

ANYWAY

המרכז
הקדמי
לב



מחלקה למדעי המחשב

פרויקט גמר בהנדסת תוכנה

אפליקציה ל **תצוגת אזורים** **בכבי ישראל**

עבור מכשירי android

מגישות: דניאל אילוז ושרה בן שמעון

מנהל מטבח האקדמיה: מר דן זילברשטיין

מנהל מטבח החברה: ד"ר דניאל הרשקביץ

תשע"ו - 2016

Department of Computer Science
Final Project in Software Engineering

Risk Zones Visualization in Israel roads

Android application

Students: Danielle Illouz and Sarah Ben Shimon
Academic Advisor: Mr. Dan Zilberstien
Technology Advisor: Dr. Daniel Hershkovitz

2016

תקציר

בשנים האחרונות עם עליית מספר תאונות הדרכים בצורה משמעותית התגלה הצורך במודעות לסכנה עבור משתמשי הדרק.

בפרויקט גמר זה המשכנו פיתוח קיים של אפליקציה לטלפון חכם בסביבת android עבור משתמשי הדרק : נהגים והולכי רגל.

האפליקציה מאפשרת למשתמש להזיהות את הסכנה שבדרך על ידי כך שהיא צובעת את מפת מדינת ישראל על פי רמת הסיכון של אזור ספציפי ולקבל התראות קוליות והתראות דחיפה בכניסה לאזור בעל רמת סיכון גבוהה.

במסגרת פרויקט זה פותח מודל חישובי - סטטיסטי המתבסס על תאונות דרכים שקרו בעבר והפקת רמת הסיכון לכל אזור על סמך על נתונים אלו. הנתונים מעוברים על ידי הלשכה המרכזית לסטטיסטיקה וארגון איחוד הצלה.

הפרויקט מורכב משני חלקים : החלק הראשון כולל פיתוח צד שרת הכלול בניתן מסד נתונים וערכונו באופן קבוע, הטמעת המודל החישובי שפותח, חישוב סיכון למפה המחוקקת לאזורים קבועים ובניית ותחזוקת תקשורת עם האפליקציה. החלק השני כולל פיתוח צד ללקוח עבור משתמשי הדרק, צביעת המפה הנראית לעיני המשתמש לפי רמות הסיכון שהתקבלו מהשרת, ייצור התראה קולית והתראות דחיפה. כמו כן, בין השרת והלקוח ישנו עורך תקשורת אשר מחובר לאורך כל הזמן.

המטרה הכלכלית של האפליקציה היא לעורר מודעות לסכנה בדרכים וכן ליצור אפשרות למשתמש לבחור בדרכים חלופיות ולהגבר את ערנותו זהירותו בקרבת מקומות מסוכנים.

מטרה נוספת של הפרויקט היא ייצור לחץ ציבוררי מצד משתמשי האפליקציה על הרשותות הציבוריות על מנת שיתקנו את ליקויי הדרך .

האפליקציה פותחה במסגרת פרויקט *uyawayum* שם לו למטרה לצמצם את מספר תאונות הדרכים על ידי הגברת מודעות הציבור אליהן.

Abstract

In the last years, as a result of the increasing number of traffic accidents, the need of awareness to road safety became more relevant than ever.

In this project we continued the development of an application to android smartphones for roads users: drivers and pedestrians.

The application lets the users to identify the dangerous places on their way by coloring the map of Israel according to the danger level of a specific area and by getting voice and push notification while entering a high risk area.

As a part of this project we developed a statistical-computational model based on accidents that happened in the past, which according to them the risk level of an area is calculated. The accidents data are given by the Central Bureau of Statistics and by “Ichud Hazola” organization.

The project has two parts: the first one includes development of a server side, creation of a data base which is updated regularly, implementation of the model that was developed, division of Israel map into equal areas and calculation of the risk level for each of them and construction and maintenance of communication with the application.

The second part included a development of the client side, coloring the map which is in front of the user according to the color the server sent and creation of voice notification and push notification.

As well, the server and the application are sharing a communication channel which stays connected as long as the application is in use.

The main goal of the application is to raise the awareness to the danger in the roads, let the user choose less dangerous driving routes and increasing the user vigilance and caution near dangerous places.

Another goal of this project is to create public pressure on the authorities by the application users which will hopefully cause infrastructure improvements.

This project was done as a part of Anyway project that aims to reduce the number of traffic accidents by raising the public awareness to them.

תודות

אין זה דבר טריויאלי לסייע בהצלחה ארבע שנים של לימודים במוסד מכובד זה, המרכז האקדמי לב ועל זאת אנו מודות לך'ה שנתנו לנו את הכוח והיכולת לסייע בהצלחה את התואר בכלל ואת פרויקט הגמר בפרט.

תודה למיר אבי טרייסטמן, רכזו הפרויקטם על העזרה במציאת פרויקט.

תודה לד"ר תרצה הרסטט, ראש החוג למחשבים על העזרה שהעניקה לנו במהלך התואר כראש החוג וכמרצה.

תודה רבה למנהל האקדמי, מר דן זילברשטיין, על העזרה הרבה, התמיכה המקצועית, הזמיןות, הידע הרחב שחלק איתנו והעшир לנו את הפרויקט ולהלויו הצמוד לאורך כל שלבי הפרויקט.

תודה למנהל מטעם החברה, ד"ר דניאל הרשקוביץ, על העזרה בכל שאלותינו, הזמיןות ואורך הרוח שהקדיש בפרויקט.

תודה לפרופסור אבא, ד"ר שאקי וד"ר שלומי פיקאלי, על עוזרתם בפיתוח המודל הסטטיסטי וסבלנותם לשאלותינו החוזרות ונשנות.

תודה לחברותינו מהלימודיים אשר תרמו את חלקן בפן המקצועי ובפן האישי לאורך כל הדרך.

תודה למשפחותינו וחברינו על הסבלנות, התמיכה והעזרה בלימודים ובכל.

תודה לצוות המסור של המרכז האקדמי לב, הרבענים הנהלה וצוות המדרשה, כולם תמכו לנו רבות במהלך לימודנו.

תודה מיוחדת לפסיכולוגית גברת נלי סימון שפתחה בפנינו את ביתה וסיפקה לנו סביבת לימודים תומכת ואוזן קשבת לכל אורך הפרויקט.

תוכן עניינים

| | |
|----|---|
| 3 | תקציר |
| 4 | Abstract |
| 5 | תודות |
| 6 | תוכן עניינים |
| 9 | רשימת איורים |
| 10 | מילון מונחים |
| 11 | מבוא |
| 11 | 1.1 הבעה הכלולת והבעיה הספציפית |
| 11 | 1.1.1 בעיה כוללת |
| 12 | 1.1.2 בעיה ספציפית |
| 13 | 1.2 המטרה הספציפית של הפרויקט ותרומתו לפתרון הבעיה הספציפית |
| 14 | 2. חלק מחקרי |
| 14 | 2.1 רקע תאורטי |
| 14 | 2.2 מטרות הפרויקט בחלק המחקרי |
| 17 | 2.3 מודל סטטיסטי |
| 19 | 2.4 נקודות ציון |
| 19 | 2.4.1 קואורדינטות גרפיות |
| 20 | 2.4.2 רשות ישראל החדשה ITM |
| 20 | 2.4.3 המריה בין נקודות ציון |
| 21 | 2.5 גודל ריבוע |
| 21 | 2.6 התאמת צבע לרכיבוע |
| 21 | 2.6.1 RGB |
| 22 | 2.6.2 aRGB |
| 22 | 2.6.3 HSL |
| 22 | 2.6.4 התאמת הצבע |
| 23 | 2.7 אלגוריתם המרת הקרכן |
| 24 | 3. תכנון הפרויקט המשי |
| 24 | 3.1 הגדרת דרישות |
| 24 | 3.1.1 ניתוח הדרישות |
| 24 | 3.2 אפיון המערכת בהתאם לדרישות |
| 25 | 3.3 בסיסי נתונים וקבצי נתונים |
| 25 | 3.3.1 SQLite |
| 26 | 3.3.2 קבצי נתונים CSV |
| 26 | 3.4 מערכת הפעלה android |
| 27 | 3.5 שפות הפיתוח |
| 27 | 3.5.1 צד לקוח |
| 29 | 3.5.2 צד שרת |

| | |
|----|--|
| 33 | 3.6 סביבות הפיתוח |
| 33 | cloud9 3.6.1 |
| 34 | סביבת Android Studio 3.6.2 |
| 36 | סביבות הרצה 3.7 |
| 36 | Drive Tests 3.7.1 |
| 37 | תקשרות 3.8 |
| 37 | תקשרות קיימת 3.8.1 |
| 38 | HTTP 3.8.2 |
| 39 | טכנולוגיות MQTT 3.8.3 |
| 40 | REST 3.8.5 |
| 41 | Websocket 3.8.4 |
| 44 | העברת מידע 3.9 |
| 44 | XML 3.9.1 |
| 44 | JSON 3.9.2 |
| 45 | סיכון 3.10 |
| 47 | 4. המערכת שנבנתה בפועל |
| 47 | פיתוח שירות החישובים |
| 47 | חלוקת המפה 4.1.1 |
| 48 | חישוב מיקום הריבוע ביחס לגבולות מדינת ישראל 4.1.2 |
| 48 | אתחול בסיס הנתונים 4.1.3 |
| 49 | תרשים UML למחלקות השירות 4.1.4 |
| 49 | DBFunc 4.1.5 |
| 51 | websocket 4.1.6 |
| 52 | MyServerProtocol 4.1.7 |
| 54 | פיתוח האפליקציה 4.2 |
| 54 | תרשים UML 4.2.1 |
| 55 | תקשרות websocket מהאפליקציה 4.2.2 |
| 56 | שליחת הודעות לשרת 4.2.3 |
| 57 | צביעת המפה 4.2.4 |
| 57 | הקפצת התראות בקרבת אזור מסוכן 4.2.5 |
| 58 | איובוד חבילות - 4.2.6 |
| 59 | פיתוח תקשורת בין שירותים 4.3 |
| 59 | 5. דיוון וסיכום |
| 59 | 5.1 דיוון |
| 59 | הערכת עבודה 5.1.1 |
| 60 | נקודות חזוק של האפליקציה 5.1.2 |
| 60 | פתרונות עתידיים 5.2 |
| 60 | צמצום בסיס נתונים 5.2.1 |
| 60 | הוספת מסד נתונים סביבתי 5.5.2 |
| 60 | שיפור האפליקציה הקודמת 5.2.3 |
| 60 | צביעת הכבישים בלבד 5.2.4 |

| | |
|----|---|
| 61 | 5.2.5 שילוב עם האתר האינטראקטיבי של חברת anyway |
| 61 | 5.2.6 חישוב סיקור על מסלול |
| 61 | 5.2.7 שילוב האפליקציה בענקית waze |
| 61 | 5.2.8 עדכון השירותים |
| 61 | 5.2.9 אבטחת התקשרות |
| 61 | 5.4 סיכום |
| 63 | 6. רשימת מאמרים |
| 64 | 7.ביבליוגרפיה |
| 67 | 8. נספחים |
| 67 | 8.1 צילומי מסך |
| 70 | 8.2 חבילות פרוטוקול websocket |
| 74 | 8.3 מדריך למשתמש |
| 78 | 8.4 Message sequence diagram |
| 79 | 8.5 צד לקוח - websocket |
| 82 | 8.6 צד שרת websocket |
| 82 | 8.7 קבצי javaDoc |

רשימת איורים

| | |
|---|----|
| איור 1-1 גраф מספר החרוגים בתאונות דרכים | 11 |
| איור 2-1 מידת ההשפעה של אזור מסוכן על נהיגה ובחירה מסלול | 13 |
| איור 1-2 תיאור קווי הרוחב וקווי האורך | 20 |
| איור 2-2 מפת ישראל על פי נקודות ITM ועל פי חלוקת קווארדייניות | 21 |
| איור 3-2 מרכיבי שיטת RGB לייצוג הצבע | 22 |
| איור 4-2 רצף הצבעים בשיטת HSL לייצוג הצבע | 23 |
| איור 1-3 המחשת עצם המסכים המתוכנן לאפליקציה | 25 |
| איור 2-3 תרשימים מערכות לפיקוח ממבט כללי | 26 |
| איור 5-3 באיור ניתן לראות כיצד מנוע TTS פועל | 30 |
| איור 6-3 השוואת בין טכנולוגיית websocket לעומת REST בזמן השליחה והተגובה | 46 |
| איור 7-3 תרשימים מערכת סופי | 48 |
| איור 1-4 פירוט עמודות בסיס הנתונים | 51 |
| איור 2-4 : תרשימים uml לצד שרת | 51 |
| איור 3-4 תרשימים uml לצד לקוח | 57 |

מילון מונחים

IP - מספר חד-חד-ערבי המשמש לזיהוי נקודות קצה, בשרות תקשורת שהן משתמשים ב프וטוקול התקשורת IP, כגון רשת האינטרנט.

URL - הוא רצף של תווים המבנה אחד המשמש לייצוג המיקום של דפי אינטרנט וקבצים בשרת האינטרנט.

שרת proxy - הוא שירות שתפקידו העיקרי לספק גישה מהירה למשתמשים חיצוניים בשרת מחשבים, יישם שרתים ש קופים המעבירים את ההודעות כפי שהתקבלו ויישם שרתים מפזרים המשנים את תוכן החבילה ומעבירים את ההודעה לאחר שינוי לרשות האינטרנט.

DNS - הוא פרוטוקול המאפשר גישה לבסיס נתונים מבוזר, על מנת שייחידות קצה בשרת האינטרנט יוכל לתרגם שמות תחום (Domain name) הנוחים יותר לשימוש אנושי טבעי (ה-URL) לכתובות הnumerיות האמתיות (כתובות IP) אליהן הם יפנו בזמן התקשורת.

SQL - היא שפת מחשב הצהרתית לטיפול ועיבוד מידע בסיסי נתונים יחסיים.

אמולטור - תוכנה המשמשת בחומרת המחשב ליצור סביבת תוכנה הדומה זו של מכשיר סלולרי ובכך מטעה את האפליקציה לזהות את המחשב כמכשיר חיצוני כלשהו המדמה פעולות על המכשיר.

Call Back Function - פונקציית תגובה הנקראת בעת שאירוע מסוים מתרחש.

SDK - סט של כלים לפיתוח תוכנה ולהעשרה יכולות תוכנה, בשירותי יישומים עבור פלטפורמה מסוימת.

ספרייה - היא אוסף של תת-טכנולוגיות המשמשות לפיתוח תוכנה. ספריות מכילות קוד או מידע, שמספקים שירותים לתכניות עצמאיות.

ממתק - הוא חלקה של מערכת החשוף למשתמש בה, כך שדרכו מתקיים הקשר בין המשתמש ובין המערכת. "מערכת", בהקשר זה היא תוכנה, מכונה, מכשיר או חפץ כלשהו, שנועד לשימושו של אדם.

API - מערכת של ספריות קוד, פקודות, פונקציות ופרוצדורות מן המוכן, בהן יכולים המתכננים לעשות שימוש פשוט, בלי להידרש לכתוב אותן בעצמם.

ממוצע - מספר שמחושב מתוך אוסף סופי של מספרים, ומתאר את "מרכזו" האוסף מבחינת גודל המספרים.

תוחלת - תוחלת של משתנה מקרי היא ממוצע הערכים אותם יכול המשתנה לקבל, משוקלל על-פי ההסתברויות של הערכים השונים.

שונות - היא ממד לפיזור ערכים באוכלוסייה הנתונה ביחס לתוחלת שלה.

1. מבוא

1.1 הבעה הכלולת והבעה הספציפית

1.1.1 בעה כללת

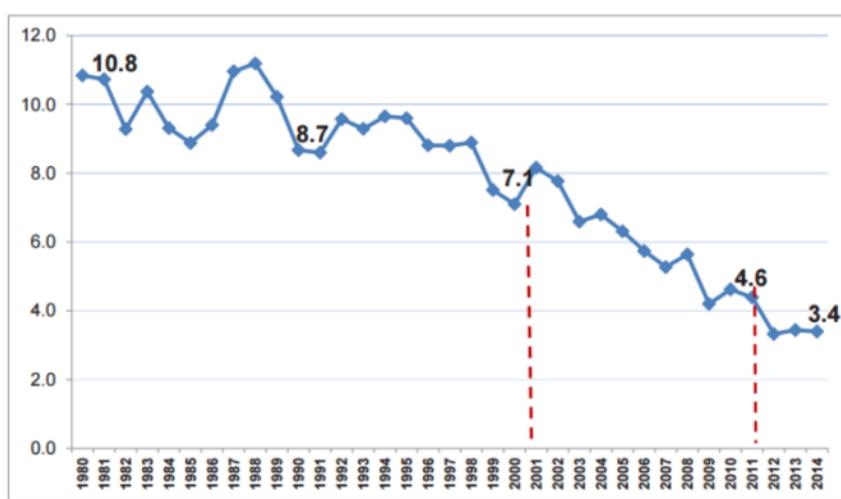
תאונות הדריכים הן בעיה כלל עולמית כאובה. על פי הערכות מידי שנה נהרגים¹ כ-2.1 מיליון אנשים* בתאונות דרכים וכ-50 מיליון אנשים נפצעים בתאונות אלו.²

עלותן הכלכלית העולמית של התאונות מסתכמת בכ- 518 מיליארד דולר בכל שנה.³ בישראל, נהגו מהתאונות דרכים מאז קום המדינה ועד ימינו למעלה מ-30,000 אנשים.⁴ אמנם ישראל נמצאת בין 5 המדינות בעלות שיעור ההרוגים הנמוך ביותר מהתאונות דרכים אך יחסית למספר כלי הרכב המהוונים בישראל שיעור ההרוגים בתאונות דרכים גבוה מברוב מדינות אירופה.

חשוב לציין כי חלק מהמחקרים והסטטיסטיקות בנושא תאונות הדריכים בישראל אינם כוללים את תאונות הדריכים שהתרחשו במחוז יהודה ושומרון.

תאונות דרכים מתרחשות בשל שלושה גורמים העיקריים: גורם אנושי- הנהג עצמו, ציוותו לחוקי התנועה, ריכוזו ותשומת לבו. גורם תשתיתתי- הכביש, תמרורים שאינם ברורים, תחזוקה לקוייה, חוסר תאורה, מפגעים בטיחותיים ועוד. הגורם האחרון הוא הגורם המכני- כלי הרכב עצמו, כשל מכני וחוסר תחזוקה. על פי נתוני ארגון הבריאות העולמי, כמחצית מנפגעי תאונות הדריכים הם משתמשי דרך פגיעים כמו הולכי רגל, רוכבי אופניים ורוכבי אופניים.⁵

הרוגים בתאונות דרכים בישראל ל-100,000 תושבים



תמונה (גרף)⁶

איור 1-1 גרף מספר ההרוגים בתאונות דרכים

* הרוג בתאונות דרכים מוגדר אדם שמת בעקבות תאונה או שמת מפצעיו בתוך 30 יום מיום התאונה

2.1.1 בעיה ספציפית

בישראל עצמה ישנים למעלה מ-350 ק"מ של כבישים המוגדרים כยอม ככבישים אדומיים - קטעי כבישים ביין עירוניים אשר נרשם בהם מספר של תאונות דרכים קשות הגובה ממוצע מספר התאונות בקטעי כביש אחרים ברוחבי הארץ⁷. על פי נתוני עמותת אור ירוק לא ניתן להציג על אזור מסוים בארץ המשופע בקטעי כביש אדומיים.

למעשה, עד לסוף שנת 2012 אורך הכבישים שהוגדרו כאדומיים היה 700 ק"מ אך בעקבות החלטה של ראש אגף התנועה במשטרת ישראל להתמקד בכבישים המסוכנים יותר ולמנוע פיזור של המאמרים למזעור מספר התאונות, הורדו ממפת הכבישים האדומים כמחצית מהכבישים אשר היו מוגדרים ככבישים אדומיים⁸.

ኖסף על כך, מספר כלי הרכב הממוגעים בישראל גדל בצורה מתמדת. נתוני המינוע בישראל מראים כי ישנים שלושה מיליון כלי רכב, מדובר על גידול של ארבעה אחוזים בשוק הרכב לעומת גידול של שני אחוזים באוכלוסייה יחסית לשנת 2014. רק בחודש ינואר האחרון, עלו 45 אלף מכוניות חדשות על כבישי ישראל, זהו זינוק של 24% לעומת ינואר אשתקד.

בישראל, ישנים כ-400 כלי רכב לכל אלף תושבים לעומת ארצות הברית יש כ-800 כלי רכב לכל אלף תושבים, באירופה ישנים 565 בממוצע וביפן 544.

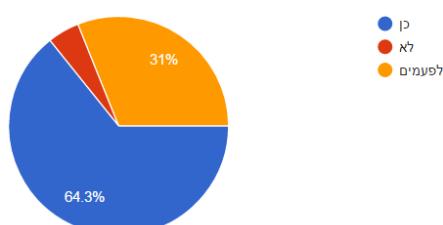
הגידול במספר הנהגים וכלי הרכב על הכביש משפייע בצורה ישירה על מספר תאונות הדרכים.

חטיבות סימון ופרסום הכבישים האדומים כמו גם כבישים מסוכנים אחרים גדולה: היא משתמשת כהתראה לנוהגים הנוסעים בכבישים אלו לנוכח במצבה זהירות ומצביעה על נקודות תורפה הדורשות מצד המדינה ורשותה מאמץ מוגבר בשיפור תשתיות ובהגברת האכיפה במקומות אלו. על פי מחקר אותו ערך בנק ישראל עולה כי שיפור התשתיות בכבישים מוגריד באופן דramatic את מספר תאונות הדרכים: כל הוסף מטר של שלולים לכביש מפחיתה את הסיכון לתאונות דרכים בכ-15% והוספת הפרדה בין נתיבים מנוגדים מפחיתה את מספר התאונות ב-32%⁹.

مسקר שערכנו וכלל 277 נשאלים בעלי רישיון נהיגה, ענו 91% מהנשאלים כי נהיגתם משתנה כתוצאה מקרבה לאזור **הידען להם** כמועד לתאונת וכן 87% מהנשאלים הודיעו כי נהיגתם משתנה באופן כלשהו בעקבות **שילוב** על אזור מסוכן. ככלומר ניתן לראות בבירור בשני המקרים שמודעות לסכנה יוצרת שינוי בconduct הנהיגה אצל הרוב המוחלט של הציבור.

בנוסף, המסקן כלל 79 נשאלים ללא רישיון נהיגה, במהלךו ענו 80% מהנשאלים כי אופן ההליכה שלהם משתנה כאשר הם מתקרבים לאזור הידען להם מסוכן.

האם נהיגתך משתנה בעקבות הودעה/שלט על אזור מסוכן? (277 responses)



בתכנון מסלול הנסעה, עד כמה בחירת המסלול מושפעת מעבר במקומות מועדים לתאונה
(277 responses)



איור 2-1 מידת ההשפעה של אזור מסוכן על נהיגה ובחירה מסלול

2.1 המטרת הספציפית של הפרויקט ותרומתו לפתרון הבעיה הספציפית

מטרתו של פרויקט זה היא לפתח על גבי אפליקציה קיימת צביעה של מפת ישראל על ידי צבע שימחיש את רמת הסיכון בכל אזור. בנוסף, נרצה לאפשר בעת השימוש באפליקציה התראה קולית והתראות דחיפה אשר ניתנת לצפייה על ידי גלילת המסך כלפי מטה- למשך התראות. כאשר משתמש נכנס לאזור המוגדר כמסוכן האפליקציה תתריע על כך בצורה קולית וגם על ידי הצגת התראות דחיפה. ביציאת המשתמש מסוכן המוגדר כמסוכן תשמע התראה קולית בלבד המודיעיה על כך.

תרומות הפרויקט לביעית תאונות הדרכים שתוארה לעיל מתבטאת בכמה מישורים: הגברת הזהירות בנהיגה והליכה סמוך למקומות מסוכנים. הגברת המודעות למקומות כאלה: הצגת אזורים עם רמות סיכון גבוהה בצורה ויזואלית אשר ממחישה את הסכנה במקומות אלו ויכולת להוביל לכך ציבורי לשיפור הבטיחות באזוריים אלו (על ידי הגברת האכיפה או תיקון ליקויים בכביש).

תרומה נוספת לביעיה זו יכולה להיות בחירה של המשתמש במסלול הליכה או במסלול נהיגה אחר אשר אינו מכיל אזורי מסוכנים. באותו הסקר המובא לעיל שערכנו, 204 מהנשאלים (למעלה מ57%) אמרו כי בחירת מסלול הנהיגה או ההליכה שלהם מושפעת באופן כלשהו ממוקדות המועדים לתאונה.

הגברת המודעות למקומות אשר מועדים לתאונות דרכיים יכולה להשפיע על בחירת מסלול הנהיגה או ההליכה של המשתמש באפליקציה ולהפוך את הדרך לבטוחה יותר עבورو.

חשוב לציין כי בעוד שתאונות הדרכים אשר מתרכחות ביוזדה ושומרון אין נכללות ברוב המהקרים והסטטיסטיות על תאונות הדרכים אפליקציה זו כוללת תאונות דרכיים אלו. בנוסף, מפת הכבישים האדומים מתיחס רק לכבישים בין עירוניים ולא קיים מיפוי הממחיש בצורה ויזואלית את הכבישים העירוניים המסוכנים.

2. חלק מחקרים

2.1 רקע תאורטי

כיוון, קיימת מפה כללית של מדינת ישראל בהש不适ה הלשכה המרכזית לסטטיסטיקה אשר מפרטת תאונות דרכים עם נפגעים בלבד. מפה זו נגישה אך ורק מדפדפן 7 ו internet explorer ומעלה ונינה נגישה כל מmachshirim סלולריים¹⁰. נוסף על כך, משתרת ישראל מתוחזק באתר המפות המשלתי מפת כבישים אדומים ומצלמות.

קיימות מפות נוספות להשלכה המרכזית לסטטיסטיקה אך הן אינן דינמיות, לא ניתן

להתקרב/ להתרחק מאזורים מסוימים.¹¹

כל המפות האלו אינן אינטראקטיביות ולא נגישות כלל: הרצת חיפוש במנוע חיפוש של המילים "מפת כבישים אדומים" או "מפת תאונות דרכים" לא מניב בתוצאות הראשונות של החיפוש קישורים למפות אלו. נוסף על כך, במפת הכבישים האדומים אין מידע על כבישים אשר אינם מוגדרים כ"אדומים" אך הם עדין מסוימים.

מפה נוספת בה ניתן לצפות בתאונות ונתוניה הנה מפה אינטראקטיבית שיצרה בפרויקט anyway אשר במסגרתו אנו מבצעות פרויקט זה. פרויקט anyway הוקם במטרה להעלות את המודעות לתאונות הדרכים בארץ על ידי מיפוי כל תאונות הדרכים, כולל תאונות דרכים ללא נפגעים, משנת 2005 וסיווגן. על המפה ניתן ליצור דיוונים בין המשתמשים אודוט מקומות מועדים לתאונות או מסוימים. כמו כן, ניתן לקבוע קритריונים לחיפוש תאונות כמו תאריך, חומרה, סוג הדרך, מזג האוויר, פni הכביש, סוג התאונה, אזור גאוגרפי ועוד. הפרויקט זמין באתר אינטרנט וגם באפליקציה למכשירי android. נתוני התאונות מגיעים מהלשכה המרכזית לסטטיסטיקה ומארגן איחוד הצלחה.

לעומת אור יロー, עמותה הנابעת בתאונות הדרכים בישראל ישנה מפה אינטראקטיבית המאפשרת דיווח על תאונות דרכים ומפגעי דרך או כביש. הדיווחים למפה מגיעים ממשתמשי האתר העומת. המפה אינה מוצגת כראוי באתר: הסימונים של תאונות הדרכים או המפגעים מופיעים אך המפה עצמה חסраה. נוסף על כך, הדיווח נעשה על ידי המשתמשים כך שהוא חלקו ולא דוקא מדויק.¹².

בארצות הברית יsono מיזם בשם metrocosm אשר מציג את מפת ארצות הברית עם נתוני התאונות מעשר השנים האחרונות. התאונות ניתנות לחיפוש ע"י כתובת או בסיכון על פי חומרת התאונה, משתמשים וגורמים אשר תרמו לתאונה.¹³ מפה דומה אם כי מכילה פחות אינפורמציה קיימת גם באנגליה.¹⁴ ישנן מפות נוספות כדוגמת Roads kill המצינית עברו רוב המדינות כמו הרוגים בתאונות ישנים לכל מאה אלפי אנשים ומציגה עוד סטטיסטיקות בנושא וכן מפנה למאמריהם הקשורים בתאונות דרכים במדינה ספציפית.¹⁵.

2.2 מטרות הפרויקט בחלק המחקרי

מטרתו של פרויקט זה כפי שציינו לעיל הוא הלאת המודעות לככיבים מסוכנים והפחחת תאונות דרכים. לצורך כך, חיפשנו מאמרים המגדירים את אופן היישוב מסוכנותו של כביש או דרך על מנת שנוכל לכתה ולהציג בצורה ויזואלית את הסכנה בכביש. בנוסף לכך, חיפשנו מאמרים החוקרים את הקשר בין תאונות וחסיפה ומודעות אליהן על מנת להיות בטוחות כי חסיפה ומודעות לתאונות דרכים יכולות אכן לעזור במניעת תאונות מעבר ליצירת לחץ ציבורי לשיפור תשתיות והחלפת מסלולי נסיעה או הליכה. ישנו אינסטינקט בסיסי שהודעה או התראה על סכנה יכולים לעזור במניעת שכן לא לחינם מוצבים שלטי אזהרה ושלטי התראה כגון "האט" ו"עצור" בצד הכביש אך רצינו בכל זאת לחפש סימוכין ואישורים להרגשה זו.

בחיפושינו אחר מאמרים בנושא מודעות וחסיפה לתאונות (באופן כללי) נתקלנו במאמר ACCIDENTS WILL HAPPEN. DO SAFETY SYSTEMS IMPROVE WAREHOUSE SAFETY PERFORMANCE? אשר ערך מחקר על תאונות במחסנים ובמפעלי עבודה בהם שיעור תאונות המלגזה מהויה למעלה מ-40% מתאונות העבודה. המחקר שנעשה התמקד בין השאר על תודעת העובדים לבטיחות ומודעותם לסכנה ואזורייה. ההשערה של החוקרים הייתה כי מודעותם לשכנה משפיעה באופן חיובי על הבטיחות. תוצאות המחקר, שנערך במלואה 50 מחסנים וככל כ-3,000 עובדים, אכן איששו את הטענה ואף יותר מכך, על פי החוקרים: "**הגולם הלאומי המשפיע על בטיחות וביצועים בטוחים הוא תודעת העובדים לבטיחות ומודעותם לשכנה**". נוסף על כך מצינוים החוקרים כי הצבת שלטים באזוריים מודעים לשכנה אכן מפחיתה תאונות. אם כן, בהשלכה על תאונות הדרכים ניתן לומר כי הعلاות המודעות לתאונות דרכים ואזוריים מסוכנים אכן יכולה להשפיע באופן חיובי על שיעור תאונות הדרכים ולעזר בהפחחתן.

עקבנו אחר מאמרים נוספים אשר עוסקו במודעות לתאונות דרכים דווקא כמו המאמר of classification driving exposure and accident rates for highway safety analysis החסיפה הנה גורם המשפיע על שיעור התאונות. המושג "חסיפה" במאמר מוגדר בכמה הגדרות שונות ביןין מספר הפעמים הפעמים שסכנה קורתה בהינתן מרחק נסיעה כלשהו ואינו חד משמעי. בשל כך ולאחר קרייה מדויקת של המאמר בחרנו שלא להביא את שלל ההגדרות למושג חסיפה שניתנו בו אך כן להסתמך על האמירה שבו כי אכן לחסיפה יש השפעה על שיעור תאונות מאחר והוכחה לכך במאמר אינה תלויות בהגדרת המושג חסיפה.

מאמר נוסף אותו קראנו ובחרנו להביא לצורך מחקר זה הוא on exposure and accident rate. במאמר טוען ומוכיח הכותב כי שעור תאונות הדרכים קשור באופן ישיר לחסיפה לתאונות הדרכים ואף יותר מכך: ככל שהחסיפה אליהן בפרק זמן מסוים עולה כך יורד שעור תאונות הדרכים. בנוסף, כותב המאמר מסביר כי שימוש בפונקציה לינארית להערכת רמת הסיכון או על מנת לחזות תאונות אינו יעיל מאחר ופונקציה

לינאריות עלולה לשנות את תוצאה רמת הסיכון אף על פי שלא נעשה שום שינוי (תחרורתי - בטיחותי) בכביש.

An Algorithm for Assessing the risk of Traffic Accident. מאמר זה מבוסס על עבודה מחקר אשר נעשתה בהונג קונג לגבי תאונות הדרכים ובמהלכו פותח אלגוריתם אשר מעריך את רמת הסיכון לתאונות באזורי כביש.

בחרנו להשתמש במאמר זה כבסיס עיקרי לבניית המודל החישובי בפרויקט זה מאחר וננוינו, כפי שיפורטו בהמשך וכן מטרותיו של המחקר והמאמר דומות לנתחים הידועים לנו ולמטרותינו בפרויקט זה. בנוסף, אלגוריתם זה נוסה והוכח כיעיל ויציב והוא גמיש ופתוח לשינויים.

תחילת נסקור בהרחבה את האלגוריתם אליו הגיעו החוקרים ובמהלך נפרט על המודל החישובי בו השתמש על מנת לקבוע את רמת הסיכון בכביש בהתבסס על מחקר זה.

במאמר הנ"ל נעשה שימוש בשיטות סטטיסטיות על מנת לנתח את תאונות הדרכים וכן בעזרת מערכת מידע גאוגרפיה המציגה נתונים על גבי מפה בצורה ויזואלית. נוסף לכך ישנו עוד שני בסיסי נתונים, האחד מכיל מידע אודוט תאונות דרכים (המחלקות לשולחה קטגוריות: קטלניות קשות וקלות) ובבסיס נתונים אשר ממנו נגזרים גורמי סיכון פוטנציאליים (כגון: סביבה כפרית, סביבת בית ספר, כביש ראשי וצדומה). מטרותיו העיקריים העיקוריים של המאמר והמחקר הן "לייסד מודל הערכה לתאונות דרכים... לזהות גורמים בעלי השפעה מכריעה על תאונות הדרכים... ולזהות אזוריים בהם ישנו סיכון גדול לתאונות...".

לפניהם שמשיך בהסביר המודל הסטטיסטי ברצוננו לפרט על הפונקציות, הרגression וההתפלגות הסטטיסטיות בו ולפרט עליהם מעט על מנת שהמודל אותו נציג יהיה ברור יותר:

$$\text{פונקציית גמא} - \text{פונקציה מרוכבת המרכיבת את המושג עצרת לכל המישור המרוכב.}$$

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$$

רגסיה לינארית - שיטה מתמטית לניטוח מוגדים סטטיסטיים בה נמצאים הפרמטרים של הקשר בין משתנה בלתי תלוי X למשתנה תלוי Y כך: $Y = aX + b$

התפלגות ברנולי - התפלגות זו היא התפלגות של משתנה מקרי אשר לגביו ניתן לומר שישנם שני מצבים: הצלחה או כישלון.

התפלגות בינומית - התפלגות המתארת את מספר ההצלחות בסדרה בלתי תליה של מניסויי ברנולי בלתי תלויים והסתברות לקבלת k ההצלחות מתוך n הניסויים אלו.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\text{כך שמתקיים: } \binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

התפלגות פואסון - ההסתברות מתקבלת כאשר סופרים אירועים נדירים שהקצב שלהם עומד על גם המתרחשים בפרק זמן קבוע. למעשה, ניתן להתייחס להסתברות זו כהתפלגות בינומית כך שמכפלת הניסויים בסיכויי ההצלחה בכל ניסוי נשארת קבועה (כלומר λ) ומספר הניסויים שואף לאינסוף.

התפלגות בינומית שלילית- התפלגות בדידה המתארת את מספר ההצלחות בסדרת ניסויי ברנולי בלתי

תלוילים המתרחשים על פי מספר נתון מראש, z , של כישלונות.

$$P_X(k) = \binom{r+k-1}{k} (1-p)^r p^k$$

התפלגות גמא- התפלגות דו פרמטרית היכולה לתאר מודל לティאור זמן ההמתנה עד לאירוע מסוים.

במאמר זה, כמו במאמר נוסף שהוא לעיל תומכים החוקרים בדעה כי שימוש בליינאריות ובפרט ברגסיה ליינארית אינה מתאימה על מנת להסיק מסקנות לגבי מקרי תאונות דרכים. לעומת זאת, התפלגות פואסונית והטפלות ביןומית שלילית מוצעות כפתרון טוב להערכת מקרי תאונות דרכים מאחר וה"מקרה" המדובר هنا נדיר ורנדומלי. במקרים מסוימים כמו כאשר הממוצע חורג מגבולות השונות החישוב עלול להיות מוטעה ועל כן הוסיפו החוקרים במאמר זה הנחה נוספת החלה על הממוצע לפיה הוא מתפלג בהטפלות גמא. כלומר: נניח כי מספר תאונות הדרכים מתפלג פואסונית עם ממוצע צפוי המתפלג על פי גמא.

בניתוח הנתונים בפועל, השונות גדולה מהממוצע ולכן כותב המאמר השתמש לティור הנתונים במונח

הסטטיסטי *overdispersion*. ככלומר, לא ניתן למדוד מהממוצע ומהשונות על טווח הפיזור של הנתונים.

רגסיה ביןומית שלילית הנה שימושית במיוחד במקרה של *overdispersion* ואך נמצאה יعلاה יותר מרגרסיות פואסון על פי החוקרים.

שלביו העיקריים של האלגוריתם כפי שהם מפורטים במאמר הם כדלקמן :

1. **מייזוג בסיסי נתוניים**- בשלב זה מיובאים נתונים תאונות הדרכים מבוסיס הנתונים וממוקמים על גבי

המפה הדיגיטלית. לאחר מכן נספרות, בעזרת SQL, תאונות הדרכים אשר הובילו למקרי מות ולסיום נערך מייזוג בין מספר תאונות הדרכים והנתונים על גורמים פוטנציאליים מסוימים. בעת, בסיס הנתונים הממזוג מהויה בסיס לשאר השלבים.

2. **מקבצי מידע**- החוקרים במאמר הגדרו ניתוח מקבצים טכנית אשר מקבצת פרטיים יחד על

בסיס דמיון או שוני. במחקר זה, אוחדו המקבצים כך שאחדותם הפנימית תהיה גבוהה ואחדותם החיצונית נמוכה מאוד.

3. **כיוול המודל**- במודל זה, המשנה הבלתי תלוי הוא מספר התאונות וailio הגורמים המסכנים הפוטנציאליים מסוימים כמשתנה תלוי. מתוך כך, חישבו החוקרים את ההסתברות לתאונות

דריכים כך :

$$P(Y = y) = \frac{\Gamma(y + \alpha^{-1})}{y! \Gamma(\alpha^{-1})} \left(\frac{\alpha \lambda}{1 + \alpha \lambda} \right)^y \left(\frac{1}{1 + \alpha \lambda} \right)^{\alpha^{-1}}$$

כך שüber $\sigma = \sqrt{\alpha}$ הוא הפרמטר של overdispersion. המשטנה σ ממחווה את מספר תאונות הדרכים והמשטנה λ הינו הממוצע הצפוי של מקרי תאונות הדרכים. לאחר וההנחה היא שההתפלגות היא בינומית

$$\lambda + \alpha \lambda^2,$$

נקודה אחת נשארה פתוחה במאמר זה והוא אופן חלוקת האזוריים באלגוריתם. במאמר, בחרו המחברים להשתמש בכללי המקבצים (clustering) של GIS. אך מאוחר ואיננו מעוניינות שפונקציית חישוב הסיכון תחשב בנפרד עבור אזוריים מרובי תאונות דרכיים ואזוריים שלא התרחשו בהם תאונות דרכיים אלא שהחלוקת תהיה באופן שוווני בחרנו לחלק את המפה כולה למרובעים שוויים בגודלם.

2.3 מודל סטטיסטי

על בסיס מאמר זה ובהתייעצויות עם מומחים לסטטיסטיקה מטעם פרויקט anyway וمراقبים לסטטיסטיקה והסתברות במרכז האקדמי לב בנינו את המודל הסטטיסטי עבור פרויקט זה. חלוקת האזוריים, כפי שהציגו לעיל, נעשית בצורה שווה כלומר, המפה חולקה לריבועים בגודל שווה כך שאורך הצלע הוא a ומתווך כך שטח הריבוע הוא a^2 קבוע. כתוצאה לכך, חישוב ההסתברות מוצטמת לנוטונים היודיעים לנו אך ורק על גבי שטח ריבוע זה כלומר: הממוצע הוא ממוצע מספר תאונות הדרכים שהתרחשו באותו ריבוע. וכן שאר הפרמטרים המרכיבים את חישוב ההסתברות כפי שפורטו לעיל, מתייחסים לריבוע ספציפי בלבד. התאמה נוספת ערכנו בין המודל המוצע במאמר לבין המודל אותו פיתחנו מתייחס לשלב הראשון של האלגוריתם הניל: מיוגג בסיסי נתונים. במאמר, בסיס הנתונים של תאונות הדרכים ממוגע עם בסיס נתונים המכיל מידע על גורמים פוטנציאליים לתאונות דרכיים כמו: אזור תעשייה, עבודות בכביש, אזור עם גני ילדים ובתי ספר, פארקים ציבוריים ועוד. בחרנו שלא להכניס מיוגג זה אל האלגוריתם שלנו מכיוון שהוא מושך ומטrhoת פרויקט זה הוא תאונות דרכיים ומণיעתן, ללא קשר לסביבה.

מעבר לכך שמיוגג בסיס הנתונים עם גורמים פוטנציאליים חורג מגבולות הפרויקט, יצרת בסיס נתונים שכזה כך שהיא עדכנית, ומיוגג שלו עלול לקחת משאים רבים ולהשפע על זמן תגובה האפליקציה. בסיס הנתונים שהוצע במאמר לא נמצא בשימוש ציבורי ומשמש גוף ממשלתי כך שזמן תגובה האפליקציה ואופן תדריות העדכון אינם בראש סדר העדיפויות של מפתחי האלגוריתם.

להלן האלגוריתם אותו פיתחנו בהתבסס על האלגוריתם אשר הוצע במאמר זה:

א. חלוקת המפה לריבועים בגודל שווה

ב. הכנסת תאונות הדרכים לתוך בסיס נתונים:

1. עברו כל ריבוע:

1.1 אם תאונת הדרכים קלה: התעלם.

1.2 אם תאונת הדרכים קשה:

1.2.1 אם התאונה התרחשה לפני שנת 2010 : התעלם.

1.2.2 אחרת : הוסף 1 למספר התאונות בשנה זו.

1.3 אם תאונת הדרכים קטלנית :

1.3.1 אם התאונה התרחשה לפני שנת 2010 : הוסף 1 למספר התאונות בשנה זו.

1.3.2 אחרת : הוסף 2 למספר התאונות בשנה זו.

כאשר ספירת התאונות מתבצעת כך, אנו מושגות מספר דברים: עדכניות- אופן הספירה מבטיח שההתאונות אשר התרחשו לפני שנת 2010 ישפיו פחות על ההסתברות שכן ייתכן ומazel היבש הפך לבטיחותי יותר וסבירתו השתניתה. מידתיות- כאשר ספירת תאונות הדרכים מתבצעת כך ניתן משקל רב יותר לתאונות דרכים קטלניים ביחס לתאונות קשות. מיקוד בתאונות קשות וקטלניות- תאונות קלות אין נספורות כך. כתוצאה לכך, ניתן לוודא כי אין בו ההסתברות לתאונות גבואה הוא איזור מסוכן בהחלט. כלומר, יידאנו שאיזור אשר קרו בו שיש תאונות דרכים קלות לא יוצג כמסוכן כפי שיוצג איזור עם שתי תאונות קטלניות.

ג. חישוב הסתברות לכל ריבוע:

חישוב ההסתברות לתאונות דרכים בכל ריבוע מתבצעת על פי החישוב אשר הוצע במאמר:

$$P(Y = y) = \frac{\Gamma(y + \alpha^{-1})}{y! \Gamma(\alpha^{-1})} \left(\frac{\alpha \lambda}{1 + \alpha \lambda} \right)^y \left(\frac{1}{1 + \alpha \lambda} \right)^{\alpha^{-1}}$$

כך ש \bar{x} הוא מספר התאונות בריבוע זה, המשתנה \bar{X} הממוצע הצפוי של מקרי תאונות הדרכים והוא הפරמטר של overdispersion. השונות והממוצע נשרו כפי שהוגדרו לעיל, באלגוריתם מהמאמר.

אם כך, נוכל לחשב את ממוצע תאונות הדרכים בהתפלגותן על פי שניים כך:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$
$$S_x^2 = \sum_{i=1}^n (x_i - \bar{x})^2$$

ולאחר שהממוצע נמצא נוכל לחשב גם את השונות על פי נוסחתה:

$$\alpha = (S_n^2 - \lambda) / \lambda^2$$

2.4 נקודות ציון

2.4.1 קואורדינטות גרפיות

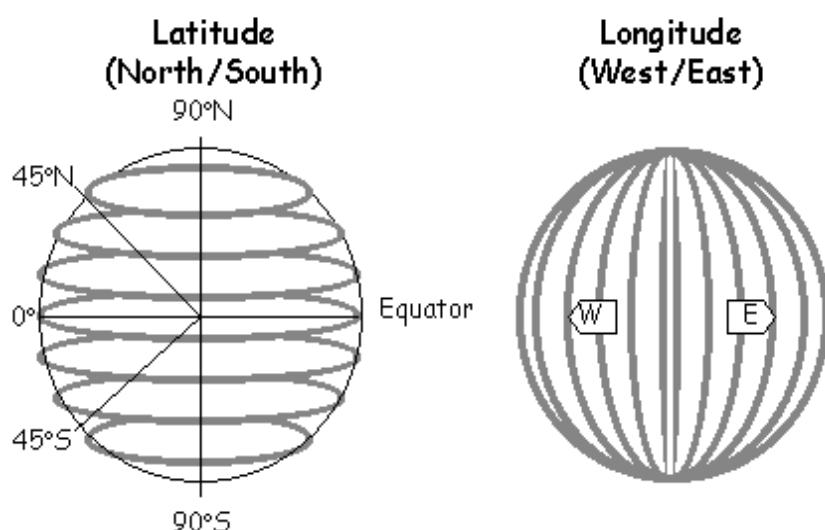
קואורדינטות גרפיות הן נקודות ציון לזיהוי מדויק של מקום על פני כדור הארץ. תוצאות הפרויקט יהיו מוצגות בסופו של דבר באפָן גרפי על גבי מפה והשימוש בקואורדינטות גרפיות יועיל להצגה עילית אינטגרלית יותר של התוצאות.

הקוואורדינטות הגרפיות מتبسطות על קו האורך והרוחב של כדור הארץ **ביחידות** של מעלות וניתן לבטא אותן גם בצורה עשרונית. הקואורדינטות נרשומות כצמד של מספרים : (latitude,longitude) כך שwidth (latitute,longitude) מצין את קו האורך.

מתייחס לקו הרוחב ואילו longitude מצין את קו האורך. עברו Latitude המתייחס לקו הרוחב נקרא גם באות היוונית φ או כפי שנקרה לו בהמשך בקיצור lat. עברו נקודה מסויימת על פני כדור הארץ, ערך lat שלה הוא הזווית הנוצרת בין קו המשווה לקו ישר העובר באותו נקודה ודרך או קרובה למרכו כדור הארץ. ערך הזווית נע בין °-90 (בקוטב הדומי) ל ° 90 בקוטב הצפוני.

החלק השני של צמד המספרים, Longitude נקרא בקיצור long/lnq והאות היוונית שלו היא λ. ערך ה long של נקודה הוא הזווית המזרחית או המערבית בין קו אורך לעוד קו שעובר דרך אותה הנקודה. המקבילה של קו המשווה בחישוב long היא קו גרייניצ' אשר מזרחה ממנו הזווית חיובית ומערבית ממנו הזווית שלילית. הערכים נעים בין °-180 ל °+180.

הערכים ניתנים במלצות וכל מעלה מחולקת ל 60 דקות ומסומנת בתג אחד ' וכל דקה מחולקת ל 60 שניות ומסומנת בשני תגים " לדוגמה, נקודות הציון של ירושלים לדוגמה היא E°0'31°N 35°47' ומיוצגת עשרונית (31.783333, 35.216667).¹⁶

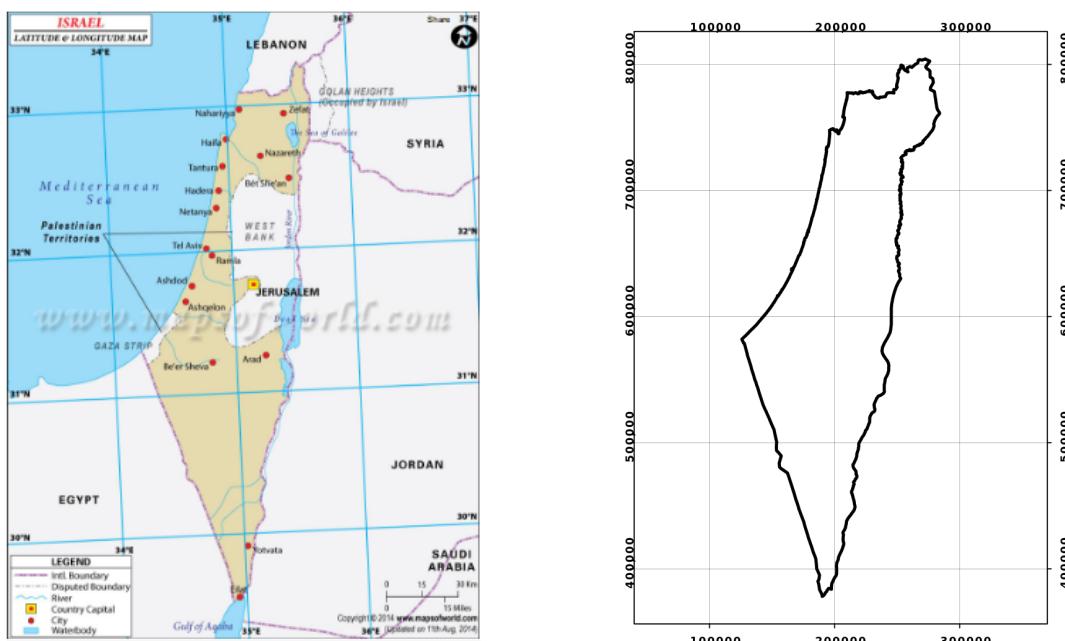


איור 2-2 תיאור קווי הרוחב וקווי האורך

2.4.2 רשות ישראל החדשה ITM

רשות ישראל החדשה, Israel Transverse Mercator, היא רשות קווארדיינטוטים ישראלית זווית המבוססת על היטל מוקורי (היטל גלילי שומר זווית אשר נוצר על ידי "עטיפת" כדור הארץ בגליל, השלقت כל נקודה מהצד ופריסת הגליל כך שנוצרת מפה מלכנית שקווי האורך והרוחב בה מאונכים זה לזה¹⁷) ועל משטח הנוצר מסיבוב של אליפסה GRS80 (אליפסה המציגת שיטה לחישוב נקודות ציון על פני הארץ¹⁸) סביב אחד מציריה.

ניתן להשתמש ברשות ישראל החדשה על מנת למפות כל נקודה בשטחה של מדינת ישראל¹⁹.



איור 2-2 מפת ישראל על פי נקודות ITM (ימין) ועל פי חלוקת קווארדיינטוטים (שמאל)

2.4.3 המרה בין נקודות ציון

המרה בין נקודות הציון נעשית ברוב מוחלט של המקרים על ידי תוכנות או ספריות מובנות מאחר והוא כוללת מאות חישובים טריגונומטריים מורכבים.

המרה מנקודות ציון מסווג ITM לקווארדיינטוט גרפיות ולהפוך נעשית באמצעות תרגום של הנקודות לאלייפסואיד או מאלייפסואיד. ההמרה נעשית על ידי מניפולציה של הנתונים הבאים: קו הרוחב במפה ישראלי החדש ביחס לקו המרידיאני, קו האורך הקבוע במפה ישראלי החדש ביחס לקו האורך הקבוע קו גריינץ' וערכי easting false และ northing false (ערכים אלו "ממירים" את טווח הנקודות של הצפון וה动员 להווח נקודות רחבות יותר כך שערכיהם אינם שליליים).²⁰

2.5 גודל ריבוע

גודל הריבוע אשר מוגדר כאזור סיכון נדרש להיות גדול מספיק על מנת שלא ליצור עומס על בסיס הנתונים על התקשרות, על האפליקציה והמשתמש באם הריבועים יהיו קטנים מידי המשמש לקבל התראות לא מדויקות ביחס למקומו ומהירות נסייעתו. אך גם קטן מספיק כך שלא יוכל יותר מכמה רחובות ויאנה על ההגדלה "אזור". בשל כך, ולאחר ניסיונות רבים שערכנו נקבע רוחב הריבוע, אותו סימנו באות α , להיות 0.0013 ואורך הריבוע, המסומן באות β , הוא 0.0018 .

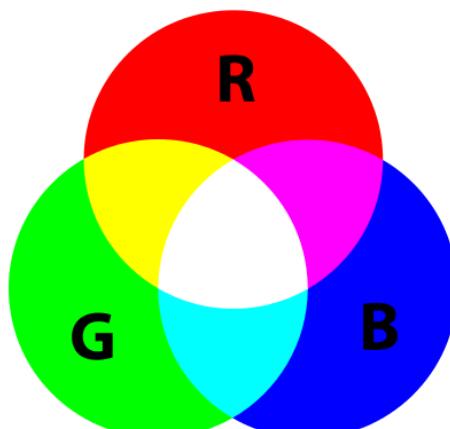
אורךו ורוחבו של הריבוע אמנים זחים אך ויזואלית על המפה הצורה שמתקבלת היא ריבוע. הסיבה לכך נעוצה בהפרש בין קווי האורך והרוחב של כדור הארץ אשר קטנים ככל שמרת欢קים מקו המשווה ומקו גרייניצי: אם ניצור מצולע שכל צלעותיו שוות הוא יהפוך למעין טרפז בחלקו העליון. בפועל, רוחב כל ריבוע עומד על 145 מטרים בקרוב ואורך כל ריבוע הוא 169 מטרים כך ששתחים של ארבעה ריבועים עומדים על דונם בקרוב.

2.6 התאמת צבע לריבוע

אחד מהתוצאות הסופיות של פרויקט זה הוא צביעת הריבועים במפה. התלבטנו רבות לגבי השיטה לייצוג צבע בו נשתמש לצביעת הריבועים ואפננו ההתאמנה של הצבע לרמת הסיכון בריבוע. בחרנו בסופו של דבר להסתמך על מודל RGB אך גם לו ישנםimplementations diverse :

RGB 2.6.1

תchosht הצביע הנוצרת כתוצאה מצפיה במסך דיגיטלי כלשהו נגרמת מאיסוף האור הנפלט שלושה סוגים פיקסלים: אדום כחול וירוק ומכאן גם נובע שמה של השיטה: eRed Green Blue. קביעת העוצמות של מרכיבי הצבע השונים יוצרת את הצבעים. קביעת העוצמות של מרכיבי הצבע השונים נעשית על ידי השמות המספריים השלמים בין 0 ל 255 לכל אחד מן המשתנים G, R, B, כאשר 0 פירושו לא להאיר בכלל את הנקודה, ו-255 מסמל האירה במלוא העוצמה. לאחר וכל צבע ניתן לייצוג ב-256 אופנים וישנם שלושה צבעים הקובעים את ייצוג הצבע ניתן להשיג 16,777,216 צבעים שונים.²¹



איור 3-2 מרכיבי שיטת RGB לייצוג הצבע

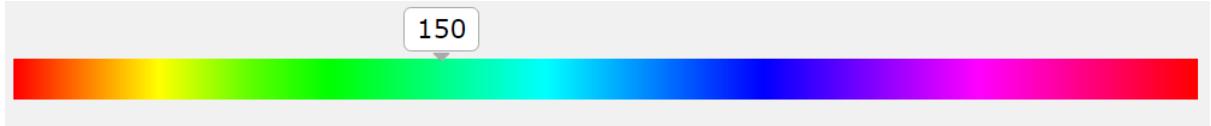
aRGB 2.6.2

שיטת צביעה זו מתבססת על שיטת RGB ומוכרת גם בשם ARGB. בשיטת ייצוג צבעים זו, RGB נשאר כפי שהוא ביצוגו המקורי ונוסף פרמטר α המתייחס לרמת אטימות הצבע. ערך זה נע בין 0 ל-100 כך שעבור 0 הצבע יהיה שקוף ובערך 100 הצבע יהיה שחור. שיטה זו דואגת מתאימה לצורכי הפרויקט שלנו מאחר ואנו לא רוצות שצביעת הריבוע תסתר לחלוון את המפה אלא רק שהצביעת תתווסף שכבה מעלייה כלומר, צביעת המפה תעשה בצבע שקוף בrama כלשהי²².

HSL 2.6.3

HSL מהוות למעשה הצגה של גליל הקואורדינטות הנמצאות במודל הצבעים RGB. הצגה באופן כזה מהוות למעשה סידור מחדש של גאומטריות הצבעים RGB בצורה יותר אינטואטיבית. גם במודל HLS בכל אות מייצגת מופיע בלבדו שהערך שיווכנס אליו ישפיע על הצבע המתkeletal. האות H אחראית על הגוון (ולכן נקראת hue) האות S אחראית על הרווחה (saturation) והאות L מתאפיינת בחירות (lightness) או (luminosity)

הציגת של הגוונים בהציגת HSL נוחה במיוחד מאחר ואנו רוצות לצבוע את המפה בצבעים הנעים בין אדום לירוק בהתאם לרמת הסיכון. הצגה של הצבע האדום היא $H=0$ ואילו הצגה של י록 מתkeletal עבור $H=150$ כך שברצף בין 0 ל 150 נמצאים כל צבעי הביניים המתאימים לצביעת ריבועי האפליקציה.



איור 4-2 רצף הצבעים בשיטת HSL לייצוג הצבע

2.6.4 התאמת הצבע

התאמת הצבע לריבוע בהתבסס על ההסתברות אשר תחושב לעיל ייעשה כך:
החלטנו כי הערך אשר ייצג את המקסימום הירוק יהיה 85 (מאחר וכל טווח הירוקים בין 85 ל-150 "ngeuu")
בגוון כחול). חישבנו על פי האלגוריתם המופיע לעיל את ההסתברות למקרה הקל ביותר של תאונה כלומר
עבור מערכ בוי ישנה תאונה אחת בלבד בשנת 2005 (למעשה, בשל אילוצי האלגוריתם זהה תאונה קטלנית)
ההסתברות עבור מערכ כזה: $[1,0,0,0,0,0,0,0,0,0,0,0]$ היא 0.079 לעומת זאת, ההסתברות עבור מערכ
כזה: $[10,10,10,10,10,10,10,10,10,10,10,10]$ שואפת לאפס.

יכרנו אם כן פונקציית התאמת לינארית בין ההסתברות המתkeletal לצבע הנבחר:

נרצה שעבור הסתברות העומדת על 0.08 להגיע לערך $H=84$ ולכן נחלק: $84 / 0.08 = 1050$. כלומר, עליינו
להכפיל את ההסתברות פי 1050 על מנת לקבל את הצבע המתאים. במקרה בו ההסתברות שואפת לאפס
הצבע, לאחר החישוב הניל יהיה קרוב מאוד גם הוא לאפס, ככלומר אדום. ערכנו מספר לא מבוטל של

ניסיונות על גבי מערכיים אפשריים של תאונות עד שהגענו להמרת הלינארית המפורטת לעיל אשר מתאפשרת בזכות העובדה שישנה סקללה וצופה של צבעים המקוריים באדום ומסתיימת בירוק.

2.7 אלגוריתם המרת الكرן

במהלך הפרויקט וכפי שנפרט בהמשך נדרשו לבעה בה עוסק אלגוריתם ההמרה של الكرן. אלגוריתם זה פותר בעיה/בעיה יסודית בתחום ה-geospatial (התחום הגיאו-מרחביבי) : בהינתן מצולע ונקודה יש למצוא אלגוריתם אשר ידע האם הנקודה שניתנה נמצאת בתוך המצולע או מחוץ לו. בהסתכלות ויזואלית, בעיה זו קלה מאוד אך כאשר מעבירים את הנתונים למחשב האלגוריתם הקЛОט בה ניתן היה לפטור את הבעיה נעלמת. ישנו עשרות אלגוריתמים אשר נותנים מענה לבעה זו אך כולם מכילים מカリ קיצון עבורם החישוב נכשל או שהם מכילים פונקציות טריגונומטריות מורכבות אשר מאריכות משמעותית את זמן החישוב.

אלגוריתם המרת الكرן הוא פתרון לא רע לבעה : אמונם נדרש במהלךו לעبور על כל קודקוד המצולע אך הוא פותר את הבעיה תמיד (מלבד במקרה בו הנקודה הנזונה הנה על גבי צלע המצולע). הרעיון העומד בסיסו של האלגוריתם הוא צייר קו ישר (קרן) מהנקודת שבסאלה ולבזר את צייר הקו כאשר הוא עבר את גבולות המלבן התוחם את המצולע. תוך כדי צייר הקו נערכז ספירה של מספר הפעמים בהם חצינו את תחומי המצולע. אם מספר הפעמים בהם תחומי המצולע נחתכו היאו אי זוגי הרי שהנקודת שבסאלה נמצאת בתוך המצולע ובאם המספר זוגי, הנקודה נמצאת מחוץ למצולע²³.

נכונות האלגוריתם מtabסת על כך שאם נעביר קו מהאינסוף לנקודה אשר לגבייה אנו תוהים וחוצה את גבולות המצולע, ייתכן אף מספר פעומים, הרי שהקו עבר לשירותין מתוך המצולע אל מחוץ למצולע ומוחוץ למצולע לתוכו וחזור חלילה. כתוצאה לכך, כל שני "מעברי גבול" שכallee הקו (או הנקודה שנעה על גביו) נמצאו מחוץ למצולע. כמובן, אם מספר החיציות הוא אי זוגי ישנה עוד "חציה" של גבול להשלים על מנת שהנקודת תהיה מחוץ למצולע ולכנן הנקודה נמצאת בתוך המצולע ולהפך. ניתן להוכיח את נכונות אלגוריתם זה באופן מתמטי בהתבסס על משפט העוקמה של Jordan. (משפט אינטואטיבי שאינו קל להוכיח לפיו כל "עיקומת גיordan" (לולאה רציפה במישור אשר אינה מצטלבת) מחלקת את השטח לחקל פנימי וחיצוני²⁴).

3. תכנון הפרויקט המעני

3.1 הגדרת דרישות

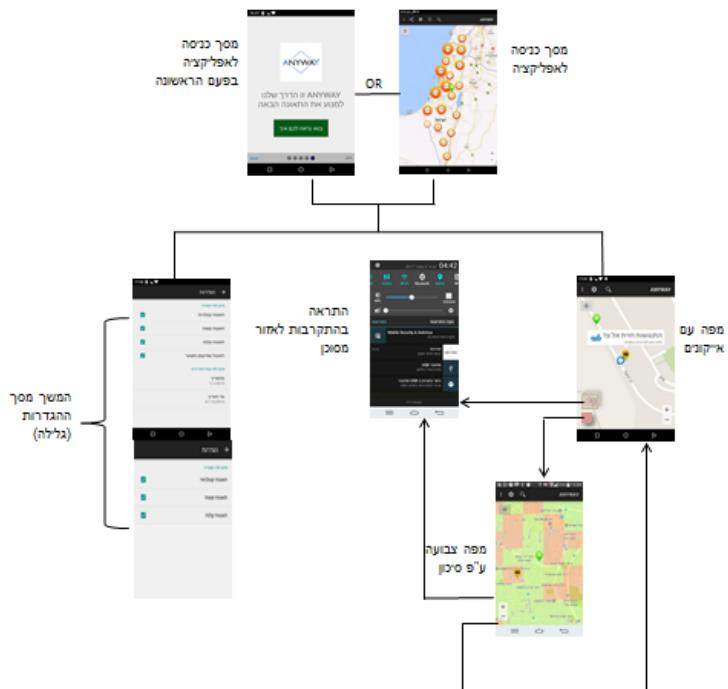
מטרת המשך פיתוח אפליקציית anyway היא צביעת מפת ישראל על פי אזורי סיכון ומטען התראתה בכניסה לאזור מסוכן.

3.1.1 ניתוח הדרישות

1. רמת הסיכון בכל אזור תחוشب ויינטן צבע הממחיש את רמת הסיכון.
2. השרטים והאפליקציה יתקשרו ביניהם.
3. המפה באפליקציה תהיה צבועה.
4. בכניסה לאזור מסוכן ישמע צליל התראאה ותשלח התראאה.
5. תצוגת האפליקציה תתעדכן בהתאם למיקום משתנה של המשתמש.
6. תיינטן אפשרות לבחור בין תצוגת מקבצים ל视uge צבועה.
7. האפליקציה תניב במהירות ככל הניתן לשינויים ותשמר על צריכת סוללה נמוכה.

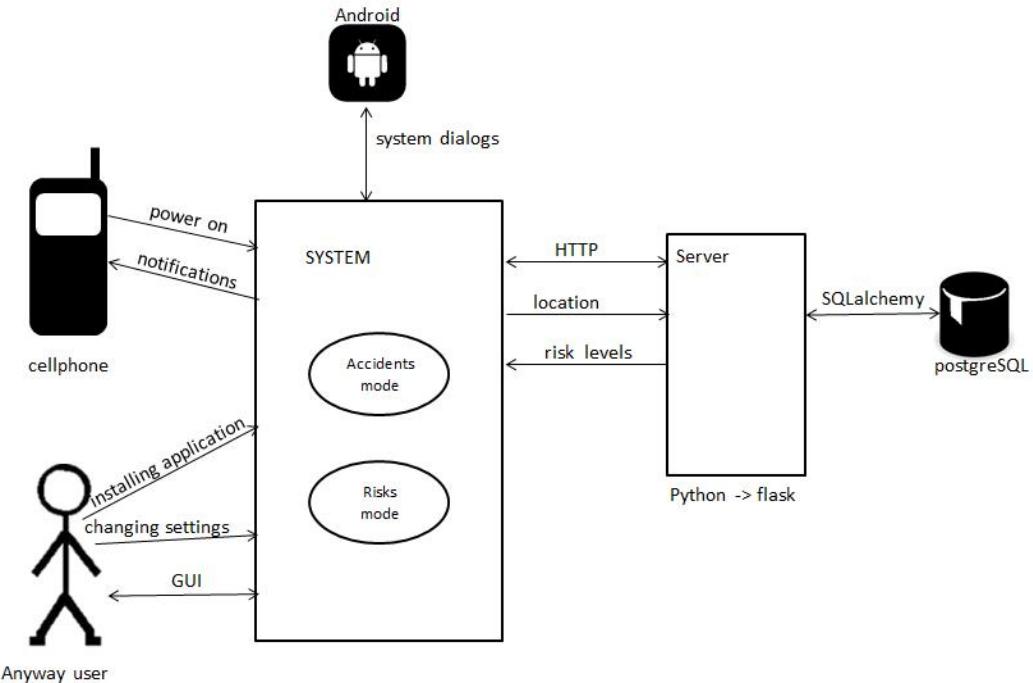
3.2 אפיון המערכת בהתאם לדרישות

על סמך רישימת הדרישות, יצרנו עץ מסכימים המתאר את המסכים אותם נרצה שהאפליקציה תכיל. עץ המסכים מכיל מסכים קיימים ומסכים שייווצרו במהלך פרויקט זה :



איור 3-1 המחשת עץ המסכים המתוכנן לאפליקציה

מתוך הנתונים שהובאו לעיל והדרישות אפיינו את המערכת בצורה ראשונית ובסיסית. למעשה, יצרנו מעין use case המתאר את הישיות והאינטראקציות ביןיהן ומהו אפין ראשון כבסיס להמשך ניתוח:



איור 2-3 תרשים מערכת לפיתוח ממבט כללי

הישיות המתווארת בתרשימים הם משתמש האפליקציה *anyway*, הטלפון החכם, מערכת ההפעלה אנדרואיד השרת ובסיס הנתונים. מנקודת מבטו כרגע, לפני שנכנסנו לעובי הקורה, מיקום המשתמש יישלח מהאפליקציה (המערכת) לשרת ורמות הסיכון יחורו כתשובה על גבי פרוטוקול התקשרות HTTP שאתו המערכת עובדת כרגע ויוצגו למשתמש על ידי המפה ובמידה והליך בחר יושם גם צליל התראה (המופיעים בתרשימים בתור *system dialogs*).

בתכנוןuproject על מספר שיקולים לגבי שימוש בשפות, בסיסי נתונים, פלטפורמות, פרוטוקולי תקשורת וסביבות עבודה שונות. להלן נביא את פירוט הלבטים השיקולים, החלטות ותיאור חלק של מצב המערכת הקיימת.

3.3 בסיס נתונים וקבצי נתונים

3.3.1 SQLite נתונים

בבחירה את בסיס הנתונים שייחסן את הריבועים בחרנו בסיס נתונים מסוג SQLite, על מנת שלא "להמציא את הגלגל מחדש" ולשכתב קוד שנכתבו כבר בשרת עם SQLite. בסיס נתונים זה הוא בסיס נתונים ייחסי משובץ המהווה למעשה ספרייה בשפת C המספקת בסיס נתונים קל. ספריה זו אינה

דורשת הממצאות של שרת נפרד, מමמשת את רוב הסטנדרט של SQL ומאפשרת גישה לבסיס הנתונים על ידי שימוש במשתנים לא סטנדרטיים של שפת השאילות של SQL. ניתן להשתמש גם בSQLite לצורך אחסון פנימי של נתונים תכנית. לבסיס הנתונים זה ישם עשרות ממשקים המאפשרים לעבוד עם בסיס הנתונים במנוגן רב של שפות²⁶. נוסף על כך, קובץ ההערכתה של בסיס נתונים זה קומפקטי במיוחד ותומך בנפח מידע עד למאה וארבעים טרה בית למד²⁷.

לבסיס נתונים זה ישם חמש סוגי טיפוסי נתונים : INTEGER - מספר חיובי או שלילי אשר ערכו יכול להגיע לעד שמונה בתים. REAL - ערך של נקודה צפה הנשמרת בשמונה בתים. TEXT - מחרוזת כלשהי אשר יכולה להיות מקודדת בצורה שונות. BLOB - טיפוס נתונים אשר שומר את הנתון בדיק כפי שהתקבל. NULL - טיפוס מסווג פלט. ניתן לשמור גם שדות מחלקה שלמה בתור עמודות הטבלה.²⁸

3.3.2 קבצי נתונים CSV

CSV היא צורה לשימור טבלאות בטקסט פשוט. בפורמט זה, כל שורה נשמרת באמצעות שבירת שורה וכל עמודה מוגדרת על ידי תו מוגדר, לרוב פסיק או tab. לצורת שמירה זו אין סטנדרט אחיד כלשהו : ייתכן ומספר השדות בכל רשומה יהיה שונה²⁹. ניתן לפתח קבצים אלו לקרוא וכתיבת כל שפה התומכת בקבצים.

3.4 מערכת הפעלה android

האפליקציה android היא אנו ממשיכות לפתח פותחה עבור מערכת הפעלה android. android היא מערכת הפעלה המיועדת לטלפונים חכמים, טאבלטים, טלוויזיות חכמת, שעונים חכמים ולמכוניות מערכת הפעלה android מבוססת על ליבת linux, המרובת פותחה על ידי חברת Android. חברת Andorid נוסדה כחברת סטארט-אף קטנה בשנת 2003 ונרכשה על ידי חברת Google בשנת 2005. המכשיר הראשון מבוסס android נמכר לראשונה בשנת 2008 ותווך כשתיים היפה android המציע הפעלהandroid למערכת הפעלה הנפוצה בעולם. נכון לשנת 2015 מערכת הפעלה android מחזיקה בנתוח שוק של מעל 80%. סمارטfonyים מפורסמים עם מערכת הפעלה android הם מכשירים בסדרת Galaxy של סמסונג ושל חברות LG, סוני אריקסון או של חברות סיניות וגם חברת מוטורולה ועוד³¹.

מערכת הפעלה זו מעוצבת במיוחד לשימוש בטלפונים חכמים וטאבלטים מבוססי מסך מגע. Google מפיתח את הקוד של אנדרואיד כקוד פתוח והדבר עוזר לרבות בפיתוח אפליקציות עבורה. לדוגמה, כאשר רוצים שרכיב מסויים במערכת יפעל בצורה שונה ניתן לראות את קוד המקור שלו ולשנות אותו, הקוד הפותח הוא תחת רישיון לתוכנה חופשית או קוד פתוח שחויב על ידי קרן התוכנה Apache. הפרויקטם של קהילת Apache מתאפיינים בתהליך פיתוח שיתופי המבוסס על הסכמה, על רישיון תוכנה פתוח, ועל רצון ליצור תוכנה באיכות גבוהה שתיהה המוביל בשוק בתחום. פרויקט הקוד הפתוח של אנדרואיד (AOSP, Project Source Open Android

ולפיתוח של מערכת הפעלה. בנוסף, לאנדרואיד יש קהילה גודלה של מפתחים אשר כותבים אפליקציות המגדילות את הfonקציונליות של המכשירים והן נכתבות בדרך כלל בשפת התכנות java את האפליקציות ניתן להוריד דרך GooglePlay או אתרי צד שלישי³². נכון לרביעון האחרון של שנת 2015 היו כ-1.4 ביליאון משתמשי android פעילים ברחבי העולם³³ וישם כ-24,000 מכשירים שונים אשר מבוססים על מערכת ההפעלה android בשנת³⁴ 2016. חנות האפליקציות של מערכת הפעלה android מכילה 2.4 מיליון אפליקציות נכון ל-2016 ועד לחודש Mai 2016 הורדו מהנوت GooglePlay למעלה מ-65 ביליאון אפליקציות³⁵

הפיתוח בandroid כולל מספר דגשים כגון: על אפליקציה של טלפון נייד להיות נטולת בעיות או קרייסות, בעלת ממשק משתמש שייהי מותאם עבור המכשירים השונים, ולהעניק חוזית משתמש חלקה. אי עמידה במשימות אלו תגרום להעדרה של המתחרות הרבות הנמצאות בשוק. יצירת ממשק למשתמש באנдрואיד זו משימה לא פשוטה כלל, שכן קיימים מכשירים שונים בעלי רזולוציות מס' שונות וגדלים שונים, על המפתח לדאוג שמשק המשתמש יוצג בצורה טובה בכל רזולוציה ובכל גודל של מסך. המכשיר הנידע עצמו הוא בעל ביצועי מעבד חלשים וכמוות זיכרונו מאד קטנה בהשוואה למחשבים המוצאים כיום, על המפתח לקחת שיקולים אלו בחשבו וכמוון גם את שיקול תצרוכת החישול במכשיר הנידע שהוא גם נושא רגש. בשל כך, האפליקציה צריכה להיות יעילה מספיק כדי שתורוץ בצורה חלקה ולא תגרום לצריכת סוללה מרובה. בעקבות העובדה הזיכרונו הקיטור במכשירים הנידעים, ההגבלה של מערכת האנдрואיד לכל שימוש היא בדרך כלל בין 16MB ל-48MB.

3.5 שפות הפיתוח

3.5.1 צד לקוח

צד לקוח אותו נפתח מיועד למכשירים אשר מותקנת עליהם מערכת הפעלה android. הפיתוח בandroid מתחלק לשתי שפות עיקריות: פיתוח native- קלומר C++ או פיתוח בשפת java. התכונות בשפה C++ לא ישרת את מטרותuproject שלנו בצורה הולמת מאוחר ולא יעשה ככל הנראה כל שימוש בdrivers או בחלקי חומרה של המערכת (שכן לשפת C++ ישנה גישה ישירה לחלק internal של linux) ולכן אין סיבה שנתכוות בשפה זו אשר גם פחות בטוחה ויציבה בהשוואה לjava. אי לכך ובהתאם לזאת בחרנו לפתח את האפליקציה בשפת java. נוסף על כך, אנו מושיכות פיתוח של אפליקציה קיימת אשר נכתבה כבר בשפת java ושינוי הקוד הקיים יתרוג בבחירה מגבלותuproject שלנו.

java הנה שפה תכנות מונחית עצמים (Oriented Object) אשר צוברת פופולריות בשנים האחרונות ומתיימרת להחליף את שפת C++ בשפה המעודפת על ידי המתכנתים. שפת java פותחה בחברת Microsystems (כיום חברתבת של Oracle) על ידי צוות בראשות James Gosling בשנת 1991 והוצאה לראשונה בשנת 1995. התחריר של השפה מבוסס במידה רבה על התחריר של C++, אך כולל הרחבות

רבות במטרה לאפשר תמייה מובנית בתהליכיונים, ביןלאומיות, אבטחה ועובדת בסביבת האינטרנט ותוכנות נוספות.

java היא שפה בעלת טיפוסיות סטטית חזקה, כלומר לכל ביתוי בשפה מתאים טיפוס ייחד. תקינות ביתויים נבדקת בזמן ההידור, במידת האפשר וכאשר אין זה אפשרי, הבדיקה תתבצע בזמן ריצה.

java תומכת בתכונות מונחה עצמים. עם זאת, java כולל מטעמי עילוות גם טיפוסים פרימיטיביים, שאינם אובייקטיבים. סיבה זו היא אחת מהסיבות שבשלה java אינה נחשבת שפה מונחת עצמים תהורה. עם זאת, לכל טיפוס פרימיטיבי בjava יש מחלוקת עוטפת, אליה וממנה מתבצעת המרה אוטומטית (Boxing). למשל, לטיפוס הפרימיטיבי int קיימת מחלוקת עוטפת Integer.

שפה java כוללת גם ניהול זיכרון אוטומטי. המתכנת פטור מן ההכרח לשחרר זיכרון המוקצה לאובייקט ברגע שאין עוד משתנים המצביעים עליו. במקרה זאת, סביבת זמן הריצה כוללת מנגנון "איסוף זבל", המבצע זאת אוטומטית.

Google play services

שירות זה הוא שירות רקע השיקיך ל-Google ומכיל גם חבילת ממשקים להתקני android. כיוון השירות מכסה מגוון רחב משירותי Google ומאפשר לאפליקציות לתקשר בקלות עם שירותי על ידי אמצעים מבוסתיים המתייחסים מאחוריה הקלעים ל-GMS: Google Mobile Services. השירותים הנכללים הם בין היתר חנות האפליקציות של android, ממשקי מיקום פלטפורמת הרשות החברתית Google+, שימוש בGoogle Maps או בGoogle Drive, שימוש בStreet View ו עוד .

שירות זה מתעדכן באופן אוטומטי דרך חנות האפליקציות Google Play בכל מכשיר אשר השירות מורכב עליו וגרסת מערכת ההפעלה שלו היא 2.3 ומעלה. כתוצאה לכך, Google יכולה לבצע עדכוני השקעות של השירות בצורה שקטה ומהירה מבלי שיצרני הטלפונים החכמים יאלצו לעדכן את הקושחה של android. ככלומר, עדכון אפליקציות המשמשות ב-Google Services אינו תלוי ביצרני הטלפונים³⁶.

לצורך פרויקט זה, המשתמש במנופת אנו מוכרכות להשתמש בשירות המפות של Google שכן מפתח האפליקציה לפניו השתמש במנופות אלו.

עם השימוש במנופת אלו מגיעות פונקציות מובנות אשר נקבעות עם התרחשות אירוע מסויים, כלומר פונקציות call back. האחריות על ה"הקשבה" לAIRONUIMS המגיעים וקריאה לפונקציה המתאימה שייכת ל- listner : עבר כל AIRONUIMS קורא לפונקציה ייחודית אותה יכול ממש המפתח כרצונו. פונקציות אלו מטפלות באירועים רבים ביניהם, למשל :

- - פונקציה זאת נקראת על ידי onLocationListner כאשר המשתמש משנה את מיקומו.

- **onCameraChange** - לפונקציה זו קורא `onCameraChangeListener` כאשר תצוגת המפה על גבי המסך השתנתה כתוצאה מהזזה של המשתמש, בעקבות פעולה אותה בחר המתכנת לבצע, בעקבות שינוי מיקום המשתמש ועוד.

TTS- Text to speech

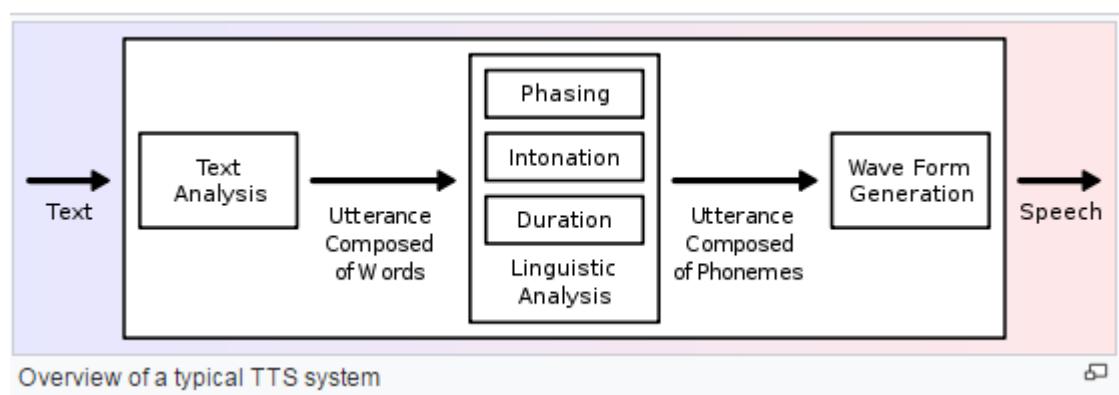
מנוע TTS ממיר טקסט משפה כלשחי הנთמכת על ידי המנווע לדיבור בשפה כלשחי הנתמכת על ידי המנווע, המנווע משתמש בסיס נתונים גדול על מנת לענה את הטקסט שהוכנס ולספק פלט קולי, ככל שמסדר הנתונים גדול יותר הפלט שיתקבל יהיה אICONטי וקרוב לטקסט המקורי.

מנוע הדיבור פותחן במקור עבור אנשים לקויי ראייה ומערכות הפעלה רבות כללו את מנגני הדיבור משנת 1990.

בפונקציית front-end ישנו שתי שימושים עיקריים, ראשית מנגני הדיבור ממיר את הטקסט שהתקבל ומכל סמלים לטקסט אלפבית ללא סמלים, תהליך זה נקרא נורמליזציה.

לאחר מכן מנווע מחלק את הטקסט שהוכנס לשפטים, פסקאות וסעיפים על פי סימני הפיסוק והסמלים שפורקו בשלב הקודם.

הbackend מקבל כקלט את ההמרה שבוצעה בfrontend וממיר את הייצוג הלשוני לציליל. מנגני הדיבור התווסף לapid android מגירסה 1.6 ומעלה.³⁷



איור 5-3 באירור ניתן לראות כיצד מנגע TTS פועל

התראות דחיפה - notification

סוג הודעה שאפשר לשולח למימושו מחוץ לתצוגת היישומון, ההודעה מופיעה תחילת כסמל בשורת התראות על מסך הטלפון וכאשר גוללים את שורת התראות נפתח אזור ההודעות ובו מופיעה הודעה שנשלחה בתוספת הסמל שהוצג בשורת התראות. הודעה נשארת על מסך ההודעות עד שהמשתמש מסיר אותה.

3.5.2 צד שרת

כאשר עסקנו בתכנון כתיבת צד השרת חכרנו את השפות המוביילות כיום בשוק לכתיבת צד שרת: השפות המוביילות והחזקות הן: `python`, `javaScript` ו- `PHP`.

השפה `PHP` היא שפת תסריט - שפת תכנות המיועדת לכתיבת תסריטים (סקרייפטים). מיועדת בעיקר לתכנות יישומי אינטרנט לצד השרת, אך יכולה לזרוץ על המחשב האישי באמצעות מהדר. השפה עצמה פותחה בשנת 1995 והתפתחה מאז לכדי השפה שאנו מכירים כיום. לשפה `PHP` ישנים מספר יתרונות ייחודיים: השפה מאוד קלה להבנה וללמידה. `PHP` היא שפה חזקה,אמינה ויציבה. נוסף על כך, שפת `PHP` מספקת אבטחה גבוהה מאוד, שמתעצמת אם היא בנויה על שרת Apache ומערכת הפעלה Linux. ניתן להריץ את מנוע `PHP` על מגוון רחב של מערכות הפעלה ושרותים, וכך ³⁸- `PHP` אינה מוגבלת לשירותים או למערכות הפעלה מסוימות. שפת `PHP` הפכה לשפת "קוד פתוח" (open source) כלומר היא חינמית ולכן בשנים האחרונות היא בטוחה הרבה פחות. `PHP` היא אחת השפות הראשונות שנבנו בייחוד לתכנות אתרים דינמיים באינטרנט וייתכן שאף הייתה הראשונה. בנוסף לעובדה ש-`PHP` היא שפת "קוד פתוח", היא מתפתחת מהר יותר ובצורה טובה יותר ממתחריה: לשפת `PHP` קיימים מגוון גדול של פונקציות ספרייה המכსות תחומים רבים ומגוונים על המפתחים. ביןיהם פונקציות להרצת פקודות shell ועוד.

JavaScript היא שפת תכנות דינמית מונחי-עצמיים המותאמת לשילוב באטרוי אינטרנט ורצה על ידי דפדפן האינטרנט לצד הלקוח. השפה מרוחיקה את יכולות שפת התגיות הבסיסית HTML ומאפשרת בכך ליצר יישומי אינטרנט מתחכמים יותר. למעשה, רוב האתרי האינטרנט המודרניים משלבים שפה זו. השפה ידועה בעיקר כשפה המוטבעת בדף HTML על מנת להציג דפים דינמיים, שמשולבת בהם תוכנה. קוד ה-`JavaScript` משולב בדף HTML מבוצע על ידי הדפדפן. מעבר לכך, לשפה יש גם סביבת ריצה עצמאית שمبוססת על השפה שמשמשת להרצת שרטוי אינטרנט (Node.js) וכן יישומיים שלוחניים.

השפה תומכת בתחביר הסטנדרטי של שפת C, מבחינת אופרטורים, מבני בקרה ופוקודות, וմבדילה בין אותיות גזלות לקטנות (case-sensitive). כמו ברוב שפות התסריט, טיפוסי השפה מקושרים לערכים ולא למשתנים, משתנה X יכול להפוך במהלך הריצה ממספר למחרוזת לפי השימוש.

השפה מבוססת "תכנות מונחה דגמי אב" (Prototypes), שהוא סוג (לא נפוץ) של תכנות מונחה-עצמיים. השפה בעירה היא שפה מבוססת אובייקטים (אפילו טיפוס בוליאני [Boolean] הוא למעשה אובייקט). האובייקטים בשפה הם מערכיים אסוציאטיביים. תכנות מונחה עצמים אפשרי, אבל לא בדרך המקובלת של הגדרת מחלקה שאינה קיימת בשפה, אלא באמצעות יצירת אובייקט והצמדת מאפיינים אליו. כל מאפיין באובייקט יכול לתפקיד כמשתנה או פונקציה. וישנם שלשה סוגי אובייקטים. מעבר לכך, השפה תומכת בהרחבת מתוודות של אובייקטים טבעיות בשפה כמו למשל מחרוזות או Date באמצעות Prototyping.

השפה היא גם שפה מונחת אירועים (Events), כך שמהלך ריצת התכנית מבוסס על אירועים שמתרחשים בזמן ביצועה. השפה תומכת גם בתכונות פונקציונלי. למעשה כל פונקציה היא אובייקט עצמה. השפה מאפשרת השמה של פונקציות למשתנים, העברת של פונקציה כארגוומנט, והגדרת פונקציות אנונימיות. השפה מתממשקת באופן טבעי עם ממשק DOM, המציג מסמכי HTML ו-XML תוך שימוש במתודולוגיה של תכנות מונחה-עצמים. לשפה יש תמייה מבנית בביטוי רגולרי.³⁹.

python היא שפת תכנות חדשנית, מכוונת עצמים, משוערכת (interpreted), דינמית ואיןטר-אקטיבית. השפה פותחה בתחילת התשעים על ידי Guido van Rossum. השפה python נועדה להיות שדרוג של שפת התכנות ABC, כך שתוכל להתאים למערכת הפעלה Amoeba ומماז השטנה השפה בצורה משמעותית⁴⁰. python תוכנה תוך שימת דגש על קריאות הקוד, וכוללת מבנים המיועדים לאפשר ביטוי של תכניות מורכבות בדרך קצרה וברורה. אחד המאפיינים הבולטים בתחום השפה הוא השימוש בהזחה להגדרת בלוקים של קוד ללא שימוש בסוגרים או מיילים שמורות.

ישנם שני כללים עיקריים אשר עמדו נגד עיניהם של מפתחי השפה: הראשון הוא "צריכה להיות דרך אחת, ועדיף אחת בלבד, לבצע זאת" והשני הוא נתינת אמון בתוכנת המקבל קוד קיים שלא יבצע בו דברים שברור שלא התכוונו אליום כתבי הקוד מלכתחילה.⁴¹

להלן התכונות העיקריות של python:⁴²

- **מחרות**- כמו ב-java, תכניות ב-python מתמקבלות לקוד-ביתיים (bytecode) שהמשערך יודע

לקראן. לאחר פעולה הקימפול לקוד-ביתיים נשמר הקובץ בשם המקורי עם הסיומת .pyc.⁴³.

פעולה הקימפול לקוד ביתיים זו מבוצעת את התכניות הכתובות ב-python ממשובותיהם של המחשב ושל מערכת הפעלה עליהם הן רצויות, ומעניק לתכניות ניידות (portability) בין פלטפורמות. בינווד ל-java, השפה python שומרת על איןטר-אקטיביות, תוכנה אופיינית של שפות משוערכות. python אינה זקופה לשלביו קומpileציה או קישור (linking) מכיוון שלא מתבצעים מידית עם הרצה. עם זאת, הקוד המקומפל למחצה רץ מהר בהרבה מזו של שפות משוערכות אחרות או שפות shell לוטגיהן.

- **תכנות מוכoon עצמאי**- python היא שפה מוכוונת עצמאים לחלוטין. יש לה את כל המאפיינים

והתכונות הדורשים לשם כך: אובייקטים, מחלקות, ירושה (גם ירושה מרובה), מודולים, זריות, שגיאה ויזיהו טיפוסים דינמי. נוסף על כך, ב-python כל דבר הוא אובייקט. עם זאת, ניתן לתוכנת בpython במספר פרדיגומות נוספות: תכנות גנרי, תכנות פרוצדורלי ובמידה מסוימת אף תכנות פונקציונלי. ישנה תמייה של השפה בתכנות מקבילי ותכנות מוכוון אירועים. בכתיבת מימוש מחלוקת כלשהי ישנו שימוש במילה השמורה self. מילה זו מתייחסת למופיע הספציפי של המחלוקת בה אנו נמצאים בעת, מעין מקבילה למילה this בשפות רבות אחרות. מעבר לכך, מפתחי השפה קבעו כי בכל מתודה השיכת מחלוקת כלשהי הfrmater הראשון הנשלח הוא self כלומר

המופיע של האובייקט שמננו המתודת נקראת. עם זאת, אין זו חובה לכתוב בצורה מפורשת בהצחרת המתודת על קבלת מופע זה כחלק מרשימה הפרמטרים וגם בשילוחה למетодה מסווג זה אין זו חובה לשולח את הפרמטר `self`. השפה `python` עשו זאת עבורנו אוטומטית⁴⁴.

- **ניהול זיכרון אוטומטי**- `python` מציעה ניהול זיכרון אוטומטי בזמן ריצה, על ידי ספירת references וaisוף זבל (garbage collection) אוטומטי של נתונים שאינם מנוצלים עוד. ישנה ערימה של גביה מוקצים האובייקטים כאשר הם נוצרו (האובייקטים במשיק `python` יכולים להיות מיוצרים על ידי קריאה למחילה, הצבה של ליטרלים בשפה, קריאה לפונקציה היוצרת אותם ועוד). המשתנים עצם, "ההפניות" (references), הם מחרוזות פשוטות המשמשות כמפתחות במילון שמנדריך את תחום ההכרה (scope) הנוכחי.
- **אסטטיות**- קוד `python` הרבה פচות "מלוכך" מסימני תחביר כגון הצורות משתנים. טוווח משתנים ("בלוק") ב-`Python` מוגדר על ידי הזזה (indentation) הזוז הקוד מהשוליים על ידי תווים או רווחים. תכונה זו מיועדת לשפר את קריאות התכנית, בכך שהיא כופה על המתכנת כתיבה בסטנדרט אחיד (פחות או יותר), וחוסכת בקוד: קוד ב-`Python` לרוב קצר פי-5-3 מקוד מקביל ב-`java`, ופי-10-5 מאשר קוד מקביל ב-`C++`.
- **התממשקות נוחה ומגוונת**- `python` ניתנת לשיבוץ בתכניות אחרות, בלי מגבלת זכויות יוצרים. ניתן לכתוב מודולים חדשים ב-`Python` בקלות וכן להשתמש במאגר הגדל של ספריות ושירותים נלוים כגון Tk, שירותי רשות, ממשקי CGI, כתיבת HTML ועוד.
- **מערכת הטיפוסים**- מערכת הטיפוסים בשפת `python` מאופיינת כחזקת לא ניתנת להפעיל על משתנה פוליה שלא מוגדרת על הטיפוס אליו הוא שייך. השפה גם динמיית התוכונה זו מושגת בכך שהאובייקטים ולא המשתנים בשפה שייכים לטיפוס ולבן כל משתנה בשפה יכול להציגו על אובייקט מכל טיפוס. שימוש שכיח בדינמיות של השפה נעשה בהגדרת פונקציה: אין הגבלה על סוג הטיפוס שיועבר בכל קריאה אליה. (חשיבות לציין שהשפה מובלעת לומר אין אכן המשתנה שהגיע לפונקציה הוא מהטיפוס לו או שהוא מציין) מושך הנה מובלעת בכלומר אין הכרזה/ הצהרה על קיום משתנים. למעשה זה יشنם יתרונות וחסרונות: מצד אחד מותאפשרת גמישות רבה בכתיבה הקוד וקייםו באופן משמעותי אך מהצד השני קשה יותר לאתר שגיאות בזמן ההידור וכן קשה יותר לזהות טעויות הקלדה.
- **טוח ההכרה הוא סטטי-קסיקלי**: שם של משתנה מוגדר רק בקטיעי קוד המוכלים בקטיע שבו בוצעה אליו פעולה כריכה. המשתנים ב-`Python` הם למעשה שמות של מצבים, references, לאובייקטים ולא שמות האובייקטים עצמם. לדוגמה, כאשר מתבצעת פעולה בשם בין משתנים בצהורה `y=x`, לא מועתק תוכן האובייקט אליו מצביע `u`. במקום זה, המשתנה `x` מוגדר כך שיצביע אל אותו אובייקט אליו מצביע `u`.

• **מפתח python מפותחת** - python היא שפה חיה וдинמית. גרסאות חדשות ותיקונים מופצים כל העת, וספריות חדשות נכתבות. בראש יש מאגר גדול של חומר על python . נוסף על כך הספרייה הסטנדרטית של python כוללת עשרות מודולים, אשר מקלים מאוד על כתיבת קוד והם מפתחים ומשתכללים.

לשפה python ישנו عشرות ממשקים וAPI להתחברות עם בסיסי נתונים מסווגים רבים : החל מממשקים גנריים, ממשקים לבסיסי נתונים יחסיים (relational data bases) התומכים גם ב-SQL, ממשקים לבסיסי נתונים לא יחסיים (כמו XML) ועד למשקים לבסיסי נתונים מסווג⁴⁵ native .
בחרנו בשפה python לפיתוח השרת ולא בשפות אחרות לאחר והשתתת הקיימים בעwayany כתוב בשפה זו.
אנו השרת אותו פיתחנו הוא לא השרת של anyway אך לאחר ובעתיד יוכל בהחלט שהוא ישתלב עם השרת הראשי וכן מאחר וישנו חלק קוד אשר נדרש לככיתה גם בשרת אותו י_crano כמו מבני נתונים לדוגמה) בחרנו לכתוב בשפה python בגרסה 2.7 לאחר ושותי החבורה כתובים גם הם בגרסה זו של שפת python .

את בסיס הנתונים בפרויקט בחרנו לעטוף במודול בשם sqlite3 אף על פי שבשרת המקורי של הפרויקט anyway נטען בסיס הנתונים במשק sqlalchemy . בחרנו כך מאחר ומודול sqlite3 מהיר יותר מהמשק sqlalchemy עד פי עשרים וחמש.

מודול זה מגיע כחלק אינטגרלי מספריית python הסטנדרטית ומהווה משק מלא לשפת השאילות SQL . המודול אינו דורש המצוות של שרת או יצרת קונפיגורציה כלשהי, הוא מהיר ושומר את כל בסיס הנתונים על קובץ אחד בלבד⁴⁶ .

על מנת לקבל עדכונים מהשרת לגבי תואנות דרכיים קיימות ספריות וממשקים רבים בpython לתזמון משימות. בחרנו במשק APScheduler לצורך תזמון משימות עדכנים בסיס הנתונים. הסיבות ששל בחרנו להשתמש דווקא במשק זה הן מאחר והמשק מאפשר שלוש סוגים תזמון למשימה : משימות עם זמן התחלה וסיום, משימות חד פעמיות ומשימות המתבצעות כל מרוחק זמן מסויים. המשק תומך בהתחברות למשקי בסיסי נתונים רבים ומשתלב עם מספר לא קטן של מערכות python שונות⁴⁷ .
ספרייה נוספת בה השתמשנו לצורך קבלת עדכונים מהשרת היא urllib2 ספריה זו מספקת לתוכנה ממשק משתמש מותאם על מנת שיוכל להשתמש במקרים האינטרנט המזוהים על ידי כתובת url . ניתן באמצעות ממשק זה ליצור בקשות HTTP מסווגים שונים כמו : GET, POST ו עוד ואך מכיל תמייה בפרוטוקולים אחרים כמו HTTPS ועוד⁴⁸ .

3.6 סביבות הפיתוח

לצורך פיתוח הפרויקט השתמשנו בשתי סביבות פיתוח עיקריות : לפיתוח השרת נעזרנו בcloud9 , ולפיתוח האפליקציה השתמשנו בסביבה העבודה android studio להלן נפרט על סביבות העבודה והסיבות בשלחן בחרנו דווקא בסביבות אלו :

3.6.1 סביבת cloud 9

סביבה זו היא סביבת פיתוח מושלבת אינטרנטית אשר פורסמה כקוד פתוח החל מגרסת 3.3 והלאה. סביבת הפיתוח תומכת במלמעלה מארבעים שפות תכנות ביניהן JavaScript, Python, Ruby, Java, C, C++, C++ ועוד. סביבת הפיתוח מאפשרת למפתחים לגשት ישר לכתיבת הקוד מאחר וסביבת העבודה עברה קונפיגורציה קודם לכן על ידי השירות האינטרנט. בנוסף, ניתן לשתף את הקוד עם עמיתים ולאפשר להם גישה לקובץ או לעריכה של הקוד. סביבת העבודה עצמה נכתבת כמעט במלואה ב-`JavaScript` ו-`Java`script (back end) מבוססת `node.js`.

מאפייני הסביבה רבים וביניהם מעין "מסוף" של סביבת הפיתוח (terminal) מובנה אשר מאפשר ביצוע פקודות `ssh` בסיסיות, סמנים מורובים לפיתוח ועריכה בו זמני, `debugger`, תמייה במאגרי קוד כמו `github`, אפשרות לפיתוח פרויקטים ציבוריים ופרטיים, הדגשה תחבירית בחלק גדול מהഫוטוניות, הצעות חכמות להשלמת הקוד⁴⁹ ועוד.

חברות רבות עוסקות עם `cloud 9` ביןיהן `heroku`, `mozilla`, `soundcloud`.⁵⁰ סביבה זו יישם כחצי מיליון משתמשים רשומים⁵¹.

סביבת `9 cloud` משתמשת ב-`Docker` כאמצעי להכנת מערכת הפעלה של סביבת העבודה האינטרנטית. Docker הוא פרויקט קוד פתוח שפורסם לראשונה בשנת 2013 אשר מכיל וווטף חלקים מתוכנה במערכת קבצים אשר כוללת את כל מה שהתקינה צריכה על מנת לרוץ: קוד, זמן ריצה, כלי מערכת, ספריות מערכת ולמעשה כל מה שנitin להתקין על שרת. עובדה זו מבטיחה שהקוד ירוץ תמיד באותו האופן ללא קשר לשרת בה הוא רץ. נוסף על כך מספקת Docker שכבת אבטרנטציה ואוטומציה ברמת מערכת הפעלה וירטואלית מבוססת `linux`.⁵² סביבת `9 cloud` משתמשת בגרסתה הנוכחית (3.0) במכונה וירטואלי עם מערכת הפעלה `ubuntu` (ויריאציה של `linux`) אותה Docker מספקת.⁵³

אחדונה של סביבת העבודה `9 cloud` מתבצע בפלטפורמת הענן של חברת Google בתוך שירות שנקרא Google Compute Engine שירות זה אשר הוקם לראשונה בשנת 2012 מאפשר למשתמשים להריץ מכונות וירטואליות על פי צרכי המשתמש ולאחר שעבר אימוט*. [מנועי המחשב של Google נגישים באמצעות קונסולת מפתח, משק תכנות יישומים התומך באופן מלא REST או משק שורת פקודה והנו בתשלום].⁵⁵

בחרנו להשתמש בסביבת עבודה זו בשל כמה מיתרונותיה העיקריים: מעבר לכך כי זו هي סביבה יציבה, ישנה אפשרות לכתוב קוד שתינו בו زمنית, לראות עדכונים בזמן אמת וכן פיתוח בסביבת העבודה הניל הנו תואם `linux`. כמו כן טמון בסביבה זו יתרון עצום: סביבת IDE מהוות למעשה service הנותן שירותים לשרת אותו נפתח כך שסביבת העבודה היא גם השרת.

*האימוט נעשה באמצעות פרוטוקול OAuth 2.0 אשר מאפשר אימוט ללא סיסמה

3.6.2 סביבת Android Studio

ישנן למעלה מעשדים סביבות עבודה המאפשרות פיתוח אפליקציית android בשפות שונות⁵⁵, לאחר ואנו ממשיכות בכתיבתה של אפליקציה קיימת אשר נכתבת בשפה java עמדו בפנינו שתי אופציות: סביבת העבודה Eclipse עם תוסף (Android Development Tool) ADT (Android Devlopment Tool)⁵⁷.Android Studio בתחילת רצינו להשתמש בסביבת העבודה Eclipse לאחר ופיתחנו בה אפליקציה מבוססת android במסגרת קורס בלימודים והסביבה מוכרת לנו. אך מאחר והמפתח הקודם של האפליקציה השתמש בערכות נושא ייחודיות לשא Android Studio אשר לא הופיעו באclipse ושיבו את התצוגה נאלצנו להשתמש בסביבת העבודה Android Studio.

סביבת העבודה Android Studio הנה סביבת הפיתוח המשולבת הרשמית לפיתוח בפלטפורמת android. הסביבה זמינה החל מנת 2013 אך רק בסוף שנת 2014 יצא הגרסה הראשונית של הסביבה.

סביבת העבודה Android Studio חינמית תחת תנאי apache license ומבוססת על גבי תוכנת 'IntelliJ IDEA' דבר שהייתה יתרון עבורנו מאחר ואנו מכירות סביבות עבודה אחרות מבוססות על אותה תוכנה. Android Studio עוצב במיוחד עבור מפתחי android והחליף את סביבת העבודה Eclipse עם התוסף ADT שהייתה עד כה סביבת הפיתוח העיקרי לפיתוח android בשפה .java⁵⁸. מאפייניה של סביבת העבודה Android Studio רבים ורובם מהווים יתרונות לא מבולטים כפי שנפרט:

- **לסביבת Android Studio ישנו Gradle מבנה.** למעשה הוא קוד פתוח

אשר מכין באופן אוטומטי גרסה מודולרית לקוד בסביבת Android Studio המבוססת על JVM. כתוצאה לכך הרבה מקונפיגורציות ההידור קורות באופן אוטומטי. Gradle תומך

בהידור מצטבר של הקוד ע"י זיהוי חכם של החלק בעץ ההידור שצריך עדכון ועדכונו בלבד, ללא החלקים האחרים, גם אלו התלויים בחלק שעבר עדכון. זיהוי חכם זה מתאפשר בין השאר בשל העבודה כי תוצר ההידור אינו קובץ XML אלא קובץ פשוט יותר שאינו במבנה של עצ. כלומר, ההידור נעשה בגישה לפיה ישנה שפת מחשב מיוחדת לכל תחום מסוים באפליקציה, בניגוד לגישה על פייה שונה שפה כללית לכל מטרה ותחום באפליקציה⁵⁹. פרויקט Gradle תומך בעיקר ב 3 שפות, ביניהם java⁶⁰. למעשה_gradle משמש כתוסף לסביבת העבודה, כך שאם ברצונו של המפתח להציג גרסת הידור שונה או להדר את הקוד שכתב בשפה שאינה java עליו בסך הכל להחליף את התוסף⁶¹.

- **תיקון שגיאות, ProGuard refactoring.** ל紧紧围绕 סביבת העבודה Android Studio ישנו מנגן

refactoring ייעודי למפתחי android. מנגן זה מפחית את סיבוכיות הקוד, משפר את קרייאותו, מקל על תחזוקתו וגורם לו להיות גמיש יותר לשינויים ורחבות⁶². כמו כן קיים מנגן תיקון שגיאות מהיר, מנגן הצעת השלמה חכם, מנגן ניתוח קוד ומנגן לזיהוי שגיאות כתיב⁶³.

- נוסף על כך, מוטמעים בסביבת העבודה Android Studio כלי ProGuard : כלי שורט פקודת המכווץ קוד בשפת `java` ו מבצע אופטימיזציה על שפת `bytecode` שלו. הכללי מזהה ומוחק פקודות אשר אין בהן שימוש⁶⁴.
- **כלי lint**- כלים אלו הנם שרות של `Android` אשר מסמן קוד כלשהו או מבנה כלשהו כ całego העולמים להתפתח לכדי באג⁶⁵. בסביבת Android Studio שרות `lint` מרייצ' למעלה מ-280 בדיקות⁶⁶ אשר מזהות בעיות ומסוגות אחרות בתחום הביצועים, שימוש, בטיחות, תאימות גרסאות ועוד⁶⁷.
 - **עורך layout**- הנו כלי ויזואלי עשיר המאפשר למפתח לగורר ולהסיר רכיבי UI וליצור תצוגה מקדימה של `layout` במגוון קונפיגורציות של מסכים⁶⁸. עורך `layout` נבנה בהתאם מלא עם `ConstraintLayout API` (ממשק המכיל אילוצים המוכלים על תצוגה במסכי אפליקציות `android`) כך שניתן בקלות להתאים את חלקי התצוגה לגודלי מסכים שונים⁶⁹.

3.7 סביבות הרצה

על מנת להריץ את האפליקציה במהלך פיתוחה נזורנו באמולטור ובמכשורי הסלולר שלנו לאחר שהועברו למכשיר "מפתח".

לעתים לצורך הצגת האפליקציה במהלך הפיתוח השתמשנו באמולטור אותו מספקת חברת `Genymobile` ונקרא `Genymotion`. אמנם סביבת העבודה `Android Studio` מגיעה עם אמולטור מובנה, אך הוא אטי באופן מחריד גם כאשר הוא רץ על גבי מחשבים בעלי מעבד חזק. האמולטור `Genymotion` לעומתו, מגיע עם וירטואлизציה של ארQUITקטורת `x86` המאפשרת למערכות הפעלה להריץ בו זמינות עד 86 " מעבדים" בדומה בטוחה ויעילה. נוסף על כך לאמולטור זה ישנה מערכת המאיצה את פעולת החומרה בשם `OpenGL` כך שהרכבת האפליקציה בפיתוח על גבי אמולטור זה הופכת ליעילה יותר. בפועל, האמולטור מורץ על גבי מכונה וירטואלית הנוצרת דרך `VirtualBox` כך שהוא תומך גם בחישנים חומרתיים כמו `GPS`, בטריה, מצלמה, רשת ועוד. אמולטור זה ניתן להרצה גם מסביבת הפיתוח `Eclipse` וגם מסביבת הפיתוח `Android studio` בעצמה⁷⁰.

כיום, המכשורים אותם ניתן ליצור וירטואלית על ידי `Genymotion` מגיעים עם מערכת הפעלה של `android` כמו כל מכשיר אחר מלבד העובדה שהם אינם מכילים את חנות האפליקציות ואת `play` `google` `services`. חנות האפליקציות ובעיקר `google play services` נדרשם לפיתוח האפליקציה הנוכחית מאחר והם מאפשרים את השימוש במפה על גבי האפליקציה. הפתרון שמצאנו לכך הוא הורדת קובץ המכיל את המאפיינים החסרים והתקנתם באופן ידני על גבי המכשיר הוירטואלי.

השימוש בסביבות ההרצה לסיורGIN נבע מה הצורך לדמות מקום משתמש, מסלול משתמש ועוד. נתקלנו בבעיות רבים בניסיונו לעורך הרצות סימולציה של האפליקציה: במרכז האקדמי לב, אשר בו אנו סטודנטיות ובמסגרתו פיתחנו פרויקט זה יישנו מסנן אינטרנט אשר מנע מהתקשרות לעבוד רפואי בזמן ההרצה. נוסף על כך, האמולטור אינו קולט מיקום מדויק של המשתמש ועל כן נאלצנו להריץ את האפליקציה במהלך פיתוחה הנו על מכשירי הסלולר האישיים שלנו לאחר שעברו התאמות וערכו גרסאות על מנת לאפשר הרצה על גיביהם והן על גבי האמולטור.

Drive Tests 3.7.1

על מנת לעורך מבחנים לעמידות האפליקציה, נכונתה, עמידתה ויציבותה גם במקרים בהם רשת הנתונים הייתה חלה ועוד ערכנו מספר נסיעות מבחן לאפליקציה על מנת לוודא שהיא אכן רצה רפואי ומגיבה רפואי למצבים קיצוניים. הרטנו את מסך המכשיר הסלולי באתן מנסיעות המבחן והרטנו יהיה זמין לצפייה לאחר שהאפליקציה תשובת תשוק באופן رسمي.

3.8 תקשורת

3.8.1 תקשורת קיימת

באפליקציה ישנה תקשורת אותה תכנן וכותב המתכנת הקודם של האפליקציה. התקשרות באפליקציה מבוססת על תהליכי נפרדים על מנת שתהליך משק המשמש לא ייעזר בהמתנה לתגובה. שליחת הבקשות לשרת מהאפליקציה בתהליך נפרד מתאפשרת באמצעות שימוש בספרייה Volley ובמחלקה AsyncTaskJava לסיורGIN (פירוט והרחבה ניתן בהמשך). בשורה התחthona, הבקשות מהאפליקציה לשרת נשלחות בפורמט JSON והן מנוהלות על ידי תור עדיפות אשר ממומש באפליקציה במחלקה בשם AnywayRequestQueue תוך שימוש העיצוב singleton. בנוסף, מוגדרת מחלקה בשם Utility עבור פונקציות העוזר השונות ובها מוגדרים דיאלוגים וMSCים שונים.

נוסף על כך, על מנת להציג מקבץ של תואנות דרכים (לDIGI כאשר המשתמש נמצא בזום) ישנו צורך לעשות את החישוב על גבי השירות. מאחר ובזמן פיתוח האפליקציה השירות של anyway לא תמך בחישוב זה נבנה שירות אחר לצורך כך ואליו הועתק מסד הנתונים רק עם השדות הרלוונטיים לצורך יצירת ושליחת מקבץ. יותר מאוחר פותחה פונקציה זו גם בשרת anyway אך מאחר ותצוגת הפונקציה מבוססת של SMARTPHONE הייתה בעייתית ואטית יותר בחר המפתח להישאר עם החישוב אותו כתוב בשרת אחר).

ספרייה AsyncTaskJava השיכת לאב Java מאפשרת לתוכנה להשתמש בשירות באופן תקין וקל בתהליכי UI בכך שהוא מאפשר לבצע חישובים ברקע ולהציג את תוצאותיהם בתהליכי UI ללא צורך במניפולציה על תהליכי או handlers. השימוש במחלקה זו לצורך חישובים ברקע אידיאלי רק עבור חישובים ופעולות קצרות שאורכם מספר שניות לכל היותר.⁷¹

ספריית Volley הנה ספרייה HTTP אשר מעבירה מידע מAPPLICATION אנדרואיד על גבי הרשת. הספרייה מתזמנת את הבקשות בצורה אוטומטית, מאפשרת להשתמש בכמה התקשרויות לרשת בו זמנית, תומכת בפורמט JSON ועוד. היתרונם בשימוש בספרייה זו היא שהיא תומכת בכך ככלומר בהפעלת פונקציות הנמצאות על גבי מחשב אחר בראש מוביל שני המחשבים יסכו ביניהם על פרטי ההתקשרות. הספרייה מכילה Pool Thread עבור תקשורת לאינטרנט ומנהל תור בקשות לאינטרנט לפי עדיפות המסודרת בתור. הספרייה אינה מתאימה להורדת קבצים גדולים או לפעולות streaming שכן היא שומרת את כל התגבות שהתקבלו במהלך ניתוח הנתונים שהתקבלו. יתרונה העיקרי של הספרייה הוא שהוא משחררת את המתכנת מכתיבת קוד התקשרות ומאפשרת לו להתרכז בלוגיקה המאפיינת את האפליקציה אותה הוא כותב. הספרייה מאפשרת מעבר לשילוח רגילה של בקשה לשירות (פנימית לשירות) למשגש של בקשה מותאמת אישית.⁷²

אחד מהבעיות בשימוש בספריית Volley היא שלספרייה אין package רשמי כלשהו. המלצת מתוחזקי הספרייה היא להוריד עותק של הספרייה ולצרכו לפרויקט, לעיתים גם נדרש מחלוקת cache אשר חסורה בספרייה.⁷³ בנוסף, הוספה של הספרייה לפרויקט משפיעה אוטומטית על גודל האפליקציה.⁷⁴ בהתייעצות עם מנהה הפרויקט ולאחר ניסיונות הרצה ובדיקות החלתו שלא להמשיך בתקשרות הקיימת עבור הפניות מהאפליקציה לשירות אותו נקים מאחר והתקשרות הקיימת אטיט ולא יעילה.

3.8.2 פרוטוקול HTTP

לצורך התקשרות בין השירות אותו נקים שיחשב את רמות הסיכון בכביש לבין השירות הקיים של anyway הצורך להשתמש בפרוטוקול HTTP. פרוטוקול HTTP הוא אחד מאבני היסוד של האינטרנט ואחד מפרוטוקולי התקשרות המוכרים והנפוצים בעולם, ראשית התיבות שלו הם HyperText Transfer Protocol והוא מהויה פרוטוקול שרת-לקוח מבוסס טקסט. אמנים הלקוח הנפוץ ביותר המשמש בפרוטוקול זה הוא דפדפן האינטרנט אך יש לפרטוקול שימושים נוספים רבים. פרוטוקול HTTP מבוסס טקסט כלומר, אין אפשרות לשלוח בו מידע ביןארי, גישה זו אמנים מקופה מאוד על תהליכי הפיתוח, הניתוח ואייתור השגיאות, ועם זאת מוסיפה נפח למיידע הנשלח. פרוטוקול HTTP "רווכב" על גבי חיבור תקשורת קיים,⁷⁵

התקשרות ב-HTTP מתחילה ביצירת שיחה בין השירות ללקוח באמצעות פרוטוקול TCP בשכבה התüberה של פרוטוקול TCP/IP, ונמשכת בסדרה של בקשות (requests) ותשובות (responses) שנשלחות על ידי הלקוח והשירות, בהתאם. ראשית, הלקוח יוצר חיבור לכתובת ה-IP ולפורט שבו השירות נמצא, בדרך כלל פורט 80. לאחר מכן נשלחת הבקשה, הכוללת את הכתובת של האובייקט המבוקש (למשל, דף HTML) ופרטים נוספים על הבקשה ועל הלקוח. השירות קורא את הבקשה, מפענח אותה, שולח ללקוח תשובה בהתאם ולרוב מנתק את החיבור ללקוח כשהשליחה הסתיימה.

מעצם הגדרתו, פרוטוקול HTTP הוא protocol - stateless חסר מצבים : אין בפרוטוקול עצמו שום אמצעי או שיטה לאחסן מידע מהמשתמש בין בקשות. הפרוטוקול מטפל בכל בקשה כתרנואקטיה נפרדת, שאינה קשורה לשום בקשה קודמת, כך שהתקשרות מרכיבת מזוגות נפרדים של בקשות ותשובות⁶. על מנת ליצור תקשורת בין הלקוח לשרת שבסיסת על ההיסטוריה הבקשות-תשובות בין השרת ללקוח נעשה שימוש בעוגיות (cookies).

בקשת HTTP מורכבת מהשדות הבאים : שיטת הבקשה- חלק מהבקשות הן בטוחות וחלק לא, דוגמאות לבקשה הן : GET-GET, POST-POST, CONNECT-CONNECT, DELETE-DELETE, מחק-DELETE. כתובות האובייקט המבוקש (כתובות מלאה או יחסית לשרת), גרסת הפרוטוקול עליו מבוססת הבקשה, שדות כוורתה המתיחשים לבקשת, ללקוח, או לתוך הנמצא בגוף הבקשה וגוף הבקשה אשר הם אופציונלי. התשובה אותה שולח השרת מכילה נתונים כמו גרסת הפרוטוקול על פייה תשובה השרת נבנתה, קוד וטקסט קצר המתאר את ניסיון השרת למלא אחר הבקשה שנשלחה, שדות מידע לגבי הודעה תשובה והשרת ולעתים קיים גם גוף הודעה.

לפרוטוקול קיימות מספר גרסאות. הגרסה הראשונה פורסמה בשנת 1991 ונקראת 0.9, בשנת 1996 פורסמה גרסה חדשה ומושפра כגרסה 1.0 השינויים העיקריים שהכילה היו שיטות בקשה שנוסףו וכן ציון גרסת הפרוטוקול. הגרסה האחרונה שהתפרסמה בשנת 1999 לפרוטוקול זה היא גרסה 1.1 ולמרות שגם נעשו ניסיונות נוספים לפרסום גרסאות חדשות הם לא יצאו לפועל. השינויים שנעשו בגרסה 1.1 הם בין היתר הוספה והסרה של מספר בקשות, אפשרות לחבר יחיד עבור מספר בקשות, תמייכה ב virtual hosts ועוד.

עד גרסה 1.0 הלקוח נאלץ לפתח ולסגור את החיבור מחדש פעמים רבות, גם כשהיא ידוע מראש על מספר בקשות שעתידות להישלח אליו שרת. סגירה של החיבור כאשר ידוע שמיד יתבקש לפתח אותו מחדש, ובקשה לפתיחת שלו מיד לאחר מכן, יוצרת עומס מיותר. לכן ל��וחות ושרותים ניסו ליים שיטה לשימוש בחיבור יחיד עבור מספר בקשות. השיטה שIOSMAה ייעלה מצביים שתוארו לעיל, אך יקרה מספר בעיות בחיבורים לשרתים proxy.

גרסה 1.1 שיפרה את המנגנון כך שהיא ניתנת להשתמש בו גם עם שירותי proxy. המנגנון נוצר כך שיתאים גם למנגנון הקודם וגם לשירותים אחרים יותר. בנוסף, תוארו מספר מנגנונים ומוסכמות לשימוש יעיל ב-persistent connections והוקדש קוד מצב מיוחד (מספרו 100) לניצול נוסף של המנגנון.⁷⁷

שימוש נוסף ב프וטוקול זה הוא HTTPS אשר למעשה מהו זה שימוש בפרוטוקול HTTP על גבי שכבת ssl/tls המאפשר יכולות אבטחה לתקשורת HTTP רגילה. שימוש HTTPS מקנה אימומות זיהוי, חיבור עמיד בפני התקפות MITM והצפנה דו כיוונית של תעבורת המידע בין הלקוח לשרת.⁷⁸.

בחרנו להשתמש ב프וטוקול זה על מנת לשרת החישובים אותו נקם ושרת anyway אשר יספק מידע לשרת החישובים יתקשרו. הבחירה בפרוטוקול זה נעשתה מאחר והוא נוח וקל למימוש, התקשרות המתוירת בין השירותים לא נעשית בתדריות גבוהה, ומכוון שבשרת של anyway ישנה תשתייה מוכנה של HTTP שנבנתה לצרכים אחרים ומוכנה לשימוש.

עם זאת, לצורך התקשרות בין האפליקציה לשרת החישובים אנו נדרש לתקשרות שתיהה מהירה מאוד, פתוחה (stream), אין צורך ליצור אותה כל פעם מחדש (每当). הנתונים אותם השירות שולח לאפליקציה קטנים מאוד. על התקשרות לתמוך בפניות מרובות לשרת ועל התקשרות להיות דו כיוונית. לצרכים אלו פרוטוקול התקשרות HTTP אינו רלוונטי לנו מאחר ואנו מחפשות מינימום של תעבורת על מנת ליעל את מהירות האפליקציה, מה שלא קורה במימוש פרוטוקול HTTP וכן מאחר וHTTP אינו מספק תקשורת דו כיוונית.

בחרנו אם כן לבדוק אופציות של פרוטוקולי תקשורת וטכנולוגיות אחרות: REST, MQTT, WebSocket

3.8.3 טכנולוגיית MQTT

בטכנולוגיה זו משתמשים בפרוטוקול מסווג Machine to Machine. הפרוטוקול הוא בעל משקל זעיר ובשל כך נטמן גם על ידי חומרות קטנות מאוד.

MQTT הננו פרוטוקול מסווג תעבורת מידע על ידי רישום והפצה וההודעות מתקבלות ונשלחות בו בצורה א-סינכרונית. הפרוטוקול עוצב כך שהוא פותר את אתגר החיבור המהיר לרשת ולכון פרוטוקול זה הננו אידאלי בצריכת חשמל נמוכה ובהעברת מידע על פני רוחב פס נמוך זאת מאחר והכותרות בו הננו מינימליות וגודל הודעה/חבילה שנשלחת בו הננו קטן ככל הניתן. לכל הודעה ניתן לצרף payload - מטען בגודל של עד 256MB.

בנוסף, ניתן להחליט על סוג השירות המתבקש: שליחת הודעה כך שהן יכולות למכת לאיבוד, שליחת הודעה באופן שהן לא יאבדו אך יתפרקו ויגיעו ליעד יותר מפעם אחת ושליחת הודעה כל שה הודעה לא תאבד ותגעה ליעד בבדיקה פעם אחד. העיקרון הנמהה בסוג השירות הוא "כל שאיכות השירות גבוהה יותר כך הביצועים נמכרים יותר".

יתרונו נוסף בפרוטוקול זה הוא קלות המימוש שלו. הקוד הוא open source וקל לאמץ ולשנות אותו בהתאם לצרכים.

באם נבחר להשתמש בפרוטוקול זה לתקשרות בין השירות ללקוח שהוא היישום במכשיר ס마רטפון נוכל להרוויח מכך שההודעות הנשלחות בו הננו קטנות וכן מכך שהפרוטוקול צריך אנרגיה נמוכה ובכך שומר על אורך חיי סוללה ארוכים.⁷⁹

REST 3.8.5

הו **REST** - **R**epresentational State Transfer העיקרי הוא הגדרת משאב מערכת שמצו בו ערך שונה ממקורו המקורי רשות. הרעיון והצרcn.⁸²

התפיסה הארכיטקטונית ב-**REST** היא תפיסת שירות-לקוח. תפיסה זו מחייבת קיום ללקוח ושרות. ללקוחות יוזמים פניות לשירותים המכילים בקשנות. השירות מעבד את הפניה, ומছזר תשובות מתאימות. הייחוד של **REST** הוא בכך שהתוכן המועבר מהלקוח לשירות הוא ייצוג של משאב. המונח משאב בהקשר זה הוא מונח כללי ורחב מאוד וכל מושג קוهرנטי יכול להיות משאב.

בכל מצב נתון הלקוח יכול להיות בתהליך של שינוי מצב או במצב מנוחה. במצב של מנוחה הלקוח יכול להיות באינטראקציה עם המשתמש, אבל איןו תופס משאים בשירות. הלקוח שולח פניות כאשר הוא מוכן לעبور למצב חדש. כאשר קיימת פניה אחת או יותר שטרם הסתיימו הטיפול בהן, הלקוח נמצא במצב של מעבר ממצב למצב. כאשר הלקוח נמצא בתהליך של שינוי מצב, השירות מעבד את המסמך המתאר את מצב הלקוח. המסמך עשוי להכיל קישורים, עשוי להיות בהם שימוש בפעם הבאה שהלקוח יבקש לשנות את מצבו.

לצורך ביצוע אינטראקציה עם המשאב היישום צריך רק את זיהוי המשאב ואת הפעולה שהוא מעוניין לבצע. הוא אינו צריך לדעת על מצבים קודמים, נתונים בזיכרון מטמון ושרותים מתווכים כגון שירות פרוקסי. הסיבה לכך היא שככל המידע נשמר בשירות.

חסורונות השיטה - לא מדובר בתקשורת דו כיוונית וכן השירות "שוכח" ללקוחות ברגע שהוא ענה על הבקשה שלהם כך שבבקשה כלשהי חדשה שוב יctrיך לעבור מידע על זהות הלקוח ולכון מדובר בחבילות גדולות בנוספּ השיטה איטית באופן ניכר ביחס לשיטות אחרות שנבדקו.⁸³

Websocket 3.8.4

פרוטוקול זה מאפשר לבצע תקשורת דו כיוונית על גבי חיבור בודד שנשאר פתוח לכל אורך התקשורת בין הלקוח לשירות. על מנת להבין את תרומתו של פרוטוקול זה לתקשורת דו כיוונית נרצה להקדים מעת ולפרט על הבעיה: רשות האינטרנט הקלאסית עובדת בתבנית המבוססת על "בקשה-תגובה" של פרוטוקול **HTTP**. הדפדפן טוען את עמוד האינטרנט ועל מנת שיקרו שינויים בשרת המשתמש צריך לבצע פעולה כלשהי בדף האינטרנט וייצור בקשה מהשרת, הדבר גורם לייצור חיבור בין הלקוח לשירות, קבלת התשובה וניתוק החיבור. אמנם טכנולוגיית **AJAX**⁸⁰ * אשר נכנסה לשוק היפה את אתרי האינטרנט לדינמיים יותר אך עדין, סוג תקשורת זה מבוסס על פרוטוקול **HTTP** ככל מרעדיין רק בהתערבות של הלקוח מתקובל מידע מהשרת.

* טכניקה ליצירת יישומי רשת אינטראקטיביים המבוססים על קוד המורץ במסגרת דף HTML בלבד, ולא כישום מרובה דפים, בכך מושגת שיפור חווית המשתמש והאצת מהירות הטעינה של דפי האינטרנט, מאחר שהיא מאפשרת לעדכן רק חלקים מובוקשים בדף האינטרנט, ללא צורך לטען את הדף כולו מחדש במחשבו של המשתמש.

אתרי האינטרנט גדלו ודרשו יותר העברות של נתונים הבעיה-ב-HTTP הייתה שבל פעם שהדפדפן היה צריך לקבל נתונים מהשרת, הוא היה צריך לשלוח בקשה לשרת כל פעם מחדש כדי לבדוק האם ישנו נתונים חדשים שעליו לקבל והאם להשתמש בו או לא. פתרון יצירתי שהומצא על מנת לפתור בעיה זו היה HTTP Long Polling, על פי שיטה זו הלקוח שולח בקשה לשרת ורק כאשר לשרת ישנו מידע להעביר ללקוח הוא עונה לבקשתה זו, כל זמן שאין לשרת מידע עבור הבקשתה שנשלחה מהלקוח השרת מחזיק את הבקשתה פתוחה, הלקוח יכול ליצור מספר בלתי מוגבל של בקשות ובכך להעמיס על השרת. שיטה זו גרמה למעשה להתקפת DoS מובנית אותה יצר מפתח האתר כלפי השרת. בנוסף, אם מדובר באתר אינטרנט, הדפדפניים מגבלים את מספר הבקשות שיכל הלקוח לשולח אל השרת על מנת שהשרת לא יוציא בבקשתם מהדפדפן. בשיטה זו ישנה צריכת משאבים גבוהה ומיותרת, שפוגעת בזמןות התקשרות הנחוצה.⁸¹

בעיה נוספת הקשורה במודול HTTP הנפוץ היא תעבורת של מידע שאינו נחוץ עבור הלקוח ונובע מה headers בפרוטוקול זה. תעבורת מידע זו יכולה להגיע עד 50% מתוכן הבקשתה או התגובה מהשרת ולהאריך משמעותית את זמן התגובה.

וכאן נכנס לתמונה פרוטוקול websocket, ה프וטוקול מתבסס לצורך הקמת התקשרות בין השירות ללקוח על פרוטוקול HTTP ורץ על אותו מודול ip/tcp. לעומת זאת, התקשרות ב-websocket מחולקת ל-2: לחיצת היד (Handshake) והעברת נתונים. כאשר אובייקט websocket נוצר מתבצעת "לחיצת יד" בין הדפדפן לשרת. הדפדפן שולח לשרת בקשה GET Upgrade ושרת "משדרג" את הפרוטוקול ל- websocket שהוא פרוטוקול מבוסס TCP. בקשה ה"שדרוג" היא גשר בין פרוטוקול ה-HTTP ל- websocket. מרגע זה ועד לסיום החיבור בין השירות ללקוח, השירות וגוף הלקוח יכולים לבחור לשולח הודעה בכל רגע נתון. אחרי לחיצת היד בין הלקוח לשרת המתבצעת פעם אחת בלבד ומאפשרת במסגרת החיבור הלקוח והשרת שולחים אחד לשני חבילות pong/opong. כשהגוף מתקבל מהשרת - הלקוח חייב לשולח חבילת pong, הפרוטוקול משתמש במנגנון זה כדי לוודא שהלקוח עדין מחובר, מעין מסגרת שליטה קבועה בה pong מיוצג על ידי המחרוזות 0x0A opcode ואילו opong מיוצג על ידי 0xA0. כאשר הלקוח מקבל pong הוא חייב לשולח בחזרה pong עם אותה הודעה מדויקת. סגירת החיבור בין הדפדפן לשרת מתבצעת על ידי שליחת חבילת סגירה (opcode=0x80), הנמען שולח מסגרת סגירה בתגובה ולאחר

מכן סגירת החיבור מתבצעת על ידי שני הצדדים⁸⁴. שדה opcode מכיל מחזורת במבנה `9x0` כאשר ההודעה נשלחת מהשרת ו-`A1x0` כאשר ההודעה נשלחת מהלקוח. Mask הוא שדה באורך 32 בתים שערכיו אקראים וمبוצעת עליו פעולה XOR בית אחר בית. פעולה זו מונעת מהדפסון להיות מנוצל על ידי תוקף. פרוטוקול websocket עצמו אינו מבחין בין חוממות אש או שרתי proxy אך הוא מבוסס על פרוטוקול HTTP (בשימוש הרגיל שלו או המוצפן- HTTPS) ולכן לפרוטוקול websocket קיימות שתי גרסאות : ws עבור חיבורים רגילים (בדומה לHTTP) וwss לחיבורים מוצפנים (כהקלה HTTPS). חלק משרתי proxy הנם שkopים והפרוטוקול יעבד אותם בצורה חלקה אך אחרים, كالו שמהליפים את headers שמנגנים מבקשת HTTP המקורית או שכאש מגיעה אליהם בקשה עם header עם אותו שם לא מכירם השרות מוחק את header שאינו מוכר לו ו מעביר את הבקשה הלאה. התנאות זו למעשה מונעת מהפרוטוקול להשלים את "לחיצת היד" בין השרת ללקוח וחוסמת את האפשרות לבצע חיבור בין השרת ללקוח.⁸⁵ אם מידע לא מוצפן של websocket עבר דרך שרת proxy מפורש או שקוฟ שאינו תומך בפרוטוקול הסיקורים שהחיבור לא יתאפשר הם גדולים. כאשר מוצפינים את המידע ישנים סיקורים טובים יותר שהמידע עבר והחיבור לא ייסגר מאוחר והחצנה יוצרת אומנת בין לקוח hsass ושרת websocket אשר עבר דרך שרת proxy אזוקן מפורשים. במקרה, אם מדובר בשרת ביןיים שקוフ ייתכן וההודעה בכל זאת תעבור. המשקנה העיקרית הנובעת מכך היא שבשימוש בגרסה המוצפנת של פרוטוקול websocket יש סיקורי טוב יותר שהתקשרות תעבוד אך יש לה מחיר- שימוש מוגבר יותר במשאים לצורך הצפנה ופונוח.

חסרוןתו של פרוטוקול זה מעבר לביעות הבטיחות בו שיפורטו בהמשך היא שישנם דפדים וגרסאות שלא תומכים בו : 9 Explorer ומטה ואנדרואיד גרסה 4.3 ומטה. בנוסף, עבר הגדרות רשות מסוימות ושרת proxy מסוימים, websockets לא יעבד⁸⁶.

כפי שנכתב לעיל, פרוטוקול websocket ישן בעיות בטיחותיות : אם תוקף יכול "להסניף" את חבילות הHTTP הרי שגם את חבילות hsass הוא יכול "להסניף". לאחר ופרוטוקול websocket הינו הראשון שאפשר מספר חיבורים בלתי מוגבל מהליך לשרת היחיד הוא מזמן למעשה התקפות DoS על אותו השרת. בפועל, שימוש הפרוטוקול לדפדים ישנה הגבלה והוא נעה בין 200 חיבורים (firefox) ל-2970 חיבורים⁸⁷ (safari). נוסף על כך, אם ניתן לעזר או להזירק את חבילות websocket אפשרי יהיה גם לעקוף אותו : אחת מהבעיות מפרצות הבטיחות בפרוטוקול נקראת **Cross-Site (WebSocket Hijacking) (CSWSH)**. עקב האכילס של ה프וטוקול, הוא שלב ה"shedrog"- חבילת HTTP שנשלחת בבקשת upgrade .get RFC של הפרוטוקול נכתב כי השרת רשאי לבחור בכל מנגנון לאימוט המשמש כל עוד המנגנון נתמך ע"י זיהוי גני של HTTP. דוגמאות למנגנונים כאלה הם cookies, TLS ועוד. בשלב זה שולח HTTP את כל פרטי ה-LWSH תוקף יכול לחטוף חבילות אלו ולגנוב את זהות הלקוח או השרת. ניתן למנוע פרצה זו על ידי בדיקה בשרת של header בשם origin ולודא שתוכנו זהה לorigin שבקש את המעבר

מפרוטוקול HTTP websocket tokens רנדומליים וייחודיים בתהיליך לחיצת הידיים ובדיקה חוזרת שלהם בשרת יכולה גם כן לעוזר.⁸⁸

פרצה נוספת דומה ב프וטוקול Cross Site Scripting (הזרקת קוד זדוני לאתר אינטרנט על ידי משתמש,⁸⁹ XSS). זאת ועוד, חומרה אש ותוכנות מסווג DLP (תכונות המונעות מהמשתמש לשלוח בטיעות מידע רגיש) אין תמיד מודעת לפוטוקול זה ולכנן לא בוחנות ומנתחות תכילות websocket ממושכות ובשל כך אין יכולות לזהות תכונות זדוניות או דליפת מידע. נקודה אחרת שיש לשים לב אליה היא שפוטוקול websocket אינו מתעסק עם נושא ההרשאה והזהות ולכנן יש לדאוג כי פרוטוקולים ברמת האפליקציה יגנו על מידע רגיש.⁹⁰

לבסוף ומהסיבות הבאות בחרנו להשתמש בפוטוקול websocket: סיבה ראשונה: כיון שצד הלוקו הוא הצד mobile ולא מותאם להודעות תקשורת כבודות העדפנו להשתמש בחיבור מסווג websocket כיון שהוא קטו בכ-50% מהודעה זהה שתשלח בפוטוקול http בגלל ריבוי headers. כך שנחסכה תקורה גבוהה מאוד.

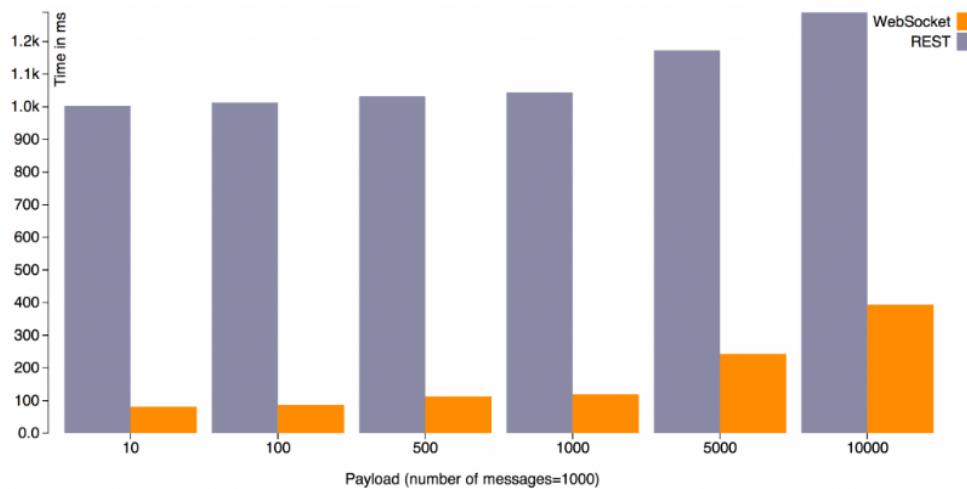
סיבה שנייה, למروת שבחברו websocket השירות צריך להחזיק את החיבורם פתוחים לכל אורך התקשרות הדבר מעודד מהירות בהעברת מידע כיון שאין צורך handshake בכל שליחת תכילה, כיון שמדובר בפיתוח לmobile מהירותו היא דבר חשוב ביותר.

סיבה שלישית, למروת פרצות האבטחה שפורטו לעיל, כיון שצד הלוקו מפותח עבור משתמשי mobile ישנה אבטחה ברמת האפליקציות של שמירת מידע ולכנן אין חשש שאפליקציה זדונית תגנוב את המידע. סיבה נוספת לבחירתנו בפוטוקול זה למטרות פרצת הבטיחות הקיימת בו שתוארה לעיל היא שעלה מנת לבצע את הפריצה בפועל יש להחזיק במספר חלונות / מסכים פתוחים של האפליקציה או התכנה בו זמנית. במכשירי סלולר הדבר כמעט בלתי אפשרי ולכנן הסיכויים שתיווצר פרצה דרך דרך התקשרות בפוטוקול זה כאשר צד הלוקו שלו הוא mobile נמוכים מאוד. זהה גם הסיבה לכך שבשימוש בפוטוקול במכשירי סלולר אין שליחה של שדה המזיגן.

בחרנו למשח חיבור שאינו מאובטח משתתי סיבות עיקריות:

א. אין לא מעבירות מידע רגיש בין השירות והлокו בגל שהמידע המועבר אינו רגיש ורצינו לשמר על תכילות מידע קטנות ככל האפשר על מנת לאפשר מהירות ולא להעמעיס על החיבור.

ב. על מנת למשח חיבור מאובטח מצד הלוקה קיים צורך **certificate** מיוחד, בהתייעצות עם החברה הוחלט שכיוון וגם כך המידע אינו ניתן לגיש אין טעם ליצור חיבור מאובטח.



איור 6-3 השוואת בין זמני השליחה והטגובה בטכנולוגיות websocket לעומת REST

3.9 העברת מידע

העברת המידע על גבי התקשרות בין השירותים עצם ובין השירות ללוקה- האפליקציה⁹¹ יכולה להיעשות במספר דרכים. להלן נביא שתי דרכים עיקריות נפוצות בהן נעשה שימוש להעברת נתונים על גבי פרוטוקולי תקשורת.

XML 3.9.1

XML הן ראשי תיבות של eXtensible Markup Language המהווה תקן לייצוג נתונים במחשבים. שימוש ב-XML מקל על החלפת נתונים בין מערכות שונות שפועלות על גבי תשתיות שונות. תקן ה-XML לא מגדיר איזה מידע יוצג אלא מגדיר כיצד ליציג מידע באופן כללי והוא שיך למשפחה שפות הסימון. מבחינה טכנית, XML אינה שפה שכן אף על פי שהתחביר (syntax) שלה מוגדר היטב, אין לה אוצר מילים ולא בקרת זרימה. תכונות שפת הסימונו `xml` כוללות ייצוג מידע באופן טקסטואלי, שמירת המטא-מידע בלבד עם המידע עצמו. ככלומר, שמירת תיאור הנתונים עם הנתונים עצמם, סידור המידע במגוון צורות ולא רק בטבלה כמו ברוב הפורמטים האחרים. ב-XML טבעי מאוד לסדר מידע באופן היררכי (עץ).

ה-XML שימושי כמעט בכל תחום של מחשבים וחשיבותו הגדולה ביותר באה לידי ביטוי בעיקר בחיבורים בין מערכות שונות⁹².

JSON 3.9.2

ראשי התיבות של JSON הן JavaScript Object Notation כלומר, ציון אובייקטים. JSON הנו מבנה קל משקל להחלפת מידע, קל עבור בני אנוש לקרוא ולכתוב וכלfür פונCTION עיי' מכונות. הוא מבוסס על

תת-מערכת של שפת התכונות Javascript. פורמט זה הוא מבנה כתוב חופשי לחלוטין משפה אך משתמש במוסכמוות המוכרות לתוכנתים של משפחות שפות C. תכונות אלו עושות את JSON לפורט אידיאלית עבור החלפת מידע.

JSON בניו על שני מבנים: אוסף של זוגות בעלי שם וערך ורשימה מסודרת של ערכים. מבני המידע האלו הם אוניברסליים כך שלפעה כל שפות התכונות המודרניות תומכות בהם בצורה זו או אחרת⁹. בחירותנו במבנה זה להעברת המידע בין השירותים נובעת מכך שפתחי השירות *wayway* בחרו בפורט זה בمعנה לשאלנות שמייעות אל שירות זה מקורות חיצוניים ושל יתרוניותו הרבים כפי שהם מצויינים לעיל.

ערכינו השוואת המונה את יתרוניותה וחסרוןותיה של כל שיטה להעברת נתונים על מנת להגיע להחלטה. יתרונות העברת נתונים באמצעות XML הם:

בשיטת העברת הנתונים על ידי XML הסימון הוא כללי: ניתן ליצור ניבים שונים (דיאלקטים) לצורך כל מטרה שהיא. נוסף על כך, יצירת סכמת XML עצמה מאפשרת ליצור טיפוסי נתונים ישירות דרך. קיימות שפות המתרגמות את המידע הכתוב בXML לצורות פלט בפורט אחר. ניתן לחלץ מידע השמור עמוק בתוך מבנים מקוונים באמצעות ספריות ומחלקות אשר פותחו לצורך כך ישנה תמייה מובנית בXML המוטמעת ב-*namespaces*.

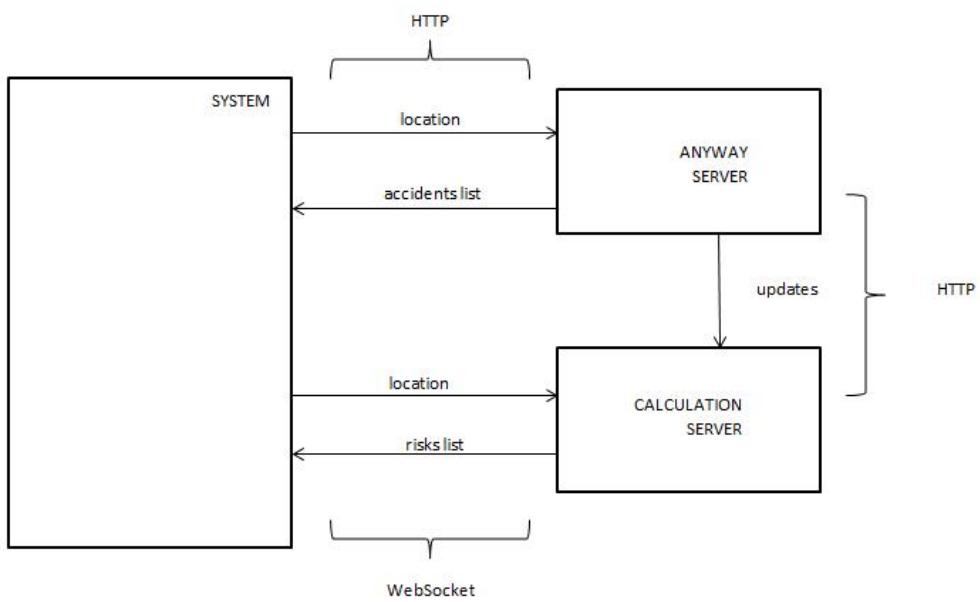
חסרונה העיקרי של השפה הוא ריבוי המלל שבבהשוואה לJSON. אותו מידע יכול לעבור ע"י מבני JSON ולהשתמש בהרבה פחות כמות מידע.

יתרונותיו של JSON הם הסינטקט הפשט שלו, אשר מוריד את התקורת משמעותית יחסית לXML קלות השימוש בו, קיימת סכמת JSON לתיאור ובקרה על מבני משתנים, ישנה ספרייה אשר מחלצת את כל הנתונים, גם אלו המקוונים עמוק בצורה חלקה ונוחה.

חסרונו מבנה JSON הוא שאין מספר רב של טיפוסי נתונים אשר נתמכים על ידו. שקלנו את היתרונות מול החסרונות כפי שהוצעו לעיל, והחליטנו להשתמש במבנה JSON לצורך העברת הנתונים. הסיבה העיקרית לבחירה זו נעוצה ברצונו לモער במידת האפשר את התקורת בתקשורת בין השירות לאפליקציה שכן מדובר במכשורי סולאר אשר נפח ומהירות הגלישה שלהם מוגבלים. בנוסף, מצאנו שהמבנה JSON נוח ואינטואטיבי עבורנו יותר מאשר שימוש ב-XML.

3.10 סיכום

לאחר סקירה זו החדשויים והחלטות שהתקבלו בה בנוו תרשימים תיאור פרויקט מדויק יותר:



איור 3-7 תרשים מערכת סופי

כפי שניתן לראות מהתרשימים, בחרנו להוסיף שרת חדש שיחוהו שרת החישובים. שרת זה יתקשר עם האפליקציה באמצעות פרוטוקול websocket ובמקביל יתקשר עם שרת anyway ויקבל ממנו עדכונים כל עשרה ימים באמצעות פרוטוקול HTTP. המערכת תתקשר עם שני השירותים בהתאם במצב בו המשתמש בוחר להיות - הצגת תאונות ומקבצי תאונות תאפשרת במאפשרות פניה לשרת anyway על גבי פרוטוקול HTTP כדי שקורה כיום ואילו הצגת אזורי סיכון על גבי המפה באמצעות צבע והתראות צליל תהיה אפשרית באמצעות פניה לשרת החישובים על גבי פרוטוקול .websocket

4. המערכת שנבנתה בפועל

בבנייה המערכת עבדנו נפרד על פיתוח שרת החישובים ופיתוח האפליקציה במודל שרת - לכה. שלב התקשרות היה השלב היחיד בו הפיתוח באפליקציה ובשרת נעשה בו זמנית ביחד ולאחר מכן כתבנו את התקשרות בין שרת החישובים. את אתחול בסיס הנתונים ביצענו נפרד וצרכנו את קובץ בסיס הנתונים לשרת החישובים.

כיוון שהייתה חלוקה מסווגת בפיתוח העדפנו להפריד גם את פירוט הביצועים שיבוא להלן:

4.1 פיתוח שרת החישובים

שרת החישובים אשר נכתב על גבי סביבת העבודה האינטרנטית²⁹, מהוות למעשה נקודת חיבור שכן הוא מתקשר עם האפליקציה ועם שרת *anyway* וכן התחילנו בפירוט פיתוחו ראשון. תחומי "אחריותו" של שרת זה הם :

- חלוקת המפה לריבועים הנמצאים אך ורק בשטחה של מדינת ישראל
- חישוב סיכון עבור כל ריבוע והתאמת צבע הממחיש את רמת הסיכון
- אתחול ושמירת בסיס נתונים חלקי של תאות הדרכים
- תקשורת עם לköחות האפליקציה באמצעות פרוטוקול *websocket*
- שליחת מידע על רמות סיכון ללקוח בהתאם בקשו

4.1.1 חלוקת המפה

חלוקת המפה של מדינת ישראל לריבועים נעשית פעם אחת, בהפעלתו הראשונה של השירות, והנתונים מגיעים בקובץ *csz*, נקודת הציון של הריבוע, רמת הסיכון שלו ומספר התאות על פי השירות ובהתאמה לאלגוריתם החישוב, ייכנסו לתוך מאגר מידע. לאחר מכן, פעם בזמנן כלשהו אם היו עדכונים ושינויים בסיס הנתונים של תאות הדרכים בשרת *anyway* יבוצע חישוב מחדש לגבי רמת הסיכון של כל ריבוע בו חל שינוי.

בחרנו להתייחס להצגה העשוריית של הקואורדינטות הגאוגרפיות אשר על המפה מטעני נוחות – הלולאה אשר תחלק את המפה לריבועים תהיה פשוטה יותר וגם הבדיקה עליה מצדנו תהיה קלה יותר לקריאה והבנה.

חלוקת עצמה נעשית על ידי לולאה מקוונת כך שהlolאה החיצונית מתחילה ב*long* - קו האורך של הנקודת הימנית ביותר של המלבן התחום את גבולות מדינת ישראל ומתקדמת בצעדים של β עד להגעה ל*long* של הנקודת השמאלית ביותר במלבן. הלולאה הפנימית מתחילה *lat* של הנקודת התחתונה ביותר

באוטו המלון ומתקדמת בצעדים של א' עד להגעתה *läa* של הנקודת הגבוהה ביותר במלון. כך נוצרו למעשה ריבועים אשר אורכם הוא β ורוחם הוא α כמתבקש.

2.1.4. **חישוב מיקום הריבוע ביחס לגבולות מדינת ישראל**

בחלוקת המפה לריבועים תחמו את גבולות מדינת ישראל במלון וחילקו אותו לריבועים וכטזאה מכך, התקבלו גם ריבועים אשר אינם שייכים למדינת ישראל. ריבועים אלו אינם מוכנסים לבסיס הנתונים של הריבועים שכן מעבר לכך שהם חרוגים מגבולות הפרויקט אין לנו מידע על תאונות שהתרחשו בהם. על מנת ליזהות האם ריבוע כלשהו נמצא בתחום שטחה של מדינת ישראל תחמו קודם את גבולות מדינת ישראל בתחום צורה מסווג מצולע (*convex*). מצולע זה הוא אוסף של נקודות ציון שמתייחת קו ישר בין נקודה אחת לשנייה על פי סדר יצור מצולע שצלעתו חוץ גבולות מדינת ישראל. לאחר מכן, בנינו פונקציה בשם *point_in_poly* המתקבלת שתי נקודות ציון אשר מהוות הפינה השמאלית התחתונה של הריבוע. פונקציה זו מחזירה אמת אם הריבוע שיק (נמצא) למדינת ישראל ושקר אחרת. החישוב לגבי שייכותו או אי שייכותו של הריבוע מתבצע על פי אלגוריתם המרת الكرן אותו מימושו. בימוש האלגוריתם ערכנו על כל שני קדוקודים סמוכים (על צלעות המצולע, במילוי אחרויות) ובדקנו אם הקו החוצה מנקודה שבשאלה חותך את הצלע שנוצרה. ישנו משטנה בוליאני המוחזק ל"שקר" ומתהפק (משקר לאמת ומאט לשקר) בכל פעם בה חוצה הקו את צלעת המצולע עד שהוא מגע לגבול המלון התוחם את המלון. ערכו של משטנה זה החזר מהפונקציה אומר אם הנקודת בשאלה נמצאת בתחום המצולע או שלא. (אמנם ישן כמה פונקציות מספריות שונות בשפת *python* המוחזק האם נקודת נמצאת למצולע, מוחזק לו, או על גבי צלעותיו. פונקציות אלו מתייחסות למצולע המוגדר על גבי מסך מחשב ולא למצולע שנקודותיו מוגדרות על ידי נקודות ציון. יכולנו להמיר את נקודות הציון היוצרות את המצולע לנקודות של (y,x) ביחס למסך מחשב (כלומר, יחסית לראשית הצירים), להמיר גם את הנקודת שבדיקה ולהשתמש בפונקציות מספריות *python*. בחרנו שלא לעשות כן בעיקר על מנת לחסוך זמן ריצה: המרת כל נקודות הציון של המצולע לנקודות יחסית לראשית הצירים ואז בדיקה של הנקודת כוללת מעבר פעמיים על כל נקודות המצולע: פעם להמרה ופעם לבדיקה. באלגוריתם בו השתמשנו, כפי שפורט לעיל ישנו מעבר פעם אחת בלבד על כל נקודות המצולע).

3.1.4. **אתחול בסיס הנתונים**

כאשר נגשנו לאותחול בסיס הנתונים נוכחנו לדעת כי החלוקה יצרה לעלה ממיליאן ושבע מאות אלף ריבועים. בשל כך, על מנת שנוכל להעריך את זמן הכנסת הנתונים ולעורך בקרה מסודרת חילקו את כל טווח קווי הרוחב לעשרה חלקים שווים ובכל פעם הרצינו לולאה חיצונית על כל טווח נקודות האורך והלולאה הפנימית התייחסה א' ורק לשערירות משטנה מקווי הרוחב. יתרוון נוסף אשר חלוקה זו מקנה הוא בעדכוון בסיס הנתונים אשר יכול להישנות בשלבים או בצורה נקודתית יחסית לשטח כלשהו במפה ישראל.

כאמור, בסיס הנתונים בו אנו משתמשות הוא SQLite אותו עוטף המודול `sqlite3`. כל רשומה בבסיס הנתונים מכילה את השדות הבאים: `id` - מספר סידורי יחיד מסוג `integer` המתחיל ב-1 ומשמש גם כ`primary key`. שני השדות הבאים, `lat`, `long`, מיצגים את הנקודה השמאלית התחתונה של הריבוע והם מסוג `real`. שדה נוסף, מסוג `integer` הוא `H` המכיל את הערך המתאים לרמת הסיכון בربיע זה. השדה האחרון הוא מסוג `blob`, אשר שומר את המידע בדיקוק כפי שהוכנס אליו. בתוך שדה זה הנקרא `acc` מאוחסן מערך בן שלושה עשר תאים: משנת 2005 ועד לשנת 2017 כך שכל תא במערך מציג את מספר התאונות שקרו בשנה מסוימת בהתאם לאלגוריתם החישוב. בחרנו לשמר את מערך תאונות זה דווקא בעמודה מסוג `blob` ולא להכין עמודה לכל שנה במערך וכן שלא לשמר את המערך כטקסט על מנת להקל על ההמרה וכן על מנת שלא ליצור מצב בו נדרש להוציא עמודה לטבלה בתחלת שנת 2018.

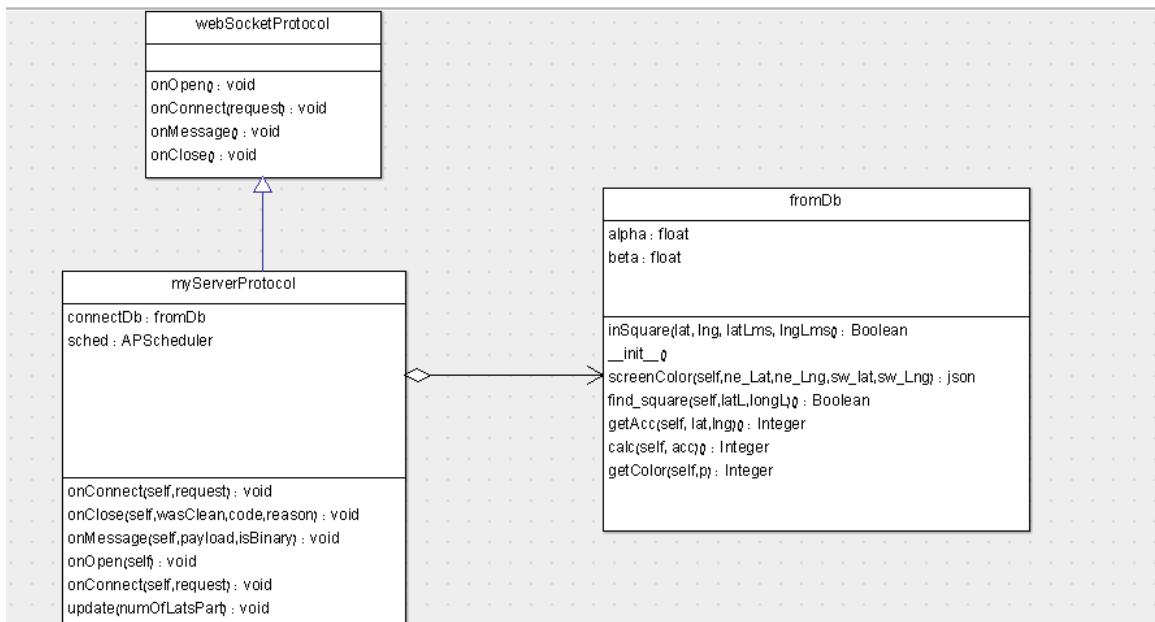
| INTEGER | FLOAT | FLOAT | INTEGER | BLOB |
|---------|-------|-------|---------|------|
| Id | lat | long | H | acc |

איור 1-4 פירוט עמודות בסיס הנתונים

הריבוע מיוצג רק על ידי הנקודה השמאלית התחתונה שלו שכן ערכי β , a ידועים לשרת ולאפליקציה ציה ובאמצעותם ניתן לשחזר את כל שאר פינות הריבוע. בכך מושג בסיס נתונים רזה יותר שתגובהו מהירה יותר וגם התקשרות בין השירות לאפליקציה קטנה משמעותית.

4.1.4 תרשים UML למחלקות השירות

להלן פירוט המחלקות, השדות והmethodות כפי שהם מופיעים בשרת:



איור 2-4: תרשים uml צד שירות

DBFunc 4.1.5 מחלקה

המחלקה DBFunc בשורת מייצגת ואחרהית על כל בסיס הנתונים : קריאה, עדכו ושליפת נתונים על פי תנאים משתנים. למחלקה ישם שני שדות סטטיים : α המייצג את רוחב הריבוע וערךו הוא $0.0013 \pm \beta$ המייצג את אורך הריבוע וערךו הוא 0.0018 . אובייקט המחלקה מורכב גם מהשדות `conn`- אובייקט החיבור לבסיס הנתונים אשר באתחול אובייקט המחלקה (במетодה `__init__`) מתחבר לבסיס הנתונים של הריבועים הנמצא בשורת והשדה `c` המשמש cursors שבאמצעות קריאה למתודת `execute` שבו ניתן לבצע פקודות מסווג SQL.

מתודות המחלקה הן :

screenColor : מетодה זו מקבלת כפרמטרים שתי נקודות ציון נגדיות ומוחירה את כל הריבועים הנמצאים בין שתי הנקודות האלו. שתי הנקודות המגיעות למетодה מייצגות למעשה את הפינה העליונה הימנית של המפה המוצגת על גבי מסך האפליקציה ונקראות `ne` (מלשון north east, צפון מזרח) והנקודה השנייה המגיעה למетодה מייצגת את הנקודה השמאלית התחתונה של המפה המוצגת על גבי מסך האפליקציה ונקראות `sw` (מלשון south west, דרום מערב).
ייתכן בהחלט כי הנקודות המגיעות למתודה נמצאות בתחום ריבוע ולאו דווקא על נקודת הריבוע כפי שהיא מצוינת במסיקת הנתונים. לכן הוספנו לנקודת ה- α ערכי β ל-`lat`, `lng`, `a`, `neLat`, `neLng`, `lat`, `lng` בהתאם ויאלו מנקודות `sw` והורדנו ערכים אלו מ-`lat`, `lng`. בכך הבטחנו עוד ריבועים שישלחו ללקוח, יוצגו על גבי המפה שבאפליקציה ויכבשו בכיסוי מלא את כל המפה המוצגת באפליקציה.

לאחר מכן, הרצנו חיפוש על בסיס הנתונים ועבור כל ריבוע שנקודת `lat` המייצגת אותו היא בין ערכו החדש של `swLat` ובין ערכו החדש של `neLat` וכן נקודת `lng` המייצגת אותו היא בין ערכו החדש של `swLng` ובין ערכו החדש של `neLng` (המקבילה של שפת python למילה Dictionary) אשר מכיל את השדות `lat`, `lng` ו- `color` כפי שהם מופיעים במסיקת הנתונים. נוסף על כך למבנה JSON אשר מכיל את שכזה שאלת העלה. כל מבנה שכזה משורשר למבנה Dictionary כליל יותר בשם `allData` מספנו כל מבנה שכזה בסדר עולה. אשר בסוף הרצת השאלת הפעלו עליו את פונקציית `dumps` מתוך ספריית `json` אשר הופכת את המבנה אשר בסוף הרצת השאלת הפעלו עליו את פונקציית `dumps` מתוך ספריית `json` אשר נשלח ללקוח.

מתודה עזר במחלקה זו נקראת `inSquare`. המתודה מקבלת שתי נקודות ציון המוחולקות ל-`lat`, `lng`, `lat`, `lng`. המתודה הראשונה היא הפינה השמאלית התחתונה של הריבוע והנקודה השנייה הנשלחת היא נקודת כלשהי. המתודה מחזירה אמת במידה והנקודה השנייה שנשלחה נמצאת בריבוע המיוצג על ידי הנקודה הראשונה שנשלחה וSKU אחרה. למעשה, אם המרחק בין הנקודה בשאלת השאלה לבין גבולות הריבוע קטן מוגדל הריבוע כדי שהנקודה בשאלת השאלה נמצאת בתחום הריבוע. בפונקציה עצמה מימשנו זאת כך :
עבור `p.lat`, `p.lng`, `s.lat`, `s.lng` נקודת הציון של הנקודה בשאלת השאלה לבין גבולות הריבוע :
$$s.lng < p.lng + \beta \quad \text{וגם} \quad s.lng >= p.lng$$

$$s.lat < p.lat + \alpha \quad \text{וגם} \quad s.lat >= p.lat$$

החזר אמת

אחרת: החזר שקר

הmethod הששית המומשת במחלקה זו נקראת **find_square** המתודה מקבלת כפרמטר נקודת ציון ומחזירה 1 במידה והריבוע בו נמצא נקודה זו הנו בהסתברות גבוהה לתאונה 0 - אחרת. הנקודה הנשלהת לפונקציה מגיעה בצורה `lat,lon` והוא הנקודה המצינית את מיקום משתמש האפליקציה. method עוברת על כל בסיס הנתונים ועובד כל ריבוע היא בודקת בעזרת הפונקציה `inSquare` אם הנקודה נמצאת בריבוע ובאים כן האם הריבוע בו נמצא נמצאת הנקודה נמצאת בהסתברות גבוהה לתאונה ומחזירה פרמטרים בהתאם כפי שפורט לעיל.

method נוספת הנמצאת במחלקה זו היא חישוב הסיכון- מימוש האלגוריתם אותו בנינו- ונקראת **calc**. הפונקציה מקבלת כפרמטר מערך בו עבור כל שנה החל מ2005 ישנו תא המכיל את מספר התאונות שקרו בשנה זו על פי אילוצי האלגוריתם. בתחילת מחשבת הפונקציה את המוצע של מערך זה ואת ערך השונות שלו. לאחר מכן, בעזרת מתודות מספרית `math` של שפת `python` כמו `som` המחשבת חזקה ו- `gamma` המחשבת ומחזירה ערכים של פונקציית גמא היישבנו את ההסתברות לתאונה בריבוע אשר התאונות שבו מוצגות על גבי המערך המתתקבל כפרמטר. אם ישנה בעיה או טעות בהכנסת הנתונים למערך method תזורך חריגת ותחזיר 1.

עוד method המומשת במחלקה זו היא **getColor** המתודה מקבלת את הפרמטר `k` המיצג את ההסתברות לתאונה ומותאם לו צבע על ידי הכפלת ההסתברות בקבוע של 1050, כפי שפורטנו לעיל. מסיבות של שמירת נתונים והמרת הצבע כאשר הוא הגיע לאפליקציה עיגלו על ידי פונקציית ההמרה של `python` את ערך `H` למספר הקרוב אליו. method זו משמשת גם כבקרה על חישוב ההסתברות. אם ערך ההסתברות המגיע אליה הוא 1- תחזיר הפונקציה ערך `H=235` כולם צבע כחול. צבע זה יציין לנו שיש לבדוק מה הבעיה נתונים ולדואג לעדכון וטיפול במקרה הקיצון.

method האخונה במחלקה זו היא **getAcc** והוא method אשר דואגת לכל התקשרות וקבלת הנתונים משרת `wayupn` וכן נפרט עלייה בהמשך.

4.1.6 websocket

קיימות ספריות רבות המממשות את ה프וטוקול ובאותם סיבע למתקנת. בתחילת בחרנו בספרייה `websocketServer` : כיוון שהספרייה פשוטה להבנה ולשימוש. הספרייה עוזרת למשתמש את לחיצת היד בין הלקוות לשרת, כיוון שהשתר שבו השתמשנו הוא שרת אינטרנט ללא כתובת IP ישירה, בתחילת התקשרנו לקשר ביניהם. לאחר שימוש צד הלקוות יצרנו חיבור בין הלקוות לשרת ולאחר מספר שניות החיבור היה מותנתק ולכן השתמשנו בתוסף של דפדפן `chrome` בשם `websocket` על מנת לבחון את השירות שלנו, גם בתוסף ה `chrome` התגלו אי יציבות בחיבור ולאחר מחקר עמוק גילינו שזה דפוס חוזר גם

אצל משתמשים אחרים בספרייה זו. כמו כן, הספרייה הייתה מותאמת באופן חלקי בלבד ל命令 python בגרסת 2.7, בתחילת חשבנו לעדכן את גרסה המודול python ל-3.3 ומעלה אך כיוון שבתביד השרת שפיתחנו אמרור יהיה להיכנס לחלק מהשרת של חברת anyway אשר כתוב בגרסה 2.7 בחרנו להשאיר עם הגרסה זו ולאור לכך בחרנו להשתמש בספרייה אחרת.

חרנו ספריות נוספות ולבסוף בחרנו להשתמש בספרייה של authbahn כיוון שלו פיתחנו הtgtalgta כיצובה ביותר וכמו כן מותאמת למ命령 python בגרסה 2.7.

בספריות של מהבahn authbahn קיימות שתי גרסאות:

שימוש ראשון: asyncio - אשר מספק תשתיית יצירת תהליכי משנה בודדים למשל: ריבוב פעולות O/I

תכנות א-סינכרוני המשובץ עשוות מונחים מתכנות סדרתי קלסי ומכלול חריגות רבות עד כדי כך שיישנו דף אינטראקטיבי המפרט את רשימת החריגות השכיחות וכייד יש להימנע מהם⁹⁴.

שימוש שני: Twisted - מנוע רשות מונחה אירופים שנכתב בשפת הפייטווח python ותומך בהרבה פרוטוקולי תקשורת נפוצים - למשל DNS⁹⁵.

בחרכנו בשימוש Twisted כיוון שהתקשרות מונחית אירופים בזמן אמיתי ובנוסף רצינו לשמר על קוד קל ופיטוי לקריאה ותחזוק.

גרסה הפרוטוקול בו אנו משתמשות היא גרסה ברירת המחדל של הספרייה: גרסה 18.

4.1.7 מחלקה MyServerProtocol

המחלקה MyServerProtocol מימושת את "לחיצת היד" בצד השרת ומשמשת לקבלת מידע מהלקוח, שימוש במחלקה FuncDb והחזרת תשובה ללקוח על סמך המידע שהתקבל ממנו.

מחלקה זו מימושת את פרוטוקול התקשרות websocket וקובעת את התשובות של השרת במצבים השונים. המחלקה MyServerProtocol ירושת מחלוקת הבסיס⁹⁶. מחלוקת הבסיס מאפשרת גישה למתודות שמנהלות את האירועים אשר מתרחשים במחלוקת היורשת וכן הגדרת factory שעליו יבוא פירוט בהמשך.

במחלקה ישנו אובייקט גלובלי connectDb שמאפשר חיבור למחלוקת .fromDb

ישנן שלוש מתודות שעל המחלוקת למש על מנת של "לחיצת היד" תתבצע, המתודות הנן מתודות מסווג callback הנקבעות בתגובה לאירוע שمفעל אותן ואלו הן:

המתודה **onOpen** נקראת כאשר ללקוח מבקש ליצור חיבור מסווג websocket המתודה מציגה הודעה על המסך כאשר ללקוח מתחבר ויוצרת את החיבור על ידי קריאה אוטומטית לפונקציית החיבור במחלוקת .WebSocketServerProtocol

המתודה **onConnect** נקראת כאשר ללקוח מבקש לחבר אובייקט מסווג websocket לשרת המתודה מציגה הודעה על המסך כאשר ללקוח מתחבר ויוצרת את החיבור על ידי קריאה אוטומטית לפונקציית החיבור במחלוקת .WebSocketServerProtocol

הmethod **OnMessage** נקראת כאשר מקבלת הודעה מהלכה, המתודה מקבלת כפרמטרים מחרוזת של בתים ודגל.

마חר ש **WebSocket** מסוגל להעביר הודעה טקסט מסווג **UTF8** ומחרוזת בינהarity, הדגל משמש אינדיקציה באם המידע שהגיע מהלכה הנו מחרוזת בינהarity או מחרוזת טקסט. כשההודעה שмагיעה מהלכה היא מחרוזת טקסט, המחרוזת מגיעה בצורה מקודדת ויש צורך לעפענה את הקידוד על ידי שימוש בפונקציה מובנית של **python** ש יודעת להמיר טקסט מקודד בקידוד **utf8** למחרוזת **.string**.

הmethod ה **onMessage** נותנת מענה לכל ההודעות המגיעות מהלכה, אנו בחרנו להתייחס לשולש סוגי הודעות ולהתעלם מההודעות אחרות.

ההודעה הראשונה הנה הודעה מסוג מחרוזת בינהarity וכתגובה להודעה זו המתודה **MDPISHE** בשרת את הודעה שהתקבלת לצורך מעקב ובראה.

ההודעה השנייה מגיעה בפורמט טקסט מכילה שתי נקודות ציון נגדיות : **sw**, **se** המציינות את גבולות המפה שモוצגת בעת על מסך האפליקציה, המתודה קוראת לפונקציה **screenColor** שאליה היא ניגשת דרך אובייקט החיבור למסד נתונים ושולחת אליה את נקודות הציון שהתקבלו לאחר המרה של הטקסט לפורמט **lat,lng** ומחזירה רשימה של אובייקטים מסוג **json**.

את הרשימה של האובייקטים המתודה שולחת בחזרה לאפליקציה בעזרת method **sendMessage** שהגיעה בירושה מהמחלקה **WebSocketServerProtocol**.

ההודעה השלישית מגיעה כמחרוזת טקסט שמייצגת נקודה **lat, lng** המתודה קוראת לפונקציה **find_square** שאליה היא ניגשת דרך אובייקט החיבור למסד נתונים ושולחת אליה את נקודות הציון שהתקבלו לאחר המרה של הטקסט לפורמט **lat,lng** ומחזירה אמת באם הריבוע שבו נמצאת הנקודה שנשלחה נמצא באזורה המוגדר כمسוכן וSKU אחרה. המתודה שולחת לאפליקציה הודעה כאשר המתודה **find_square** מחזירה אמת.

שליחת הודעה לאפליקציה מתבצעת בעזרת method **sendMessage** שהגיעה בירושה מהמחלקה **WebSocketServerProtocol**.

הmethod השלישי הוא **onClose** מקבלת כפרמטרים את קוד סגירת החיבור, סיבת סגירת החיבור ומשתנה בולייאני שמצוין האם סגירת החיבור התרבצעה בצורה מסודרת על פי פרוטוקול "לחיצת היד" או מכשל כלשהו. המתודה מדפסה בשרת שישה בקשה סגירה ולצורך מעקב שולחת בחזרה לאפליקציה הודעה שהחיבור נסגר וסגורת את החיבור על ידי קריאה אוטומטית לפונקציית הסגירה במחלקה **97** **WebSocketServerProtocol**.

לאחר הגדרת המחלקה אשר קובעת את התנוגות השרות קיים צורך להפעיל את השרת ואנו עושים זאת על ידי קריאה מפונקציית **main**.

בפונקציית `main` אנו יוצרים אובייקט בשם `factory` אשר מתחבר למחלקה הבסיסית ממנה המחלקה `WebSocketServerFactory` המגדירה את ה프וטוקול החיבור יורשת `MyServerProtocol`. האובייקט מתחול את החיבור לשרת על ידי שליחת כתובת השירות (`url`) ו-`port` ולאחר מכן המכון הגדרת הפרוטוקול של אובייקט `factory` לפרטו של הגדרכנו כמפורט לעיל.

חלק מפונקציות האובייקט אנו מתחולות את מספר החיבורים המקורי ל-20 חיבורים (זאת מכיוון שהשרת הוא שרת שרצ על גבי סיבכה אינטרנטית שנועדה לפיתוח ובעתיד הקרוב ירוץ על גבי שרת חזק יותר ונוכל לשנות את מספר החיבורים המקורי בהתאם).

כמו כן הגדרכנו את מרוחת הזמן בין שליחת `pings` ל-5 שניות ואת זמן ההמתנה לגובה עד 30 שניות. כדי שהשרת ירוץ ללא הפסקה השתמשנו ב-`reactor`, שמתפרק כלולאה אשר מגיבה לירוצים בהתאם לסוג האירוע שהתרחש, הולאה רצה כל עוד לא היא לא נעצרה.

הגדרנו את ה-`reactor` להאזין לחיבור TCP על `port` מס' 8081 ברגע שmagiu אירוע `reactor` מתיכון אליו כ"תליך" ומגידיר לו פונקציות טיפול בהתאם לירוע, פונקציות הטיפול נלקחות מה-`factory` שבו הגדרכנו את התנאיות השרות `reactor` ממתיין לירוצים וזמן ההמתנה `reactor` רדום, ברגע שmagiu אירוע על `port` 8081 שלו בחרנו להאזין, האירוע נרשם לרשימה האירועים, ה-`reactor` מתעורר ומתחליל לטפל בירוצים שהגיעו.

ה-`reactor` בשרת מוגדר לרוץ תמיד.⁹⁸

אובייקט נוסף המוגדר בחלק המוביל מחלוקת זו הוא אובייקט התזמון המאפשר לבסיס הנתונים להתקען באמצעות תקשורת בין השירותים. על מנת שאובייקט זה יוכל לחברם בסיס הנתונים הנהנו לו פונקציה עוטפת במחלוקת `MyServerProtocol` בשם `update` אשר קוראת למетодה `getAcc` מתוך האובייקט מסווג `DBFunc`. ההרחבה על אובייקט תזמון משימות זה ואופן פעולתו תופיע בהמשך.

4.2 פיתוח האפליקציה

כאמור, פיתוח האפליקציה נעשה בשפת `java` על גבי סביבת העבודה `android studio`. במהלך פרויקט זה פיתוח האפליקציה נעשה כהמץ לפיתוח קיים. להלן רשימת הדרישות מאפליקציית הפרויקט:

- הקמת תקשורת יציבה עם שרת החישובים
- צביעת מפה ישראל בהתאם לנתונים המגיעים משרת החישובים
- הקפצת התראות (קוליות ודחיפה) בכניסה לאוצר מסוכן

4.2.1 תרשימים UML

להלן תרשימים UML המתאר את מחלוקותיו וקשריו העיקריים של צד הלוקה- האפליקציה.



איור 3-4 תרשימים uml צד לקוח

4.2.2 תקשורת websocket מהאפליקציה

במיושן צד הלקוח חשבנו בתחילת הפעלה לחשוף authbahn ספרייה של מנגנון לשימור על אוניברסיטת אולום ספרייה צד הלקוח בשפת java הייתה לא יציבה והיה קשה לשמור חיבור פתוח ויציב בין הרשת והלוקות. לבסוף בחרנו בספרייה java-websocket כיוון שהייתה יציבה יותר ועל כן אמינה יותר.

הספרייה מגדרה פונקציות אבסטרקטיות שיש למש על מנת שኒצור חיבור מסוג `websocket` עם השירות. מימושן של הפונקציות הנ"ל נמצא בפונקציה `connectWebSocket`.

במחלקה **MainActivity** ישנו אובייקט גלובלי `mWebSocketClient` מסוג `WebSocketClient` במתוך מימוש "לחיצת היד" בפתחת התקשרות מול השירות:

הפונקציה `connectWebSocket` מכילה את הפונקציות הנדרשות על מנת ליצור חיבור לשרת החישובים ולמש את לחיצת היד בהצלחה. הפונקציה יוצרת אובייקט `webSocketClient` מסוג `webSocket` ומגדירה לו את כתובות השירות על ידי זווע במקורה של כישלון בהגדרת הזוא הפונקציה מציגה הודעה שגיאיה.

במידה והפונקציה הצליחה ליצור קישור לכתובות זווע של השירות הפונקציה מגדרה את שלושת פונקציות לחיצת היד אשר מוחיב הפרוטוקול כמפורט לעיל.

פונקציה פנימית `onOpen` שמקבלת כפרמטר מידע על שינוי הפרוטוקול שהגיע מהשרת, למשל מידע מחבילת `http` המשודרגת. הפונקציה שולחת הודעה לשרת על הלקוח שהתחבר ולאחר מכן ישנה הפעלה אוטומטית של פונקציית חיבור בספריה.

הפונקציה השנייה הינה `onMessage` שמקבלת כפרמטר אובייקט מסוג `string` שמכיל את ההודעות שmagiuות מהשרת, הפונקציה מפעילה `Runnable` מסוג `UiThread` על מנת להציג נתונים על התצוגה על סמך מידע שmagiu מהתהילך שאינו מסוג `UiThread` אשר פועל ברקע.

בחרנו להשתמש `Runnable` מכמה סיבות:

1. אי אפשר לרשות מכמה מחלקות ולכון לא יכולנו לרשות מהמחלקה `Thread`.
2. מימוש של `Runnable` הוא מימוש של `interface` ולא מוסף הדבר פונקציונליות מסוובכת.
3. ישנה אפשרות להשתמש שוב ב-`Runnable` שסיים בשונה מהתהילך שלא ניתן לעשות בו שימוש נוספת **מוסך מרצע שסויים**.
4. ישנה אפשרות להשתמש `Runnable` על תהליך נפרד או על תהליך שנעשה בו שימוש בכל מקרה לפונקציה ישנו משתנה בוליאני המאותחל ל'שקר', במידה והתבצעה כניסה לאזרור מסוכן בפעם הראשונה, הפונקציה מת裏עה בצורה קולית על ידי המחלקה `TextToSpeech` אשר מתרגמות טקסט להודעה קולית וכן מציגה הודעה נוטיפיקציה ומשנה את ערך המשתנה הבוליאני לערך `True` לאחר מכן הפונקציה מקבלת עדכונים מהשרת, במידה והלקוח עובר בין אזרור מסוכן לשנהו לא תתקבל התראה נוספת, רק כאשר הלקוח עבר לאזור בטוח ערך המשתנה הבוליאני يتעדכן לערך 'אמת' ותתקבל התראה קולית על כן. במידה והלקוח עבר מאזור בטוח לאזרור מסוכן תתקבל התראה מתאימה.

לאחר מכן, הפונקציה מקבלת מהשרת רשימה של אובייקטים מסוג `JSON` ושולחת את האובייקטים אחד אחר אחד לפונקציית הצבעה `getColor`, בנוסף הפונקציה קוראת לפונקציה `clearMap` על מנת למחוק צבעה קודמת.

הפונקציה `onClose` מקבלת כפרמטרים קוד סגירה וסיבת הסגירה, הפונקציה שולחת הודעה לשרת על סגירת החיבור וסיבת הסגירה וסגורת את החיבור לאחר מכן ישנה הפעלה אוטומטית של פונקציית סגירה

בספריה, הfonקציה מנקה את המפה באמצעות הfonקציה clearMap לקרה שהחיבור ייסגר על ידי השרת.

הfonקציה onError - הfonקציה מציגת הודעה שגיאת במידה והיתה שגיאת באחת מהfonקציות הקודמות על גבי מסך log.

לאחר מימוש פונקציות לחיצת היד אנו מחברים את האובייקט m WebSocketClient דרך הfonקציה connect של הספרייה.

4.2.3 שליחת הודעות לשרת

מהאפליקציה נשלחות שתי סוגי הודעות לשרת. סוג ראשון : מידע על מיקומו הנוכחי מהfonקציה onLocationChangeLisnter אשר עוקבת אחרי מיקומו הנוכחי של המשתמש, ההודעה מתחילה במילה location. לפני שליחת ההודעה, נערך בדיקה האם החיבור מול השירות עדיין תקין ואם לא, הfonקציה מנסה ליצור חיבור מחדש לשרת.

הודעה שנייה שנשלחת לשרת היא גבולות המסך, כאשר גבולות המסך מיוצגים על ידי נקודה צפונית מזרחית והנקודה הדרומית מערבית, ההודעה מתחילה במילה lat/long. גם כאן, לפני שליחת ההודעה מתרכשת בדיקת החיבור לשרת ויש ניסיון להתחברות מחדש במידה הצורך.

בנוספ', במקרה שMegיעה תשובה מהשרת שאינה מותאמת לתשובה הצפוייה מסוגי הודעות אלו בלבד, האפליקציה מתעלמת.

4.2.4 צביעת המפה

המפה המוצגת על גבי מסך האפליקציה היא מפה googleMaps עם API V2. צביעת המפה נעשית על ידי הfonקציה getcolor, הfonקציה מקבלת צבע ונקודה lat,lng, הfonקציה ממירה את הצבע אשר מיוצג בשיטת HSL לצבע argb כאשר d מייצג את רמת השקיפות של הצבע והוא קבוע לערך 80. ערכי S ו L קבועים והם שווים ל 0.5 בהתאם. הfonקציה מוסיפה למפה מצולעים בצורה ריבוע הנבנה מהנקודה שהתקבל מהשרת אשר מהוות הנקודה השמאלית התח桐ונה של הריבוע.

הfonקציה ה"נגידית" לפונקציית הצביעה היא הfonקציה clearMap קוראת לפונקציה המובנית של googleMaps המשנה את מיקום המשתמש למקומו הנוכחי על ידי קריאה לפונקציה onLocationChanged. כמו כן, הfonקציה מסירה את כל הסמלים שהוצגו על גבי המפה. כך, אנו מונעים מצב בו המפה תהיה עמוסה בסמלים אשר אין לה צורך בהם (הם נמצאים על המפה מעבר לגבולות המסך המשמש) ונמנע מצב של ריבוע הנציג מספר פעמים (על ידי יציאה וחזרה לאותו אזור).

4.2.5 הקפצת התראות בקרבת אזור מסוכן

במסגרת ההתראות בחרנו למשתמש שתי אפשרויות שבעתיד הלקוח יוכל לבחור האם להשתמש באחת מהן או בשתיهن.

סוג ראשון של התראות - notification :
ההתראות נבנות בפונקציה **addNotification**, הפונקציה בונה את מבנה הודעה וכן את סמל הודעה, כוורת ההתראה ותוכן הודעה.

לאחר מכן הפונקציה מקשרת בין העדכון הראשי למבנה הודעה כאמור לעיל, הפונקציה בונה את הודעה בעזרת מנהל ההתראות.

הפונקציה נקראת מתוך websocket onMessage של רק כאשר השרת מחזיר הודעה שאכן מדובר באזור מסוכן.

סוג שני - התראות קוליות, TTS :
על מנת להפעיל את ההתראות הקוליות יש צורך באתחול ובצינר onCreate אשר ממומשים בפונקציה onCreate של העדכון, בפונקציית האתחול - onInit ישנה בדיקה האם המנווע זמין, במידה וכן הפונקציה מאתחלת את שפת הדיבור של המנווע לאנגלית. ישנה אופציה לבחור שפות נוספות, כמו עברית היא לא חלק מהם ללא הורדת נוספת נוספת, בחרנו באנגלית שהנה שפה בינלאומית מוכרת. במידה והשפה מוכרת למנוע תתקבל הודעה Hello בכניסה לאפליקציה, במידה והשפה אינה מוכרת מקבל הודעה שגיאה במסך הוגן ולא תשמע התראה.

על מנת להשתמש הגדכנו את הפונקציה speak המקבלת מחרוזות וממירה אותה לפט קולי במידה ויש ריבוי הודעות, הודעה נוספת נכנסת לתור ונקראות אחת לאחר השניה לפי סדר הכניסה.

הפונקציה speak נקראת בשני מצבים :
1. מהפונקציה onMessage כאשר נכנסים לראשונה לאזור מסוכן ומתקבל טקסט - 'be carefull' שהוא

היא ממירה לפט קולי שנשמע באפליקציה בבירור.

2. בפונקציה onMessage כאשר המקום משתנה ועובר לאזור בטוח, הפונקציה speak מקבלת כקלט מחרוזת 'out of danger' שהיא היא ממירה לפט קולי שנשמע באפליקציה בבירור.
בפונקציה onDestory אנו עצורות את מנוע הדיבור וכן מכבות אותו.

4.2.6 איבוד חבילות -

נפרד לשני מקרים :

מקרה ראשון, השרת מאבד חבילה שנשלחה מהלקוח :

נפרד מקרה זה על פי סוגי החבילה :

סוג ראשון - הלקוח לא הגיב להודעת Ping אשר נשלחה מהשרת, במקרה כזו השרת ממתין פרק זמן של 30 שניות במהלךו נשלחות 6 הודעות pong רק במידה והלקוח לא ענה בהודעת pong לאף הודעה במהלך הזמן הניל החיבור ביניהם יתנתק והודעת סגירה תשלח ללקוח.

סוג שני - חבילת מידע שנשלחת מהלקוח מתוך הפונקציה `onLocationListner` - Ciyouo שקצת שליחת ההודעות עברו שינוי מיקום מתמשך בתדרות כה גבוהות ייתכן והלקוח בכלל לא יבחן באטיות התקשרות.

סוג שלישי- חבילת מידע שנשלחת מהלקוח מתוך הפונקציה `onCameraChange` כתוצאה מהזאת המשך על ידי המשמש לשרת. במידה וההודעה הולכת לאיבוד המשך לא יצבע ולכון אלו מאמיניות כי הלקוח ייזו את המשך שוב במטרה לקבל תגובה. שליחת הודעה שנייה זו על שינוי גבולות מסך נמצאת בסיכויים גבוהים להגיע לשרת אם התקשרות תקינה והאיבוד הקודם היה חד פעמי וחרג. כך יקבל הלקוח שוב את המידע שביקש (עם שינוי מזערי).

מקרה שני, הלקוח מאבד חבילה שנשלחה מהשרת:

נפרד לשני סוגי הודעות:

הסוג הראשון, הודעה `ping` שנאבה: כל עוד מסגרת הזמן שהקבעו לסגירת החיבור (30 שניות) לא עברה, החיבור ישאר פתוח ויתקבלו 5 הודעות Ping נוספת מהשרת שעליהן הלקוח יכול לענות.

סוג שני: הודעה תגובה על מידע שהועבר מהלקוח לשרת. גם את סוג הודעה זה נפרד לשני סוגים:

א. הודעה שהגיעה בתגובה לשינוי המיקום של הלקוח מתוך הפונקציה `onChangeListner`, קצב ההודעות שנשלחות הן מהיר ולכון ייתכו בהחלטה שהלקוח לא יבחן שאבדה חבילה ואיובודה לא יפריע לחווית השימוש שלו.

ב. הודעה שהגיעה בתגובה לכך שמסך האפליקציה השתנה מתוך הפונקציה `onCamerachange`, במקרה של חוסר תגובה מהשרת אלו מאמיניות שהלקוח ייזו בשנית את מסך האפליקציה ויקבל תגובה.

4.3 פיתוח תקשורת בין שרתים

חלק זה של הפיתוח אחראי על עדכון בסיס הנתונים מול שרת `anyway`. עדכון הנתונים נעשה באמצעות התקשרות בין שרת החישובים לשרת `anyway` לצורך קבלת עדכנים ומחר ובבסיס הנתונים גדול מאוד וקיים לנו חלוקה פחות או יותר שוויונית שלו אשר נעשתה בזמן האתחול החלנו בכל יום להריץ משימה אשר תבודוק עדכנים עבור עשירית מבוסיס הנתונים כך שעדכון בסיס הנתונים מושלים במלואו פעם בעשרה ימים.

כפי שנכתב לעיל, אובייקט `Tzomun` המשמשות, מוגדר בתוך חלק `main` המריץ את המחלקה `MyServerProtocol`. לאובייקט זה הוספנו את הפונקציה `update` הנקרה מתוך אובייקט `factory` פעם ביום, בשעות הערב(00:00) על מנת שלא ליזור עומס על השירותים.

הפונקציה `update` משתמשת במשתנה סטטי אשר מתאפס כל עשרה ימים וMOVVER בקריאה לפונקציית `getAcc` השויכת למופע המחלקה `DBFunc`.

בהתאם לחלוקת שנעשתה בשרת הפונקציה `getAcc` שולחת בקשות לשרת `anyway`, על מנת לעדכן את החלק המתאים מבסיס הנתונים. הפונקציה `getAcc` משתמשת בספריית `urllib2` דרך היא שולחת לשרת `anyway` את נקודות הציון של הריבוע עליו היא מעוניינת לקבל מידע, אשר התרחש בין התאריך האחרון בו היא ערכה עדכון לריבוע והຕאריך הנוכחי. במס הthesובה שהתקבלה (בפורמט JSON) מכילה תאונה חדשה הפונקציה תכנס את התאונה למערך התאונות בהתאם לאלוצי האלגוריתם ובמידת הצורך, תקרא לפונקציית חישוב הסיכון, תקבל את ערך ההסתברות החדש ותעדכן את מערך הריבוע והציב המצויין את הסיכון שלו בבסיס הנתונים.

5. דיוון וסיכום

5.1 דיוון

5.1.1 הרכבת עבודה

העבודה על הפרויקט נעשתה בהנחיה צמודה של המנהה האקדמי : נפגשנו עמו מידי שבוע. בפגישה סיכנו את הנעשה בשבוע החולף והגדכנו מושימות לשבוע לאחר מכן.

במסגרת הפרויקט נפגשנו עם המנהה מטעם החברה כשלה צורך ולא בזמנים קבועים, חשוב לציין שהמנהה מטעם החברה היה זמין בצורה יוצאת דופן וענה במהירות לכל השאלות שלנו גם בשעות לא שגרתיות.

עבור נושאים שחרגו מהבנתנו בתחום הסטטיסטיקה ו**big data** פגשנו מומחים. לכל אורך הפרויקט עבדנו ביחד ביחס לצוות והעבודה הייתה נעה ופורייה.

5.1.2 נקודות חזק של האפליקציה

לאפליקציה אותה פיתחנו ישן מספר נקודות חזק :

- **עמידה בעקרון Open Closed Principle**- לפי עקרון זה, רכיבי המערכת צריכים להיות פתוחים להרחבה ועם זאת לא לאפשר שינויים בלבד הרחבות. אם נרצה להרחיב רכיב מסוים ולהוסיף לו פונקציונליות נוספת, יוכל לעשות זאת ללא שינויים בקוד המקור. שמירה על עיקרון זה מבטיחה פיתוח יעיל הלוקח בחשבון הרחבות עתידיות. אפליקציית `anyway` שומרת על עיקרון זה, ומtówך תכנון לפיתוחים עתידיים, כל הוספה פונקציונלית חדשה לא תדרוש שינוי בקוד הקיים.

- עמידה בעקרון **Interface Segregation Principle**- לפי עקרון זה הממשק מחולק לחלקים קטנים, באופן שינויים יוכלו להתבצע בקבוצות בלי להתעסק עם קוד ארוך שברובו הוא אינו רלוונטי לערכיה. האפליקציה שפיתחנו שומרת על עקרון זה, ולא מחייבת את המפתח שירצה להרחיב אותה או לעורך בה שינויים לקרה את כל הקוד ולשלוט בו, וכך הוא יוכל לעורך את השינויים בקוד בקלות. לדוגמה : שינוי בעיצוב מסכי האפליקציה לא דרש הבנה ובקיאות בפונקציונליות.

2. פיתוחים עתידיים

2.1. צמצום בסיס נתונים

פיתוח עתידי אפשרי לאפליקציה זו יכול להיות צמצום בסיס הנתונים לבסיס נתונים שיכיל אך ורק ריבועים אשר יש בתוכם חלק מככיבש או דרך כלשהו. באופן כזה לא רק שבסיס הנתונים יקטן משמעותית אלא גם הזמן התגובה והתקשרות יכולים להשתפר משמעותית ולהשאיר את מתכנתה האפליקציה ממוקדים יותר בריבועים ובמקומות בהם חלו תאונות דרכיהם.

2.2. הוספת מסד נתונים סביבתי

הוספת מסד נתונים המכיל מידע סביבתי כמו אזורי תעשייה, פארקים, גני ילדים וכדומה ומיזוגו עם בסיס הנתונים של תאונות הדרכים באופן המפורט במודל המובה במאמר חישוב הסיכון比亚 לתוכאות סיכון מדויקות ומשופרות.

2.3. שיפור האפליקציה הקודמת

שיפור התקשרות שנעשתה על ידי המפתח הקודם ועובדת על `volley` ו-`http` כפי שפורט לעיל ומאתה את האפליקציה

2.4. צביעת הכבישים בלבד

פיתוח עתידי משמעותי בו ישנו צורך ליצור מסד נתונים כלשהו שמכיל נתונים על כבישי ישראל בלבד ולצבע רק את קטעי הכביש לפי רמת סיכון.

פיתוח שכזה יספר את נראות האפליקציה וכן יבהיר היכן נועצת הסכנה באופן ספציפי יותר.

2.5. שילוב עם האתר האינטראקטיבי של חברת `anyway`

צירוף המודל החישובי וצבעת המפה על גבי האתר האינטראקטיבי ולאחר מכן למשתמשים לראות את הסיכון על גבי המפה הצבעה ובכך לעורר מודעות לסכנה.

2.6. חישוב סיכון על מסלול

בהתנתק נקודת התחלה ונקודת סיום האפליקציה תוכל להציג מספר מסלולים ומהמשמש יוכל לבחור את המסלול שבו הסיכון קטן יותר ביחס למסלולים אחרים.

7.2.7 שילוב האפליקציה בענקית waze

שילוב מידע של החלק המחקרי של הפרויקט שלנו עם waze על מנת שימוש באפליקציית waze יוכל לקבל מידע על הסיכון במסלול שנבחר.

8.2.8 עדכון השירותים

כיום, צד השרת ממומש בשפת `python` בגרסת 2.7, בעוד שאר העולם התקדם לגרסהות מתקדמות יותר וכן בಗל שישן ספריות מתקדמות אשר לא עובדות עם גרסאות ישנות של שפת `python` יש צורך לעדכן את השירותים לגרסהות מתקדמות יותר.

9.2.9 אבטחת התקשרות

בגרסה זו של האפליקציה לא חתיכנו לאבטחת המידע בחיבור `websocket` בין השירות ללקוח כיון שהמידע שעובר בין הצדדים אינו רגיש וכיימת אבטחה כלשהי באפליקציית המובייל,➥ בעtid כדי לבצע בדיקת אבטחה על ידי שליחת שדה `origin` בין הלקוח והשרת.

5.3 שיווק

בימים הקרובים לאחר עדכון מסד הנתונים בנתונים חדשים משנת 2016 האפליקציה תועלה לעדכון Google Play Store לאפליקציה הקיימת בـ

5.4 סיכום

מחוקרים וכתבות רבות שנעשו בנושא תאונות הדרכים בישראל מציגים תמונות מצב עוגמה בעקבות כך עולה הצורך בפתרון הבעיה.

פרויקט `anyway` שם לו למטרה להלחם בתופעת תאונות הדרכים על ידי הגברת המודעות לתאונות הדרכים וכן בניסיון להפעיל לחץ ציבורי על הרשויות על מנת שישופרו התשתיות ברחבי הארץ.

מטרת הפיתוח בـ`android` הייתה הנגשת המידע ולהגיע לחשיפה גדולה יותר שכן כיום סマרטפון הננו מוצר צריכה בסיסי ומספר הטלפונים במדינה עולה על מספר האנשים.

במסגרת הפרויקט פותח מודל חישובי על מנת לסוג את רמת המסוכנות של הכבישים בארץ וצביית המפה על מנת להציג את הנתונים بصورة ויזואלית ברורה.

מחוקרים שקרו לנו אנו ופורטו לעיל הסקנו כי התראת יוצרת הרתעה ולכן בכניסה לאזוריים ברמת סיכון גבוהה כפי שהוגדר במודל החישובי מוצגת הודעה אזהרה וכן מושמעת הודעה קולית.

נחשפנו לפרויקט `anyway` בעקבות כתבה שפורסמה בתקורת וכיוון שהנושא נגע לבנו יצרנו קשר עם מפתחי הפרויקט, כיון שהפרויקט מבוסס על התנדבות חשבנו על הפיתוח הנ"ל בעצמנו.

הकשיים שהיו לנו בתחילת הדרכן היו רבים, החל משלב הגדרת הפרויקט, כיוון שהפרויקט לא הוגדר לנו על ידי גורם חיצוני וננתנו לנו חופש רב בקבלת החלטות שאפנו גבוהה ואנו מרצות מאוד מה מוצר שפיתחנו.

קושי נוסף היה החלק המחקרי משום שלא היה לנו רקע מתמטי עמוק בסטטיסטיקה בילינו שעוטה רבות בלימוד מונחים, ניתוח נתונים פגישות עם מומחים לסטטיסטיקה וקריאה מאמרים רבים על מנת להוכיח את הצורך בהתראה וכן על מנת לבנות מודל חישובי.

במסגרת הקמת השירות היה علينا להתמודד עם מסד נתונים גדול, בניית מסד הנתונים הנוכחי לוותה בקשאים אשר התגלו בזמן אמת וגרמו לכך שהייתה علينا לחשב מסלול מחדש, גם כאן נעזרנו במומחים ל *big data* שבuzzards בחרנו מסד נתונים מהיר יותר אשר מותאם יותר לצורכינו.

בחלק המעשי היה علينا להביא לידי ביטוי את כל המידע הרחב שלנו בשלב המחקר, למשם מודל חישובי, לחלק את מפת ישראל לריבועים שווים בגודלים, לנחל ותקורת חכמה בין השירות ללקוח שבה זמן התגובה בין הלקוח לשירות היה מרכיב קריטי ולצבעו ולעכן את המפה ביחס למיקום הנוכחי של המשתמש.

התוצר הסופי של עבודתנו הוא אפליקציה שעובדת בתקשורת על גבי פרוטוקול *websocket* מול השירות אשר מחשב את רמות הסיכון ביחס למיקום הנוכחי של המשתמש וגבולות המסך, האפליקציה בתגובה צובעת את המסך על פי חלוקה לריבועים אשר נתונים עליהם מגיעים מהשירות, האפליקציה שולחת התראות קוליות והתראות דחיפה למשתמש כאשר המשתמש נכנס לאזור המוגדר כمسוכן.

על ידי הגדרת הסיכון והציגתו בצורה ויזואלית הנגשנו את המידע ואנו תקווה שעל ידי כך מספר תאונות הדרכים בארץ יפחית ובאופן ישיר יופחת מספר הנפגעים.

ולו חי אדם אחד יינצלו בזכותו הפיתוח שלנו נראה זאת כשכרנו מן השמיים.

6. רשימת מאמרים

להלן רשימת המאמרים עליהם התבססו במהלך פיתוח המודל הסטטיסטי :

Koster, M.B.M. de, Balk, B.M., Davelaar, I. & Martens, M. Accidents will happen. Do safety systems improve warehouse safety performance?, Progress in Handling Research: 2010 (pp. 49-60), Netherlands, 2010.

Philip S. Carroll. classification of driving exposure and accident rates for highway safety analysis, Accid. Anal. & Prec. Col. 5, pp. 81-94. Pergamon Press, 1973.

Hauer Ezra, On exposure and accident rate, Traffic Engineering and Control 36(3): 134-138 ,1995.

Kwok-suen Ng , Wing-tat Hung, Wing-gun Wong, An algorithm for assessing the risk of traffic accident, Journal of Safety Research 33(3): 387-410, 2002.

7.ביבליוגרפיה

1. <http://www.rsa.gov.il/meidamechkar/pirsumim/documents/nekudattazpit9.pdf>
2. http://www.who.int/features/factfiles/roadsafety/02_en.html
3. https://he.wikipedia.org/wiki/%D7%AA%D7%90%D7%95%D7%A0%D7%AA_%D7%93%D7%A8%D7%9B%D7%99%D7%9D
4. <http://www.oryarok.org.il/?p=1226>
5. http://www.who.int/features/factfiles/roadsafety/03_en.html
6. <http://mida.org.il/2016/02/17/%D7%94%D7%9E%D7%A6%D7%91-%D7%94%D7%90%D7%9E%D7%99%D7%AA%D7%99-%D7%A9%D7%9C-%D7%AA%D7%90%D7%95%D7%A0%D7%95%D7%AA-%D7%94%D7%93%D7%A8%D7%9B%D7%99%D7%9D-%D7%91%D7%99%D7%A9%D7%A8%D7%90%D7%9C/>
7. <http://news.walla.co.il/item/2622809>
8. <http://www.mako.co.il/cars-cars-news/safety/Article-72d89ccb78fca31006.htm>
9. <http://www.themarker.com/dynamo/cars/1.2279316>
10. <http://teunot.cbs.gov.il/teunotm/>
11. http://www.cbs.gov.il/publications14/acci13_1572/pdf/map_h.pdf
12. <http://www.oryarokmap.co.il/>
13. <http://metrocasm.com/10-years-of-traffic-accidents-mapped.html>
14. <http://www.mapsdata.co.uk/mapsdataapp/#>
15. <http://roadskillmap.com/>
16. https://en.wikipedia.org/wiki/Geographic_coordinate_system
17. https://he.wikipedia.org/wiki/%D7%94%D7%99%D7%98%D7%9C_%D7%9E%D7%A8%D7%A7%D7%98%D7%95%D7%A8
18. https://en.wikipedia.org/wiki/GRS_80
19. https://he.wikipedia.org/wiki/%D7%A8%D7%90%D7%9C_%D7%94%D7%97%D7%93%D7%A9%D7%94
20. <http://tx.technion.ac.il/~zvikabh/software/ITM/isr84lib.cpp>
21. https://he.wikipedia.org/wiki/%D7%99%D7%91%D7%A2_%D7%91%D7%9E%D7%97%D7%A9%D7%91
22. https://en.wikipedia.org/wiki/RGBA_color_space
23. <http://geospatialpython.com/2011/01/point-in-polygon.html>
24. https://en.wikipedia.org/wiki/Jordan_curve_theorem
25. https://en.wikipedia.org/wiki/Point_in_polygon
26. <https://docs.python.org/2/library/sqlite3.html>
27. <https://he.wikipedia.org/wiki/SQLite>
28. <https://www.sqlite.org/datatype3.html>
29. <https://he.wikipedia.org/wiki/CSV>
30. <https://www.statista.com/statistics/263453/global-market-share-held-by-smartphone-operating-systems/>
31. <https://he.wikipedia.org/wiki/%D7%90%D7%A0%D7%93%D7%A8%D7%95%D7%90>

- %D7%99%D7%93_(%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%
A4%D7%A2%D7%9C%D7%94)
- 32. [https://he.wikipedia.org/wiki/%D7%90%D7%A0%D7%93%D7%A8%D7%95%D7%90%D7%99%D7%93_\(%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94\)](https://he.wikipedia.org/wiki/%D7%90%D7%A0%D7%93%D7%A8%D7%95%D7%90%D7%99%D7%93_(%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94))
 - 33. <http://www.androidcentral.com/google-says-there-are-now-14-billion-active-android-devices-worldwide>
 - 34. <http://qz.com/472767/there-are-now-more-than-24000-different-android-devices/>
 - 35. <https://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play/>
 - 36. https://en.wikipedia.org/wiki/Google_Play_Services
 - 37. https://en.wikipedia.org/wiki/Speech_synthesis#Android
 - 38. <https://he.wikipedia.org/wiki/PHP>
 - 39. <https://he.wikipedia.org/wiki/JavaScript>
 - 40. <http://groups.engin.umd.umich.edu/CIS/course.des/cis400/python/python.html>
 - 41. <https://he.wikipedia.org/wiki/%D7%A4%D7%99%D7%99%D7%AA%D7%95%D7%9F>
 - 42. <http://vlib.eitan.ac.il/python/general/preface/whatispython.htm>
 - 43. <http://www.network-theory.co.uk/docs/pytut/CompiledPythonfiles.html>
 - 44. <http://stackoverflow.com/questions/2709821/what-is-the-purpose-of-self>
 - 45. <https://wiki.python.org/moin/DatabaseInterfaces>
 - 46. <http://pythoncentral.io/introduction-to-sqlite-in-python/>
 - 47. <http://apscheduler.readthedocs.io/en/latest/>
 - 48. <https://pymotw.com/2/urllib2/>
 - 49. <https://www.quora.com/What-are-the-benefits-of-Cloud>
 - 50. <https://c9.io/?redirect=0>
 - 51. <https://c9.io/site/about>
 - 52. [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))
 - 53. <https://www.infoq.com/news/2014/07/cloud>
 - 54. <https://en.wikipedia.org/wiki/OAuth>
 - 55. https://en.wikipedia.org/wiki/Google_Compute_Engine
 - 56. <http://tekeye.biz/2014/list-of-android-app-development-ides>
 - 57. https://en.wikipedia.org/wiki/Android_software_development
 - 58. https://en.wikipedia.org/wiki/Android_Studio
 - 59. https://en.wikipedia.org/wiki/Domain-specific_language
 - 60. <https://en.wikipedia.org/wiki/Gradle>
 - 61. <http://stackoverflow.com/questions/16754643/what-is-gradle-in-android-studio>
 - 62. https://en.wikipedia.org/wiki/Code_refactoring
 - 63. <https://developer.android.com/studio/features.html>
 - 64. [https://en.wikipedia.org/wiki/ProGuard_\(software\)](https://en.wikipedia.org/wiki/ProGuard_(software))
 - 65. [https://en.wikipedia.org/wiki/Lint_\(software\)](https://en.wikipedia.org/wiki/Lint_(software))
 - 66. <https://developer.android.com/studio/features.html>

67. https://en.wikipedia.org/wiki/Android_Studio
68. https://en.wikipedia.org/wiki/Android_Studio
69. <https://developer.android.com/studio/features.html>
70. <https://www.sitepoint.com/improved-android-emulation-genymotion/>
71. <https://developer.android.com/reference/android/os/AsyncTask.html>
72. <https://developer.android.com/training/volley/index.html>
73. <https://blog.jayway.com/2015/03/04/pros-and-cons-of-some-android-frameworks/>
74. <http://www.truiton.com/2015/02/android-volley-vs-asynctask-better-approach/>
75. <http://icode.co.il/%D7%A4%D7%A8%D7%95%D7%98%D7%95%D7%A7%D7%95%D7%9C-http-%D7%9E%D7%93%D7%A8%D7%99%D7%9A-%D7%9C%D7%9E%D7%AA%D7%97%D7%99%D7%9C%D7%99%D7%9D/>
76. [https://he.wikipedia.org/wiki/%D7%9E%D7%A6%D7%91_\(%D7%9E%D7%93%D7%A2%D7%99_%D7%94%D7%9E%D7%97%D7%A9%D7%91\)](https://he.wikipedia.org/wiki/%D7%9E%D7%A6%D7%91_(%D7%9E%D7%93%D7%A2%D7%99_%D7%94%D7%9E%D7%97%D7%A9%D7%91))
77. https://he.wikipedia.org/wiki/Hypertext_Transfer_Protocol
78. https://he.wikipedia.org/wiki/HTTP_Secure
79. <http://mqtt.org/>
80. [https://he.wikipedia.org/wiki/AJAX_\(%D7%AA%D7%9B%D7%A0%D7%95%D7%A](https://he.wikipedia.org/wiki/AJAX_(%D7%AA%D7%9B%D7%A0%D7%95%D7%A)
81. <http://netcraft.co.il/blog/?p=8954>
82. <https://he.wikipedia.org/wiki/REST>
83. <https://www.pubnub.com/blog/2015-01-05-websockets-vs-rest-api-understanding-the-difference/>
84. <http://www.digitalwhisper.co.il/files/Zines/0x41/DW65-3-WebSocket.pdf>
85. <http://stackoverflow.com/questions/28827349/proxy-blocking-websockets-how-to-get-round>
86. <https://internet-israel.com/%D7%9E%D7%93%D7%A8%D7%99%D7%9B%D7%99%D7%9D/html-5-%D7%9E%D7%93%D7%A8%D7%99%D7%9B%D7%99%D7%9D/web-sockets/>
87. https://media.blackhat.com/bh-us-12/Briefings/Shekyan/BH_US_12_Shekyan_Toukhan_Hacking_Websocket_Slides.pdf
88. <http://www.christian-schneider.net/CrossSiteWebSocketHijacking.html>
89. <https://he.wikipedia.org/wiki/XSS>
90. <http://resources.infosecinstitute.com/websocket-security-issues/>
91. <http://stackoverflow.com/questions/4862310/json-and-xml-comparison>
92. <https://he.wikipedia.org/wiki/XML>
93. <http://www.json.org/json-he.html>
94. <https://docs.python.org/3.4/library/asyncio.html>
95. <http://twistedmatrix.com/trac/>
96. http://autobahn.ws/python/_modules/autobahn/twisted/websocket.html#WrappingWebSocketAdapter.onClose
97. <http://autobahn.ws/python/websocket/programming.html>
98. <http://stackoverflow.com/questions/35111265/how-does-pythons-twisted-reactor-wor>

k

8. נספחים

8.1 צילומי מסך

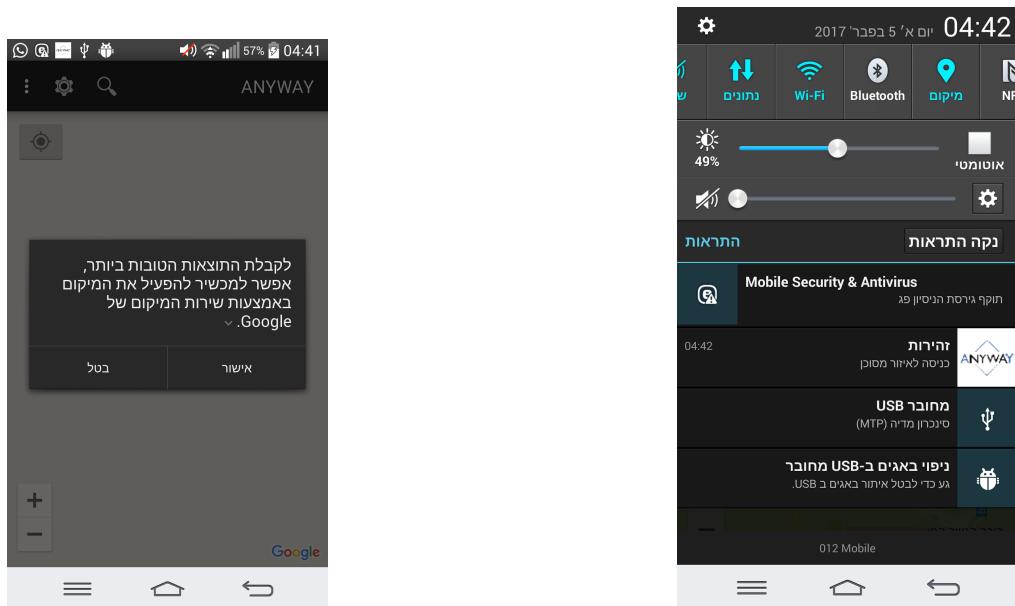
להלן נציג את צילומי המסך של האפליקציה. לאחר והפרויקט מהוות המשך פיתוח של אפליקציה קיימת בחרנו להתייחס ולהביא בדוחיה זה את צילומי המסך שאנו כתבנו תחילה ולאחר מכן להציג את צילומי המסך הקיימים של האפליקציה, על מנת שההקשר בין המ██דים לסדר הופעתם ומשמעותם יהיה ברור.

המסך הבוער כפי שהוא מופיע בכניסה לאפליקציה :



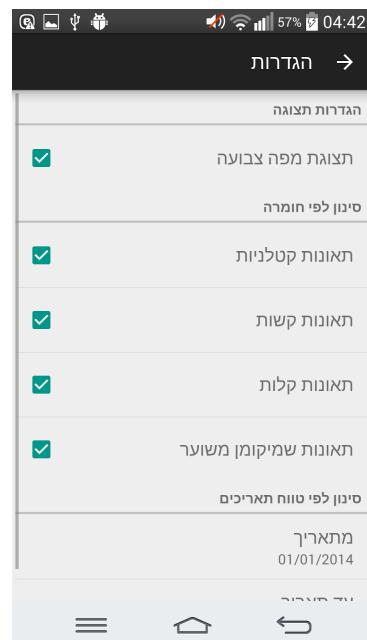
באם שירותים המוקום אינם פתוחים בטלפון החכם, האפליקציה תבקש לפתוח אותם על ידי הוצאת החתירה הבאה :

בצילום המשך הבא ניתן לראות את הנטויפיקציה אותה שולחת האפליקציה בכניסה המשמש לאזור

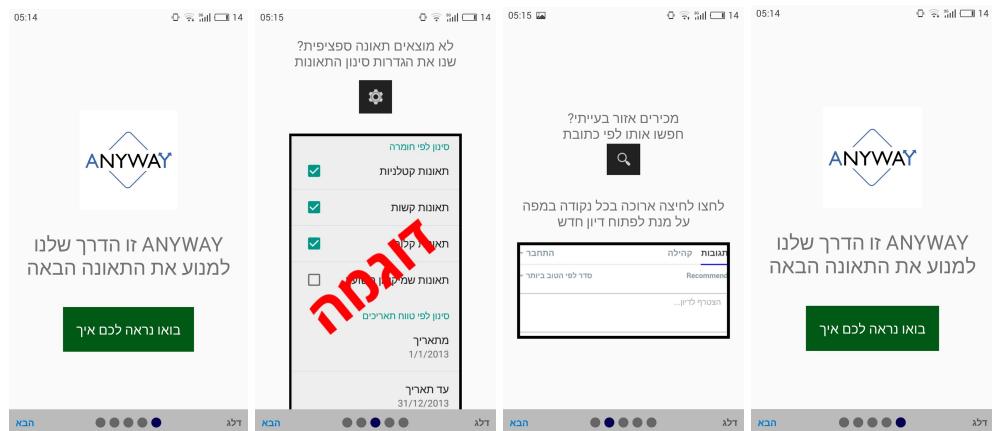


מסוכן :

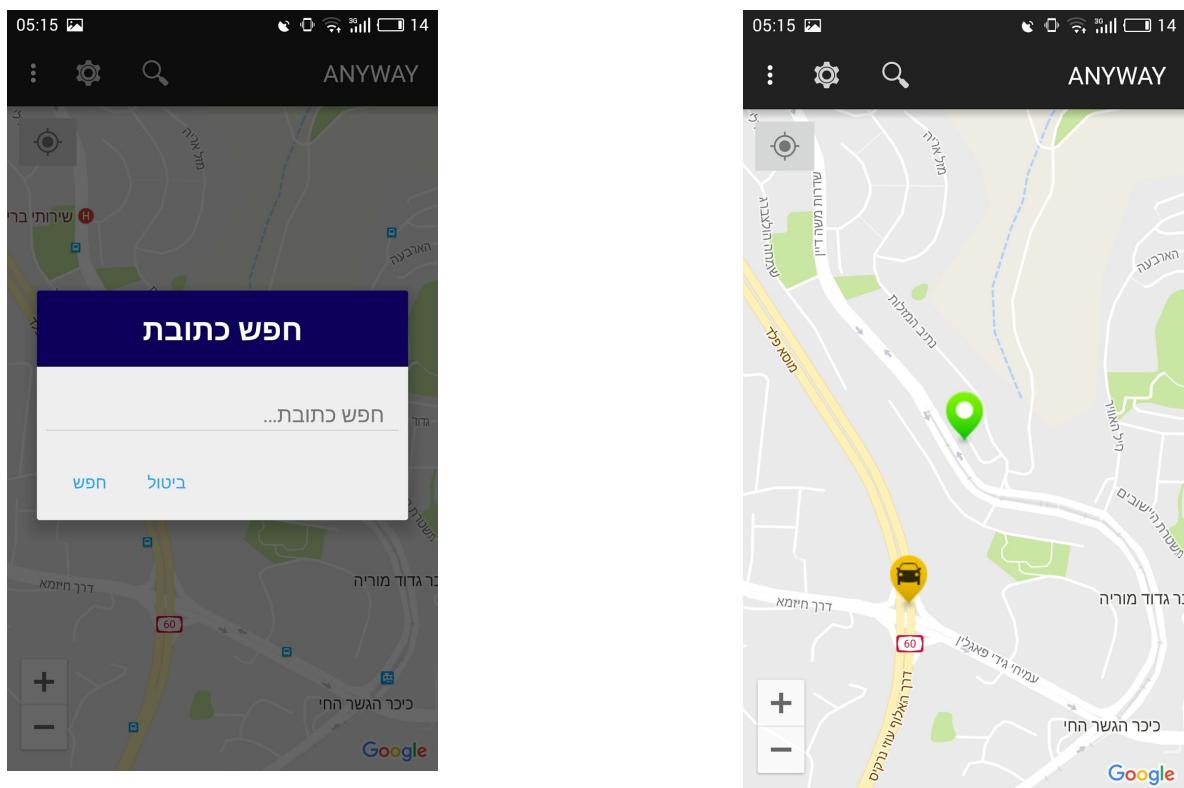
המשך ההגדירות של האפליקציה נראה כך והגישה אליו נועשית באמצעות לחיצה על גלגל השינויים הנמצא בחלקו העליון של מסך האפליקציה הראשי המציג את המפה :



שאר מסכי האפליקציה הקיימים נקראים כך :
בכניסה הראשונית לאפליקציה ישנו מסכי הסברים -



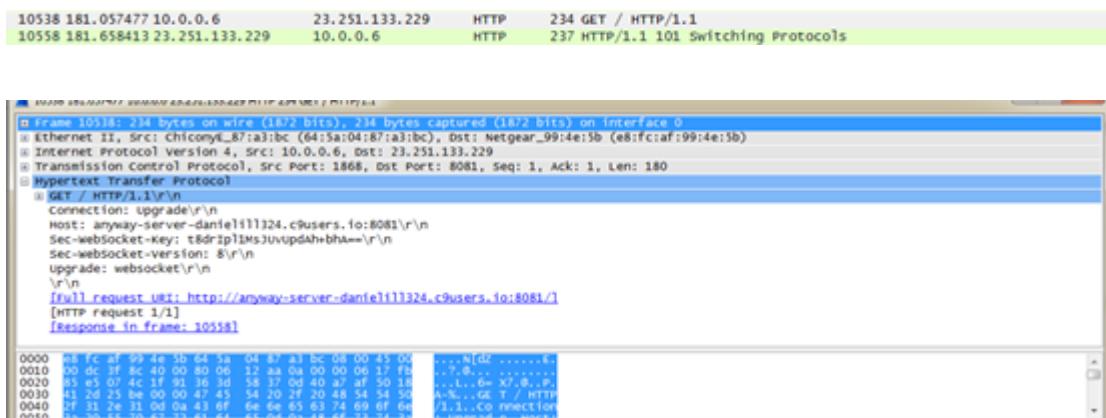
לאחר מכן ניתן לראות את המפה עם מקבצי תאותה דרכים ולהציג על סימן זכוכית המגדלת תוביל לחיפוש אחר כתובות מסיימת:



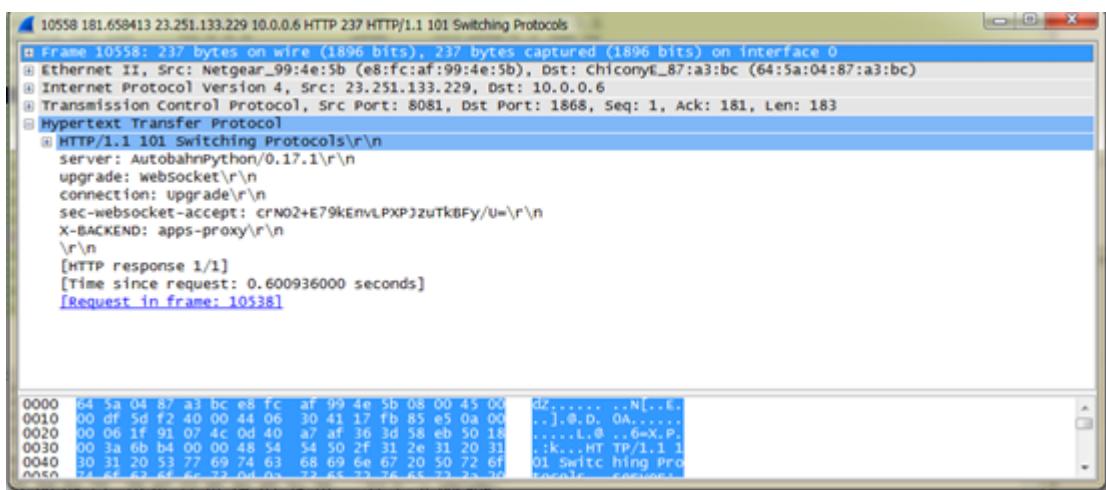
8.2 חבילות פרוטוקול websocket

צילומי המסך הבאים נלקחו מתוך תכנית הקלטת חבילות התקשורת wireshark והם מציגים את תהליך ההתחברות, התקשורת וההתקנתו של השירות והאפליקציה כיצה על גבי פרוטוקול websocket אותו מימושנו באמצעות הספריות המוצגות לעיל.

בתמונה הבאה ניתן לראות את שרת websocket (שרת החישובים) שכותבת IP שלו היא 23.251.133.229 יוצר את "לחיצת הידיים" עם האפליקציה שכותבתה היא 10.10.0.6 .



בתמונה הבאה מומוחש שלב ho101 שאריו השירות והאפליקציה מתקשרים ביניהם באמצעות פרוטוקול .websocket



```
10558 181.658413 23.251.133.229 10.0.0.6 HTTP/2.0 [HTTP/1.1 101 Switching Protocols]
Frame 10558: 237 bytes on wire (1896 bits), 237 bytes captured (1896 bits) on interface 0
Ethernet II, Src: Netgear_99:4e:5b (e8:fc:af:99:4e:5b), Dst: ChicomE_87:a3:bc (64:5a:04:87:a3:bc)
Internet Protocol Version 4, Src: 23.251.133.229, Dst: 10.0.0.6
Transmission Control Protocol, Src Port: 8081, Dst Port: 1868, Seq: 1, Ack: 181, Len: 183
Hypertext Transfer Protocol
HTTP/1.1 101 Switching Protocols\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 101 switching Protocols\r\n]
Request Version: HTTP/1.1
Status Code: 101
Response Phrase: Switching Protocols
server: AutobahnPython/0.17.1\r\n
upgrade: websocket\r\n
connection: upgrade\r\n
sec-websocket-accept: crNO2+E79kEnvLPXPJzuTkBFy/U=\r\n
X-BACKEND: apps-proxy\r\n
\r\n
[HTTP Response 1/1]
[Time since request: 0.600936000 seconds]
[Request in frame: 10538]
```

המשיכר על גביו רצה האפליקציה :

10562 18L736382 10.0.0.6 23.251.133.229 WebSocket 126 WebSocket.Text [FIN][MASKED]

- Frame 10562: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on Interface 0
- Ethernet II, Src: Chicony_E_87:a3:bc (64:5a:04:87:a3:bc), Dst: Netgear_99:4e:5b (e8:fc:af:99:4e:5b)
- Source: Chicony_E_87:a3:bc (64:5a:04:87:a3:bc)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.0.0.6, Dst: 23.251.133.229
- 0100 = Version: 4
- 0101 = Header Length: 20 bytes (5)
- Differentiated Services Field: 0x00 (DSSCP: CS0, ECN: NOT-ECT)
- Total Length: 112
- Identification: 0x3f97 (16279)
- Flags: 0x02 (Don't Fragment)
- Fragment offset: 0
- Time to live: 128
- Protocol: TCP (6)
- Header checksum: 0x130b [validation disabled]
- [Header checksum status: unverified]
- Source: 10.0.0.6
- Destination: 23.251.133.229
- [Source GeoIP: unknown]
- [Destination GeoIP: unknown]
- Transmission Control Protocol, Src Port: 1868, Dst Port: 8081, Seq: 181, Ack: 184, Len: 72
- Source Port: 1868
- Destination Port: 8081
- [Stream index: 146]
- [TCP Segment Len: 72]
- Sequence number: 181 (relative sequence number)
- [Next sequence number: 253 (relative sequence number)]
- Acknowledgment number: 184 (relative ack number)
- Header Length: 20 bytes
- Flags: 0x02B (PSH, ACK)
- Window size value: 36639
- [Calculated window size: 65536]
- [Window size scaling factor: 4]
- CHECKSUM: 0xb0af [unverified]
- [Checksum status: unverified]
- Urgent pointer: 0
- [SEQ/ACK analysis]
 - This is an ACK to the segment in frame: 10558
 - [The RTT to ACK the segment was: 0.077969000 seconds]
 - [RTT: 0.154577000 seconds]
 - [Bytes in flight: 72]
 - [Bytes sent since last PSH flag: 72]
 - [PDU size: 72]
- WebSocket
 - 1.... = Fin: True
 - .000 = Reserved: 0x0
 - 0000 = opcode: Text (1)
 - 1.... = Mask: True
 - .100 0010 = Payload Length: 66
 - Masking-Key: 31310fe6
 - Masked payload
 - Payload
- Line-based Text Data
 - Hello from Genymotion Google Nexus 5X - 6.0.0 - API 23 - 1080x1920

להלן צילום מסך המפרט את החבילה שנשלחה מהלכה לשרת עם הودעה מהלכה המצינית את גבולות המסך בקובוארדינטות גրפיות:

```

46459 858.827336 10.0.0.6 23.251.133.229 WebSocket Text [FIN] [MASKED]
Total Length: 141
Identification: 0x6f4c (28492)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to Live: 128
Protocol: TCP (6)
Header checksum: 0xe338 [validation disabled]
[Header checksum status: unverified]
Source: 10.0.0.6
Destination: 23.251.133.229
[Source GeoIP: Unknown]
[Destination GeoIP: unknown]
Transmission Control Protocol, Src Port: 1868, Dst Port: 8081, Seq: 1803, Ack: 125662, Len: 101
    Source Port: 1868
    Destination Port: 8081
    [Stream Index: 146]
    [TCP Segment Len: 101]
    Sequence number: 1803 (relative sequence number)
    [Next sequence number: 1904 (relative sequence number)]
    Acknowledgment number: 125662 (relative ack number)
    Header Length: 20 bytes
    Flags: 0x018 (PSH, ACK)
    Window size value: 16665
    [Calculated window size: 66660]
    [Window size scaling factor: 4]
    checksum: 0xc36a [unverified]
    [checksum status: unverified]
    urgent pointer: 0
    [SEQ/ACK analysis]:
        [RTT: 0.154577000 seconds]
        [Bytes in flight: 101]
        [Bytes sent since last PSH flag: 101]
        [POU size: 101]
    [WebSocket]:
        1... .... = Fin: True
        .0... .... = Reserved: 0x0
        ...001 = opcode: Text (1)
        1... .... = Mask: True
        .10 1111 = Payload length: 95
        Masking-Key: d2d8328d
        Masked payload
        Payload
    [Line-based text data]
        Tst/Lng: (31.769571704194544,35.01204438507557)
Frame (155 bytes) | Unmasked data (95 bytes)

```

השרת, כתגובה לשילוח הלקוח את מיקומו ואת גבולות המסך שולח ללקוח רשימה של הריבועים לעליו
לצבע במסך בפורמט JSON :

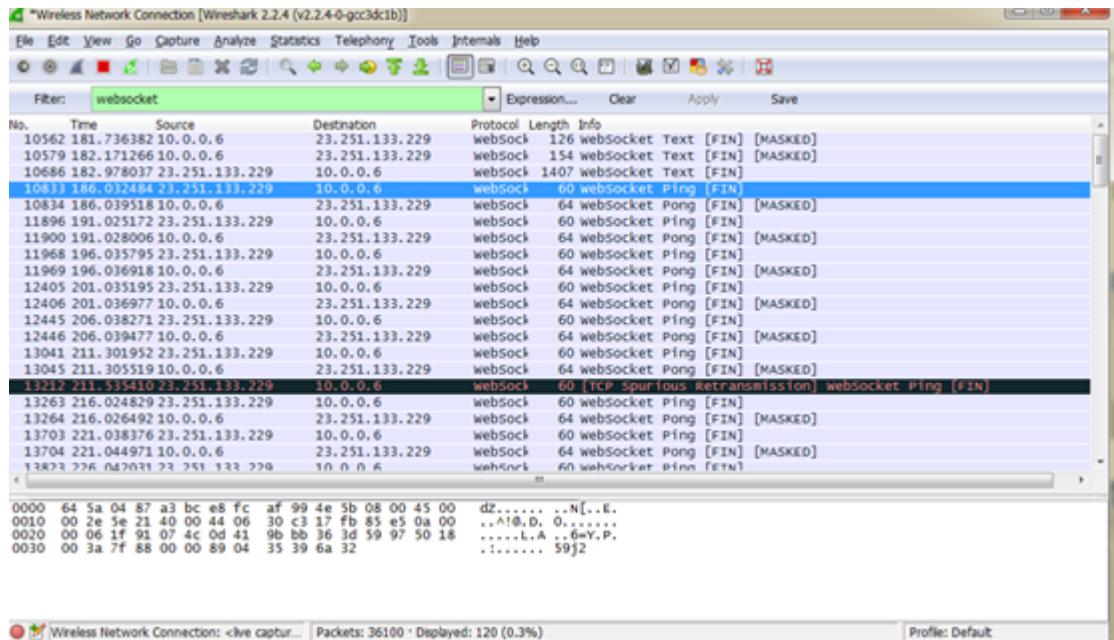
```

3006 181.879037 23.251.133.229 10.0.0.6 WebSocket Text [FIN]
3006 181.879037 100 Bytes captured (33356 bytes), 140 Bytes captured (13326 bytes) on interface 0
# IP/Port: 181.879037/23.251.133.229, Src: 10.0.0.6/Dst: 23.251.133.229, Prot: 6 (TCP)
# Source: 181.879037 (eth0:1/10.0.0.6)
# Destination: 23.251.133.229 (eth0:1/23.251.133.229)
# Type: IPv4 (0x0800)
# Internet Protocol Version 4 (IPv4)
#     Version: 4 (0x04)
#     Header Length: 20 bytes (0)
#     Differentiated Services Field: 0x00 (0x00P|0x00, ECN: Not ECN)
#     Total Length: 140
#     Identification: 0x3e20 (24096)
#     Flags: 0x02 (Don't Fragment)
#     Fragment offset: 0
#     Time to Live: 128
#     Protocol: TCP (6)
#     Header checksum: 0x3285 [validation disabled]
#     [Header checksum status: unverified]
#     Source: 10.0.0.6
#     Destination: 23.251.133.229
#     [Source GeoIP: Unknown]
#     [Destination GeoIP: unknown]
#     Transmission Control Protocol, Src Port: 8081, Dst Port: 1868, Seq: 40124, Ack: 351, Len: 135
    Source Port: 8081
    Destination Port: 1868
    [Stream Index: 148]
    Sequence number: 40124 (relative sequence number)
    [Next sequence number: 42877 (relative sequence number)]
    Acknowledgment number: 351 (relative ack number)
    Header checksum: 0x3285
    Flags: 0x018 (PSH, ACK)
    Window size value: 38
    [Calculated window size: 29898]
    [Window size scaling factor: 302]
    checksum: 0xc36a [unverified]
    [checksum status: unverified]
    urgent pointer: 0
    [SEQ/ACK analysis]:
        [RTT: 0.154577000 seconds]
        [Bytes in flight: 2773]
        [Bytes sent since last PSH flag: 4828]
        [POU size: 135]
        YEP - Line-based text data (135 bytes)
    [44 reassigned TCP segments (62293 bytes): #10600(1420), #10601(1420), #10601(1420), #10604(1420), #10606(1420), #10607(1420), #10609(1420), #10610(1420), #10611(1420), #10614(1372), #10618(1420), #10629(1420)]
    [WebSocket]:
        1... .... = Fin: True
        .0... .... = Reserved: 0x0
        ...001 = opcode: Text (1)
        0... .... = Mask: False
        1... .... = Payload length: 126 Extended Payload length (16 bits)
        Extended Payload length (16 bits): 60289
        Payload
    [Line-based text data]
        (Truncated)[{"Tst": 30.732795000000146, "color": "#000000"}, {"Tst": 34.99227200000017, "color": "#000000"}, {"Tst": 30.734795000000147, "color": "#000000"}, {"Tst": 34.99227200000017, "color": "#000000"}, {"Tst": 30.734795000000147, "color": "#000000"}]
Frame (155 bytes) | Unmasked data (95 bytes)

```

בתמונה הבאה, ניתן לראות את ויזוא התקשורת בין השירות ללקוח על ידי שליחת ping pong מידי כמו

עширויות השניה :



ניתן לראות את המהירות בה נשלחות ומתקבלות ההודעות בין השירות ללקוח (אפליקציה) על ידי התבוננות בזמןי השליחה והקבלה של ההודעות :

| | | | | | |
|------------------|----------------|----------------|---------|-----|-------------------------------|
| 46459 858.827336 | 10.0.0.6 | 23.251.133.229 | webSock | 155 | websocket Text [FIN] [MASKED] |
| 46587 859.488175 | 23.251.133.229 | 10.0.0.6 | webSock | 136 | websocket Text [FIN] |

8.3 מדריך למשתמש

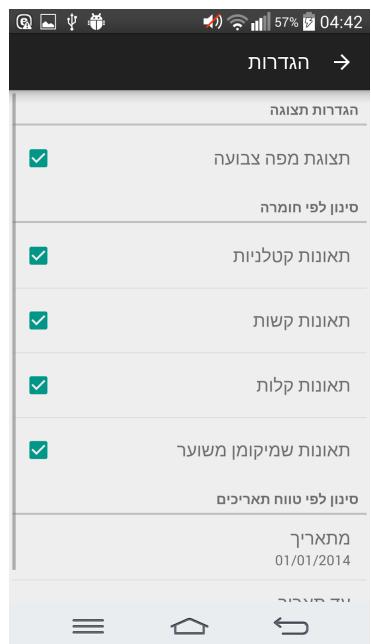
משתמש יקר, במטרה למנוע שימוש שאינו נכון באפליקציה וכן הגדלת היעילות השימוש באפליקציה
אני קרא את ההוראות הבאות:

בכוניסתך לראשונה לאפליקציה יופיע עבורך מסך כניסה להסבירים על שימוש באפליקציה אני קרא אותו
בעיוו, על מנת להשתמש באפליקציה יש לאפשר גישה לשירותי מיקום.

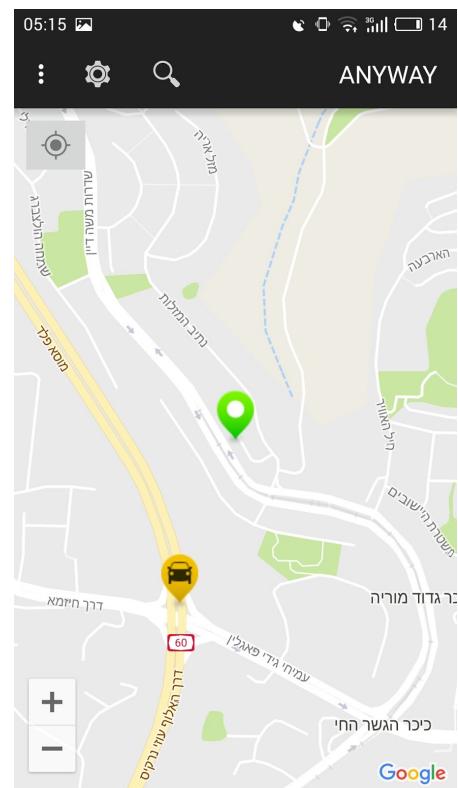
מסך כניסה בפעם הראשונה יראה כך:



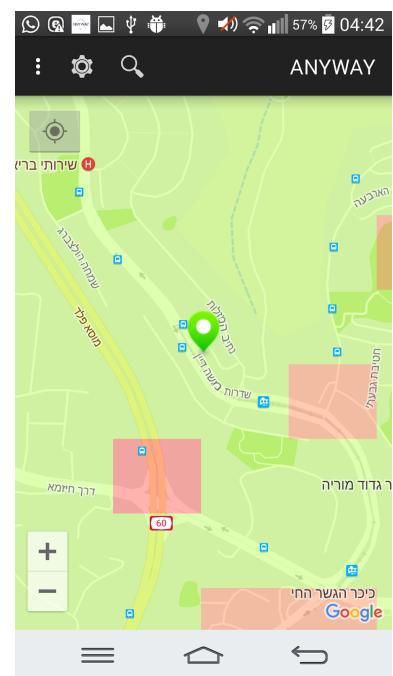
במסך ההגדרות תוכל לבחור האם תציגת האפליקציה תהיה מפה צבועה על פי רמת הסיכון שבדרך או
להישאר בתציגת תאונות בדמות סמלים על המפה



מסך בתציגות איקוניים:

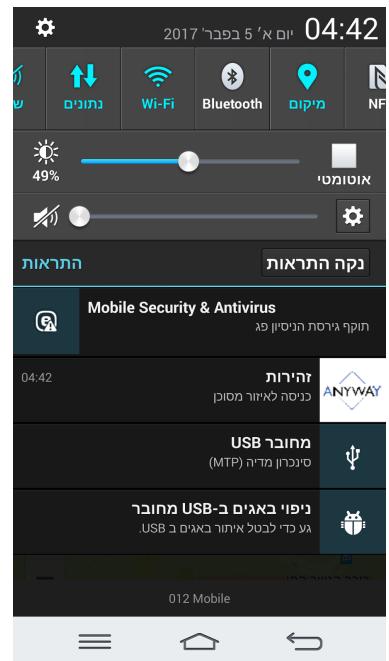


מפה צבואה על פי ריבועים :



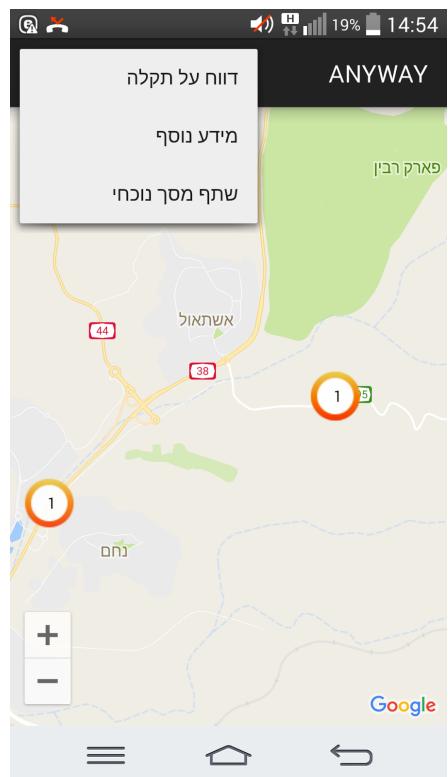
בבחירה בimap צבואה יתקבלו התראות עברו כניסה לאזור שצובע באדום על גבי המפה וגם הודעות התראה.

דוגמא להתראה המתקבלת מהאפליקציה :



חשוב לציין כי מעבר בין איזור אדום לשנהו תהיה התראה רק במידה ועברת באיזור שצבוע בצבע יrox לפניו כן, קלומר מריבוע שצבוע באדום, מעבר לאיזור שצבוע בירוק וחזרה לריבוע אדום. על מנת לצאת מהאפליקציה יש ללחוץ פעמיים מהירות על הכפתור 'חזרה לאחרו'.

בכל מקרה של תקלת אני דוחה :



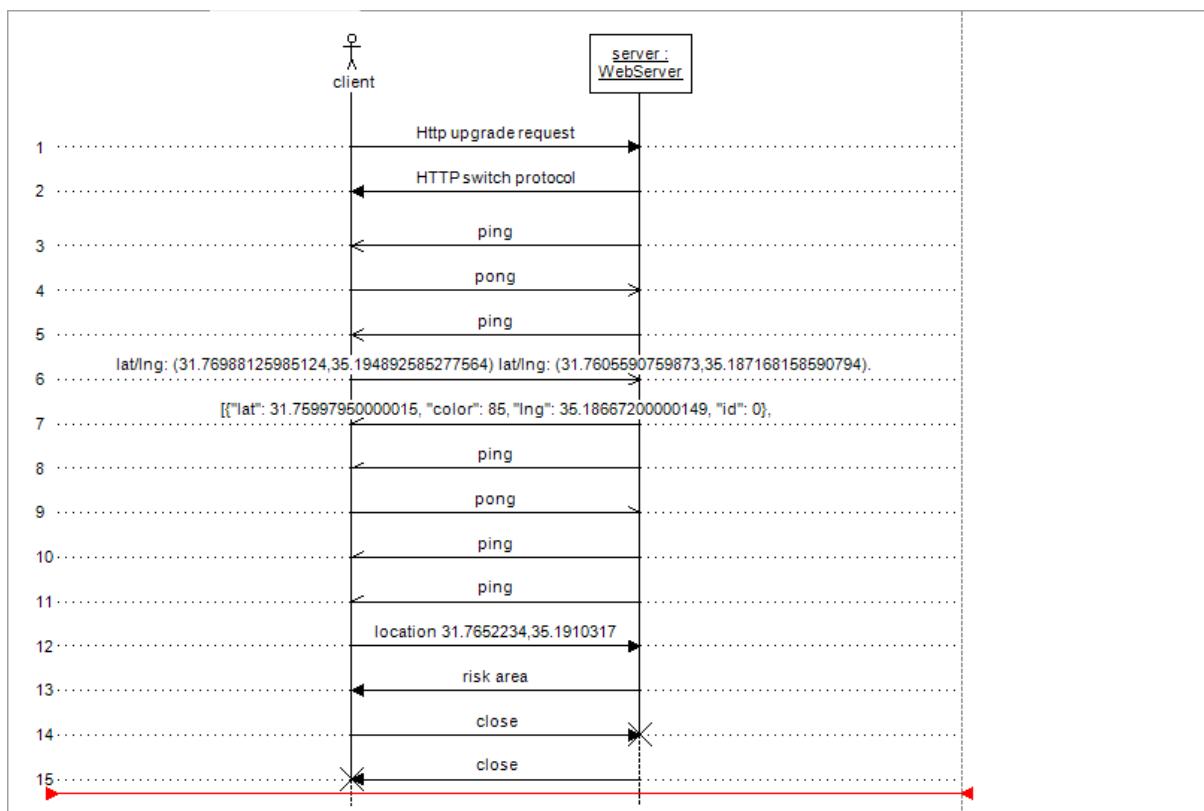
שימוש פורה ונעימים,

דניאל ושרה,

Anyway

Message sequence diagram 8.4

להלן מוצג תרשים המתאר את השתלשלות ההודעות בין השרת והלקוח על פי התקשרות שתוארה:



websocket - צד לקוח 8.5

```
/**
 * this function handle all the issues of the websocket connection
 */
private void connectWebSocket() {
    URI uri;
    try {
        uri = new URI(WS_URI);
    } catch (URISyntaxException e) {

        Toast.makeText(getApplicationContext(),"uri failed", Toast.LENGTH_SHORT).show();
        e.printStackTrace();
        return;
    }
    mWebSocketClient = new WebSocketClient(uri) {
        @Override
        /**
         * create connection with the server
         */
        public void onOpen(ServerHandshake handshakedata) {
            Log.i("Websocket", "Opened"+ handshakedata.getHttpStatusMessage());
            mWebSocketClient.send("Hello from " + Build.MANUFACTURER + " " + Build.MODEL);

        }

        @Override
        /**
         * the function received a message from the server, and create a runnable.
         * the function react due to the message context
         */
    };
}
```

```

public void onMessage(String s) {
    final String message = s;
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Log.i("Websocket", "got message: " + message);
            if(message.equals("2") || message.equals("3"))

            {
                if( message.equals("2")) {
                    if (!mInDanger) {
                        addNotification();
                        mInDanger = true;
                        try {
                            speak("be carefull");
                        } catch (Exception e) {
                            Log.i("ring", "problem");
                        }
                    }
                }
            }
            else //message of 3 arrived
            {
                if (mInDanger) { //was in not dangerous area
                    mInDanger = false;
                    try {
                        speak("out of danger");
                    } catch (Exception e) {
                        Log.i("ring", "problem");
                    }
                }
            }
        }
    }
}
else
try {
    clearMap();
    JSONArray jsonArr = new JSONArray(message);

    for (int i = 0; i < jsonArr.length(); i++)
    {
        JSONObject jsonObj = jsonArr.getJSONObject(i);

        Log.i("json", String.valueOf(jsonObj.getDouble("lat")));
        getColor(jsonObj.getInt("color"), jsonObj.getDouble("lat"), jsonObj.getDouble("lng"));
    }
}

```

```
        log.i("json", JSONUtil.toJsonString(jsonObj));
        getColor(jsonObj.getInt("color"), jsonObj.getDouble("lat"), jsonObj.getDouble("lng"));
    }

} catch (JSONException e) {
    e.printStackTrace();
}

});

}

@Override
/***
 * closing the connection with the server, send the server close message
 * if the reason of calling that function is not from on destroy the function will try to reconnect
 */
public void onClose(int code, String reason, boolean remote) {
    mWebSocketClient.send("closing" + reason);
    mWebSocketClient.getConnection().close(code, reason);

    Log.i("Websocket", "Closed " + reason);
    clearMap();
    try {
        if (reason != "onDestroy") {
            mWebSocketClient.connect();
        }
    } catch (Exception e) {
        Toast.makeText(getApplicationContext(), "connection failed", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onError(Exception ex) {
    Log.i("Websocket", "Error \n" + ex);
}

};

mWebSocketClient.connect();
}
```

websocket צד שרת 8.6

```
def onConnect(self, request):
    print("Client connecting: {0}".format(request.peer))

def onOpen(self):
    print("WebSocket connection open.")

def onMessage(self, payload, isBinary):
    if isBinary:
        print("Binary message received: {0} bytes".format(len(payload)))
    else:
        print("Text message received: {0}".format(payload.decode('utf8')))
        if((payload.split()[0]=="lat/lng:")):
            try:
                numbers = re.findall(r"(\d+\.\d+)", payload)
                answerFromDb = connectDb.screenColor(numbers[0], numbers[1], numbers[2],numbers[3])
                self.sendMessage(answerFromDb)
            except:
                pass
        else:
            try:
                if((payload.split()[0]=="location")):
                    numbers = re.findall(r"(\d+\.\d+)", payload)
                    inSquare=connectDb.find_square(numbers[0], numbers[1])
                    if(inSquare==1):
                        self.sendMessage("2")
                    else:
                        if(inSquare== -1):
                            self.sendMessage("3")
            except:
                pass

def onClose(self, wasClean, code, reason):
    print("WebSocket connection closed: {0}".format(reason))
    self.sendMessage("closed")
    #WebSocketServerProtocol.send_close(code,reason)
```

javaDoc קבצי 8.7

Field Summary

| Fields | |
|---|--------------------------------------|
| Modifier and Type | Field and Description |
| double | alpha |
| double | beta |
| int | i |
| private java.lang.String | LOG_TAG |
| private boolean | mDoubleBackToExitPressedOnce |
| private android.support.v4.app.FragmentManager | mFragmentManager |
| private com.google.android.gms.common.api.GoogleApiClient | mGoogleApiClient |
| private boolean | mOpsDialogNeverShowed |
| private boolean | mInDanger |
| private static int | MINIMUM_ZOOM_LEVEL_TO_SHOW_ACCIDENTS |
| private java.util.List<AccidentCluster> | mLastAccidentsClusters |

```

private static int START_ZOOM_LEVEL;
private android.speech.tts.TextToSpeech TTS;
private static java.lang.String TUTORIAL_SHIMMER_KEY;
private java.lang.String WS_URL;

Fields inherited from class android.app.Activity
DEFAULT_KEYS_DECALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL, DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, FOCUSED_STATE_SET, RESULT_CANCELED, RESULT_FIRST_USER, RESULT_OK

Fields inherited from class android.content.Context
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_CIPS_SERVICE, APPDIRECTOR_SERVICE, AUDIO_SERVICE, BATTERY_SERVICE, BND_ANONYMIZE_CLIENT, BND_ANONYMIZE_ACTIVITIES, BND_ALLOW_OWN_MANAGEMENT, BND_APP_CREATE, BND_APP_DESTROY, BND_EXTERNAL_SERVICE, BND_IMPORTANT, BND_NOT_PERMISSIONED, BND_WAIT_PRIORITY, BROADCASTS_SERVICE, CAMERA_SERVICE, CAPTIONING_SERVICE, CARBONIZER_CARDSERVICE, CLIPBOARD_SERVICE, CONNECTIVITY_SERVICE, CONSUMER_ID_SERVICE, CONTEXT_INCLUDE_CODE, CONTEXT_RESTRICTED, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE, DYNAMICSERVICE, FINGERPRINT_SERVICE, HARWARE_PROPERTIES_SERVICE, INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYBOARD_SERVICE, LAUNCHER_APP_SERVICE, LAUNCHER_INFLATER_SERVICE, LOCATION_SERVICE, MEDIA_PROJECTION_SERVICE, MEDIA_SCANNER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE, MODE_APPLIANCE, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_NO_LOCALIZED_COLATORS, MODE_PRIVATE, MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_STATS_SERVICE, NOTIFICATION_SERVICE, Nfc_SERVICE, POWER_SERVICE, PRINT_SERVICE, RECEIVER_SERVICE, SENSOR_SERVICE, SHORTCUT_SERVICE, STREAM_SERVICE, SYSTEM_HEALTH_SERVICE, TELECOM_SERVICE, TELEPHONY_SERVICE, THREADSCRIPTION_SERVICE, TEXT_SERVICE, UNKNOWN_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, USER_SERVICE, VIBRATOR_SERVICE, WALLPAPER_SERVICE, WiFi_SERVICE, WiFi_SERVICE, WiFi_SERVICE

Fields inherited from Interface com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks
GAINED_NETWORK_LIST, GAINED_NETWORK_UNREGISTERED

Fields inherited from Interface android.content.ComponentCallbacks2
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE, TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE, TRIM_MEMORY_UI_HIDDEN

Constructor Summary
Constructors
Constructor and Description
MarkActivity()

Method Summary
All Methods Instance Methods Concrete Methods
Modifier and Type Method and Description
void addAccidentMarker(com.google.android.gms.maps.model.Accident accident)
Add accident marker to map

void addAllAccidentCluster(java.util.List<com.google.android.gms.maps.model.Accident> accidents)
Add all AccidentCluster set to markers to map

void addAllDiscussionMarker(com.google.android.gms.maps.model.Discussion discussion)
Add Discussion marker to map

void addMarkersOnMap()
get all accidents and discussions that is not on the map from MarkersManager and add them to map

private void addNotification(com.google.android.gms.maps.model.LatLng)
function that builds the notifications logo+ an Activity size: 64x64 px intent that connect the notification to main activity build the notification by systemService

protected void initGoogleApiClient()
Builds a GoogleApiClient.

protected void initLocationListener(com.google.android.gms.location.LocationRequest)
Uses a LocationSettingsRequest to build a LocationSettingsRequest that is used for checking if a device has the needed location settings.

private boolean checkIfAppHasPermission(Intent)
check if user opened the intent from a link shared with him, like: http://www.away.co.il/...parameters

protected void checkLocationSettings(com.google.android.gms.location.LocationRequest)
Check if the device's location settings are adequate for the app's needs using the SettingsApi.checkLocationSettings(GoogleApiClient, LocationSettingsRequest) method, with the results provided through a PendingResult.

private void clearMap()
Clear the map from markers and re-adding search and location markers

private void connectWebsocket()
this function handle all the issue of the websocket connection

protected void createLocationRequest()
Sets up the location request.

private void getCoordinates(double lat, double longi)
get coordinates from URL and set app preferences and map location accordingly

private void getMarkersFromServer()
get parameters from URL and start building accidents for current view and settings

void onBackgroundFreed()
onCameraChange(com.google.android.gms.maps.model.CameraPosition cameraPosition)

void onConnected(android.os.Bundle bundle)
update the map when the user connect

void onConnectIsEnabled(com.google.android.gms.common.ConnectionResult connectionResult)
onConnectIsEnabled(int i)

protected void onCreate(android.os.Bundle savedInstanceState)
onCreate(android.os.Bundle savedInstanceState)

boolean onCreateOptLocations(android.view.Menu menu)
onCreate()

protected void onDestory()
onDestory()

void onInfoWindowClick(com.google.android.gms.maps.model.Marker marker)
onInfoWindowClick(android.location.Location location)

void onMapLoaded()
onMapLoaded()

void onMapLongClick(com.google.android.gms.maps.model.LatLng latLang)
onMapLongClick(com.google.android.gms.extensions.GoogleMap googleMap)

boolean onMarkerClick(com.google.android.gms.extensions.Marker marker)
onMarkerClick(com.google.android.gms.extensions.Marker marker)

void onMyLocationButtonClick(android.view.View view)
Handle for "MyLocation" button

boolean onOptLocItemClicked(android.view.MenuItem item)
onOptLocItemClicked()

void onPause()
onPause()

protected void onRestart()
onRestart()

void onSetup()
onSetup()

private boolean setMapToLocation(com.google.android.gms.maps.model.LatLng latLang, int zoomLevel, boolean animate)
move the camera to specific location, if the map is ready

private boolean setMapToLocation(android.location.Location location, int zoomLevel, boolean animate)
move the camera to specific location, if the map is ready

private void setSettings(java.util.Date start_date, java.util.Date end_date, int show\_fatal, int show\_error, int show\_light, int show\_info, int zoom, double latitude, double longitude)
Save all parameters to SharedPreference and move to lat,log specified

private void setViewport(com.google.android.gms.maps.model.LatLngBounds)
setViewport()

void showAboutInfoDialog()
showAboutInfoDialog()

```


MINIMUM_ZOOM_LEVEL_TO_SHOW_ACCIDENTS

```
private static final int MINIMUM_ZOOM_LEVEL_TO_SHOW_ACCIDENTS
```

See Also:

Constant Field Values

TUTORIAL_SHOWED_KEY

```
private static final java.lang.String TUTORIAL_SHOWED_KEY
```

See Also:

Constant Field Values

LOG_TAG

```
private final java.lang.String LOG_TAG
```

mMap

```
private com.androidmapsextensions.GoogleMap mMap
```

mFragmentManager

```
private android.support.v4.app.FragmentManager mFragmentManager
```

mMapFragment

```
private com.androidmapsextensions.SupportMapFragment mMapFragment
```

mNewInstance

```
private boolean mNewInstance
```

mMapIsInClusterMode

```
private boolean mMapIsInClusterMode
```

```
mDoubleBackPressedOnce  
private boolean mDoubleBackPressedOnce  
  
mSentUserLocation  
private boolean mSentUserLocation  
  
mShowedDialogAboutInternetConnection  
private boolean mShowedDialogAboutInternetConnection  
  
mShowedTutorialBefore  
private boolean mShowedTutorialBefore  
  
mInDanger  
private boolean mInDanger  
  
mLastAccidentsClusters  
private java.util.List<AccidentCluster> mLastAccidentsClusters  
  
mSharedPrefListener  
private android.content.SharedPreferences.OnSharedPreferenceChangeListener mSharedPrefListener  
  
mGoogleApiClient  
private com.google.android.gms.common.api.GoogleApiClient mGoogleApiClient  
  
mLocation  
private android.location.Location mLocation
```

```
mPositionMarker  
private com.androidmapsextensions.Marker mPositionMarker  
  
mSearchResultMarker  
private com.androidmapsextensions.Marker mSearchResultMarker  
  
mLocationRequest  
private com.google.android.gms.location.LocationRequest mLocationRequest  
  
mLocationSettingsRequest  
private com.google.android.gms.location.LocationSettingsRequest mLocationSettingsRequest  
  
mGpsDialogNeverShowed  
private boolean mGpsDialogNeverShowed  
  
WS_URI  
private final java.lang.String WS_URI  
See Also:  
Constant Field Values  
  
mWebSocketClient  
public org.java_websocket.client.WebSocketClient mWebSocketClient  
  
poly  
public com.androidmapsextensions.Polygon poly  
  
alpha  
public final double alpha
```

beta

```
public final double beta
```

See Also:

[Constant Field Values](#)

i

```
public int i
```

tts

```
private android.speech.tts.TextToSpeech tts
```

Constructor Detail

MainActivity

```
public MainActivity()
```

Method Detail

onCreate

```
protected void onCreate(android.os.Bundle savedInstanceState)
```

Overrides:

[onCreate in class android.support.v7.app.AppCompatActivity](#)

speak

```
private void speak(java.lang.String text)
```

Parameters:

`text`, - convert the text to audio

onDestroy

```
protected void onDestroy()
```

Overrides:

onDestroy in class android.support.v7.app.AppCompatActivity

onStart

```
protected void onStart()
```

Overrides:

onStart in class android.support.v4.app.FragmentActivity

onStop

```
protected void onStop()
```

Overrides:

onStop in class android.support.v7.app.AppCompatActivity

onRestart

```
protected void onRestart()
```

Overrides:

onRestart in class android.app.Activity

onResume

```
public void onResume()
```

Overrides:

onResume in class android.support.v4.app.FragmentActivity

onPause

```
protected void onPause()
```

Overrides:

onPause in class android.support.v4.app.FragmentActivity

setUpMapIfNeeded

```
private void setUpMapIfNeeded()
```

onMapReady

```
public void onMapReady(com.androidmapsextensions.GoogleMap googleMap)
```

Specified by:

onMapReady in interface com.androidmapsextensions.OnMapReadyCallback

```
buildLocationSettingsRequest
protected void buildLocationSettingsRequest()
Uses a LocationSettingsRequest.Builder to build a LocationSettingsRequest that is used for checking if a device has the needed location settings.

checkLocationSettings
protected void checkLocationSettings()
Check if the device's location settings are adequate for the app's needs using the SettingsApi.checkLocationSettings(GoogleApiClient, LocationSettingsRequest) method, with the results provided through a PendingResult.

onConnected
public void onConnected(Bundle bundle)
update the map when the user connect
Specified by:
onConnected in interface com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks
Parameters:
bundle - 

onConnectionSuspended
public void onConnectionSuspended(int i)
Specified by:
onConnectionSuspended in interface com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks

onConnectionFailed
public void onConnectionFailed(ConnectionResult connectionResult)
Specified by:
onConnectionFailed in interface com.google.android.gms.common.api.GoogleApiClient.OnConnectionFailedListener

onLocationChanged
public void onLocationChanged(Location location)
Specified by:
onLocationChanged in interface com.google.android.gms.location.LocationListener
Parameters:
location - get tcurrent ocation send to server the new location

onCreateOptionsMenu
public boolean onCreateOptionsMenu(Menu menu)
Overrides:
```

onInfoWindowClick

```
public void onInfoWindowClick(com.androidmapsextensions.Marker marker)

Specified by:
onInfoWindowClick in interface com.androidmapsextensions.GoogleMap.OnInfoWindowClickListener

Parameters:
marker - click on info button marker show details on the accident
```

onMapLongClick

```
public void onMapLongClick(com.google.android.gms.maps.model.LatLng latLng)

Specified by:
onMapLongClick in interface com.google.android.gms.maps.GoogleMap.OnMapLongClickListener

Parameters:
latLng -- the position of the click in, is the user press long click on the map he can add new discussion
```

onCameraChange

```
public void onCameraChange(com.google.android.gms.maps.model.CameraPosition cameraPosition)

Specified by:
onCameraChange in interface com.google.android.gms.maps.GoogleMap.OnCameraChangeListener

Parameters:
cameraPosition -- the position to move to the position is LatLng of me and se the function send to the server this new position
```

checkIfAppOpenedFromLink

```
private boolean checkIfAppOpenedFromLink()

check if user opened the intent from a link shared with him, like: http://www.anyway.co.il/...parameters

Returns:
true if user open the app from the link, false otherwise
```

getDataFromSharedURL

```
private boolean getDataFromSharedURL()  
  
get parameters data from URL and set app preferences and map location accordingly  
Returns:  
true if all parameters are correct, false otherwise
```

setSettingsAndMoveToLocation

```
private void setSettingsAndMoveToLocation(java.util.Date start date,
```

setMapToLocation

```
private boolean setMapToLocation(android.location.Location location,  
                                int zoomLevel,  
                                boolean animate)
```

move the camera to specific location, if the map is ready

Parameters:

location - location to move to

zoomLevel - zoom level of the map after new location set

animate - animate the map movement if true

```

addMarkersToMap
public void addMarkersToMap()
get all accidents and discussions that is not on the map from MarkersManager and add them to map

addAccidentToMap
public void addAccidentToMap(Accident a)
Add accident marker to map
Parameters:
a - Accident to add

addDiscussionToMap
public void addDiscussionToMap(Discussion d)
Add Discussion marker to map
Parameters:
d - Discussion to add

showAboutInfoDialog
private void showAboutInfoDialog()
Show information about Anyway in dialog

updateMapFromSearchResult
public void updateMapFromSearchResult(com.google.android.gms.maps.model.LatLng searchResultLocation,
java.lang.String searchResultAddress)
Move map to location and add marker to specify searched location When search result selected in the result dialog(@dialogs.SearchDialogs.searchAddress) it call this method
Parameters:
searchResultLocation - Location of search result
searchResultAddress - Address of search result

onBackPressed
public void onBackPressed()
Overrides:
onBackPressed in class android.support.v4.app.FragmentActivity

clearMap
private void clearMap()
Clear the map from markers and re-adding search and location markers

onMyLocationButtonClick
public void onMyLocationButtonClick(android.view.View view)
Handler for 'MyLocation' button
Parameters:
view - Button clicked

showTutorialIfNeeded
private void showTutorialIfNeeded()
Show tutorial activity if it never shown before Also save the state of activity as 'shown' in Shared Preferneces

connectWebSocket
private void connectWebSocket()
this function handle all the issues of the websocket connection

```

```
getColor

private void getColor(int H,
                      double lat,
                      double lng)

Parameters:
H -- hue of the square color
lat -- latitude of the left bottom corner of the square
lng -- longitude of the right bottom of the square the function draw a polygon on the map and color it according to the H parameter

addNotification

private void addNotification()

function that builds the notifications logo= awlsixty size: 64*64 px intent that connect the notification to main activity build the notification by systemService
```