Problem addressed:

The address book is currently used as a tool to solely manage contact information. The military plans currently use it to contact service members when required. However there are plans to extend its functionality as a tool for military discipline.

The Singapore Armed Forces (SAF) has plans to trial a system and process to track and reward all soldiers within a particular battalion.

The plan is to evolve the address book desktop application into one that comprises the ability to tag each user with a series of Tags that the user can create.

There can be different tags created either for offences or rewards. The tags will be displayed on the user profile in the address book.

There should also be a memo that can be open, read and edited for each tag of a contact in the address book.

There are also plans to expand the functionality to allow it to track serviceman leave and performance appraisal.

Target User Profile:

Admin staff such as the battalion command clerk that handles the update of the address book regarding changes of soldiers and updating performance record and leave application in addition to recording offences or rewards.

Commanders such as the Commanding Officer or Officer Commanding to view the list of offences and rewards committed to arrange punishment or award ceremonies.

Project Directions:

Enhanced Address Book for offences committed, rewards, performance and annual leave management used by military officials.

1. To allow creation of groups and to be able to add persons into these groups for the types of offences committed. For example, one group would be for persons who committed Do not call offences, while another would be Scam call offences. A summary describing the offences committed by each person in a particular group is possible.

Military/Class Grading system (similar structure with the first one)

1. Create groups of users who have achieved Gold, Silver or Bronze during their Physical Assessment tests, or track offences like late coming, insubordination .. etc.

User Profiles (Week 4):

Mike is an army officer in his early 20s. He has been assigned the task of updating soldiers' performance in his camp. He needs to carry out his task alone. He is not familiar with CLI, thus a GUI is needed.

Michael is an army technician in his 30s. He needs to update the system when new enlist personnel joined in or soldiers discharged from reservice. He updates the system alone, with entering the record one by one, or importing from a txt file. He also deletes a group of records in one time.

Problem Addressed:

An enhanced address book software which will include more information of the soldier such as rank, group, and performance. The user is able to manage records as a bulk without the hassle of editing records one by one.

User Stories:

As an army officer, I can add a new record for the newly enlisted soldier. (Achieved?)

As an army officer, I can delete a record when the soldier finished his reservice. (Achieved?)

As an army officer, I can update the record. (Achieved?)

As an army officer, I can create a new group, so that I can classify soldiers by the groups for different purposes.

As an army officer, I can delete a particular group, so that I can remove the unwanted group.

As an army officer, I can view a person and find out which group he belongs to, so that I can know his performance.

As an army officer, I can list all groups in the system so that I can keep track of what has been created in the system.

As an army officer, I can view a list of persons in a particular group so that I can give a punishment or a reward.

As an army officer, I can rename a particular group, so that I can update the purpose.

As an army clerk, i want to track the addresses and contact of all service members, so that i can contact them if necessary

As an army clerk, i need to track the appraisal of the service members, so that I can reward them

As an army clerk, I need to track the punishment for each service man, so that I can understand the details of his punishment and arrange for them.

As an army clerk, i need to ensure that a tag can be created for each reward/punishment for each service member so that i can understand the details of this contribution

As an army clerk, i need to be able to create different tag with different names for each soldier, so that i can associate different things to them for reward or punishment

As an army clerk, i need to be able to create a memo for each tag, so that i can view their details without losing anything specific.

As an army technician, I can import a bulk of records with the help of a txt file.

As an army technician, I can delete a certain group of records, as it is no longer required. Constraint is that all members need to be removed from the group first.

As an army technician, I can export the records as a txt file.

V1.2 user stories to achieve:

As an army officer, I can add a new record for the newly enlisted soldier. (Achieved?)

As an army officer, I can delete a record when the soldier finished his reservice. (Achieved?)

As an army officer, I can update the record. (Achieved?)

As an army officer, I can create a new group, so that I can classify soldiers by the groups for different purposes.

As an army officer, I can delete a particular group, so that I can remove the unwanted group.

As an army officer, I can view a person and find out which group he belongs to, so that I can know his performance.

As an army officer, I can list all groups in the system so that I can keep track of what has been created in the system.

V1.3

Load groups from json file even though there is no person assigned to the group. - Jiancheng
Delete the group, and assign the soldiers under the group into the Default group - Januarius
Delete all persons under the group without deleting the group itself. - Jun Wen
Rename a group which will update the soldiers under the group at the same time. - Shijie

Update Developer Guide (do after v1.3)

Jun Wen - (C-Help)
Jian Cheng - (C-DetectDuplicates)
Shi Jie - (C-Tagging)
Janurius - (C-Update)

Feature Lists (Week 5)

- Create group/tag - to create a new group which can have 0 to many NS men
- List person - to show the details of the selected person with information such as his group and performance
- List all groups/tag - to list all of the groups created in the system.
- Rename group/tag - To rename an existing group.
- List all persons in a group/tag - To see all persons in a particular group.
- Delete group/tag - to delete specific group based on given ticket number
- Help guide - to view the list of command and their usage
- Exit - to quit and exit the program

A group is a generic container for a list of related tags
A tag can be created for a group
A tag can be associated with an NS man after it has been created for a group.
An NS man cannot be tagged to a tag belonging to a group he is not in

Example:
(Tags are in lowercase letters, Groups are in UPPERCASE)

FITNESS

- Ippt gold
- Ippt silver

COMBAT

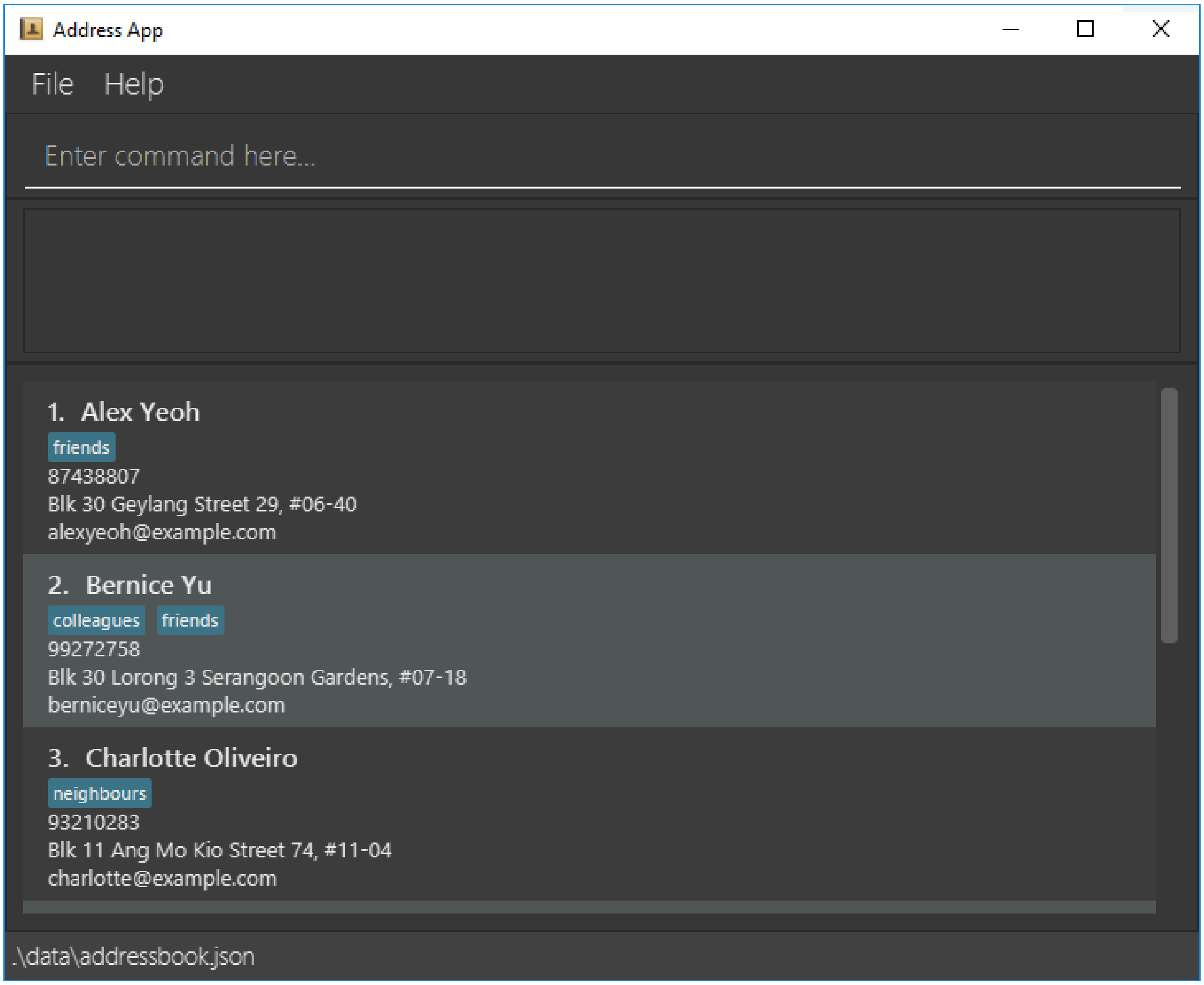
- Hand to hand combat training
- Marksman

User Guide

~~AddressBook NS Edition (ABNS) is a **desktop app for managing NS per, optimized for use via a Command 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 `11` or above installed in your Computer.~~
2. ~~Download the latest `addressbook.jar` from here.~~
3. ~~Copy the file to the folder you want to use as the *home folder* for your AddressBook.~~
4. ~~Double-click the file to start the app. The GUI similar to the below should appear in a few seconds. Note how the app contains some sample data.~~



1. Alex Yeoh
friends
87438807
Blk 30 Geylang Street 29, #06-40
alexyeoh@example.com

2. Bernice Yu
colleagues friends
99272758
Blk 30 Lorong 3 Serangoon Gardens, #07-18
berniceyu@example.com

3. Charlotte Oliveiro
neighbours
93210283
Blk 11 Ang Mo Kio Street 74, #11-04
charlotte@example.com

.\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 p/98765432 e/johnd@example.com a/John street, block 123, #01-01` : Adds a contact named `John Doe` to the Address Book.~~
   - ~~**`delete`**`3` : Deletes the 3rd contact shown in the current list.~~
   - ~~**`clear`** : Deletes all contacts.~~

- ~~`exit` : Exits the app.~~

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

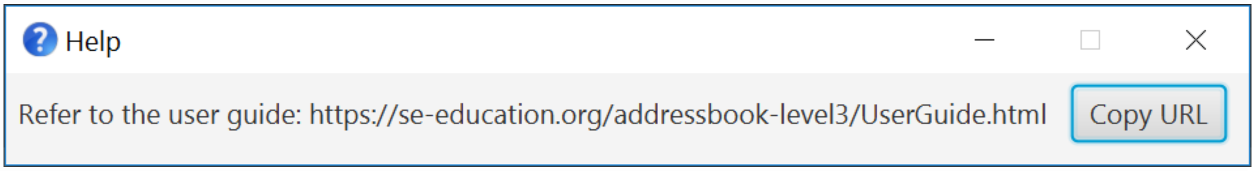
Features

ℹ️ **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.
- If a parameter is expected only once in the command but you specified it multiple times, only the last occurrence of the parameter will be taken.
  e.g. if you specify `p/12341234 p/56785678`, only `p/56785678` will be taken.
- 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`.

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`

Creating a group: `create -g`

Creates a new group to the address book.

Format: `create -g n/NAME`

Examples:

- `create -g n/FITNESS`
- `create -g n/COMBAT`

Listing all groups : `list all existing groups`

Shows a list of all groups in the address book.

Format: `listGroup all`

~~Rename the group : `rename -g`~~

~~Rename an existing group to another name.~~

~~Format: `rename INDEX [n/NAME]`~~

~~Rename the group at the specified `INDEX`. The index refers to the index number shown in the displayed group list. The index **must be a positive integer** 1, 2, 3, …~~

- ~~Existing values will be updated to the input values.~~

~~Examples:~~

- ~~`rename 1 n/PERFORMANCE` Rename the group with index 1 into `PERFORMANCE`.~~

List all persons in a group: `list -g [group name]`

Lists all persons in a particular group.

Format: `list -g [group_name]`

- The group name is case-insensitive. e.g `IPPT` will match `ippt`

Add a person to the group: `assignptg n/[person_name] g/[group_name]`

Add a person to a particular group.

Format: `assignptg n/[person_name] g/[group_name]`

- The person name is case-insensitive. e.g `James` will match `james`
- `The group name is case-insensitive e.g. Group will match group`

Deleting a group: `delete -g`

Deletes the specified group from the address book.

Format: `delete INDEX`

- Deletes the group 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 group in the address book.
- `List all` followed by `delete 1` deletes the 1st group in the results of the `find` command

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 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.

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 p/PHONE_NUMBER e/EMAIL a/ADDRESS [t/TAG]…` e.g., `add n/James Ho p/22224444 e/jamesho@example.com a/123, Clementi Rd, 1234665 t/friend t/colleague`
Clear	`clear`
Delete	`delete INDEX` e.g., `delete 3`

Edit	`edit INDEX [n/NAME] [p/PHONE_NUMBER] [e/EMAIL] [a/ADDRESS] [t/TAG]…` e.g.,`edit 2 n/James Lee e/jameslee@example.com`
Find	`find KEYWORD [MORE_KEYWORDS]` e.g., `find James Jake`
List	`list`
Help	`help`

MPV (First Iteration):

1. Create group	Jun Wen
2. Add person to group	Zheng Shijie
3. List all groups	Jun Wen
4. List all persons in group	Jiancheng
5. Delete group	Januarius

JW Bug Fix - Create Group , Show All Group Created, Delete All Person in Group
JC Big Fix -