MIS 284N: PROJECT MILESTONE 5

Team Members: Vikram Gupta, Yeggi Lee, Chetna Singhal

1. **Challenges faced in each step**
    a. **Step 1 -** None (had a functional solution from Milestone 4)
    b. **Step 2** - Extracting the required data points from API response and integrating these data points into a JSON string for transfer to Raspberry Pi. While parsing the API response, Android Studio prevented us from directly using the variable '1h' since it cannot compile variables beginning with numbers. Our initial solution was to send all the data including some parts that we did not need. Eventually, we were able to send only the minimum temperature, maximum temperature, and precipitation level by making a custom object in JSON and then resolving the '1h' issue by converting the resulting JSON string into a map. The final result is sent in the form of a JSON string to the Raspberry Pi.
    c. **Step 3** - Making sure that all the data points required for regression are of float data type since the input received from Android is in string format.
    d. **Step 4** - Parsing through the API response to fetch data points (min temp, max temp, precipitation values (using 3h now instead of 1h)) corresponding to tomorrow's date and then aggregating them using the logic: 1. Minimum of min_temp, Maximum of max_temp, Sum of Precipitation values throughout the day for transfer to Raspberry Pi.
    e. **Step 5** - None. We moved bluetooth advertise URL functionality from a forever loop to button A press leading to controlled uBit communication to Raspberry Pi.
    f. **Step 6** - None. Used simple print statements to display required information on Raspberry Pi terminal.
    g. **Step 7** - Programming timers, alert dialogs, disable buttons features in Kotlin.
    h. **Step 8** -
        i. Using Eddystone beacon to both transmit messages from the microbit to the Raspberry and vice versa.
        ii. Using Eddystone beacon and bluezero microbit at the same time i.e. getting steps from the microbit using Eddystone and sending steps to the microbit using bluezero.microbit package.
        iii. Debugging bluetooth pairing issues between microbit & Raspberry Pi
    i. **Step 9** - None. Used Intent class to launch settings page in Android device to help the user to switch network.

        (startActivity(Intent(Settings.ACTION_WIFI_SETTINGS))

2. **The trade-offs between this device to some actual health app (apart from accuracy & prediction)**
    a. **Limited health metrics tracked / No personalization** - Presently, the microbit can only track step count. No additional fitness data can be transmitted by the

microbit (for e.g. weight, height, heart rate, blood pressure etc.). Hence, our health app is recommending the same number of steps to all individuals irrespective of their physical attributes. For instance, if a user is obese and their goal is to lose weight, then a standard step count recommendation by our app might not be effective.

b. **Minimal analytics** - Additionally, the microbit is also not capturing other performance metrics like distance covered, time taken etc. Hence, our app is unable to give any feedback to the user regarding improvement in performance over a period of time. Moreover our app is not storing the user step count in any database so no meaningful analytics can be shared with the user for e.g. spread of step count during the day, average steps walked during the week/month/year, performance comparisons week-on-week/month-on-month/year-on-year etc.

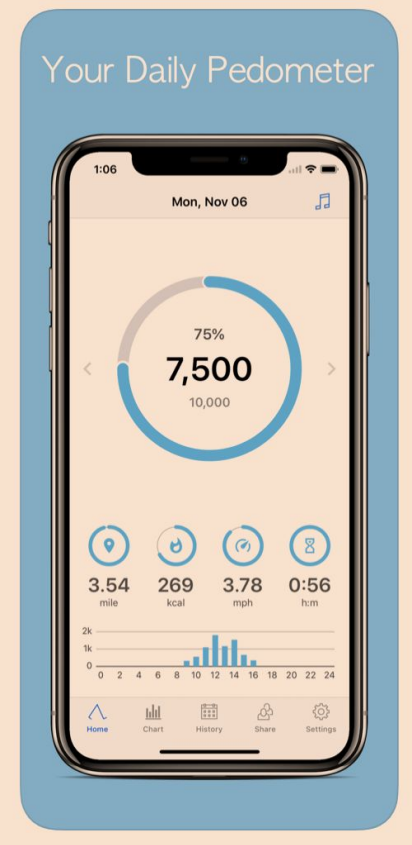

Figure: Expected analytics from a basic pedometer app

c. **Fragmented User Experience** - App cannot connect to the microbit (via Raspberry Pi network) and the Internet at the same time. The user has to manually switch networks to gather information from the Internet and pass it to the Raspberry Pi to display the recommendation on the microbit.

d. **Limited Operational Range** - To connect to the broker (hosted on Raspberry Pi) both the clients (microbit and Android app) have to be physically present in the

limited operational range of Raspberry Pi. That means the user has to carry the microbit, Raspberry Pi and the Android device together at all times to get real-time recommendations.

3. **Thoughts on how to commercialize it, knowing the disadvantages**
    a. Sell it as the cheapest pedometer in the market; Target students, beginners or sceptics (minimal investment)
    b. Deliberately market it as a low-tech pedometer which is best suited for senior citizens
    c. No concerns regarding confidential data - Since this pedometer is not tracking your location or any other vital biometric data or storing your walk history in any database, it can be marketed it as a low-risk pedometer in the digital age.
    d. Sell it as a starter kit to individuals/students interested in self-learning IoT and building small IoT projects.

4. **One advantage & one disadvantage of our prediction model**
    a. **Advantage:** It uses the three most important features (Min temp, max temp, precipitation levels) from the OpenWeather API feature set to give a generalized prediction of expected step count for the user.
    b. **Disadvantage:** All parameters have been given positive weights when it comes to step count prediction. Ideally, precipitation and max temperature value should have a negative impact on predicted step count i.e. if the max temp or rainfall is higher than a certain threshold or min temp is lower than a certain threshold, lower step count goal should be recommended to the user for that day.

<demo instructions included on next page>

5. Instructions

a. Send your pedometer step count for evaluation by pressing button B on the microbit.
b. Check if the Android phone is connected to the Internet.
c. Open the app named “SampleMQTTAndroidApp” in Android phone.
d. Press “RETRIEVE WEATHER” button.
e. Current weather temp will be displayed on the app page.
f. You will next receive an alert for connecting the Android device to the Raspberry Pi. On pressing “Understood” you will be navigated to the Wi-Fi settings page wherein you have to connect to IOT-MIS-21 (Raspberry Pi) network.
g. On successful connection, press the back button located on the bottom left on the phone screen.
h. You are now redirected to the homepage of the “SampleMQTTAndroidApp” app. Press “CHECK GOAL PROGRESS” button to evaluate your step count performance for the current day. If you receive the message “Work harder”, it implies that you walked less steps than your daily goal and if you receive the message “Goal achieved, keep going!”, it implies that you completed or exceeded the daily step count goal as calculated by our prediction model.