Wei Boon - Project Portfolio

Overview

My team and I were tasked with enhancing AddressBook3 - a given CLI (Command Line Interface) application into a better product. Through the ideation phase, we decided to morph the application into the Njoy Teaching Assistant. We aim to improve the quality of life for teachers by assisting them with daily administrative tasks. In particular, the Njoy Teaching Assistant enables teachers to maintain student records to manage students better ; set questions and quizzes to enhance students learning; and keep track of their schedules with an interactive timetable.

This is what our project looks like:

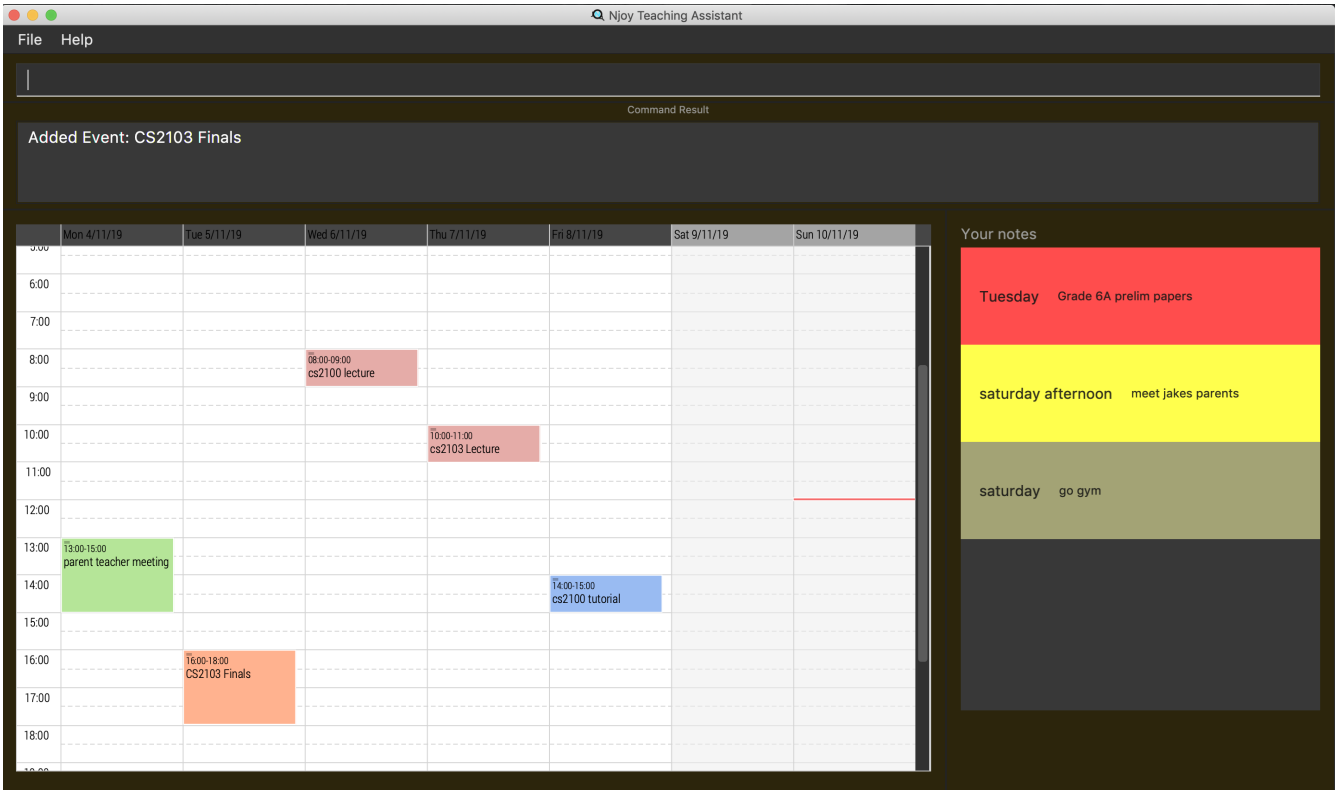


Figure 1. The graphical user interface for Njoy.

My role was to design and write code for the event feature. This includes commands for manipulating and viewing events such as `screenshot`, `export`, `view`, etc. The rest of the sections will cover the summary of my contributions to the codebase, the user guide and the developer guide.

The following are icons and symbols that I will be using for the Project Portfolio:

`This indicates a component, class or object in the architecture of the application.`

This indicates important text.

Summary of contributions

This section entails a summary of my specific enhancements, code contributions and other helpful increments towards the Njoy Teaching Assistant.

I implemented the commands related to the event feature, which include:

- **Adding a event to the event record**
- **Getting the index of a event from the event record**
- **Removing a event from the event record**
- **Editing a event in the event record**
- **Viewing the calendar in different modes**
- **Taking a screenshot of the calendar**
- **Exporting all events in the event record to a ics file**

I'll be sectioning the functionality into the following format (where applicable for the features):

1. **Creation**
2. **Deletion**
3. **Editing**
4. **Display**

Enhancements

Event

Event Creation and Deletion

- **What it does:** Users can create events and delete events to keep track of their schedule. These events are created by specifying the event name, start date time and end date time. These events also can be set to recur weekly or daily, reducing the hassle for repeated events. They can also have their own unique colors to allow users to easily identify and categorize their events. Users can also delete any events which they have mistakenly created, using the delete event command.
- **Justification:** Teachers usually have activities and events to attend to. These include delivering lectures, meeting with parents, having consultations, etc. I wanted to create a feature which allows teachers to easily manage their time, not just during the office hours, but also to schedule errands or other preparation work to be done outside working hours.
- **Highlights:** This enhancement works with existing as well as future commands. Some challenging portions include, ensuring there were no duplicate events created as well as implementing the recurring feature.

Event Editing

- **What it does:** Users can edit events that are currently in their calendar.
- **Justification:** There is a possibility that events have to be rescheduled in the calendar, and this functionality allows teachers to do exactly that. In fact, teachers are allowed to edit any of the fields in an event including its color and recurrence type.

Event Viewing

- **What it does:** Users can view their events in calendar form and set the target date of interest as well as the viewing mode of the calendar (weekly or daily).
- **Justification:** Teachers may want to view the calendar in different forms, for different purposes. They might want to have a over-arching view of the upcoming events of the week, thus they may wish to view their events in a weekly form. They may also wish to go into the details of a specific day to see exactly what they have planned for that day. Additionally, when making appointments with other people, they may need to go to that target date of interest to check their availability.
- **Highlights:** This enhancement works with existing as well as future commands. Notably, this feature also remembers the preferences of the user in the same instance of the application. For example, after switching to the student list and then switching back to the calendar, it remembers the target date of interest as well as the viewing mode (weekly or daily), and renders it accordingly.

Event Screenshot

- **What it does:** Users can take a screenshot of the calendar and save it to a png file.
- **Justification:** Teachers may want to share their weekly schedule or simply have a image of their events to be viewed on their mobile devices. This feature allows them to do exactly that by taking a screenshot of whichever date and time the user is currently viewing and saving it into a png file.
- **Highlights:** This enhancement works with existing as well as future commands. The implementation of this feature was rather challenging, as different users have different screen sizes, and taking a screenshot may not capture all the information desired by the user. As such, the screenshot feature, actually opens the calendar into full-screen mode before taking the screenshot. This is a "best-effort" approach to capture as much detail as possible within the confines of the user’s display.

Event Export

- **What it does:** Users can export all their events into a ics file.
- **Justification:** Teachers may want to view and manager their events in alternative applications such as Google Calendar or iCalendar. The ics file type is one of the most common and well recognised calendar file types, and thus allows users to easily import their NJoy events into other calendar applications.

- **Highlights:** This enhancement works with existing as well as future commands. The implementation of this feature was rather challenging, as every event in NJoy teaching assistant had to be converted into a ics-compliant format, which represents all its information including the recurrence type.

Code contributed

Please click these links to see a sample of my code.

Add Event

[Functional code] [Test code] *{Links to samples of my code.}*

Index Event

[Functional code] [Test code] *{Links to samples of my code.}*

Edit Event

[Functional code] [Test code] *{Links to samples of my code.}*

Delete Event

[Functional code] [Test code] *{Links to samples of my code.}*

Screenshot Event

[Functional code] [Test code] *{Links to samples of my code.}*

Export Event

[Functional code] [Test code] *{Links to samples of my code.}*

View Event

[Functional code] [Test code] *{Links to samples of my code.}*

Other contributions

- Project management:
  - Resolved the issues found by others related to the user interface and storage components of the project.
  - Resolved the issues found by others related to my feature on Github.
  - Updated the AboutUs page in GitHub for my team's repository.
  - Updated the ReadMe description.

- Enhancements to existing features:
  - Wrote additional tests for existing features to increase coverage from 51% to 60% : #188
- Documentation:
  - Made cosmetic tweaks to the existing contents of the User Guide: #176
  - Made relevant changes for the Developer Guide: #188

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. The following are specific portions of the NJoy Assistant's User Guide that I have selected. I'll only show one example of creation, editing and display for the features that I have created, as they are repetitive.

The following is an example of `Event` screenshot section the User Guide:

Taking screenshot of schedule

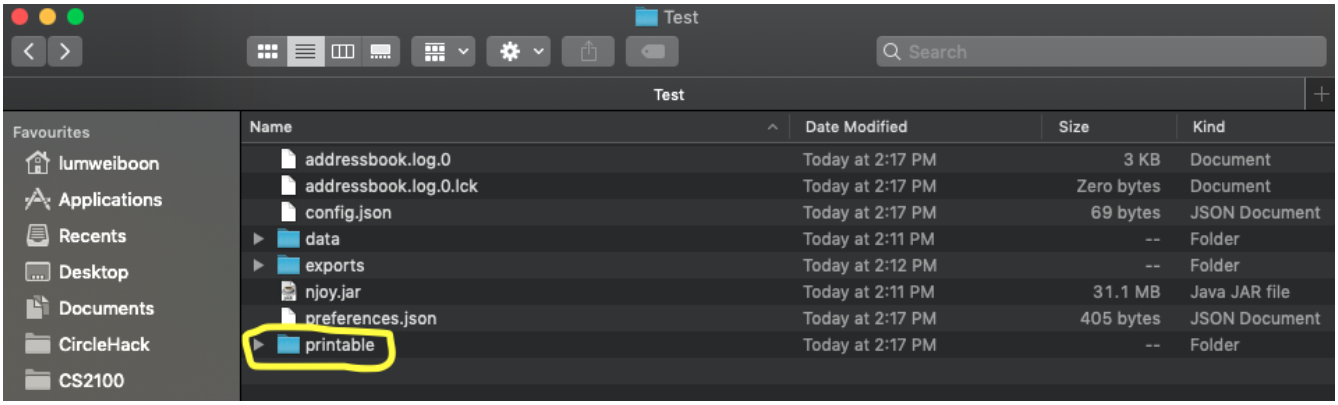
Takes a screen shot of the calendar as PNG file
Format: `event screenshot`

NOTE	Scroll to which portion of the calendar is to be taken, and the application will then open a separate full screen window to maximise the content captured of the screenshot.
NOTE	The event schedule screenshot is stored in the same place as where the JAR file is installed. The screenshot can be found under a newly created printable directory. If the directory exists beforehand, no new directory is created. See statistics print feature
NOTE	The screenshot will be saved based on the current settings of your event schedule. For example if you are viewing 2019-11-23 on a weekly mode, the filename will be saved as WEEKLY_2019-11-23.

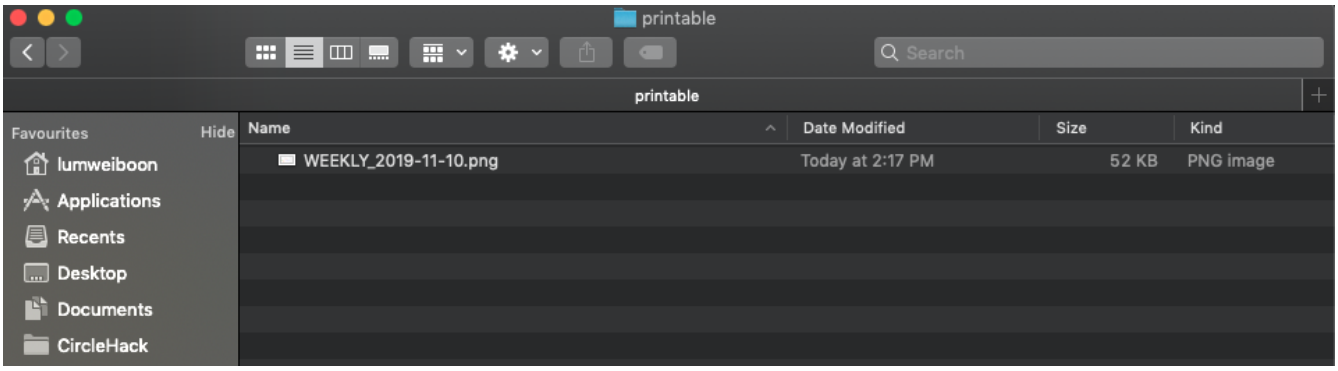
- Example: `event screenshot`
  Takes a screenshot of the current calendar and saves it into Printable folder.

The following example demonstrates how to find the screenshot:

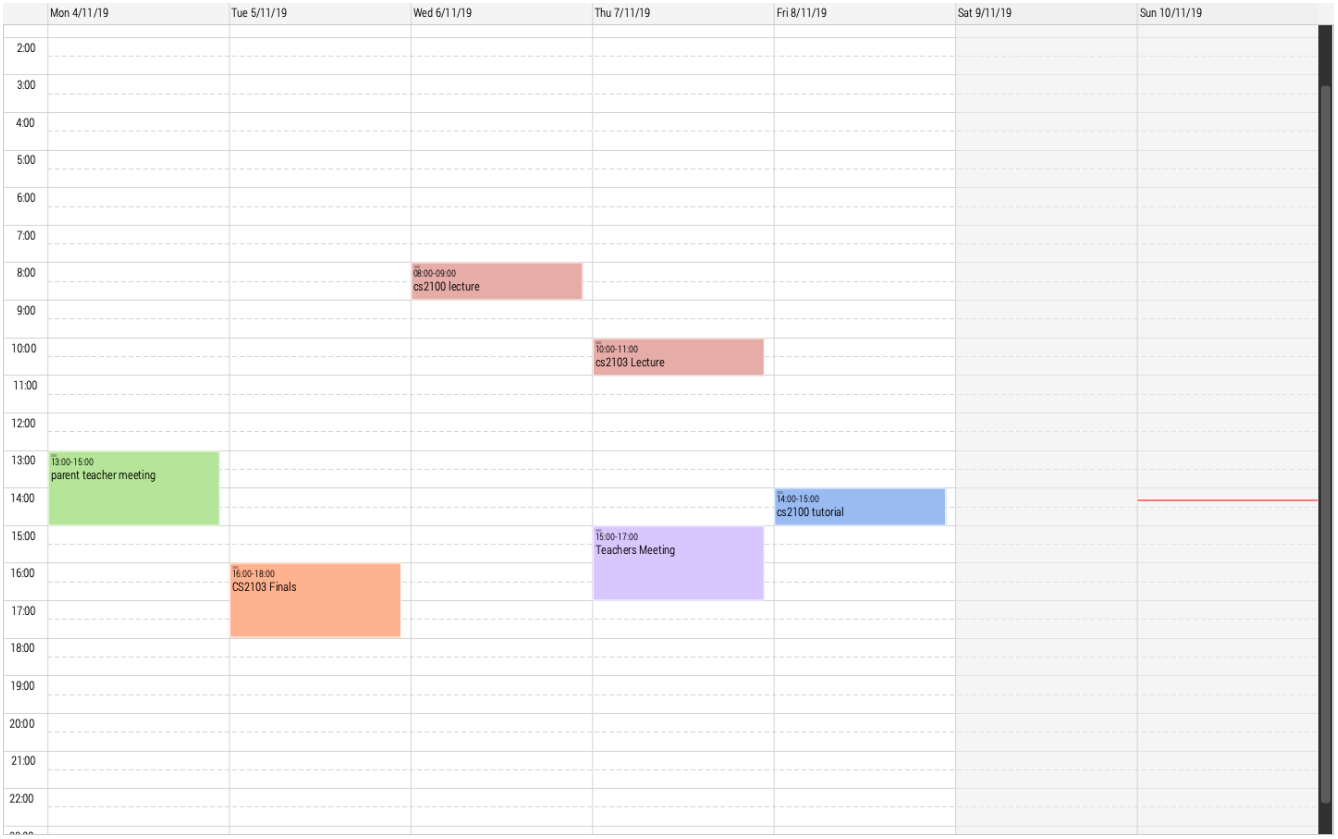
Step 1. After executing the event screenshot command, the user should navigate to their current directory. An printable folder as shown in the image below should be created!



Step 2. The png file should be named according to your schedule settings in the format VIEWMODE_TARGETDATE.png



Step 3. Now you can share your calendar screenshot with others (or yourself)!



The following is an example of `Event` view section the User Guide:

Viewing all events

Show all your events in the calendar. All fields are optional Format: `event view scheduleMode/… targetDate/…`

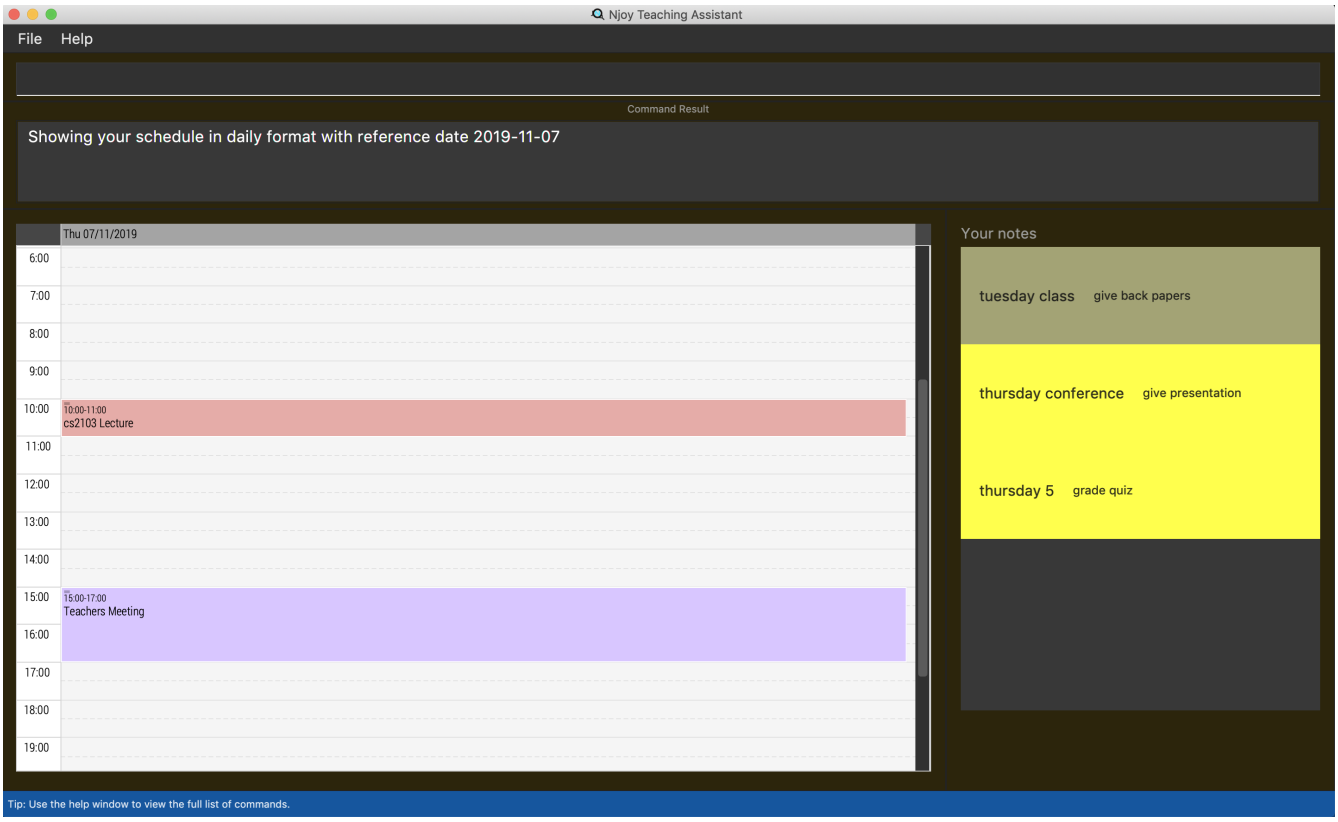
The options supported by this feature includes:

1. `scheduleMode` - Schedule viewing mode of the calendar. Either weekly or daily.
2. `targetDate` - The target date to show for the week. If in daily mode, simply show the events in the date. If in weekly mode, show the week which includes the specified date

NOTE	The target date option must be specified in the following format: yyyy-mm-dd. E.g. 2019-11-23

Example:

- `event view scheduleMode/daily targetDate/2019-11-07`
  Opens the calendar view in daily mode with the target date 7 November 2019.



Viewing the calendar in daily mode with target date 7 November 2019

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. Again, I'm only going to include the most relevant portions of the guide, especially the UML diagrams that I have created. Also, some of the command hyperlinks will obviously not work because I have omitted them for brevity.

The following is the `EventRecord` class overview section of the Developer Guide:

Class Overview

The figure below describes the interactions between event-related classes in the `Model`. Note how the `EventRecord` class has a dependency on `Event` object in its constructor, but only has a `VEvent` attribute: `vEvents`. This highlights a mapping between the Event and VEvent object within the `EventRecord` class. Althought the methods of the `EventRecord` class are omitted for brevity, they are mostly `VEvent` based, which again highlights that interactions with the `Logic` and `UI` layers will mostly be done in `VEvent` type.

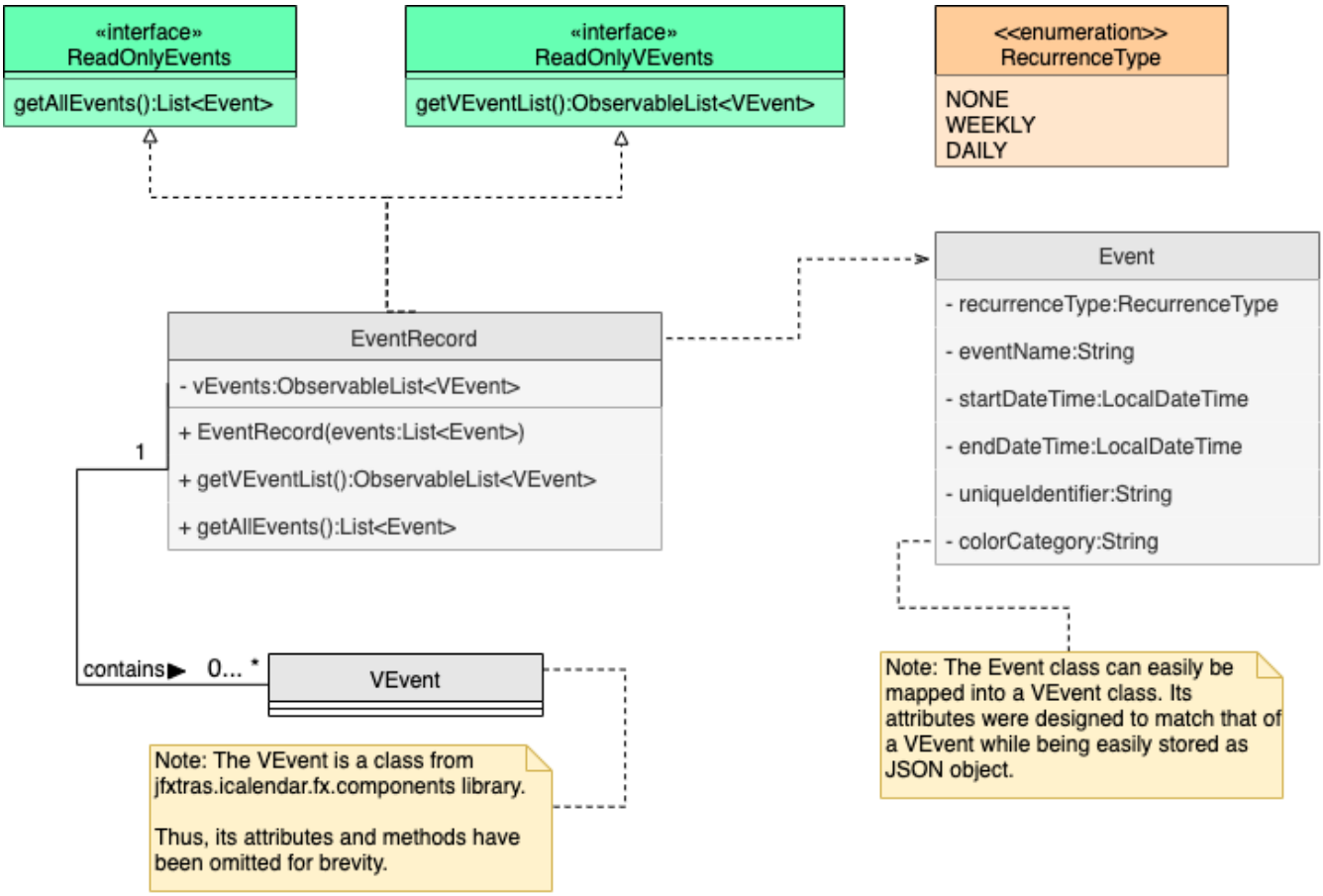


Figure 1. Class Diagram for EventRecord

The following is the delete `Event` section of the Developer Guide:

1. If successful, a success message will be generated by the and a new `CommandResult` will be returned with the generated success message. Otherwise, an error message showing proper note command syntax is thrown as `CommandException`.
2. If the command syntax was valid and `VEvent` was removed from the `EventRecord`, `LogicManager` calls `Storage#saveEventRecord(ReadOnlyEventRecord eventRecord)` which saves the new notes

record in JSON format after serializing it using the `JsonEventRecord`.

The following is a sample sequence diagram of the `EventDeleteCommand`. Other commands under the notes feature follow a similar program flow; their diagrams have been omitted for brevity.

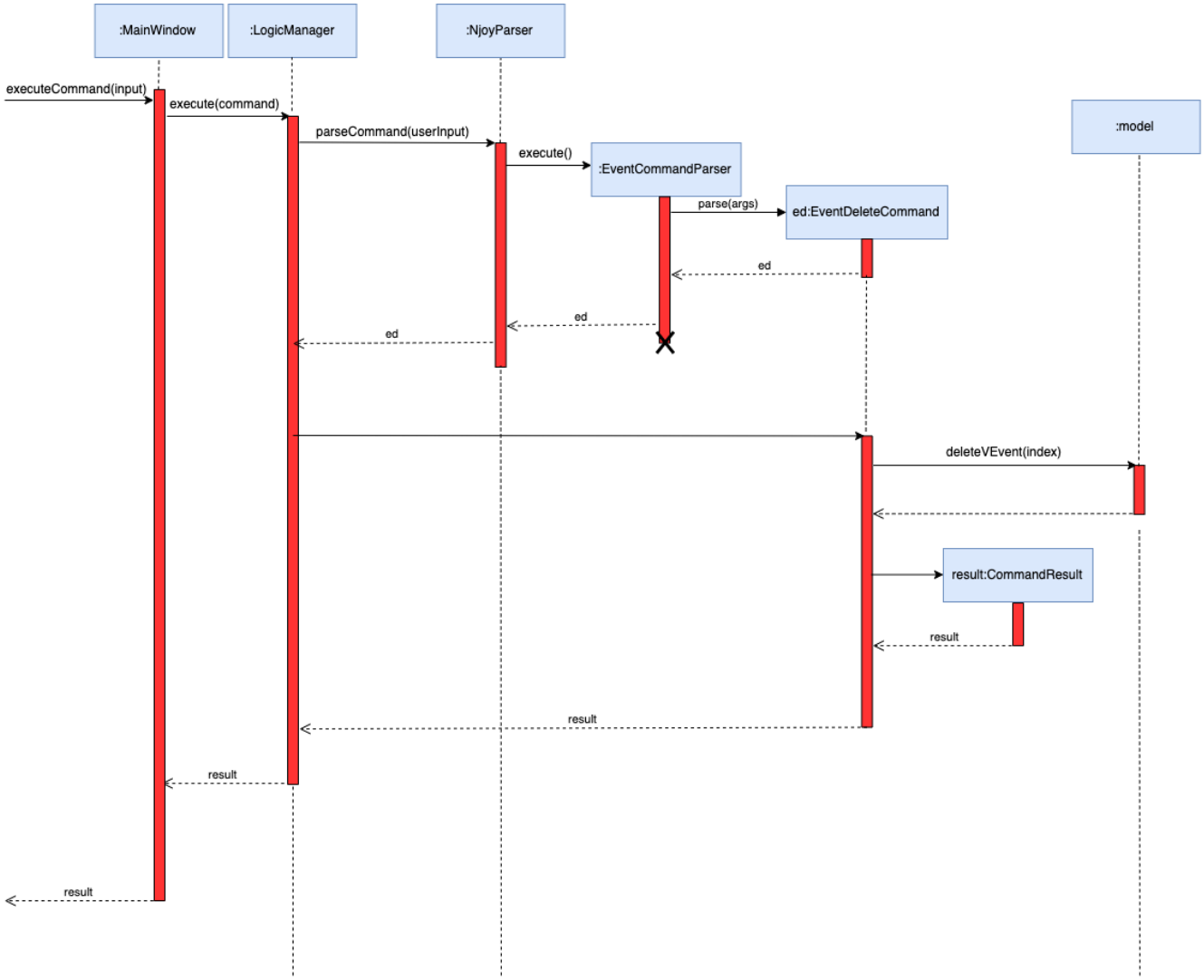


Figure 2. Sequence Diagram for Deleting Events

View event command

This command changes the mode which the `EventSchedulePanel` is in. The command has 2 optional parameters being `targetViewDateTime` and `eventScheduleViewMode`. The `EventScheduleViewMode` enum is used to represent the the skins of the iCalendarAgenda: weekly and daily. The `targetViewDateTime` sets the reference date time to be rendered in iCalendarAgenda, which will in turn show the corresponding week (if it is in week mode) which contains the reference date time. Otherwise, in daily mode, it will simply show the VEvents for that day.