**מטלה 1 - מונחה עצמים**

תאור כללי

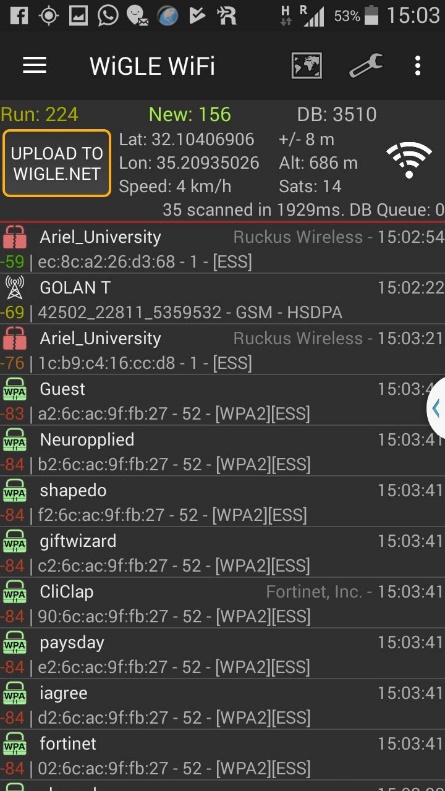
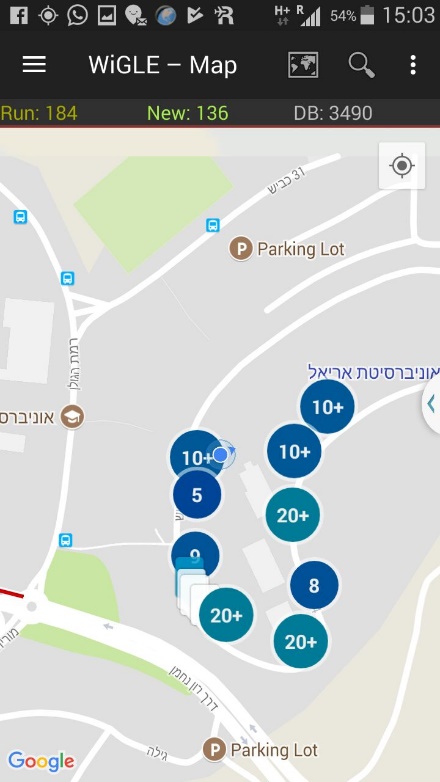
מטרה- איסוף נתונים על רשתות wifi עריכת המידע והצגתו ע"ג אפליקציית מפות

תכנית הגאווה שכתבנו מחולקת לשני חלקים עיקריים:

חלק א- מקבל שם של תיקייה עם קבצי csv , קבצים אלו מכילים מידע על דגימות WiFi שהתקבלו מהקלטות שביצענו על ידי שימוש באפליקציית Wigle-Wifi . את ההקלטות אנחנו ממיינים ומוצאים את עשרת הדגימות הכי חזקות לפי עוצמת האות בדיוק של שניה, לאחר מכן כותבים את הכול לקובץ csv חדש שמאחד בתוכו את עשרת הדגימות החזקות בכל שניה .

חלק ב- קורא את הקובץ csv שייצרנו בחלק א יודע לסנן אותו לפי דרישות המשתמש על הזמן המיקום והID- . לאחר מכן ממיר את הקובץ לפורמט KML שאת קובץ זה ניתן לקרוא ע"י שימוש באפליקציית מפות דוגמאת Google Earth .

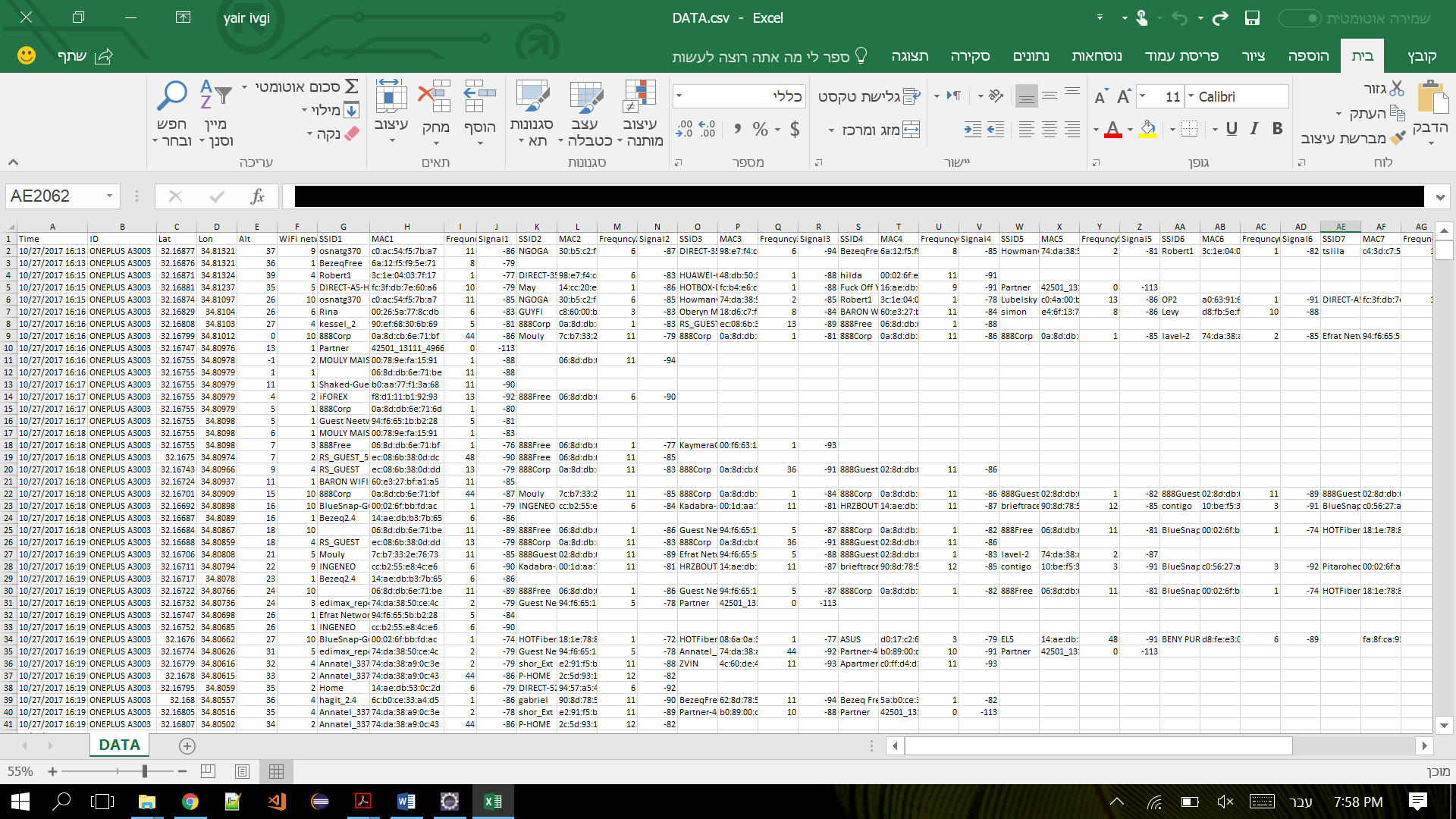
תמונות מהאפליקציה במהלך ההקלטה ברחבי הקמפוס העליון מיום 9-11-17



תמונות מGoogle earth לאחר עיבוד הנתונים והמרתם לקובץ KML



מידע לאחר עריכת והדפסתו לקובץ



כלי תוכנה

התוכנה נכתה בשפת JAVA בסביבת עבודה eclipse .

הנתונים נאספו ע"י שימוש באפליקציית Wigle-Wifi.

המידע הוצג תוך שימוש באפליקציית Goole Earth

באגים:

ליאיר אין פלאפון מבוסס אנדרואיד יש לו אייפון , לכן השתמשנו גם במידע שהעלה בועז.

חלקי המערכת:

[Main](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/Main.java)

המחלקה הראשית שממנה אנו מפעילים את חלי התכנית השונים.

[CsvReader](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/CsvReader.java)

מחלקה זו קוראת את כל הקבצים בתיקייה והיא האחראית לעיבוד הקובץ לפי דרישותינו.

[WifiSpot](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/WifiSpot.java)

מחלקה זו מייצגת נקודת wifi אחת עם הנתונים הרלוונטים.

[RawData](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/RawData.java)

מחלקה זו מייצגת שורה שמכילה את כל הנקודות wifi שנקלטו באותו זמן. בתוכה יש את המתודה organizeData שמאפשרת למיין את הנתונים לפי עוצמת האות ולוקחת את עוצמת 10 נקודות ה wifi הכי חזקות.

[WriteCsv](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/WriteCsv.java)

מחלקה זו אחראית לכתיבה של המידע לקובץ הcsv החדש לאחר שטיפלנו במידע.

[IFilter](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/IFilter.java)

ממשק זה אחראי לכל פונקציות הסינון.

[FilterById](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/FilterById.java)

מחלקה זו מטפלת בנושא הסינון על פי שם ומחזירה את המידע בצורה מסוננת.

[FilterByTime](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/FilterByTime.java)

מחלקה זו מטפלת בנושא הסינון על פי זמן ומחזירה את המידע בצורה מסוננת.

[FilterByLocation](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/FilterByLocation.java)

מחלקה זו מטפלת בנושא הסינון על פי מיקום ומחזירה את המידע בצורה מסוננת.

[ReadAndWriteWithFilter](https://github.com/YairIvgi/OOP_project/blob/master/WIfi_Location2/src/ReadAndWriteWithFilter.java)

מחלקה זו אחראית לכתיבת קובץ הcsv לקובץ בפורמט KML.

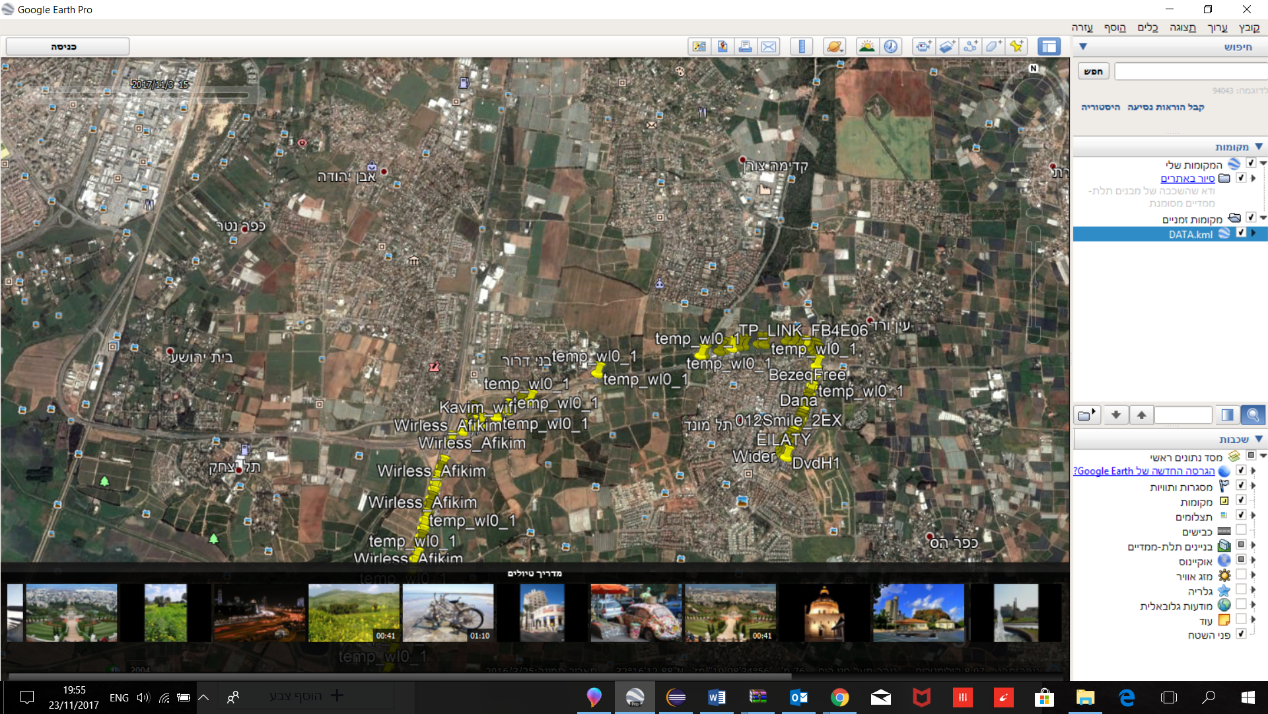
בנוסף, קיימות מספר מחלקות של Junit שבודקות עבור כל מחלקה אם הפונקציות שם עובדות כמו שצריך.

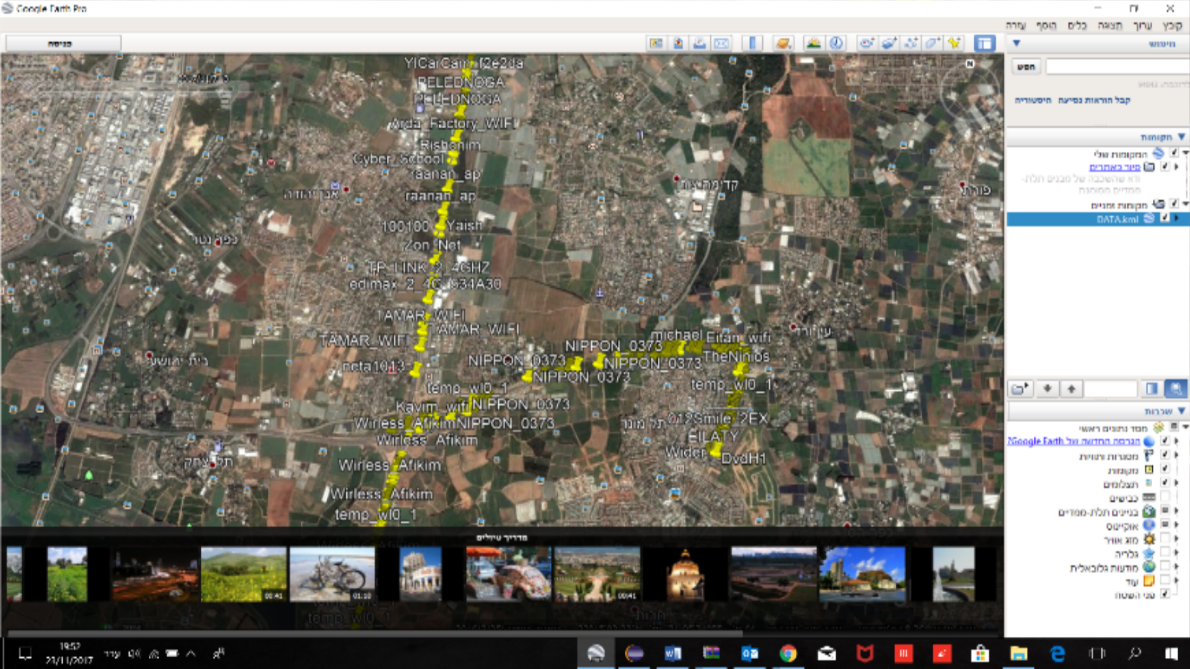
קובץ KML :

בחרנו בחבילה kml frame work ליצירת KML.

בחרנו בחבילה זו כי ראינו המלצות במספר מקומות ב stack overflow, בדקנו את החבילה וראינו שיש לה תיעוד סביר והיא ממלאת את כל הפונקציות הנדרשות בתרגיל.

המידע ב Google Earth:

לפני קליטת מספר נקודות:

לאחר קליטת מספר נקודות: