Ng Jin - Project Portfolio

Project: AddressBook - Level 4

AddressBook - Level 4 is a desktop address book application used for teaching Software Engineering principles. The user interacts with it using a CLI, and it has a GUI created with JavaFX. It is written in Java, and has about 6 kLoC.

Code contributed: [https://github.com/CS2103AUG2017-F10-B3/main/blob/master/collated/main/Labradorites.md] [https://github.com/CS2103AUG2017-F10-B3/main/blob/master/collated/test/Labradorites.md] {give links to collated code files}

Enhancement Added: Search for persons whose phone numbers contains searched keywords

External behavior

Start of Extract [from: User Guide]

Locating persons by phone: `findPhone`

Finds persons whose phones contain any of the given keywords.
Format: `findPhone KEYWORD [MORE_KEYWORDS]`

- Only the phone number is searched.
- Numbers containing keywords will be matched e.g. `111` will match `911100`
- Persons matching at least one keyword will be returned (i.e. `OR` search). e.g. `222 333` will return `0222`, `9333`

Examples:

- `findPhone 000`
  Returns `10001` and `191000`
- `find 111 222 333`
  Returns any person having phone numbers `1111`, `0222`, or `9333`

End of Extract

Justification

Allowing the user to search for his contacts by phone introduces more flexibility and user-friendliness of AddressBook. User's friend may have received a call from a phone number that is unsaved. Hence with findPhone, user is able to help check if the number happens to be in his AddressBook.

Implementation

Start of Extract [from: Developer Guide]

findPhone Command mechanism

The findPhone mechanism is able to search for persons whose phone numbers contain any number being searched.

Finds persons whose phones contain any of the given keywords.
Format: `findPhone KEYWORD [MORE_KEYWORDS]`

As long as contact has phone number that contains KEYWORD [MORE_KEYWORDS], it will be matched and displayed.

Java files that have been modified:
- `PhoneContainsKeywordsPredicate.java`
- `AddressBookParser.java`
- `FindPhoneCommand`
- `FindPhoneCommandParser`
- Some test units have been updated.

End of Extract

Enhancement Added: Search for persons whose tags exactly matches searched keywords

External behavior

Start of Extract [from: User Guide]

Locating persons by phone: `findTag`

Finds persons whose tags contain any of the given keywords.
Format: `findTag KEYWORD [MORE_KEYWORDS]`

- Only the tag is searched.
- Tags containing keywords will be matched e.g. `friends` will match `friends`
- Persons matching at least one keyword will be returned (i.e. `OR` search). e.g. `friends family` will return `friends`

Examples:

- `findTag friends`
  Returns `friends`
- `findTag friends family neighbours`
  Returns any person having tags `friends`, `family`, or `neighbour`

End of Extract

Justification

Allowing the user to search for his contacts by tags introduces more flexibility and user-friendliness of AddressBook. User is able to filter and see his a group of his contact that are of the same tag which he can then use for relating people or to mass send messages by groups. Hence with findTag, user is able to help check if the number happens to be in his AddressBook.

Implementation

Start of Extract [from: Developer Guide]

findTag Command mechanism

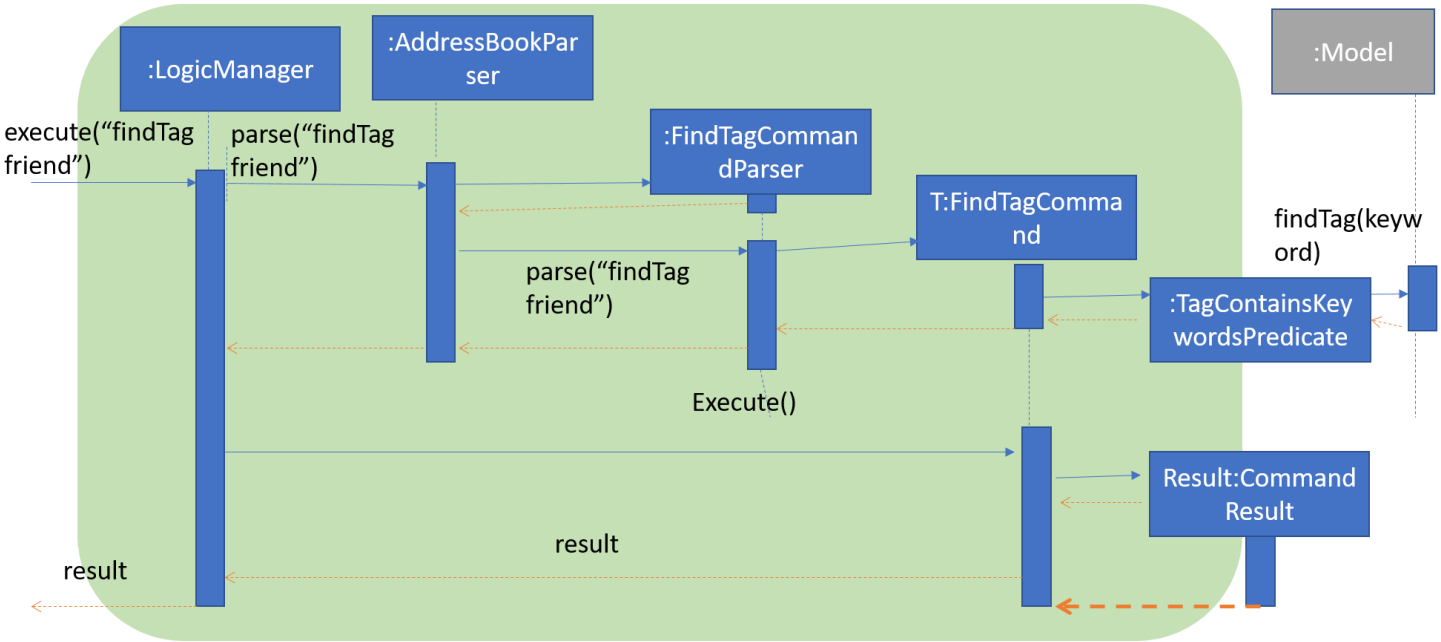
The findTag mechanism is able to search for persons whose tags contain any keywords being searched.

Finds persons whose tags contain any of the given keywords.
Format: `findTag KEYWORD [MORE_KEYWORDS]`

As long as contact has tags that contains KEYWORD [MORE_KEYWORDS], it will be matched and displayed.

Java files that have been modified:

- `TagContainsKeywordsPredicate.java` : Newly created.
- `AddressBookParser.java` : Edited to include findTag command.
- `FindTagCommand.java` : Newly created.
- `FindTagCommandParser.java` : Newly created.
- `Person.java` : Altered to get tags as a string.
- `ReadOnlyPerson.java` : Altered to get tags as a string.



End of Extract

Enhancement Added: List all tags added to AddressBook and all tags of most recent listing

External behavior

Start of Extract [from: User Guide]

Listing all tags : `listTags`

Shows a list of unique all tags in the AddressBook or persons currently displayed in PersonPanelList.

Tags listed are unique (no duplicates) and sorted alphabetically.

Format: `listTags [f]`

- `f`, is optional, allows users to choose between displaying all unique tags added to AddressBook or unique tags of persons currently listed in the PersonPanelList.
- `listTags` displays list of all unique tags added to AddressBook.
- `listTags f` displays list of all unique tags of persons listed in the PersonListPanel. `f` referring to "filtered". *

End of Extract

Justification

Allowing the user to list tags in 2 ways introduces more flexibility and user-friendliness of AddressBook. With listTags, user can see all tags that he has added to AddressBook and can refer to his old tags if he wants to reuse the same name. With listTags f, user is able to filter and see a range of tags which he can then use for relating people. Hence with listTags, user is able to help check if the number happens to be in his AddressBook.

Implementation

Start of Extract [from: Developer Guide]

The listTags implementation

`listTags` is able to seive out all unique tags that can be found in the AddressBook. ` f` can be appended to list unique tags of persons currently listed in the PersonListPanel. `listTags` does not display duplicate tags that are tagged to the contacts in the AddressBook and is sorted alphabetically. `listTags f` displays a list of unique tags of persons currently shown in the PersonListPanel.

Java files that have been modified:
- `Model` : Modified to include a signature in the Model interface for the getFilteredTagsList(), getNormalTagsList() method. Takes in no parameter and returns a List<Tag>. getTagsListAsString() is added to retrieve the list of tags as String.
- `ModelManager` : Modified to include the getFilteredTagsList() and getNormalTagsList() methods that takes in no parameter and returns a List<Tag> that contains all tags. getTagsListAsString() takes in a List<Tag> and returns List<String>.
- `ListTagsCommand` : Created to convert the List<Tag> to strings by calling the above functions and print the list of tags in the ResultsDisplay box.
- `Tag` : modified to include an accessor for tagName.

End of Extract

Enhancement Added: Search for person's address according to searched index

External behavior

Start of Extract [from: User Guide]

Searching person's address on Google Maps `gAddress`

Searches person's address on Google Maps with respect to `INDEX` keyed in.
Format: `gAddress INDEX`

> - Only the address is searched on Google Maps according to index keyed in.
> - The index refers to the index number shown in the most recent listing.
> - The index **must be a positive integer** 1, 2, 3 …

- `gAddress 1`
  BrowserPanel now shows searched address of person at index 1 in Google Maps.
- `gAddress 3`
  BrowserPanel now shows searched address of person at index 3 in Google Maps

End of Extract

Justification

Allowing users to search for his friends' address on Google Maps at his fingertips makes visiting friends convenient. User may want to visit his friend and want to know the surroundings of his friend's neighbourhood so that he does not get lost. Being able to check his friend's address on Google Maps at ease makes it easier for him to travel to his friend's house.

Implementation

Start of Extract [from: Developer Guide]

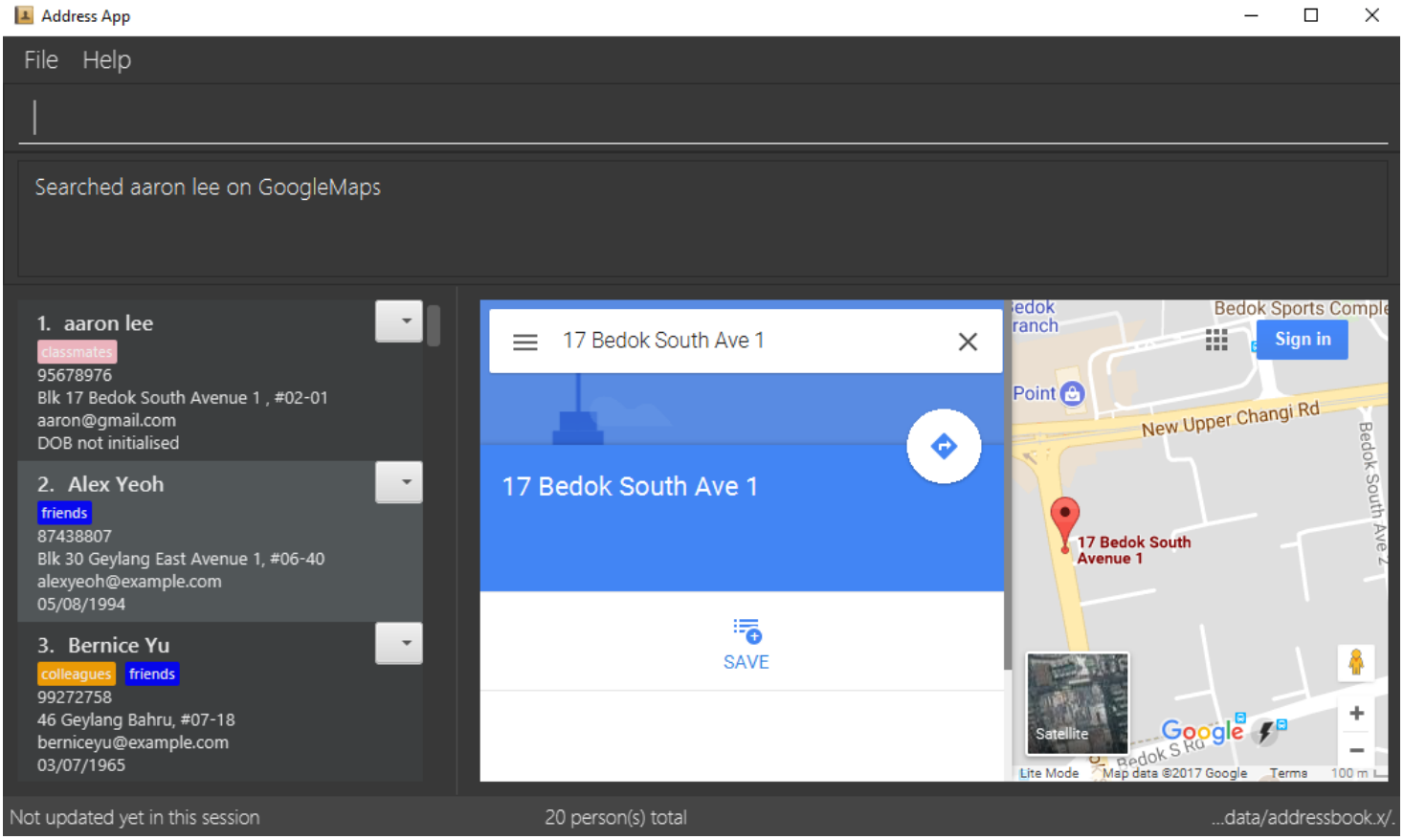
Search for address on Google Maps

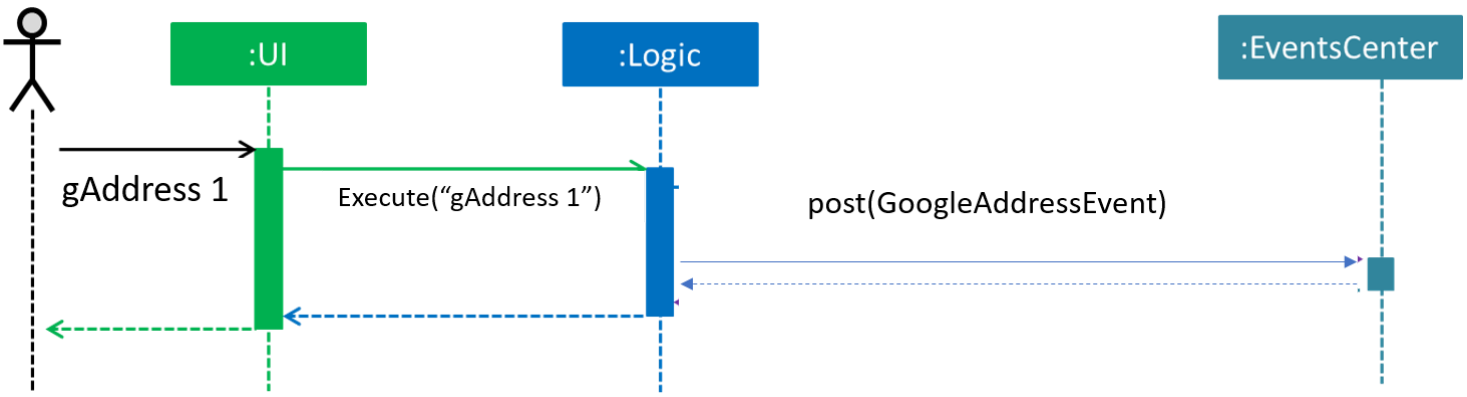
An improvement from searching address on Google Maps by selecting panel; converted to CLI form.

Searches for person's address on Google Maps by index in most recent listing.

Format: `gAddress INDEX`

Person whose index is keyed in will have its address searched in the browser panel on Google Maps.





End of Extract

Last updated 2017-11-14 02:56:49 +08:00