Design Document #4

A description of how your program does or could follow the 7 principles of universal design (from the ethics lecture).

Preface: Our program satisfies all of the Principles of Universal Design, however, there are a couple of features that COULD be implemented to further adhere to certain principles.

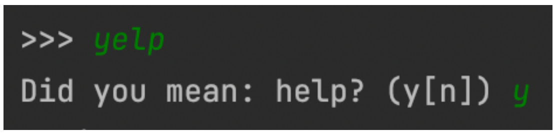
Table of contents:

Principle 1: Equitable Use

- We provide the same level of security to all users by enforcing a minimum length for their usernames and passwords.
- We could create a text-to-speech feature for the visually impaired, in order to provide the same means of use for all users.
- To make the design more appealing, we could create a GUI and make it aesthetically pleasing with colours and images.
- We also provide the info about logging for all users. This way they can track whether anyone accessed their account without their knowledge.

Principle 2: Flexibility in Use

- The help command lists all the possible commands the user can call. These commands are enumerated hence giving the user an option to either enter the name or the number of the command they want to call.
- We have also implemented a feature to accommodate spelling mistakes by the user. The program corrects the spelling of the user if they are off by a few characters and allows them to respond yes/no to whether they wanted to call that specific command instead of typing out the whole command again.


```
>>> yelp
Did you mean: help? (y[n]) y
```

- Users of our program only need a keyboard to access our program which means there won't be any issue with users being either left-handed or right-handed.

Principle 3: Simple and Intuitive Use
- To make the user interface simpler/intuitive to use, there is a designated 'help' command available to the user that allows them to see all possible actions they can take at the given moment.
- We use simple and to-the-point vocabulary for our command names, which accommodates users with all sorts of literacy skills and eliminates unnecessary complexity.
- In the future, we can implement the presenters in different languages to accommodate users who are more comfortable speaking in a language other than English. This will not be too difficult because we followed Clean Architecture, and so we only have to change our presenters a bit to achieve this.
- We arrange information consistent with its importance by providing a command to access all prescriptions and another command to access non-expired prescriptions. Furthermore, we have commands to show prescriptions with their details or to only show the name of the prescription.
- Finally, we always provide messages when a user enters a new menu and provide success or error messages whenever a user inputs a command that requires one. For example, when creating a new account, changing your contact information, creating a prescription, and more.

Principle 4: Perceptible Information
- Using our command design in the terminal, we can provide a verbal mode of communication to users. Just like with the principle of Equitable Use, we can create a GUI to help us display images and use the pictorial mode of communication, and create a text-to-speech feature that will allow our program to be compatible with the visually impaired.
- By using colons, we are able to distinguish between the text providing context and the essential information. For example, This provides contrast and maximizes the legibility of important information.

Principle 5: Tolerance for Error
- Throughout our program, there are multiple warnings set up for actions that the user can take that will have significant repercussions. Some instances of this can be:
    1. If an admin user wants to delete the account of any user, they will be first prompted with a warning message stating the consequences of the action.

2. If a secretary wants to cancel an appointment for a patient, then they are first prompted with a warning message stating what action they are about to undertake.

- We provide fail-safe features for instances when any user might forget their password. All admins are allowed to change the password of any user in the instance they forget their password.
- Other failsafe features included are the ability to reschedule/cancel appointments, and delete reports/prescriptions in the case any error has been made while creating them.
- We used the Levenshtein Distance to increase tolerance for error. It enables the program to suggest an alternate spelling on the terminal if the user makes a spelling mistake. The user can then accept the alternate spelling which would then call the related command or the user can reject the spelling.

Principle 6: Low Physical Effort

- Across our program, we have maintained a uniform naming convention for all commands the user can call. This makes it easier for the user to remember commands and it means that the user wouldn't have to call 'help' every time he has to call a command.
  An example of this can be the use of the 'back' command as an alias for unloading patients from and signing out of different stages of the program.
- As a measure to reduce physical effort, we allow the user to call a command by the number associated with it in the help directory instead of having to type out the entire command every single time.

Principle 7: Size and Space for Approach and Use

- By using the command pattern in the terminal, users need only to use their keyboard to interact with our program. This eliminates the need for a mouse and accommodates people with a small mouse space or with people who have bad aim with a mouse.
- Furthermore, because users are only interacting with their keyboard, if they are comfortable with using a keyboard then they are comfortable with using our program.