Yang Zi Yun - Project Portfolio

PROJECT: Duke Email Manager

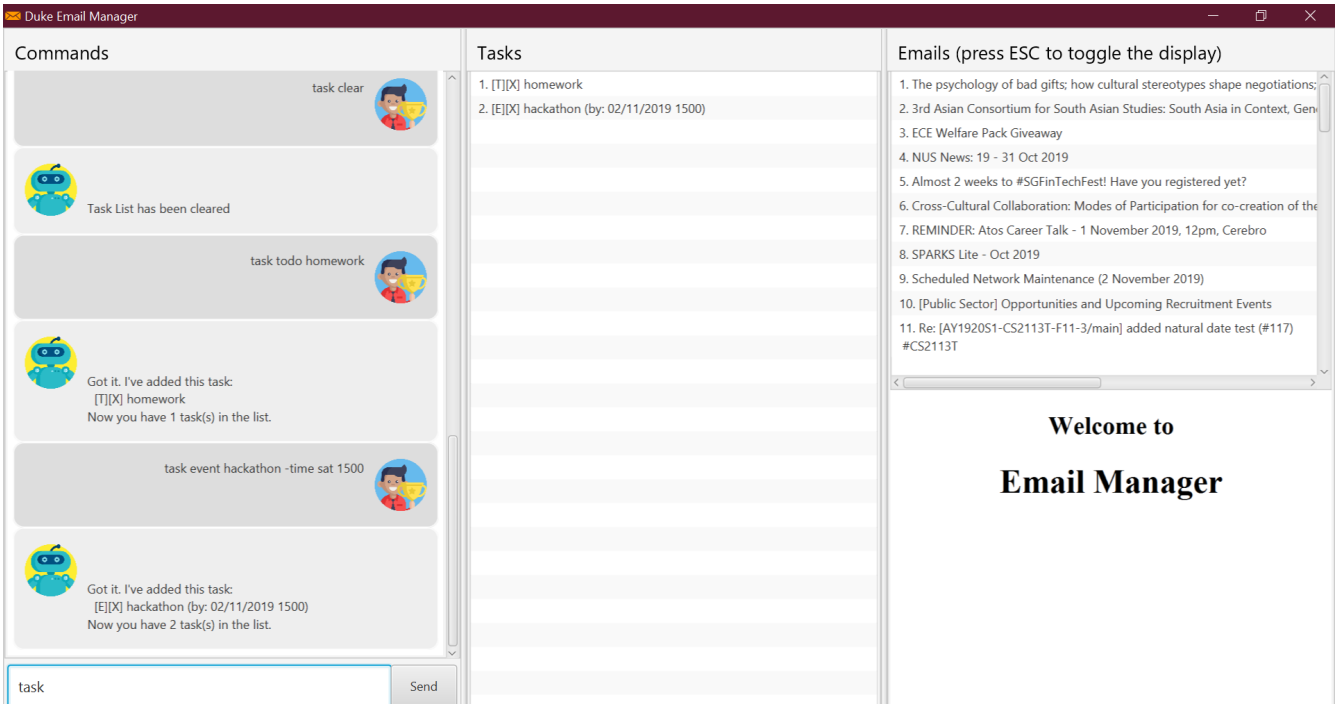
1. Product Overview

My team and I developed an Email Manager as a solution for the overwhelming emails received daily in our mailbox.

We were tasked with enhancing a basic Command Line Interface (CLI) task manager for our Software Engineering project. We chose to morph it into an email manager while enhancing the task manager. We also provide ways to link the management between tasks and emails.

Duke Email Manager is an email and task manager desktop app, specifically designed for NUS School of Computing Students to manage their emails and busy schedules. As a text-based application, it is optimized for those who prefer typing and working with CLI. Email Manager also has a developed Graphical User Interface (GUI) that allows users to view email and task details in an appealing, well-organized format.

The figure below shows interface of our app:



My role was mainly to design and write the codes for email managing including email showing, email tagging and email filtering. 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.

2. Summary of contributions

This section shows a summary of my coding, documentation, and other helpful contributions to the team project.

- **Code contributed**: [Summary of code contribution]
- **Enhancement added**:
  1. Email Tagging
     - What it does: Students can tag their emails with the command `email update <index> -tag <TagName1> -tag <TagName2>`
     - Justification: By tagging the emails, students can have a clearer overview of their list of emails. Hence, they can attend important emails in the fastest time. This also minimizes the chances of missing out crucial information being informed by email.
     - Highlights: Multiple tags can be added to an email. Duplicated tags can be detected and not added to the email again.
     - Code contributed: #82
  2. Email Filtering by Tag(s)
     - What it does: Students can filter the emails by the tags added with the command `email list -tag <TagName1> -tag <TagName2>`
     - Justification: This allows the students to view the list of emails being tagged with particular tag(s), helping them to filter out relevant emails to keep their email organised.
     - Highlights: Maximum number of input tags allowed is two, this will shows emails that are tagged with both tags.
     - Code contributed: #97 #105
  3. Email Showing
     - What it does: The command `email show <index>` will display the email in html format.
     - Justification: Student can view the content of email in the application without have to find the email in other places. This helps the student to access all the information in the email all in one place.
     - Highlights: Pressing `ESC` key can toggle the display between email list view and email content view.
     - Code contributed: #71
  4. Switching mode between email and task
     - What it does: Students can easily switch the mode between `email` and `task` with the command `flip`. In the user input text field, a prefix either `email` or `task` will be displayed with respect to the current mode.
     - Justification: Students will need to frequently change their input when they try to handle their tasks and emails, this feature allows them to navigate between both modes easily.

        - Highlights: The prefix in the user input text field is design to reduce the amount of typing required from the students. The prefix is non-deletable.
        - Code contributed: #54

- **Other contributions**:
    - Project management:
        - Managed releases versions `v1.2.1` on GitHub (#v1.2.1)
    - Documentation:
        - Update User Guide and Developer Guide
    - Community:
        - PRs reviewed (with non-trivial review comments)
        - Reported bugs and offered suggestions for other teams in the class ()
    - Tools:
        - Integrated continuous integration (Travis) to the team repo (#36) (#37) (#38)
        - Integrated coverage report (Coveralls) to the team repo (#107 #102)
    - GUI enhancement:
        - Implemented a different background colour for UserDialogBox and DukeDialogBox for clearer layout and view purpose (#114)
        - Resize window to fit screen (#71)
        - Display of email using WebView and toggling email list view and email content view by pressing `ESC` key (#71)
    - Other functionalities or feature:
        - Task detection of anomalies (#32)
        - Basic email class implementation (#48 #49 #50)
        - Wrote additional tests for existing features to increase instruction coverage from 19% to 29%, and increase branch coverage from 15% to 20% (#114)
        - Implemented logger (#131)
        - Added key binding functionality to create keyboard shortcut (#78 #70)

3. Contributions to the User Guide

We had to update the User Guide with instructions for the enhancements that we had added. The following is an excerpt from our Email Manager User Guide, showing additions that I have made for the email managing features.

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

4. 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.