VolunCHeer - User Guide

By: `CS2113T T08-01` Since: `Feb 2019` Licence: `MIT`

1. Introduction

Hello and welcome to VolunCHeer! VolunCheer is a desktop application for project managers who want **an easy way to keep track of their beneficiaries, volunteers, and any ongoing / upcoming projects.**

VolunCheer is optimised for users who prefer a **Command Line Interface (CLI)** while still being able to view important data on the **Graphical User Interface(GUI)**. If that suits your tastes come give it a try!

We guarantee that VolunCHeer will save you the need for multiple documents and folders just to store all your information. And with our ability to **filter out suitable volunteers based on their data**, you can effectively say goodbye to more cumbersome sorting methods like Excel!

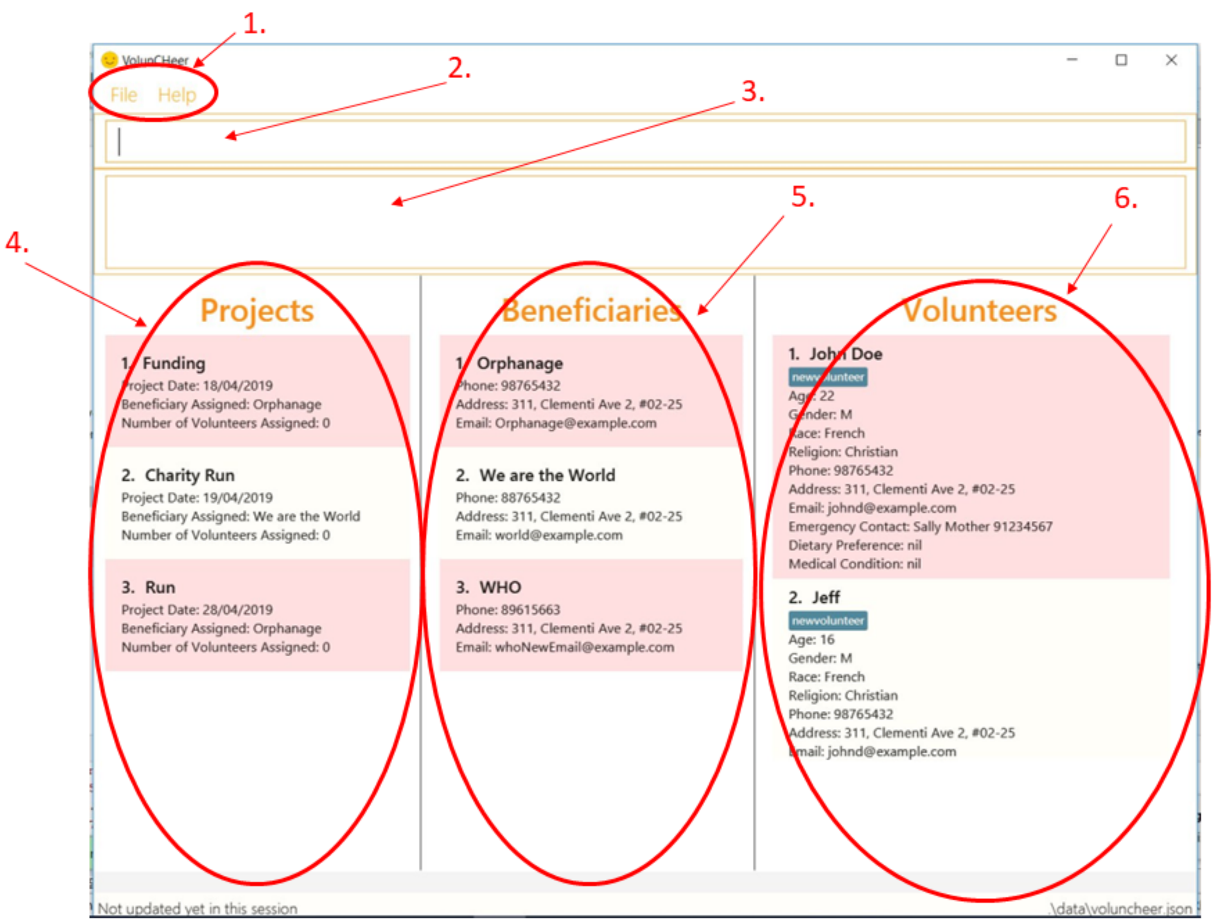
Eager to get started? Click on Section 2. "Quick Start" to jump ahead, though we highly recommend reading the overview below!

1.1. Overview

To facilitate your reading, we have a few icons to take note of:

TIP	Useful tips are generally located here!
IMPORTANT	Information here is quite important and should be carefully noted.
NOTE	Important information can be found here.

Now let us go though a quick overview of our VolunCHeer application.



Projects

1. Funding
Project Date: 18/04/2019
Beneficiary Assigned: Orphanage
Number of Volunteers Assigned: 0

2. Charity Run
Project Date: 19/04/2019
Beneficiary Assigned: We are the World
Number of Volunteers Assigned: 0

3. Run
Project Date: 28/04/2019
Beneficiary Assigned: Orphanage
Number of Volunteers Assigned: 0

Beneficiaries

1. Orphanage
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

2. We are the World
Phone: 88765432
Address: 311, Clementi Ave 2, #02-25
Email: world@example.com

3. WHO
Phone: 89615663
Address: 311, Clementi Ave 2, #02-25
Email: whoNewEmail@example.com

Volunteers

1. John Doe
newvolunteer
Age: 22
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Emergency Contact: Sally Mother 91234567
Dietary Preference: nil
Medical Condition: nil

2. Jeff
newvolunteer
Age: 16
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com

Not updated yet in this session
.\data\voluncheer.json

Figure 1. VolunCHeer user interface

1. Menu Bar: This is where you can exit the app (under File) or find the Help page
2. Command Box: This is where you will be typing in your commands.
3. Command Result: This is where the results of all your commands are displayed.

TIP	Useful tips and error messages are also shown here. If it seems like somethings is not working, try looking here!

4. Project List: This column shows the list of projects you have currently stored in VolunCHeer.
5. Beneficiary List: This column shows the list of beneficiaries currently in VolunCHeer.
6. Volunteer List: Similar to the other lists, this one shows the list of beneficiaries you currently have.

2. Quick Start

To start using VolunCHeer, all you have to do is follow these simple steps below:

1. Ensure you have Java version `9` or later installed in your Computer. If you are not sure what version you have, check out this site for more info.
2. Download the latest `VolunCHeer.jar` here.

3. Copy the file to whichever folder you would like to store VolunCHeer in.
4. Double-click the file to start the app. The GUI should appear as below in a few seconds.

   [UiClean] | *UiClean.png*

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. Some example commands you can try:
   - **`help`** : get a list of all the commands we have (highly recommended).
   - **`addProject`** n/Project Sunshine d/20190320: adds a project named "Project Sunshine" in the project list.
   - **`deleteProject2`** : deletes the 2nd project portfolio in the current list of projects.
   - **`exit`** : exits the app
7. For a more detailed explanation of each command please refer to Section 3, “Features” .

3. Features

Before we go on to explain our features, this section will give a brief introduction on how to interpret our explanations.

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 you supply, like `add n/John Doe`. Parameters are generally necessary, unless mentioned as below.
- Parameters in square brackets are **optional** e.g in `addVolunteer n/NAME [t/TAG]`, we can input `addVolunteer n/John Doe t/friend` or `addVolunteer n/John Doe`.
- Items with `…` after them can be used multiple times, including zero. e.g. `[t/TAG]…` can be used as   (i.e. 0 times), or `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 perfectly acceptable.

3.1. Viewing help : `help`

If you are stuck and cannot figure out what to do, do not fear! Instead of screaming for help, simply type it into the command bar and we will give you everything you need to use this app.

Format: `help`

3.2. Project Management

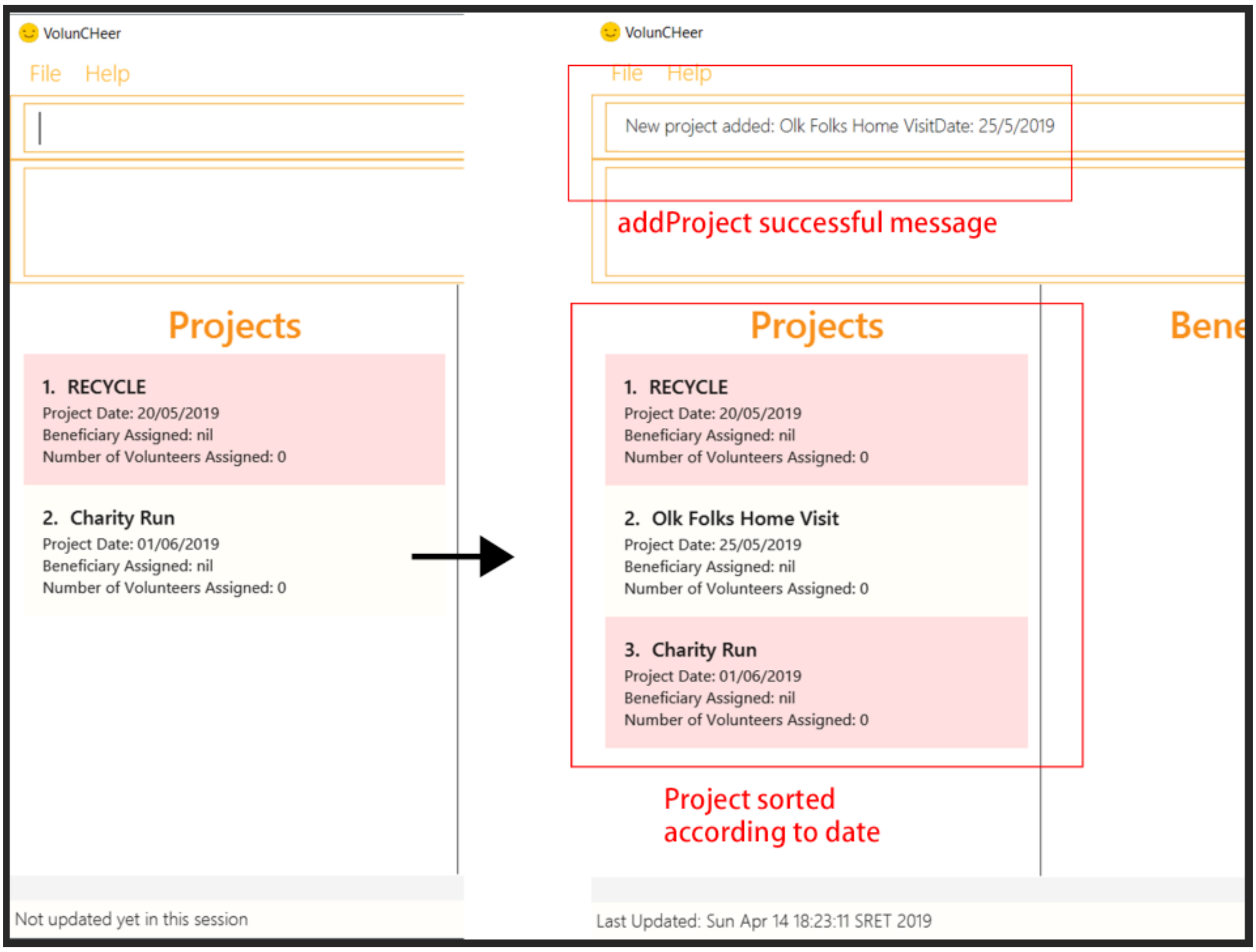
3.2.1. Adding a project: `addProject` / `ap`

One of the first things to do when you use the app is to start adding projects to track, and this is the command to use.

Format: `addProject n/PROJECT_NAME d/DATE`

- Please enter DATE in DD/MM/YYYY format, making sure that the date should be after today.
- Project list does not accept duplicates in Project Titles, so make sure you name everything differently!
- Projects are automatically sorted in ascending date order for easier tracking or Project tasks.

Now let us look at what happens when the command `addProject p/Old Folk Home Visit d/25/05/2019` is entered on screen.


VolunCHeer
File Help
Projects
Bene
addProject successful message
Project sorted according to date

New project added: Olk Folks Home VisitDate: 25/5/2019

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Charity Run
Project Date: 01/06/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Not updated yet in this session

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Olk Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

3. Charity Run
Project Date: 01/06/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Last Updated: Sun Apr 14 18:23:11 SRET 2019

Figure 2. When `addProject p/Old Folk Home Visit d/25/05/2019` is executed.

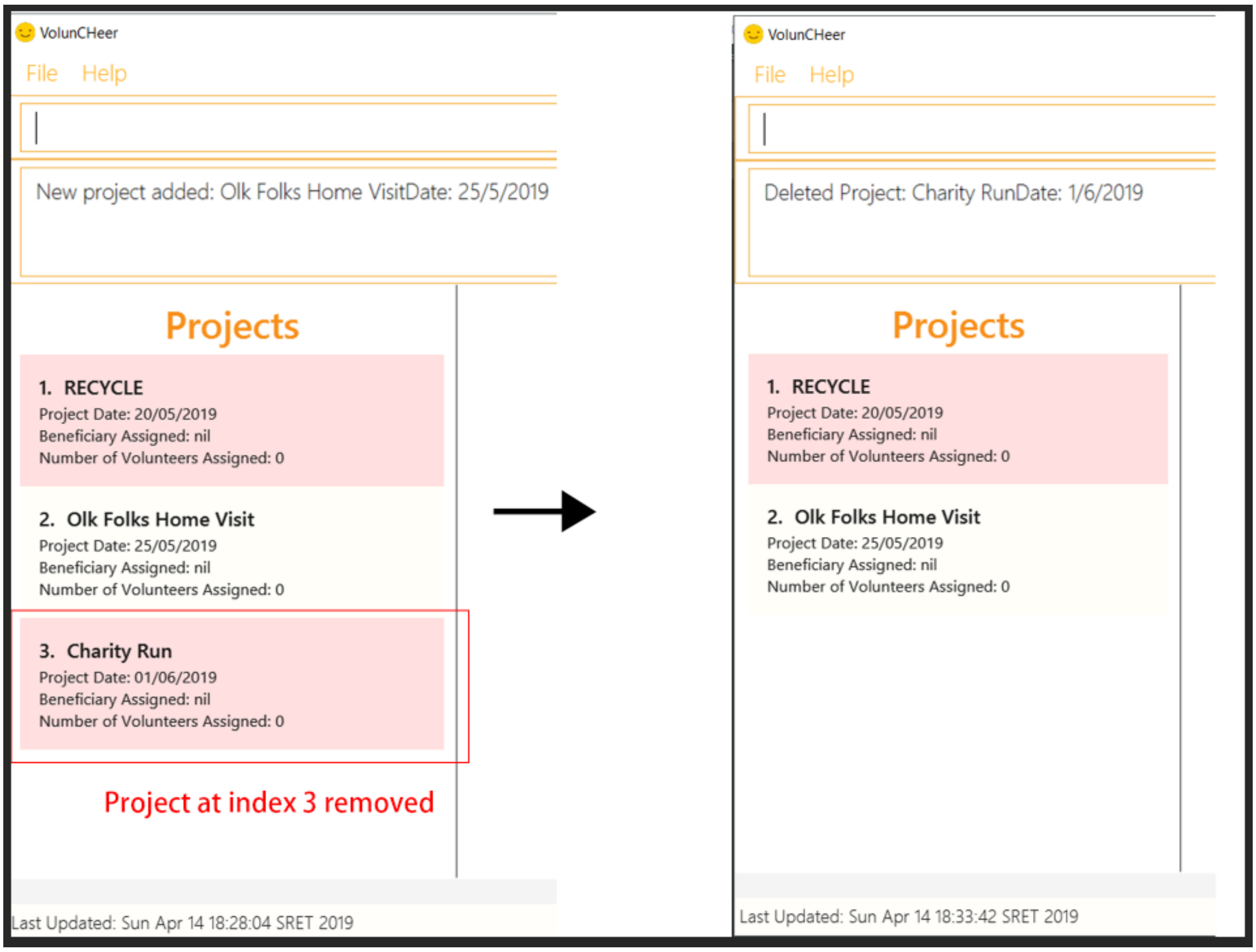
3.2.2. Deleting a project : `deleteProject` / `dp`

When a project is completed or cancelled, VolunCHeer allows you to easily delete it by stating the project order in the list.

Format: `deleteProject INDEX`

- This INDEX refers to the index of the project in the project list. If you are unsure of the order, **PLEASE** use 'listProject' to view all projects and get the correct index. If you delete the wrong projects, please refer to Section 3.7, "Undoing previous command : `undo`".
- Error message is shown if the INDEX entered is invalid

This is how the project list changes upon execution.


VolunCHeer

New project added: Olk Folks Home VisitDate: 25/5/2019

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Olk Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

3. Charity Run
Project Date: 01/06/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Project at index 3 removed

Last Updated: Sun Apr 14 18:28:04 SRET 2019

VolunCHeer

Deleted Project: Charity RunDate: 1/6/2019

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Olk Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Last Updated: Sun Apr 14 18:33:42 SRET 2019

Figure 3. When `deleteProject 3` command is executed.

3.2.3. Listing all projects : `listProject` / `lp`

When you want to take a look at all your projects, this command helps you do so.

Format: `listProject`

3.2.4. Assigning a Beneficiary to Project: `assignB`

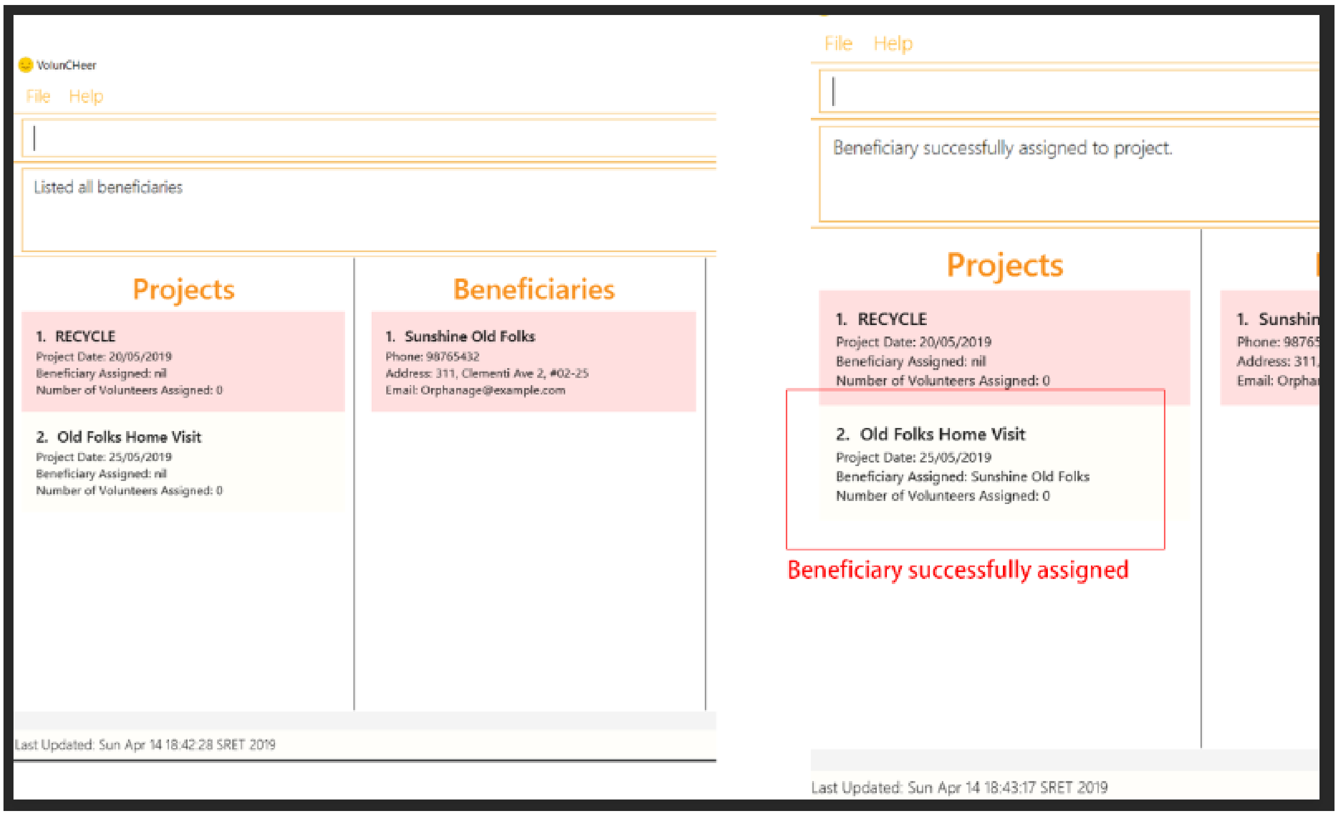
Projects are generally associated with certain beneficiaries. VolunCHeer allows you to attach them easily with this command. It assigns the Beneficiary at the provided INDEX to the Project with ProjectTitle indicated.

Format: `assignBeneficiary p/PROJECT_TITLE i/INDEX`

- The assigned Beneficiary can then be seen under the Project card as shown below.
- There can be only one beneficiary for each project, however, one beneficiary can be assigned to multiple projects.

IMPORTANT The index **must be a positive integer** `1, 2, 3, …`

After assigning a beneficiary, the project will have its data updated as seen below.


VolunCHeer

Listed all beneficiaries

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Beneficiaries

1. Sunshine Old Folks
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

Last Updated: Sun Apr 14 18:42:28 SRET 2019

Beneficiary successfully assigned to project.

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 0

Beneficiary successfully assigned

1. Sunshin
Phone: 98765
Address: 311,
Email: Orpha

Last Updated: Sun Apr 14 18:43:17 SRET 2019

Figure 4. When `assignB p/Old Folks Home Visit i/1` command executed

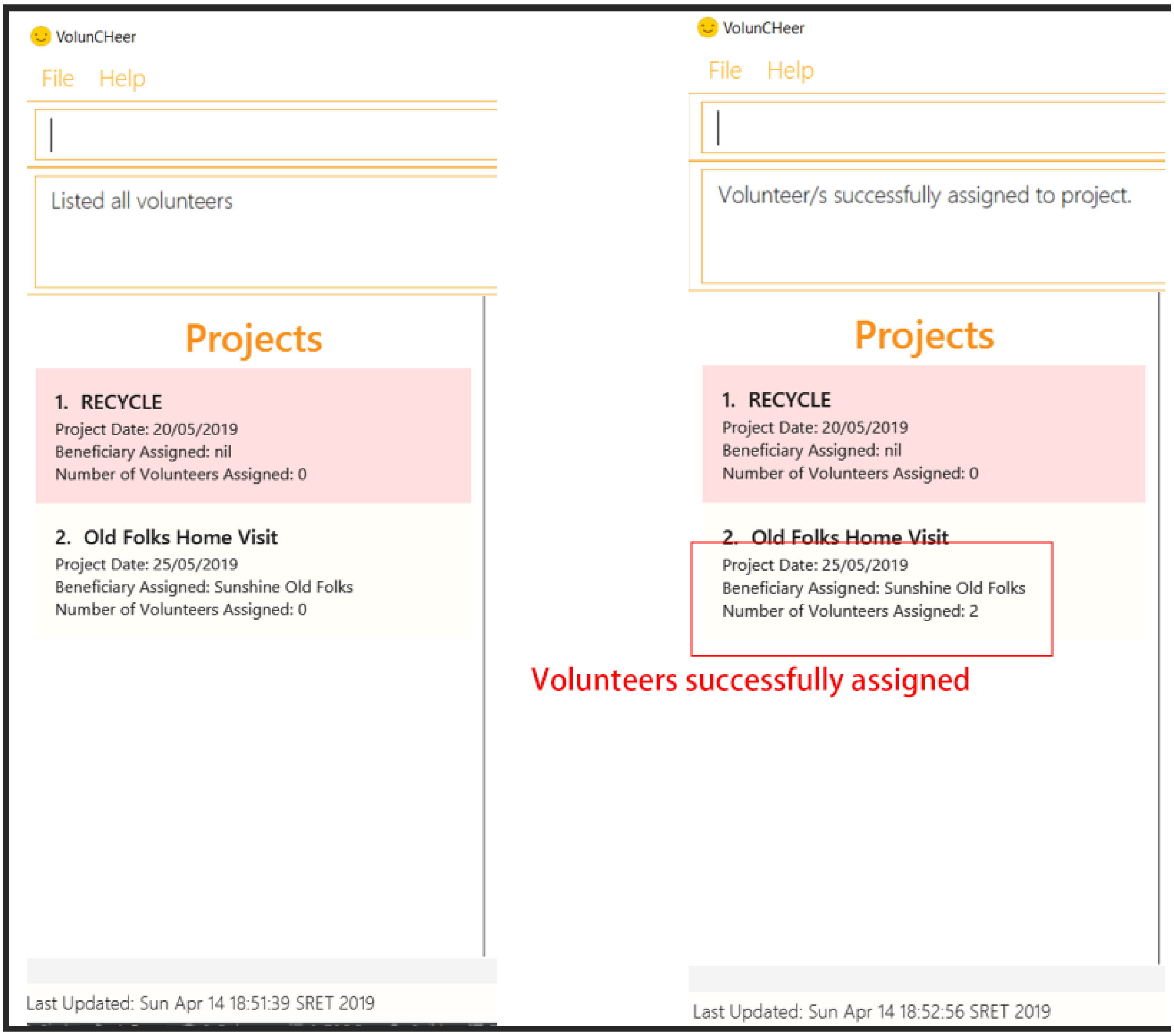
TIP Use listBeneficiary to view a full list of Beneficiary to assign. Use summaryBeneficiary command to view the Projects attached to each Beneficiary.

3.2.5. Assigning one or more Volunteers to Project: `assignV`

We also provide an easy method to assign a specific number of volunteers to the indicated Project

Format: `assignVolunteer p/PROJECT_TITLE rv/REQUIRED_NUMBER_OF_VOLUNTEERS`

TIP The number of volunteers assigned to the Project can be seen under the Project card as shown below.


VolunCHeer

Listed all volunteers

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 0

Last Updated: Sun Apr 14 18:51:39 SRET 2019

VolunCHeer

Volunteer/s successfully assigned to project.

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 2

Volunteers successfully assigned

Last Updated: Sun Apr 14 18:52:56 SRET 2019

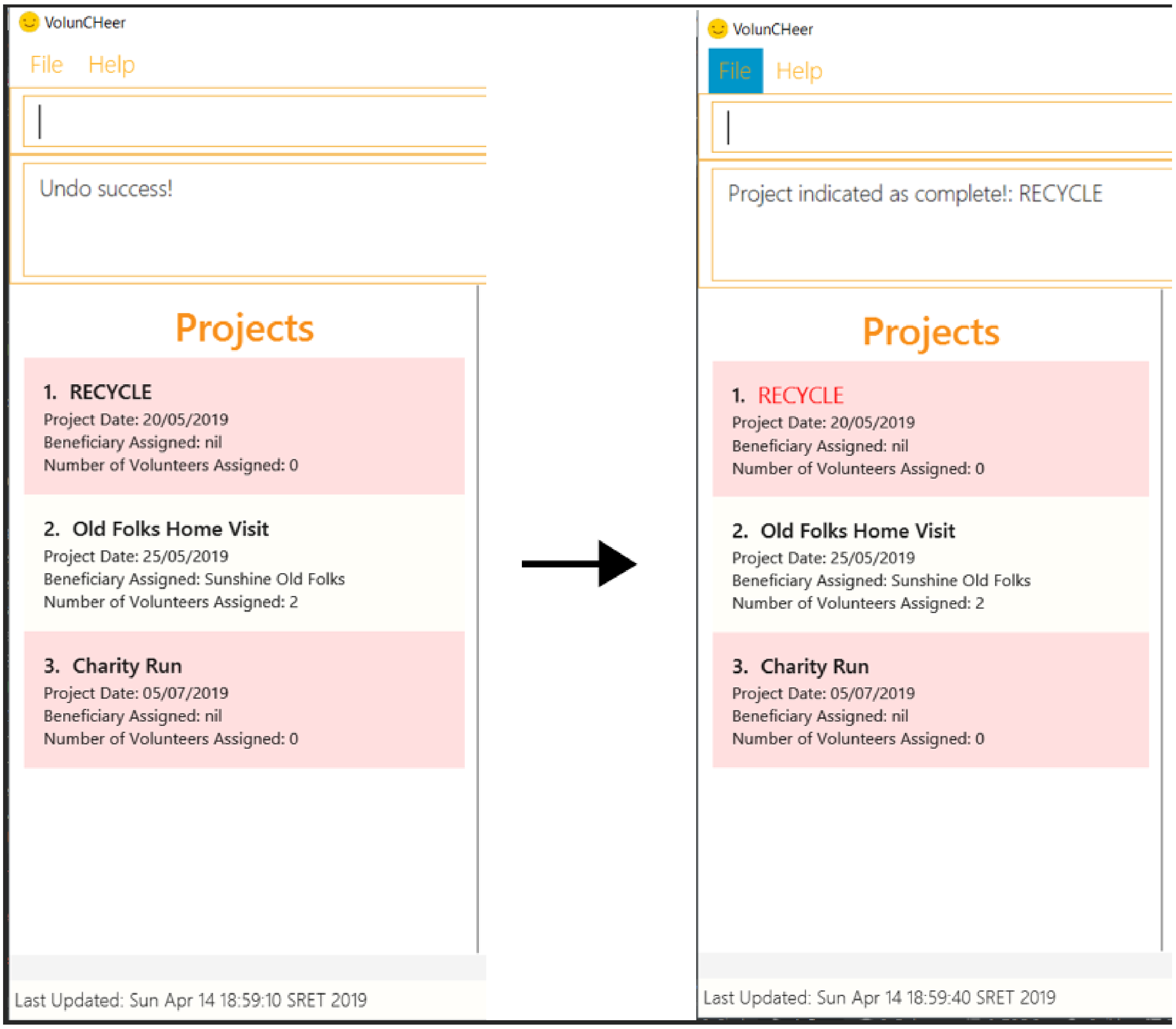
Figure 5. When `assignV p/Old Folks Home Visit rv/2` *is executed.*

TIP Use the commands listed in Section 3.5, “Filtering & Exporting” to filter out the desired list of volunteers.

3.2.6. Mark project as complete: 'complete'

Once a `project` is done, you can mark it as complete to distinguish it from your other projects. Simply provide an INDEX to indicate which project you would like to complete.

Format: `complete i/INDEX`



Undo success!

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 2

3. Charity Run
Project Date: 05/07/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Last Updated: Sun Apr 14 18:59:10 SRET 2019

Project indicated as complete!: RECYCLE

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 2

3. Charity Run
Project Date: 05/07/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Last Updated: Sun Apr 14 18:59:40 SRET 2019

Figure 6. When `complete i/1` *command is executed*

NOTE Once marked as complete, project title will be displayed in red colour font

3.3. Beneficiary Management

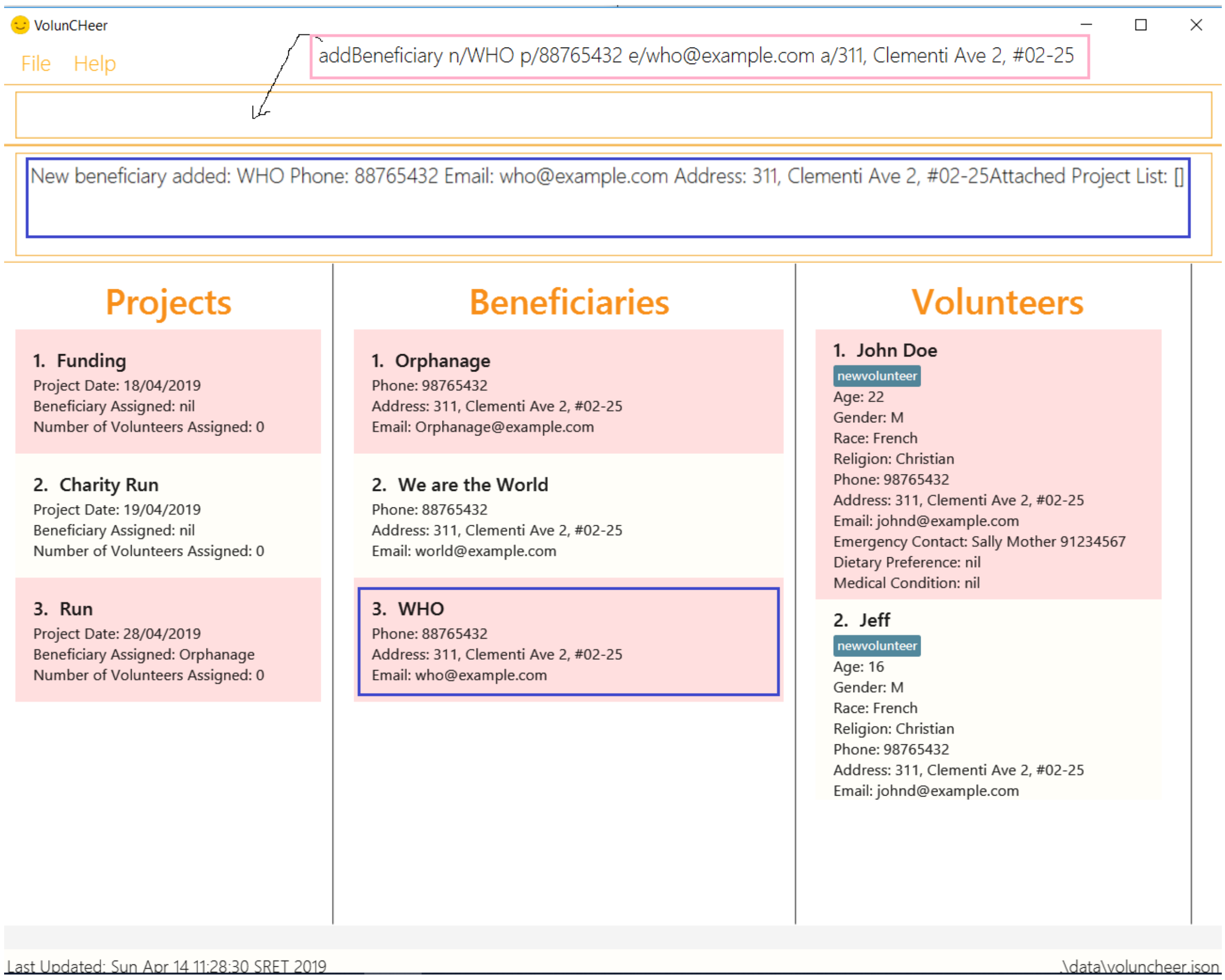
3.3.1. Adding a beneficiary: `addBeneficiary` / `ab`

Similar to the previous adding command, this adds a beneficiary to the list of Beneficiaries

Format: `addBeneficiary n/NAME a/ADDRESS e/EMAIL p/PHONE_NUMBER`

Example:

- `addBeneficiary n/Orphanage p/98765432 e/Orphanage@example.com a/311, Clementi Ave 2, #02-25`



```
addBeneficiary n/WHO p/88765432 e/who@example.com a/311, Clementi Ave 2, #02-25
```

```
New beneficiary added: WHO Phone: 88765432 Email: who@example.com Address: 311, Clementi Ave 2, #02-25Attached Project List: []
```

Projects

1. **Funding**
Project Date: 18/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. **Charity Run**
Project Date: 19/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

3. **Run**
Project Date: 28/04/2019
Beneficiary Assigned: Orphanage
Number of Volunteers Assigned: 0

Beneficiaries

1. **Orphanage**
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

2. **We are the World**
Phone: 88765432
Address: 311, Clementi Ave 2, #02-25
Email: world@example.com

3. **WHO**
Phone: 88765432
Address: 311, Clementi Ave 2, #02-25
Email: who@example.com

Volunteers

1. **John Doe**
newvolunteer
Age: 22
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Emergency Contact: Sally Mother 91234567
Dietary Preference: nil
Medical Condition: nil

2. **Jeff**
newvolunteer
Age: 16
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com

Last Updated: Sun Apr 14 11:28:30 SRET 2019 .\data\voluncheer.json

Figure 7. Add Beneficiary Command Result (pink: input, blue: output)

In the figure above, after the add command, we can observe a new beneficiary card is shown on the GUI.

- The beneficiary will be used to assign to a project, this means that the project will benefit this beneficiary, i.e. Orphanage Home, Nursing home, etc.
- When add a new beneficiary, the project lists assigned to it will be empty. You can assign projects to it by assign command stated.

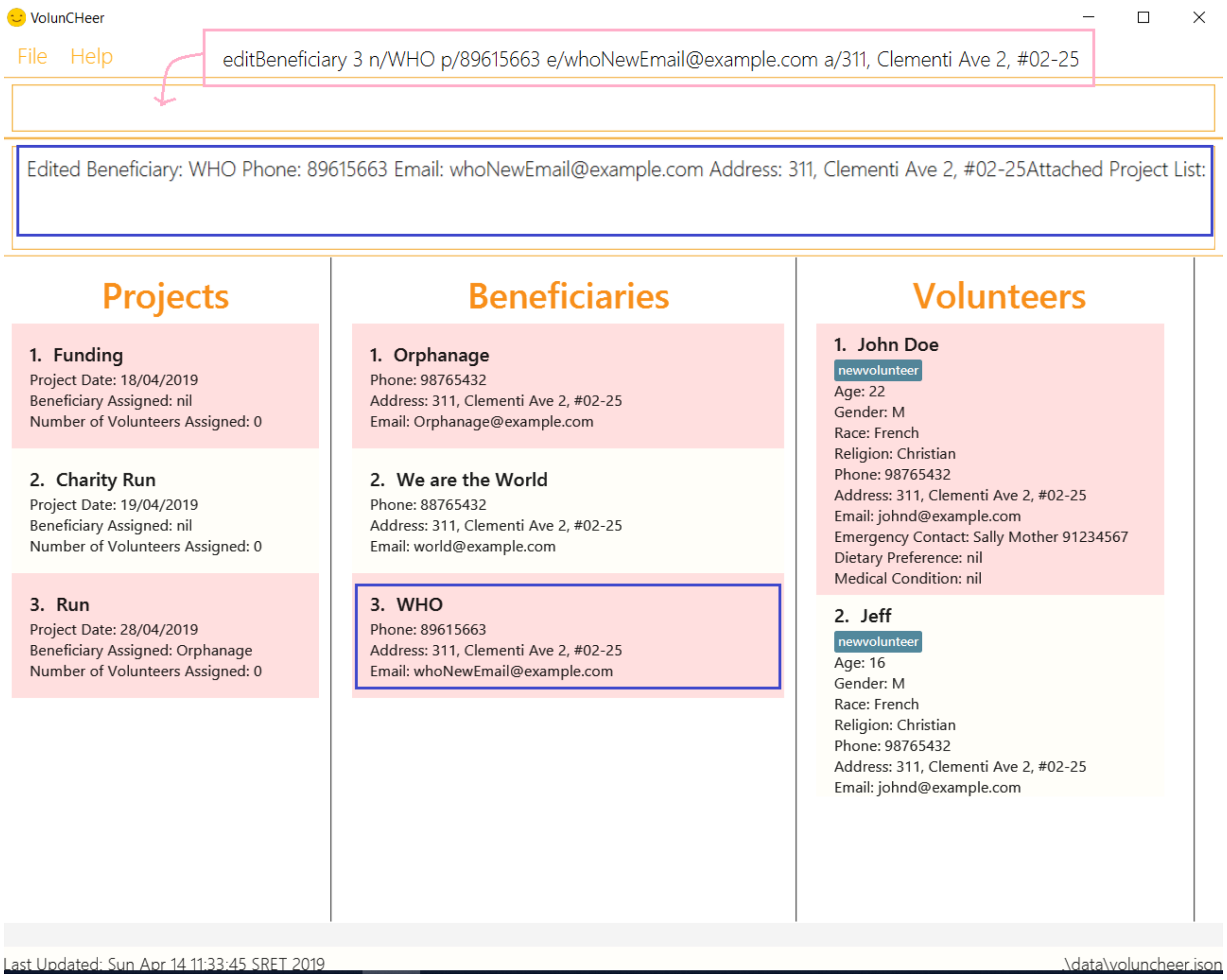
3.3.2. Editing a beneficiary: `editBeneficiary` / `eb`

In case of incorrect information, we also allow you to edit the beneficiary at the given INDEX

Format: `editBeneficiary INDEX (must be a positive integer) [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS]`

Examples:

- `editBeneficiary 1 n/Old Folk Home p/91234567`


VolunCHeer

```
editBeneficiary 3 n/WHO p/89615663 e/whoNewEmail@example.com a/311, Clementi Ave 2, #02-25
```

```
Edited Beneficiary: WHO Phone: 89615663 Email: whoNewEmail@example.com Address: 311, Clementi Ave 2, #02-25Attached Project List:
```

Projects

1. Funding
Project Date: 18/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Charity Run
Project Date: 19/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

3. Run
Project Date: 28/04/2019
Beneficiary Assigned: Orphanage
Number of Volunteers Assigned: 0

Beneficiaries

1. Orphanage
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

2. We are the World
Phone: 88765432
Address: 311, Clementi Ave 2, #02-25
Email: world@example.com

3. WHO
Phone: 89615663
Address: 311, Clementi Ave 2, #02-25
Email: whoNewEmail@example.com

Volunteers

1. John Doe
newvolunteer
Age: 22
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Emergency Contact: Sally Mother 91234567
Dietary Preference: nil
Medical Condition: nil

2. Jeff
newvolunteer
Age: 16
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com

Last Updated: Sun Apr 14 11:33:45 SRET 2019 .\data\voluncheer.json

Figure 8. Edit Beneficiary Command Result (pink: input, blue: output)

In the figure, we can see that the WHO information including phone number and email has changed, compared to the last figure.

NOTE	When a beneficiary is edited, the data of the beneficiary in its attached projects is in sync, meaning that that data is automatically updated in the mentioned projects.

3.3.3. Deleting a beneficiary: `deleteBeneficiary` / `db`

Of course, once a beneficiary is no longer associated with you, it can be removed by providing the INDEX.

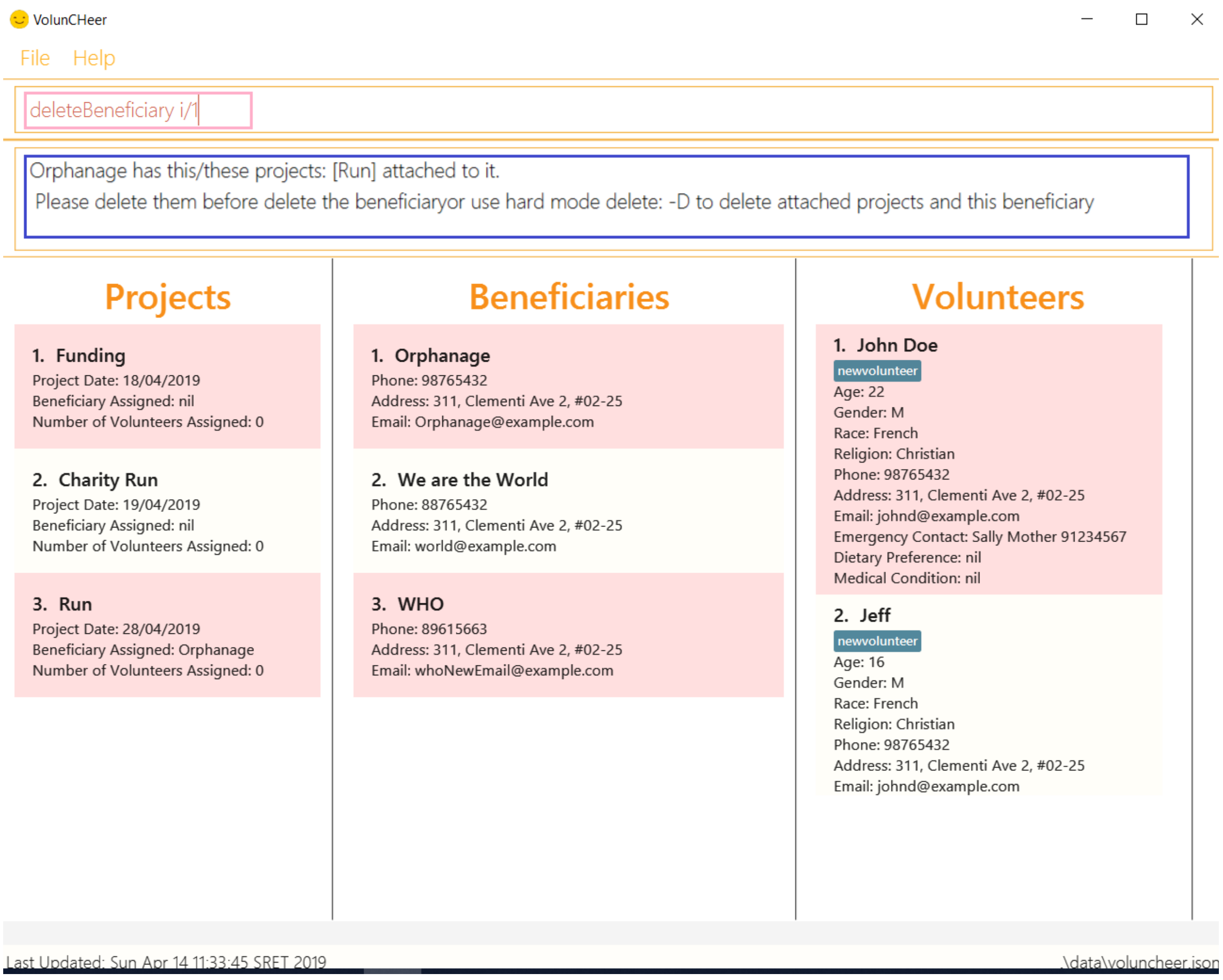
Format: `deleteBeneficiary i/INDEX -D`

IMPORTANT	`-D` is optional and should not be misused (see below)

- There are two modes of deletion: **soft delete mode** and **hard delete mode**.
- In the **soft delete mode**, there is a safe check to help you avoid deleting beneficiary that has attached projects, leaving the projects unassigned.
- In the **hard delete mode**, the beneficiary and all its attached projects will be deleted.
- Default is **soft delete mode**. To switch to **hard delete mode**, include `-D` in your command.

Examples:

- `deleteBeneficiary i/1` **soft delete mode**
- `deleteBeneficiary i/1 -D` **hard delete mode**



```
deleteBeneficiary i/1
```

```
Orphanage has this/these projects: [Run] attached to it.
 Please delete them before delete the beneficiaryor use hard mode delete: -D to delete attached projects and this beneficiary
```

Projects

1. Funding
Project Date: 18/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Charity Run
Project Date: 19/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

3. Run
Project Date: 28/04/2019
Beneficiary Assigned: Orphanage
Number of Volunteers Assigned: 0

Beneficiaries

1. Orphanage
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

2. We are the World
Phone: 88765432
Address: 311, Clementi Ave 2, #02-25
Email: world@example.com

3. WHO
Phone: 89615663
Address: 311, Clementi Ave 2, #02-25
Email: whoNewEmail@example.com

Volunteers

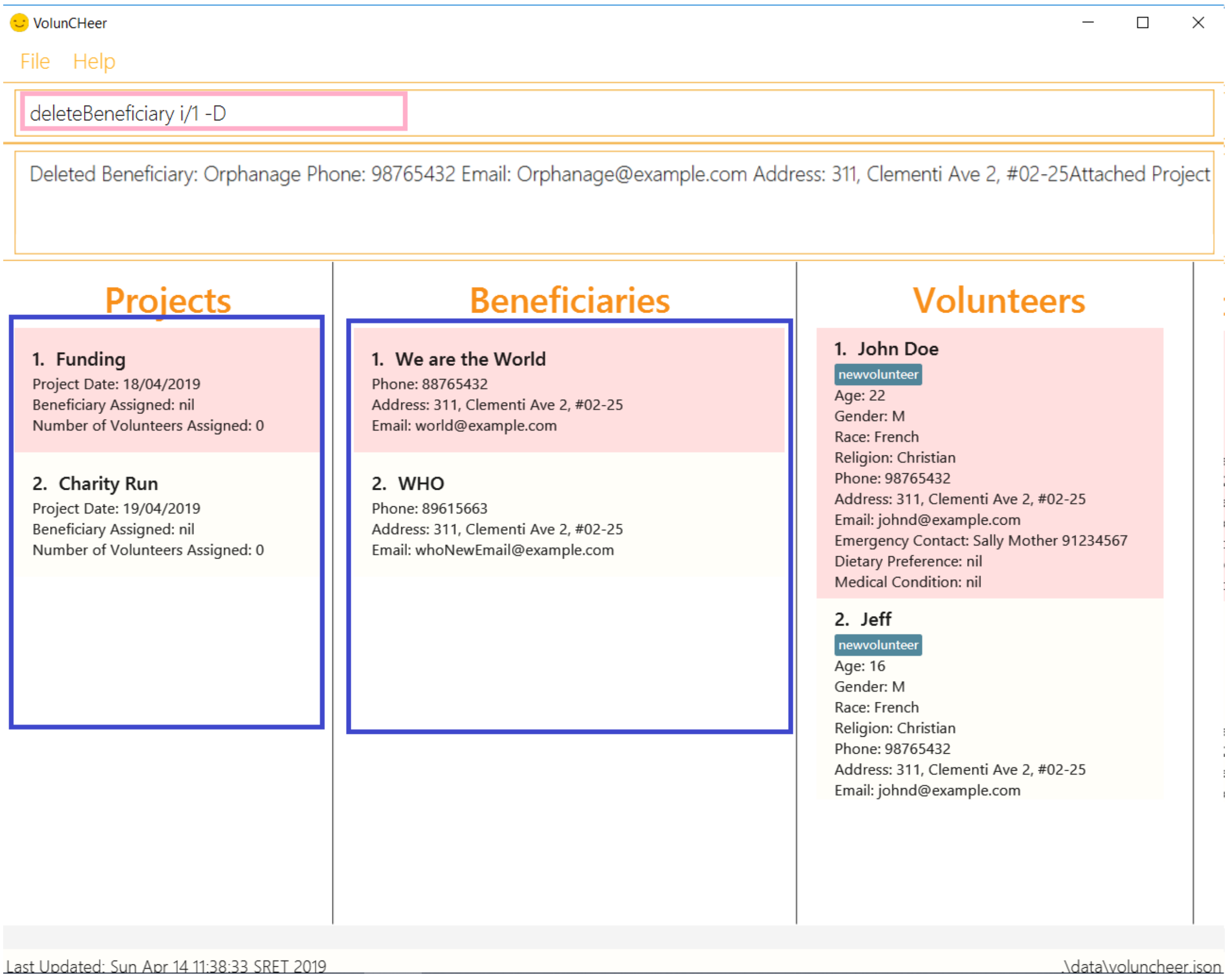
1. John Doe
newvolunteer
Age: 22
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Emergency Contact: Sally Mother 91234567
Dietary Preference: nil
Medical Condition: nil

2. Jeff
newvolunteer
Age: 16
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com

Last Updated: Sun Apr 14 11:33:45 SRET 2019
.\data\voluncheer.json

Figure 9. Delete Beneficiary Command (Soft Delete Mode) Result (pink: input, blue: output)

In Figure 3, we are trying to soft delete a beneficiary which was assigned to project **Run**. Hence, a message appears and informs us to switch to hard delete mode.


VolunCHeer

File Help

deleteBeneficiary i/1 -D

Deleted Beneficiary: Orphanage Phone: 98765432 Email: Orphanage@example.com Address: 311, Clementi Ave 2, #02-25Attached Project

Projects

1. Funding
Project Date: 18/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Charity Run
Project Date: 19/04/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Beneficiaries

1. We are the World
Phone: 88765432
Address: 311, Clementi Ave 2, #02-25
Email: world@example.com

2. WHO
Phone: 89615663
Address: 311, Clementi Ave 2, #02-25
Email: whoNewEmail@example.com

Volunteers

1. John Doe
newvolunteer
Age: 22
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Emergency Contact: Sally Mother 91234567
Dietary Preference: nil
Medical Condition: nil

2. Jeff
newvolunteer
Age: 16
Gender: M
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com

Last Updated: Sun Apr 14 11:38:33 SRET 2019 .\data\voluncheer.json

Figure 10. Delete Beneficiary Command (Hard Delete Mode) Result (pink: input, blue: output)

In Figure 4, the beneficiary and its attached projects have been deleted successfully.

3.3.4. Listing all beneficiaries: `listBeneficiary` / `lb`

As before, you can show a list of all Beneficiaries in the beneficiary pool.

Format: `listBeneficiary`

TIP The command can be used to get back to full list after several commands which change the list.

3.3.5. Locating beneficiaries by name: `findBeneficiary` / `fb`

TO facilitate searching for beneficiary, you can locate a specific one easily with via given keyword/keywords.

Format: `findBeneficiary KEYWORD [MORE_KEYWORDS]`

- The search is case insensitive. e.g `orphanage` will match `Orphanage`
- The order of the keywords does not matter. e.g. `Orphanage Nursing` will match `Nursing Orphanage`
- Only the name is searched.
- Only full words will be matched e.g. `Orphan` will not match `Orphanage`
- beneficiaries matching at least one keyword will be returned (i.e. `OR` search). e.g. `Orphanage Nursing` will return `Orphanage Rainbow` and `Nursing Home`

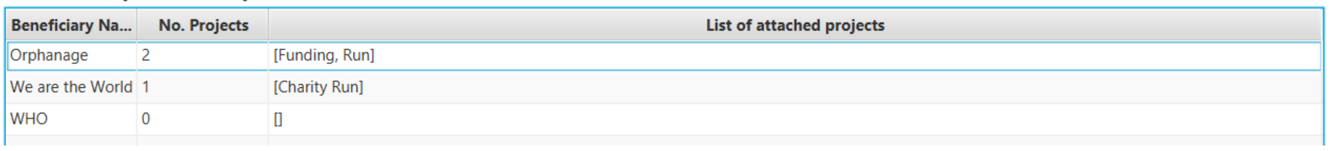
Examples:

- `find Nursing`
  Returns `Nursing Home` and `Nursing Center`

3.3.6. Summarising all beneficiaries: `summariseBeneficiary` / `sb`

Sometimes we have a beneficiary assigned to many projects and we just want to see a list of everything it is attached to. This command opens a pop up summary table of the beneficiaries for easy view. You can use even the arrow in header cells **number of Projects** to sort beneficiaries by the number of attached projects.

Format: `summariseBeneficiary`



Beneficiary Na...	No. Projects	List of attached projects
Orphanage	2	[Funding, Run]
We are the World	1	[Charity Run]
WHO	0	[]

Figure 11. Beneficiary Summary Table

TIP The command can be used to consider future partners or fundraising.

3.4. Volunteer

3.4.1. Adding a volunteer: `addVolunteer` / `av`

As like before, this adds a volunteer to the volunteer pool

Format: `addVolunteer n/NAME y/AGE g/GENDER r/RACE [rg/RELIGION] a/ADDRESS e/EMAIL p/PHONE_NUMBER ec/EMERGENCY_CONTACT [dp/DIETARY_PREFERENCE] [m/MEDICAL_CONDITION] [t/TAG]…`

Alternative Format: `av n/NAME y/AGE g/GENDER r/RACE [rg/RELIGION] a/ADDRESS e/EMAIL p/PHONE_NUMBER ec/EMERGENCY_CONTACT [dp/DIETARY_PREFERENCE] [m/MEDICAL_CONDITION] [t/TAG]…`

- "Add Successful!" message is prompted upon successfully adding a volunteer
- An invalid message will be prompted if a Volunteer with the same exact name is present in the existing database
- Parameters for Religion, Dietary Preference, Medical Condition are optional and set to 'nil' by default

TIP A volunteer can have any number of tags (including 0)

Examples:

- `addVolunteer n/John Doe y/18 g/male r/eurasian rg/christian a/John street, block 123, #01-01 e/johnd@example.com p/98765432 ec/Mary, Mother, 92221111 dp/vegetarian m/asthma`
- `av n/Sarah Soh y/22 g/female r/chinese rg/buddhist a/betsy ave 6, 02-08 e/sarah08@example.com p/92345678 ec/Johnny, Husband, 81234568`

3.4.2. Deleting a volunteer : `deleteVolunteer` \ `dv`

After a volunteer has left, it can be deleted by this command by referencing its index in the list.

Format: `deleteVolunteer INDEX`
Alternative Format: `dv INDEX`

- Deletes the volunteer at the specified `INDEX`.
- The index refers to the index number shown in the displayed volunteer list.
- The index **must be a positive integer** 1, 2, 3, …
- Error message is shown if the given index is invalid

Examples:

- `listVolunteer`
  `deleteVolunteer 2`
  Deletes the 2nd volunteer in the volunteer list.
- `findVolunteer Betsy`
  `dv 1`
  Deletes the 1st volunteer in the searched volunteer list.

TIP Use the list volunteers commands to check the correct index of the volunteer to be deleted

3.4.3. Editing a volunteer : `editVolunteer` \ `ev`

Similar to beneficiary, we can update volunteer particulars by the given index.

Format: `editVolunteer INDEX [n/NAME] [y/AGE] [g/GENDER] [r/RACE] [rg/RELIGION][p/PHONE]` `[a/ADDRESS]` `[e/EMAIL]` `[ec/EMERGENCYCONTACT]` `[dp/DIETARYPREFERENCE]` `[mc/MEDICALCONDITION]` `[t/TAG]…`

Alternative Format: `ev INDEX [n/NAME] [y/AGE] [g/GENDER] [r/RACE] [rg/RELIGION][p/PHONE] [a/ADDRESS] [e/EMAIL] [ec/EMERGENCYCONTACT] [dp/DIETARYPREFERENCE] [mc/MEDICALCONDITION] [t/TAG]…`

- Edits the volunteer at the specified `INDEX`. The index refers to the index number shown in the displayed volunteer 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 volunteer will be removed i.e adding of tags is not cumulative.
- You can remove all the volunteer's tags by typing `t/` without specifying any tags after it.

Examples:

- `editVolunteer 1 p/91234567 e/johndoe@example.com`
  Edits the phone number and email address of the 1st volunteer to be `91234567` and `johndoe@example.com` respectively.
- `ev 2 n/Betsy Crower t/`
  Edits the name of the 2nd volunteer to be `Betsy Crower` and clears all existing tags.

3.4.4. Locating volunteers by name: `findVolunteer` \ `fv`

Searching for volunteers works similarly to beneficiaries.

Format: `find KEYWORD [MORE_KEYWORDS]`

Alternative Format: `fv 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`
- volunteers matching at least one keyword will be returned (i.e. `OR` search).
- e.g. `Hans Bo` will return `Hans Gruber`, `Bo Yang`

Examples:

- `findVolunteer John`
  Returns `john` and `John Doe`
- `fv Betsy Tim John`
  Returns any volunteer having names `Betsy`, `Tim`, or `John`

3.4.5. Listing all volunteers : `listVolunteer` \ `lv`

Shows a list of all volunteers in the volunteer pool.

Format: `listVolunteer`

Alternative Format: `lv`

3.5. Filtering & Exporting

3.5.1. Assigning mapping index to each volunteer : `map`

We know that some volunteers suit a certain project better than others. To help with finding these volunteers, the map command assigns the volunteers with points 3, 2 or 1 according to the selection criteria that you set.

Format: `map t/(POINTS)(CRITERIA) t/(POINTS)(CRITERIA) t/(POINTS)(CRITERIA)`

- The t/ refers to any of the following tags.
- There are three types of tags, the age of volunteer (y/), race (r/) and medical condition (m/).
- You can enter at most 3 tags and at least 1 tag as the selection criteria.
- Each volunteer is internally assigned points which will be used used for sorting later on.
- The age criteria has comparators >,<,= which relate to the age given afterwards.
- See examples below for a clearer picture.

Examples:

- `map y/3>18 r/2chinese m/1NIL` Gives volunteers above the AGE of 18 3 points, RACE chinese 2 points and MEDICAL_CONDITION of NIL 1 point.
- `map m/3NIL` Only gives volunteers with no MEDICAL_CONDITION 3 points.

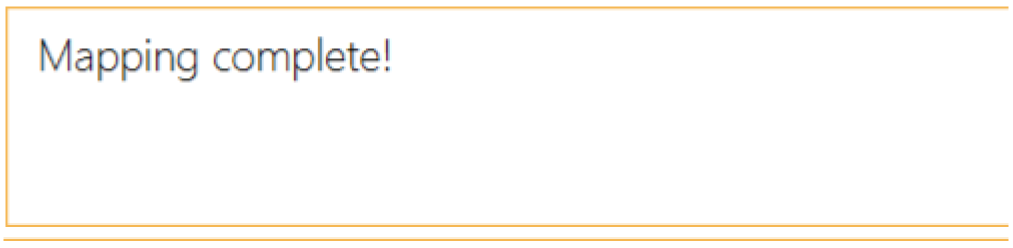

Mapping complete!

Figure 12. map command execution

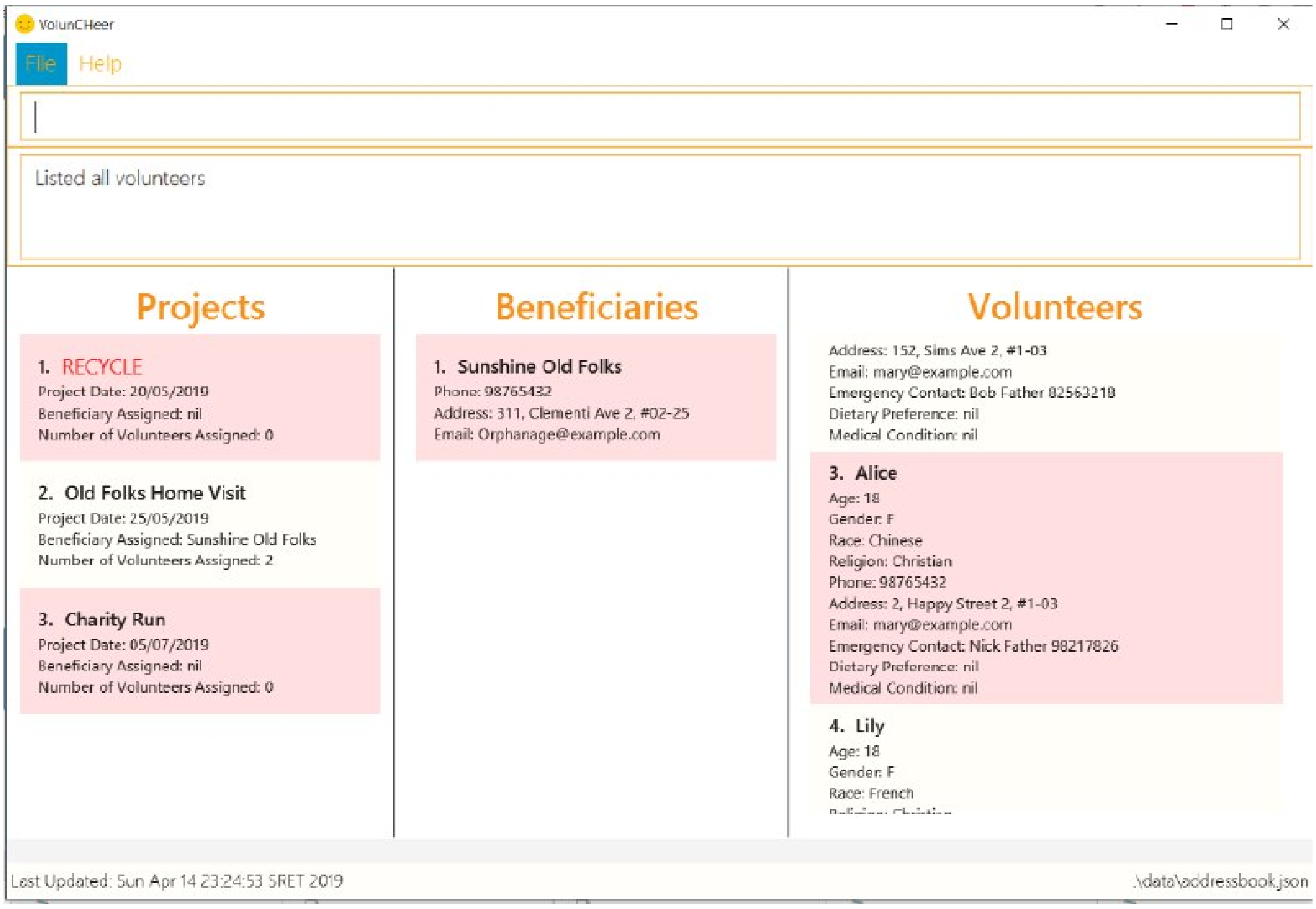
Upon executing a successful map command, the message on figure 12 will appear.

3.5.2. Sorting volunteers according to points : `sort`

After mapping, we can then sort the volunteers into order, with the most suitable volunteers being on top.

Format: `sort`

- The map function should be called before sort to generate the points
- Volunteers with equal points will not be sorted in any particular order
- Selection of the volunteers, such as with the assignV command, can be done after sorting.



Listed all volunteers

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 2

3. Charity Run
Project Date: 05/07/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Beneficiaries

1. Sunshine Old Folks
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

Volunteers

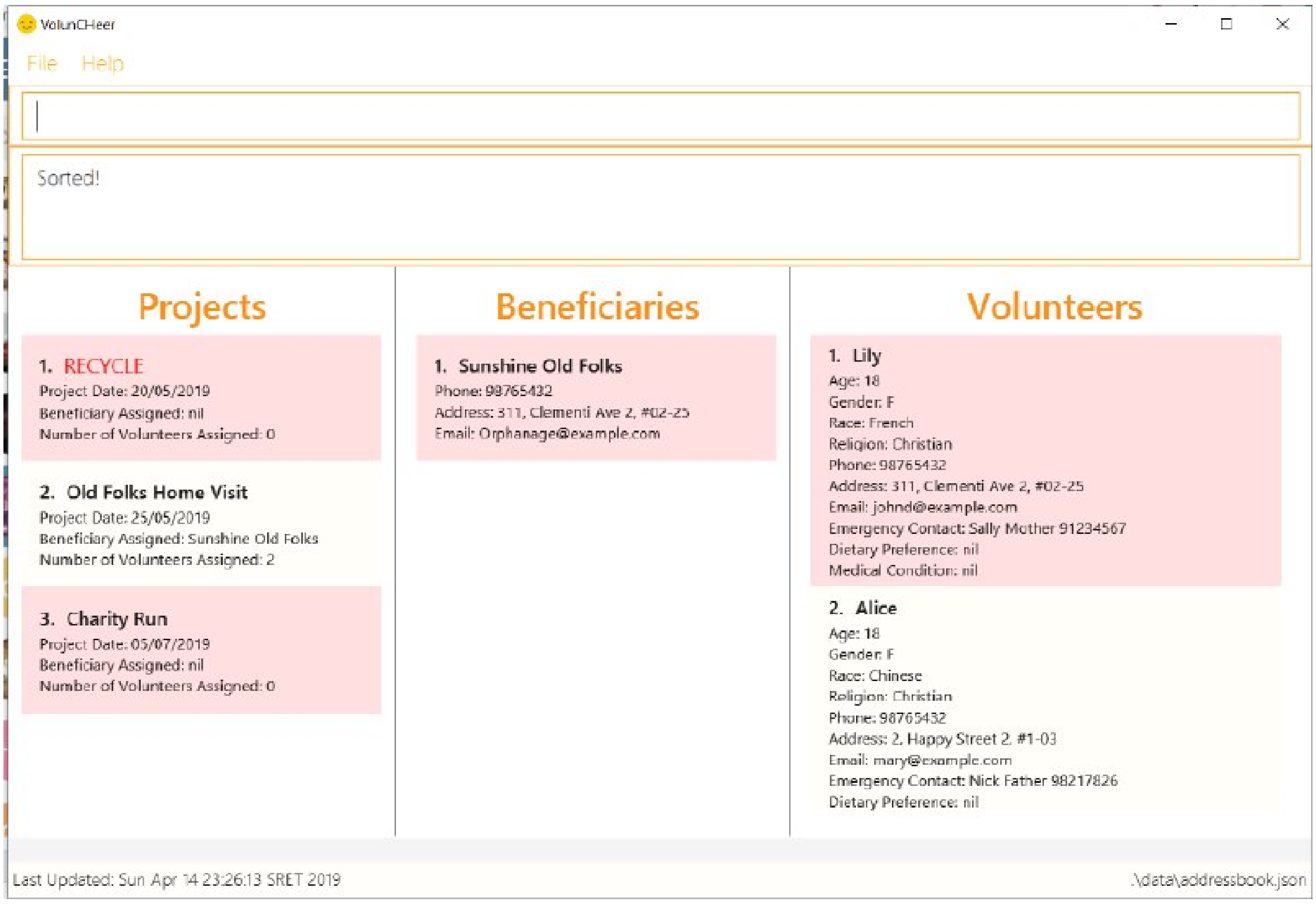
Address: 152, Sims Ave 2, #1-03
Email: mary@example.com
Emergency Contact: Bob Father 82563210
Dietary Preference: nil
Medical Condition: nil

3. Alice
Age: 18
Gender: F
Race: Chinese
Religion: Christian
Phone: 98765432
Address: 2, Happy Street 2, #1-03
Email: mary@example.com
Emergency Contact: Nick Father 98217826
Dietary Preference: nil
Medical Condition: nil

4. Lily
Age: 18
Gender: F
Race: French

Last Updated: Sun Apr 14 23:24:53 SRET 2019
.\data\addressbook.json

Figure 13. Before sorting


VolunCHeer

File Help

Sorted!

Projects

1. RECYCLE
Project Date: 20/05/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

2. Old Folks Home Visit
Project Date: 25/05/2019
Beneficiary Assigned: Sunshine Old Folks
Number of Volunteers Assigned: 2

3. Charity Run
Project Date: 05/07/2019
Beneficiary Assigned: nil
Number of Volunteers Assigned: 0

Beneficiaries

1. Sunshine Old Folks
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: Orphanage@example.com

Volunteers

1. Lily
Age: 18
Gender: F
Race: French
Religion: Christian
Phone: 98765432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Emergency Contact: Sally Mother 91234567
Dietary Preference: nil
Medical Condition: nil

2. Alice
Age: 18
Gender: F
Race: Chinese
Religion: Christian
Phone: 98765432
Address: 2, Happy Street 2, #1-03
Email: mary@example.com
Emergency Contact: Nick Father 98217826
Dietary Preference: nil

Last Updated: Sun Apr 14 23:26:13 SRET 2019 .\data\addressbook.json

Figure 14. After sorting

As can be seen in figure 13 Alice was previously at index 3. After sorting, she has shifted up to index 2 in figure 14.

3.5.3. Extracting multiple volunteers from sorted list : `extract`

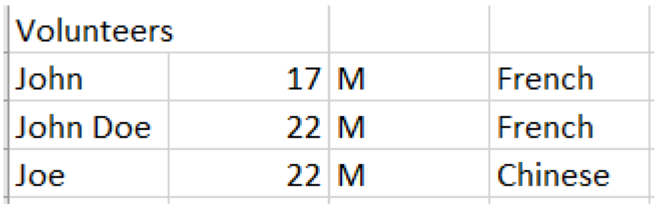
Not everyone will have VolunCHeer, which is frankly their loss. Nonetheless, this command allows you share a list of certain volunteer particulars by extracting it into a Microsoft Excel file.

Format: `extract NUMBER_OF_VOLUNTEERS t/PARTICULAR [t/OTHER_PARTICULARS]…`

- This command requires at least one type of particular from the volunteers, up to all type of particulars.
- If the NUMBER_OF_VOLUNTEERS exceeds the total number of volunteers in the list, the file will just extract all volunteers in VolunCHeer.
- This command can be called before map and sort if order is not an issue.

Examples:

*`extract [1][20]` Extracts the first 20 volunteers in the sorted list. *`extract [5][15]` Extracts volunteer number 5 to 15 in the list.



Volunteers			
John	17	M	French
John Doe	22	M	French
Joe	22	M	Chinese

Figure 15. Extracted volunteer details

The Excel file will look like figure 15.

3.6. Listing entered commands : `history`

Lists all the commands that you have entered in reverse chronological order.
Format: `history`

NOTE Pressing the <code>&uarr;</code> and <code>&darr;</code> arrows will display the previous and next input respectively in the command box.

3.7. Undoing previous command : `undo`

Restores the VolunCHeer application to the state before the previous *undoable* command was executed.
Format: `undo`

NOTE Undoable commands: those commands that modify the VolunCHeer application's main content (`addProject`, `addVolunteer`, `delete`, `edit` and `clear`).

Examples:

- `delete 1`
  `list`
  `undo` (reverses the `delete 1` command)
- `select 1`
  `list`
  `undo`
  The `undo` command fails as there are no undoable commands executed previously.
- `delete 1`
  `clear`
  `undo` (reverses the `clear` command)
  `undo` (reverses the `delete 1` command)

3.8. Redoing the previously undone command : `redo`

Reverses the most recent `undo` command.
Format: `redo`

Examples:

- `delete 1`
  `undo` (reverses the `delete 1` command)
  `redo` (reapplies the `delete 1` command)
- `delete 1`
  `redo`
  The `redo` command fails as there are no `undo` commands executed previously.
- `delete 1`
  `clear`
  `undo` (reverses the `clear` command)
  `undo` (reverses the `delete 1` command)
  `redo` (reapplies the `delete 1` command)
  `redo` (reapplies the `clear` command)

3.9. Clearing all entries : `clear`

Clears all entries from the specific list requested by user.
Format: `clear`

3.10. Exiting the program : `exit`

Exits the program.
Format: `exit`

3.11. Saving the data

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

3.12. Attendance taking `[coming in v2.0]`

Track attendance of the volunteers and award frequent volunteers with certificates or promote to team leader.

3.13. Manage funding and sponsorships `[coming in v2.0]`

Manage funds and sponsors for individual projects and track project spending.

3.14. Auto-completion of command `[coming in v2.0]`

Quick Auto-completion of command to enhance typing speed

4. 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 VolunCHeer application folder.

5. Command Summary

- **AddProject** `addProject n/PROJECT_TITLE d/DATE b/BENEFICIARY [t/TAG]…`
  e.g. `addProject n/Charity Run d/081219 b/Sunshine Old Folks Home`
- **AddVolunteer** `addVolunteer n/NAME y/AGE a/ADDRESS e/EMAIL p/PHONE_NUMBER g/EMERGENCY_CONTACT r/RACE d/DIETARY_PREFERENCE m/MEDICAL CONDITION [t/TAG]…`
  e.g. `addVolunteer n/John Doe y/18 a/John street, block 123, #01-01 e/johnd@example.com p/98765432 g/98292998 r/chinese d/vegetarian m/asthma`
- **AddBeneficiary** `addBeneficiary n/NAME a/ADDRESS e/EMAIL p/PHONE_NUMBER`
  e.g. `addBeneficiary n/Orphanage p/98765432 e/Orphanage@example.com a/311, Clementi Ave 2, #02-25`
- **EditBeneficiary** `editBeneficiary INDEX (must be a positive integer) [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS]`
  e.g. `editBeneficiary 1 n/Old Folk Home p/91234567`
- **DeleteBeneficiary** `deleteBeneficiary i/INDEX -D` e.g. `deleteBeneficiary i/1 -D`
- **ListBeneficiary** `listBeneficiary`
- **FindBeneficiary** `findBeneficiary KEYWORD` e.g. `findBeneficiary Old`
- **SummariseBeneficiary** `summariseBeneficiary`
- **List** : `list`
- **EditProject** `editProject PROJECT_NAME [n/NAME] [d/DATE] [b/BENEFICIARY] [t/TAG]…`
  e.g. `editProject Charity Run d/010319`
- **EditVolunteer** `edit INDEX [n/NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [t/TAG]…`
  e.g. `editVolunteer 1 p/91234567 e/johndoe@example.com`
- **Find** : `find KEYWORD [MORE_KEYWORDS]`
  e.g. `find James Jake`
- **DeleteProject** : `delete PROJECT_TITLE` e.g. `delete Charity Run`
- **DeleteVolunteer** : `delete INDEX`
  e.g. `delete 3`
- **Select** : `select INDEX`
  e.g.`select 2`
- **Map** `map t/SELECTION t/SELECTION t/SELECTION`
  e.g. `map y/18 > r/chinese m/NIL`
- **Sort** `sort`
- **Extract** `extract VOLUNTEERS_REQUIRED+` e.g. `extract 20`

- **History** : `history`
- **Undo** : `undo`
- **Redo** : `redo`
- **Clear** : `clear`
- **Export** : `export`
- **Import** : `import`
- **Exit** * : `exit`
- **Help** : `help`