Chua Sim Nee - Project Portfolio

PROJECT: ORGANice

1. Introduction

This document aims to show an overview of my involvements in the project, ORGANice. My role in this project was to implement the task list for the users to track the administrative tasks that needs to be done.

1.1. Project Description

ORGANice is developed by a team of five students in National University of Singapore (NUS). All of us are year 2 Computer Science students,taking the Software Engineering module, CS2103T.

1.2. Project Scope

ORGANice was made for our CS2103T module's team project (TP). Over the course of 6 weeks, we were tasked with either enhancing or morphing a Command Line Interface (CLI) desktop addressbook application. We chose to morph it into an organ transplant manager, ORGANice. It has a Graphical User Interface (GUI) which is created with JavaFX.

The target audiences of ORGANice are the hospital administrative staffs as we aim to help ease their workload when trying to match all their patients with all the donors' organs they have.

1.3. Project Summary

ORGANice is an organ transplant manager. Currently, hospital administrative staffs have many patients and organ donors information in their records. Whenever, they have a new patient who needed an organ, they often have to go through many manual search and matching to find a compatible donor. We aim to reduce such menial work using ORGANice and speed up this process.

ORGANice is able to:

- `add` and `edit` patients, donors and doctors information.
- `list` the entire database information.
- `match` the patients and show potential compatible donor's information.
- `sort` the result after using `match` command.
- `find` a specific patient, donor or doctor in ORGANice.
- `processing` a patient and a donor who are compatible to generate a list of administrative tasks to do for the patient and donor.

- `done` processing a patient and donor pair and determine if they can be removed from the matching pool.

This is what ORGANice looks like:

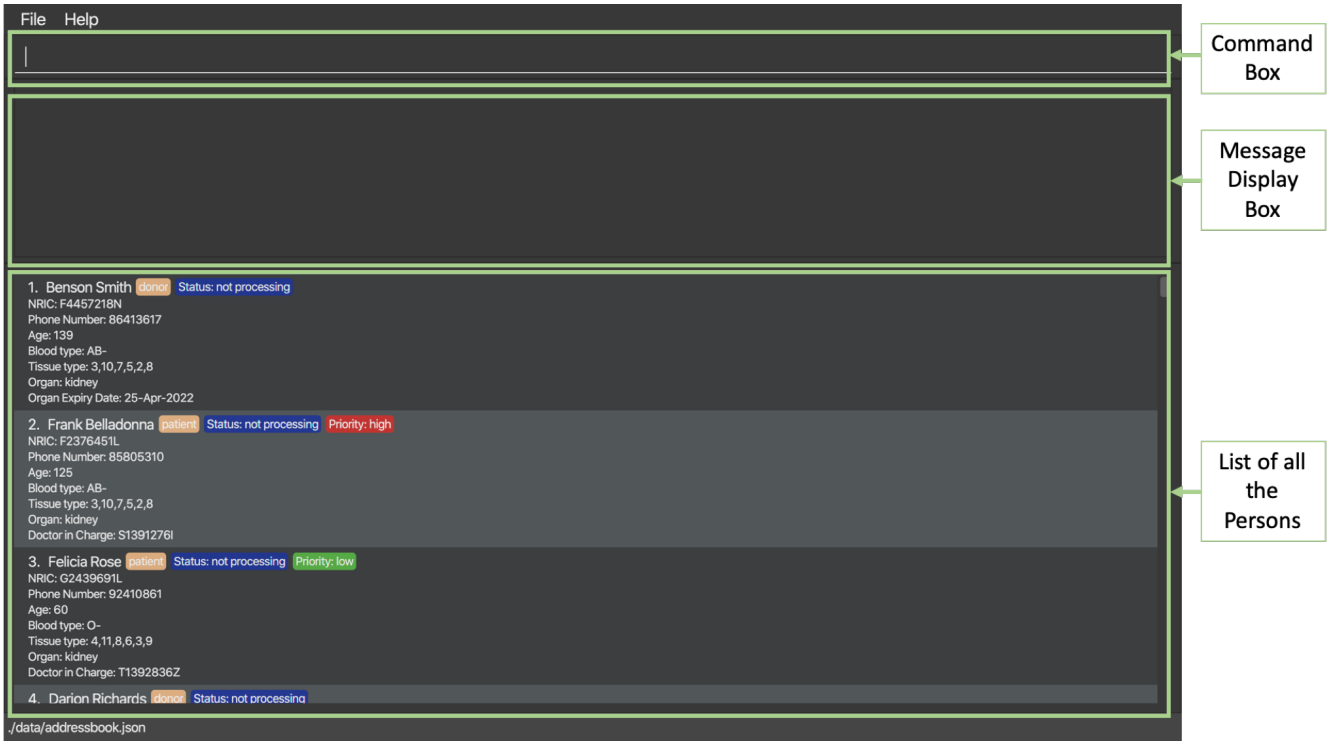

Figure 1. The graphical user interface for ORGANice.

1.4. Key formatting

2. Summary of Contributions

This section lists the codes and documentations I contributed to ORGANice.

I was responsible of implementing the `processing`, `processingMarkDone` and `done` commands. I also ensured the code quality is up to standard.

2.1. Main Feature : `Processing`

This section will explain the command, `processing` I implemented for ORGANice.

- What it does:
  It allow user to view a whole list of administrative tasks which needs to be done to arrange for cross-matching test between a patient and a donor.
- Justification:
  Hospital staffs often need to do a lot of paper work and administrative tasks. There is now an increasing number of patients admitting to hospitals and if they need an organ, there will be a standard operating procedure which need to be followed. With ORGANice, staffs will not need to manually keep track of which tasks are not yet done for every patients.
- Highlights:

A default task list will be generated when the user use the command with valid NRICs of donor and patient. The task list generated belong to the donor and patient only. This will not lead to any confusions as it is impossible to have a donor matching with more than one patient or a patient matching with more than one donor.

2.2. Main Feature : `ProcessingMarkDone`

This section will explain the command, `processingMarkDone` I implemented for ORGANice.

- What it does:
  It allows users to mark a task on the list of administrative tasks as done.
- Justification:
  This command will allow the hospital staffs to have a better overview of what are the existing tasks which need to be done. This will reduce the chance of hospital staffs doing a single task for more than one time or miss out a task by accident.
- Highlights:
  This feature cannot be used if the patient and donor have not been processed before. This will ensure that the correct list is edited instead of a wrong one. The list will also be generated with either a cross, □, or a tick, ☑ beside each task. It will be very intuitive for the hospital staffs to determine which tasks are done already.

2.3. Main Feature : `Done`

This section will explain the command, `done` I implemented for ORGANice.

- What it does:
  It allows users to determine if a patient and a donor is all done. Meaning, the patient and donor have gone through a test and we know know if the patient rejected the organ from the donor or not. If the patient is not compatible with the donor, then both the patient and donor will be saved back into ORGANice and available for future organ matching detection. However, they will not be able to match each other anymore. If the patient is compatible with the donor, then both of the patient and donor will be removed from ORGANice.
- Justification:
  Our algorithm is only able to find the compatibility of the blood type and tissue type of the patient and donor pair. However, a cross-matching test is needed to determine if the patient will reject the donor's organ. Hence, there is a chance that the result of the cross-matching test is negative and the patient and donor will need to find another match.
- Highlights:
  This feature will save a history of all rejected patients a donor have so as to not allow the donor to be matched again with them. This will ensure that the same patient and donor do not go through another round of cross-matching just to return the same negative result.

2.4. Code Contributed

Please refer to this link to view the code I wrote: [RepoSense]

2.5. Other Contributions

- **Enhancements**
- **Documentations**
- **Community**

3. Contributions to User Guide

3.1. Current Enhancement

3.2. Proposed Enhancement for v2.0

4. Contributions to Developer Guide

4.1. Current Enhancement

4.2. Proposed Enhancement for v2.0