Name: Mfonisoabasi James
S/N: 20034498
Concepts & Ideas learned in class and incorporated into app development

User Experience

"Are we making the content in our software solutions to our customers findable, valuable, discoverable and meaningful?", was a question posed during the lecture. I learned that having good user experiences is what can change your customers from not wanting to engage with the application to picking it up and using it. From the design phase of my application to the development phase my goal has been to have a very simple and clean user interface without any hassle. I spent some time researching colour palettes as I am aware that colours make us feel things as humans and if used in the right context, can make people feel positive. So the application is built around having a simple user interface that anyone can get behind, with a pleasing, positive and vibrant colour scheme that helps to keep users using the application. There are 3 colours that make up the palette for this project:

1. Orange represents warmth and joy. It is an uplifting and optimistic hue that encourages creativity. Essentially a perfect blend for those picking up vocational skills
2. Blue tends to have a calming effect on our minds, making us feel secure and confident. This certainly is a feeling I would love the community to get while using the app
3. White, very often associated with simplicity and perfection. This was a big criteria in our design phase and having elements of white in there certainly helps tie the rest of the app's simplicity together.

Keeping it "Dry"

The "DRY" acronym as taught during the lecture stands for "Don't repeat yourself". Our job as designers and developers is that we are knowledge managers. We collect, organize, maintain and harness knowledge. Maintenance as a software steward happens at the very beginning of our idea iterations and is something that constantly happens throughout the development process. Knowledge management and maintenance bleed into one of our software design motos which is the idea of keeping it DRY vs WET (write everything twice). I definitely faced a WET situation when trying to build custom headers and widgets for my application. As such, it was important to create a function for those custom widgets where I can easily pull from anywhere in the app without having to rewrite the same code every time I needed it's implementation. This helped save me a lot of time in the coding process and also helped me keep my code base more manageable with less room for breakage.

Quality

Quality is very situational and can mean a lot of different things in different contexts. In the context of a software engineer and software steward, some applications of quality to me and how they have translated to my app development would include:

1. Removing code smells: ensuring that I keep things DRY and refactoring my code. To separate and encapsulate code is always a good practice.

2. Improving performance: making sure that the app is not laggy or slow by using software design patterns such as MVC, to help make sure that the entire program isn't written in one page and has a logical pattern or flow.
3. Stability: ensuring a quality app means I have to test different versions of operating systems to try and correct any bug that may exist in the code. Which would lead to refinement and an overall more stable application.

These are steps that I have taken to help enhance the quality of my application to help achieve better user experiences.