**Year 2 – semester A – Object Oriented Course – Report EX0**

**Names**: Alex Fishman & Alona Kondratenko

Input: folder path that contains several CSV format files, those files contains data from Wiggle\_WIFI application. The folder contains also another folder named 'OUT'.

Output:

* Can provide one big CSV file that contain data from all CSV's in the folder, this Union CSV is filtered by several categories that will be explained later.
* Can provide KML file (with or without the filter) from the data in CSV file/s and be open on google earth.

How the output works: the program takes all the CSV format files that there is on the given path and creates new CSV file with the fields: time, model, longitude, altitude, latitude. The Union CSV is filtered by time and ID.

The chosen fields: 6 permanent columns that represents the time and the coordinates, and 40 dynamic columns that filtered by the strongest signal (he is a part of the data). In addition – the file can be filtered by tome and ID.

The program combined from 4 classes:

* **CSV\_header\_row** – describes the head line (the labels) on the input file, it contains private Strings that matching the labels on the original file, getters of the values and one constructor that gets the relevant values from the header line.

**Main use**: saving data about the devise that collect the data.

* **CSV\_row** – describes the structure of the rows in input files.

This class **implements** from interface: Comparable<CSV\_row>.

Has been done an @Override to CompareTo function (that not in use right now).

This class contains private String variables and private CSV\_header\_row variable, getters of the values and one constructor that gets the values of the first line, and CSV\_header\_row object.

There is a function called getCriterionForGroup that returns the key for filtering the rows (by: time, model, longitude, altitude, latitude).

In addition – was defined five comparators for the filter by: time, model, longitude, altitude, latitude, SSID. For each comparator object has been done @Override to the Compare function.

**Main use**: creates the data rows on Union CSV file.

* **CSV\_Merged\_row –** this class represents a row from Union output file.

Contains private String variables: 5 key labels and one more " extends " key that contains the 40 more dynamic variables. Getters and one constructor that gets 2 Strings: the first is the keys of data (they separated by ',') and the second String is the "extension" (also separated by ','). Also – the constructor make sure that the extension does not het over 40 (if we lack of data – it will be spaces).

The class contains five comparators (ID, time, alt, lon, lat) for each comparator object has been done @Override to the Compare function.

The class also contains CompareByTime function and CompareByPlace function that we will use on the filter.

* **toCSVtoKML –**

in this class we define List<CSV\_row> variable that gets data from input file, and List<SCV\_Merged\_row> that gets data that will appear on output file.

readCSV function – gets file name of input path and read the information from this file into rowList object.

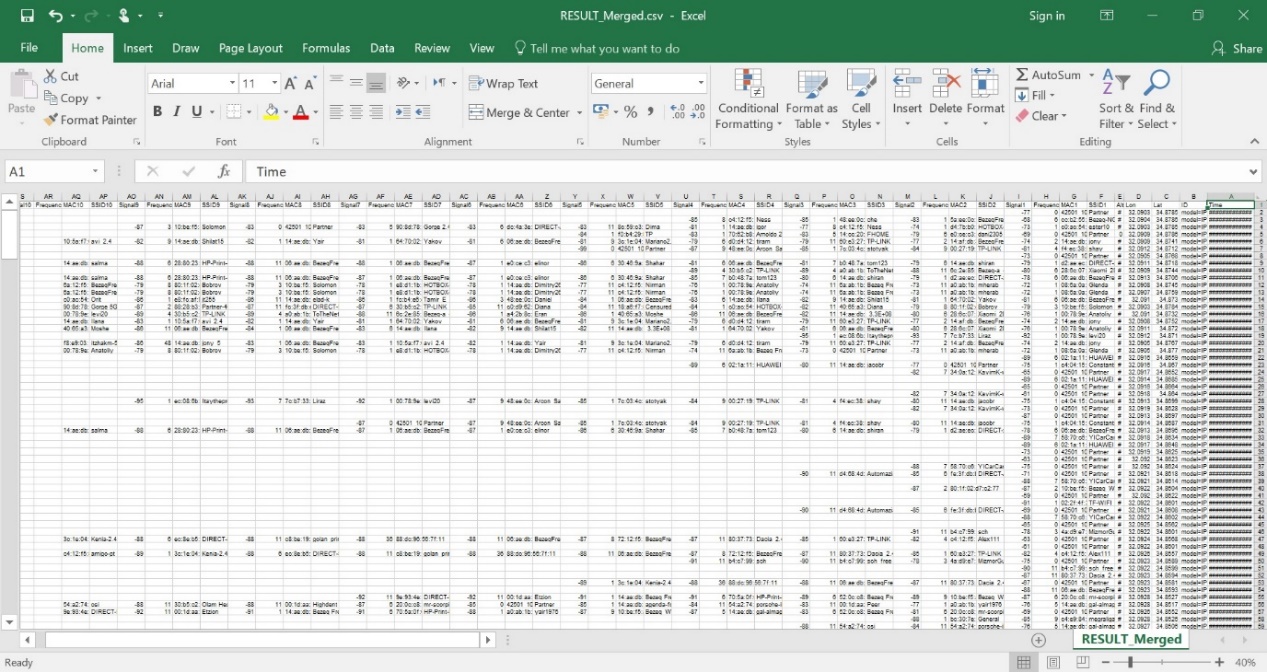
mergeData function – collects filtered data from rowList (by the signal) and make another object called rowMergeList, and now we able to filter by several criteria's (ID, time) by the comparator object that defined in SCV\_Merged\_row class.

createCSV\_Merged\_File function – gets output name and the mergedRowList object and creates the output file.

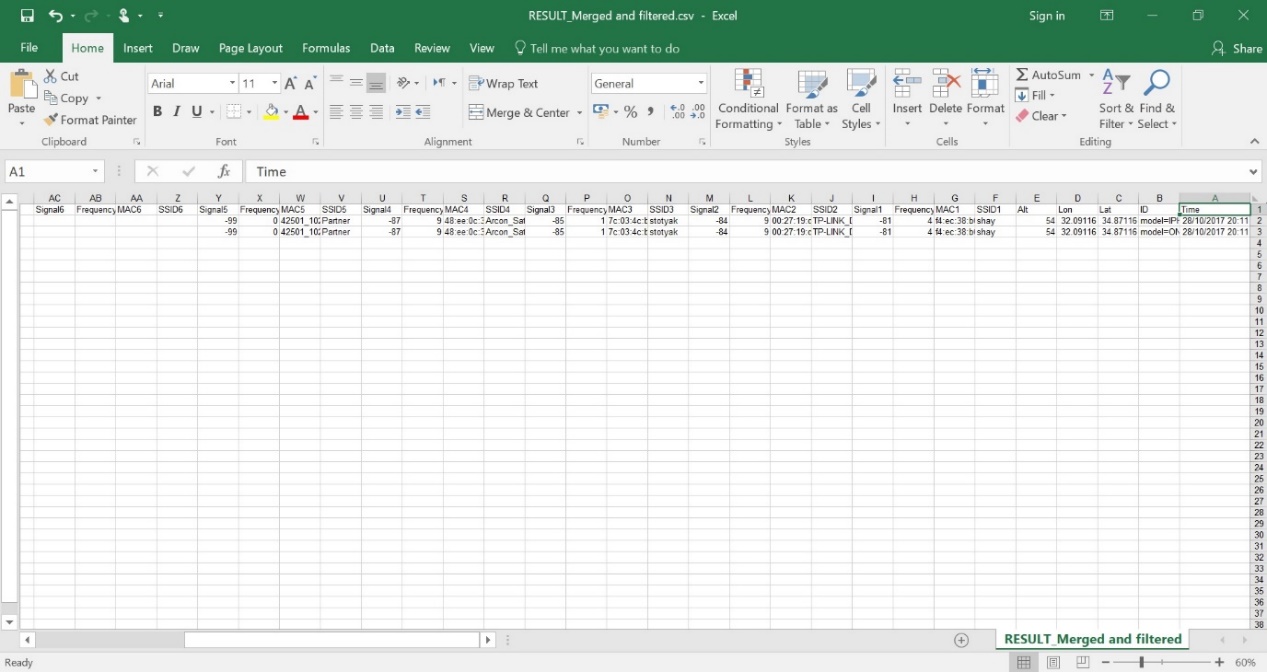
toKML function – gets the path of Union CSV file that was created by us and create KML file into our 'OUT' folder.

**Examples of the usage of this program:**

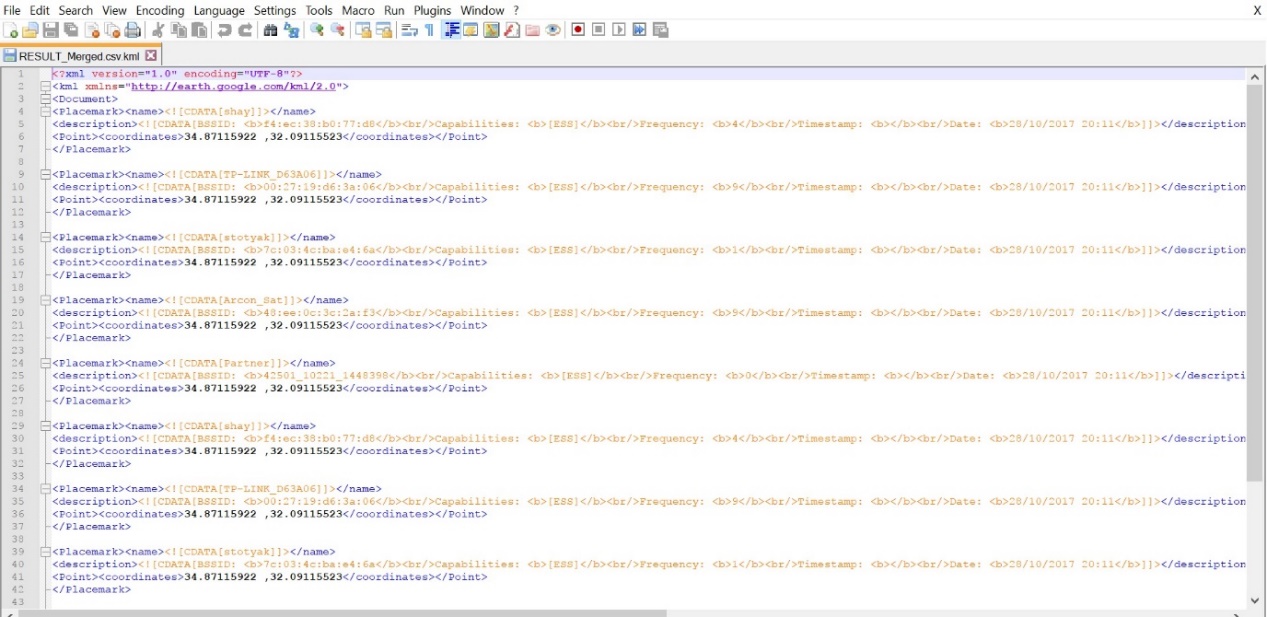
Union CSV file:



Filtered CSV file:

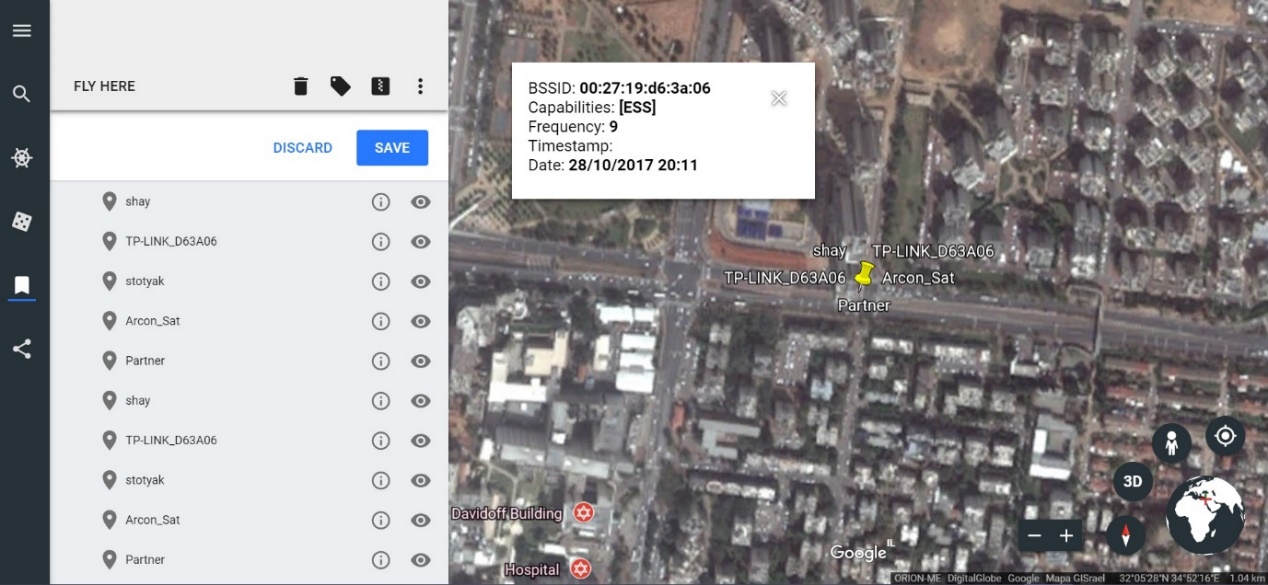


KML format file:



Opening the files on google earth:

Filtered file:



Unfiltered file:

