Adrian Posadas

Alex Lee

Ethan Reidel

CSE 165

5/10/22

Final Project Report

Scheduling App

We were able to develop an app that keeps track of the user's daily schedule and outputs it to an easy to understand UI. We figured that a scheduling app would be best to create simply because we thought we would be able to benefit from using it in our daily lives. We gained inspiration from the reminders and calendar app that Apple and Android provides us and our main goal was to create a hybrid between the two apps but cater towards a daily schedule as opposed to a monthly calendar.

In this final project we created a simple scheduling app using CSS for styling and C++ for functionality. We decided to go with QT Creator due to its simplicity when designing user interfaces and it allowed us to implement C++. Before coding the app we took an efficient approach. This app started off as a Figma diagram mock up to give us direction of how we want the app to look. After multiple versions of designing the UI it was time to start getting our hands dirty and dive into the mainwindow.ui file. We tried our best to replicate the Figma diagrams, but we decided to go the web app route instead of creating a mobile app. Our reasoning was because it would make more

sense to have a scheduling app that populates the user's monitor since it would be easiest to view the user's schedule as opposed to a smaller mobile screen.

We included many features such as a text box for the user to enter whatever task they may have, a time and date widget, multiple buttons (add, back, login, etc.) and an area for the user to scroll through their tasks. We figured that simplicity was the best way to go about the UI. There have been numerous situations where I have used an app that had great functionality however its layout was extremely overwhelming and difficult to use. We countered that by going minimalistic while still maintaining all the features that a good scheduling app would require.

Adrian was in charge of styling the app through the style sheet using CSS. I (Adrian) manually had to access each object's stylesheet within the UI designer to code up the colors, fonts, and layout. I had to refer to the Figma diagrams to match certain colors of objects along with their placement. I also had to set up the UI which was something I found very interesting because designing UI's was something completely new to me along with using QT. I applied my previous knowledge from the labs we did (specifically the calculator app) since the button styling was somewhat similar. One major issue that I encountered was using QT's free version of the app which prevented me from having a dedicated CSS styling sheet. However, I was able to adapt to the issue and applied the styling directly through the UI editor. The entire styling process was something that I thoroughly enjoyed because I was able to bring visualizations to life and cater towards something that we found to be visually pleasing.

Alex created the original Figma UI and button functionalities. I (Alex) created the add function to append items to the list of tasks and the remove completed tasks when

the user clicked on them from the list. Along with the appending of items, I implemented the date/time creation of each task that the user would define. I was also responsible for the button connections of the entire app. The “backend” aspect of the app was done in C++. One problem that I encountered would be deleting all selected items from the list at once as opposed to deleting objects individually. We decided to stick with single deletion due to how our UI layout was made and to reduce user redundancy of checking off items when they could just click the task and delete it manually. I made the starting UML diagram of the project through multiple mock ups of how the app would look with different color schemes, fonts, and layouts. This was made possible by using Figma and showing my group members a visual of what the app should look like which would give us all a sense of direction.

Ethan was responsible for bridging multiple UI windows which consisted of the main window and the sign in window. I (Ethan) worked specifically on connecting multiple UI windows and matching the styling of both UI’s. This was possible by using multiple .ui files and another header file in the project folder. One issue that I encountered was trying to connect two main windows instead of having one of the windows listed as a dialog box. By doing this it allowed me to successfully have both windows coincide with each other and give the app the “new window” once the user signed in to their account. If the user were to enter an incorrect username and password then a pop up window would appear alerting the user that their credentials are incorrect and if the user entered the correct credentials then it would display a message saying login successful and redirect the user to the app.

Some problems that we have encountered were the connecting of the buttons on QT. We were able to create the buttons through the .ui file however to add functionality was a difficult portion of the project. Another problem that we encountered was getting QT to recognize when the mouse was being used. We had to create multiple functions to detect when the user was hovering over an object, check if the mouse had been clicked and to check when the user released the click button. We used the Youtube links from the labs as a main platform of support to assist us when we got stuck and constantly referenced our previous QT labs. Another problem that came up was the tasks printing in the incorrect place of the UI. We had it printing on the frame instead of the scrolling area at the bottom. We fixed this by having the object point to the scroll area object instead of the frame object.

Throughout the development process we have learned how to properly deploy an app starting from the basics of a UML Diagram all the way to having a working product. We have familiarized ourselves heavily with the importance of communication between all teams for this project. The functionality team had to communicate with the multiple UI team and the design team had to make sure both teams approved of the layout and color scheme.

Overall, we have learned how much really goes into creating an app from scratch along with how important time management/ planning is. This project has taught me how difficult it is to deploy an app along with the amount of quality assurance testing that was involved. Lastly, the entire app development process was a huge learning curve for all of us due to it being our first time developing something like this from scratch but we all managed to contribute equally and created something spectacular!