Ko Gi Hun - Project Portfolio

PROJECT: Typee

Overview

Typee is an engagement management application, where users can manage engagments and schedule accordingly. Our main users are targeted in corporation's receptionists and secretaries whose main tasks is to manage and arrange engagements.

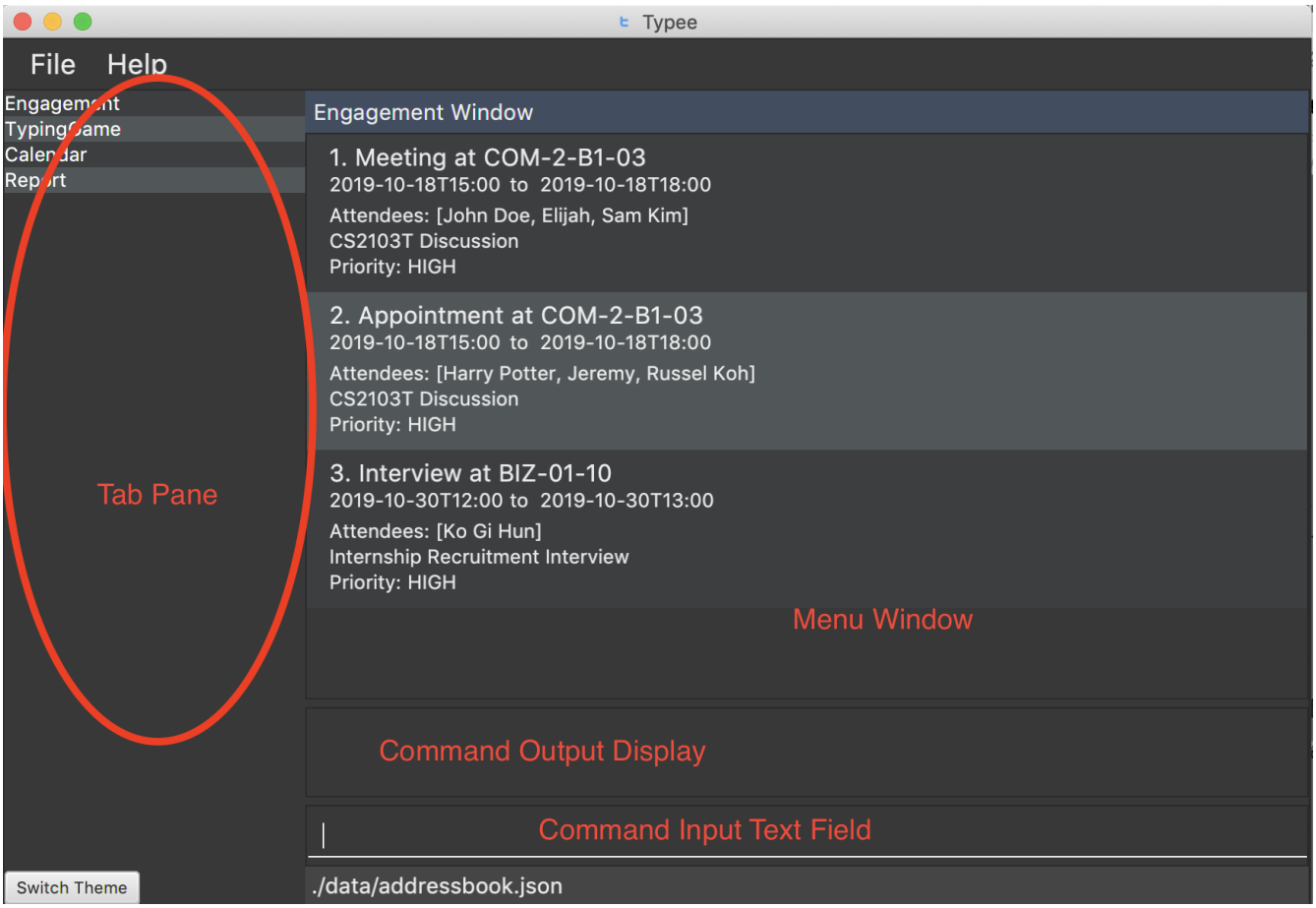


Figure 1. default window of Typee upon start-up

- Main UI is composed of 4 parts;
  - Tab/Menu Pane: List of Name/Feature/Menu is displayed here.
  - Menu Window: Each feature/menu window is displayed and can be interacted.
  - Command Output Display: Command output after execution displays here.
  - Command Input text field: User enters command
- Typee divides engagements into 3 types; Appointments, Meetings and Interviews.
- Typee application interacts with the users by separating application features into different menu windows. Each menu name is displayed in the left-end of the application window, where user can switch to different windows anytime by using command.

- Typee is a CLI (Command Line Interface) based application, with a mixture of simple GUI interacting features.

Summary of contributions

- **Major enhancement**:
  1. added **the ability to switch windows to other menu windows**
     - What it does: allows the user to switch the main window to load different menu windows to utilise different features any time.
     - Justification: In terms of user interface, by placing different feature interactions and forms in separate windows makes the application look more organised.
     - Highlights: New class `Tab` was implemented in the model and existing class `CommandResult` had to check if the command if the command type is `tab` command. This was a necessary implementation as the `MainWindow`, which is the root UI component of the main UI had to load external fxml file to the empty component during the command process. Structure of the MainWindow had to be modified appropriately so that each separate fxml files are linked with different JAVAFX controller classes other than `MainWindow` controller class.
     - Credits: SEDU team, where much of the architecture of the Main Window UI structure was referenced in order to customise our UI architecture to allow performing tab switching command.
  2. added **the ability to generate reports into a document form (pdf) based on the selected engagement**.
     - What it does: allows users to generate a report into pdf format document, where user can simply select the engagement that the user wants to generate into document and attach it to email to the corresponding receiver.
     - Justification: Secretaries and receptionists are not responsible only in task management. They have other tasks that consume alot of their time during office work. In order to improve their task performance more time efficiently, the feature allows users to save time instead of manually typing the document while referring to the engagement information one by one. With a simple 1 line command, typee generates a document, so that the user only need to attach the document their email. Document template is fully customisable, hence it makes the application flexible so that any corporations can use our application to fit their company document conventions formats.
     - Credits:
       - Commons-IO : file extension validation method is used for
       - iTextPdf
- **Minor enhancement**: Updated the existing AB3 CSS dark theme file to customise our Typee UI color schemes; TreeListView, Functional button skins (Delete button, Refresh button, Next/Previous month button)
- **Code contributed**: [Functional code] [Test code]
- **Other contributions**:

- Project management:
  - Managed releases `v1.3` - `v1.5rc` (3 releases) on GitHub
- Enhancements to existing features:
  - Updated the GUI color scheme (Pull requests #33, #34)
  - Wrote additional tests for existing features to increase coverage from 88% to 92% (Pull requests #36, #38)
- Documentation:
  - Did cosmetic tweaks to existing contents of the User Guide: #14
- Community:
  - PRs reviewed (with non-trivial review comments): #12, #32, #19, #42
  - Contributed to forum discussions (examples: 1, 2, 3, 4)
  - Reported bugs and suggestions for other teams in the class (examples: 1, 2, 3)
  - Some parts of the history feature I added was adopted by several other class mates (1, 2)
- Tools:
  - Integrated a third party library (Natty) to the project (#42)
  - Integrated a new Github plugin (CircleCI) to the team repo

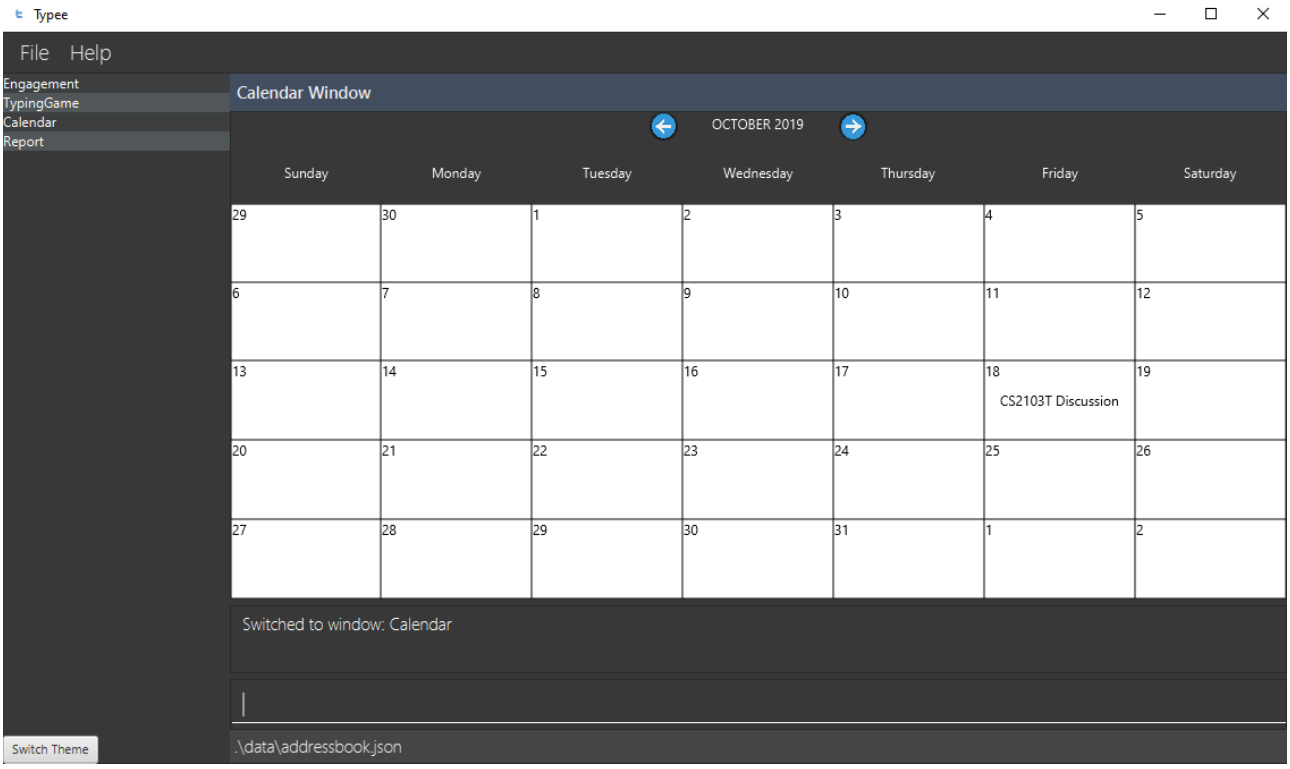
{you can add/remove categories in the list above}

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.

Quick Start

1. Ensure you have Java 11 or above installed in your Computer.
2. Download the latest `typee.jar`.
3. Copy the file to the folder that you want to use as the home folder for your Typee application.
4. Enter the command `java -jar typee.jar` on your terminal to start the app. The GUI should appear in a few seconds.
5. Type a command in the text box and press `Enter` to execute it.



1. Type the command in the command box and press kbd:[Enter] to execute it.
   e.g. typing **`help`** and pressing kbd:[Enter] will open the help window.
2. Some example commands you can try:
   - **`list`** : lists all appointments.
   - **`tab game`**: switches main window to start window of the game.
   - **`add d/Meeting on Monday`** : adds an appointment with the description `Meeting on Monday` to the appointment manager.
   - **`pdf i/1 to/Jason from/Harry`**: generates a document of the selected engagement of index 1, by setting the sender as `Harry` and receiver of the document as `Jason`.
   - **`delete 3`** : deletes the 3rd appointment shown in the current list.
   - **`exit`** : exits the app.

Refer to [Features] for details of each command.

Switch to different windows: `tab`

Switches to a different menu in the application window.
Format: `tab [menu_name]`

- Typee application has mainly 4 major features/menus. User needs to switch to respective windows to perform different commands and features of the system.
  - Engagement
  - TypingGame
  - Calendar
  - Report

NOTE `tab game` is used instead of `tab typinggame` to simplify typing.

- `Engagement` window will be the default window upon start up of the application.

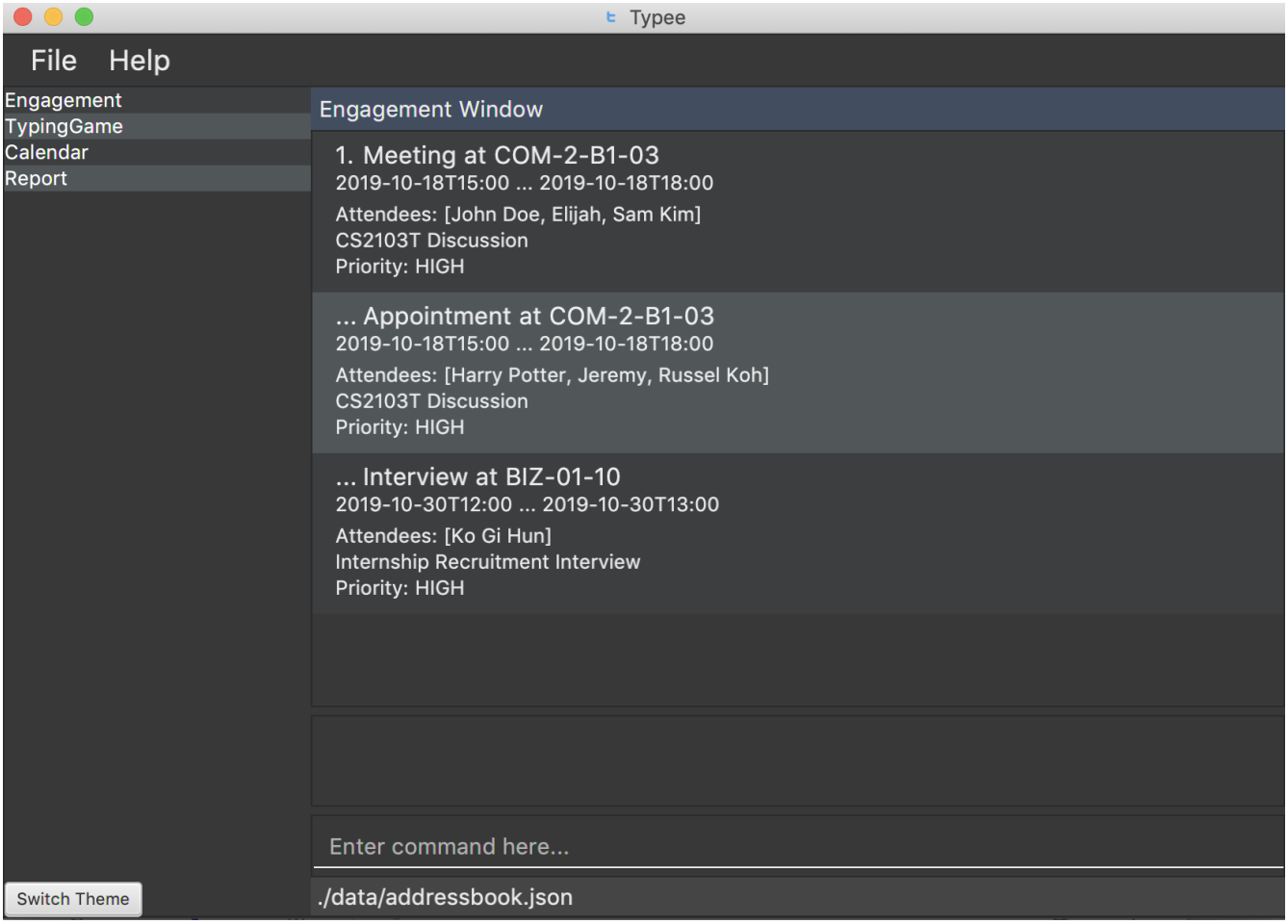


Figure 2. Engagement (default) window displayed on start-up of the application.

Now, if the user wants to switch to different windows, simply enter the tab command with refer to the menu name listed on the left end of the application window. For example, user enters `tab calendar` switch to calendar view window.

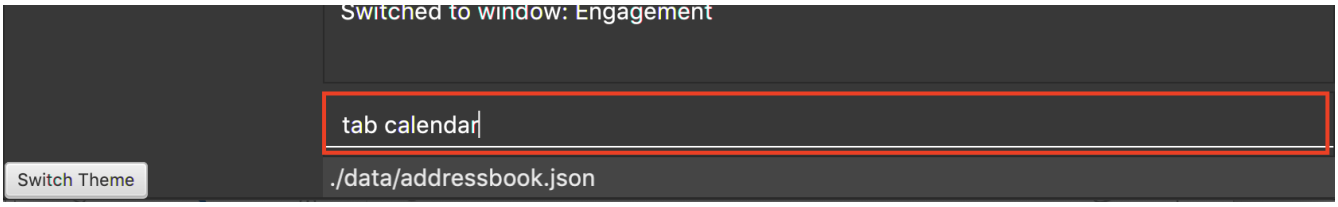
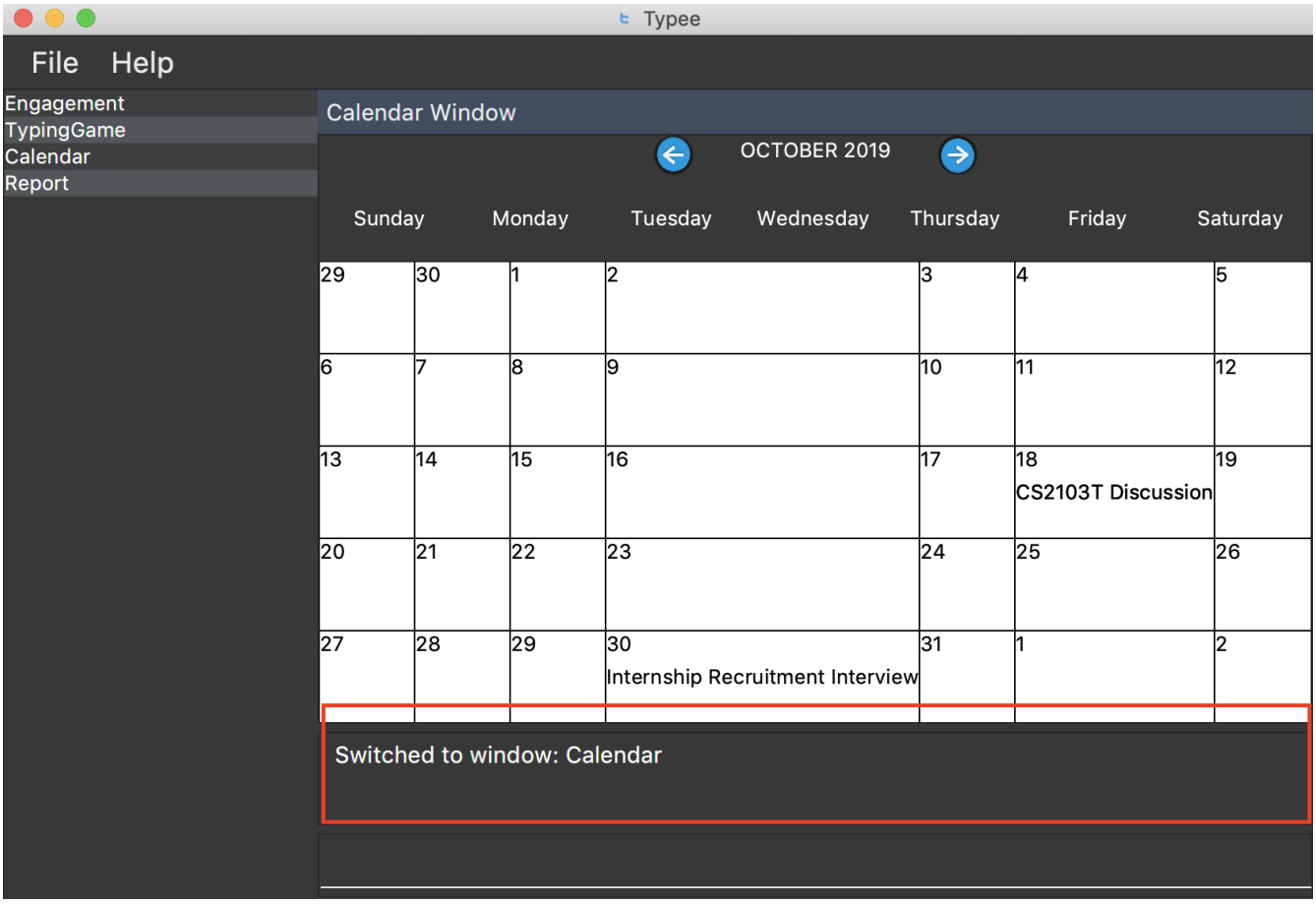


Figure 3. input section with tab command entered `tab calendar`

Below is the screenshot after entering the command.

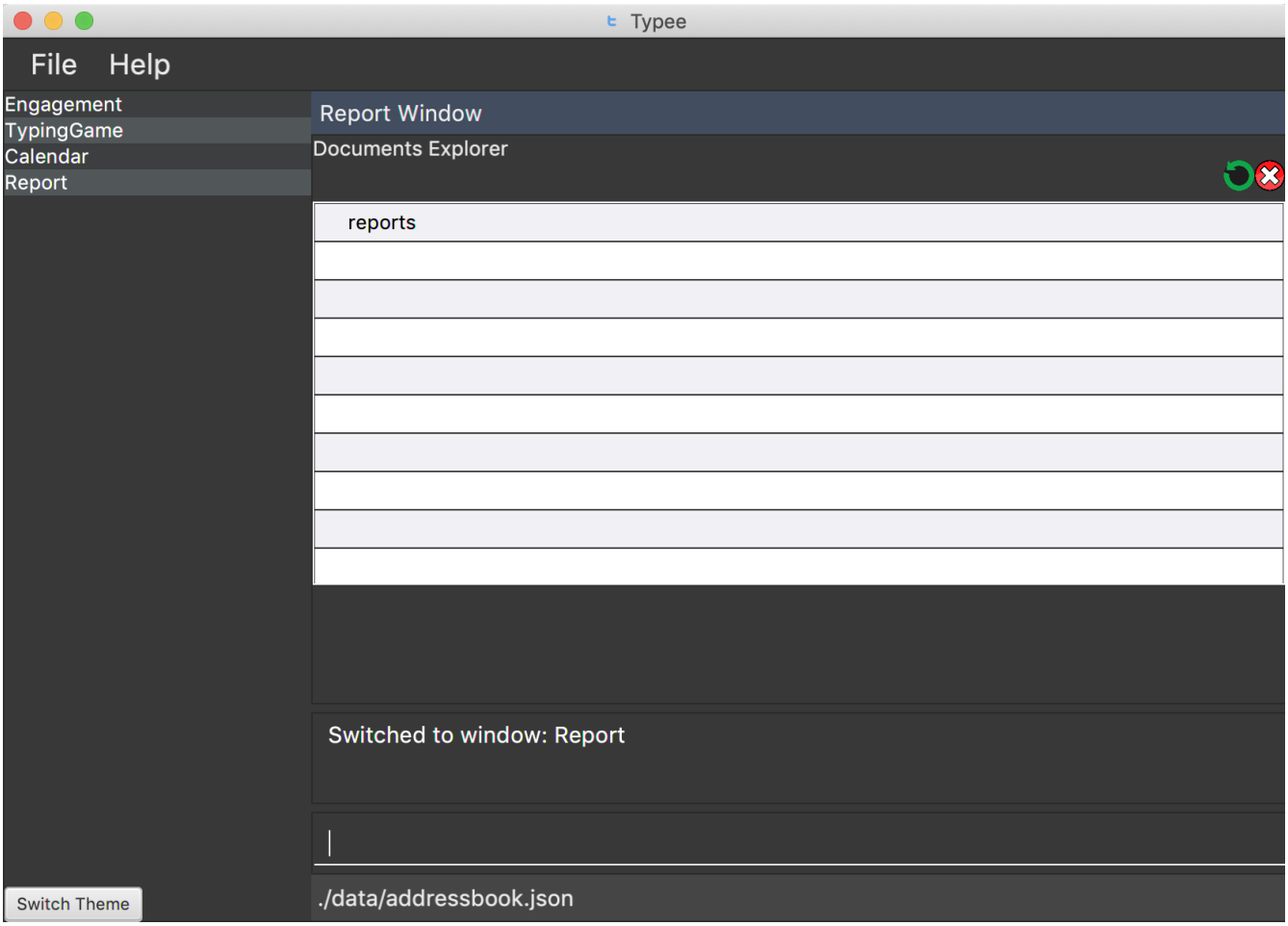


Generating a PDF file of engagement : `pdf`

Pdf Command allows user to create a document of selected engagement in a given format of document template. Document template can be customised based on the customers's requirements, however, default document format will be in an eamil format, where the user can set who the user is going to send this document to inform an engagement.

Format: `pdf i/[index] to/[receiver] from/[sender]`

For example, if the user wants to create a document of an engagement, which has a list index of 1, which can by observed in `engagement` window. User sets the sender as `John`, which is the user's name, and sets `receiver` as `Harry`. Hence, user enters `pdf i/1 to/Harry from/John` to generate the document.



Once user enters the command, system will display the command result in the output panel, showing "Engagement Report successfully generated." Now, if the user clicks the green refresh button on top right of the documents explorer, explorer will display the generated pdf as a list item.

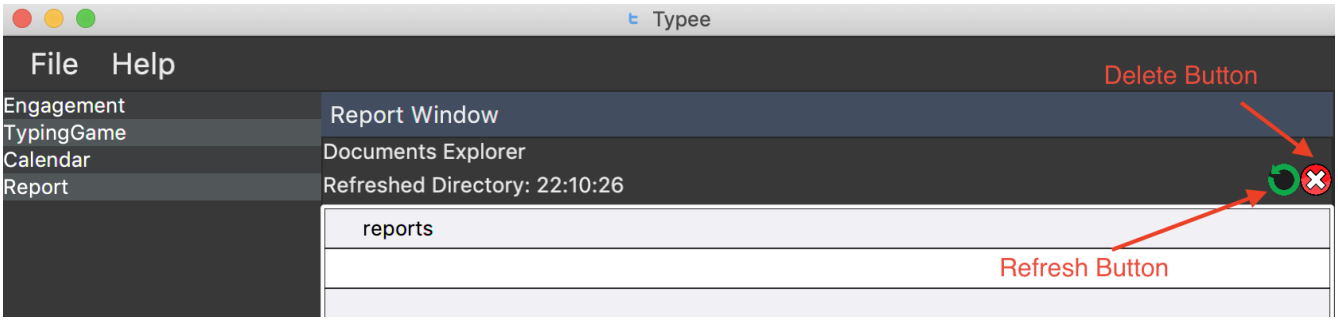


Figure 4. refresh button and delete button in documents explorer.

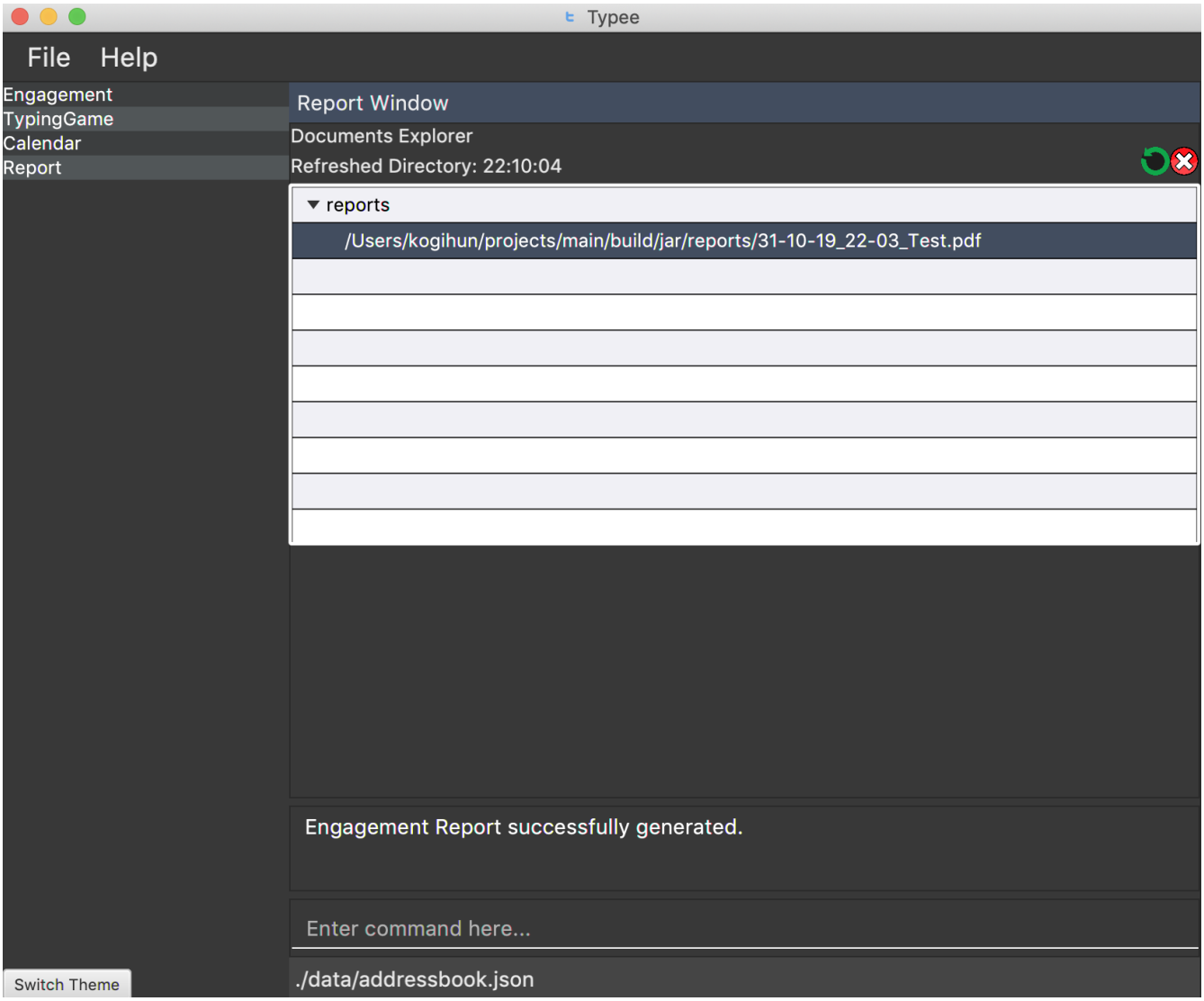


Figure 5. System after clicking refresh button

User can double click the list item to open the document file on their local computer file system. Below is the sample of generated document from our system.

Dear Jason,

You are invited for the following meeting below. Please confirm the meeting information. Should you have any inquiries, please contact via the secretary's email.

Engagement Description:	Test
Venue:	SR-10
Time:	31-10-19_22-03 - 31-10-19_22-03
Attendees	- Uggi

Warm Regards,
Harry
Secretary | SC09-F14-3
e0388888@typee.sg
31 Science Park Rd, Singapore 117611
Typee Pte Ltd

----------This is a computer generated document----------

Figure 6. pdf document sample

If the user does not want to keep the document, instead of directing the actual directory in the local stystem, user can simply click the red `x` button, next to the refresh button to delete the selected document list item. Once system displays the popup message to confirm the user's decision, user will click the `OK` button to confirm deletion. Delete function will only available when user has pre-selected the document list item and the selected item must be a .pdf form, not directory. Below is the screenshot after user clicks the delete button.

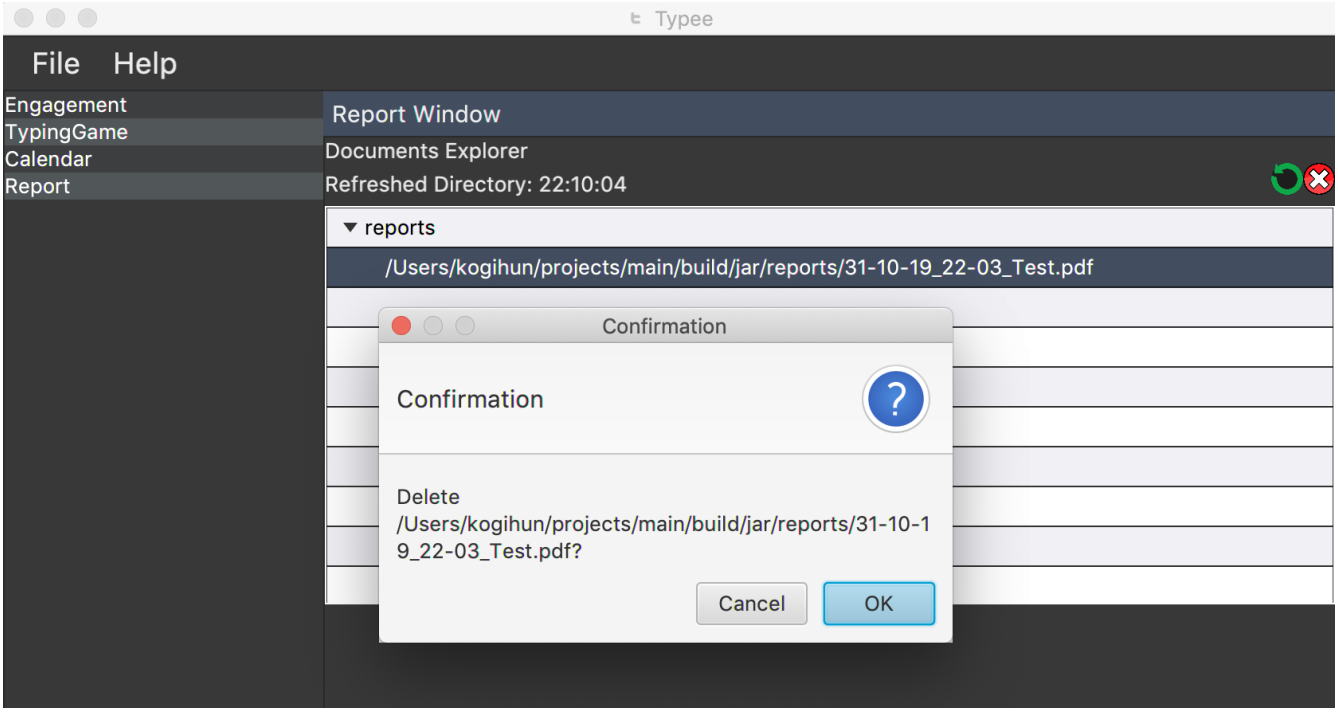
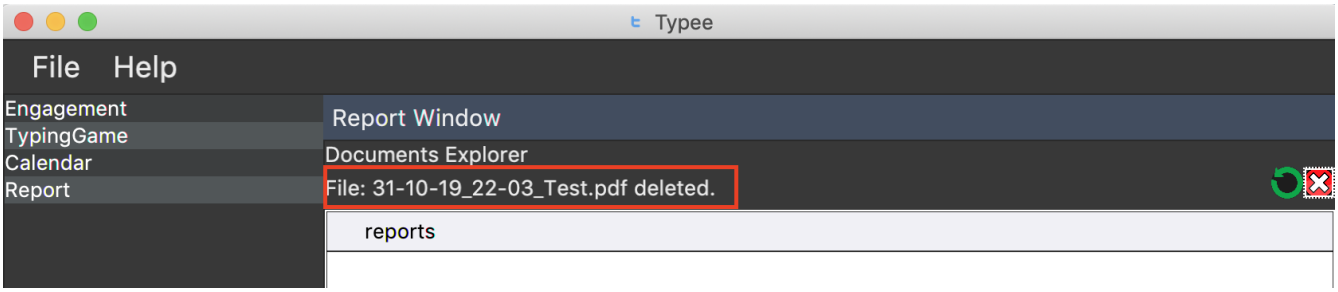


Figure 7. pdf delete pop up message

Now, once system successfully deleted the selected document, system will display the status message above the documents explorer. Below is the screenshot of the system status message after deletion.



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.

Tab Switch Feature

Tab switch feature is a type of Command that allows users to switch to respective windows for using different features in the system. System UI structure is generally divided into 2 parts;

Implementation Structure

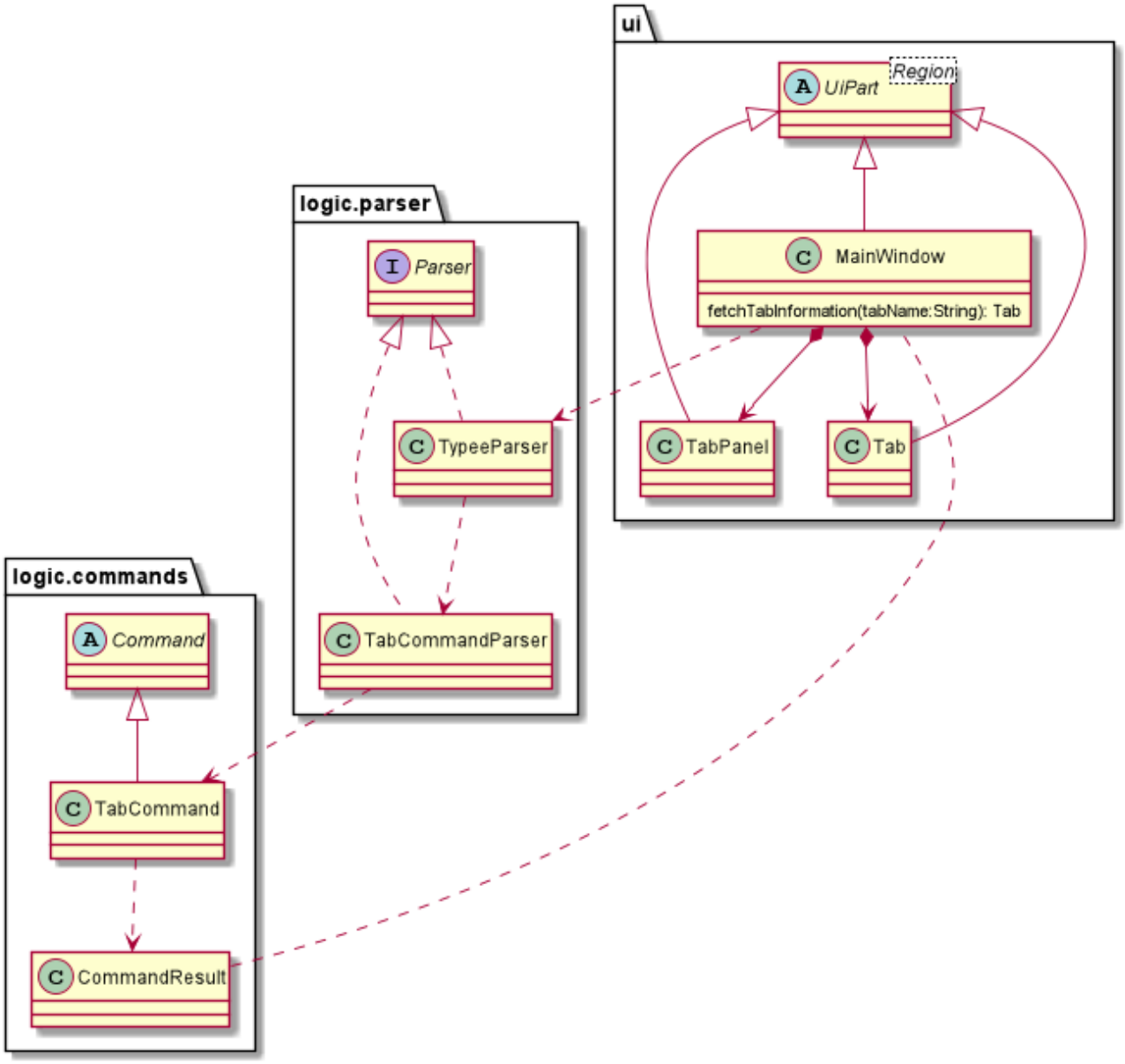


Figure 8. TabCommand Implementation Class Diagram

1. `MainWindow` which contains necessary text input for `Command` inputs and output display for displaying the `CommandResult`
2. Window component, which is a type of `VBox`, that acts as a container for other different UI windows.
3. Tab Menu List, which displays the name of the tab/menus, which has respective separate UI windows. User has to use a specific `Command`; `TabCommand` in order to switch the window to another window.
    - Upon first startup of the system, by system default, system will display the `EngagementList` window.
4. New Ui Model class `Tab` is implemented to contain the respective fxml controller classes in a OOP manner. Below is the class diagram for `Tab` class

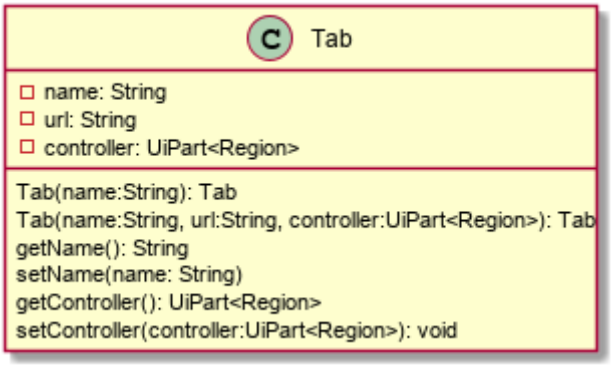


Figure 9. Tab Class Diagram

From figure 9, MainWindow will have an additional method `fetchTabInformation(tabName)`. After the parser executes the `TabCommand`, it will return a CommandResult with `Tab` property. The method will compare the `Tab` in `CommandResult` and once there is a matching result, the method will load the respective fxml file in the MainWindow to display the respective feature window. Below is the basic activity diagram of fetchTabInformation method

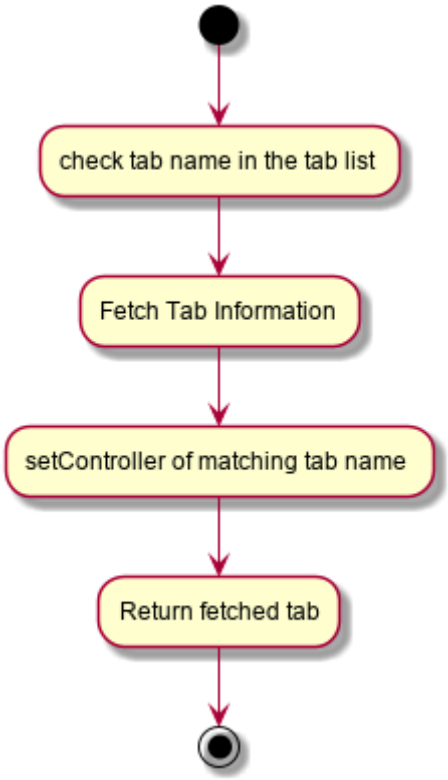


Figure 10. fetchTabInformation method Activity Diagram

Engagement Report Generation

This feature allows user to generate a pre-selected engagement in to a report and save it as a document file. The document file will be created in a .pdf format.

Implementation Structure

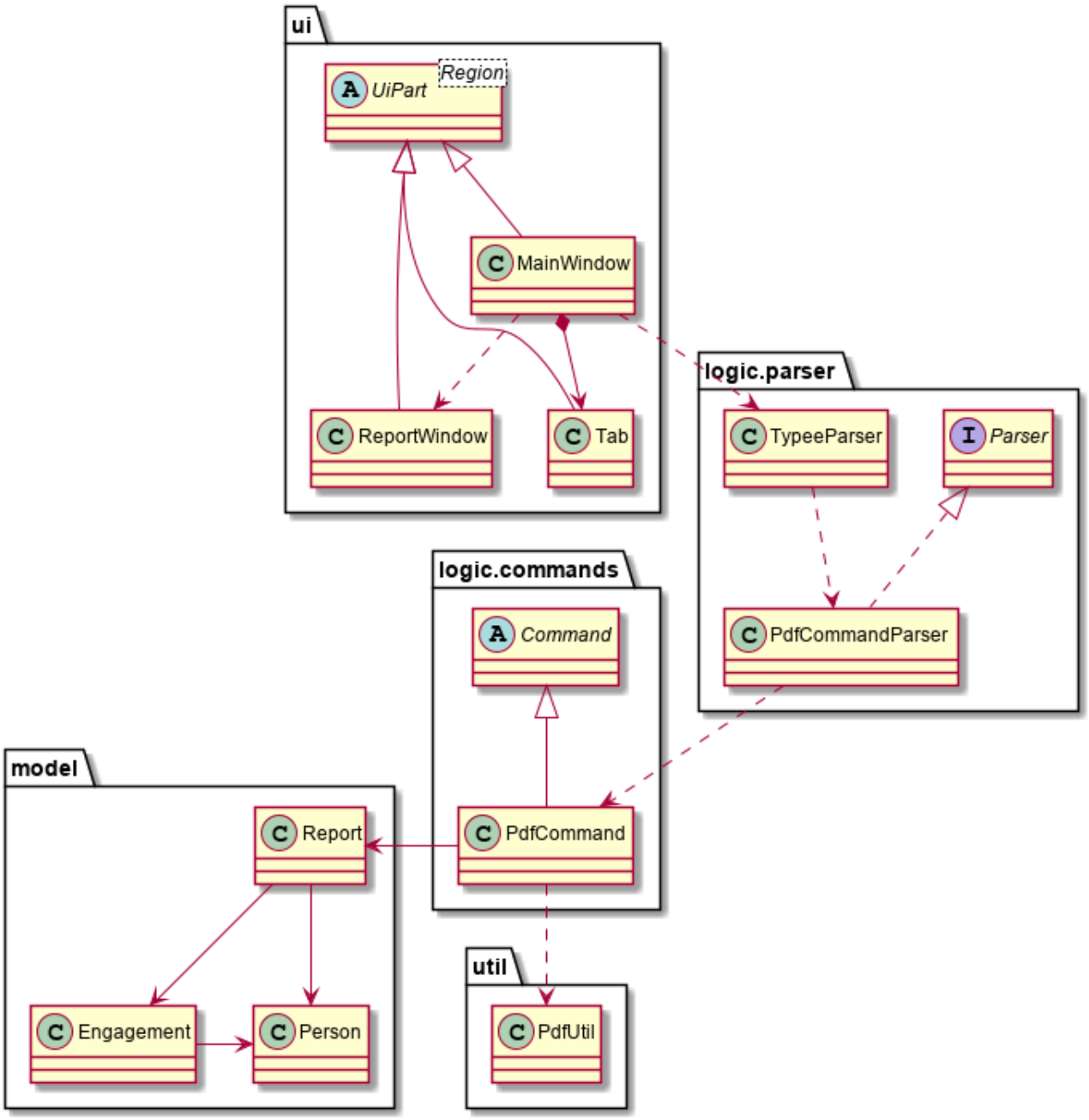


Figure 11. PdfCommand Implementation Class Diagram

The feature will be implemented as an additional type of Command; `PdfCommand`

- Proposed syntax of the PdfCommand is as follow: `pdf i/[engagementList_index] to/[Person] from/[Person]`.

Util class `PdfUtil` will be implemented for handling all pdf document creation related methods.

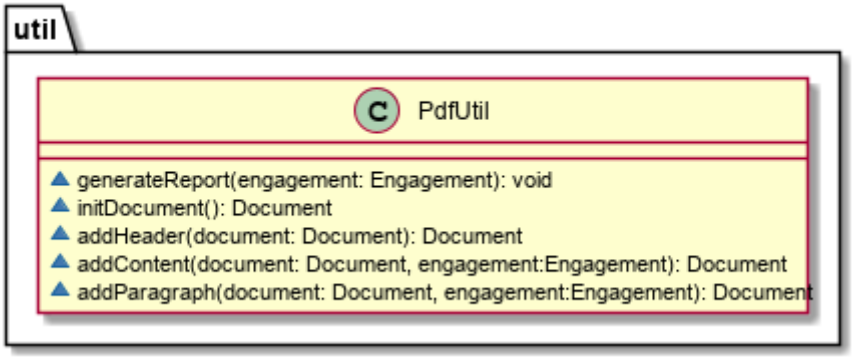


Figure 12. PdfUtil Class Diagram

PdfUtil class will be implemented under `util` package and it will be able to deliver few essential features that are necessary for document creation.

- Able to generate a full report document based on the `Engagement` as input.
- Use different templates for each other types of `Engagement` such as `Appointment`, `Interview` and `Meeting`. Document template will follow general email format: Receiver, Content contained in a table, Sender and signature with address and company logo (Refer to Figure 12). To fulfill the document format, `Report` class needs to be implemented in order to model all necessary properties that has to be in the document. It 3 following properties;
  - `engagement`: specific engagement information to include in the document
  - `to`: A `Person` who is receiving the document
  - `from`: A `Person` who is either a receptionist or a secretary who is sending the document.

UI Design

`ReportWindow` will be the UI container which helps the user to interact using `PdfCommand`. UI will consist of 2 main scrollable panes; Engagement list with index numbers, and the directory explorer for `reports/` as a item list.

UI Components & Features

Table below explains the components that are included in the `ReportWindow` with its purpose and features.

Table 1. Report Window UI Components

	UI Component Type	Feature	Purpose
`EngagementList`	Scrollable Stack Pane	Displays the sorted engagement list	To guide user to help recognize the list index number that the user wants to generate document from.

	UI Component Type	Feature	Purpose
`File_Explorer`	Scrollable Stack Pane	1. Displays the list of document generated previously and stored under the directory `reports/`. 2. Each list item is clickable with a MouseClick action of opening the document.	To allow user to manage documents more time efficiently.

Use case: (UC15) Generate appointment document in PDF format

MSS

1. User requests for generating PDF of an engagement for a selected engagement.
2. System generates a PDF file in a specific external directory
3. System shows successful message

   Use case ends.

Extensions

3a. User enters invalid credentials.

  3a1. System shows error message.

  Use case resumes at step 2.

3b. System fails to generate pdf file.

  3b1. System shows error message.

  Use case resumes at step 2.