

## **Introduction - Design Phase**

My primary concern for the touchscreen mirror display was fitting all of the necessary information into the space available. So addressing this was a priority in the sketch phase of the project. Ultimately I decided that much of the required displays could be accessed at the touch of a corresponding button, thus freeing up significant portions of the mirror for, well, *mirroring* and allowing the use of larger font sizes to improve the visibility of the interface.

I also decided early on that I would like to implement lights in the frame of the mirror, on the left and right sides. This is shown in the preliminary design sketch. Our faces are generally less flattering when lit from above or below so I believe this was the best choice for the personal/vanity mirror design I was aiming to create.

I did my best to make the frame look like stainless steel in the sketch, as such would be an ideal material to protect the vulnerable electronics from water and rust in the bathroom.

At this point I looked for a set of fonts to use so that all text in the design could be varied, but remain consistent. I found a nice free set of fonts called SourceCodePro that fit the slick, modern aesthetic nicely.

For the sketch, I first drew out a rough idea by hand, which involved much erasing. Once I was satisfied with the most significant design aspects, I re-created the sketch using Gimp, a free image manipulation software (this is the sketch I have included in the repository).

## **Implementation**

I used Processing to implement the prototype of the design. Each section of the design is created using a class for constructing UI “widgets” of various sizes and colors. By defining these sections early, altering the dimensions and coordinates of UI elements later on was easier.

The flagship feature of my prototype is the “button bar” near the bottom of the mirror. It features 5 fully functional buttons to imitate the interactivity of a touchscreen. When pressed, each button will open a new window in a predefined area of the mirror. This feature allows most information to be hidden away until the user wants to see it. These subsections, as I call them, are the personal event and news feed, a weekly weather forecast, weekly fitness metrics, personal sleep stats, and a power button (was originally going to be additional mirror settings but I ran out of time). The buttons are represented with universally recognizable icons.

The personal events, news, and weather sections all pull data from .csv files, which are included in the repository. The fitness metrics, however, are randomly generated each time the window is opened. This was helpful when debugging the time series plots in the fitness section, and I ended up keeping it that way to demonstrate another method of generating data points. My original plan was to fully implement the sleep stats button, but I saw that it was not required to implement ALL of the health statistics and decided to prioritize other features.

Outside of the buttons, the top  $\frac{1}{3}$  section of the mirror is displayed as long as the mirror is turned on. This section features the time and date (fetched dynamically from the user’s computer), a stopwatch to keep track of the time spent in front of the mirror, and a randomly generated representation of the user’s weight.

## **Conclusion - Advanced Features**

As it stands, I did not explicitly implement any of the advanced features from the project description (other than a rough representation of lights in my sketch). Instead, I prioritized a consistent and clean aesthetic with simple and elegant functionality. The most advanced feature would have to be the buttons, and implementing them in a way that is functional, satisfying and looks good was the greatest challenge of the project.