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

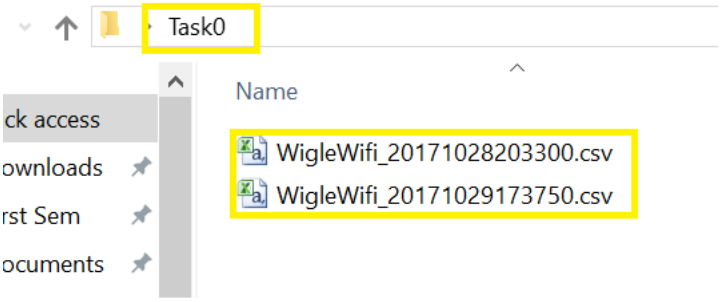
1. ***mergeCSVfiles* –** This class gets a folder path, takes only CSV files and creates a new CSV file that contains all the top ten strongest (signal) Wi-Fi networks and their information for every timestamp from all those files, arranging them in an ascending order.
2. ***WiFiNetwork* –** This class 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.

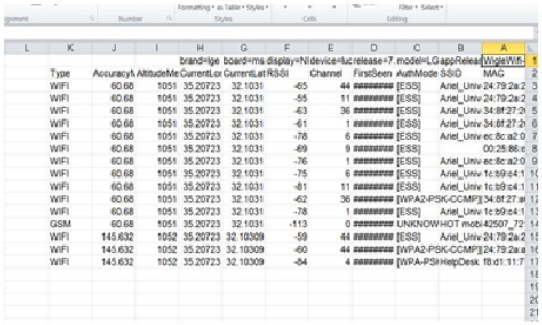
4. ***Filter –*** This class gets the uses choice of filter and his/her input (using Scanner).
5. ***LocPoint –*** This class creates a ***point(latitude, longitude).*** It has a function ***pointInCircle()*** that checks if a given point is within the radius from the current point.
6. ***Main -*** The main class.

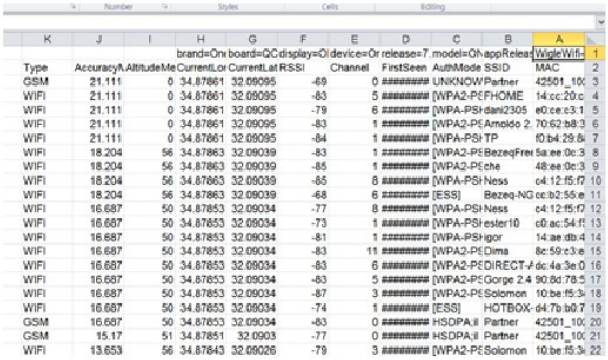
Some examples:

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

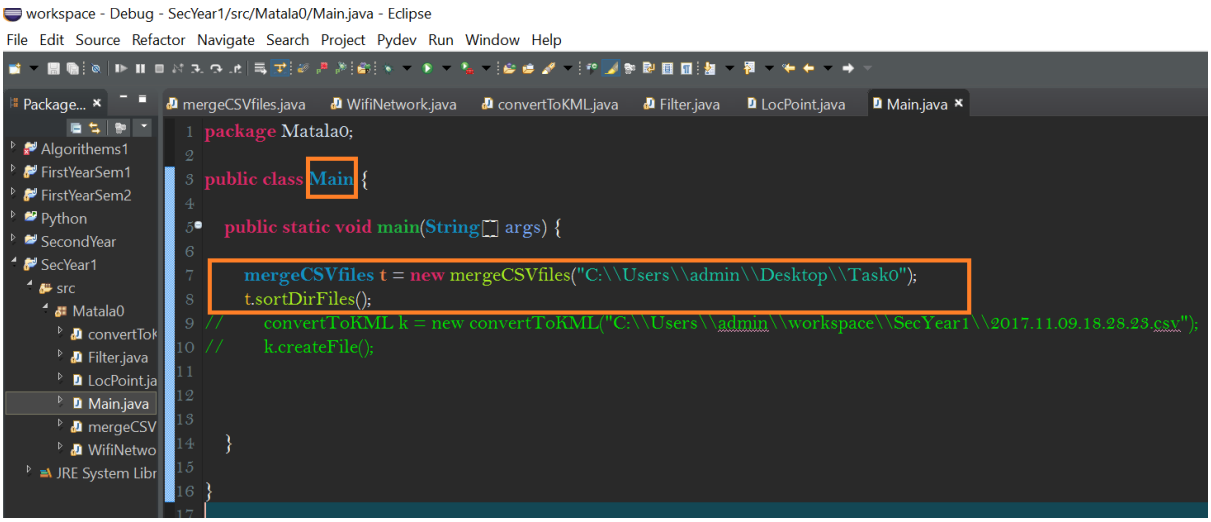


Here's a look inside them:

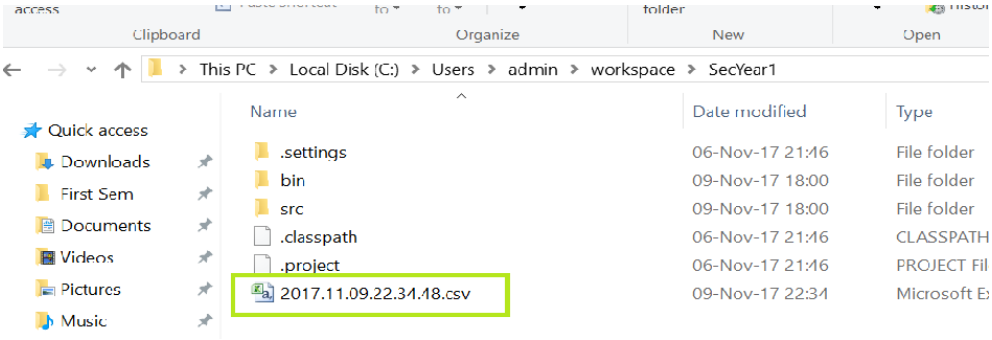




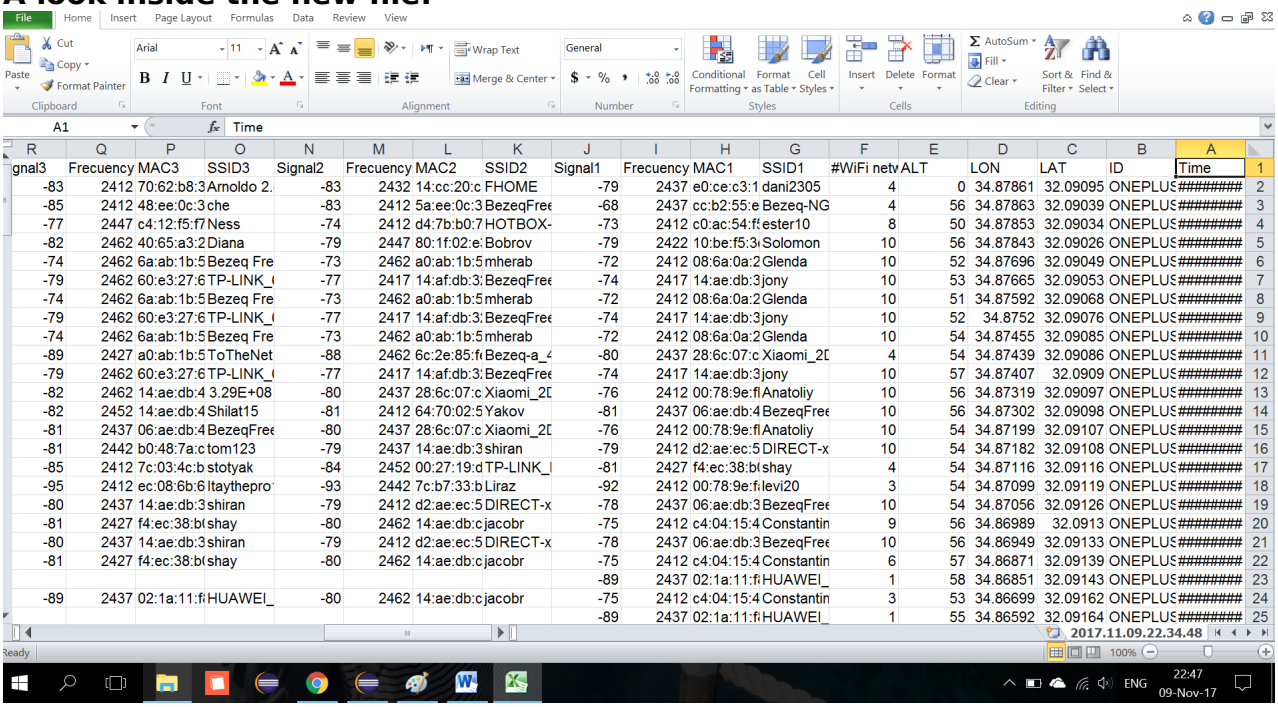
We run the program in the class Main:



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



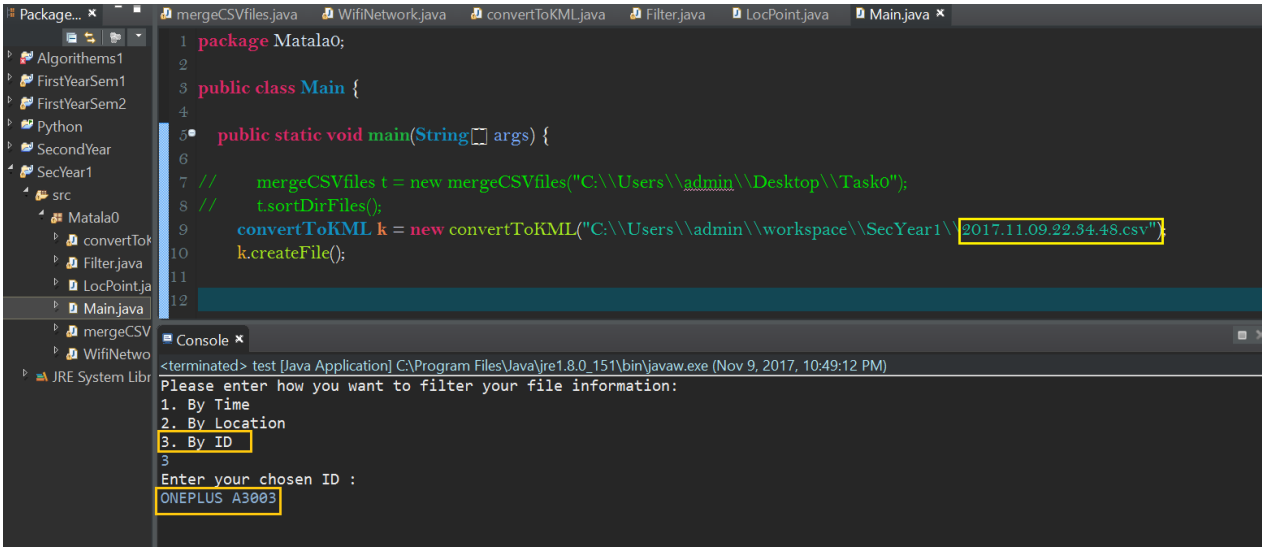
A look inside the new file:



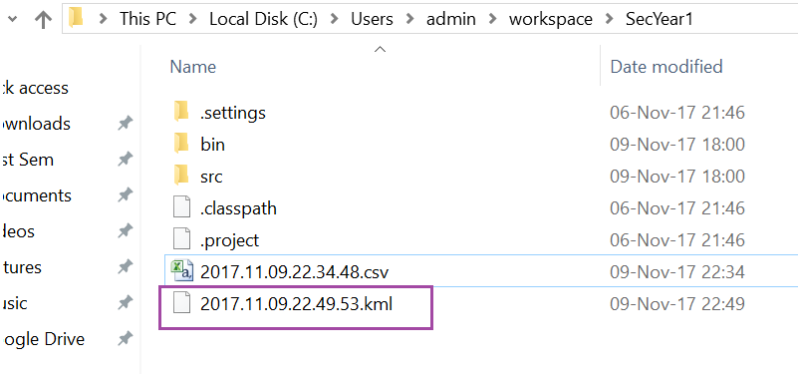
R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
gnal3	Frecuency	MAC3	SSID3	Signal2	Frecuency	MAC2	SSID2	Signal1	Frecuency	MAC1	SSID1	#WiFi netv	ALT	LON	LAT	ID	Time
-83	2412	70:62:b8:3	Arnoldo 2.	-83	2432	14:cc:20:c	FHOME	-79	2437	e0:ce:c3:1	dani2305	4	0	34.87861	32.09095	ONEPLUS	########
-85	2412	48:ee:0c:3	che	-83	2412	5a:ee:0c:3	BezeqFree	-68	2437	cc:b2:55:e	Bezeq-NG	4	56	34.87863	32.09039	ONEPLUS	########
-77	2447	c4:12:f5:f7	Ness	-74	2412	d4:7b:b0:7	HOTBOX-	-73	2412	c0:ac:54:f	ester10	8	50	34.87853	32.09034	ONEPLUS	########
-82	2462	40:65:a3:2	Diana	-79	2447	80:1f:02:e	Bobrov	-79	2422	10:be:f5:3	Solomon	10	56	34.87843	32.09026	ONEPLUS	########
-74	2462	6a:ab:1b:5	Bezeq Fre	-73	2462	a0:ab:1b:5	mherab	-72	2412	08:6a:0a:2	Glenda	10	52	34.87696	32.09049	ONEPLUS	########
-79	2462	60:e3:27:6	TP-LINK_(	-77	2417	14:af:db:3	BezeqFree	-74	2417	14:ae:db:3	jony	10	53	34.87665	32.09053	ONEPLUS	########
-74	2462	6a:ab:1b:5	Bezeq Fre	-73	2462	a0:ab:1b:5	mherab	-72	2412	08:6a:0a:2	Glenda	10	51	34.87592	32.09068	ONEPLUS	########
-79	2462	60:e3:27:6	TP-LINK_(	-77	2417	14:af:db:3	BezeqFree	-74	2417	14:ae:db:3	jony	10	52	34.8752	32.09076	ONEPLUS	########
-74	2462	6a:ab:1b:5	Bezeq Fre	-73	2462	a0:ab:1b:5	mherab	-72	2412	08:6a:0a:2	Glenda	10	54	34.87455	32.09085	ONEPLUS	########
-89	2427	a0:ab:1b:5	ToTheNet	-88	2462	6c:2e:85:f	Bezeq-a_4	-80	2437	28:6c:07:c	Xiaomi_2[	4	54	34.87439	32.09086	ONEPLUS	########
-79	2462	60:e3:27:6	TP-LINK_(	-77	2417	14:af:db:3	BezeqFree	-74	2417	14:ae:db:3	jony	10	57	34.87407	32.0909	ONEPLUS	########
-82	2462	14:ae:db:4	3.29E+08	-80	2437	28:6c:07:c	Xiaomi_2[	-76	2412	00:78:9e:f	Anatoliy	10	56	34.87319	32.09097	ONEPLUS	########
-82	2452	14:ae:db:4	Shilat15	-81	2412	64:70:02:5	Yakov	-81	2437	06:ae:db:4	BezeqFree	10	56	34.87302	32.09098	ONEPLUS	########
-81	2437	06:ae:db:4	BezeqFree	-80	2437	28:6c:07:c	Xiaomi_2[	-76	2412	00:78:9e:f	Anatoliy	10	54	34.87199	32.09107	ONEPLUS	########
-81	2442	b0:48:7a:c	tom123	-79	2437	14:ae:db:3	shiran	-79	2412	d2:ae:ec:5	DIRECT-x	10	54	34.87182	32.09108	ONEPLUS	########
-85	2412	7c:03:4c:b	stotyak	-84	2452	00:27:19:d	TP-LINK_[	-81	2427	f4:ec:38:b	shay	4	54	34.87116	32.09116	ONEPLUS	########
-95	2412	ec:08:6b:6	Itaythepro	-93	2442	7c:b7:33:b	Liraz	-92	2412	00:78:9e:f	levi20	3	54	34.87099	32.09119	ONEPLUS	########
-80	2437	14:ae:db:3	shiran	-79	2412	d2:ae:ec:5	DIRECT-x	-78	2437	06:ae:db:3	BezeqFree	10	54	34.87056	32.09126	ONEPLUS	########
-81	2427	f4:ec:38:b	shay	-80	2462	14:ae:db:c	jacobr	-75	2412	c4:04:15:4	Constantin	9	56	34.86989	32.0913	ONEPLUS	########
-80	2437	14:ae:db:3	shiran	-79	2412	d2:ae:ec:5	DIRECT-x	-78	2437	06:ae:db:3	BezeqFree	10	56	34.86949	32.09133	ONEPLUS	########
-81	2427	f4:ec:38:b	shay	-80	2462	14:ae:db:c	jacobr	-75	2412	c4:04:15:4	Constantin	6	57	34.86871	32.09139	ONEPLUS	########
								-89	2437	02:1a:11:f	HUAWEI_	1	58	34.86851	32.09143	ONEPLUS	########
-89	2437	02:1a:11:f	HUAWEI_	-80	2462	14:ae:db:c	jacobr	-75	2412	c4:04:15:4	Constantin	3	53	34.86699	32.09162	ONEPLUS	########
								-89	2437	02:1a:11:f	HUAWEI_	1	55	34.86592	32.09164	ONEPLUS	########

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

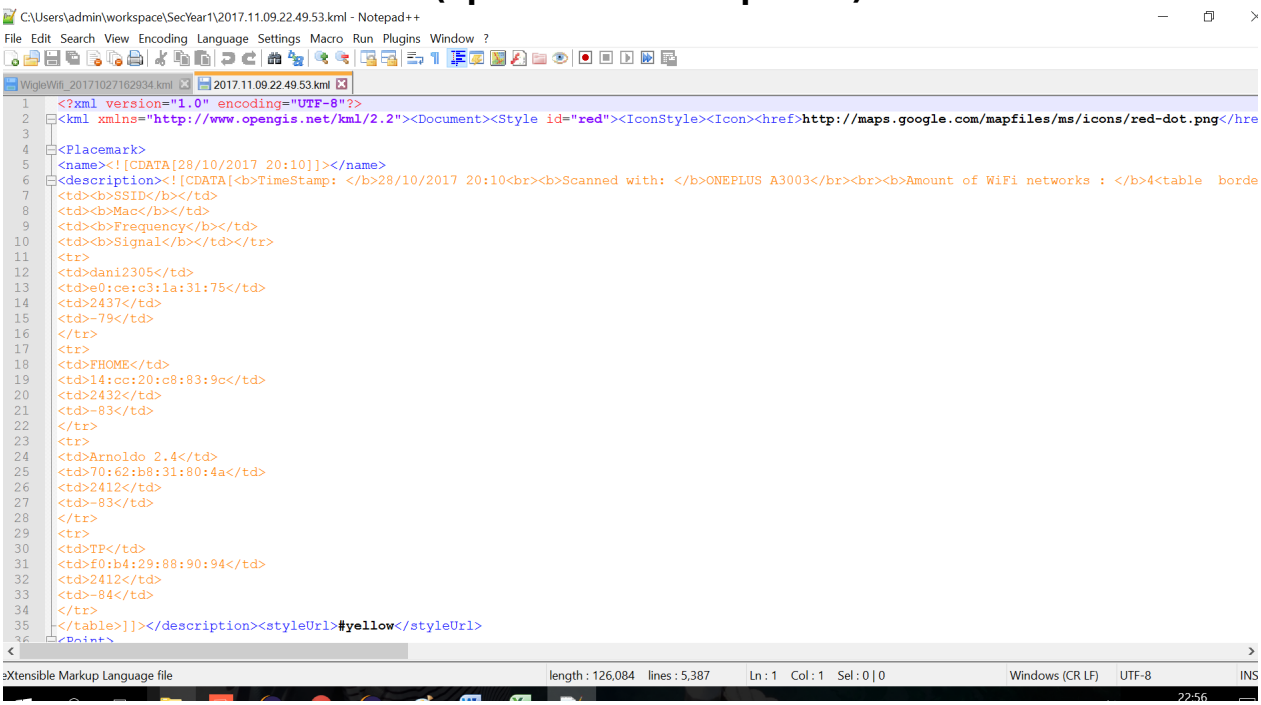
We were asked to choose how we want it to filter, we chose by ID and then we wrote the name of the device we wanted it to use – ONEPLUS A3003:



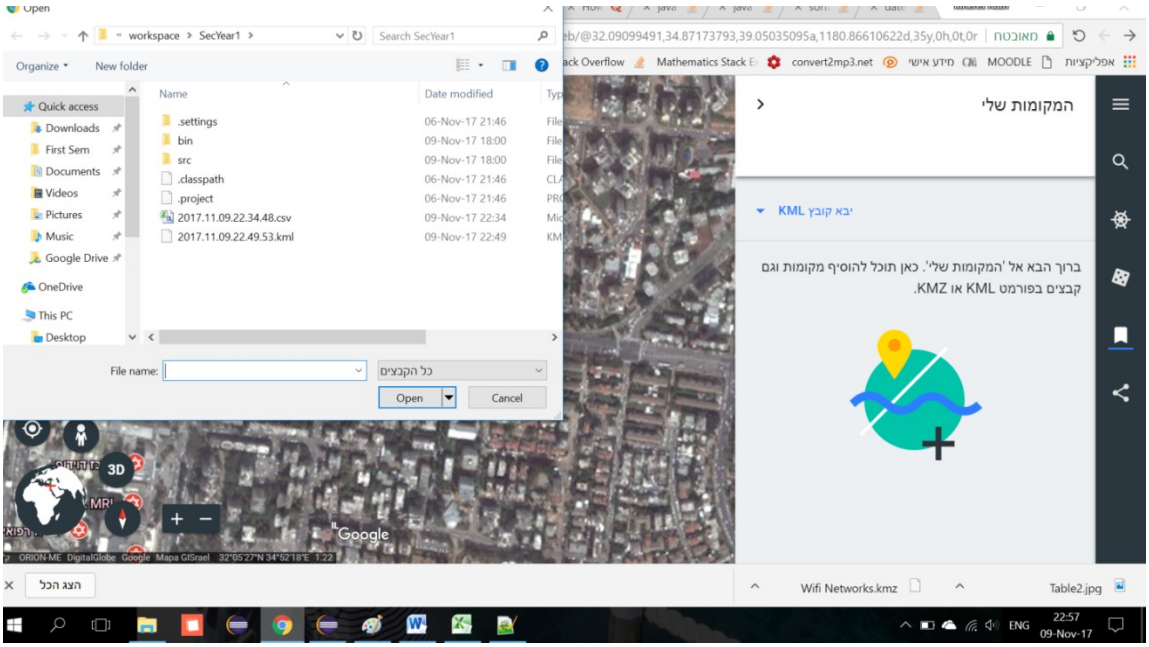
The KML file was created (timestamp as its name):



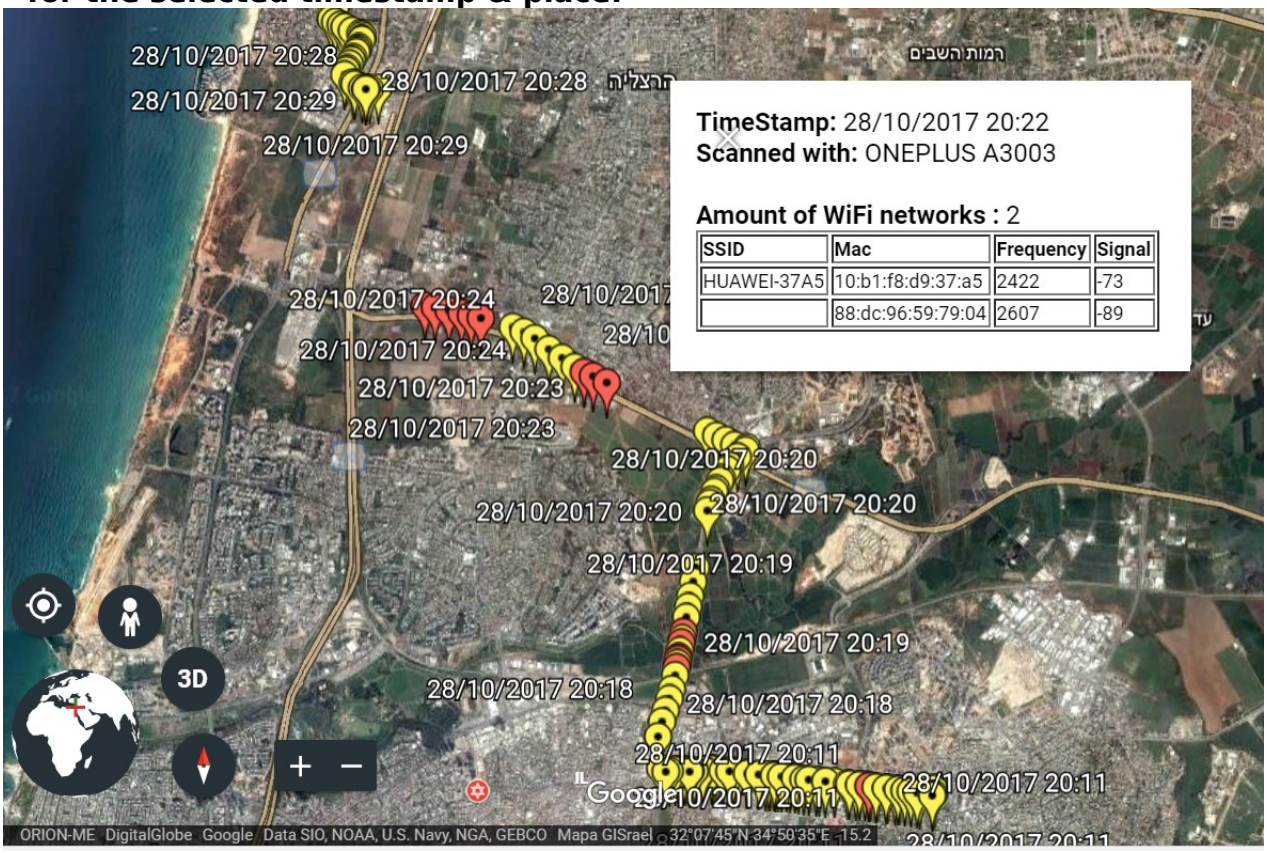
A look inside the KML file (opened with Notepad++):



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.***