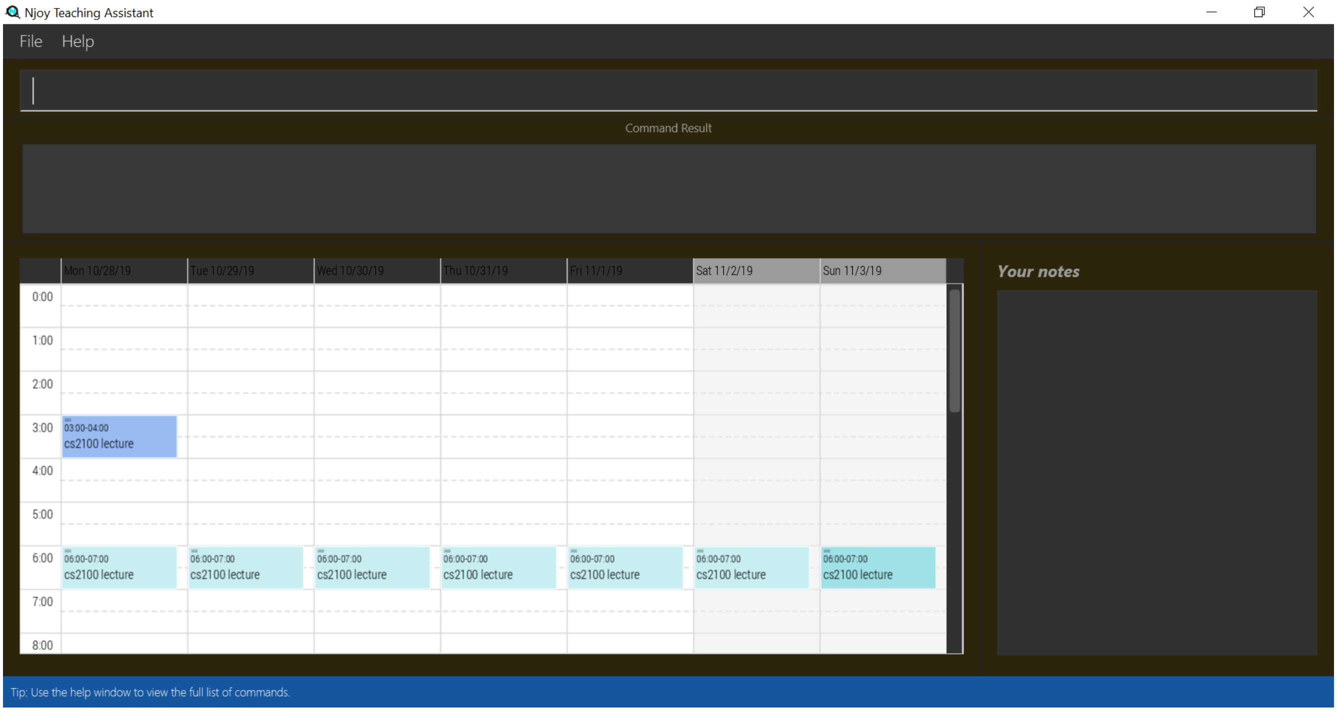
Jeong Yu Han - Project Portfolio

Introduction and Overview

The purpose of this document is to display the author's (Jeong Yu Han) contribution to the software engineering project and his technical skills.

My team and I were tasked with enhancing a command line interface Address Book application for our Software Engineering project. We chose to morph it into a teaching assistant called the Njoy Teaching Assistant, targeted at tech-savvy teachers who wish to improve the efficiency of their daily tasks such as keeping schedules, creating quizzes, keeping track of student performances and keeping timely reminders.

This is what our project looks like:



	Mon 10/28/19	Tue 10/29/19	Wed 10/30/19	Thu 10/31/19	Fri 11/1/19	Sat 11/2/19	Sun 11/3/19
0:00							
1:00							
2:00							
3:00	03:00-04:00 cs2100 lecture						
4:00							
5:00							
6:00	06:00-07:00 cs2100 lecture	06:00-07:00 cs2100 lecture	06:00-07:00 cs2100 lecture	06:00-07:00 cs2100 lecture	06:00-07:00 cs2100 lecture	06:00-07:00 cs2100 lecture	06:00-07:00 cs2100 lecture
7:00							
8:00							

Tip: Use the help window to view the full list of commands.

Figure 1. The graphical user interface for Njoy.

My role was to design and write code for the notes and statistics features. The rest of the sections will cover the summary of my contributions to the codebase, the user guide and the developer guide on how I implemented the features.

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

NOTE	This symbol indicates a note to the user.
TIP	This symbol indicates a user friendly tip.

IMPORTANT	This symbol indicates something important.

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

This indicates important text.

Summary of contributions

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

I implemented the notes specific commands which include:

- **Creating a note**
- **Editing a note**
- **Deleting a note**
- **Listing the notes**
- **Sorting the notes by priority**

I'll be sectioning the functionality into the following format:

1. **Creating**
2. **Editing**
3. **Displaying**
4. **Sorting/Processing**

Enhancements

Creating Notes

- **What it does:** Users can create notes with a title and description. Should the user please, the user is able to assign a priority to the note when creating it. Adding a note will append it to the bottom of the list, and is shown in the User Interface according to its priority. The default priority is unmarked, which has grey panels, while the high, medium and low priority notes are red, yellow and green respectively.
- **Justification:** Teachers often have a dizzying array of tasks to handle. Apart from taking care of their family, they still have to attend to the students, their departmental heads et cetera. They often use applications such as Sticky Notes or Google Calendar to keep track of all their tasks. The notes feature separates the work and home worries that every teacher has to deal with. Teachers also want to be able to look at the most pressing issues at one glance, for nobody wants to miss an important conference.
- **Highlights:** This enhancement works with existing as well as future commands. The implementation for this feature was required serious contemplation in terms of design considerations. While the implementation of additional features such as re-doing, un-doing was

considered, I decided that the main UX principle of the notes feature was that it was supposed to be super lightweight and easy to use. I decided to give the simplest set of commands for this feature while making prioritising on optional utility.

Editing Notes

- **What it does:** Users can choose to edit their notes after creation. The user can do so by specifying the index of the note it is displayed in in the current list shown. He then gives fields to edit, such as the note title, description or priority.
- **Justification:** Users often make mistakes while typing. Teachers often suddenly see shifts in priority as important school events arise, and the flexibility to edit a targeted note or reminder is important.
- **Highlights:** This enhancement works with existing as well as future commands. The implementation for this feature was relatively challenging because of the validation issues. Given any number of optional fields that the user can specify to edit, the feature had to ensure that the edit command was valid - that it maintained the unique property of the note, that there was at least one editable property specified and that the new note was different from the original one.

Displaying Notes

- **What it does:** Shows the list of notes in the command result box. This feature is an alternative solution to the issue of notes having variable length - the notes can have content that exceed its allocated space in the User Interface.
- **Justification:** A teacher might want to keep notes that are quite long but do not fit into its allocated space. This is due to the fact that the notes appear as a panel on the side, leaving space for other features such as the timetable. Should the content be truncated in Njoy, the user can opt for this listing feature to see its full content in the command box.
- **Highlights:** This enhancement works with existing as well as future commands. The implementation for this feature was not difficult but required some consideration about the user's experience.

Code contributed

Please click these links to see my code contributions dashboard.

Other contributions

- Project management:
  - Reviewed team's code coverage.
  - Resolved the issues found by others related to my feature on Github.
- Enhancements to team documentation:
  - Updated README and ContactUs with missing acknowledgement and developer contacts.
  - Updated Glossary and Product Survey : #24

- Enhancements to team code:
    - Wrote the initial skeleton for storage using the Student entity : #47
    - Wrote the skeleton for the generic printable interface for print/export functions : #105
    - Refactor poor practices to follow better scalability and readability : #91
    - Wrote additional tests and generic model stubs for existing features : #123
- Documentation:
    - Made additions to the appendix for Developer Guide and add introductory statements to improve readability: #25
    - Made relevant changes for implemented features in the Developer Guide: #77, #97, #181, #190
- Community:
    - Reviewed Pull Requests of Other Teams: #190, review under username 'lumwb' : #23
- Tools:
    - Change team's libraries to be compatible with both Mac and Windows.

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 to display. This is because some features are repetitive. Therefore, I'll only show one example for each of the main features that I have written about in the user guide.

NOTE The following is an example of **Note Editing**:

Creating a note

Create lightweight, digital notes and reminders quickly and easily as specified.
Format: `note note/… desc/… priority/… (optional)`

The options supported by this feature includes:

1. `note` - The note title.
2. `desc` - The description of the note.
3. `priority(optional)` - The level of priority placed on this note. It can be of type `unmarked`, `low`, `medium` or `high`. It is by default set to unmarked if field is omitted.

NOTE All non-optional fields are required and also cannot consist exclusively of whitespace characters. All leading and trailing whitespaces of the fields will be ignored.

NOTE All priority fields can be in either lowercase or UPPERCASE. (e.g. `note note/sample title desc/sample desc priority/low` is the same as e.g. `note note/sample title desc/sample desc priority/LOW`).

NOTE `priority` levels are distinguishable by their colour in the User Interface. `unmarked` or default notes have grey panels, while `low`, `medium` and `high` priority notes have green, yellow and red panels respectively.

NOTE The notes panel is not responsive. The User Interface only supports title and description of limited length. Notes that exceed this length will be truncated. For information on how to view these notes, see notes list command.

Examples:

- **Unmarked Note**: `note note/Friday morning class 6A desc/give back prelim papers`
  Creates an unmarked note with title *'Friday morning class 6A'* with description *'give back prelim papers'*.
- **High priority note**: `note note/Tuesday 1pm desc/Science Conference priority/high`
  Creates a high priority note with title *'Tuesday 1pm'* with description *'Science Conference'*.

Editing a note

NOTE The following is an example of **Statistics Generation**:

Generating a Statistics report

Create statistics reports of student scores using excel data files. The statistics generated is based on the calculated weighted score of the input data.
Format: `statistics file/… print/…(optional)`

The options supported by this feature includes:

1. `file` - The absolute/relative file path of the data file.
2. `print(optional)` - The filename of the printable report you wish to generate.

NOTE The statistics feature only supports file type in the 'Excel' format. Data files are to end with the *'.xlsx'* extension.

NOTE The `print` option allows the generation of a *'.png'* file containing a snapshot of the statistics report generated. Regardless of the print specifications, a new window is opened with the statistics report.

NOTE The statistics report is stored in the same place as where the JAR file is installed. The report can be found under a newly created `printable` directory. If the directory exists beforehand, no new directory is created.

NOTE Filenames without `` `.png` `` extension will automatically have the extension appended to the end of the file name.

NOTE If the file name specified already exists in the printable directory, it will overwrite the existing file.

Examples:

- **View Statistics Report (without saving)**: `statistics file/C:\Users\SampleUser\Desktop\6B Prelim Scores.xlsx`
  Shows the statistics report for excel file named *'6B Prelim Scores'*.
- **View and Save Statistics Report**: `statistics file/C:\Users\SampleUser\Desktop\6B Prelim Scores.xlsx print/6B Prelim Performance Report`
  Shows the statistics report for excel file named *'6B Prelim Scores'*. It also saves an image file containing a snapshot of the statistics report in the *'printable'* directory with name *'6B Prelim Performance Report'*.

Data File Specifications

The data file needs to be specified in the following format to ensure that the statistics report is successfully generated.

NOTE Failure to adhere to the following specifications might result in either the excel file being rejected or erroneous statistics. If the report is generated, it might be correct but the behavior is not guaranteed should the specifications mentioned below be violated or overlooked.

- **Entries starts at cell A1 with the cell `Students`, regardless of whether there is score data.**
- **First row begins with the cell `Students`, followed by their names. (e.g. row 1 of 4 cells having `Students`, `Jason`, `Mike` and `Peter`)**
- **First column states the different subjects starting from the second topmost cell. (e.g column 1 of 4 cells having `Students`, `Math`, `Science` and `English`.)**
- **There is at least one student.**
- **There is at least one subject.**
- **All student names have unique identifiers. (e.g. two students named Jason could be identified as `Jason 1` and `Jason 2`)**
- **All subject names have unique identifiers. (e.g. two subjects named Math could be identified as `Advanced Math` and `Elementary Math`)**
- **All students have a corresponding score for all subjects**
- **All scores are numeric characters (integer or decimals)**
- **There are no gaps between rows and columns.**
- **All cells within the row and column range have at least one non-whitespace characters.**
- **All cells outside the row and column range are empty.**

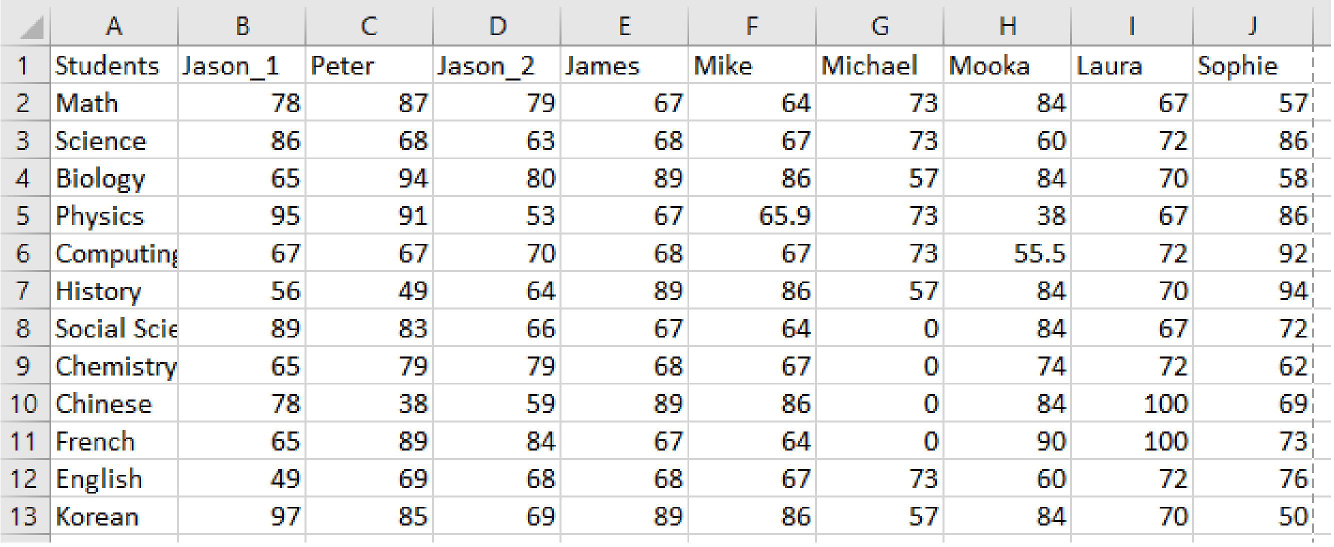
NOTE Failure to adhere to the following specifications might result in some of the statistical data generated not being useful to the user.

- **All scores should be in percentage terms. (0 ~ 100)**

NOTE The general rule for the file format is to ensure all entries in the file are as compact to the top left as possible; the entries should form a rectangular shape on the top left corner of the excel sheet. The rectangle should have no empty entries while all cells outside the rectangle are empty entries.

The statistics report generated uses weighted average scores across the different subjects to perform analysis. All scores carry equal weight. The statistics generated allow you to see some of the common descriptive statistics and their relative distributions both in terms of frequency and percentile terms.

The figure below illustrates a sample data set that fulfills all of the above specifications.



	A	B	C	D	E	F	G	H	I	J
1	Students	Jason_1	Peter	Jason_2	James	Mike	Michael	Mooka	Laura	Sophie
2	Math	78	87	79	67	64	73	84	67	57
3	Science	86	68	63	68	67	73	60	72	86
4	Biology	65	94	80	89	86	57	84	70	58
5	Physics	95	91	53	67	65.9	73	38	67	86
6	Computing	67	67	70	68	67	73	55.5	72	92
7	History	56	49	64	89	86	57	84	70	94
8	Social Scie	89	83	66	67	64	0	84	67	72
9	Chemistry	65	79	79	68	67	0	74	72	62
10	Chinese	78	38	59	89	86	0	84	100	69
11	French	65	89	84	67	64	0	90	100	73
12	English	49	69	68	68	67	73	60	72	76
13	Korean	97	85	69	89	86	57	84	70	50