Indoor Air Monitoring

Android App Utilizing Machine Learning Model and IoT Devices

Alec Borman, jjc171230@utdallas.edu; Hayden Iske, hji170000@utdallas.edu; Dang Truong, dmt180000@utdallas.edu; Jay Castro, jjc171230@utdallas.edu; Angel Torres, axt190026@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

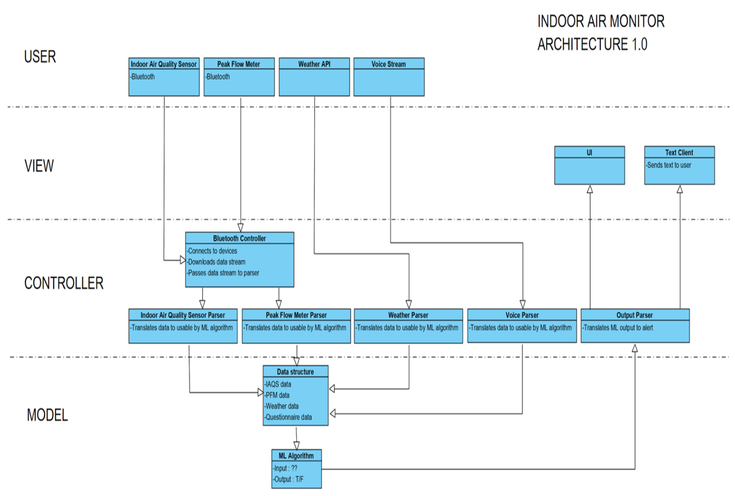
Improved an Android application that takes information from the user's devices, including an indoor air monitoring device and peak flow meter, and local weather data and alerts the user to when he/she will fall sick and need to take medicine using a machine learning model.The machine learning model relies on data from two IoT devices, user-answered questions via mobile app, and local weather data gathered via a web API. The user's data is backed into a cloud-based server for the model to access and then improve based on previously asked questions.

Keywords: Mobile App, Cloud-Based, Machine Learning

Architecture

Core

The core of the app was built using a Model-View-ViewModel (MVVM) interface. With this design, the business model of the application is split from the presentation, maintaining clean separation between application logic.



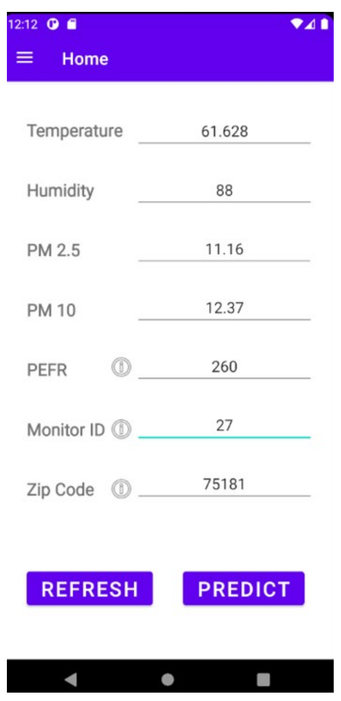
Software

- Android Studio
  - Kotlin / Gradle
- Cloud Server
- Purple Air API
- Amazon AWS
- DynamoDB
- Amazon REST API
- Peak Flow Meter

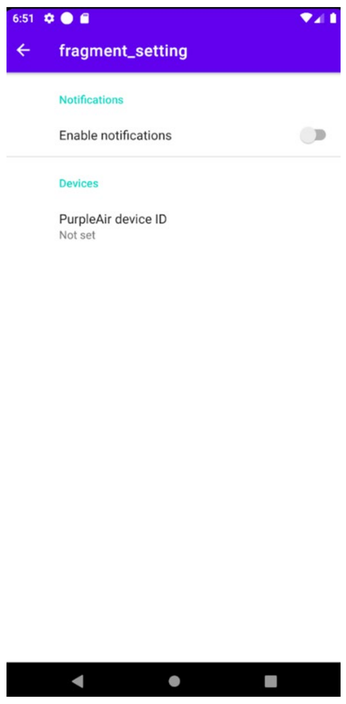
Impact

- User can connect their devices to the app and receive predictions on when to take their medication to prevent health problems
- Model uses predictive learning to continuously improve behavior and become more efficient and handling illness flare-ups
- Full backend implementation and integration with android mobile applications

Results



The user enters values for PEFR, Monitor ID, and Zip Code. Refreshing the app updates the values.



New Settings page where the user can save their Monitor ID to local storage.

Performance

- Polling new data to the app refreshes the app once every 20 seconds
- Objective of updating the app with latest API's and new features was accomplished
- Company mentor feedback: Met the requirements and was pleased with the the new additions and improvements of the app
- Weekly task completion rate: 90 %
  - Some features were not completely implemented

Summary

- Refurbished MVVM to improve design and overall clean architecture for the app which helps developers build onto the app with ease
- Demonstrated a functional UI portion for the app which shows relevant data needed and retrieved from the device and model (PM2.5, PM10, etc.) as well as the questionnaire
- Updated Purple Air API to have a functional device that measures air quality,temperature, and humidity in an environment
- Created AWS Database and REST API for App Use
- Refactored using RetroFit2
- Integrated Machine Learning Model into the app for predictive learning