Smart Healthcare Monitoring System

Project Idea

There has been an increasing use of IoT in the Healthcare field, specifically as a monitoring tool to improve patients' health. Activity trackers, wearable biometric sensors, and real-time tracking devices have become integrated into our lives to monitor our well-being on a daily basis. We developed a healthcare monitoring system that connected to various IoT Healthcare devices and aggregated the data into one comprehensive view. The three devices were a smart scale, fitness watch, and sleep tracker. All of these tracked different metrics for a person, helping them reach their health and fitness goals.

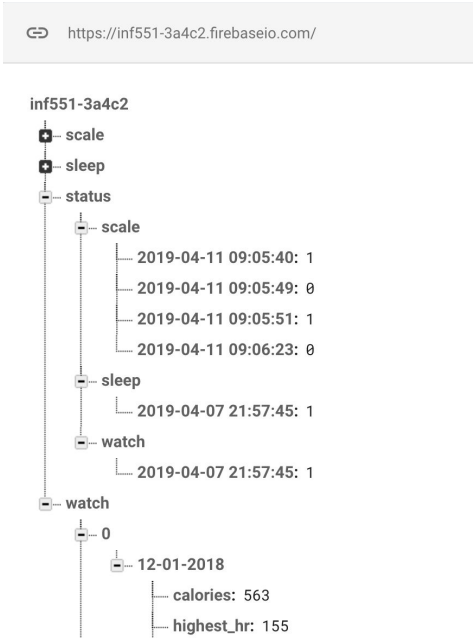
Components

There were three main components in the system that we developed. The Python script that produced the data, Firebase that stored the data, and the user interface that allowed users to control the current state of the devices and get a summary and detailed view of their health data.

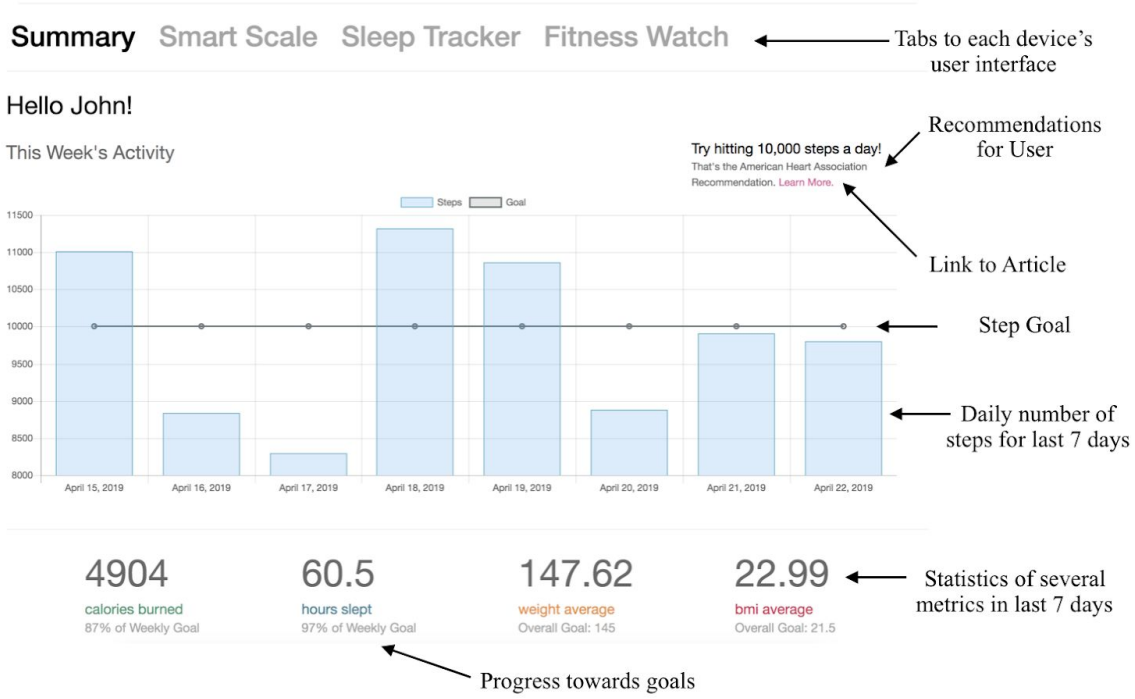
The Python script represented the three devices which produced data on a daily basis from December 1, 2018 to April 22, 2019. The script is titled "device_data.py" in the code submitted, and a screenshot of part of it is shown below.

```
dic = {}
dic['scale'] = []
dic['watch'] = []
dic['sleep'] = []
dic['status'] = {'scale': {'2019-04-07 21:57:45': 1, '2019-04-07 21:57:46':0}, 'watch': {'2019-04-07 21

for day in dates:
    day = day.replace('/','-')
    date_dic = {day: {}}

    # Smart Scale

    weight = round(random.uniform(145,150),1) # in pounds
    height = 68 # in inches
    bmi = round((weight * 0.45) / (height * 0.025) ** 2,2) # percentage
    body_fat = round((1.39 * bmi) + (0.16 * 24) - (10.34 * 1) - 9,3) # Equation for body fat

    date_dic[day]['weight'] = weight
    date_dic[day]['height'] = height
    date_dic[day]['bmi'] = bmi
    date_dic[day]['body fat'] = body_fat

    dic['scale'].append(date_dic)

    # Sleep
    date_dic = {day: {}}

    hours_slept = round(random.uniform(6,9),1)
    avg_temp = round(random.uniform(97.0,99.5),1)
    avg_breathing = random.randint(14,18)

    date_dic[day]['hours'] = hours_slept
```

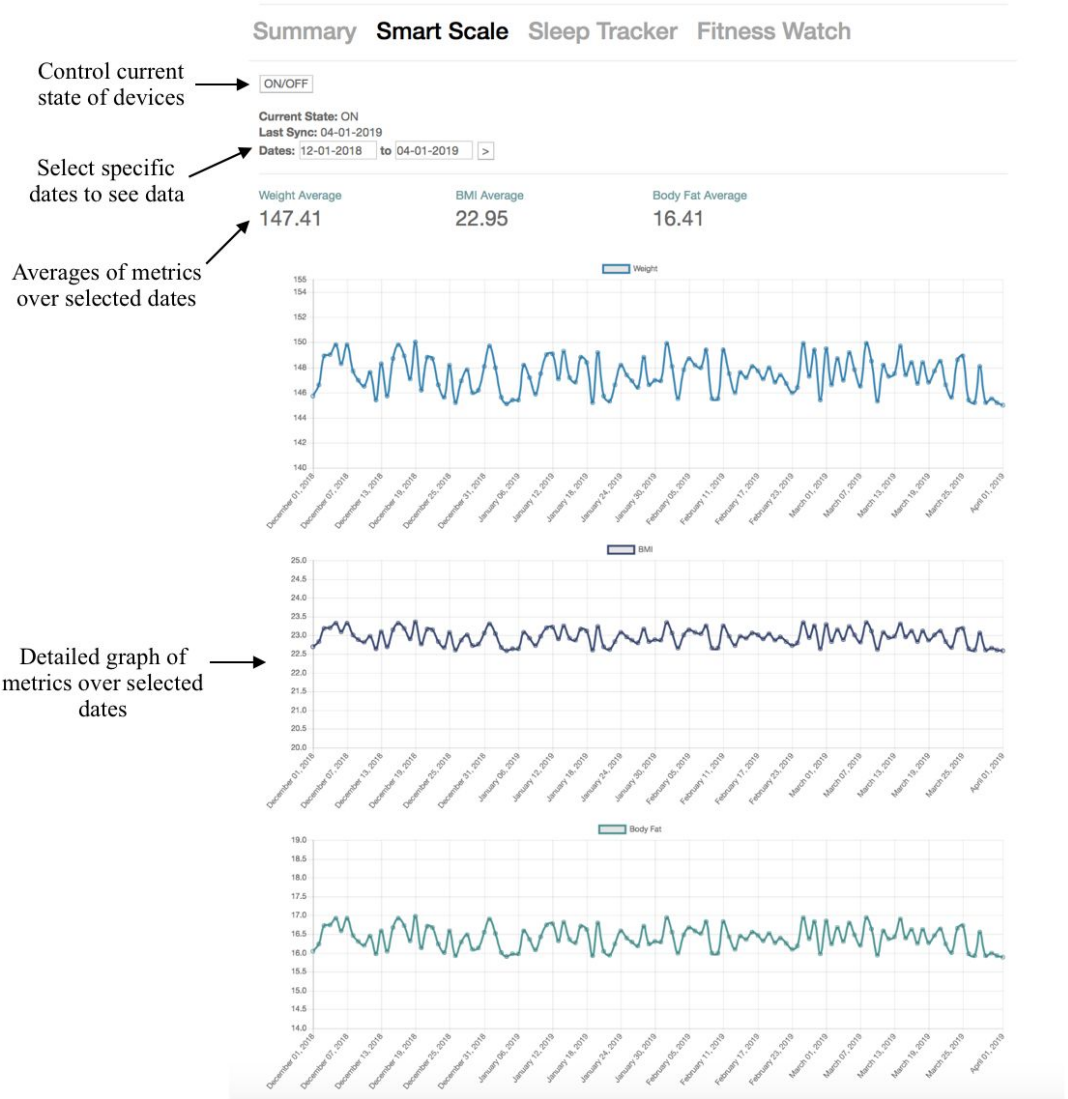
Firebase was the cloud database that stored the data. The link to the database is: https://inf551-3a4c2.firebaseio.com/ . More details regarding the design and layout of the database is included in the implementation section. Part of the database is shown to the left.



The Summary page is the first page the user will see when they open the Web App. It provides a summary of the user's health in the last 7 days by showing statistics of several metrics, progress towards their goals, recommendations, and tabs to each device's user interface. All of the data on this page is pulled from Firebase.

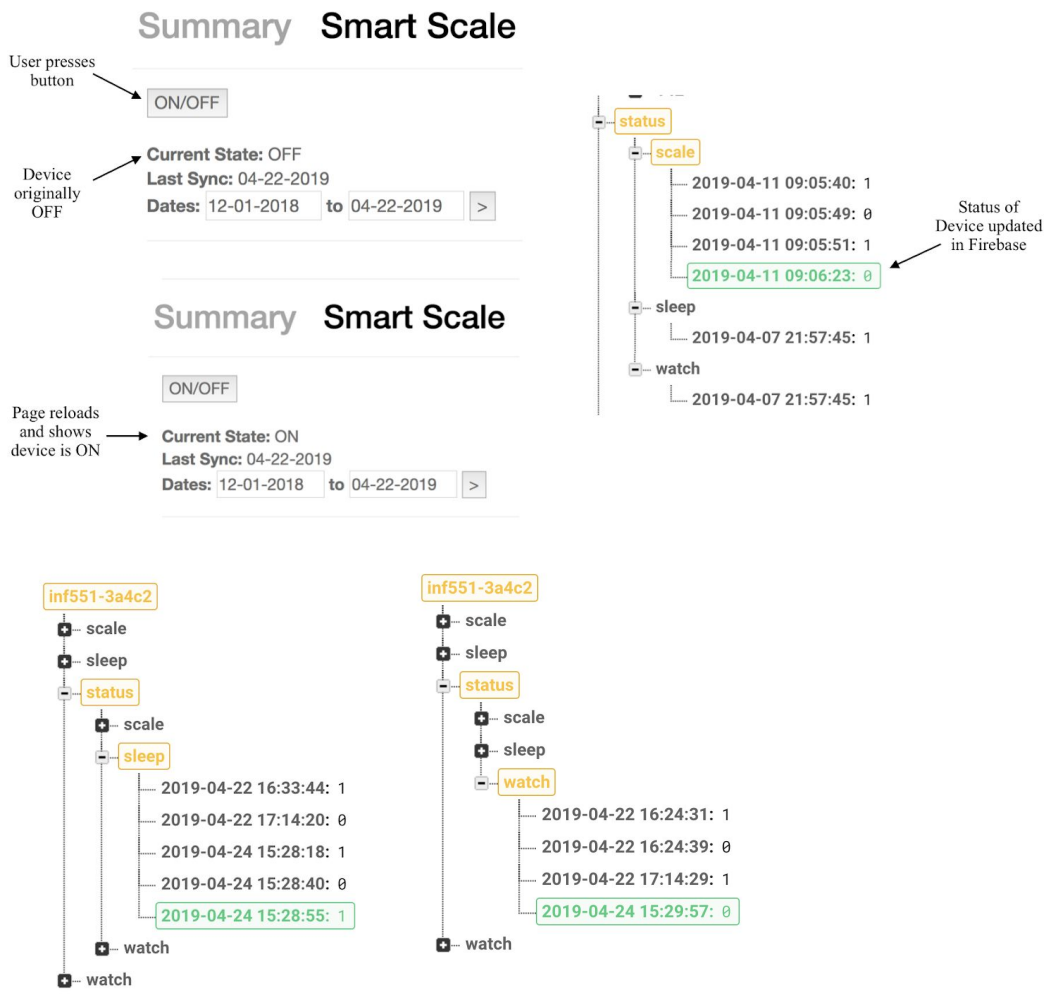


Each of the device's user interfaces allows the user to see and control the current state of the device, view the last sync date, and select specific dates to see statistics and graphs of their health data. All of the data on each device's interface is pulled from Firebase. The user interface for the Smart Scale is shown below. Last sync is when there was last data from the devices, while current state refers to if the device is currently on or off.



The user interfaces for Sleep Tracker and Fitness Watch are the same layout and functionalities as the one above, except the metrics are different. For Sleep Tracker, it has hours slept, breathing rate, and temperature. For Fitness Watch, it has steps, calories burned, and highest heart rate. Screenshots of these two web pages are included in the appendix.

As stated earlier, the user can turn each device on and off. The status of each device is stored in Firebase with a timestamp of when the device was on and off. The current state displayed pulls the most recent status of the device from Firebase. When the user presses the ON/OFF button, a timestamp with the updated status is inputted into Firebase and the page refreshes now showing the new state. If the status is 0, it means the device is off, while if the status is 1, it means the device is on.

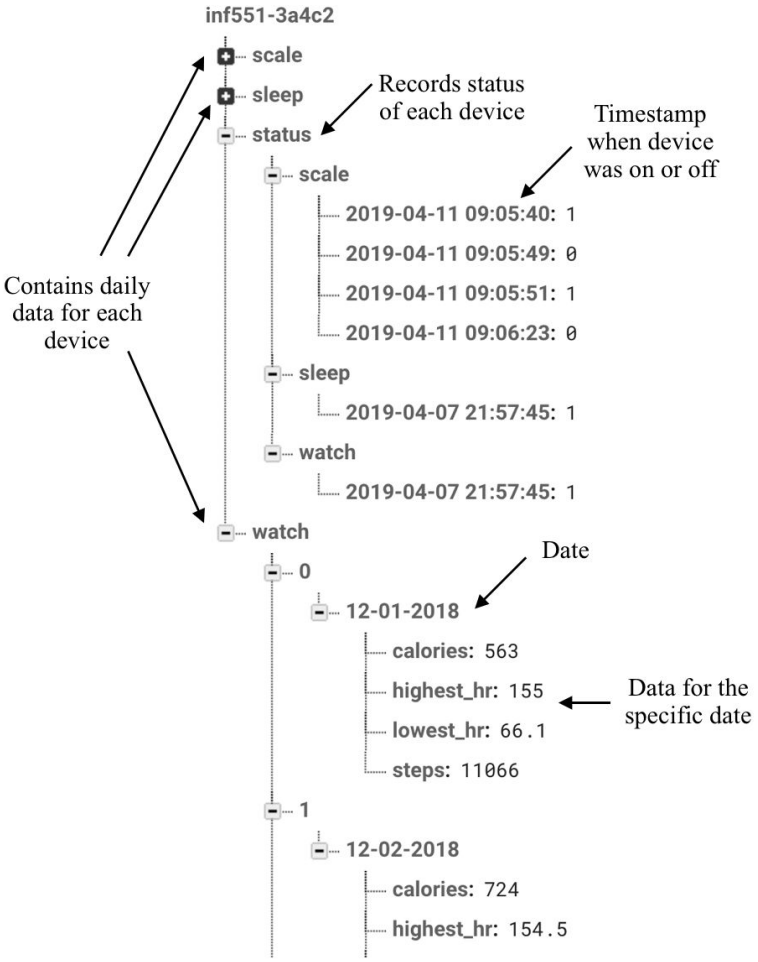


Above shows that a new timestamp and status will be inputted into Firebase database when the ON/OFF button on the Sleep Tracker and Fitness Watch user interface is pressed.

By default, data for all dates will be displayed. However, users can also select specific dates which they would like to see their health data for. The user interfaces will then refresh and display the updated averages and graphs for data within that time frame.

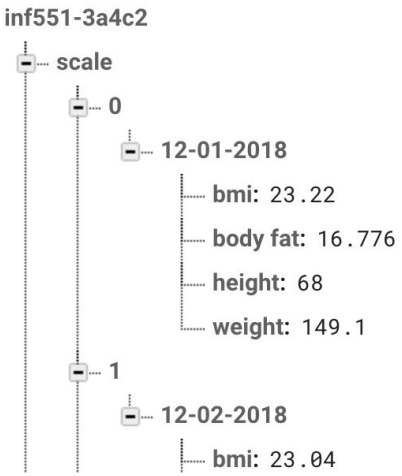
Implementation Details

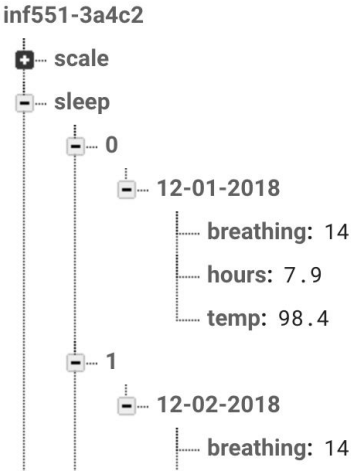
Initially, we planned to create device simulations to use IBM Watson IoT Platform to generate data. However, due to limitations, such as unable to produce random float numbers, we decided to develop a Python script that would generate around 6 months of data from December 1, 2018 to April 22, 2019. Different data metrics were collected from smart scale, fitness watch, and sleep tracker for each day. The collected data was saved into a JSON file then imported into a Firebase Cloud Database.



As seen to the left, the overall database is split into four main nodes. One for each of the devices and one for the status of these device. The status node is contains a status dictionary for each device. Each dictionary contains the timestamps of when the device was on and off.

The layout for the device nodes are the same for each device. The node for Watch is shown to the left. The node contains a list of dictionaries. Each dictionary key is a date and value is a dictionary of metrics and it's associated values for that date. The metrics stored are: weight, BMI, and body fat for smart scale, hours slept, breathing rate, and temperature for sleep tracker, and calories, highest heart rate, lowest heart rate, and steps for fitness watch. The nodes for Smart Scale and Sleep Tracker are shown below.





We then used the data to create mockups of our device user interfaces in Tableau to better understand how we wanted them to look like. Here are the links to the Tableaus:

- **SleepTracker Dashboard:** https://public.tableau.com/views/SleepTracker_15538471748170/SleepTracker?:embed=y&:display_count=yes&publish=yes
- **Fitness Watch Dashboard:** https://public.tableau.com/views/FitnessWatch_15538478781570/FitnessWatch?:embed=y&:display_count=yes
- **Smart Scale Dashboard:** https://public.tableau.com/views/SmartScale/SmartScale?:embed =y&:display_count=yes

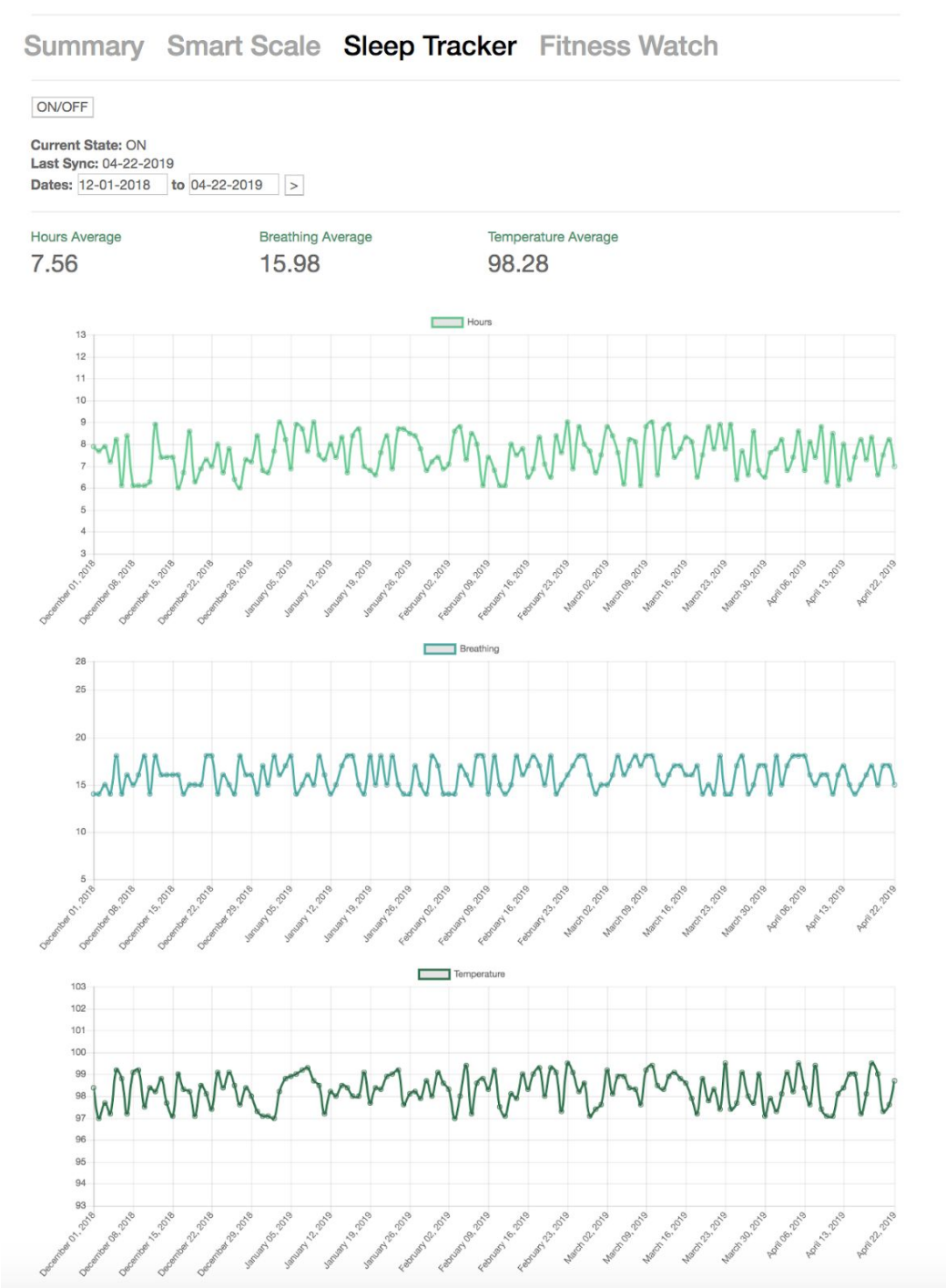
We initially started to develop our user interfaces using Javascript, but found it to be challenging since both of us did not have prior experience with this language and with web development. Therefore, we ended up deciding to use Django, a Python-based open-source web framework, because we already knew how to program in Python. All the code for the user interface can be found in the code submitted under “mysite” folder. We started by developing the Smart Scale device user interface. We worked on figuring out how to read and write data to and from Firebase through Django. Once we had successfully connected to the database and could interact with it, we decided to utilize chart.js, an open source Javascript library, that allows developers to easily plot data in web applications. There were a lot of tutorials on how to integrate chart.js into Django which helped us implement this library. We created the three plots for each metric over time and the average of those metrics before adding in the feature that allowed the users to select specific dates to view data. One issue that we ran into was that our web pages were taking a very long time to load. To fix this, we ended up redesigning our database to how it is shown above to reduce the amount of times necessary to get data from Firebase. In addition, we did most of the data manipulation within our Django Python script rather than retrieving data from the database multiple times. Next, we worked on allowing users to control the state of the devices from the user interfaces by adding a button to turn the device on and off. This button writes to the database by updating the current state of the device. It was relatively easy to input data to

Firebase from the user interface, but the challenging part was having the web page automatically refresh such that the current state of device shown would be updated.

After the Smart Scale user interface was complete, we replicated the user interface for Sleep Tracker and Fitness Watch, but changed what data it got and wrote from/to Firebase based on the metrics and status of each device. We also added a header portion with tabs that allowed users to navigate between the different user interfaces. We checked the functionality on each page to make sure that each device's user interface was able to interact successfully with Firebase. Our last step was to develop the Summary page. This page was relatively easy to implement since it was utilizing similar code as the previous user interfaces, but aggregating all the metrics together in a summarized view. We added a few features, such as recommendations, progress towards goal, and welcome sign, so that it would be similar to other health summary dashboards that are currently on the market.

Appendix

Sleep Tracker User Interface



Fitness Watch User Interface