Task 1

The data collected on my sensor (Android application) was the current GPS/network location and the sound intensity. The data is collected by the phone's sensors and it is parsed to a CSV format. The CSV formatted data (sound intensity, latitude, longitude) was sent to an AWS Kinesis Firehose stream on the cloud. The Kinesis stream was connected to an S3 bucket. The stream considers the phones running the application as the producers and the S3 bucket at the end of the stream as a consumer. The data produced by the phones are deposited into the S3 bucket. The kinesis stream allows the system to scale the number of data producers dynamically.

The motivation to collect both sound intensity and GPS data was to get a sense of how badly Dublin was impacted by noise pollution in different areas. If there are many users using the application while walking to work or on lunch break, you could get a rough idea of real-time noise pollution levels in dublin. This system relies on the existence of many active users. The data stored in the cloud was read by a python script running on a computer and the data was used to produce a heatmap. The read areas should show high noise pollution and the bluer areas indicate lower noise pollution. This heatmap is stored in a HTML file that is updated periodically, displaying near real-time data. In theory the HTML file produced can be uploaded to an Apache web server running on a publicly accessible EC2 instance. This would allow users to view the noise pollution data.

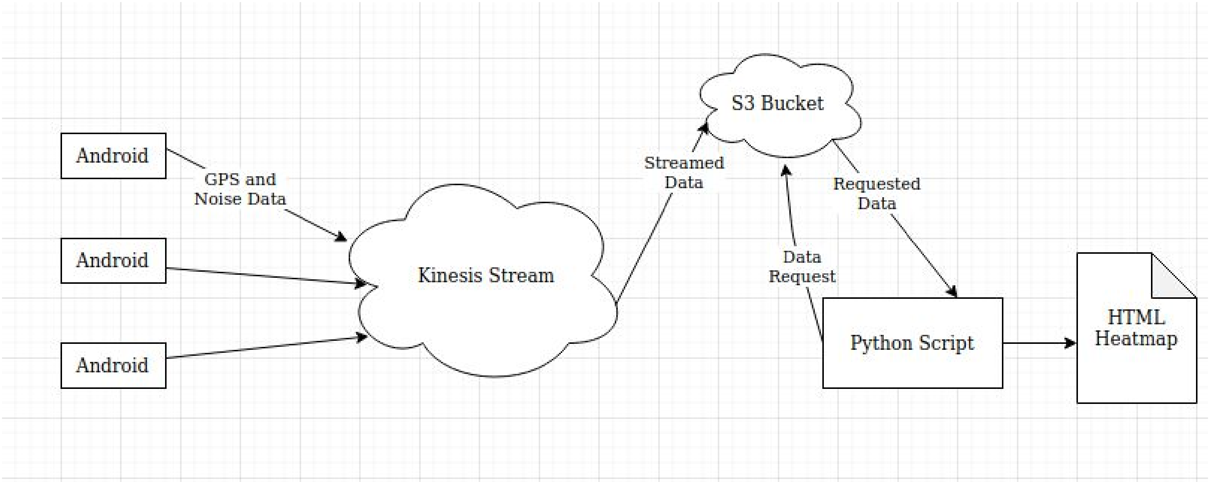

Fig 1.1 This is a technical diagram of the system end to end. This is a centralised architecture.

The Amazon Amplify Android SDK was used to send the data to the cloud. The Android application used in the assignment was written by me using the Android Java framework. The data was read from the cloud using the Boto3 package in Python. The downloaded data was parsed into pandas dataframes. The data frames were used to clean the data and make it workable. The heatmap was made with the Folium package. Folium provides commands to make many types of interactive maps.

Link to Github repository: https://github.com/andrejliadov/Urban_Computing

The branch called Assignment 4 will contain the code for this assignment. I will append some of the relevant code to this report. On the android app, most of the work was done on the SendRunnable class. This implemented and sent runnables to the sender thread.

The use of Kinesis streams allows the system to dynamically scale to many users. Kinesis Firehose is a managed service on the AWS platform. This saves cost and increases availability of the system.

Task 2

The data used for data visualization in this assignment was acquired from an open data source. I would have used the data gathered and stored in task 1. This was not possible since I was not able to generate enough data with my app.

To alleviate this issue I have decided to use the open data source that I found in the third assignment. The data set contains all of the available parking spaces in Amsterdam. It is being updated live and my program downloads the data and stores it on the cloud. The parking data included the GPS coordinates of the parking space. I used the GPS coordinates and the number of parking spaces to create a heatmap of real-time parking availability in Amsterdam.

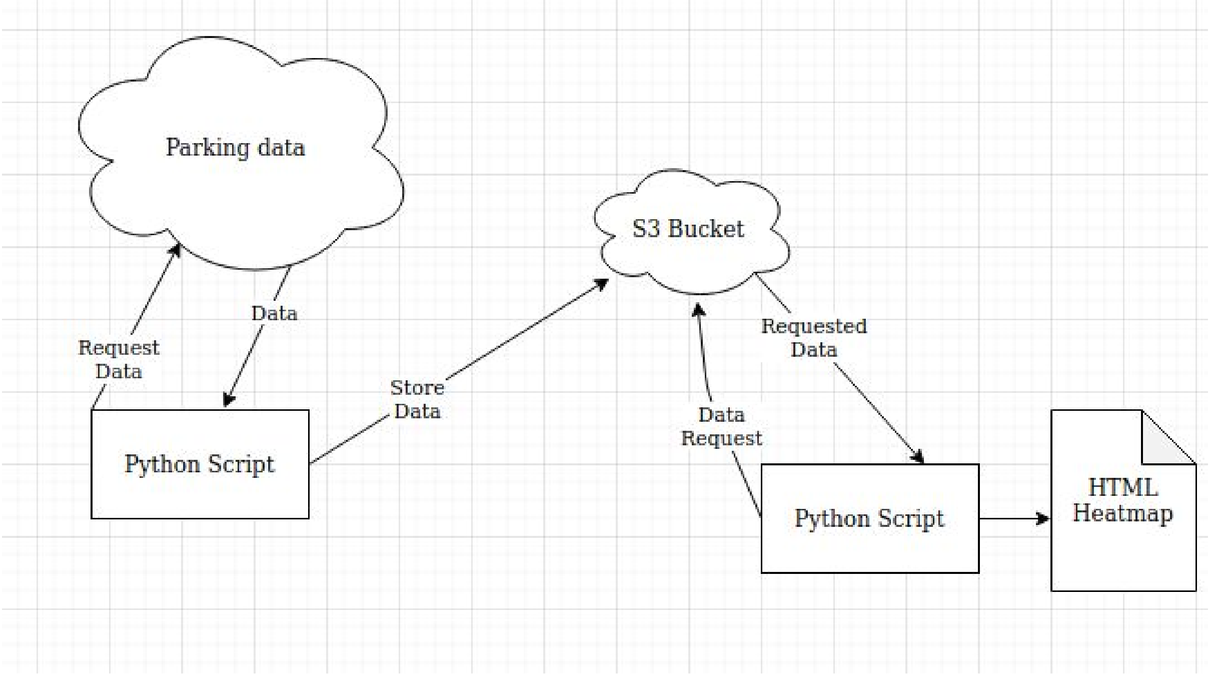

Fig 2.1 This is a technical diagram of the data visualization system. Note that the parking data is requested from a website.

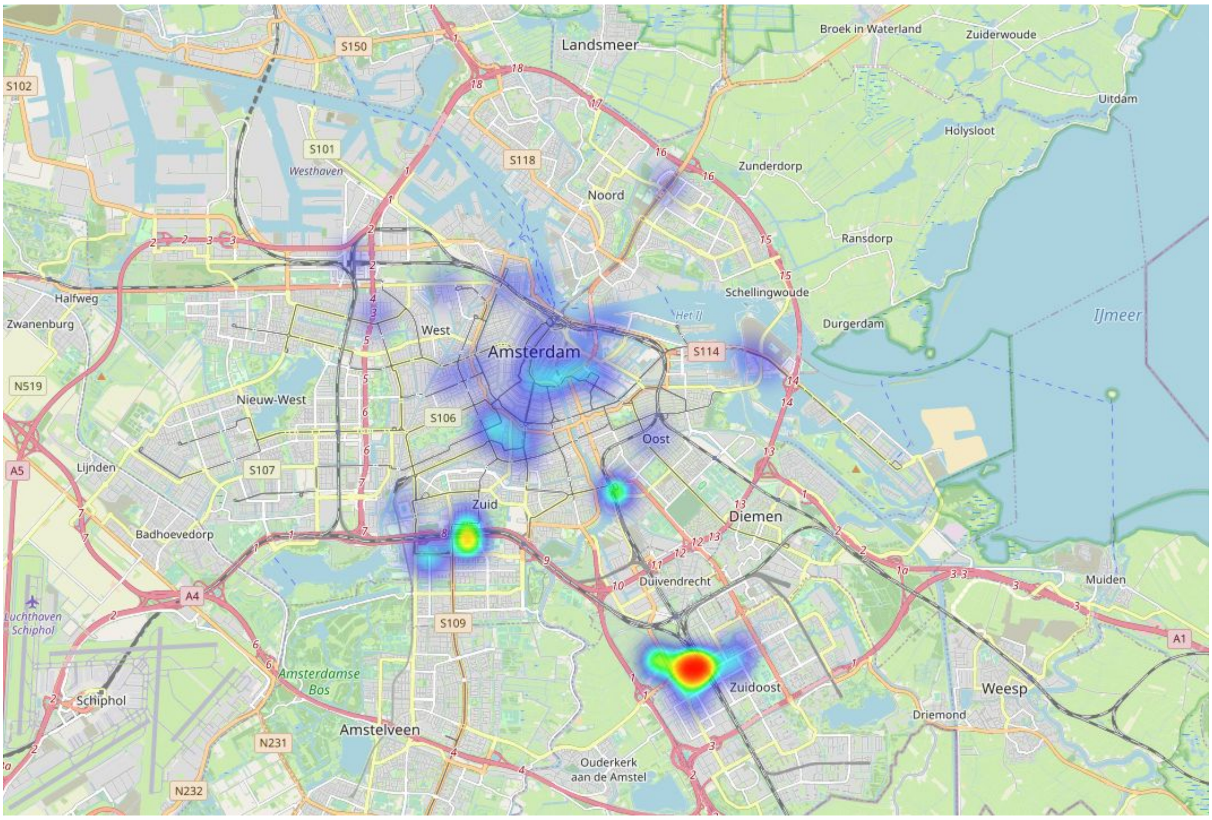

Fig 2.2 This is the heatmap generated with the parking availability data. It is fully interactive and can be used to view the whole world if needed.

Link to data: http://opd.it-t.nl/data/amsterdam/ParkingLocation.json

Task 3

In assignment 3 I was collecting noise data and I was transmitting it to the cloud in a similar fashion to this assignment. In this assignment I decided to add the GPS measurements. In theory this was supposed to work by getting a real-time updating GPS measurement. I added the location manager and location listener systems to my Android application code. I was able to collect the first GPS location but the problem was that it did not update the GPS reading when the position of the mobile phone was changed. Therefore on the S3 bucket I only got one GPS location.

The GPS data would have allowed a way to make a heatmap of noise pollution. I would have looked for an open data source for average income by district in Dublin. I would analyse the correlation between the prosperity of a region and noise pollution of the area in given regions.