**Implementation Task List - 05/11/15**

**Android Application**

Sprint 1:

1. Setup login activity and basic UI
2. Setup dashboard activity and basic UI
3. Setup feedback list activity and basic UI
4. Setup historical chart activity and basic UI
5. Setup route recording activity and basic UI
6. Establish activity links (mocked where necessary)

Sprint 2:

1. Construct basic versions of dashboard sections
2. Construct REST API access layer (against API plan)
3. Connect relevant UI actions to API layer
4. Construct route recording/GPS service
5. Implement basic graphing on historical chart screen

Sprint 3:

Plans to improve UI and make it ‘complete’.

Test API layer works against mocked API on the server side.

Sprint 4:

Implement any UI relevant to secondary goals and ‘nice to have’ features.

Final polish of essential UI.

Testing against running API.

Full testing of the application.

**Application Program Interface (API)**

Sprint 1:

1. Setup REST API framework using Node.js
2. Setup connection between Node.js API server and PostgreSQL database
3. Setup API function for user authentication using token-based authentication
4. Setup API function for uploading routes for a specific user and store them in the database for that specific user

Sprint 2:

1. Setup API to return pollution exposure index value for a particular route (where the route identifier would be the request and the index value would be the response)
2. Establish and implement an algorithm to efficiently calculate the pollution exposure index value of a specified route.
3. Setup API function for user feedback which will store feedback data for a specific user
4. Incorporate user feedback score as a factor into the algorithm which calculates the pollution exposure index value for a particular user
5. Setup API function to return the historic exposure levels for a particular route

Sprint 3:

Start testing API with real data from front-end Android application

Update API to serve requests using the same data format as the front-end Android app

Extensively test API with all sorts of inputs

Sprint 4:

Implement functionality for least-polluted route finder (and any other ‘nice to have’ features)

Test route-finder functionality with front-end application

Full testing of API with front-end application

**Data Mining/Predictions**

Sprint 1:

1. Testing data mining methods
2. Choose appropriate model

Sprint 2:

1. Forecast weather data scraping implementation
2. Implementation of calculation and storage of predicted pollutant levels

Sprint 3:

1. Improve algorithm efficiency and accuracy
2. Incorporation of other relevant variables into the model

Sprint 4:

1. Final testing and improvements