Chng Zhi Xuan - Project Portfolio

Project: Tourist-Book

Code contributed: [Functional code (https://github.com/CS2103AUG2017-F09-B2/main/blob/master/collated/test/Chng-Zhi-Xuan.md)] [Test code (https://github.com/CS2103AUG2017-F09-B2/main/blob/master/collated/test/Chng-Zhi-Xuan.md)]

Enhancement Added: Bookmarking

External behavior

Start of Extract [from: User Guide]

Bookmark a place: `bookmark`

Adds a bookmark tag to the specified place from the address book.
Format: `bookmark INDEX`

- Bookmarks the place at the specified `INDEX`.
- The index refers to the index number shown in the most recent listing.
- The index **must be a positive integer** 1, 2, 3, …

Examples:

- `list`
  `bookmark 1`
  Bookmarks the first place in the address book.
- `find n/attractions`
  `bookmark 3`
  Bookmarks the 3rd place in the results of the `find` command.

Display bookmarks: `show_bookmark`

List all bookmarked places in Tourist Book

Format: `show_bookmark`

Clearing all bookmarks: `clear_bookmark`

Clears all bookmarks from the address book.

Format: `clear_bookmark`

End of Extract

Justification

Bookmarking was a key feature part of the user story, as the tourist would like to track favourite places. Implementation was done via tag manipulation since the Tag class was well suited to support this feature. However, tag manipulation was cumbersome in the old addressbook, only from the edit command which was very slow.

My solution has seperate commands to speed up the process of adding/removing a specific tag ("Bookmarked") from each place, by passing the need to use the slower edit command.

(Pull requests #13 (https://github.com/CS2103AUG2017-F09-B2/main/pull/13), #93 (https://github.com/CS2103AUG2017-F09-B2/main/pull/93))

Implementation

Start of Extract [from: Developer Guide]

Bookmark & Clear Bookmark mechanism

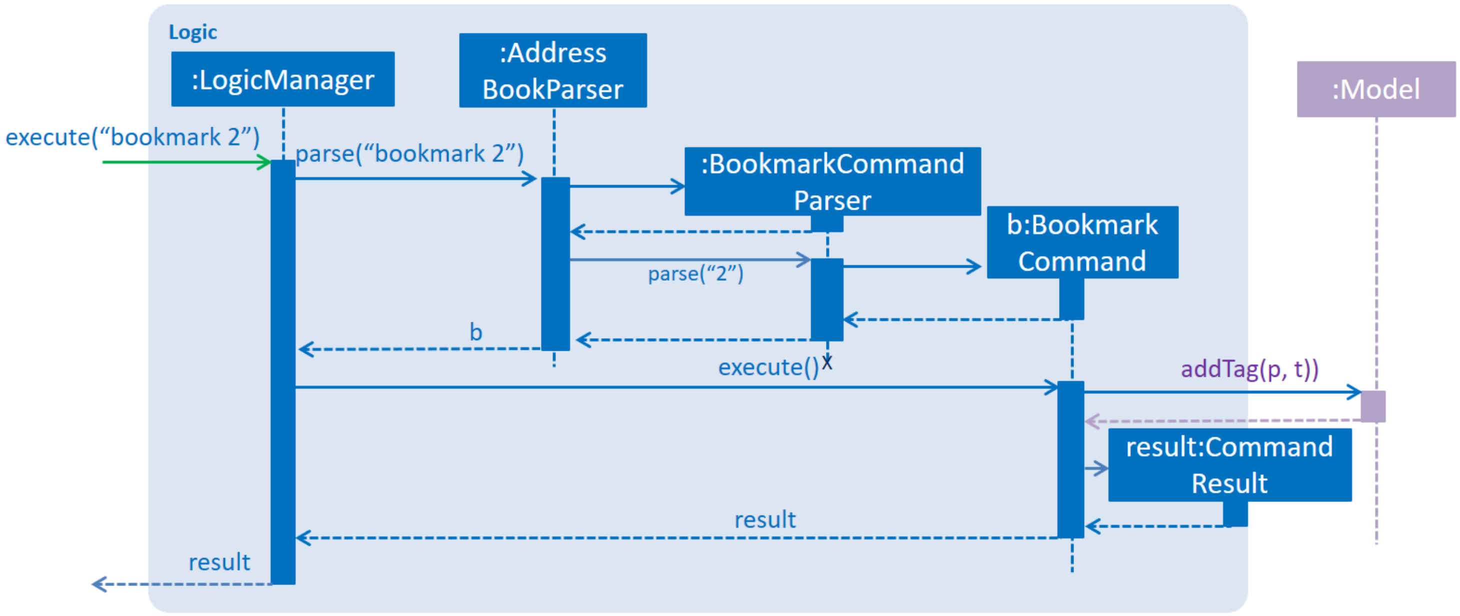
The underlying implementation of `Bookmark` and `Clear Bookmark` is tag manipulation in the model component. Two methods were added to `Model` interface to support this feature (`addTag` and `removeAllTags`).

Both `Bookmark` and `Clear Bookmark` extends from `UndoableCommand` as shown in **Figure 2.3.2** , which allows Undo/Redo by the user. Also, `Bookmark` is an index based command like (`delete` or `edit`)

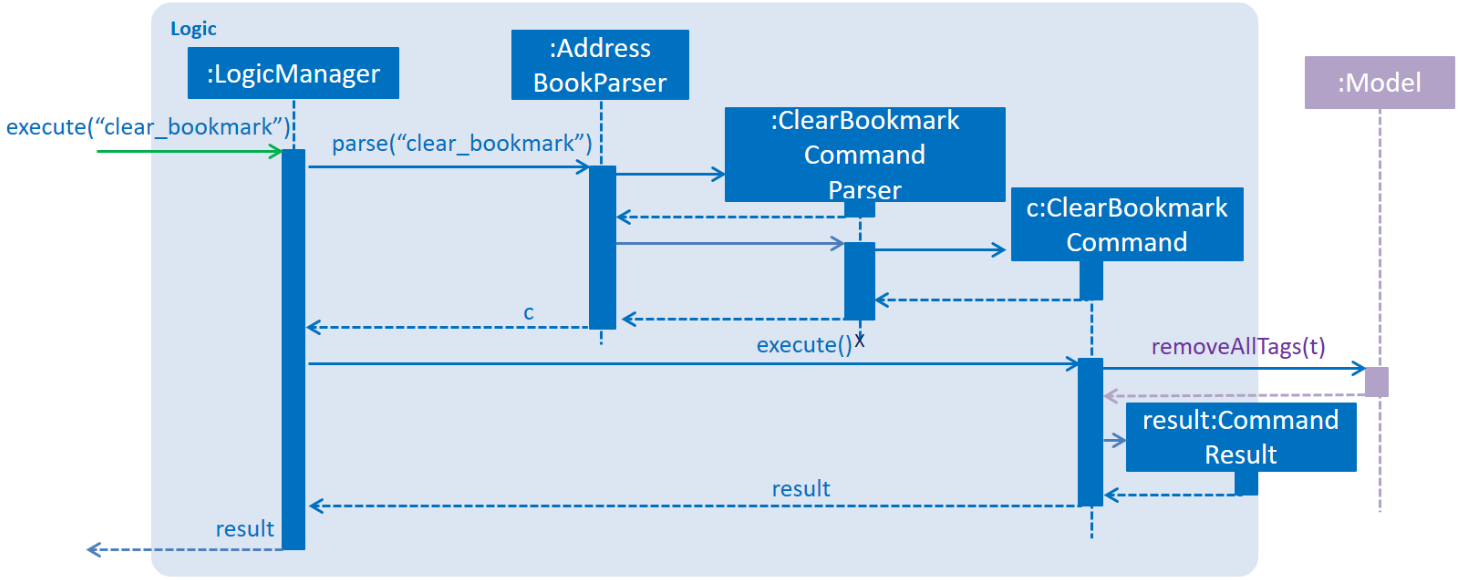
The following are the sequence diagrams each command

`bookmark` command

> Bookmarking a place which is already bookmarked will throw `DuplicateTagException` since `addTag` can be used by other commands or methods.



`clear_bookmark` command



Both commands are quite similar, only slightly difference is additional index argument for `bookmark` , and calling different methods in `ModelManager` .

Design Considerations

Aspect: Implementation of `bookmark` / `clear_bookmark`

Alternative 1 (current choice) `Tag` manipulation through methods in `ModelManager`

Pros: Easy to implement over existing `Tag` class.

Cons: Tags are randomly ordered, bookmark tag position is not consistent in the place list.

Alternative 2: Create a new class and have it as a new attribute in `Place`
Pros: Able to have fixed positioning of bookmark when listing places.
Cons: Tedious to implement, testing has added complexity.

Aspect: User experience for `bookmark` / `clear_bookmark`
Alternative 1 (current choice) Have 2 dedicated commands for manipulation on single specific tag
Pros: Standardization of "bookmark" tag, user cannot vary the tag for the same purpose E.G ("favourite", "saved", "like").
Cons: Only 1 bookmark "list" for user to put their favourite places.
Alternative 2: A more general command for user to change tags of specific place
Pros: More concise version of `edit` , allows user to have different bookmark "lists" E.G ("Near_West", "Near_East").
Cons: A general version of `clear_bookmark` will be `remove_tag` , which would allow the user to delete important pre-allocated tags in the initial list.

End of Extract

Enhancement Added: Direction from place to place

External behavior

Start of Extract [from: User Guide]

Get directions to selected destination

Displays the directions from location A to selected destination.
Format: `dir [INDEX_FROM] [INDEX_TO]`

- Displays the location from `INDEX_FROM` to `INDEX_TO` . The index refers to the index number shown in the last place listing.
- Both fields must be provided.
- The index **must be a positive integer** >= 1 and within the range of the list shown

Examples:

- `dir 1 2`
  Returns the directions from the location at index 1 to index 2 of the places listing.

- `dir 4 1`
  Returns the directions from the location at index 4 to index 1 of the places listing.

End of Extract

Justification

Highlighted as a "nice to have" feature in the user story. The usage would be the tourist has a schedule to visit several places in a day. It is very likely the tourist would seek direction to get from one listed place to another listed place.

"dir" command is implemented as a shortcut to manually entering 2 places on Google maps. It uses the name and postal code data found in the Tourist-Book to help the tourist search for the direction in the browser panel.

(Pull request #62 (https://github.com/CS2103AUG2017-F09-B2/main/pull/62))

Implementation

Start of Extract [from: Developer Guide]

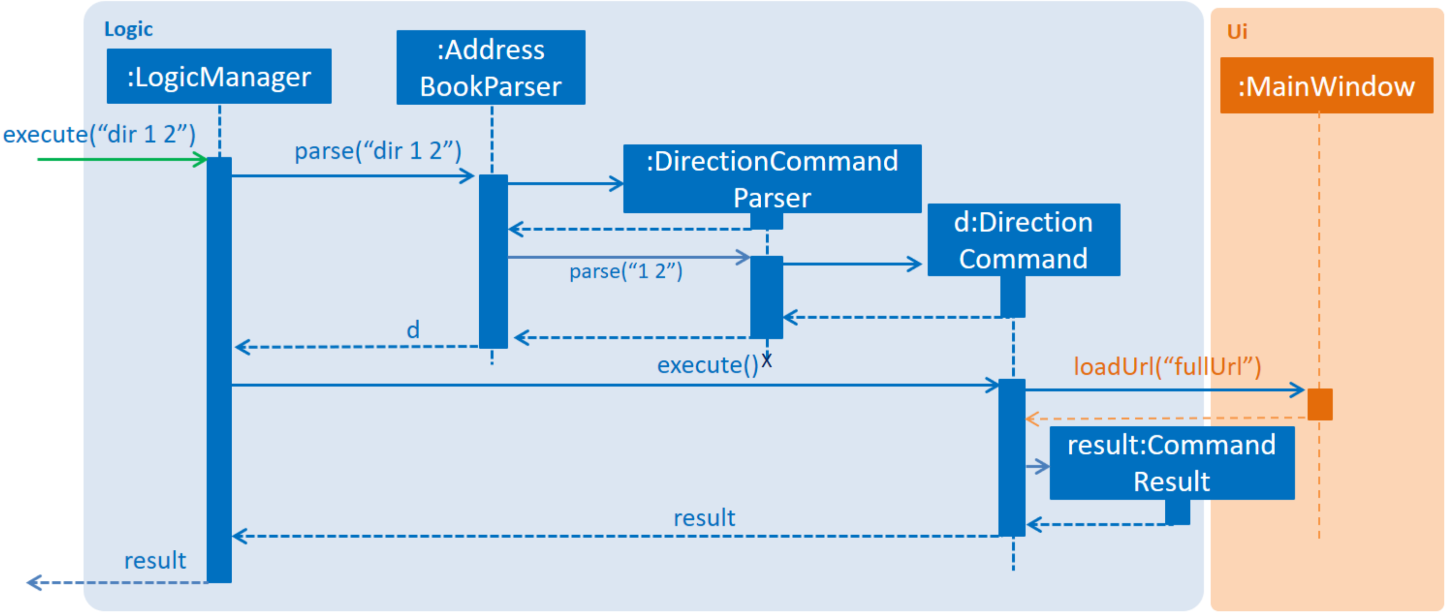
Direction Implementation

The underlying principle of direction is to retrieve 2 places from the index given by the user and convert each place’s name and postal code into a usable URL. The URL is then passed to the Ui component, within `MainWindow` to change the URL of the `BrowserPanel` .

`DirectionCommandParser` gets indexes with the help of `parseIndexFromPosition` in `ParserUtil` . It extracts the first 2 indexes from a given string using zero-based positions 0 and 1.

> ℹ Direction uses 1 based indexing from the user, index 0 is not used. Same index values will be rejected by the parser, and if either index are out of bounds, it will throw a `CommandExceptionError` during execute().

The following is the sequence diagram for `dir` command



Design Considerations

Aspect: Implementation of `dir` .

Alternative 1 (current choice) place's name and postal code transformation to URL.

Pros: Least amount of maintenance and takes advantage of browser panel to do more work.

Cons: Command can only work with an internet connection.

Alternative 2: Cache directions from one place to all other places within a attribute of a place.

Pros: Same directions given as google maps and works offline.

Cons: Huge amount of space required (complete graph) and tedious implementation through Google Maps API.

Aspect: Amount of places user can "chain" for getting directions.

Alternative 1 (current choice) Maximum of 2 places is used, with directions given between them.

Pros: No additional data structure needed to support this, less scope means quicker deployment and less prone to bugs.

Cons: User must use "dir 1 2" then "dir 2 3" to plan their schedule for places 1 2 3.

Alternative 2: Implement a queue structure.

Pros: User can type "dir 1 2 3" to store all the data needed. Use another command like "next_dir" to show directions instead of typing the `dir` command again.

Cons: Data structure implementation requires extensive test coverage, longer to deploy and needs a limit on how many places can it store.

End of Extract

Enhancement Proposed: Add place attribute : "Description"

Currently, there isn’t a short summary on what the various places offer, the tourist would need to select a place and visit the its website to find out more. However, the website might load slowly, and would require a few mouse clicks to find what is the place about.

To alleviate this issue, we can pre-load the description into a new place attribute<optional> and have it display on the place list.

Other contributions

- Refactor 1.9K LoC, from "person" to "places" (Pull request #44 (https://github.com/CS2103AUG2017-F09-B2/main/pull/44))
- Added user stories (Pull request #2 (https://github.com/CS2103AUG2017-F09-B2/main/pull/2))
- Tested for group T09-B1 (Issues raised #81 (https://github.com/CS2103AUG2017-T09-B1/main/issues/81), #82 (https://github.com/CS2103AUG2017-T09-B1/main/issues/82), #83 (https://github.com/CS2103AUG2017-T09-B1/main/issues/83), #84 (https://github.com/CS2103AUG2017-T09-B1/main/issues/84))

Last updated 2017-11-13 14:43:10 +08:00