**Object Oriented – Project 0**

**Kineret Ruth Nahary - 206903684.**

**Yakir Amar - 204590095.**

The program merges CSV files from exported android app called ***"WiGLE WiFi Wardriving"*** into one file – taking Wi-Fi networks only and arranging them by time and place. For every timestamp it takes the top 10 networks with the strongest signals and arranges them in an ascending order.

***"WiGLE WiFi Wardriving"***  app – it is an open source network observation, positioning and display client from the world's largest queryable database of wireless networks. This app can be used for site survey, security analysis and competition with friends. It collects networks for personal research (information was taken from the android app store).

<https://play.google.com/store/apps/details?id=net.wigle.wigleandroid&hl=en>

The program also takes an arranged file and converts it into a KML file that can be used on the "Google Earth" site. By uploading the file to "Google Earth" site, we can look up all the Wi-Fi networks we wanted to see on the map. We can choose in the program either to filter the network's list by a specific date and hour, by a specific ID (=device), or by choosing a point and a radius to show all the networks inside this specific radius.

"Google Earth" – is a geobrowser that accesses satellite, aerial imagery and other geographic data over the internet to represent the Earth as a three dimensional globe. This product has many features one of them is the ability to show mappable data by reading KML files that had been uploaded to it.

<https://serc.carleton.edu/sp/library/google_earth/what.html>

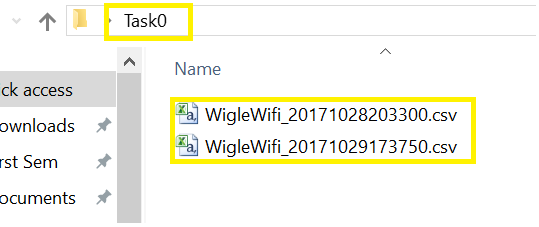
Further look into the program we've built:

**We have so far 5 classes:**

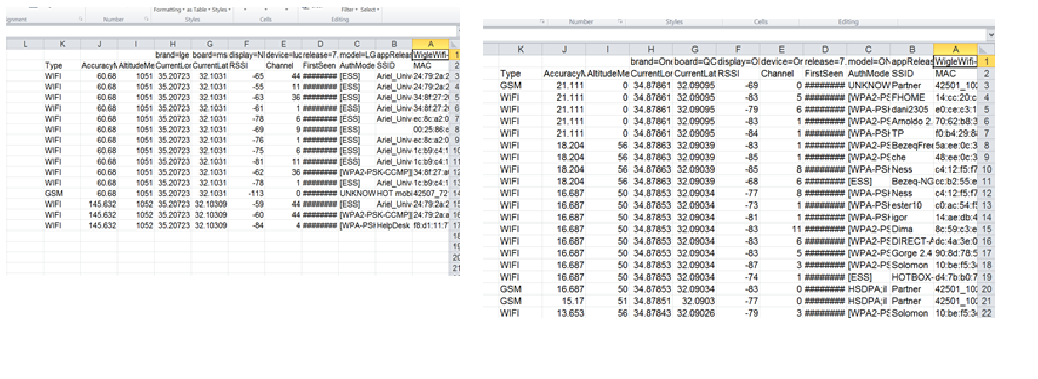
1. ***mergeCSVfiles* –** This class gets a folder path, takes only CSV files and creates a new CSV file that contains all the top ten strongest (signal) Wi-Fi networks and their information for every timestamp from all those files, arranging them in an ascending order.
2. ***WiFiNetwork* –** This class represent an object called ***WiFiNetwork*,** it contains all the information of a specific Wi-Fi network that had been scanned by the app and added into the file. The information has the device ID, the network's name, its MAC, signal, frequency, longitude, latitude, altitude and its timestamp- the time it had been scanned. The class has a function that converts a channel into frequency, because the app only gives us the channel and we wanted to know and save it as frequency for later use with **"Google Earth".**
3. ***convertToKML –*** This class gets an arranged CSV file (a file that was created with ***mergeCSVfiles***) and converts it into a KML format. It gives the user 3 filter options of what he wants the file to have, to filter by time- specific date and hour, filter by a point on the map and a radius to get the networks that are within this radius, or to filter by ID – the name of the device. There is a function called ***getColor()*** that gives every pin a color that represent its top network (with the strongest signal between the networks from the same timestamp). A green color is given when the signal is higher/equal to -70, a yellow when it's between -70 and -90, and a red when it is lower than -90.
4. ***Filter –*** This class gets the uses choice of filter and his/her input (using Scanner).
5. ***LocPoint –*** This class creates a ***point(latitude, longitude).*** It has a function ***pointInCircle()*** that checks if a given point is within the radius from the current point.
6. ***Main -*** The main class.

**Some examples:**

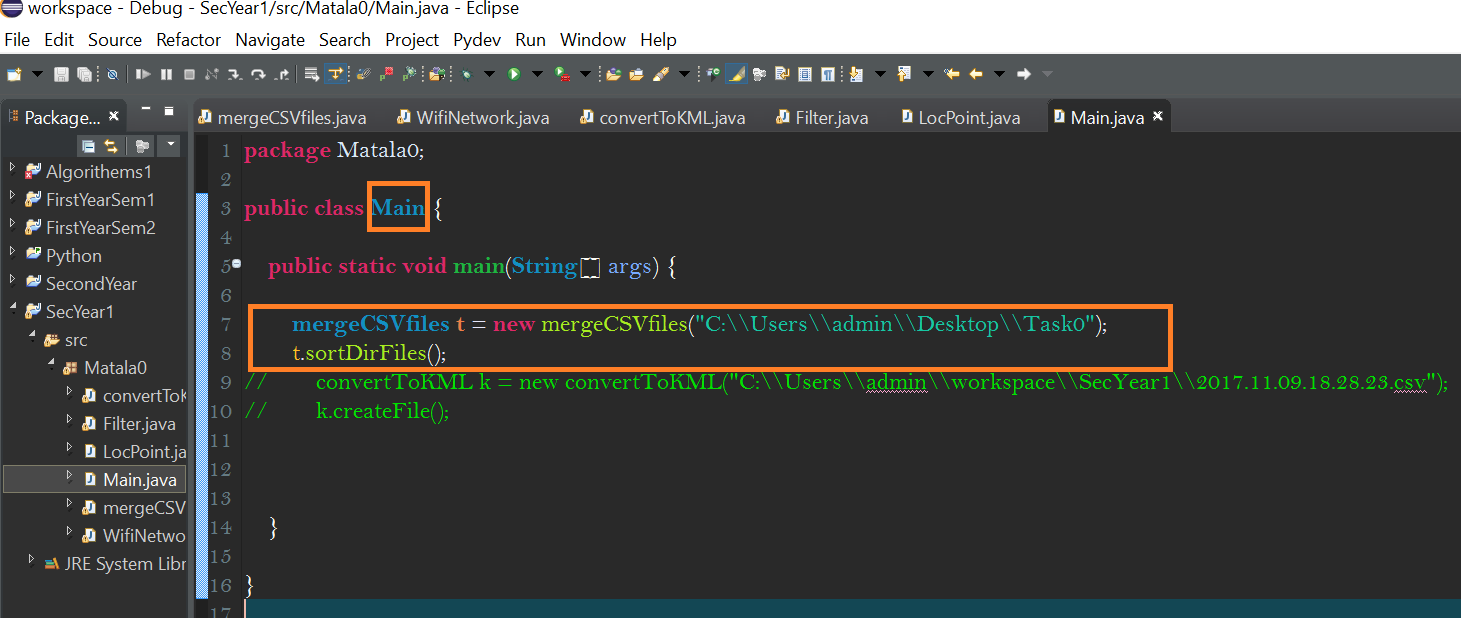
**First, here we have the csv files exported from the  *"WiGLE WiFi Wardriving"*  app (our file, and Boaz's):**

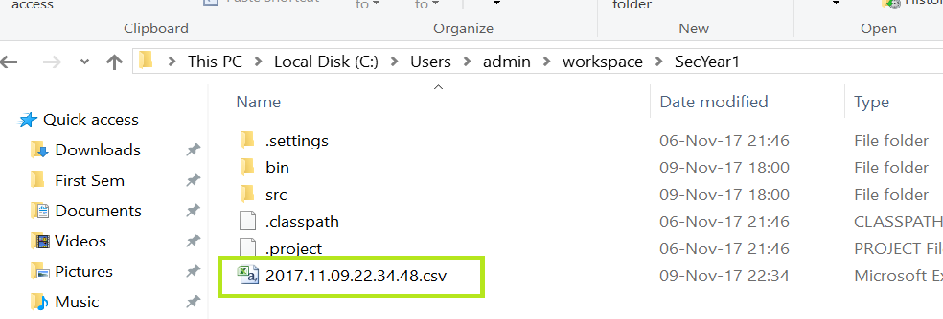


**Here's a look inside them:**

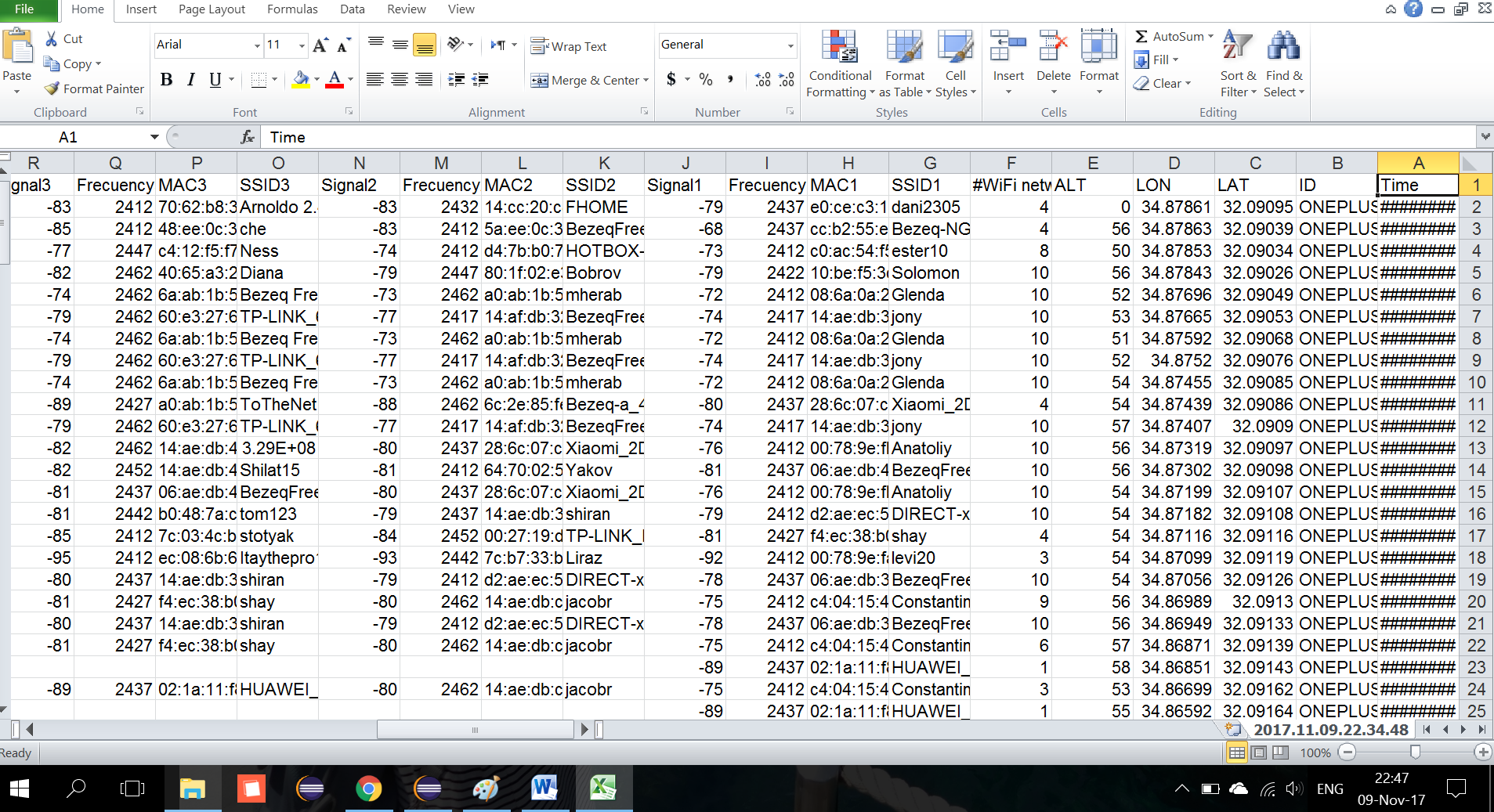


**We run the program in the class Main:**



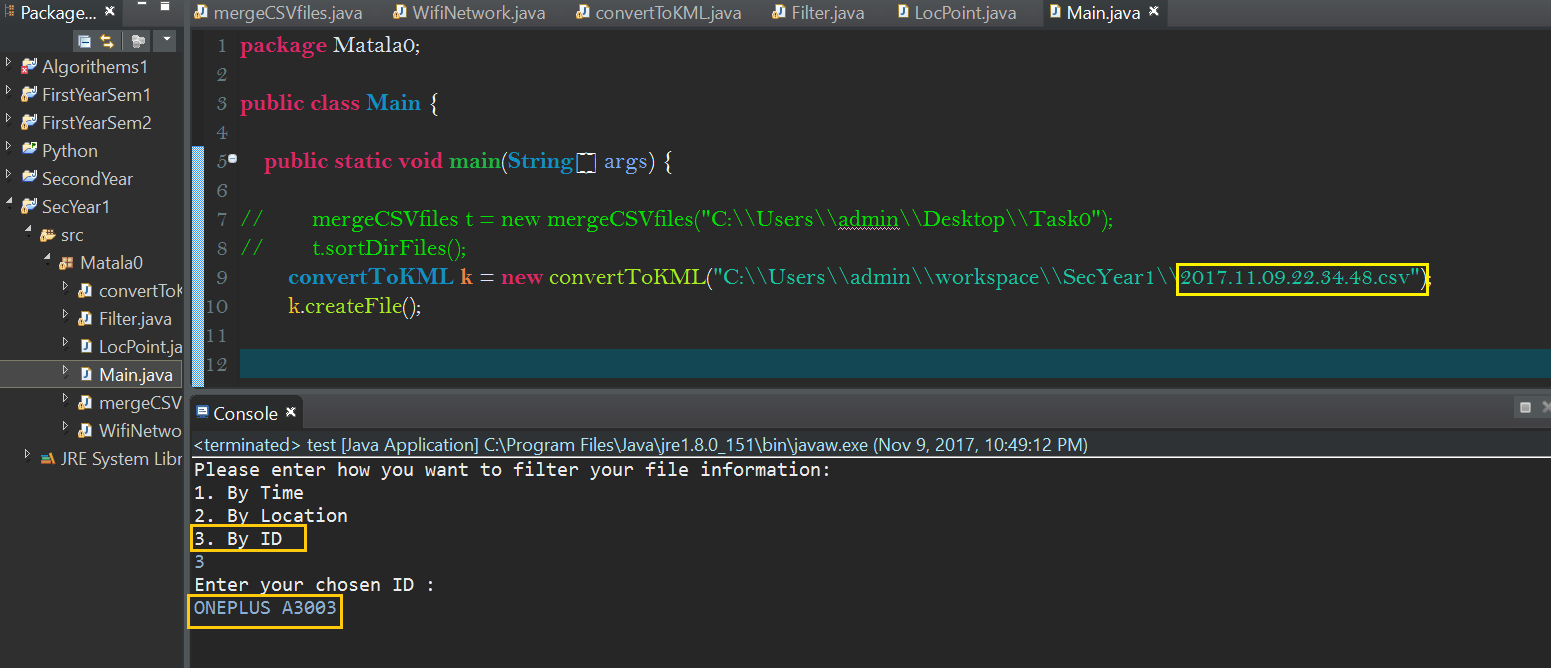
**The file was created (the timestamp as its name):** 

**A look inside the new file:**

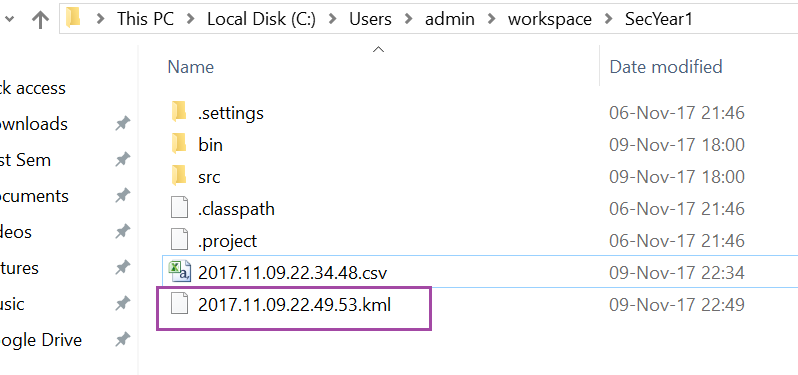


**We run the program again, this time to get a KML file to upload on "Google Earth".**

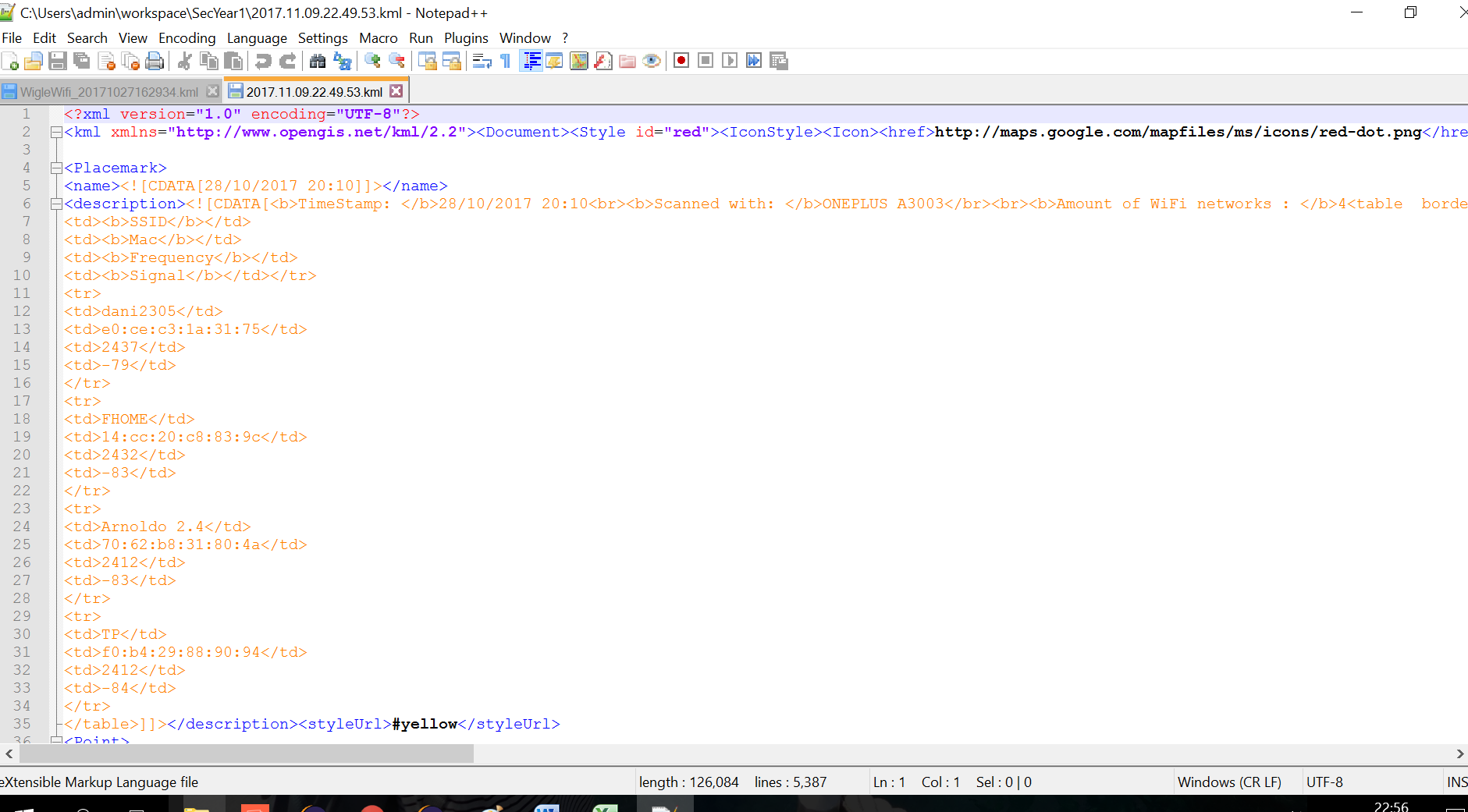
**We were asked to choose how we want it to filter, we chose by ID and then we wrote the name of the device we wanted it to use – ONEPLUS A3003:**



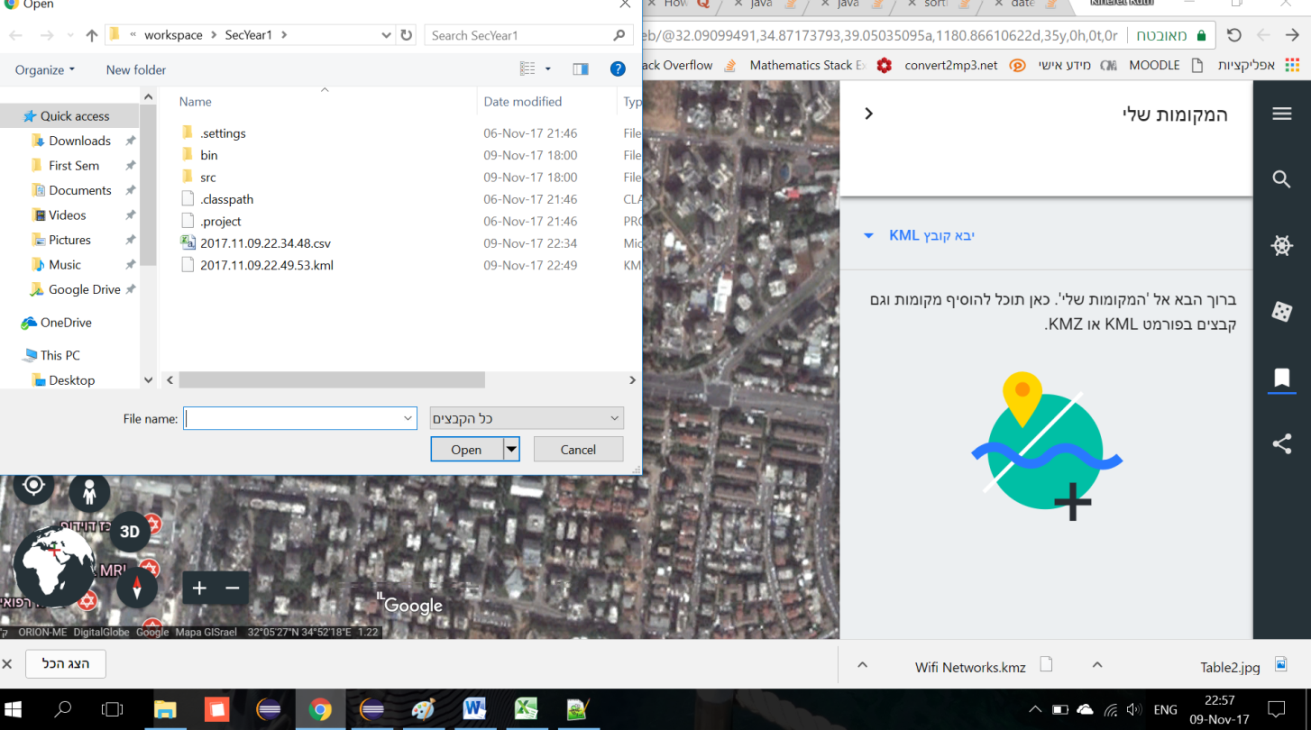
**The KML file was created (timestamp as its name):**



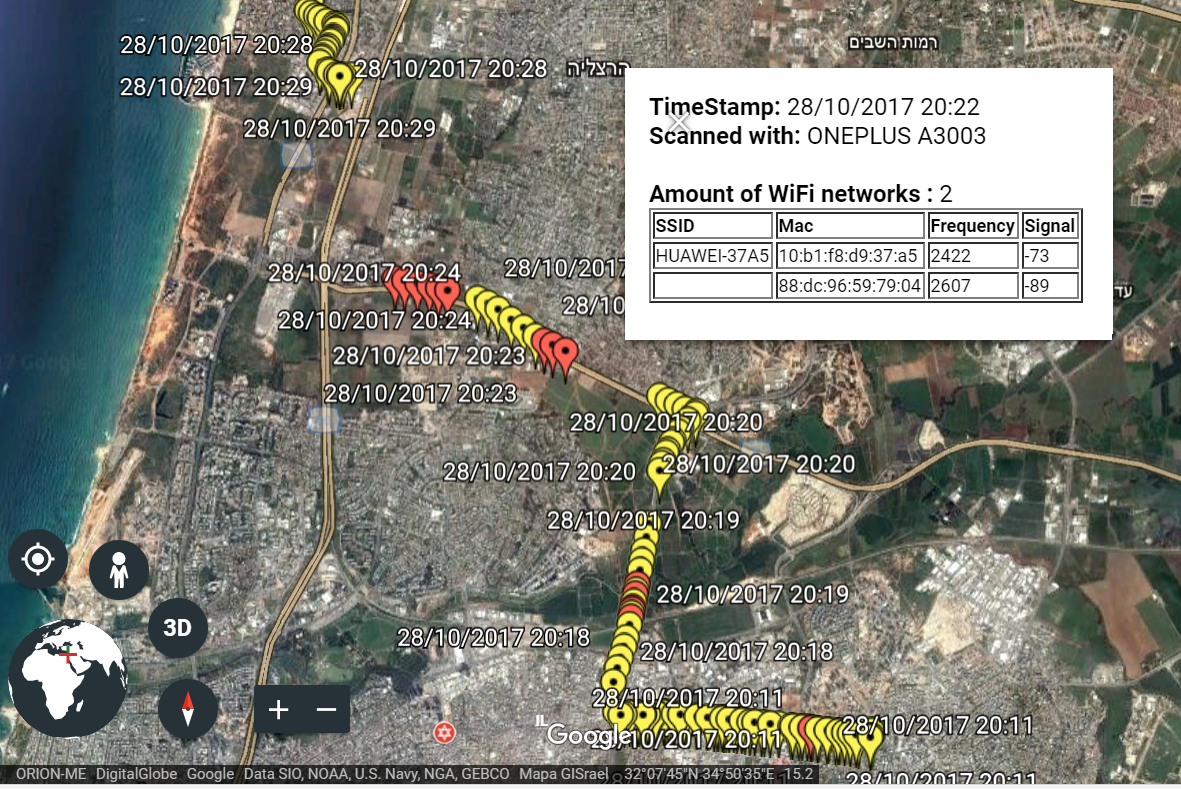
**A look inside the KML file (opened with Notepad++):**



**We uploaded the file on "Google Earth" web option:**



**Here is what it looks like - by clicking on a pin it opens the information for the selected timestamp & place:**



**Information we got from the internet for our code:**

* <https://stackoverflow.com/questions/2784514/sort-arraylist-of-custom-objects-by-property>
* <https://stackoverflow.com/questions/16425127/how-to-use-collections-sort-in-java-specific-situation?answertab=votes#answer-16425169>
* <https://stackoverflow.com/questions/23068676/how-to-get-current-timestamp-in-string-format-in-java-yyyy-mm-dd-hh-mm-ss>
* <https://stackoverflow.com/questions/19012146/how-can-i-a-coordinate-point-by-radius-with-another-coordinate-point-and-distanc>
* <https://developers.google.com/kml/documentation/kml_tut>
* <https://www.quora.com/How-do-I-create-a-table-in-xml>
* Also the convert channel to frequency function in the class ***WiFiNetwork.***