Social Distancing App - 590U A5

Idea/Motivation

A large population of the public now-a-days have smartphones. Secondly, almost all smartphones of the present day have bluetooth and GPS capabilities. The GPS gives the users' realtime location, whereas the bluetooth can be used to continiuously discover other bluetooth smartphones (which can easily be used as a proxy for other people nearby!) in the near vicinity. Hence, these sensors when used in sync hold immense potential to apply ubiquituous to the problem of social distancing.

Given the above motivation, we can come up with the following idea for the app;

Use the bluetooth to discover new smartphone devices in close vicinity, while outside home, at constant intervals of time (every 10 to 15mins). Combine this with the realtime location data to identify crowded regions on a map and provide this info to the user in realtime. Lastly, utilize the average count of bluetooth devices seen throughout the day as a logical metric for social distancing.

UI/UX Design

The user interface has been explored in detail in the video. I'll just give a short description of the interface and the main buttons here; - There is just one activity/screen in this app for ease of usability. All important details such as "Nearby People", "Tracking Status" and the current distancing score estimate (*which updates on pressing the "My Score" button*); is displayed on the same screen. - The majority of the screen is covered with the Google Maps, which is something most smartphone users now-a-days are comforatble working with. The app has access to the Places API, which allows the users to interact freely with the UI. - The button on the top right of the screen called "Set Home", lets the user set their home coordinates on the map with a green marker. Touching the marker displays the location on top of the marker. - The "Track" button on the bottom left initiates the constant time duty cycle for bluetooth device scanning. It starts sensing for nearby bluetooth devices every 20 seconds (this duration will be kept around 10 or 15mins in a practical scenario). For the purposes of the demo we are scanning for all devices, but in a real world scenario, we would only scan for smartphones. **The code has commented lines of code that just sense for other bluetooth smartphones.** - The "Stop" button in the middle ends the bluetooth scanning cycle. - The "My Score" button updates the user's score metric which is displayed right above the button. A lower score is more desirable, and thats why a lower score appears in green and a high score appears in red. - The UI also displays the number of "Nearby People" below the map. This is using the number of bluetooth smartphones as proxy. This number is updated every 10-15mins, after every bluetooth scan cycle. - Lastly, whenever the user is in a crowded region (with a lot of bluetooth smartphones detected), the map displays a red marker at that location. Touching this marker tells the user about the number of devices that were in his/her vicinity in that location, along with the time of visit.

Scoring Metric Engineering

The scoring metric used is pretty intuitive and straightforward. The metric and its intuition is described below; - The score of a user is nothing but the sum of bluetooth devices detected, divided by the number of duty/scanning cycles. Therefore in this case, a lower score is desireable, because the user wants to be in less crowded places in as many of the duty cycles (i.e. as long a duration) as possible. - A score from 0 to 3 is considered green/safe. Score from 3 to 5 is considered moderate, and finally a score greater than 5 is red/risky. The user can look at the score anytime when outside(and using the tracking) and guage how likely he/she may have been exposed to the virus, and take corrective measures.

Evaluation

- Given the current circumstances, I was only able to go down for a walk to test the app. The walk is recorded in the video.
- All features of the app discussed above worked fine.

Highlighting Contribution

I used the starter code and made the changes I felt were necessary to realize my idea of a distancing app; - UI/UX changes, to keep the UI simple and focused, keeping in mind the privacy of both, the user and the people in the vicinity. - Automating Bluetooth Scanning, at a constant time interval to get a count of new smartphones in the vicinity. - Filtering bluetooth devices, so that only smartphones are counted as proxy for people nearby. - Interactive map markers for crowded regions, which show both the time of day when the user visited that particular location, along with the number of people around at that location. The user can see this data visually on the map, relative to his/her home marker which is shown in green. - Devising a Social-distancing scoring metric and computing it in near-realtime to give as feedback to the user.

Video and Github Links

Github Link

Video Link