

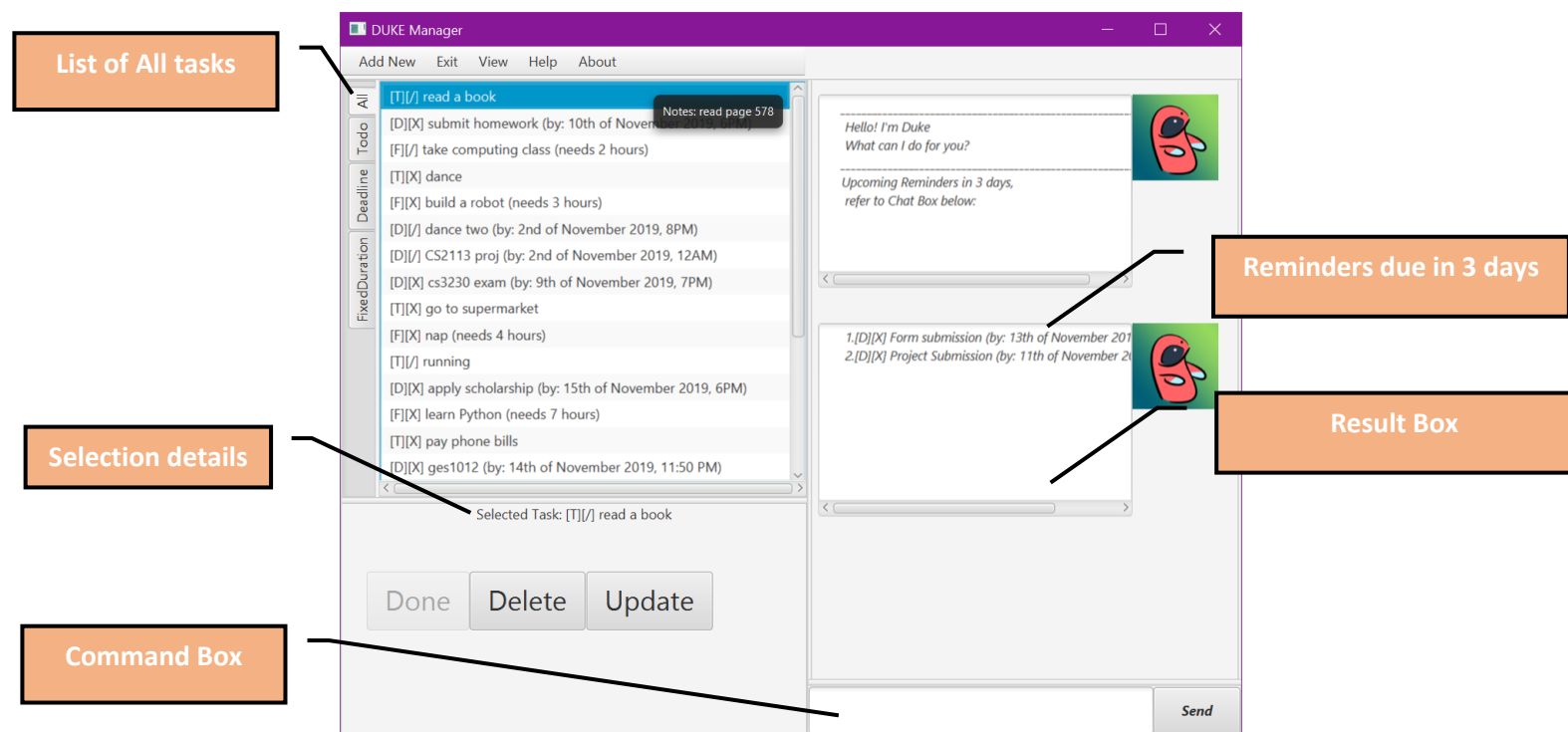
# Gervaise Ang Hui Xuan - Project Portfolio

## PROJECT: DUKE Manager

### Overview

My team of 4 software engineering students and I were tasked with enhancing a basic command line personal assistance interface for our Software Engineering project. We chose to extend our initial interface into a Java-based digital academic handbook for engineering students called **DUKE Manager**, in short of Diary (of the) Undergraduate Kommon Engineer (D.U.K.E.) Manager. This enhanced application enables engineering students to manage student tasks and modules with or without deadlines; show upcoming reminders; and track academic expenses.

This is what our project looks like:



My role was to design and implement codes for the reminders and help features. The following sections illustrate these enhancements in more detail, as well as the

relevant sections I have added to the user and developer guides in relation to these enhancements.

Note the following symbols and formatting used in this document:

---



This symbol denotes information that you may want to take note of.



This symbol denotes a tip that you may find useful when using our application.

Additionally, you may also encounter the following text styles:

**reminders**

A grey highlight (called a mark-up) indicates that this is a command, parameters and file paths will be in this format.

- Important messages will be **bold**

## Summary of Contributions

---

This section contains a summary of my code, documentation and other contributions to the team project.

### Enhancements Added:

- i. Ability to retrieve **reminders** for existing tasks.
  - What it does: allows the user to make use of the saved progress of the tasks keyed in earlier and be reminded of these existing tasks due in 3 days.
  - Justification: This feature improves the product significantly because a user can be reminded of the highest priority or urgent tasks to be completed with a due date in 3 days.

- Highlights: This enhancement affects existing commands and commands to be added in future. It required an in-depth analysis of design alternatives.
- ii. Ability to retrieve **help** commands if a user does not know where to start.
- What it does: allows the user to undo all previous commands one at a time. Preceding undo commands can be reversed by using the redo command.
  - Justification: This feature improves the product significantly because a user can be reminded of the highest priority or urgent tasks to be completed with a due date in 3 days.
  - Highlights: This enhancement affects existing commands and commands to be added in future. It required an in-depth analysis of design alternatives.

**Code contributed:** [\[Code Contribution Dashboard\]](#) [\[Commits\]](#)

#### **Other contributions:**

- Project management:
  - Assisted in approving, reviewing and merging pull requests.
  - Added user stories into GitHub issues.
- Documentation:
  - Updated Developer Guide on **reminder** and **help** features alongside model component.
  - Updated User Guide to ensure **reminder** and **help** features added by me are up to date.
- Community:
  - PRs reviewed (with non-trivial review comments): [#42](#) [#123](#) [#168](#) [#226](#) [#231](#)
  - Reported bugs [#1](#) [#2](#) [#3](#) [#4](#) [#5](#) [#6](#)
- Tools:
  - Integrated Codacy **Code analytics tool** [#231](#)

## Contributions to User Guide

---

We had to update our enhanced DUKE Manager User Guide for the enhancements that we had added. The following is an excerpt from our **DUKE Manager User Guide**, showing additions that I have made for the **reminder** and **help** features.

This section also contains an excerpt for the improvised help feature that I have planned for the next version (v2.0) of DUKE Manager.

### Retrieving upcoming reminders of tasks due in 3 days:

Upon **proper exiting** of Duke Manager through the exit command, the progress of all the data will be saved. Next, open/double click the Jar File once again, this returns to the program after exiting for the first time previously.

This will then prompt Duke Manager to load the saved progress of the inputs you have keyed in earlier as existing tasks. Additionally, the interface on the right will retrieve and display the upcoming reminders of tasks **due in 3 days from the current date today**.



Does not require any additional input unless it does not meet prerequisites.

#### Prerequisites:

- Ensure that there is an existing Priority-Related Task Input (task with a time and date) added previously.

Example: **deadline** <task description> /by <date and time>

- Ensure that the existing Priority-Related Task added previously <date and time> is set to between 1-3 days after <today'sDate>.

Example: Today's date is 01/11/2019. Must exist a task whereby it is already saved in the program. eg. **deadline** <task description> /by 03/11/2019 18:00

## View all available functions or commands: help

View all the functions and commands that DUKE Manager have.

Format: help



A generated message shows "Help window will pop up: Shows all the commands available in Duke Manager" before the Help Window GUI is opened.

## Opens the interface guide: help

Shows a friendly help tutorial on how to use DUKE Manager. This opens a new help window to allow the user to view commands via user friendly interface.

Format: help

# Contributions to the Developer Guide

---

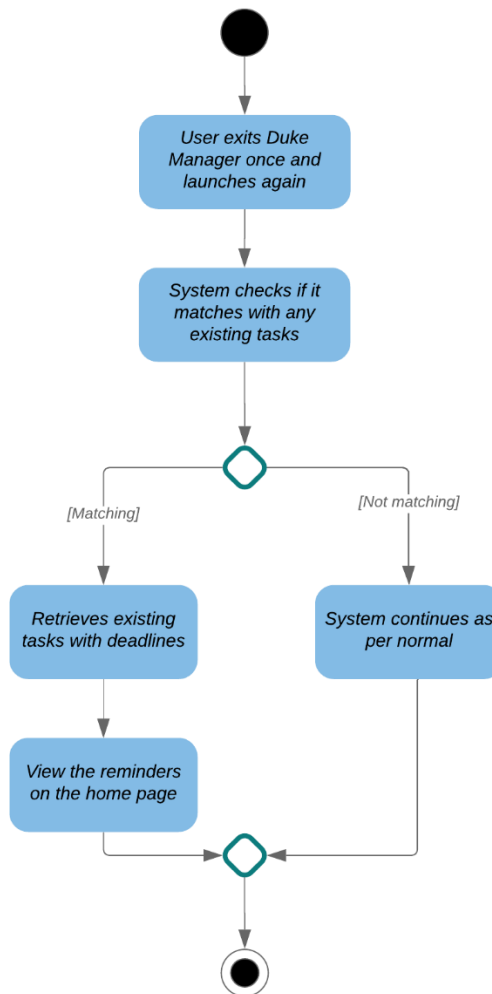
The following is an excerpt from our **DUKE Manager Developer Guide**, showing additions that I have made for the reminder and help features.

## Reminders Feature

The reminders feature will allow the user to retrieve the existing tasks with a due date and time specified in the task list of Duke Manager and display it as a reminder. This will allow the user to be reminded of the task which have a due date in 3 days from today's date at the home page.

## Implementation

The following diagram shows the activity diagram of getting reminders in Duke Manager.



The feature is activated upon properly exiting Duke Manager the first time previously and launched again for the second time. Upon entering the class Reminders, the following steps are taken by the application:

The `getReminders()` method does not require any additional input. However, ensure that there is an existing Priority-Related Task Input (task with a time and date) added previously and this same existing task <date and time> is set to 1-3 days after <today'sDate>.

**Step 1.** DUKE Manager is launched for the second time. It attempts to load for each existing tasks in storing in `TaskList()` array, which also refer to the directory of the saved data files in file explorer.

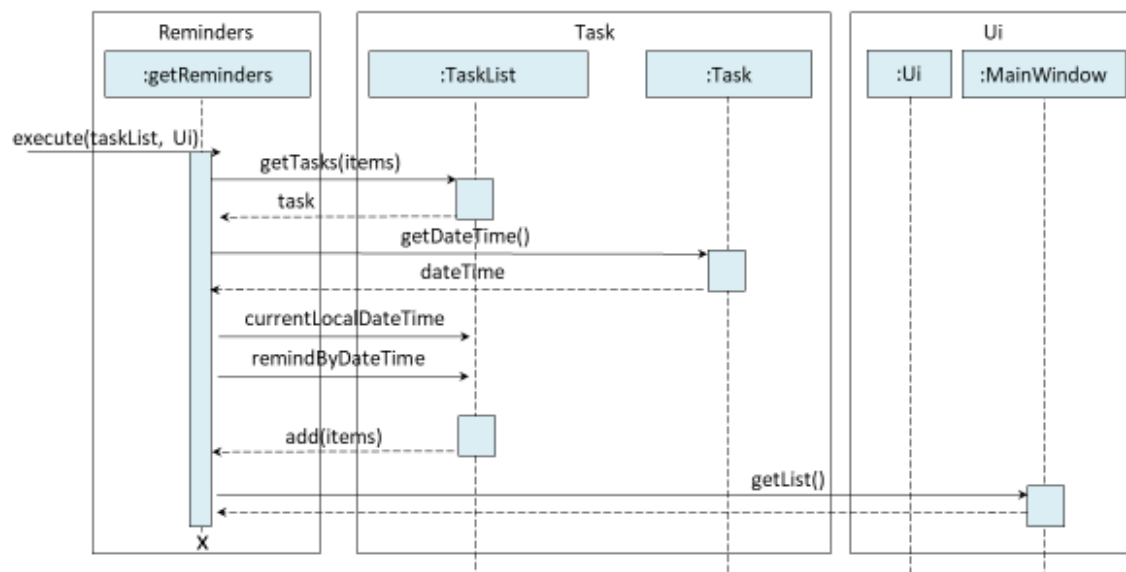
**Step 2.** After successfully retrieving existing tasks with a due date and time, it gets the current date and time where the `LocalDateTime` is tracked.

**Step 3.** To set a reminder for the task due in 3 days, it compares the current date and time with the existing task due in that <date and time>.

**Step 4.** If the task is due after 3 days from the current `LocalDateTime` and the task is due before the `remindByDateTime`, the `add()` method implemented in `TaskList()` will be invoked.

**Step 5.** Therefore, at the home page, there will be a section on upcoming reminders where existing tasks in the TaskList() will call the getReminders() method which is set to remind only existing tasks between 1-3 days after today's date.

Below is the sequence diagram of retrieving the list of tasks due in 3 days as reminders.



## Design Considerations

Aspect: Importance of being able to remind user of upcoming tasks

Due to our natural daily lifestyle which consists of tasks and personal goals to finish these tasks by a certain date and time. A reminder feature is useful in this case after user inputs all his tasks and returns to the program after exiting for the first time. Therefore, it was implemented so that upon proper exiting of Duke Manager through the exit command, the updated progress of all the data will always be saved where it will retrieve tasks which have a due date in 3 days from today's date.

## Help Feature

This feature allows users to view all available commands in Duke Manager. It comprises of the following functions: Selecting a command, followed by a short description, format and an example of a particular command.

## Implementation

The help Function is facilitated by the MainWindow from the UI Component. A drop-down list to select a command which requires help is created in this process.

The exact implementation is shown below:

```
@FXML
public void createHelpWindow() {
    try {
        FXMLLoader fxmlLoader = new FXMLLoader(Main.class.getResource( name: "/view/HelpWindow.fxml"));
        AnchorPane ap = fxmlLoader.load();
        Scene scene = new Scene(ap);
        Stage stage = new Stage();
        stage.setScene(scene);
        stage.setAlwaysOnTop(true);
        fxmlLoader.<HelpWindow>getController().setHelpWindow();
        stage.show();
    } catch (IOException e) {
        Logr.log(Level.SEVERE, msg: "Unable to load help window", e);
        e.printStackTrace();
    }
}
```

When the help command is entered, the MainWindow from the UI Component prompts the user-friendly interface of help window so that the user can navigate freely to be aware of the functions available in Duke Manager and what each command specifically does.

```
else if (cmd instanceof HelpCommand) {
    duke.saveState(cmd);
    response = Ui.showHelpGui();
    createHelpWindow();
    dialogContainer.getChildren().add(
        DialogBox.getDukeDialog(response, dukeImage)
    );
}
```



## Design Considerations

### Aspect: Types of help

- Alternative 1: Dynamic help command to suggest format when help followed by a space is being entered.
  - Pros: Can dynamically retrieve help instantly when unsure of a command format.
  - Cons: As the user is unsure of the system, this method might not allow user to be aware of the functions available in Duke Manager and what each command specifically does.
- Alternative 2 (current choice): Help window pops up with a friendly user interface for user to fuss-free navigation and usage of Duke Manager.
  - Pros: Very simple for the user to receive help for a command format he/she is unsure of.
  - Cons: View the entire description of one single command one at a time.

## Instructions for Manual Testing

### Receiving Reminders

1. Deadline Task
  - a. Test case: deadline <task description> /by 03/11/2019 18:00  
[Expected: Returns reminders for tasks due in upcoming 3 days before the deadline]
  - b. Test case: deadline <task description> /by 03/10/2019 18:00  
[Expected: Returns no existing reminders if the task deadline is over]