Special Topics Networking

Project Report

Social Distance Monitoring using Bluetooth

Under the Guidance of

VP nguyen

By

Keerthana Pyata 1002029148

Harsha Vardhani Kakarla 1002028585

Sai Samhith Praptagiri 1002060952

Introduction

Due to the COVID-19 pandemic, social distance has become the new norm. In order to prevent the spread of the virus, it was advised that people keep a physical distance of six feet from one another. In public spaces, such as offices, schools, hospitals and grocery stores, there were positions designated with a six-foot distance for people to stand in lines. However, in a few locations, there were no such markings, necessitating the use of a program that can determine the distance between two people.

Aim

To create a robust, accurate, and user-friendly application that promotes and monitors social distancing measures in public spaces. We will follow best practices in software development, such as testing and documentation, to ensure a high-quality outcome.

Planning and Designing of the Project

In this project, we used the Kivy framework to create and plan the Social Distance Monitoring mobile application. In order to help users follow social distance rules and lower the risk of virus transmission, the program uses Bluetooth technology to track and estimate the proximity of people in diverse places.

Libraries Used

We have chosen to use the following libraries:

- Kivy - to create the UI and handle touch events
- PyBluez - to access Bluetooth functionality
- Numpy - to handle signal processing and data analysis

Implementation

We planned to implement this application in several different stages

1. Design and develop the UI interface using Kivy and PyCharm IDE.
2. Integrate Bluetooth functionality using PyBluez.
3. Detects and stores Bluetooth signals using Numpy.
4. Process and classify Bluetooth signals to estimate distance and detect proximity violations.
5. Implement filtering, proximity zones, alerts based on the features as directed in the code.
6. Test and debug the application on various devices with bluetooth capabilities.

Features

This application has has many features like

- ➢ Bluetooth Functionality
- ➢ Signal Processing
- ➢ Filtering
- ➢ Alerts
- ➢ User Interface
- ➢ Proximity zones
- ➢ Custom Device Name Filter

Bluetooth Functionality

We have used PyBluez library to implement Bluetooth functionality in the app. The app will detect the bluetooth signals generated by other devices and store them for further processing.

Signal Processing

Numpy library is mainly used for numeric functions, so here for this application also we have used Numpy library to process the bluetooth signals and calculate the accurate distance between the devices carried by the person. Here we have applied a distance estimation algorithm that takes into account factors such as signal strength, noise and interference to improve accuracy.

Filtering

This application will allow the user to filter devices based on several parameters such as signal strength, device type, manufacturer, time, and custom tags or identifiers. This feature will enable the user to focus on relevant devices and reduce noise in the data. We have implemented the filtering system using a combination of PyBluez and Numpy libraries.

Alerts

This application helps the user by alerting the user if they are closer than the recommended safe distance from other devices. We used different types of modes like customizable vibrations, sounds and visual indicators to provide feedback to users. These alerts will be triggered based on the estimated distance between devices and will be adjusted based on the user preferences.

User Interface

We have created a user friendly interface using the Kivy framework and PyCharm IDE. The interface allows the user to interact with the application features such as filtering, alerts and device information. We have used a combination of buttons, sliders and text input to provide a seamless user experience.

Proximity Zones

This application will implement proximity zones like immediate, near and far zones, based on the estimated distance between devices. Users can choose to filter devices within specific proximity zones and the implementation of this feature is by using a combination of PyBluez and numpy libraries.

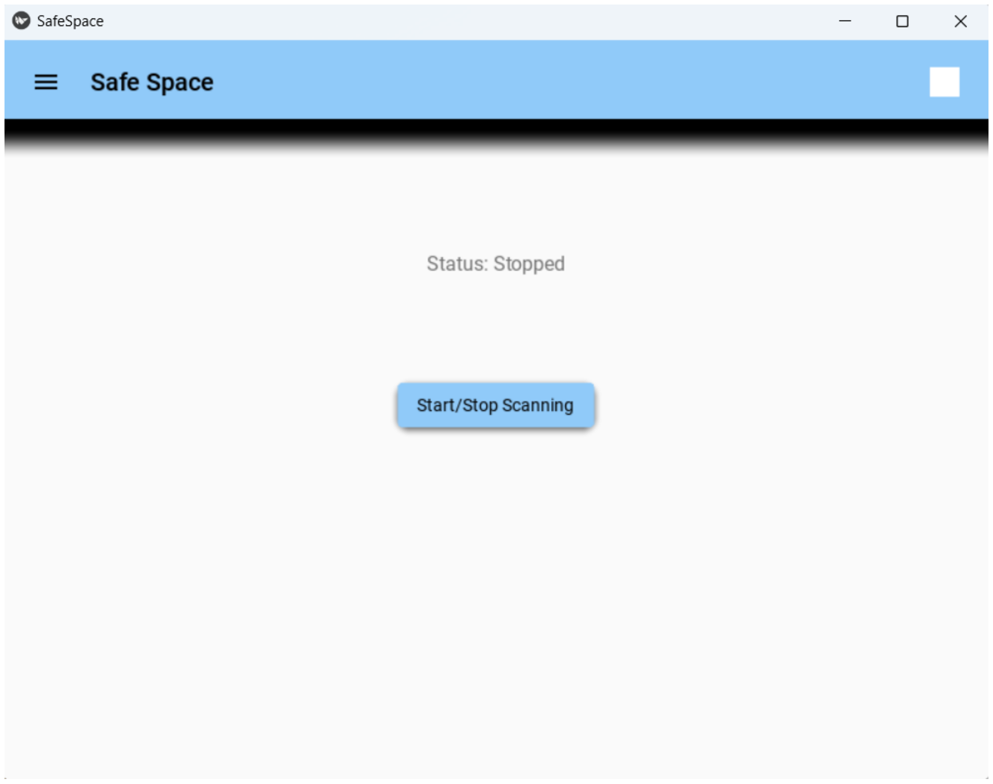
Custom Device Name Filters

We tried to pass a more advanced filtering system by allowing users to customize device name filters in the application user interface This feature enables the user to search for specific devices by name and further adjust their filtering based on the criterias.

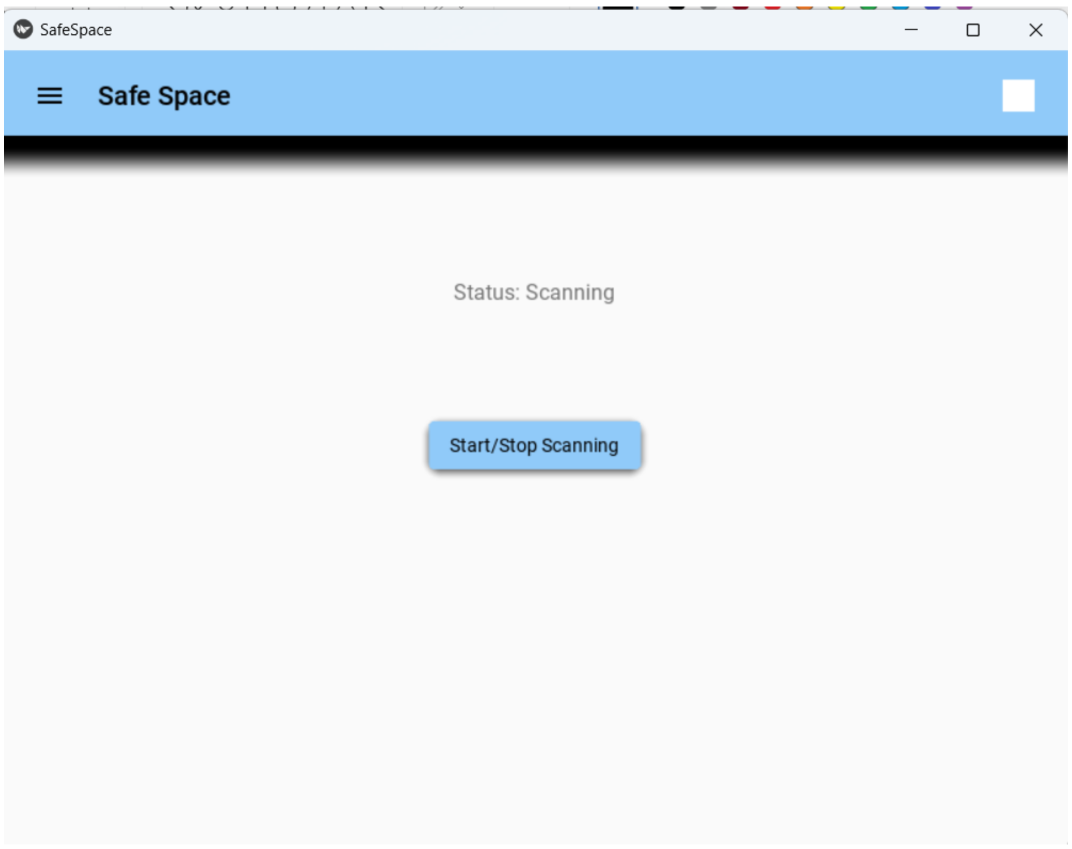
Working

We have started this application firstly by giving it a logo. Yes, we have given a name for this application as SafeSpace.

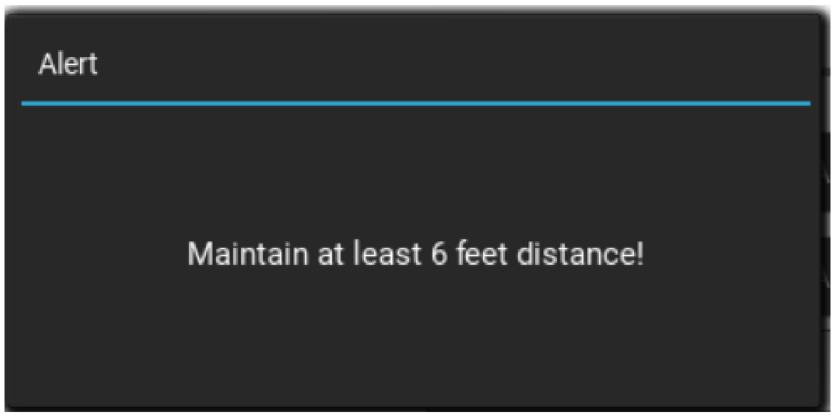
For this application we have used the Python 3.10 version for writing the backend codes and for the front end development we have used Kivy and PyCharm.
As this application does not have any user profiles thus can be downloaded on any device. After the user enters the application they find themselves in an UI application where they can see a button saying Start/Stop



And when the user clicks on that button the application starts scanning the surroundings by taking the bluetooth signals from nearby devices. We have used a single button to stop or to start the application for scanning nearby devices. It uses ‘BleakScanner’ imported from a python package called ‘bleak’ to scan for the devices. In this app, we can set some filters for range of scanning i.e., scan for devices with weak bluetooth signals or for strong bluetooth signals etc. Once the scanner detects any signal it calculates the distance between two devices through the function ‘calculate_distance’.



Once the distance is calculated the data gets compared by the previously stored data imputed where it has Proximity zones like immediate, near and far zones.
If the distance between two devices is less than 6 feet, an alert is sent to the user as a notification.
And that notification or alert can be in a pop up message or a vibration depending on the customer setting as the user has freedom to choose which type of alert they want.
And the user even has the ability to customize the device by selecting which one they want.


Alert

Maintain at least 6 feet distance!

Contributions:

As this is a team project, we have allocated tasks to each person.

Sai Samhith: Is responsible for creating Alert functionality which is to send an alert to the user if the distance is less than 6 feet, has made use of popup and label from 'kivy.uix' package to send the notification , main function in which MDApp and Builder was used to create application window and has made contribution in creating layout

Keerthana Pyata: Is responsible for Bluetooth Scanning functionality which uses packages from requests, asyncio and Bleakscanner libraries to scan for devices and filter them , distance calculation function which uses rssi and tx_power to calculate distance between two devices.

Harsha Vardhani Kakarla: Is responsible for User interface and Safespace_layout.py, has used asyncio function to scan for devices in timed intervals and properties package from kivy to filter devices accordingly and return the distance as immediate, near and far based on the calculated distance , and creating kv file for layout(Filters layout to select filters for scanning), used MDApp from kivy library to create labels and align them.

Lesson Learned from this Project

While developing this project we got to know different types of libraries. Firstly, we have learned a lot on Kivy as it was the first time for all of us to use it. We came up with Kivy for our frontend while doing a lot of research on developing UI indexes using python and got to know that Kivy is an easier version to use.
And even the process for collecting bluetooth signals using Bleaksanner libraries and training them with the data which has previously been trained and calculating the distance. While doing this project we got stuck many times and everytime when we got stuck we again did research till we get a solution and ran the application smoothly.
When developing this application we could learn how to develop an app using Kivy, how we can scan the bluetooth signals and the way alert signals are popped up.

LINKS FOR PROJECT:
GIT: https://github.com/team3stn/Safespace
YOUTUBE: https://www.youtube.com/channel/UCHDZSTfOxeMs_Y_-Kak_Y3g