Six:

Introduction of Members:

* My name is Six Wires, I’m a frontend developer, and working with me I have Chandler Odenweller, Minsoo Mun, Junsik Seo, and Nilay Jasapara.

Overview of the project:

* The project we were assigned was an indoor air quality monitoring app that utilizes a machine learning algorithm to help predict lung failure within users.

Mission given:

* Create an application that uses the given machine learning algorithm to help preemptively predict and reduce the risk of asthma attacks.

Objective:

* Our objective was to update the skeleton application into a fully-fledged application that was user friendly and aesthetically pleasing.

Approach:

* Our team divided into frontend and backend. The frontend developers (me and Chandler) worked on updating the application’s look and feel, adding a few screens along the way. We added the splash screen and updated the settings screen. Our backend team was devoted to connecting the application to the machine learning algorithm and receiving and using the values from the bluetooth devices we’ve implemented.

Time: 2:30

Chandler: (2:04)

* I worked specifically on the final redesign of the main home page of the app and implementation of the settings page. Since this project led on from a previous group, one of the main difficulties faced was taking and learning the app already given by the previous semester’s group and understand it well enough to add on top our own features that do not conflict and instead meld with their features. This process of learning the code base took up a significant portion of the process of implementing our redesigns within the app.

(33 seconds)

* The home page takes inspiration from other weather apps and implements their features in a minimalistic fashion that matches the rest of the theme of the app. The most visually interesting and eye-catching feature of many weather apps is the symbol at the top describing whether it is rainy, sunny, cloudy, or any other weather condition that would correctly describe the day.

(25 seconds)

* However, this simple feature turned out to be one of the most frustrating features to implement. Simply trying to add in the symbol at the top of the home screen depicting what the weather currently was required learning the intricacies around Gradle and Kotlin and their many, many functions that sound and act very similar but are used for different niche purposes. This eventually led to me finally implementing the feature with much frustration but it truly became the key point of the home page of the app.

(33 seconds)

* When implementing the settings page I took most of the inspiration partly from Six’s work on the setup page, and partly from the previous group’s code within their Questionnaire implementation. The setup page uses shared preferences in order to record the zip code and monitor ID code and these same shared preference values is what allows the settings page to successfully change the values. From the previous group’s code base, I was able to implement the design of the settings page within their style of coding so it would mesh well with the rest of the app.

(33 seconds)

Junsik:

I worked on connecting our app to a medical spirometer device and updating the app with the latest API’s. The spirometer device was prepared by the previous group and they used an app called Smartone to get PEFR value. I tried to connect our application via bluetooth to a medical spirometer device directly but I wasn’t able to manage the pre-built device to send a signal to our app or get a bluetooth signal. I checked the smartone app and it generates PDF files after every testing. So I decided to retrieve PDF files from the smartone app to our app. I was able to get PDF and image files from the storage. However, because it was my first time using Kotlin, I couldn't figure out how to retrieve value from the PDF files. I need location or name of files to retrieve value, but it turns out that smartone app only gives files, not a name. So our team decided to enter the PEFR value manually.

(1 minutes 23 seconds)

Nilay:

* My contribution to the project consisted mostly of refactoring and debugging the code. I assisted with debugging the AirMonitor api, ensuring that the device was correctly connected and that we were receiving accurate data. I double-checked that the data was successfully supplied to the ML algorithm and that the generated results were correct. Helped the backend team as needed.
* Because this is my first time working with Android studios, I had to learn by looking up a lot of information online. This project was built in Kotlin, which was also new to me, and learning the basics of the language took a couple of weeks, which slowed down the app's development significantly.

Total time: 1 minutes

Minsoo:

* Hi, my name is Min-Soo Mun and my contribution to the project was mostly testing. I found and fixed bugs in the project and collaborated with our backend team on fixes to those problems when necessary. I did testing with individual values to make sure each value was correct and being read into the devices properly. I am planning to use some testing libraries such as Junit and Mockito to test the app further if possible. I also helped our backend team when needed.

* At first, it was hard for me to work on this project because it was my first time using Android Studio. I didn’t even know how to run the project. However, as I kept working on it with my teammates and learned how to use Android Studio, I started to understand how to read the code and was able to figure out any bugs to fix and improvements in the project.

Total time: 1 min