**המחלקה להנדסת תוכנה**



**חיבור זה מהווה חלק מהדרישות לקבלת תואר ראשון בהנדסה**

מאת:

דני וולפוב 306319138

אסף אסל \*\*\*

מנחה אקדמי: ד"ר יעל נצר אישור:\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_

רכז הפרויקטים: ד"ר ראובן יגל אישור:\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_

# תקציר

כיום כ-5% לערך מכלל האוכלוסייה בישראל מגדלת כלב בתור חיית מחמד. אנו מאמינים כי נובע הצורך להפגיש בין הכלבים ובעליהם בזמן הטיול היומי על מנת ששני הצדדים יפיקו את המרב מהטיול – הכלב ייהנה מחברה של בני מינו וטיול ארוך יותר, בעוד הבעלים ירוויח איש שיחה וחבר לעתיד.

משיחות עם חברים ומכרים בעלי כלבים הבנו שאין כלי ייעודי למטרה זו. הכלי שנפוץ כיום הוא אפליקציית "וואטסאפ" אשר משמשת ליצירת קבוצות משתמשים לכל גינה שכונתית המתפתחת מפה לאוזן הנקראות "גינות שכונתיות".

הפרויקט עוסק ביצירת אפליקציה חברתית שיתופית לאנדרואיד בה אנשים יוכלו לשתף את מיקומם בזמן הטיול היומי עם הכלב. האפליקציה עובדת ע"י עדכונים של משתמשים בזמן אמת, משתמשי האפליקציה יכולים ליצור קשרים חדשים עם בעלי כלבים אחרים באמצעות הודעות, לקבוע מפגשים ולעקוב אחר הכלב. האפליקציה מסתמכת על מיקום של המשתמשים ועל סביבת הטיול שלהם. . הפרויקט בנוי מצד שרת שנכתב בNodeJS ואפליקציה למכשירי אנדרואיד מבוססי מערכת בשפת JAVA .

נבעה בעיקר מהרצון ללמוד לפתח IOSחשוב לציין שהבחירה שלנו לפתח את האפליקציה דווקא בסביבת אנדרויד תוך כדי אינטגרציה עם מערכות אחרות. בנוסף אנדרויד היא מערכת הפעלה נפוצה בעולם בשנים האחרונות ובעלת תמיכה נרחבת.

בשלב הראשון ניתן יהיה לעלות לאוויר עם אנדרויד בלבד.

# הצהרה

הפרויקט נעשה בהנחיית ד"ר יעל נצר, במחלקה להנדסת תוכנה.

המכללה האקדמית להנדסה ירושלים– עזריאלי. החיבור מציג את עבודתנו האישית ומהווה חלק מהדרישות לקבלת תואר ראשון בהנדסה.

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | https://github.com/Bu1ly/Bark-Android-app |
| 2 | יומן | https://trello.com/b/xKaw3Hmn/barkapp |
| 3 | ניהול פרויקט (אם בשימוש) | https://github.com/Bu1ly/Bark-Android-app/issues |
| 4 | הפצה | https://barkandroid.herokuapp.com/ |
| 5 | סרטון אב-טיפוס | https://www.youtube.com/watch?v=jOKgC-VX0ak |

# תוכן עניינים:

1. מילון מונחים.................................................................................7
2. תיאור מסגרת הפרויקט .................................................................8
3. תיאור הבעיה ...............................................................................9
   1. מודל עסקי .........................................................................9
   2. דרישות ואפיון הבעיה ..........................................................9
   3. הבעיה מבחינת הנדסת תוכנה .............................................11

4.תיאור הפתרון .............................................................................11

* 1. מהי המערכת ......................................................................11
  2. תהליכים ונתוני המערכת .......................................................11
  3. תיאור הפתרון המוצע ............................................................11
  4. תיאור הכלים המשמשים לפתרון .............................................12

1. תיאור המערכת שמומשה ............................................................13
2. תוכנית בדיקות .........................................................................14

6.1 טבלת בדיקות ....................................................................14

1. סקירה ספרותית ........................................................................17
2. סיכום ומסקנות..........................................................................17
3. ספרות ....................................................................................18

11. קישורים למערכות ניהול הפרויקט ובקרת תצורה ...........................10

1. נספחים .................................................................................19

## מילון מונחים

, תיאור למקבצי ספריות קוד – Application Programming Inteface – **API**

פקודות פונקציות בהן המתכנת יכול לעשות שימוש פשוט מבלי לכתוב אותן בעצמו.

. Operation System מערכת הפעלה – **OS**

. Integrated Development Environment סביבת עבודה – **IDE**

**Android** – מערכת הפעלה לטלפונים חכמים.

**Android Studio** – סביבת עבודה לפיתוח אפליקציות לטלפונים חכמים.

**Node.js** - סביבת פיתוח צד שרת מונחת אירועים בשפת .JavaScript

**WebStorm** – סביבת העבודה בה נכתב השרת בNode.js.

**N**ot **O**nly **SQL** – **NoSQL** –קטגוריה בבסיס נתונים שאינו ממודל במבנה טבלאי.

**MongoDB** - בסיס נתונים בNoSQL מבוסס אובייקטים JSON, בסיס נתונים הנשען על מבנה של מסמך.

מתחילהHTTP התקשורת ב־ – Hypertext Transfer Protocol - **HTTP**

ביצירת שיחה בין השרת ללקוח באמצעות פרוטוקולTCP בשכבת התעבורה של פרוטוקול TCP/IP ונמשכת בסדרה של בקשות(requests) ותשובות(responses) שנשלחות על ידי הלקוח והשרת, בהתאמה.

## תיאור מסגרת הפרויקט

הפרויקט הוא פרויקט יזמות שבא לפתור בעיה שעולתה משיחות עם חברים וכן, נתקלנו בה בעצמנו.

# תיאור הבעיה

בשנים האחרונות עם כניסת האינטרנט לשימוש המוני ויומיומי המידע הופך להיות נגיש יותר. כניסת הטלפונים החכמים והאפליקציות הפך את אפשרויות החיבור לנגישות וזמינות יותר. מאז כניסת הרשתות החברתיות לפני מספר שנים האנונימיות שהייתה נהוגה ברשת מתמעטת ורואים יותר רצון של אנשים לשתף חברים ואנשים בחוויות שהם עוברים. השילוב בין הרצון לשתף וזמינות המידע יצר המון אפשרויות של ניצול חוכמת ההמונים לטובת המשתמש הפרטי.

הצורך ביצירת מערכת שימושית לבעלי כלבים, אשר תחליף את שיטות ההתקשרות הלא אחידות הקיימות כיום, כגון: פייסבוק, ״וואטסאפ״ (גינות שכונתיות), אשר תיתן מענה ופלטפורמה ייעודית על בסיס רשת חברתית המאפשרת ניהול קשרים, גילוי, ניהול קבוצות בצורה יעילה וידידותית.

* בחירת מסד נתונים אשר יתאים לנתונים של המשתמשים כולל שמירת המיקום שלהם ויעבוד בזמן אמת ומספיק טוב ומהר.
* תכנון הלוגיקה שמאחורי ממשק המשתמש בשפת Java , לצורך אפליקציית האנדרואיד.
* תכנון אלגוריתם ״חכם״ לצורך מציאת אנשים (בעלי כלבים) בזמן אמת אשר מטיילים עם הכלב שלהם בסביבה שלי.
* מציאת פלטפורמה מתאימה למפות – סימונים לפי נק׳ ציון, תמיכה ב Real Time Data.
* תכנון ואפיון תבניות תיכון לאפליקציה כולל כל הממשקים אשר מתממשקים עם דגש לסדר קל לתפעול וידידותי למשתמש.
* מציאת שרת אחסון אשר יהיה מספיק מהיר ויעמוד בקצב הגידול של המשתמשים באפליקציה.

# הבעיה מבחינת הנדסת תוכנה

# ארכיטקטורת צד שרת

* כתיבת מערכת תוכן שיושבת על צד שרת שמקבל תוכן ממשתמשים.
* ניהול API המאפשר למכשירים מרוחקים להתחבר, להעלות ולהוריד מידע מהשרת.
* מערכת ניהול משתמשים – הרשמה , התחברות עדכון נתונים אישיים , שמירת סיסמאות בצורה מוצפנת.
* תכנון אלגוריתם ״חכם״ לצורך מציאת אנשים (בעלי כלבים) בזמן אמת אשר מטיילים עם הכלב שלהם בסביבה שלי.
* מציאת פלטפורמה מתאימה למפות – סימונים לפי נק׳ ציון, תמיכה ב Real Time Data.
* תכנון ואפיון תבניות תיכון לאפליקציה כולל כל הממשקים אשר מתממשקים עם דגש לסדר קל לתפעול וידידותי למשתמש.
* בחירת מסד נתונים אשר יתאים לנתונים של המשתמשים כולל שמירת המיקום שלהם ויעבוד בזמן אמת ומספיק טוב ומהר.
* מערכת ניהול לנתונים עבור מקומות מפגש – שמירת המיקום, שמות ותמונות.

# ארכיטקטורת צד לקוח

* ניהול ארכיטקטורה לאפליקציית האנדרואיד שתפריד בין החלק הגרפי\ויזואלי, לבין המודל עצמו.
* יצירת מנגנון באפליקציה שידע לשמור חלק מהנתונים לוקאלית, ואת חלקם לעדכן בזמן אמת דרך חיבור לשרת.
* שימוש בAPI של אנדרואיד לצורך חיבור למיקום, למפות, לחיבורים חיצוניים כגון רשת ומצלמה.

• שימוש בספריות נוספות המספקות תשתיות נחוצות לפעילות האפליקציה, כגון תקשורת בפרוטוקול HTTP, ולצורך העלאה והורדת תמונות.

**מודל עסקי**

האפליקציה היא חינמית, המטרה שלנו להשיג כמה שיותר משתמשים ובכך להפוך את האפליקציה למשמעותית. אוכלוסיית בעלי הכלבים הולכת וגדלה בעולם ולכן דרך ליצירת קשר היא חשובה. יצירת משתמשים פעילים בתחום יכולה להוות מקור מידע ומפגש, היכול להיות מנוצל לטובת הכלבים ובעליהם.

# **תיאור הפתרון**

הצורך ביצירת פלטפורמה לבעלי כלבים הביא אותנו לרעיון של אפליקצייה לטלפון חכם.

רעיון האפליקציה הוא לחבר בין בעלי הכלבים בזמן הטיול, לתת יכולת מעקב אחר משתמשים , לעקוב אחר חיסוני הכלב ועוד כמה פ'יצרים אשר יתנו את המענה הכי טוב בכל מקום אשר נהיה בו!

לצורך העבודה חילקנו תחומי אחריות:

דני וולפוב – צד שרת ומסדי נתונים.

אסף אסל – צד לקוח.

# תיאור המערכת

לאחר איפיון ותיכון האפליקציה הסכמנו שצד הלקוח יהיה באנדרויד, ה-UI יעבוד ישירות עם שרת ה-NODEJS שבחרנו ע"י שליחת בקשות HTTP עם אובייקטי JASON וקבלה של המידע חזרה לאחר חישוב השרת או פנייה למסד הנתונים באובייקטים של JASON גם כן.

כמו כן החלטנו לממש את השרת ומסד הנתונים בתבנית SINGLETONE.

המערכת בנויה בתצורת "שרת – לקוח". אנו עובדים לפי מודל "מפל המים".

חלק מתהליך החשיבה על פתרון הבעיה היה להתאים את התממשקות בין רכיבי המערכת : שרת, מסדי נתונים, צד לקוח בצורה מהירה תוך כדי מחשבה על העבודה העתידית. ביצענו בדיקה מעמיקה על סוגי המידע המועברים בין רכיבי המערכת, החלטנו לעבוד עם קבצי JASON מפני שהעברת המידע בג'ייסונים היא קלה, נוחה לבדיקה ונוחה ליצירת אובייקטים חדשים. בעקבות החלטה זו בחרנו לעבוד עם סיפריית VOLLEY אשר תומכת בקבצי ג'ייסון, שרת NODEJS היודע לפרסס ג'ייסונים ומסד נתונים MONGODB אשר מקבל אובייקטים של ג'ייסון.

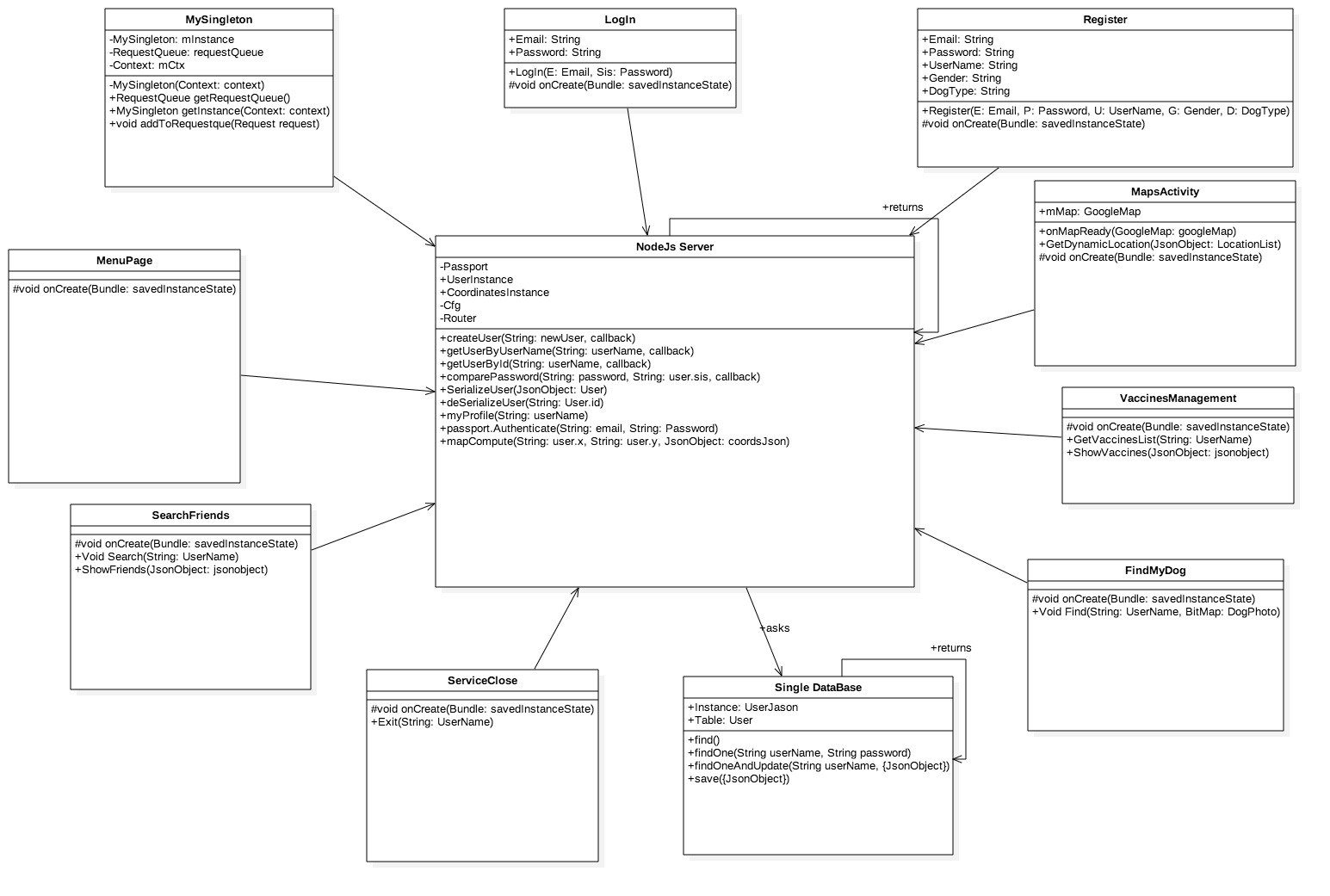
**צד שרת** – הבחירה הייתה בשרת מסוג "NodeJS" המאוחסן ב-HEROKU, תומך ב-REAL TIME, עובד עם ג'ייסונים, בעל פאנל שליטה וויזואלי, CLI נוח ופשוט, חינמי!

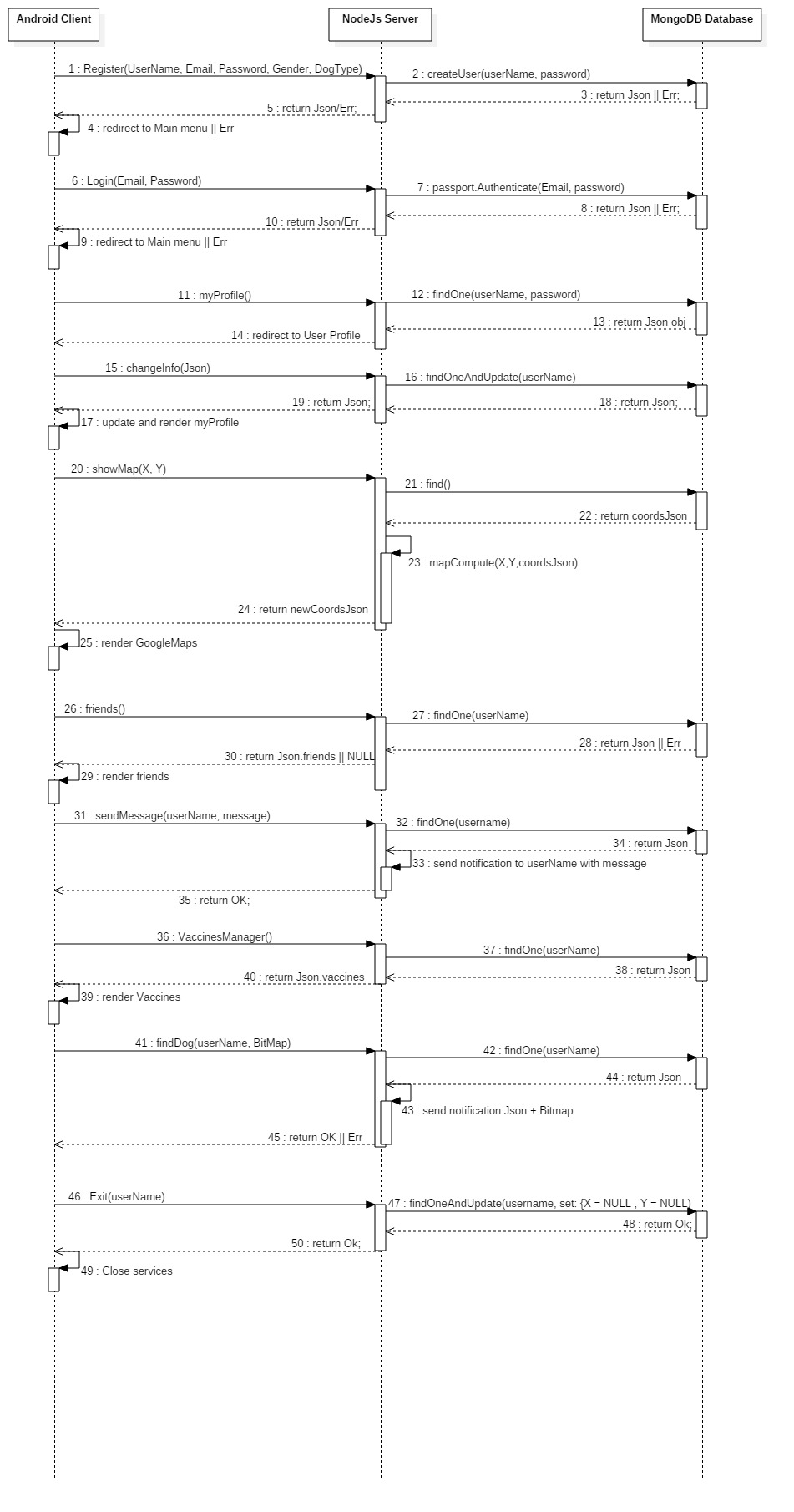
**מסד נתונים** – הגורם העיקרי הוא המחיר, לכן היו לנו כמה אופציות לבחירה,

שרת של גוגל לדוגמא – MLAB שמספק 500GB בחינם!

**צד לקוח** – בחרנו לפתח את האפליקציה לפלטפורמת האנדרואיד (בשפת JAVA), הבחירה בוצעה בפלטפורמה זו מפני שאנו שואפים לשפר את מיומנותנו באנדרואיד בנוסף לניסיון עבר במכללה, ומצד שני שהדרישה שלנו היא לאפליקציה שתהיה על מכשיר סלולארי עם התממשקות לשירותי מיקום.

## מהי המערכת

**Class Diagram**

**Sequence Diagram**

\*תרשימים ברזולוציה מלאה ניתן לראות במערכת ניהול הפרויקט ((GitHub

## הכלים המשמשים לפתרון

- Android studio – עורך קוד לצד הלקוח.

- GitHub – מאגר הפרוייקט.

- Google Map API – שירות המפות לאפליקצייה.

- WebStrom- עורך קוד לצד שרת .

-Mlab DataBase - מסד הנתונים המסופק ע"י MLAB ומאוחסן בשרתי גוגל.

- HEROKU NODEJS - דומיין המאחסן את שרת הNODEJS שלנו.

- TRELLO – מערכת לניהול יומן הפרוייקט.

- מכשירי אנדרויד.

תכנית בדיקות

הבדיקות בפרוייקט זה יבוצעו במספר סביבות:

**סביבה פרויקטאלית** - בה יבוצעו בעיקר בדיקות פונקציונאליות של הפרויקט.

**סביבה ארגונית** - בה יבוצעו פעולות ובדיקות האינטגרציה עם מערכות משיקות, ובו יערכו בדיקות הביצועים, כשל והתאוששות, אבטחת מידע .

**סביבה באתר לקוח** - יבוצעו בדיקות המדמות תהליכי לקוח בתנאי עבודה וקונפיגורציה של מערכות המדמות מצבים אמתיים.

# תשתית וטכנולוגיה

|  |  |  |
| --- | --- | --- |
|  | סוג | פירוט |
| 1 | תחנותעבודה | 2 מחשבים עם SSD 512MB, מערכות הפעלה MAC ווינדוס מינימום 16 GB ראם. 2 מכשירי אנדרויד-LG נקסוס 5, LG 4G. |
| 2 | בסיס נתונים | בסיס נתונים מבוסס mongoDB המאוחסן בשרתי גוגל, נותן השירות mlab. |
| 3 | כלים לניהול הבדיקות | Android studio debugger, Webstorm Jetbrains debugger. |
| 4 | כלים לביצוע הבדיקות | Android Studio,Webstorm Jetbrains, Postman, ,מכשירי פלאפון אישיים. |
| 5 | אמצעים להדמיית הסביבה האמתית | Android studio Emulator,פלאפונים. |
| 6 | ציוד/תוכנות לסביבת העבודה | מחשב עם חיבור לאינרנט (פורטים לא חסומים), פלאפון חכם עם חיבור לאינרנט. |

# גרסאות

|  |  |  |
| --- | --- | --- |
|  | שם המוצר | גרסה |
| 1 | אנדרויד | 7.0.1 |
| 2 | NODEJS | **6.9.5** |
| 3 | MONGODB | 4.7.7 |
| 4 | Passport | 0.3.2 |
| 5 | bcrypt | 2.4.3 |
|  | npm | 3.10.10 |

# טבלת בדיקות

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **תוצאה נצפית** | **תוצאה צפויה** | **מודול** | **שם בדיקה** | **מספר**  **בדיקה** |
| **תקין** | האפליקציה נטענת. | **מערכת** | זמינות מערכת. | **1** |
| **תקין** | ניווט תקין ועמודים נטענים. | **מערכת** | ניווט כללי באפליקציה. | **2** |
| **תקין** | התראה למשתמש שהערכים לא חוקיים/לא קיימים/חסרים ומתן אפשרות להכנסת ערכים מחדש. | **Login** | הכנסת ערכים לא חוקיים/לא קיימים/אי הכנסת ערכים. | **3** |
| **תקין** | המשתמש מנותב למסך הראשי, נתונים נשמרים על המכשיר להתחברות אוטומטית בכניסה הבאה. | **Login** | הכנסת ערכים תקינים ונכונים. | **4** |
| **תקין** | התראה למשתמש שהשם משתמש לא תקין ומתן אפשרות לבצע מחדש. | **Register** | הרשמה עם שם משתמש ריק. | **5** |
| **תקין** | התראה למשתמש שהסיסמא לא תקינה ומתן אפשרות להכנסת סיסמא מחדש. | **Register** | הכנסת סיסמא ריקה למשתמש תקין. | **6** |
| **תקין** | פתיחת חלון חדש לניסיון חוזר ומתן הודעה למשתמש לנסות שוב. | **Register** | בעיית תקשורת. | **7** |
| **תקין** | מתן הודעה שמשתמש כבר רשום. | **Register** | הכנסת מספר של משתמש רשום. | **8** |
| **תקין** | מתן הודעה שהרשמה בוצעה בהצלחה,ניתוב למסך הראשי והכנסת המשתמש והסיסמה לשמירה במכשיר וב DB בשרת. | **Register** | הכנסת שם משתמש וסיסמא נכונים. | **9** |
| **תקין** | המפה מוצגת למשתמש עם המיקום המדויק בו הוא נמצא. | **Map** | טעינת מפה ומיקום המשתמש. | **11** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **תקין** | המפה מוצגת למשתמש עם כל המקומות המועדפים שיש בסביבה שלו ולפי רדיוס שהגדיר. | **Map** | טעינת מקומות הבילוי על המפה לפי מיקום המשתמש. | **11** |
| **תקין** | טעינה עם הכתובת של המקום בו המשתמש לחץ. | **Map** | הצגת משתמשים במפה עם שם המשתמש. | **14** |
| **תקין** | ניתן להכניס תמונה מהגלריה או מהמצלמה. | **New place** | הוספת תמונה מגלריה או מהמצלמה. | **16** |
| **תקין** | מתן הודעה למשתמש שישנם שדות ריקים. | מערכת | חוסר באחד השדות החשובים)שם המקום,בחירת קטגוריה(. | **17** |
| **תקין** | הכנסה של המקום לDB. | **New place** | הכנסת ערכים נכונים ותקינים. | **18** |
| **תקין** | לא ניתן להכניס ערכים לא תקינים. | **Preferences** | הכנסת ערכים לא תקינים. | **19** |
| **תקין** | נתונים נשמרים על השרת ומתן הודעה שהנתונים נשמרו. | **Preferences** | ערכים תקינים-שמירה. | **21** |
| **תקין** | הרשימה מציגה את כל המשתמשים לפי המרחק שהוגדר. | **Places list** | תצוגת רשימה לפי העדפות ומרחק. | **21** |
| **תקין** | טעינה תקינה של המידע מהDB והצגה על הכרטיסיות. | **Places list** | תצוגת תוכן על הכרטיסיות. | **22** |
| **תקין** | הכרטיסייה נפתחת ומוצג המשתמש הנמצא שם. | **Places list** | לחיצה על כרטיסיה של מקום בילוי. | **23** |
| **תקין** | שם משתמש ומספר טלפון נשמר בCollection זמני כולל ספרת אימות. | **Server - Register** | קבלה של מייל ושם משתמש ופרטים נוספים. | **26** |
| **תקין** | במידה וכל הפרטים תקינים– הכנסה של המשתמש החדש לCollection של משתמשים רשומים וסיסמה מוצפנת, אחרת הודעת שגיאה. | **Server - Register** | קבלה של סיסמה עם מייל. | **27** |
| **תקין** | משתמש קיים יחזיר אובייקט עם נתוני המשתמש, נתונים שגויים יחזירו אובייקט ריק. | **Server - Login** | הכנסת מספר טלפון ומייל. | **28** |
| **תקין** | הוספה של המקום החדש לפי מדינה ועיר. במקרה של תקלה החזרה של הודעת שגיאה. | **Server –**  **Add new place** | קבלה של אובייקט עבור מקום חדש. | **29** |
| **תקין** | שמירה של התמונה על הדיסק שהשרת יושב עליו,במקרה של תקלה החזרה של אובייקט ריק. | **Server –**  **Add new place** | קבלה של תמונה עם מקום בילוי. | **31** |
| **תקין** | שמירת התמונה על הדיסק בו יושב השרת, במקרה של תקלה החזרה של אובייקט ריק. | **Server –**  **Add status** | הוספת תמונה למשתמש. | **32** |
| **תקין** | האובייקט של המשתמש ישתנה בהתאם למה שהשרת קיבל מהשתמש, במידה של תקלה החזרה של אותו אובייקט. | **Server – change**  **preferences** | שינוי הפרטית של המשתמש. | **35** |

סקירת עבודות דומות בספרות והשוואה

**אפליקציות דומות:**

**Tagg**

אפליקציית מעקב אחרי הכלב לפי רכיב ג'י פי אס המותקן בקולר הכלב, נותן התראות כשהכלב מתרחק מהאזור המוגדר, עוקב אחר הפעילות הפיזית של הכלב( ריצה, טיולים).

**MAPMYDOGWALK**

מעקב אחר התזונה של הכלב, סימון מסלול הטיול היומי תוך חישוב כמה קלוריות הורדנו

**FourSquare**

אפליקציית מפות אשר מציגה מפה באזור המגורים תוך סימון כל המקומות המתאימים לכלב – שירותי ווטרינריה, גינות, חנויות אוכל

# נספחים

# רשימת ספרות \ ביבליוגרפיה

[www.developers.google.com](http://www.developers.google.com)

[www.passportJS.org](http://www.passportJS.org)

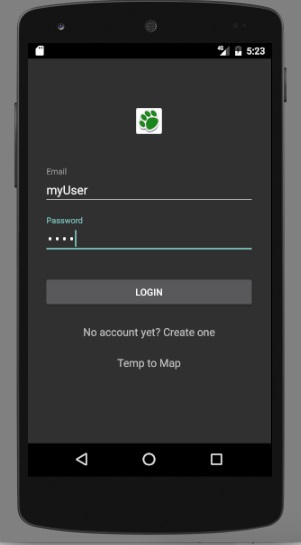
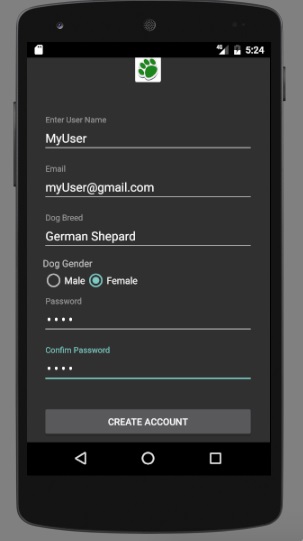
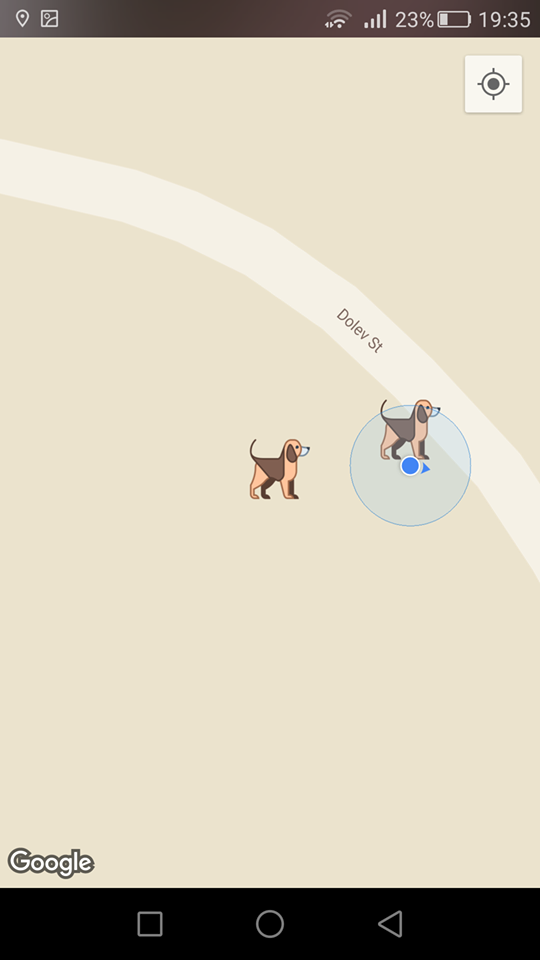
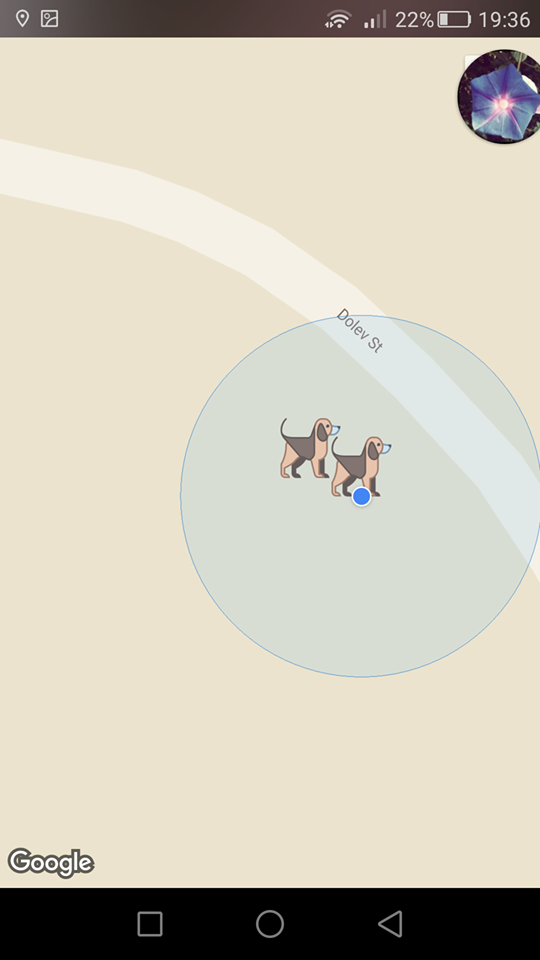
[www.mlab.com](http://www.mlab.com)

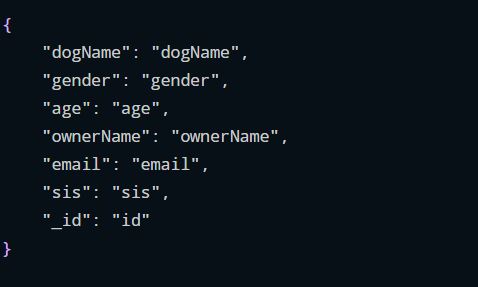
<https://developer.android.com>

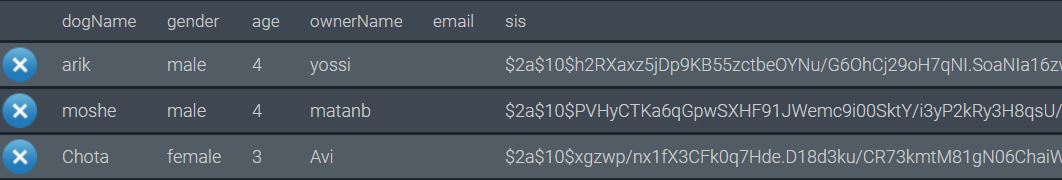
<https://www.jetbrains.com/idea/documentation>

<http://www.tutorialspoint.com/index.html>

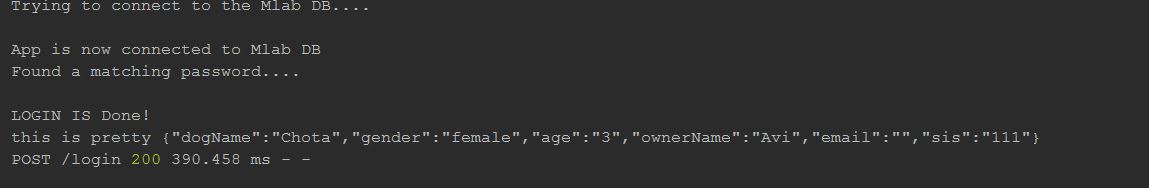
[http://stackoverflow.com](http://stackoverflow.com/)

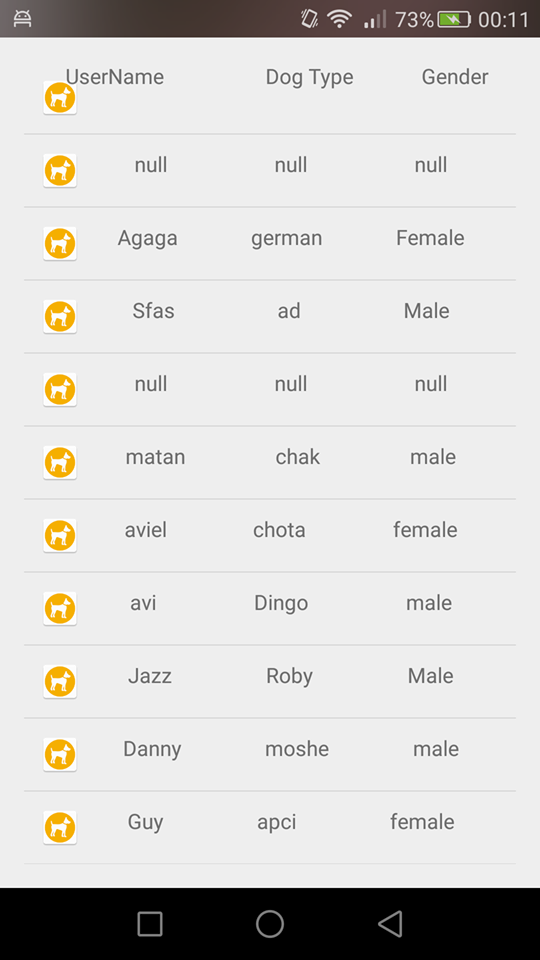
**   **



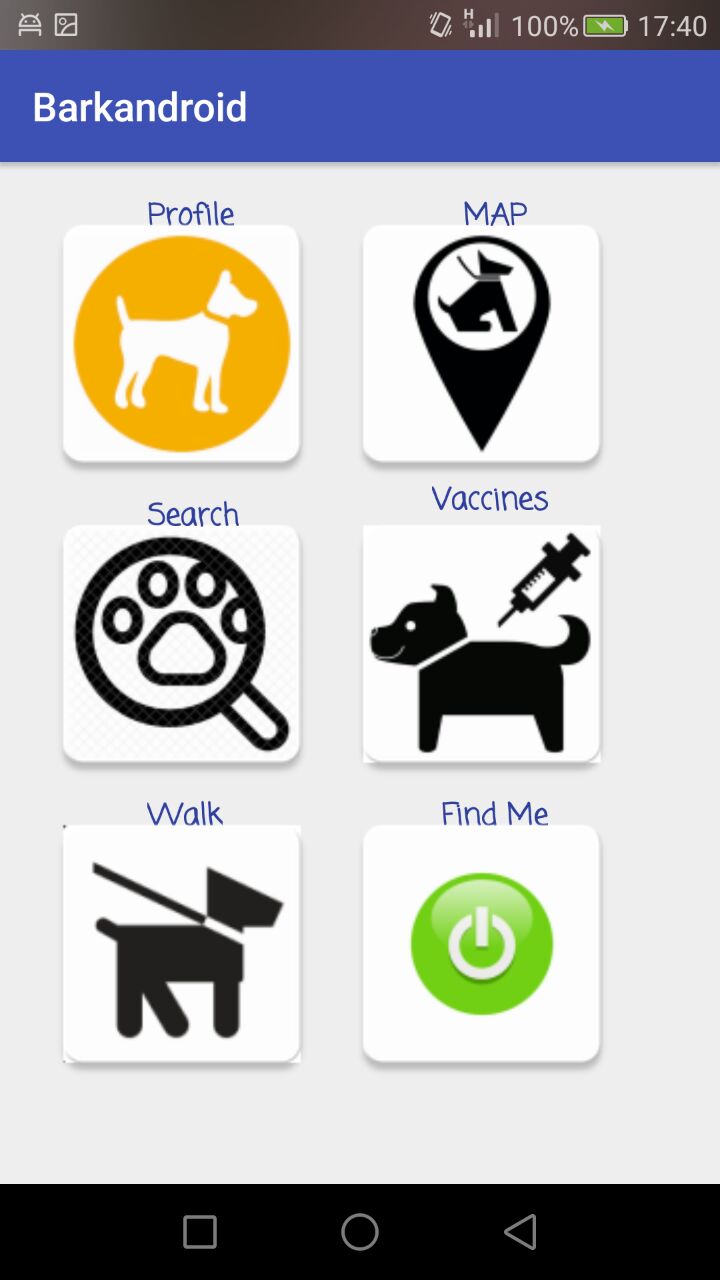
****







# C:\Users\Danny\AppData\Local\Microsoft\Windows\INetCacheContent.Word\b7ecef64-1d40-40c2-b6c3-5cb6181e306b.jpg



# סיכום/מסקנות

# 

Android מימוש הפרויקט טמן בחובו אתגרים רבים, במהלך הפרויקט למדנו רבות על פיתוח בסביבת פיתוח צד שרת בנוד ושימוש במסד נתונים מונגוJSON – שמירת הנתונים מתבצעת בעזרת אובייקט שלושה דברים שהיינו צריך ללמוד בצורה טובה.

במהלך הפיתוח הן בצד הלקוח והן בצד השרת נתקלנו בקשיים רבים עקב העובדה שלא היה לנו ידע ממשי בפיתוח מערכת מסוג זה ובכך שבחרנו סביבת עבודה חדשה.

בעזרת הקורס "פיתוח בסביבת אנדרואיד" בסמסטר א' ולמידה ממדריכים רבים בנושא השרת, האפליקציה ומסד הנתונים פיתוח המוצר החל לצבור תאוצה וניתן היה לראות תוצאות.

כמו כן תוכנית הבדיקות שערכנו תרמה לנו להבנה עמוקה יותר של המערכת ולפתרון של בעיות. בנוסף תחום הפיתוח באנדרואיד אינו פשוט כמו שהיה נדמה לפני שהתחלנו, ישנו צורך להבין את רכיבי התוכנה החיצוניים והספריות השונות, אנו בטוחים שאם היה לנו ידע מוקדם לגבי הספריות השונות והמודלים השונים, תזמון הפרוייקט יכול היה להיות טוב יותר.

בפרויקט למדנו רבות על סביבות עבודה חדשות, שרתים, תקשורת עם מסד הנתונים ופיתוח אפליקציות באנדרואיד תוך כדי התמודדות עם חיי היום יום – לימודים, עבודה חיי משפחה ולחץ יום יומי.

המסקנה העיקרית שלנו היא שפיתוח תוכנה מ-0 כולל בתוכו הרבה אתגרים מעבר לכתיבת קוד, הוא מצריך חשיבה מעמיקה יותר שכוללת תכנון נכון של המערכת, ניהול זמנים בצורה יעילה ונכונה, ניהול משימות, תוכנית בדיקות מקיפה על מנת למנוע כמה שיותר תקלות, התמודדות עם תקלות שצצות במהלך הפרוייקט וכמובן בראש ובראשונה תקשורת טובה בין העמיתים לעבודה, הרי ללא תקשורת טובה בינינו לא היינו מסוגלים להתקדם בפרוייקט זה.

.