

MILESTONE 3: HIGH FIDELITY VERTICAL PROTOTYPE

Team P

Members:

- Aseespal Singh Sehgal – sehgalas@myumanitoba.ca
 - Brayden Falloon – falloonb@myumanitoba.ca
 - Eran Efron – efrone@myumanitoba.ca
 - Mostafa Raad – kamalmr@myumanitoba.ca
- 

Contents

Technology overview	2
Design deviations and evolutions	3
Usability Sales Pitch	4
Visibility.....	4
Feedback	4
Constraints.....	4
Consistency	5
Diagnose/recover from errors	5

Technology overview

Our prototype was made using HTML, CSS, and Javascript. Additionally, we used a charting library called CanvasJS in order to display the user's history in graph format for each health metric. Our approach to building this prototype was to each take all the feedback we had received on our previous prototypes, and individually design a page for each metric. We then compared results and, with the additional input of some volunteers, decided on one uniform design to use across all metric pages that took the best qualities of each independently designed page. Once we had come up with a satisfactory design, we were able to switch to a more collaborative model to complete the prototype, in which all team members contributed to connecting, polishing, and debugging each page of the prototype so that it could be completed in a timely manner.

Design deviations and evolutions

The user feedback we got turned out to be very useful and it also reiterated the fact that what the developer might find adequate can be very different from what the end users do. An example of this in our prototype was the medication page. In our original paper prototype, the medication page had various kinds of data displayed to the user to make the page more informative, but this was perceived as a clustered webpage instead of an informative one. To remedy that, we stripped down the page to the most essential information, which led to the page looking much simpler and more user friendly. This also allowed all the pages to maintain a more consistent style, which was missing in the paper prototype. Another change was the addition of a goals page. The paper prototype was organized such that the different health goals could be set only on their respective webpages, however, to maintain consistency, we added a goals page where the user can set all their health goals in one place rather than by jumping around from page to page. We also switched to having a button to show trend graphs instead of an arrow to indicate that the user should swipe the screen. This was done because the users found the arrow confusing and didn't know whether to swipe up or down, and what would happen when they swiped. The graphs were also changed from scatterplots, which intimidated some users, to histograms, which makes far more sense for tracking daily goals. We also made the act of completing a goal more satisfying by adding a green border to the icon in the menu, as well as making the incomplete task indicators explode into a firework once that goal is complete. All that being said, negative feedback isn't the only useful feedback, and we received a lot of positive comments on the simple clean presentation that didn't overwhelm with information, giving the team confidence that there may be a market for this app.

Usability Sales Pitch

Visibility

We tried to focus on visibility by making the clickable areas of the screens stand out against the rest of the window, making use of high contrast colour combinations to do so. For example, on our main gradient page we used red dots on top of a blue/green field in order to draw the user's attention and prompt investigation. In all our other pages we used a simple black and white colour scheme with large black buttons that ensure users will have no difficulty reading or finding where they should interact.

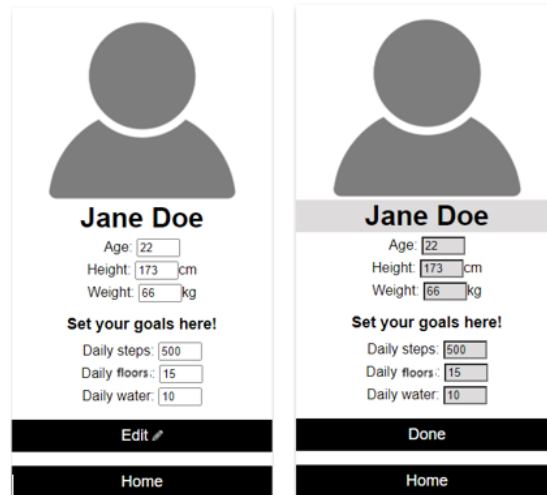


Figure 1: User feedback when editing a profile

Feedback

As shown in figure 1, we focused on providing feedback to the user by changing the pointer symbol whenever a user hovers over a clickable object and by showing a blinking cursor whenever a user can type something. When editing the goals page, we also change the colour of the editable fields to let the user know that the fields are now editable.

Constraints

We have put in a few constraints in place to prevent users from making errors. For instance, the user cannot put in an alphabet while putting in their age, height, weight, or their health goals. The application will also prevent inappropriate input such as negative numbers, or numbers that are outside the scope of human possibility for age, height, and weight. However, the user is still given freedom to make their name out of any characters, as some may prefer to use an online persona that might include numbers or symbols. Pages with adjustable values, such as the goals page or the medication tracking page have an editing mode to prevent users from accidentally altering any values when visiting the pages for other reasons.

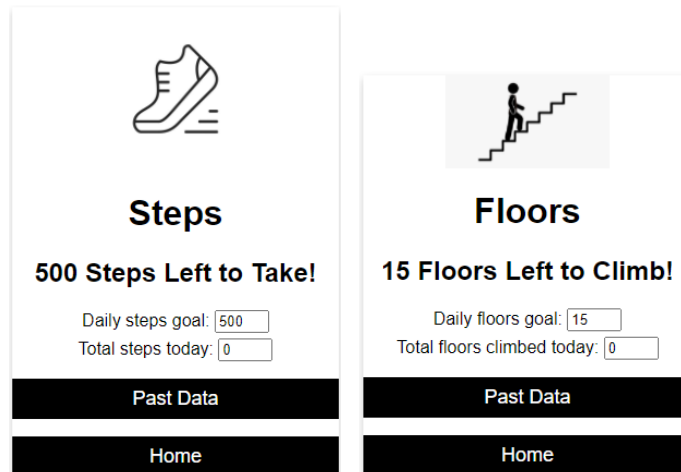


Figure 2: Design consistency between pages

Consistency

We have tried to bring consistency in this prototype by keeping the design of all the different pages within the app similar to one another, a clear example of which is shown in figure 2. This allows users to get comfortable with different screens in the app very easily, aiding with the learnability of the system, as well as ensuring that they know that they are still using the same website. In addition to this, our approach for the project was to assume the possibility of a port to mobile platforms down the road, as the idea fits well with that form factor, thus we used cards for our pages so that a user moving from desktop to mobile would already be comfortable with the new layout, as everything will not only behave the same, but appear the same as well.



Figure 3: Indication to the user what input is causing problems and why

Diagnose/recover from errors

If an incorrect input is given, the user will get a message detailing what was wrong and an indication that they will have to do it again with a different specified input. The system then resets to the default value of zero/null until the user puts in the right value. Resetting the input to 0 may take some control away from the user, but they are given a clear explanation for why their input was rejected and defaulting back to zero ensures that the system is always in a viable state. Further, the user will see the field is populated with a zero, giving yet another indication that their previous input was not accepted, and they need to try setting their information again.

