**Implementation Plan**

**Data Scraper**

Members: Andrei

Overall Purpose/Description: The purpose of the data scraper is to provide a way to obtain data about London air quality and weather. The program would have to connect to various data sources, download all the data for the past 3 years, and store it in our own database used for predictions. It also needs functionality which allows downloading new data which is recorded daily.

Requirements:

Technologies: Java, PostgreSQL

Week 1: Look for data sources which would preferably provide an API. Analyse and decide which of them to get data from.

Week 2: Implement java program which can download hourly air quality and weather data from selected sources. Implement feature for parsing and storing downloaded data into Postgres database. Design and implement database structure.

Week 3: Implement feature which makes the program run daily and get data for the current day (past hours). Implement feature for downloading and storing forecasted weather data.

Week 4: Create automated tests to check if program is reliable, fix bugs. Upload program to server and make it run hourly.

**Data Mining**

Members: Kirthi, Mégane

Overall Purpose/Description: The purpose is to design a prediction analysis model that would enable us to predict the pollution levels periodically. The pollution levels would be determined using the past pollution levels and the weather conditions. The parameters that we use would be:

1. Pollutants - pm2.5, pm10, NO2.
2. Temperature
3. Rain
4. Wind - direction and speed

The regression model along with Principal Component Analysis is likely to be used to achieve the result. The algorithm would predict the pollution levels and update the database, which would then be used by the API.

Requirements:

Technologies:

Week 1: Analyse the different prediction models and methods. Choose the appropriate model, language and technologies that we would use. Coordinate with the research team to know more about the data that would be used for prediction.

Week 2: Design the abstract model and sample a few predictions based on the past pollution and weather conditions (temperature, rain and wind speed).

Week 3: Making the algorithm more efficient and faster and incorporating other parameters like wind direction.

Week 4: Testing and evaluation.

**Front-End**

Members: Sam, Andreas

Overall Purpose/Description:

Requirements:

Technologies:

Week 1: Make a project up, define structure, basic UI, main links, research into methods for REST APIs and GPS in the background

Week 2: Begin work on the API layer and GPS service and the main dashboard/UI

Week 3: Keep on working on the tasks already started in week 2

Week 4: Finishing and polishing + extra features if time permits

**API**

Members: Andrei, Vino

Overall Purpose/Description: The purpose of the API is to provide an interface for the front-end Android application to allow it to fetch/update data. The API will also serve as an interface for our pollution prediction algorithms that we will incorporate into the back-end system. The Android application will be able to fetch this generated data without having to process raw data from the database. In general, the API will allow us to do most of the heavy computation on the server-side and just allow the results of computation to be accessed from the Android device. From the end-users perspective this system architecture would be good on performance as the speed at which data is processed will not be dependent on the type of Android device used. The API functions implemented will include: Authorisation, Feedback, Route index/avg, Route finder, and also

Requirements:

* Functional:
* Non functional:

Technologies: Node.js and PostgreSQL

Week 1: (refer to Implementation Task List)

Week 2: (refer to Implementation Task List)

Week 3: (refer to Implementation Task List)

Week 4: (refer to Implementation Task List)