Marlene Koh - Project Portfolio

PROJECT: NUSCouples

Overview

NUSCouples is a desktop application targeted at couples studying in the National University of Singapore (NUS). The user interacts with it using a CLI, and it has a GUI created with JavaFX. It is written in Java, and has about 10 kLoC.

Summary of contributions

- **Major enhancement**: added **a timetable field and the ability to view the NUSMods timetable of the person stored in NUSCouples**
  - What it does: allows the user to view the schedule of their partner with just one command/one click.
  - Justification: this feature allows the user to keep updated with their partner's school schedule.
- **Minor enhancement**: added a command to compare and display the common breaks in the NUSMods timetable of the person stored in NUSCouples along with one other given timetable.
- **Code contributed**: [Functional code] [Test code]
- **Other contributions**:
  - Project management:
    - Managed releases `v1.3` - `v1.5rc` (3 releases) on GitHub
  - Enhancements to existing features:
    - Add code to add a timetable field to person (Pull request #71)
    - Add code to parse timetable given an NUSMods URL (Pull request #76)
    - Reimplemented select and created deselect command to view and hide timetable respectively (Pull requests #127, #155)
    - Add command to compare timetables (Pull request #161)
    - Wrote tests (Pull requests #74, #178)
  - Community:
    - PRs reviewed (with non-trivial review comments): #148

Contributions to the User Guide

> *Given below are sections I contributed to the User Guide. They showcase my ability to write documentation targeting end-users.*

Timetable

Adding your partner's timetable [Since v1.3]

Refer to [Adding your partner : `add`]

Editing your partner's timetable [Since v1.3]

Refer to [Editing your partner : `edit`]

Viewing your partner's timetable : `tview`[Since v1.4]

Shows the current saved timetable of your partner.
Format: `tview`
Alias: `tv`

TIP	`Click` your partner's details in the list panel on the left to view your partner's timetable. `Ctrl` + `Click` your partner's details to go back to calendar view.

Comparing timetable : `tcompare` [Since v1.5]

Displays the common breaks shared by the given timetable and your partner's timetable in a timetable format.
Format: `tcompare tt/TIMETABLE_URL`
Alias: `tc tt/TIMETABLE_URL`

- The timetable url provided has to be a valid NUSMods short URL.
- Your partner must exist in *NUSCouples* before this command can be executed.

Examples:

- `tcompare tt/http://modsn.us/IO4n5`
- `tc tt/http://modsn.us/wNuIW`

Contributions to the Developer Guide

Given below are sections I contributed to the Developer Guide. They showcase my ability to write technical documentation and the technical depth of my contributions to the project.

Timetable View and Compare feature

Adding a Timetable

The Timetable Viewer feature is implemented by `Timetable`, which will reside in `ModelManager`.

Users are able to add a shortened NUSMods timetable URL to their existing partner in *NUSCouples*.

```
Sample shortened NUSMods URL: http://modsn.us/wNuIW
```

We pass the shortened URL through a `HttpURLConnection` to get the expanded URL.

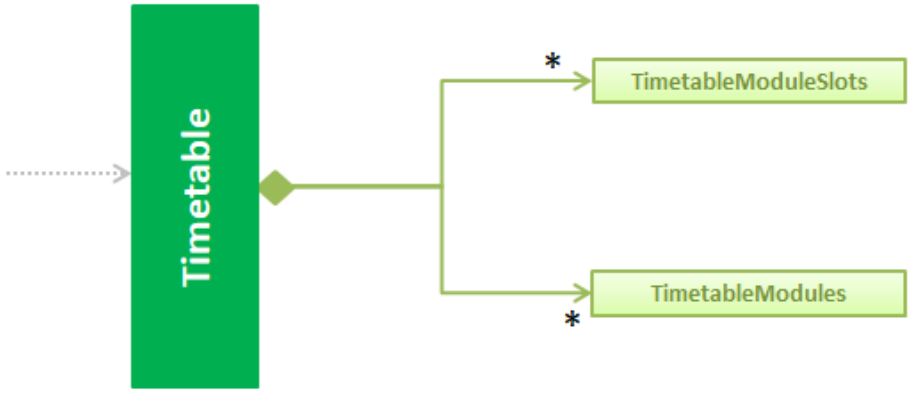
```
Sample expanded NUSMods URL: https://nusmods.com/timetable/sem-
2/share?CS2101=SEC:C01&CS2103T=TUT:C01&···
```

The expanded NUSMods URL can be generalised and represented in the format `···/timetable/sem-[SEM_NUM]/share?[MODULE_CODE]=[LESSON_TYPE]:[CLASS_NUM]&[MODULE_CODE]=[LESSON_TYPE]:[CLASS_NUM]&···`

We can parse this expanded NUSMods URL to get the `SEM_NUM`, as well as the `MODULE_CODE`, `LESSON_TYPE` and `CLASS_NUM` for each of the modules in the timetable.

Using NUSMods API, we can get the `WEEK_TEXT`, `DAY_TEXT`, `START_TIME`, `END_TIME` and `VENUE` of each module.

The following diagram shows how the Timetable class is represented.



A TimetableModule represents one NUSMods module.
The TimetableModuleSlots represents a particular class session of a TimetableModule. (e.g. Tutorial, Lecture, etc)

Design Considerations

Aspect: Implementation of add NUSMods timetable URL

- **Alternative 1 (current choice):** Accept short URLs only

- Pros: Easier to implement.
- Cons: Less user friendly as users can only add one type of URL.

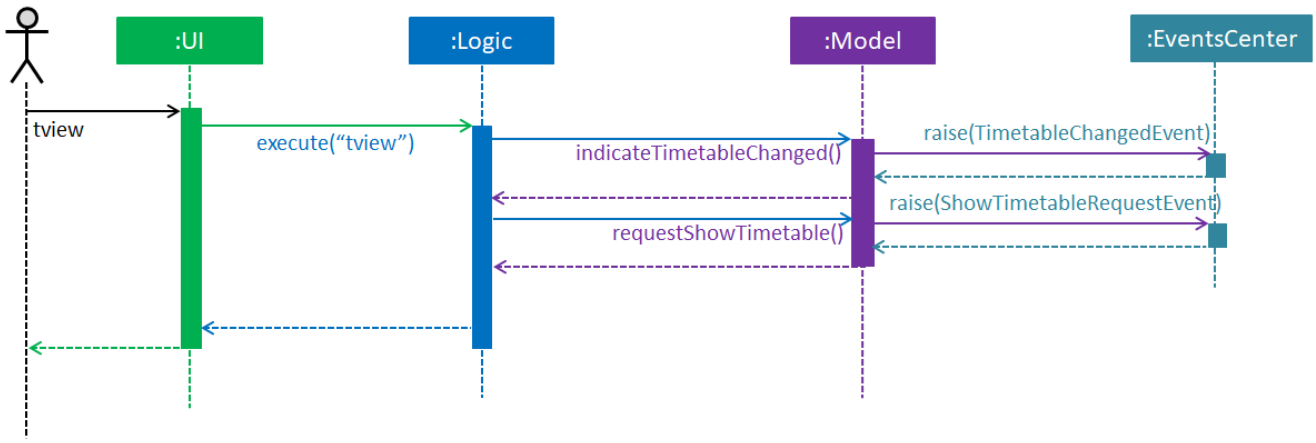
- **Alternative 2:** Accept both short URLs and expanded URLs
  - Pros: More user friendly as users have the choice to add either short or expanded URLs.
  - Cons: Difficult to check if given expanded NUSMods URL is a valid.

Aspect: Data Structure to support implementation of Timetable

- **Alternative 1 (current choice):** Store information by days of the week and by modules taken
  - Pros: Easy to add new functions on top of this implementation, more flexible.
  - Cons: May be a bit messy to implement due to the need to manage both structures.
- **Alternative 2:** Store information by days of the week
  - Pros: Easy to add new functions on top of this implementation such as compare timetables by days.
  - Cons: Have to sort information by day during parsing which can be tedious.
- **Alternative 3:** Store information by modules taken
  - Pros: Easier to implement due to how NUSMods API is structured.
  - Cons: Difficult to extract out information for a particular time slot on a particular day.

Viewing a Timetable

The following image shows how the tview Command works.



The `TimetableChangedEvent` is handled by `StorageManager` which will save the new timetable details into the relevant timetable display files.

```
    @Subscribe
    public void handleTimetableChangedEvent(TimetableChangedEvent event) {
        setUpTimetableDisplayFiles(event.timetable.getTimetableDisplayInfo());
        setUpTimetablePageHtmlFile();
        raise(new ShowTimetableRequestEvent());
    }
```

The `ShowTimetableRequestEvent` is handled by both `ListPanel` and `MainWindow`. The following code snippets show how they are handled.

```
    @Subscribe
    private void handleShowTimetableRequestEvent (ShowTimetableRequestEvent event) {
        logger.info(LogsCenter.getEventHandlingLogMessage(event));
        scrollTo(PARTNER_INDEX); // selects Partner in ListPanel
    }
```

```
    @Subscribe
    private void handleShowHelpEvent(ShowHelpRequestEvent event) {
        logger.info(LogsCenter.getEventHandlingLogMessage(event));
        handleHelp();
    }

    public void handleShowTimetable() {
        browserPanel.loadTimetablePage(); // Loads Timetable Page in Browser Panel
        if (!browserPlaceholder.getChildren().contains(browserPanel.getRoot())) {
            browserPlaceholder.getChildren().add(browserPanel.getRoot());
        }
    }
```

Design Considerations

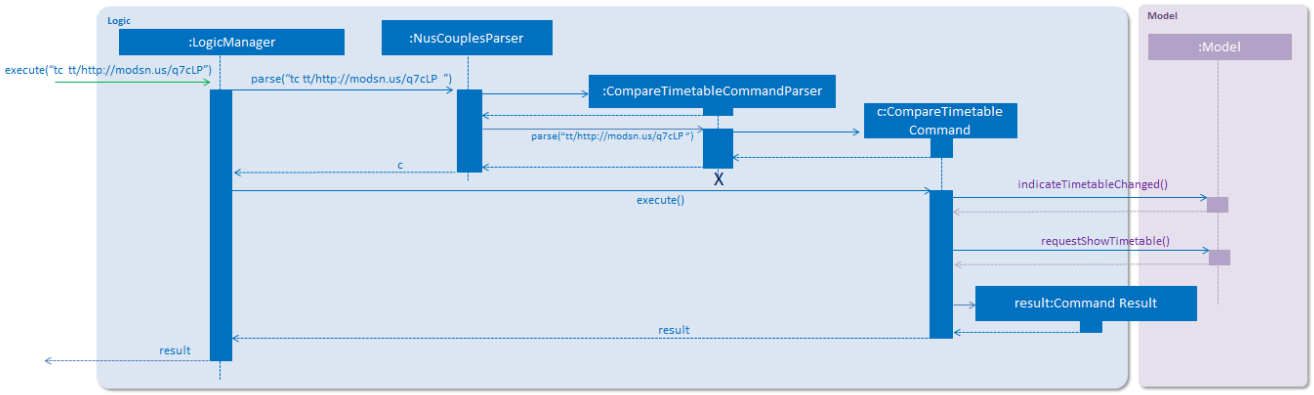
Aspect: Implementation of storing Timetable Information

- **Alternative 1 (current choice):** Stores Information in a HTML file. Edits the javascript array in the HTML file to change the contents of the tables.
    - Pros: Easy to implement.
    - Cons: GUI will be a static web page.
- **Alternative 2:** Use JavaFX
    - Pros: Provides a friendlier GUI (able to drag and drop table view).
    - Cons: Takes longer to load and display.

Comparing Timetables

The sequence diagram below shows interactions within the `Logic` Component for the `execute("tc`

tt/http://modsn.us/q7cLP") API call.



Similar to Viewing a Timetable, the CompareTimetableCommand raises two Events: ShowTimetableRequestEvent and TimetableChangedEvent. This updates the relevant files and refreshes the Timetable Page displayed.

We use a boolean isComparing in each TimetableModuleSlot to differentiate between the text to display for the normal Timetable and compared Timetable.

In *setUpTimetableInfoCompare()* in TimetableUtil:
We call a method *setUpUnsortedModuleSlotsForComparing()* that contains the following loop:

```
for (TimetableModuleSlot t : allUnsortedModulesSlots) {
    t.setComparing(true);
}
```

In TimetableModuleSlot:
We override the toString() method:

```
@Override
public String toString() {
    if (isComparing) {
        return isEmpty
                ? "\"\"" : "\"X\"";
    } else {
        return isEmpty
                ? "\"\"" : "\"" + moduleCode + "\"";
    }
}
```

The list of TimetableModuleSlots are formatted to look like an array, and is subsequently stored in a String which is stored in a file called timetableDisplayInfo.

FileTimetableStorage replaces the timetable array declared in TimetablePage.html, which replaces the contents in the table cells of the HTML file.

Design Considerations

Aspect: Implementation of displaying common breaks

- **Alternative 1 (current choice):** Replace all busy slots with 'X'
    - Pros: Easy to implement.
    - Cons: Not visually pleasing.
- **Alternative 2:** Shade the cells a different colour.
    - Pros: More intuitive for the user.
    - Cons: More tedious to implement.