Object Oriented Programming - Project

Serving: Tal Adari – 203037007

Aviad Ariel – 204326409

Noya Yaron – 205746720

How the system works:

The system is divided into 2 main classes: CsvProccessor and KMLCreator.

The CsvProccessor class reads csv files which was built by wiggle wifi app from a given folder path, sorts them by time & signal power and outputs the wifi data to a new formatted csv file.

The data for each network that being read from the csv files is saved into 2 classes: “WifiRecord” which includes the time, location and id of the network, and “Wifi” which includes the ssid, mac, signal and frequency.

The CsvProccessor class reads the csv files lines into a list of WifiRecords which is inserting the data sorted by time.

Each WifiRecord in the list has it’s own list of Wifi, which is inserting the data sorted by signal power.

The KMLCreator class reads a csv file formatted as our output from the CsvProccessor class, and outputs the data into a kml file, sorted by either time, id or location of the network.

Using filter’s classes the program is filtering the data and outputs it to a new kml file.

Software:

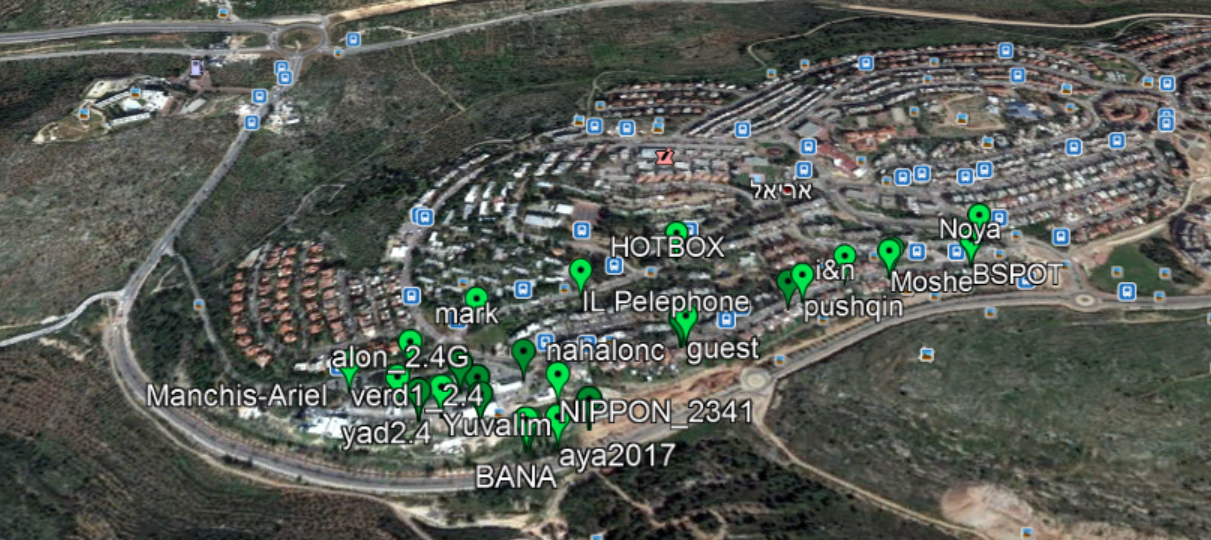
We used Eclipse as our IDE for developing the project, included joda datetime library, junit library.

Unit Testing:

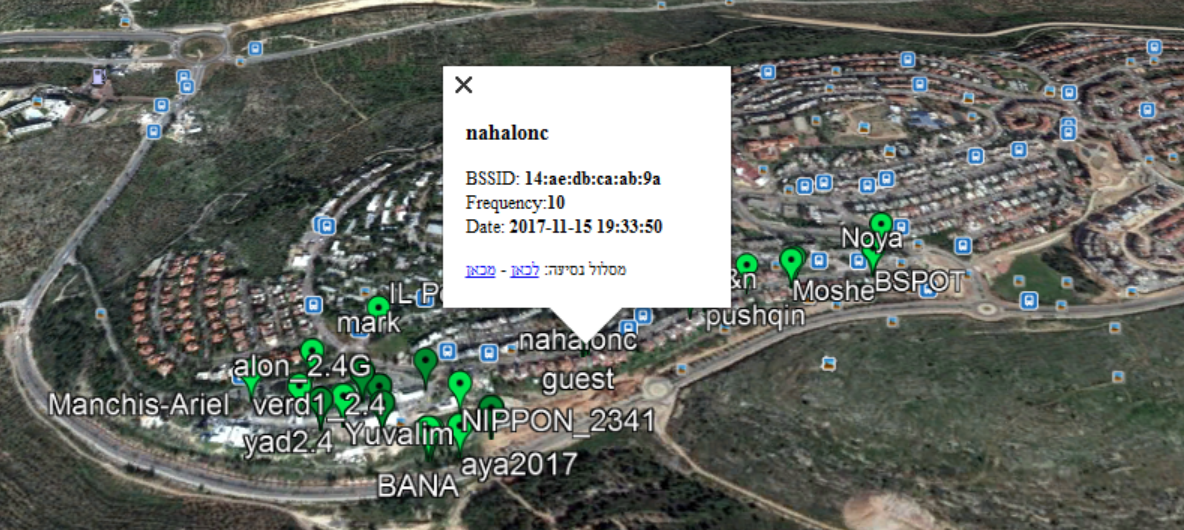
There are folders with test files attached with the project inside the src folder, we created tests for each class, focusing on testing behavior.

Ex 0

We made a test drive with the wiggle wifi app in Ariel.



Data from one of the networks we found



Ex 01

2. <https://github.com/taladari/navproject>

3. Done

4. Done, there are example files in the src folders.

5. We found JAK as the most known KML API for java, though we found it not so usefull, as we created the KML file with our own implementation.

6. We added unit testing classed using the JUnit library, we made tests for each project class.

7. We added timestamp section to our KML file, the timeline view option is working on google earth pro.