

# Team - Group 4

*Rathi Kannan rk27867*

*Gowtami Khambhampati gk6952*

*Danqing Wang dw33369*

*Shreya Bakshi sb59344*

## (1) challenges you faced in each step

- **Step 1 - 3:**
  - No issues in implementation
- **Step 4:**
  - We were able to call the API to fetch the forecasted weather but when we printed the forecast date, depending on the time of the day we ran the program, the date was not reflecting the following day's date all the time. This was because the forecast API was returning 7 days of forecast split for every three hours. So, we changed the Kotlin code to include date logic to return the forecast for the right date.
- **Step 5:**
  - No issues in implementation
- **Step 6:**
  - Weights based on the dataset from entire class gave us very high numbers for predicted steps. Our step counter, was originally set up to increment by 1, so we were never able to predict the situation for goal achieved. We mocked the data by changing microbit code to increment by thousands to test both conditions (work harder/Goal achieved).
- **Step 7:**
  - This was a fun step to integrate UI with ease of usability of devices. We did not face any particular challenge here.
- **Step 8:**
  - We had trouble using bluetoothctl to send messages to Microbit. Although, we were able to discover the Microbit programmatically, the message transfer was not successful. Serial connection between Pi and Microbit helped resolve this issue.
- **Step 9:**
  - Misunderstood this requirement initially to perhaps hardcode and switch the ip addresses within the Android code. Once we understood that we had to only launch the wireless settings page to change the network, implementation was doable.

### Common challenges that we faced in all the steps

- **Connectivity issue to Pi Wifi**
  - One of the challenges we faced was maintaining a stable connection to Raspberry Pi using IOT wifi. We had to switch the wifi multiple times back and forth to maintain the connectivity and it required a lot of manual effort and was time consuming.
- **Connectivity issue to MQTT broker**
  - MQTT broker would often get disconnected and we would not be able to publish or subscribe any messages to and from Pi and the error message -"broker not accessible" would get displayed . We had to manually restart the Android phone multiple times to establish connection to the broker.
- **Time lag between Pi and Microbit**
  - We established a reliable communication from microbit to Pi, only upon pressing button A step count will get displayed on the Pi . However , often there would be a lag between microbit and Pi and though step counts would change on the microbit but it would take some time to display the updated step counts on the Pi terminal.

## (2) the trade-offs between this device to some actual health app (apart from accuracy and prediction)

- We are employing multiple devices in our custom Stepcount app. An actual health App uses one device to aggregate the jobs of Microbit/Pi and Android App.
- Applying gestures/ accelerometer to increment step count is an involved process and the sync doesnt not appear immediately. The transfer of information appears slow.

## (3) your (brief) thoughts on, if you were to sell this product, knowing the disadvantages, how would you possibly commercialize it;

- Given that the sync across devices is slow, we would perhaps add additional features to play music or display engaging advertisements to keep the user enthralled/ glued to the App.
- The Microbit could be designed in custom encasing and promoted as an add on exciting addition to footwear, like a clip on - similar to the colorful Crocs pins.

## (4) at least one advantage and disadvantage of our prediction model?

- **Advantage**
  - Prediction model was easy to understand and implement.
- **Disadvantage**
  - Testing the model based off the actual steps from our Microbit was challenging. For Milestone5, the weights were based on the class step count so the numbers were generally higher and so actual steps had to be manipulated to jump by 1000 to create different test scenarios.

[Pi Github](#)

[Android Github](#)