

המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ו

אפליקציה חברתית לבעלי כלבים

Bark – Social android app for dog Owners

מאת

אסף אסל

דני וולפוב

מנחה אקדמי: ד"ר יעל נצר	אישור:	תאריך:
אחראי תעשייתי: -----	אישור:	תאריך:
רכז הפרויקטים: ד"ר ראובן יגל	אישור:	תאריך:

מערכות ניהול הפרויקט: Github

#	מערכת	מיקום
1	מאגר קוד	https://github.com/Bully/Bark-Android-app
2	יומן	https://trello.com/b/xKaw3Hmn/barkapp
3	הפצה	https://barkandroid.herokuapp.com/
4	סרטון אב-טיפוס	https://www.youtube.com/watch?v=jOKgC-VX0ak



תוכן העניינים

1	מבוא
2	תיאור הבעיה
3	תיאור הפתרון
4	תכנית בדיקות
5	סקירות עבודות דומות בספרות והשוואה
6	סיכום \ מסקנות
7	נספחים

1. מבוא

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".

2. תיאור הבעיה

הצורך ביצירת פלטפורמה 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) 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.

3. תיאור הפתרון

הצורך ביצירת פלטפורמה לבעלי כלבים הביא אותנו לרעיון של אפליקציה לטלפון חכם. רעיון האפליקציה הוא לחבר בין בעלי הכלבים בזמן הטיול על בסיס שירותי המיקום. בדומה לאפליקציות שנכנסו לחיינו בשנה האחרונה, האפליקציה תהיה מבוססת שירותי מיקום של המכשיר הסלולארי ועל בסיס השירות החינמי - Google Map. זה יבוא לידי ביטוי בכך שהרעיון המרכזי הוא שכאשר משתמש אחר יהיה באזור שרלוונטי למשתמש (פארק קרוב לדוגמא) האפליקציה תתריע על כך למשתמש ובכך ייווצר מפגש חברתי לא מתוכנן. בנוסף תתווסף לאפליקציה היכולת לעקוב אחר חיסוני הכלב ועוד. המטר היא לתת את המענה הכי טוב בכל מקום אשר נהיה בו! לצורך העבודה חילקנו תחומי אחריות: דני וולפוב – Server Side, Database. אסף אסל – Android Developer – Client side.

תיאור המערכת:

לאחר אפיון ותיכון האפליקציה הסכמנו שצד הלקוח יהיה באנדרואיד, ה-UI יעבוד ישירות עם שרת ה-NODEJS שבחרנו ע"י שליחת בקשות HTTP עם אובייקט 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 ספריית VOLLEY אשר תומכת בקבצי

ג'ייסון, שרת NODEJS היודע לפרסם ג'ייסונים ומסד נתונים MONGODB אשר מקבל אובייקטים של ג'ייסון.

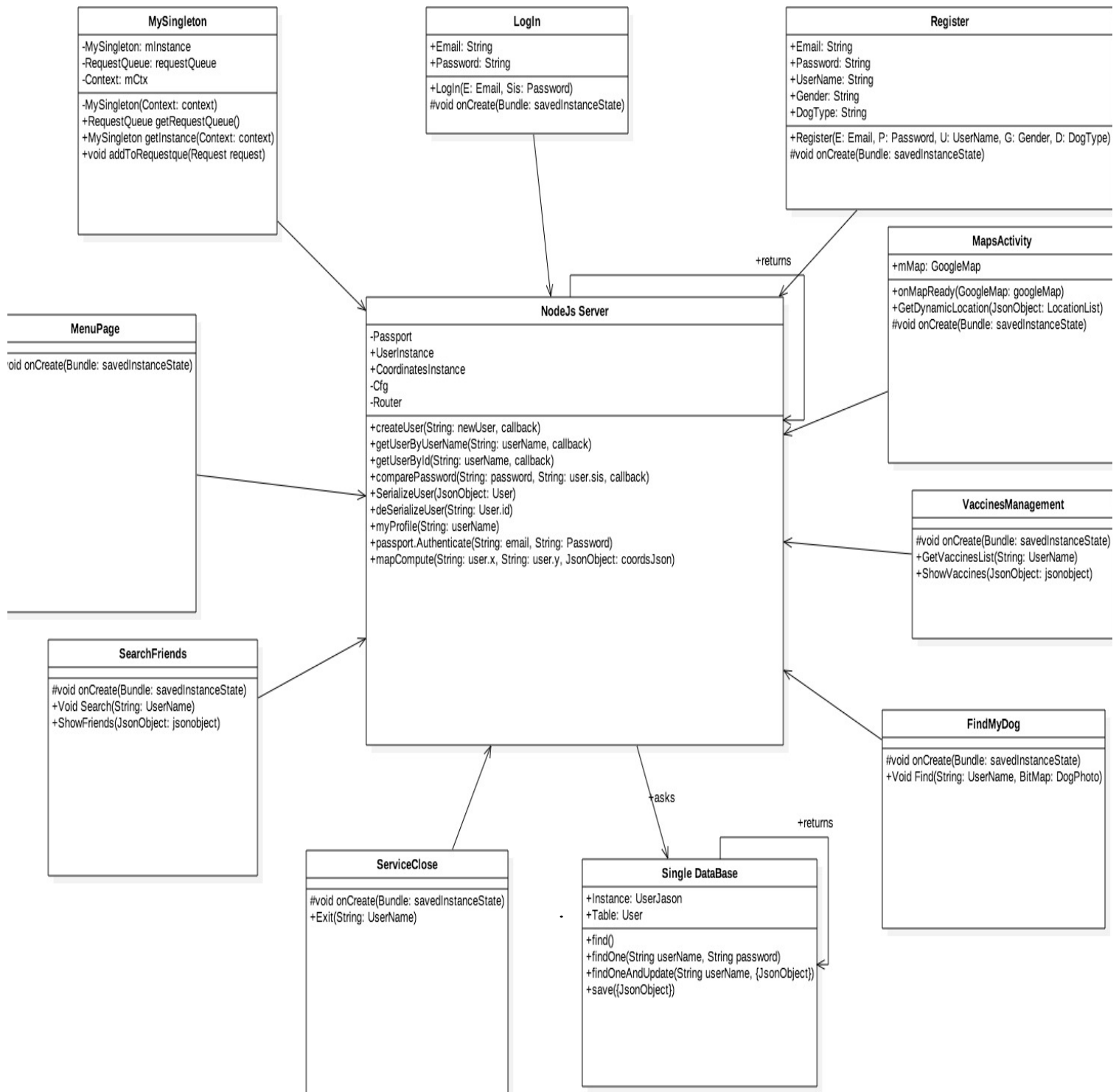
צד שרת – הבחירה הייתה בשרת מסוג "NodeJS" המאוחסן ב-HEROKU, תומך ב-REAL TIME, עובד עם ג'ייסונים, בעל פאנל שליטה וויזואלי, CLI נוח ופשוט, חינמי!

מסד נתונים – הגורם העיקרי הוא המחיר, לכן היו לנו כמה אופציות לבחירה, שרת של גוגל לדוגמא – MLAB שמספק GB500 בחינם!

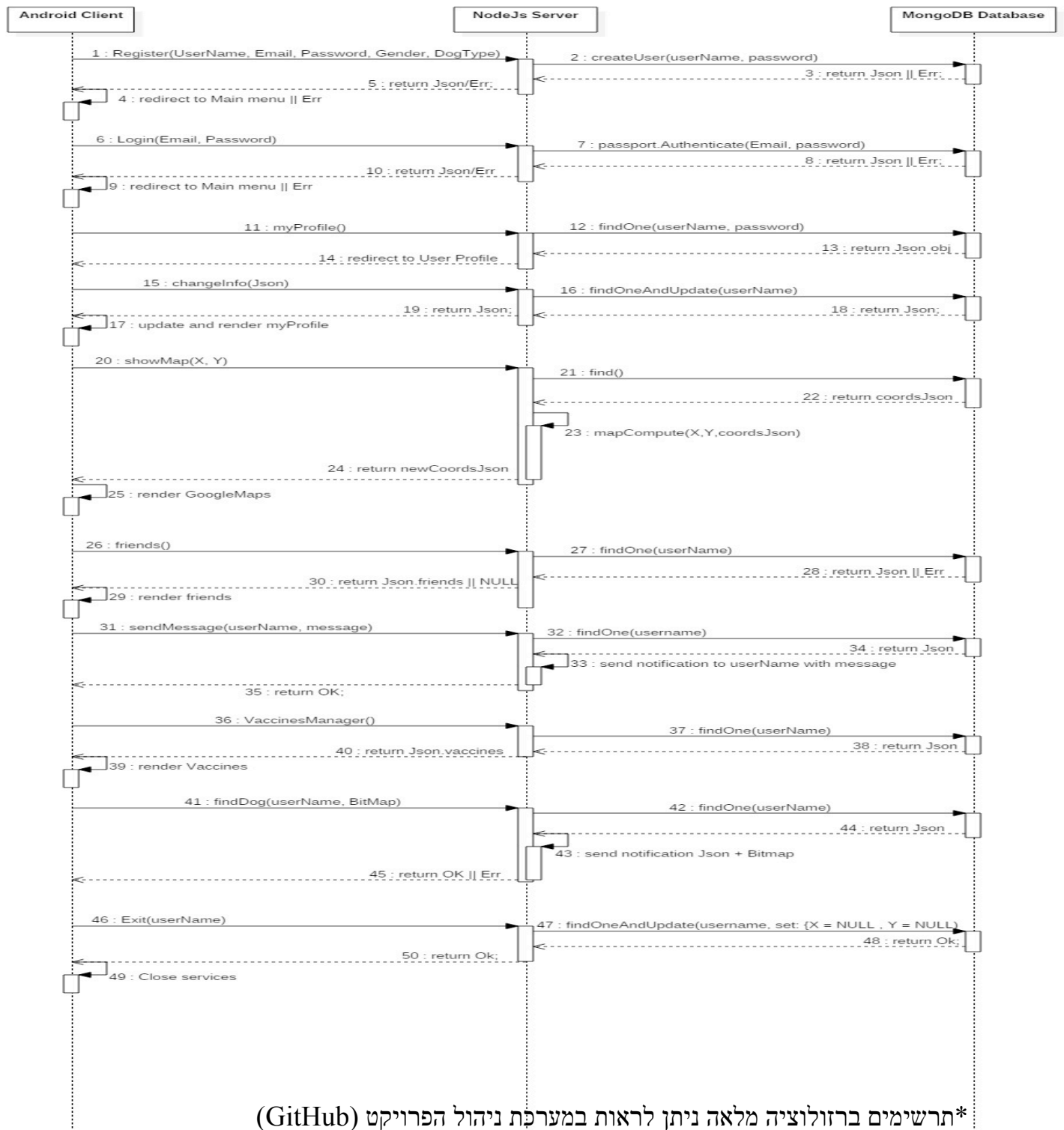
צד לקוח – בחרנו לפתח את האפליקציה לפלטפורמת האנדרואיד (בשפת JAVA), הבחירה בוצעה בפלטפורמה זו מפני שיש לנו ניסיון עבר עם אנדרואיד, ומצד שני שהדרישה שלנו היא לאפליקציה שתהיה על מכשיר סלולארי עם התממשקות לשירותי מיקום.

מהי המערכת

Class Diagram



Sequence Diagram



הכלים המשמשים לפתרון

- Android studio – עורך קוד לצד הלקוח.
- GitHub – מאגר הפרויקט.
- Google Map API – שירות המפות לאפליקציה.
- WebStrom- עורך קוד לצד שרת .
- Mlab DataBase - מסד הנתונים המסופק ע"י MLAB ומאוחסן בשרתי גוגל.
- NODEJS HEROKU - דומיין המאחסן את שרת הNODEJS שלנו.

4. תכנית בדיקות

הבדיקות בפרויקט 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 אמיתיים.

תשתית וטכנולוגיה

סוג	פירוט
1	תחנות עבודה 2 מחשבים עם MB512 SSD, מערכות הפעלה MAC ווינדוס מינימום GB 16 ראם. 2 מכשירי אנדרואיד-LG נקסוס 5, G4 LG.
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 לאינטרנט (פורטים לא חסומים), פלאפון חכם עם חיבור לאינטרנט.

גרסאות

גרסה	שם המוצר	
7.0.1	אנדרואיד	1
6.9.5	NODEJS	2
4.7.7	MONGODB	3
0.3.2	Passport	4
2.4.3	bcrypt	5
3.10.10	npm	

צד לקוח

בכל עמוד וויזואלי שיצרנו עם שדות קלט, בדקנו קבלת קלט לא טוב. ווידאנו קבלת קלט נכון ועקבנו אחרי זרימת המידע באפליקציה – קבלת מידע מהמשתמש, שמירת המידע במעבר בין דפי האפליקציה, שליחתו בפורמט הנכון לצד השרת (אובייקט ג'ייסון). בדיקת האינטגרציה עם שירותי גוגל.

בכל עמוד נבדקה תקינות הפונקציונאליות ושהכל עובד כמו שתוכנן. בנוסף בדקנו את תקינות שליחת בקשות HTTP לשרת ואופן קבלת המידע (כיצד הוא נשמר בצד הלקוח).

שרת

בדקנו קבלה ושליחה של בקשות HTTP לצד הלקוח וזמינות השרת בכל שעות היום. בדיקת החיבור לבסיס הנתונים שמירת המידע ומחירון השליפה. נבדקה קבלת בקשות HTTP תוך הקפדה על הפורמט הרצוי של המידע המועבר. בדיקת תהליך ההצפנה של הסיסמאות של כל משתמש הנרשם למערכת.

מסד נתונים

בדיקת זמינות מסד הנתונים וחיבור אליו.
נבדקה קבלה ושליחה של אובייקטים למסד הנתונים וממנו.
בדיקת נכונות הטבלאות והמידע שמוזן לתוכן.

	תיאור	חומרה 1-5	סטאטוס	תחום אחריות
1	צד קליינט לא מצליח לשלוח בקשות לשרת	5	סגורה	אסף
2	צד שרת מחזיר מידע בפורמט שגוי בעת Login	4	סגורה	דני
3	שרת לא מתחבר לDB	5	סגורה	דני
4	שרת לא זמין בשעות מסוימות	3	מושהית	דני
5	מעבר שם המשתמש מדף Login לדף הפרופיל האישי	2	חדשה	אסף
6	צד קליינט לא מציג API של גוגל	4	סגורה	אסף

5. 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

6. סיכום \ מסקנות

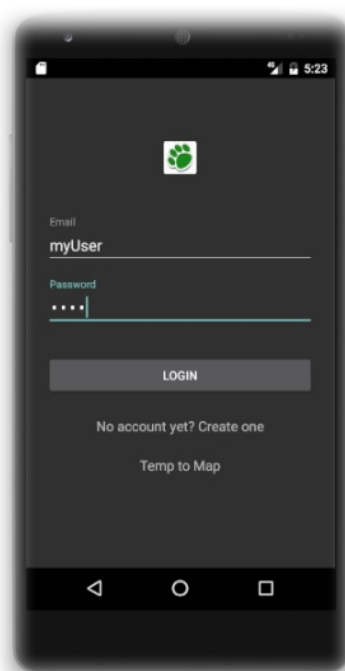
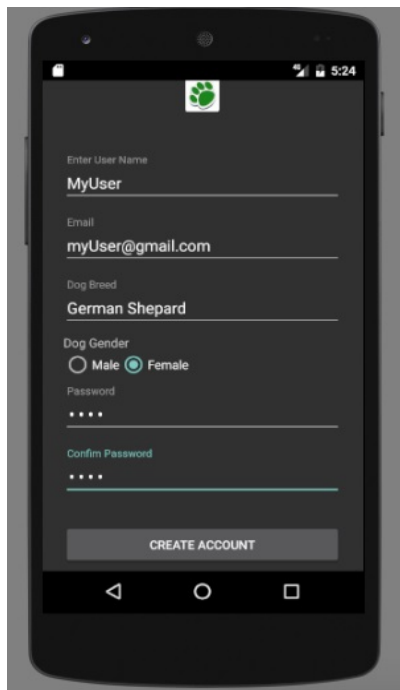
עד כה בוצע העלאת שרת מסוג NodeJS והתממשקות מולו בהצלחה. פונקציית Login / Register עובדות כרגע מול השרת והדבר מבוצע בהצלחה. בנוסף התממשקות DB אל מול השרת ושמירה / אחסון הנתונים ובנוסף גם קבלתם. מבחינת האפליקציה כרגע השירות של google map עובד בהצלחה ומציג נתונים נדרשים, כרגע המידע הוא סטטי, ובהמשך המידע יהיה דינמי ומשתנה בהתאם למשתמשים באפליקציה שיוזנו בהתאם למפה.

המסקנות שלנו כרגע לגבי העבודה מול השרת הינם - אנו נצטרך לפתח אלגוריתם מספיק חכם שימנע עומס יתר על השרת מכל משתמש שיהיה מחובר לאפליקציה (ע"י חוקים ברורים מראש ורדיוס של מיקום אשר מוגדר מראש לכל משתמש). בנוסף עלינו להשקיע מחשבה על איך לשמור את הקורדינטות (המשתנות) של כל משתמש ב-DB כדי להימנע מיצירת עומס על ה-DB ובאותו הזמן שהמידע יקושר לאובייקט של הuser הספציפי.

7. נספחים
א. רשימת ספרות \ ביבליוגרפיה

www.developers.google.com
www.passportJS.org
www.mlab.com
<https://developer.android.com>
<https://www.jetbrains.com/idea/documentation>

ב. תרשימים וטבלאות



```
{
  "dogName": "dogName",
  "gender": "gender",
  "age": "age",
  "ownerName": "ownerName",
  "email": "email",
  "sis": "sis",
  "_id": "id"
}
```

	dogName	gender	age	ownerName	email	sis
✕	arik	male	4	yossi		\$2a\$10\$h2RXaxz5jDp9KB55zctbeOYNu/G6OhCj29oH7qNI.SoaNla16z
✕	moshe	male	4	matanb		\$2a\$10\$PVHyCTKa6qGpwSXF91JWemc9i00SktY/i3yP2kRy3H8qsU/
✕	Chota	female	3	Avi		\$2a\$10\$Xgzwp/nx1fX3CFk0q7Hde.D18d3ku/CR73kmtM81gN06ChaiW

```
Trying to connect to the Mlab DB...

App is now connected to Mlab DB
Hey, new User registration...
POST /register 200 390.361 ms - -
```

```
Trying to connect to the Mlab DB...

App is now connected to Mlab DB
Found a matching password...

LOGIN IS Done!
this is pretty {"dogName":"Chota","gender":"female","age":"3","ownerName":"Avi","email":"","sis":"111"}
POST /login 200 390.458 ms - -
```

ג. תכנון הפרויקט

פגישת הכרות עם המנחה	4.10.16
לימוד עצמי	1.12.16- 1.11.16
חיבור השרת לצד הלקוח	5.12.16
העלאת קבצי השרת לדומיין	8.12.16
התחלת יצירת UI	15.12.16
חיבור השרת לDB	15.12.16
מתן פונקציונליות לרכיבי האפליקציה	1.1.17-1.2.17
בדיקות חלקיות לאב טיפוס	1.2.17-8.2.17
פגישה שנייה עם המנחה – התקדמות, שאלות.	7.2.17
הגשת אב טיפוס	12.2.17
אלגוריתם מפות	12.2.17-5.3.17
מערכת הודעות	5.3.17
מערכת חיסונים	15.3.17
סיום צד שרת ולקוח	22.3.17
בדיקות	23.3.17

ד. רשימת/טבלת דרישות וסיכונים

תאריך אחרון לסיום	סבירות (1-5)	חומרה (1-5)	סיכונים	זמן עבודה מוערך בשבועות	תיאור השלב
11.12.16	1	4	אפיון לא מתאים לפרויקט, תיכון לא טוב.	2	אפיון תיכון ועיצוב תבניות תיכון למבנה הפרויקט
25.12.16	2	5	בעיות בשאילתות , מידע לא נשמר\נאבד, שרת איטי.	2	בחירת מסד נתונים, התחברות לשרת
12.2.17	2	3	לא נצליח לבצע אינטגרציה בין הפיצ'רים השונים.	4	בניית אב טיפוס
5.2.17	1	5	אלג' לא טוב.	3	תכנון ובניית האלגוריתם למציאת אנשים
25.2.17	2	4	אי קבלת הודעות.	3	מערכת הודעות
19.3.17	1	2	זמן רב ללמוד את ה API.	3	אינטגרציה של המפות עם האפליקציה
2.4.17	3	3	באגים שלא נצפו מראש.	3	בקרה ובדיקה - Debug
20.5.17	3	4	אי עמידה בזמנים.	4	כללי – אינטגרציה סופית ותיקון במידת הצורך.
				Total: 24 weeks	