may click on the words in blue to jump directly to the relevant section of the user guide.

To get an overview of the guide, you can refer to the [Table of contents](#) below.

# Table of Contents

1. About Medibook
  2. Using this User Guide
  3. Table of Contents
  4. Installation Guide
  5. Person-attribute
  6. Features
    - Viewing Help
    - Adding a person
    - Listing all persons
    - Editing a person
    - Locating persons by name
    - Finding patients
    - Finding nurse
    - Deleting a person
    - Clearing all entries
    - Exiting the program
    - Assigning a nurse to a patient
    - Delete nurse assignment from patient
    - Schedule checkups
    - Viewing a nurse or patient
    - Saving the data
    - Editing the data file
  7. FAQ
  8. Known Issues
  9. Command Summary
-

# Installation Guide

1. Ensure you have Java `17` or above installed in your Computer.
  - Steps to check your current Java version, click [here](#).
  - Steps to install Java in your computer, click [here](#).
  - **For Mac users:** Ensure you have the precise Java version prescribed [here](#).

## Why do I have to install Java?

- *Java is a versatile programming language that lets apps run on any device with a Java Virtual Machine (JVM). Once installed, you can run any Java app, including MediBook. You can find out more about Java [here](#).*

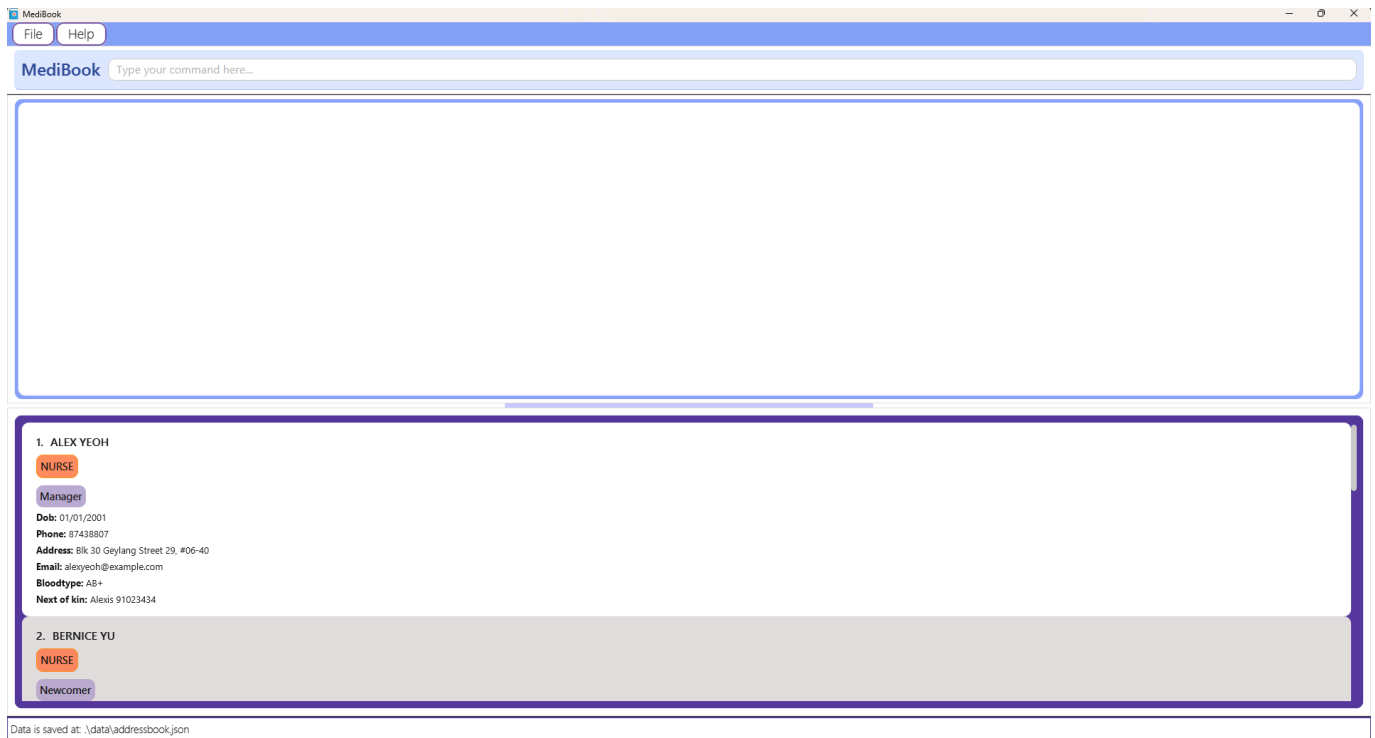
1. Download the latest `.jar` file from [here](#).
  - [What is a JAR file?](#)
2. Copy the file to an empty folder you want to use as the *home folder* for MediBook.
  - MediBook will create a data folder in this *home folder* to store your [data files](#).
3. Open a [command terminal](#), run the following commands:

```
cd home_folder
java -jar MediBook.jar
```

Example: if your `MediBook.jar` is in the folder `C:/Users/user/Documents`, Run the commands:

```
cd C:/Users/user/Documents
java -jar MediBook.jar
```

A GUI similar to the below should appear in a few seconds. Note how the app contains some sample data.



4. 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/01/01/2001 p/98765432 e/johnd@example.com a/311, Clementi Ave 2, #02-25 b/AB+ ap/Patient nok/Jane 91234567 t/Big Boy mh/High Blood Pressure** : Adds a contact named **John Doe** to MediBook.
- **edit 2 b/O+** : Edits the blood type of the 2nd contact shown in the current list to O+.
- **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.

5. Refer to the [Features](#) below for details of each command.

 [Back to Table of Contents](#)



---



## Overview of Person Attributes

This section provides an overview of all the attributes a person in MediBook can have, including whether they're required, what format they should follow, and any special notes.


Attribute	Prefix	Required?	Valid Format	Example
Name	n/	✓ Yes	<ul style="list-style-type: none"> <li>- Alphabetical</li> <li>- <b>Non consecutive</b> special characters (hyphens, slash, apostrophes)</li> <li>- Spaces</li> <li>- Cannot start or end with a special character</li> <li>- Case-insensitive</li> </ul>	E.g., Joh 'n-S/0-Doe
Date of Birth	dob/	✓ Yes	dd/mm/yyyy	E.g., 11/11/2003
Phone Number	p/	✓ Yes	<ul style="list-style-type: none"> <li>- Max allowable is 17 digits</li> <li>- Spaces are allowed, but each group requires min 3 digits</li> <li>- Only numericals and spaces are allowed, no other symbols are allowed</li> </ul>	E.g., 1234 567 890
Address	a/	✓ Yes	<ul style="list-style-type: none"> <li>- Free text</li> <li>- Case-sensitive</li> </ul>	E.g., 10 NUS Road, #03-21
Blood Type	b/	✓ Yes	<ul style="list-style-type: none"> <li>- Standard blood group types</li> <li>- Case-insensitive</li> </ul>	E.g., A+ , AB- , etc.

Attribute	Prefix	Required?	Valid Format	Example
Appointment Type	ap/	✔ Yes	<ul style="list-style-type: none"> <li>- Appointment type of personnel</li> <li>- Case-insensitive</li> </ul>	E.g., <code>Nurse</code> or <code>Patient</code>
Email	e/	✘ Optional	<ul style="list-style-type: none"> <li>- <code>local-part@domain.com</code></li> <li>- Local-part (min 2 characters) with special characters (plus, underscore, dot, hyphen)</li> <li>- Local-part must start with alphanumeric character</li> <li>- Domain (min 1 character)</li> <li>- <code>com</code> can be replaced with minimum 2 characters</li> </ul>	E.g., <code>john@example.com</code>
Next of Kin	nok/	✘ Optional	<ul style="list-style-type: none"> <li>- <code>Name Phone</code></li> <li>- A space is required between <code>Name</code> and <code>Phone</code></li> <li>- <code>Name</code> Same format constraints as the name attribute above</li> <li>- <code>Phone</code> Same format constraints as the phone number attribute above</li> <li>- Only one Next of Kin allowed</li> </ul>	E.g., <code>Jane 91234567</code>
Tags	t/	✘ Optional	<ul style="list-style-type: none"> <li>- Only alphanumeric and spaces allowed</li> <li>- Case-sensitive</li> </ul>	E.g., <code>2 Smart</code>

Attribute	Prefix	Required?	Valid Format	Example
			<ul style="list-style-type: none"> <li>- Multiple tags allowed</li> <li>- Colour based</li> </ul>	
Medical History	m/	 Optional*	<ul style="list-style-type: none"> <li>- Alphanumeric and spaces</li> <li>- Special characters (commas, hyphens, slash, rounded brackets, colon)</li> <li>- Only allowed for <b>patients</b></li> <li>- Multiple medical histories allowed</li> </ul>	E.g., (Very, sick): – urgent/need12
Checkups	(N/A)	 Optional	Managed via schedule command <ul style="list-style-type: none"> <li>- Each checkup is fixed 30 minutes</li> <li>- Checkups cannot be scheduled in the past.</li> </ul>	Not included in add command

 **Note:** Optional fields are marked with square brackets in command formats for readability. **Do not include square brackets in your actual command input.** 

Correct: e/john@example.com  Incorrect: [e/john@example.com]

 **Email Constraints:** The email format given [above](#) is a simplified version, for tidiness, you can find the full constraints, through a warning message, when you enter an invalid email in the app.

 [Back to Table of Contents](#)



# Features

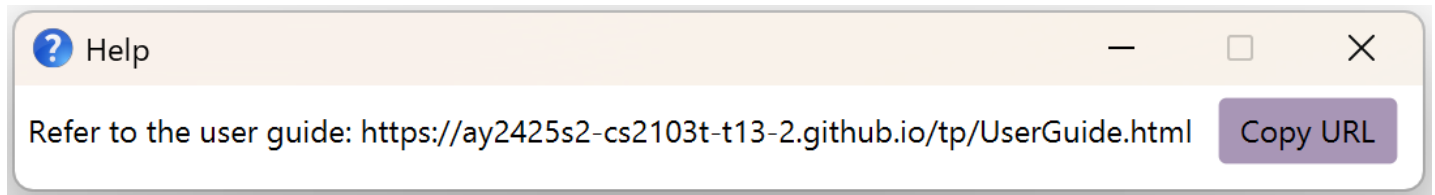
## Notes about the command format:

- 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 — do not include the brackets themselves when typing commands.  
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`, `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.

Quick Navigation: [Viewing Help](#) • [Adding a person](#) • [Listing all persons](#) • [Editing a person](#) • [Locating by name](#) • [Finding patients](#) • [Finding nurse](#) • [Deleting](#) • [Clearing all entries](#) • [Exiting program](#) • [Assigning nurse](#) • [Delete nurse assignment](#) • [Schedule checkups](#) • [Viewing person](#) • [Saving data](#) • [Editing data file](#) •

## Viewing help: `help`

Launches a popup window with a link for our User Guide. Clicking on `Copy URL` will copy the URL for you to paste in your browser.



Format:

 [Back to Features](#)

## Adding a person:

Adds a person to the address book.

### Details


- 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.

### Format

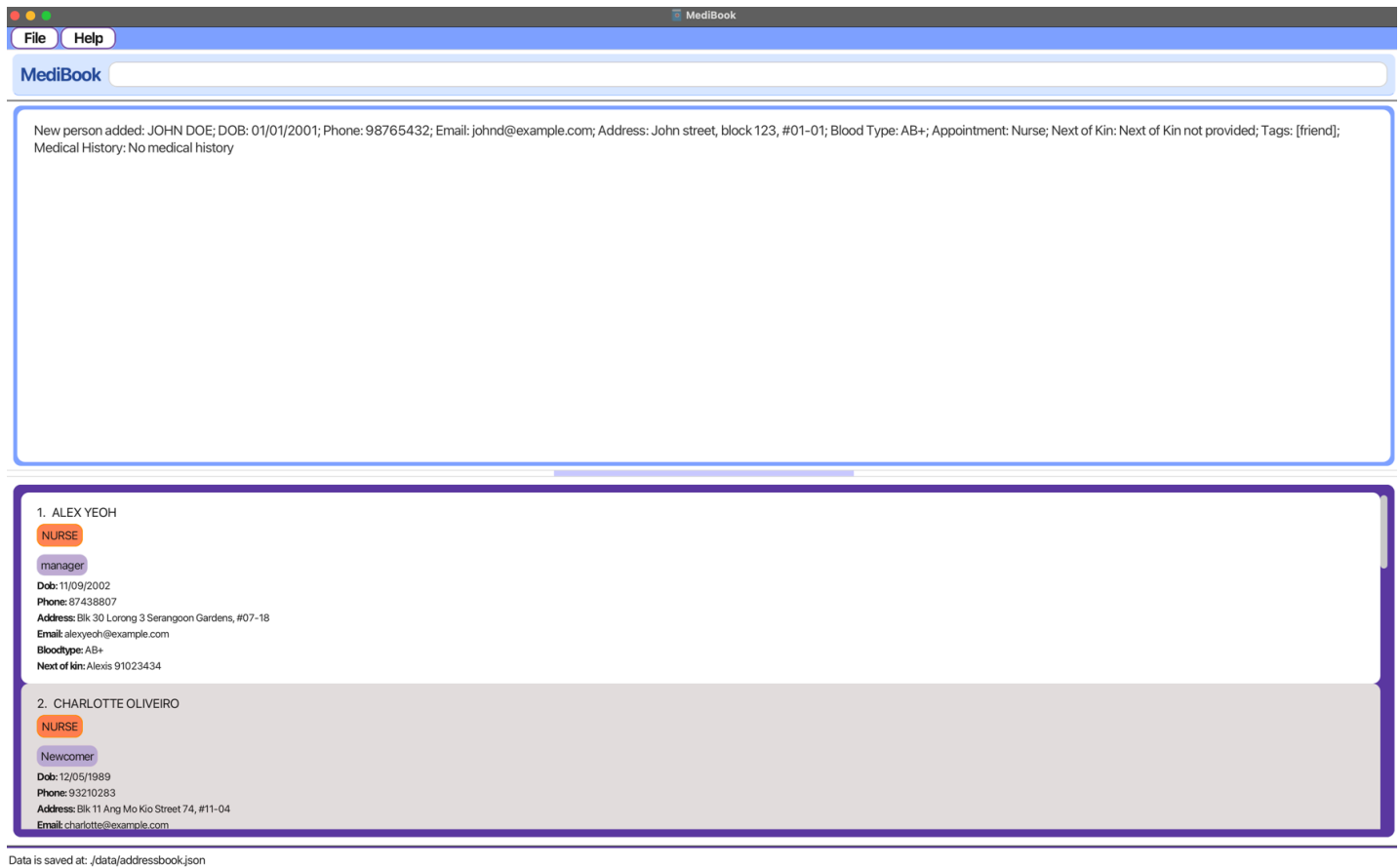
```
add n/NAME dob/DOB p/PHONE_NUMBER a/ADDRESS b/BLOOD_TYPE ap/APPOINTMENT  
[e/EMAIL] [nok/NEXT_OF_KIN_NAME_PHONE] [t/TAG]... [mh/MEDICAL_HISTORY]...
```

### 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`

 **Tip:** You can refer back to [here](#), for more details on the person attributes and their formats.

 **Constraints:** MediBook has a duplicate policy which you can view [here](#).



[← BACK](#) [Back to Features](#) [View Person Attributes](#)

## 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.

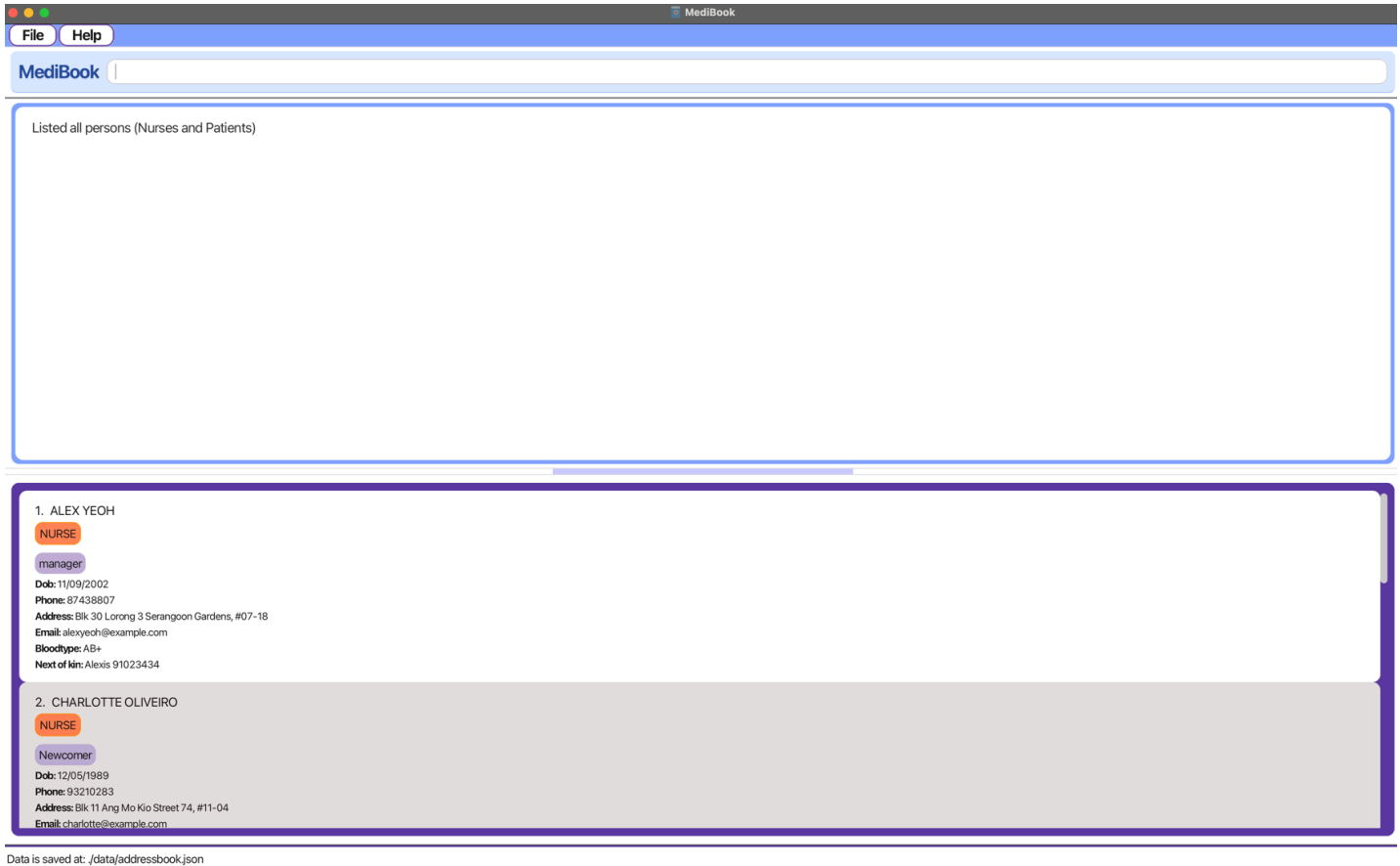
## Formats

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



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

[← BACK](#) [Back to Features](#)

## Editing a person: `edit`

Edits an existing person's information in MediBook.

### Details

- 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. 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 add by editing in a medical history to a nurse.
- If a change of appointment is required, patient to nurse, do ensure medical history is cleared before changing appointment.
- If a change of appointment is required, nurse to patient, do ensure the nurse has no patients assigned to them before changing appointment.
- If a change of nurse name is required, do ensure the nurse has no patients assigned to them before changing name.

## Format

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

## 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 to the respective values.

**⚠ Constraints:** MediBook has a duplicate policy which you can view [here](#).

The screenshot shows the MediBook application window. At the top, there's a menu bar with 'File' and 'Help'. Below it is a title bar 'MediBook'. The main content area displays the details of a person: 'Edited Person: ALEX YEOH; DOB: 01/01/2001; Phone: 8549 9584; Email: alexyeoh@example.com; Address: Blk 30 Geylang Street 29, #06-40; Blood Type: O+; Appointment: Nurse; Next of Kin: Alexis 91023434; Tags: [Manager]; Medical History: No medical history'. Below this, there's a list of users. The first user is '1. ALEX YEOH' with a 'NURSE' tag and a 'Manager' role. The second user is '2. BERNICE YU' with a 'NURSE' tag and a 'Newcomer' role. At the bottom, a status bar indicates 'Data is saved at: .\data\addressbook.json'.

 [Back to Features](#)  [View Person Attributes](#)

## Locating persons: `find`

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

### Details

- 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`

### Format

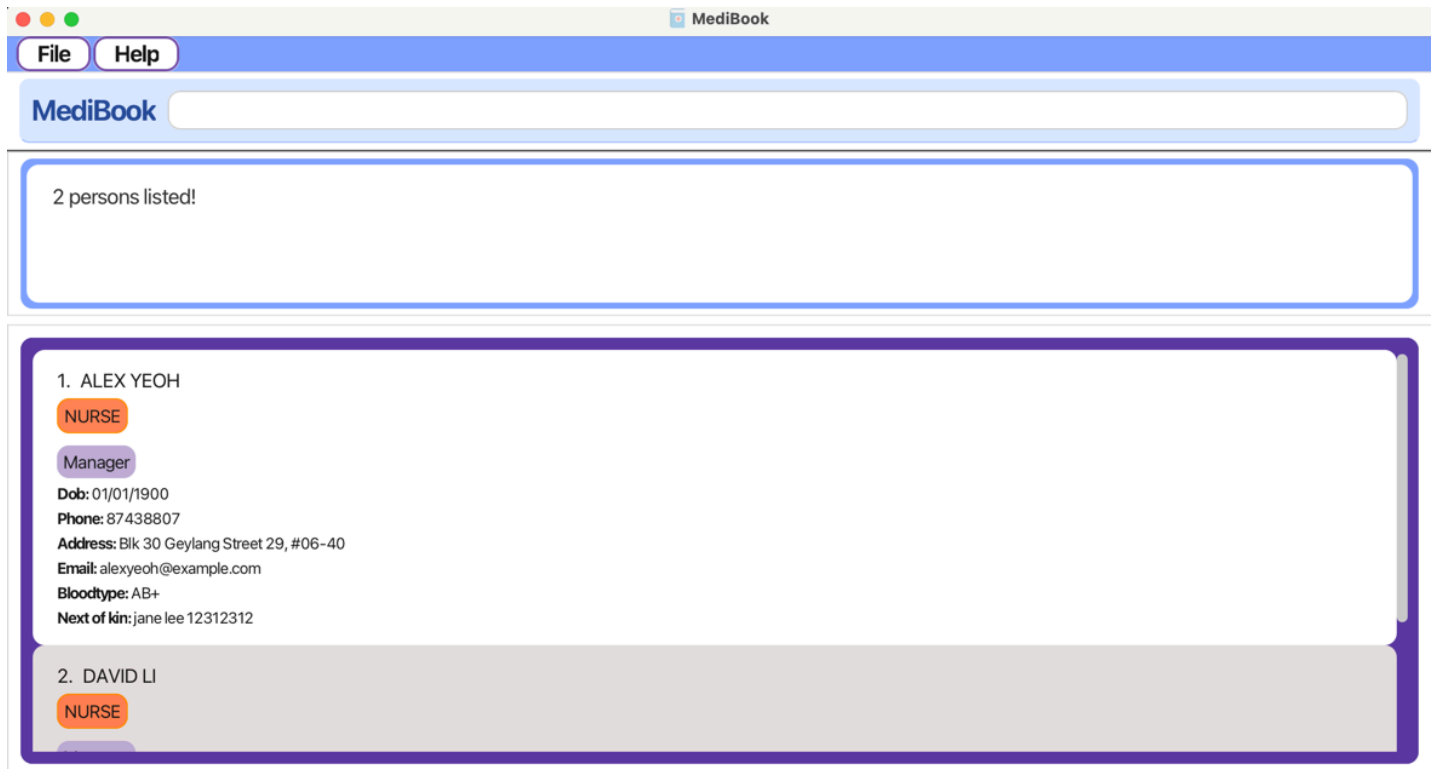
```
find KEYWORD [MORE_KEYWORDS]
```

### Parameters

- `KEYWORD` : The keyword to search for in a person's name. May be a full or partial name.
- `[MORE_KEYWORDS]` : Additional keywords (optional) to further find more than one person.

### Examples:

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



[← BACK](#) [Back to Features](#)

## Finding patient `find patient of nurse`

Finds patients assigned under a specified nurse.

### Details

- Finds patients assigned to the nurse at `INDEX`.
- If no patients are assigned to the nurse, the program will return that no patients were found.

### Parameters

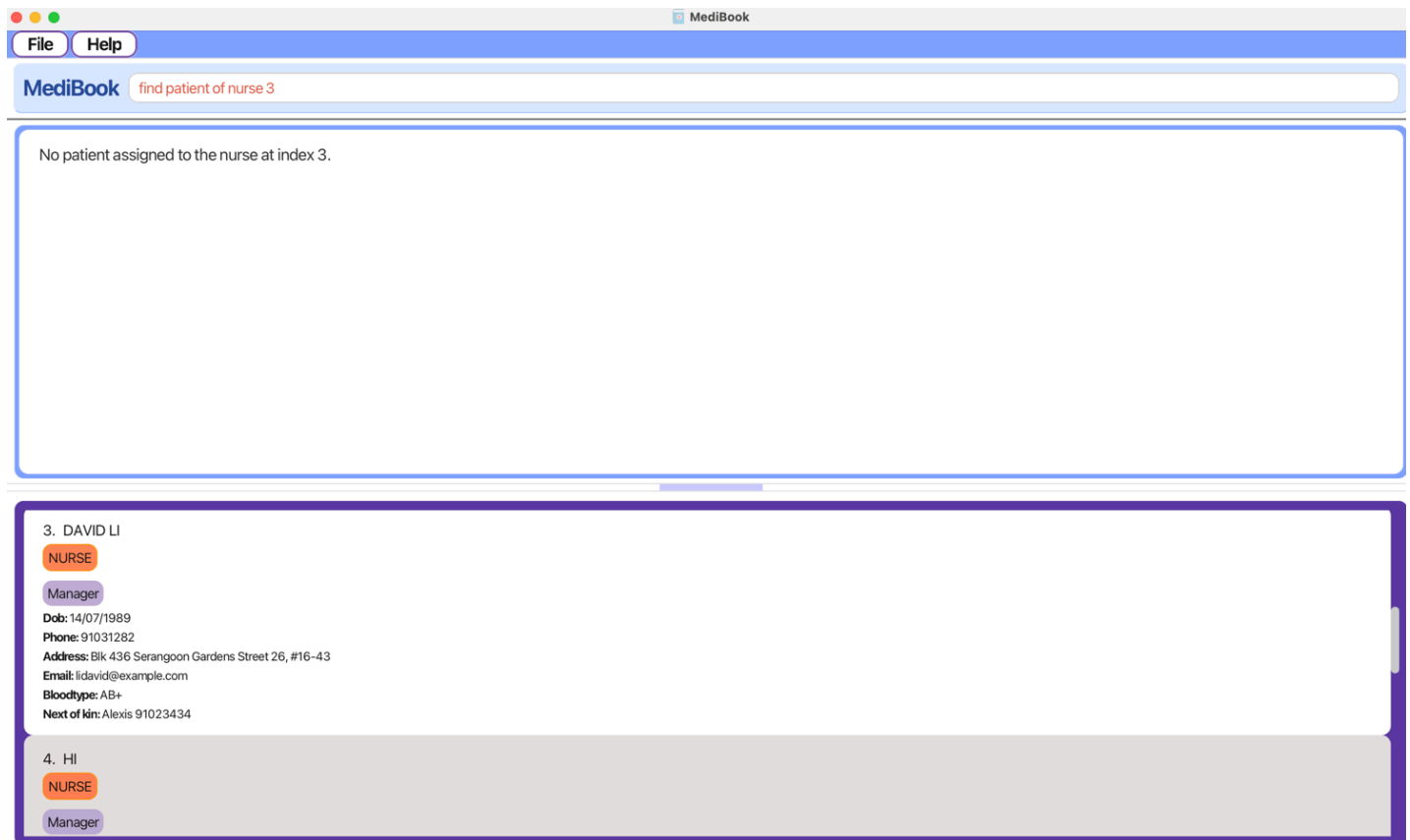
- The index refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3, ... and be within the total number of person in the list.

### Format

```
find patient of nurse INDEX
```

## Examples

- `find patient of nurse 1` returns e.g: Patient(s) assigned to nurse ALEX YEOH: ROY BALAKRISHNAN.
- `find patient of nurse 3` returns e.g: No patient assigned to the nurse at index 3.



[← BACK](#) [Back to Features](#)

## Finding nurse `find nurse of patient`

Finds nurse(s) assigned to a specified patient.

## Details

- Finds nurse at assigned to the patient at `INDEX`.
- If no patients are assigned to the nurse, the program will return that no patients were found.

## Parameters

- The index refers to the index number shown in the displayed person list.



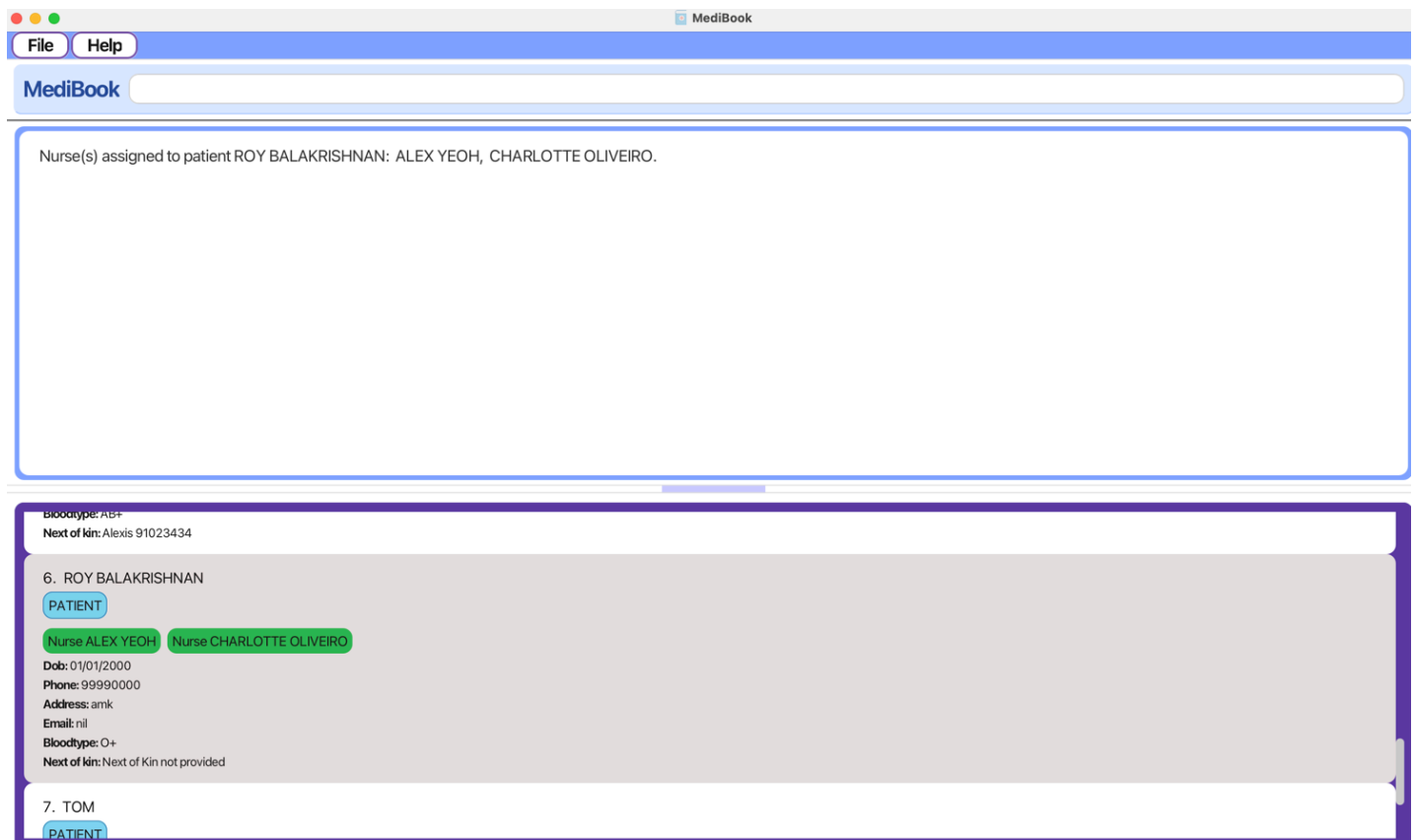
- The index **must be a positive integer** 1, 2, 3, ... and be within the total number of person in the list.

## Format

find nurse of patient INDEX

## Examples

- find nurse of patient 6 returns e.g: Nurse(s) assigned to patient ROY BALAKRISHNAN: ALEX YEOH, CHARLOTTE OLIVEIRO.
- find nurse of patient 7 returns e.g: No nurse assigned to the patient at index 7.



[← BACK](#) Back to Features

## Assign a nurse to a patient : assign

Assigns a specified nurse to a specified patient.

## Details

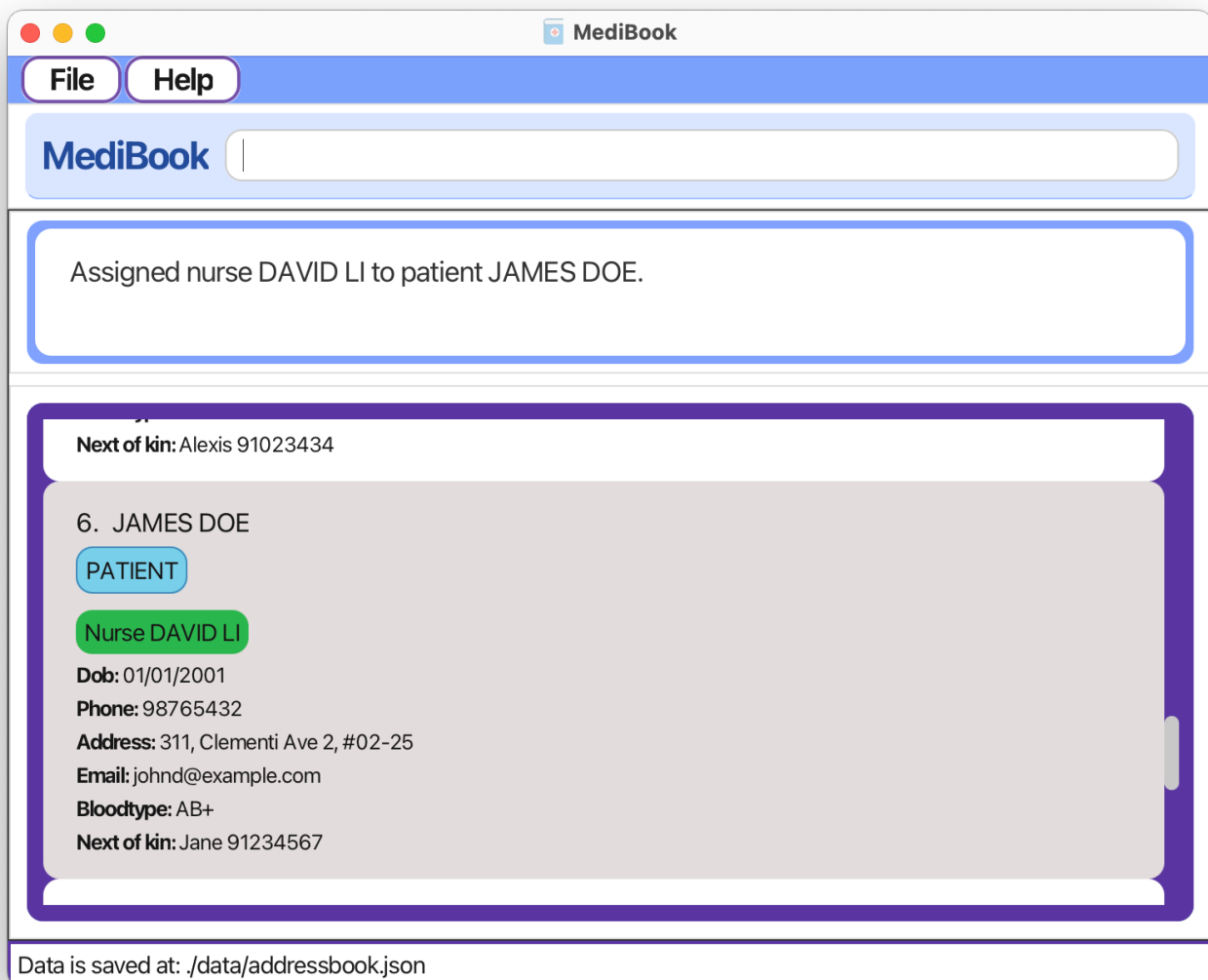
- Assigns the nurse at `NURSE_INDEX` to the patient at `PATIENT_INDEX` .
- `NURSE_INDEX` and `PATIENT_INDEX` both refer to the index number shown in the displayed person list.
- At most 2 nurses can be assigned to one patient.
- The index **must be a positive integer** 1, 2, 3, ...

## Format

```
assign PATIENT_INDEX NURSE_INDEX
```

## Examples

- `assign 6 4` assigns the nurse at index 4 to the patient at index 6.



## Delete nurse assignment from a patient : `assign delete`

Removes a specified assigned nurse from a specified patient.

### Details

- 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, ...

### Format

```
assign delete NURSE_NAME PATIENT_INDEX
```

### Examples

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

 [Back to Features](#)

## Schedule Checkups: `schedule add for patient` / `schedule delete for patient`

Schedules or deletes checkup sessions for patients.

### Details

- Checkups cannot be scheduled for:
  - Dates or times in the past (as of the current date and time).
  - Times outside of working hours (9:00 AM to 5:00 PM).
  - Times that are not in 15-minute increments (e.g., `00`, `15`, `30`, `45` ).
  - Times that are within 30 minutes of an existing checkup.
- Default timing checkups are for 30 minutes.
- The date and time must be in the format `DD/MM/YYYY HHMM`.
- A warning will be issued if the target patient doesn't have an assigned nurse.

## Actions

- **Add checkup:** Schedules a new checkup session.
- **Delete checkup:** Removes an existing checkup session.

## Format

- Add checkup:

```
schedule add for patient INDEX DATE TIME
```

- Delete checkup:

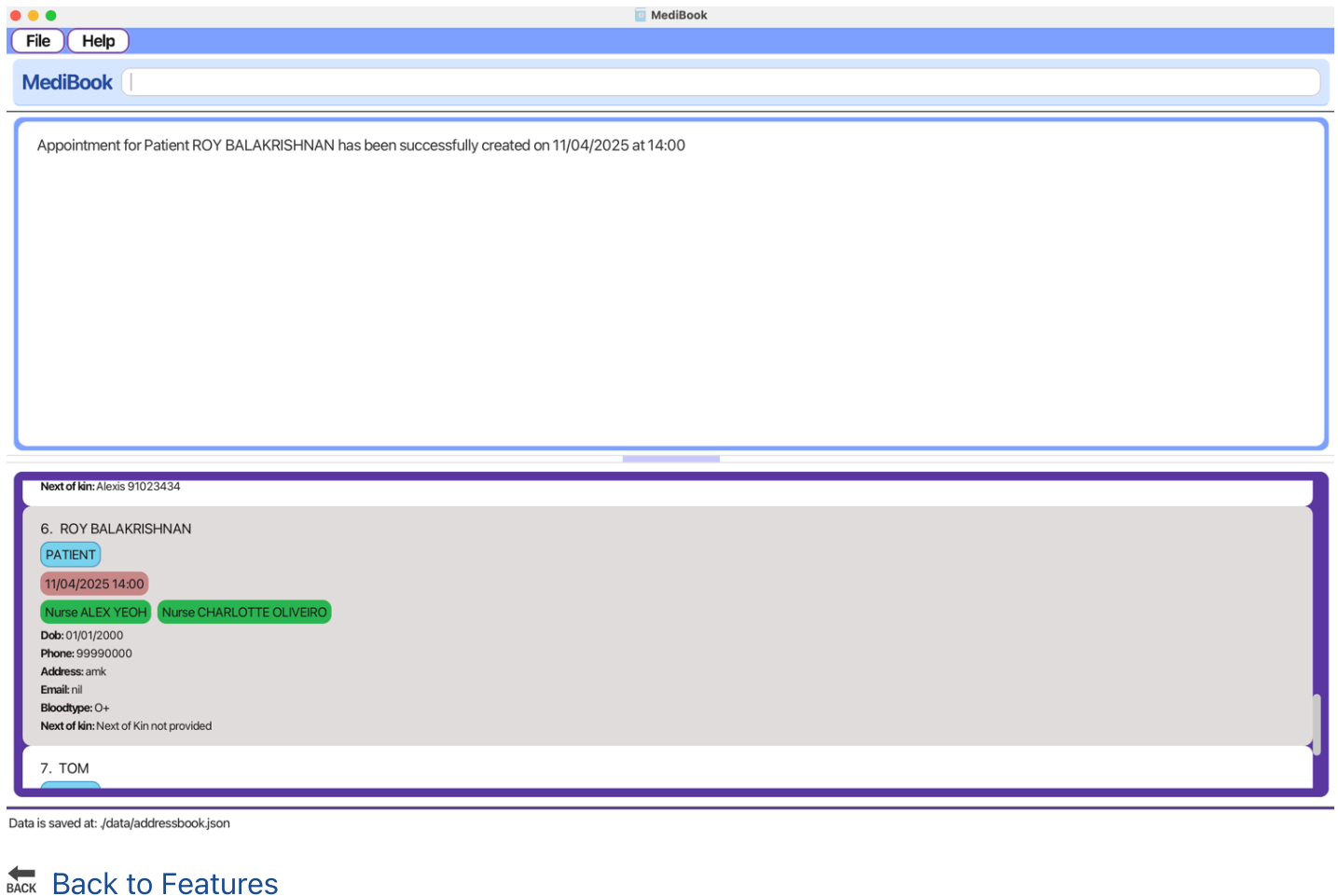
```
schedule delete for patient INDEX DATE TIME
```

## Parameters

- **INDEX** : Refers to the index number of the patient in the displayed list. The index must be a positive integer (1, 2, 3, ...).
- **DATE** : The date for the checkup, in **DD/MM/YYYY** format.
- **TIME** : The time for the checkup, in **HHMM** format (24-hour clock).

## Examples

- `schedule add for patient 6 11/04/2025 1400` : Schedules a checkup for the patient at index 6 on April 11, 2025, at 14:00 PM.
- `schedule delete for patient 6 11/04/2025 1400` : Deletes a checkup for the patient at index 6 on April 11, 2025, at 14:00 PM.



## 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.

### Details

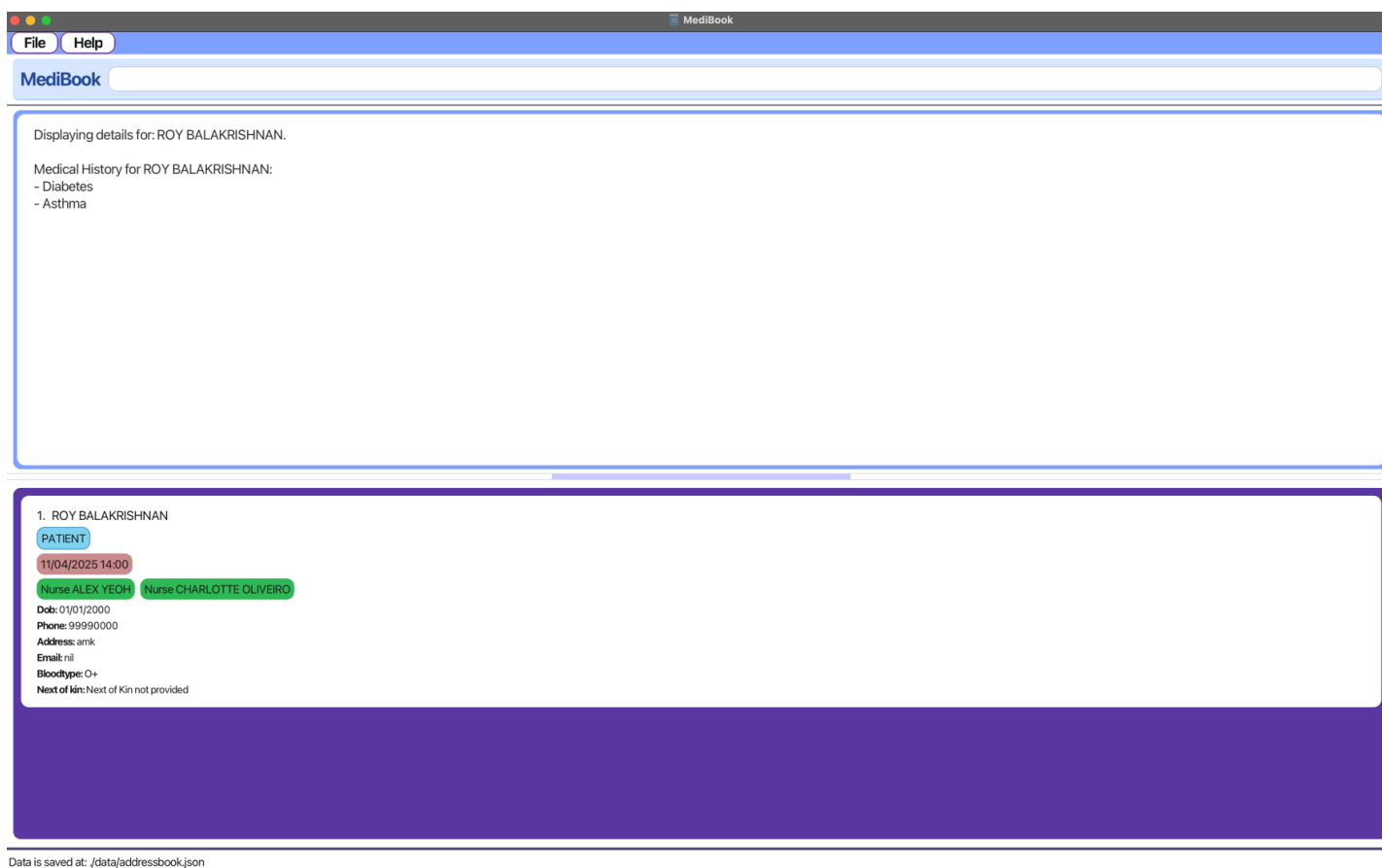
- Displays the details of the person at `INDEX`.
- If the person is a patient, then the patient's medical history is shown (if any).
- The index refers to the index number shown in the displayed person list.
- The index **must be a positive integer** 1, 2, 3

### Format

`view INDEX`

### Example

- `view 7` : Views the patient medical history



[← BACK](#) [Back to Features](#)

## Deleting a person : `delete`

Deletes the specified person from the address book.

## Details

- 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, ...

## Format

`delete INDEX`

## Example

- `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.

 [Back to Features](#)

## Clearing all entries : `clear`

Clears all entries from the address book.

### Format

`clear`

 [Back to Features](#)

## Exiting the program : `exit`

Exits the program.

### Format

`exit`

 [Back to Features](#)

## 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.

 [Back to Features](#)

## 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.

 [Back to Features](#)

 [Back to Table of Contents](#)

---

## FAQ

**Q:** How do I check my current Java version?

**A:** Open a command terminal and type `java --version`. If you see a version number, Java is installed. If not, you need to install Java.

**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.

 [Back to Table of Contents](#)

---

## 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.





[Back to Table of Contents](#)

---

## Command summary

Action	Format, Examples
<b>Add</b>	<pre>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]...</pre> <p>e.g., <code>add n/John Doe dob/01/01/2001 p/98765432</code>  <code>e/johnd@example.com a/John street, block 123, #01-01 b/AB+</code>  <code>ap/Nurse</code></p>
<b>Clear</b>	<code>clear</code>
<b>Delete</b>	<pre>delete INDEX</pre> <p>e.g., <code>delete 3</code></p>
<b>Edit</b>	<pre>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]...</pre> <p>e.g., <code>edit 2 n/James Lee e/jameslee@example.com</code></p>
<b>Find</b>	<pre>find KEYWORD [MORE_KEYWORDS]</pre> <p>e.g., <code>find James Jake</code></p>
<b>Find Patient</b>	<pre>find patient of nurse INDEX</pre> <p>e.g., <code>find patient of nurse 1</code></p>
<b>Find Nurse</b>	<pre>find nurse of patient INDEX</pre> <p>e.g., <code>find nurse of patient 2</code></p>
<b>List</b>	<code>list</code> <code>list nurse</code> <code>list patient</code> <code>list checkout</code>
<b>Help</b>	<code>help</code>
<b>Assign</b>	<pre>assign PATIENT_INDEX NURSE_INDEX</pre> <p>e.g., <code>assign 2 1</code></p>

Action	Format, Examples
<b>Assign</b> <b>Delete</b>	<code>assign delete NURSE_NAME PATIENT_INDEX</code> e.g., <code>assign delete john doe 2</code>
<b>Schedule</b>	<code>schedule add for patient PATIENT_INDEX DATE_TIME</code> e.g., <code>schedule add for patient 2 01/01/2025 1100</code> <code>schedule delete for patient PATIENT_INDEX DATE_TIME</code> e.g., <code>schedule delete for patient 2 01/01/2025 1100</code>
<b>View</b>	<code>view INDEX</code> e.g., <code>view 2</code>

[↑ TOP](#) [Back to Table of Contents](#)

## Duplicate Policy

MediBook allows all persons to have identical attributes through `edit` and `add` commands. User will be warned if it results in having two person sharing any of these attributes: `Name` , `Date of Birth` and `Phone Number` .

Example:

[↑ TOP](#) [Back to Table of Contents](#)

## Colour Legend

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

[↑ TOP](#) [Back to Table of Contents](#)