

## I.N.S – Indoors Navigation System

### Application document

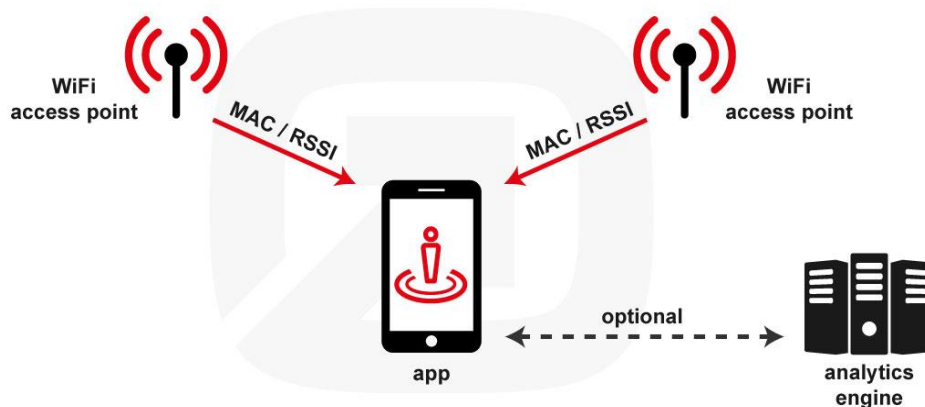
#### תוכן עיניינים

1	מבוא
2	מבנה האפליקציה
3	מסך ראשי
3	מעקב אחר ה-WiFi Daemon
4	Pinpoint
4	אלגוריתם ההתמצאות
5	אלגוריתם הניווט
5	מבנה ה-Database
5	טבלת BSSID
6	טבלת locations

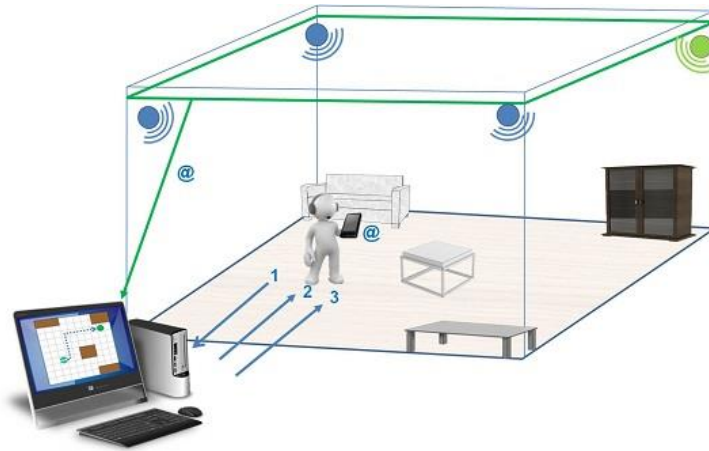
#### מבוא

Indoors Navigation System הינה אפליקציה המאפשרת לכל משתמש לנווט במבנים סגורים, המערכת מאפשרת לכל משתמש למפות מבנים, לציין נקודות על המפה ולנווט במבנה סגור (גם כשאין קליטת GPS).

הרעיון פשוט, כל משתמש יכול לקבוע נקודת ציון בה הוא נמצא כעת, המשתמש יכול להסוּף תאור ותמונה לנקודת ציון. בעת קביעת נקודת ציון המערכת אוספת נתונים אודות שירותי ה-WiFi הנקלטים ע"י המכשיר, המערכת יוצרת קבוצה של MAC address (BSSID) ושל עצמת ה-signal (RSSI). אוסף הנקודות המאפיין נ"צ נקרא fingerprint.



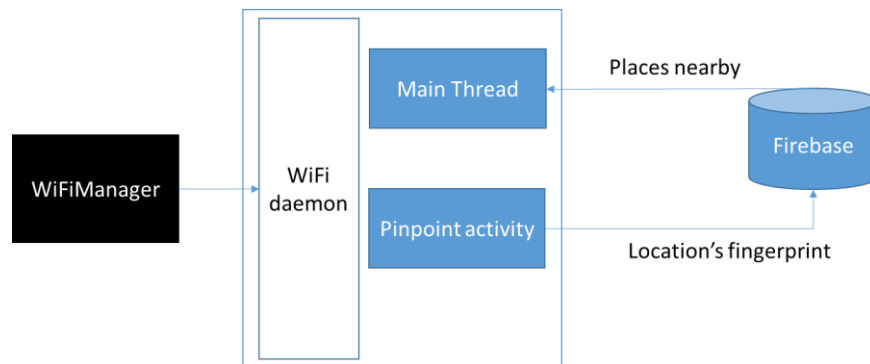
כאשר האפליקציה פועלת היא מחפשת רשתות WiFi ומנסה למצוא את המקום הקרוב ביותר מתוך מאגר ה-fingerprints הקיים. המאגר משותף לכל המשתמשים ולכן מספיק אדם אחד שימפה בניין שלם עבור קבוצת משתמשים.



## מבנה האפליקציה

האפליקציה בנויה משלושה חלקים מרכזיים:

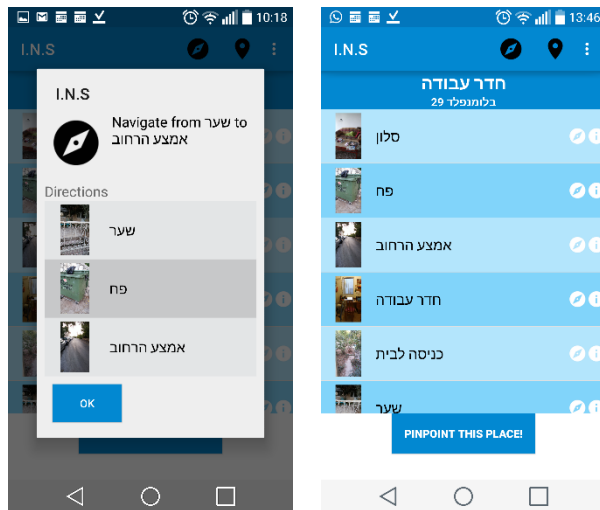
1. **WiFi daemon** – זהו Thread הפועל לאורך כל חיי האפליקציה, תהליך זה באופן שיטתי מפעיל סריקה מחודשת של ה-Wifi (מפעיל את ה-WifiManager) מעבד את התשובות ומעביר אותם לאפליקציה Main Activity.
2. **Firestore DB** – מסד הנתונים המרכזי של המערכת.
3. **Main Activity** – התהליך המרכזי של האפליקציה, בהתאם לנתונים מה-DB ומה-Wifi Daemon קובע את המקום הנוכחי של המשתמש.



ה-Main Thread הוא החלק היחיד החשוף למשתמש, הוא מציג את המקום הנוכחי של ההמתנתמש. בכל רגע נתון המשתמש יכול לבצע מספר אפשרויות: קביעת נ"צ חדש (pinpoint), ניווט בין מקומות סמוכים ומעקב אחר רשתות WiFi פעילות:

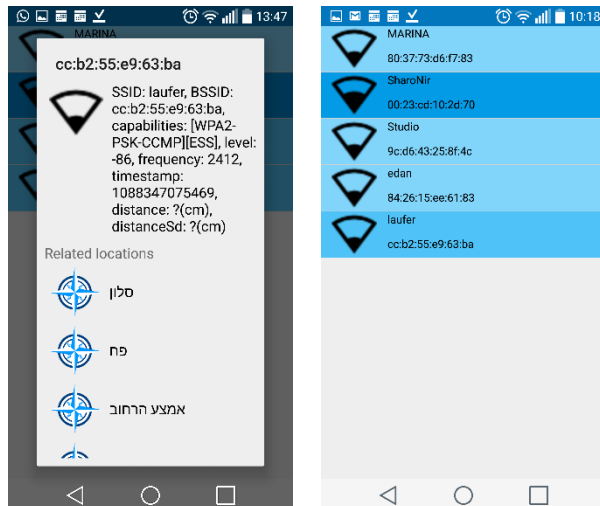
## מסך ראשי

במסך הראשי ניתן לראות את המקום הנוכחי בו אני נמצא כעת. בנוסף נקבל רשימה של מקומות סמוכים. לחיצה על Info תציג פרטים על המקום הנבחר. לחיצה על "ניווט" תציג מסלול של כל התחנות שעלינו לעבור מנקודה A ועד לנקודה B.

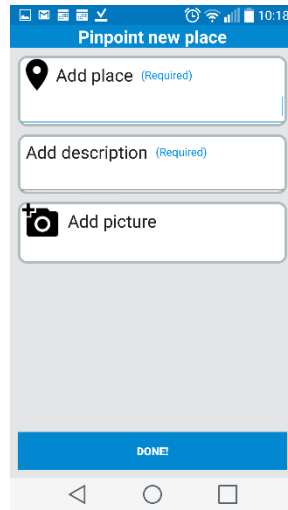


## מעקב אחר ה-WiFi Daemon

במסך זה אפשר לנטר בזמן אמת אחר פעולת ה-WiFi Daemon ולראות את כל רשתות ה-WiFi הקרובות. לכל מכשיר נוכל לראות את מספר ה-BSSID שלו ואת ה-RSSI (עצמת ה-signal). לחיצה על רשת כלשהי תציג לנו את כל הני"צ בהם הרשת הני"ל היא חלק מה-fingerprint.



## Pinpoint



מסך זה מאפשר לבצע Pin point – הזנת נקודת ציון חדשה. המערכת באופן אוטומטי יוצרת fingerprint לנקודה. המשתמש חייב לציין את שם המקום ולספק תאור קצר. המשתמש יכול להוסיף תמונה של המקום (אופציונאלי). לחיצה על "Done!" תעדכן את הנייץ החדש ב-DB.

## אלגוריתם ההתמצאות

התמצאות היא היכולת של האפליקציה לדעת איפה היא נמצאת ברגע נתון.

אלגוריתם ההתמצאות מורכב מ-2 יוריסטקות:

1. היסטוגרמת מקומות
2. חישוב מרחקים

נסמן fingerprint באופן הבא:  $F = \{(bssid_1, rssi_1), \dots, (bssid_n, rssi_n)\}$  כל fingerprint הוא קבוצה של זוגות, BSSID (הMAC הייחודי של מכשיר) וה-RSSI (עוצמת ה-signal).

נסמן ב- $f'$  את ה-fingerprint בזמן  $t$  (למעשה  $f'$  הוא ה-fingerprint שהאפליקציה חישה ברגע זה).

בנוסף נסמן ב- $BSSID_i$  את כל ה-Fingerprints בהם משתתף ה-bssid שכתובתו  $i$ :

$$BSSID_i = \{F_1, F_2, \dots, F_n \mid i \in F_i\}$$

**היוריסטיקה הראשונה** הינה חישוב היסטוגרמת Fingerprints באופן הבא: לכל  $bssid \in f'$  נספור בכמה Fingerprints מופיע.

$$\forall F_{i_{score}} = \sum_{bssid \in f'} \begin{cases} 1 & F_i \in BSSID_{bssid} \\ 0 & F_i \notin BSSID_{bssid} \end{cases}$$

נאסוף את כל ה-Fingerprints עם הציון הגבוה ביותר (הקבוצה  $best - locations$ ):

$$best - location = \max(F_{1_{score}}, F_{2_{score}}, \dots, F_{n_{score}})$$

נשים לב כי היסטוגרמה שכזו שקולה לחיתוכים של קבוצות, דוגמא: ידוע לנו כי  $BSSID_1$  מופיע במקומות א,ב,ג. לעומת זאת  $BSSID_2$  מופיע במקומות ב ו-ג ובנוסף  $BSSID_3$  מופיע במקומות א,ב.

בדגימה מקבלים כי  $f' = \{BSSID_1, BSSID_2, BSSID_3\}$  ונקבל את ההיסטוגרמה הבאה:

א	2
ב	3
ג	2

לכן בסבירות גבוהה אנחנו נמצאים במקום ב' (הרי  $\{b\} = best - locations$ ).

**יוריסטיקה שנייה** הינה חישוב של סכום הפרשי המרחקים בין  $f'$  לבין כל המקומות המופיעים ב- $best - place$ . כלומר:

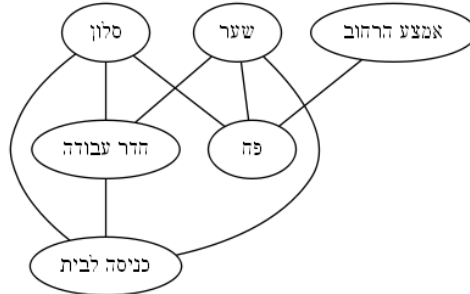
$$\forall F \in \text{best}_{\text{locations}} F_{\text{score}} = \sum_{rssi \in F} |rssi - f'_{rssi}|$$

היוריסטיקה הנ"ל נותנת ציון המביע מרחק בין המרחק הנוכחי שלי ובין הנ"צ. לכן נבחר את הנקודה עם הציון הנמוך ביותר. הנקודה שנבחר היא תבחר להיות המיקום הנוכחי של המשתמש.

## אלגוריתם הניווט

אלגוריתם הניווט מבוסס על הערכה של מקומות סמוכים. במידה והמערכת שמה לב לשינוי במיקום המשתמש (דוגמא: מעבר ממיקום א' למיקום ב') המערכת תזהה שיש קשר גאוגרפי בין א' ל-ב' ומבחינת המערכת קיים מסלול בין א' ל-ב'. מידע זה נשמר ב-Database ומתעדכן עבור כל המשתמשים! מספיק שמשתמש אחד יטייל בין מספר נקודות ציון שונות כדי ליצור מסלול.

האפליקציה מסוגלת לחשב גרף המקשר בין נ"צ שונים (באופן ויזואלי):



Wifi relations map  
Nir Moshe

לאחר חישוב הגרף, המערכת מבצעת BFS בין המקור ליעד. במידה והמערכת מצליחה היא מציגה מסלול למשתמש.

## מבנה ה-Database

מאגר הנתונים של המערכת מבוסס על מערכת Firebase. למערכת 2 "טבלאות" (טבלאות מסוג Key-value ולא טבלאות רלציוניות).

### טבלת BSSID

טבלת BSSID מכילה את כל ה-BSSID שנצפו ע"י המערכת, ולכל BSSI נציין באיזה locations מופיע ובאיזה עוצמה (כפי שניתן לראות בתמונה).

bssid	
00:23:cd:10:2d:70	
10:be:f5:36:5b:70	
10:be:f5:39:97:0e	
14:ae:db:3c:75:1a	
60:e3:27:ff:8b:d2	
6c:19:8f:48:e6:ee	
7c:b7:33:2b:af:21	
80:37:73:d6:f7:83	
1474814000653	חדר עבודה -83
1474814034747	סלון -76
1474814070061	כניסה לבית -79
1474815612251	שער -83
1474815664147	פה -80
1474815700357	אמצע הרחוב -80
84:26:15:ee:61:83	
94:57:a5:4a:d4:78	
ac:f1:df:9f:28:09	
c0:ac:54:f8:e4:51	
c0:ff:d4:7e:16:7a	
c4:12:f5:3f:d9:40	
c4:3d:c7:4c:a1:92	
cc:b2:55:e9:63:ba	

## טבלת locations

טבלת Locations מכילה את כל הפרטים אודות נ"צ:



1. שם
2. תאור
3. תמונה (אופציונאלי) מקודדת ב-Base64
4. Points
5. Near (מקומות קרובים)
  - a. רשימה של כל ה-BSSI והעצמה שלהם המאפיינים את הנ"צ.
  - a. רשימה של כל המקומות הסמוכים לנ"צ. רשימה זו מתעדכנת תוך כדי תנועת המתמש, זו הדרך בה המערכת קובעת קשר בין מקומות.