**המחלקה להנדסת תוכנה**

**פרויקט גמר – תשע"ו**

**אפליקציית רשתות**

**Wireless Application**

**מאת**

**ליאור ספיר**

**מנחה אקדמי: דר' מרים אללוף אישור: תאריך:**

**רכז הפרויקטים: ד"ר ראובן יגל אישור: תאריך:**

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | github.com/liorsap1/ SW-Final-project-JCE- |
| 2 | יומן | https://www.targetprocess.com/ |
| 3 | ניהול פרויקט (אם בשימוש) |  |
| 4 | הפצה |  |
| 5 | סרטון אב-טיפוס |  |

תוכן העניינים

* מילון מונחים, סימנים וקיצורים

**תקציר**

* במסמך זה אתאר את פרויקט הגמר שלי בתואר הראשון להנדסת תוכנה, ובמסגרת הפרויקט אצור מוצר תוכנתי (אפליקציה) המספק מענה לניידות המידע והגרת הנתונים בתחום הנתבים (הסבר מהו נתב בהמשך) הביתיים.
* המוצר המפותח בשלבי הפרויקט הוא אפליקציית (יישום תוכנתי) היושבת על גבי מערכות הפעלה מבוססות גרסאות Android, האפליקציה מקודדת בשפת התכנות JAVA ומפותחת בסביבת Windows 10. האפליקציה (יישום) המפותחת בפלטפורמת Android Studio בגרסה האחרונה הזמינה באתר הראשי לפלטפורמה (לינקים יצורפו בהמשך). האפליקציה הינה (בשלבי פיתוח לכתיבת מסמך זה),היא תהיה מסוגלת לבצע משיכת נתונים מרשת אלחוטית המחוברת לנתב בייתי ובהמשך תתאפיין ביכולת דחיפת נתונים וביצוע קונפיגורציה לנתב הביתי.
* הפרויקט יתנהל תחת המנחה דר' מרים אללוף ואופי הפרויקט יהיה כמודל הדומה לפרויקטים מתוכננים היטב ואינם זריזים כדוגמת Agile. בפרויקט ישתלבו שתי חלקי פיתוח לאפליקציה. בשלב הראשון אעבוד על שאיבת נתונים מן הנתב. בשלב השני אעבוד להצלחת ביצוע קונפיגורציה להגדרות הנתב. עבור הפרויקט הציוד הנדרש הינו מס' נתבים ביתיים, מחשב, תוכנת Android Studio, כבל עם מחבר אנדרואיד וטלפון חכם מבוסס מערכת ההפעלה Android.

# **מבוא**

הנתב (Router) הוא אחד מהרכיבים המרכזי מאוד בתעבורת המידע באינטרנט . נתב (ROUTER) - הוא רכיב תקשורת מחשבים שנועד לקביעת נתיבן והפצתן של חבילות נתונים ברשתות תקשורת נתונים. רכיב שהוא התקן חומרה המאפשר לשתף חיבור אינטרנט אחד בין המחשבים ברשת התקשורת. משימת הניתוב מתבצעת בשכבה השלישית של מודל השכבות (OSI) ומשתמשת בכתובות IP.

נבין בקצרה להעשרה, את עיקר פעולת הנתב, צעד לפני שנרצה לגשת אל נתוניו - הנתב מקבל חבילת מידע (packet), המגיעה אליו מרכיב תקשורת אחר ועל פי יעד החבילה הנתב ישלח את חבילת המידע להמשך מסלולה אם במידה ולא עשתה מספר צעדים יותר ממה שמוסכם. כל רכיב תקשורת שבו עוברת החבילה נחשב צעד נוסף. במידה והנתב קורא כי כמה מסלולים אפשריים ליעד את החבילה, הנתב יבחר בנתיב אליו החבילה תצעד את מספר הצעדים הקטן ביותר. בנוסף למספר הצעדים, הנתב מתחשב בנתונים רבים המשפיעים על החלטות ניתוב חבילות המידע.

מעבר לתפקידו של הנתב כגשר להעברת חבילות מידע, הנתב מתוכנן ויודע גם לספק שירותים נוספים. הנתב ממשמש כדלת כניסה ויציאה לרשת מקומית – ביתית לאינטרנט. כל המחשבים המחוברים לרשת זו יכולים לתקשר בינם לבין עצמם ובינם ובין הנתב ללא כל צורך ביציאה לאינטרנט. עדיין יהיה ניתן לשתף ולהעביר קבצים בין מחשבים שונים המחוברים לאותו נתב.

שירות נוסף אותו מספק הנתב הוא בתחום אבטחת המחשבים המחוברים לרשת המקומית. כאשר מגיעה לנתב חבילת מידע מכוון האינטרנט לרשת המקומית, הנתב בודק בטבלאות האם החבילה מגיעה מיעד כשיר ואמין ובהתאם לכך פועלת לנתב את החבילה ליעדה או משליכה אותה. דהיינו הנתב יוצר "חומת אש" (firewall) שמסננת ברמה מתקדמת את תעבורת חבילות המידע ברשת בהתאמה לרשימה שחורה אשר הנתב מחזיק. יש הרבה סוגי נתבים, בפרויקט שלי, אתעסק בעיקר בשני סוגי נתבים: ביתיים וארגוניים.



WIFI

רשת אלחוטית במרחב מקומי.

ה – WIFI או WI-FI היא רשת אלחוטית המאפשרת למכשירים אלקטרונים להעביר נתונים באופן אלחוטי (ברשת אלחוטית) באמצעות גלי רדיו. רשתות אלה מבוססות לרוב על תקני IEEE 802.11 ו-WI-FI. תקנים אלו מאפשרים פריסת רשת שבה מועברות חבילות IP בין רכיבי תקשורת שונים למרחקים של כמה עשרות או מאות מטרים תלוי בעוצמת השידור. תחום התדרים וכמות העברת הנתונים בצורה אלחוטית בנתב נקבעת לפי הקונפיגורציה שנותן המשתמש. כפי שצוין כבר לפני כי כוחו של הנתב ליצור תשדורת אלחוטית – כוחו של ה WIFI הוא בפשטות הפריסה של הרשת האלחוטית. אין צורך ברכיבים רבים, כבלים ופרוטוקולים מורכבים שיושבים על המחשבים. ניתן להפעיל את רשת ה – WIFI בשני מצבים:

AD-HOC – הרשת אינה מאורגנת וכי אין בקר ראשי המנהל את טופולוגית הרשת. רכיבי הרשת מתקשרים בינם לבין עצמם וכולן ממשות את אותו פרוטוקול תקשורת לפי הסטנדרט המתאים ל-WIFI.

INFRASTRUCTURE – תשתית רשת המנוהלת על ידי נקודת גישה או יותר (מדובר בנתבים) בה או בהם עוברות תעבורת חבילות המידע ומנותבות ליעדן המתאים. נזכיר כי האפליקציה המתוכננת אמורה לשאוב מידע מנקודת גישה אחת ולנתח את המידע עבור אותה נקודת גישה.

* כיצד נכנס הרכיב האמור לתמוך בשאיבת הנתונים מאותו נתב\נקודת גישה? הרי האפליקציה המתוכננת אמורה לתמוך ביעדי הפרויקט. בפרויקט זה אפתח אפליקציה על פלטפורמה של Android Studio ובסביבה של רכיב המופעל במערכת הפעלה של Android.

Android – מערכת הפעלה (Operating System), המיועדת לרכיבים אלקטרונים המבוססים יל ליבת לינוקס. בין הרכיבים נמצאים – טלפונים חכמים, טלוויזיות חכמות, טאבלטים, שעונים, מחשבים וכולי.. אנדרואיד הינה המערכת ההפעלה הנפוצה בעולם ותופסת נתח של כ70 אחוזים מכלל שוק הטלפונים החכמים ונמכרים מיליוני טלפונים מבוססי אנדרואיד ברחבי העולם. גרסת המערכת הראשונה פותחה בשנת 2003 על ידי חברת “Android” Inc. ונרכשה על ידי חברת google העולמית תוך זמן קצר. מאז פותחו גרסאות רבות למערכת ההפעלה וחברות גדולות כגון Samsung, HTC, LG ועוד. כיום לכתיבת הפרויקט, הגריסה הכי עדכנית הקיימת בשוק היא Android 7.0.

פיתוח האנדרואיד וגרסאותיו לרוב נעשות על ידי עובדי GOOGLE

ויצרני רכיבים העובדים עם אותה מערכת יכולים להוסיף מאפיינים

משלהם לאותה גרסה. הפיתוח על גבי מערכת ההפעלה אנדרואיד

נעשה על גבי הרבה אפשרויות ופלטפורמות רבות. בפרויקט זה כמו

שהוזכר, פיתוח האפליקציה לטלפון חכם יעשה על גבי Android Studio. לרוב פיתוחי קוד האפליקציות שרצות על גבי אנדרואיד, נכתבות בשפת JAVA ולכל אפליקציה ניתן להתאים SDK המתאים לפיתוח האפליקציה. ניתן לכתוב גם בשפת תכנות אחרת כמו C, C++ או C# ועוד. בהינתן תכנון קוד בשפה אחרת, יש להתקין תוסף הנקרא NDK. הסבר להבהרה על SDK.

SDK – (Software Development Kit), ערכת פיתוח תוכנה שהיא מאגדת סט כלים (פונקציות או הרשאות) לפיתוח יישום ולהעלות את יכולתו לעבוד במספר רב יותר של פלטפורמות, סביבות עבודה שונות ועל רכיבים שונים. SDK יכול להיות ספריה של ממשק לתכנות יישומים, יכול להכיל רישיון לבניית תוכנה מסוימת או לכלול חומרה לעבוד בתקשורת עם תוכנה מסוימת.

בפרויקט זה אתמקד בניתוח והתעסקות של מידע עבור נתבים ביתיים דרך אפליקציה.

* הכוח החזק הטמון בנתב הוא היכולת לשדר חבילות מידע אל רשת אלחוטית. לאותה רשת אלחוטית (WIFI), בה משדר הנתב.

**האפליקציה שלי**

אפתח אפליקציה הניגשת לנתב מרחוק המסוגלת לקבל נתונים בזמן אמת על מצב הנתב ועל מדדיו.

מה אני מצפה מהאפליקציה? על האפליקציה להביא שינוי ראשוני בניידות יכולת שאיבת והגדרת נתוני הנתב מרחוק. מה יש היום? היום ניתן להתחבר לנתב דרך דף אינטרנט המחובר לנתב. האפליקציה עונה על נושא הניידות בחיבור לנתב וגישה מרחוק, בצורה אלגנטית ונעימה יותר.

**הצורך באפליקציה**

כאשר משתמש ירצה להתחבר לנתב (במידה והוא לא גורם זר), לגלות נתונים עליו וגם להגדיר במידת הצורך את מאפייניו, יהיה נותר לו לעשות זאת בצורה הכי נוחה שיש ובכמה שפחות פעולות המצריכות מהמשתמש להדליק את המחשב, להתחבר לנתב ולהזדהות מולו

(לבצע אימות עם שם משתמש וסיסמא).

לאחר פיתוח האפליקציה, כל שיהיה על המשתמש לעשות הוא לפתוח את אותה אפליקציה ולהזדהות רק בפעם הראשונה מול הנתב. כיום רוב האנשים בעולם משתמשים בדגמי הטלפון הסלולרי החכמים שיכולים להריץ אפליקציה כזאת שאני מתאר בפרויקט הנ"ל.

# **תיאור הבעיה**

.מחשבי לוח ()במהלך העשייה, הבנתי כי נכון לבצע הפרדה לשלבים בביצוע העבודה (הטכנית). כאן:

בעקבות השלבים המתוכננים, עולות בעיות בכמה תחומים: הנדסת תוכנה, הנדסת אנוש ואתגרים יצירתיים.

* בעיית הניידות וגישה מרחוק לנתב המוצגת בתחילת המאמר, מעוררת את בעיה המרכזית הבאה: עת לכתיבת דו"ח זה, אין אפליקציית אנדרואיד שיודעת לבצע את הדרוש בפרויקט עבור היישום שברצוני לפתח. אפליקציה כזו, תביא את נושא הניידות לנתב, בצורה משמעותית מאוד והאפליקציה שתפותח, תהייה החלוצה בהבאת פתרון לנושא. כאן אצטרך לחשוב על דרכים יצירתיות וחכמות להתמודד עם הגדרת תכני הנתב, ביצוע קוד נכון מבחינת הנדסית ואנושית כך שיוכל לתת מענה לבעיה.

## **דרישות ואפיון הבעיה**

לאחר חשיבה על אופי האפליקציה וכיצד אמקסם את הנוחות המקסימלית למשתמש, על מנת שיוכל לקבל את כל נתוני הנתב מופרדים ומוצגים בצורה קלה להבנה ולחולל שינויים בהגדרות הנתב, הגעתי להחלטה כי אלו המסכים העיקריים שבהם האפליקציה תציג את ממשקה.

אפליקציית הנתב תכיל 4 מסכים:

* מסך ראשי, שהוא גם המסך ההתחלתי והראשוני כאשר מפעילים את האפליקציה (Launcher).
* מסך נתונים, המציג את נתוני הנתב (לאחר חיבור WIFI עם המכשיר).
* מסך טווח, המציג את חוזקת החיבור האלחוטי עם הנתב ויוצג מד המראה את חוזקת חיבור בהתאם למרחב הפיזי מהנתב.
* מסך Login, מסך המציג שדות אימות הנועדות לספק אמצעי אימות מול הנתב הביתי, על מנת לקבל את דף הגדרות הנתב.
* מסך קונפיגורציה, המראה את אפשרויות התאמת הגדרות חדשות לנתב הביתי.
* כעת לשלב האב טיפוס, יכולים להיות שינויים עתידיים בממשק המשתמש של האפליקציה. המסכים אשר מוצגים כאן במסמך, הם באמת אב-טיפוס למסכים העתידיים שאמורים להיות באפליקציה. אלו הם מסכים לדוגמה וקיימים בשלב זה באפליקציה.

****

**מסך ראשי:**

במשך זה יופיעו נכון לעכשיו שלושה כפתורים בסיסיים שמטרתן

להעביר את המשתמש למסכים יעודים שם יקבל את השירות המופיע

על הכפתור במסך הכניסה.

שלושת הכפתורים במסך הכניסה:

Press for indices – קבלת מדדים על טווח קליטת הרשת.

Press to get info – קבלת הנתונים על הנתב עצמו כפי שמפורט

במסך הנתונים.

Configure Router – מעביר אותי למסך אימות מול הנתב הביתי

על מנת לגשת להגדרות של הנתב.

**מסך נתונים:**

אפליקציית הנתב תהייה בעלת יכולת להציג את נתוני הנתב המחובר למכשיר

ממנו פועלת האפליקציה.על האפליקציה להציג את הנתונים האלו:

SSID – שם הנתב המוצג למשתמשים ברשת ולאלו הרוצים ליצור חיבור

אלחוטי עם הנתב.

Frequency – התדר שבו עובד הנתב.

Ip Address – כתובת IP של הנתב אל מול הספק.

Mac Address – כתובת הפיזית (MAC) של המכשיר המחובר לנתב.

BSSID – כתובת הפיזית (MAC) של הנתב.

Link Speed – מהירות החיבור אל הנתב.

****

**מסך טווח:**

במסך זה יוצג בהמשך מד, אשר מראה את טווח חוזקת

הקליטה של הנתב אל מול המכשיר המחובר.

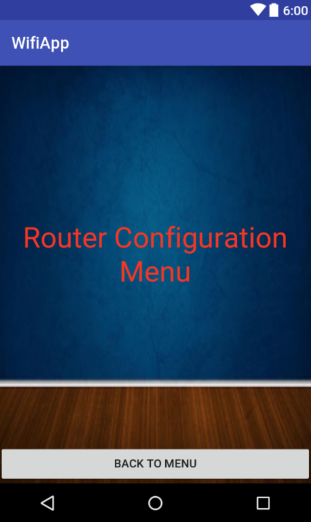
המטרה ליישם מסך כזה באפליקציה היא לנתח את חוזקת הקליטה

של הנתב ברחבי הבית או המשרד. כתוצאה מהמידע אפשרי להסיק

מהו המקום הטוב ביותר לשים את הנתב בהתחשב בטווח הקליטה.

**מסך Login:**

מסך זה עוד לא מוכן בשלב זה. התכנון העתידי עבור המסך הוא לאפשר אימות והזדהות מול הנתב על מנת לקבל את כל הנתונים הניתנים להגדיר מחדש בנתב.

**מסך קונפיגורציה:**

בדף זה תפתח רשימה של כל האופציות והמאפיינים הניתנים

להגדרה מחדש על ידי האפליקציה.

לאחר סיום ההגדרות, יופיע חלון המאשר כי ההגדרות

אכן התבצעו ונטמעו בהצלחה.

## **הבעיה מבחינת הנדסת תוכנה**

בכל מסך של האפליקציה אדרש להציג את נתוני הנתב הקיימים ובחלק מהמסכים אצטרך לממש ייצוג ויזואלי לנתונים כמו למשל מד טווח שידור הנתב – אצטרך להציג מד (Gauge) שיתעדכן בזמן אמת כאשר המכשיר ישנה מיקום. התחברות לנתב על ידי API של אנדרואיד על מנת למשוך נתונים מהנתב דורש גם כן הבנה מעמיקה בנושא תהליכים ברקע שירותי האפליקציה. גישה לנתוני הנתב על מנת להגדיר את הנתב מחדש, היא פעולה מסובכת שמצריכה מחקר מעמיק בדרכי התקשורת לנתב והפונקציות המפעילות את השירותים המתאימים להגדרה. האתגרים הצפויים לי במהלך הפרויקט הם מציאת הכלים הנחוצים (API) כדי להשיג את המידע הדרוש למלא את נתוני מסך "מסך נתונים". התאמת פלאג-אין גרפי לקוד ה"מסך טווח" כך שהנתונים יתקבלו בזמן אמת ומחזורי. התחברות ואימות בדרך כלשהי (עדין לא נעשה בשלבים אלה) לנתב. דחיפת נתונים לנתב תוך מטרה לשנות את נתוני הנתב מציבים לי אתגר תוכנתי גדול, מפני שהנתבים אינם מספקים API כללי שבעזרתו אוכל לקדד שינויים בנתב.

# תיאור הפתרון

## **מהי המערכת**

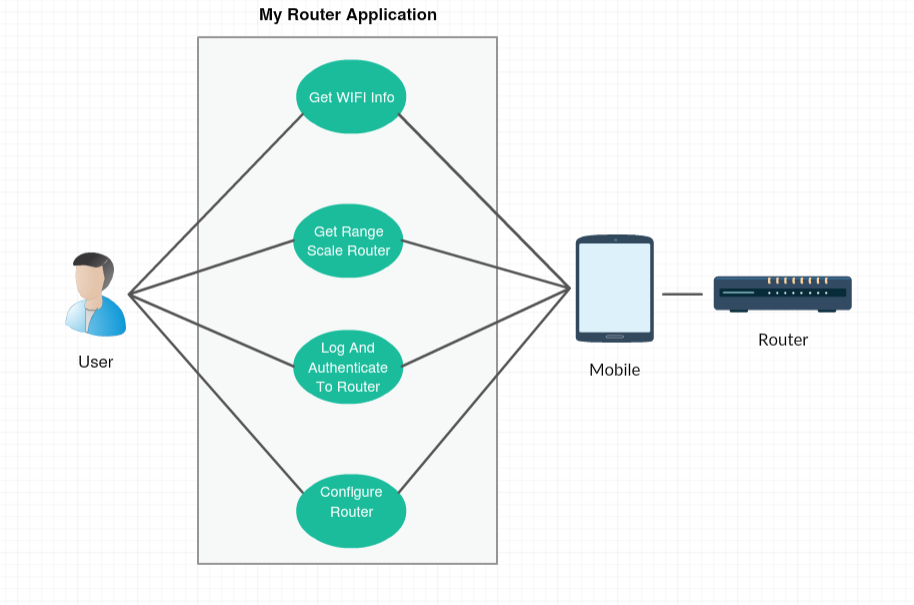
מערכת האפליקציה מורכבת מחמישה מסכים (יתכנו מסכים נוספים לפי דרישות עתידיות), העונות על הצורך להציג נתונים, לבצע אימות ולהגדיר נתונים חדשים.

אם כך נבין קודם, מהם תרחישי השימוש באפליקציה:

* קבלת נתונים מהנתב.
* קבלת טווח קליטה מהנתב.
* אימות מול הנתב לצורך קבלת הגדרות הנתב.
* הגדרות חדשות לנתב.

ניתן לראות באיור המוצג את התרחישים מול המשתמש.

משתמש (user) : מכשיר חכם עליו מותקנת האפליקציה : הנתב הביתי



## תהליכים ונתוני המערכת

**בפרק זה אפרט על המענה לקבלת נתונים מהנתב והצגתם במסך האפליקציה.**

## תהליך שאיבת הנתונים מהנתב:

## על מנת שהאפליקציה תקבל נתונים מהנתב הביתי, עשיתי תחילה מחקר בכל הנוגע לספריות המספקות לי איזשהו מידע על רכיב הWIFI שבמכשיר הסלולרי שלי. לאחר מחקר שביצעתי באתרי האינטרנט שסוקרים את ספריות האנדרואיד בנוגע לעניין הבנתי כי תהליך שאיבת הנתונים מהנתב לאפליקציה מתבצע בעזרת מחלקה של אנדרואיד הנקראת WifiManager והיא דורשת יבוא של הספרייה android.net.wifi.WifiManager. ספריה זו מכילה את כל האספקטים של חיבור רכיב הWIFI לנתב. לרשות המחלקה הזאת הרבה פונקציות הפועלות ברמת המכשיר ומקבלות נתונים מן החיבור לנתב.

על מנת לגשת ולבקש את השירות מהרכיב עלינו לקבל הרשאות:

לכן נכתוב את הפקודה הבאה בקובץ הרשאות האפליקציה AndroidManifest.xml את השורה

<uses-permission *android:name=*"android.permission.ACCESS\_WIFI\_STATE"/>

## לאחר שיש הרשאה וניתן להתחיל לכתוב את זרימת המידע, תחילה יש לאתחל את השירות לקבלת נתונים מרכיב הWIFI במכשיר הסלולרי על ידי הפקודה:

wifi = (*WifiManager*) getSystemService(*Context*.WIFI\_SERVICE);

המשתנה wifi הוא אובייקט של המחלקה WifiManager. הפעולה getSystemService קוראת לספריה שדורשת מופע של מחלקה המממשת את ממשק הספרייה לקבלת נתונים מהאובייקט הדרוש כדי לבצע פעולות כלשהן. כאן *Context*.WIFI\_SERVICE אני בעצם מבקש את הספרייה הכוללת בתוכה גישה לכל הפונקציונליות הדרושה לשאוב או לבצע שינויים בהגדרות רכיב הWIFI של המכשיר המעפיל את האפליקציה.

לאחר מכן יש לקבל את נתוני הנתב על ידי הפקודה

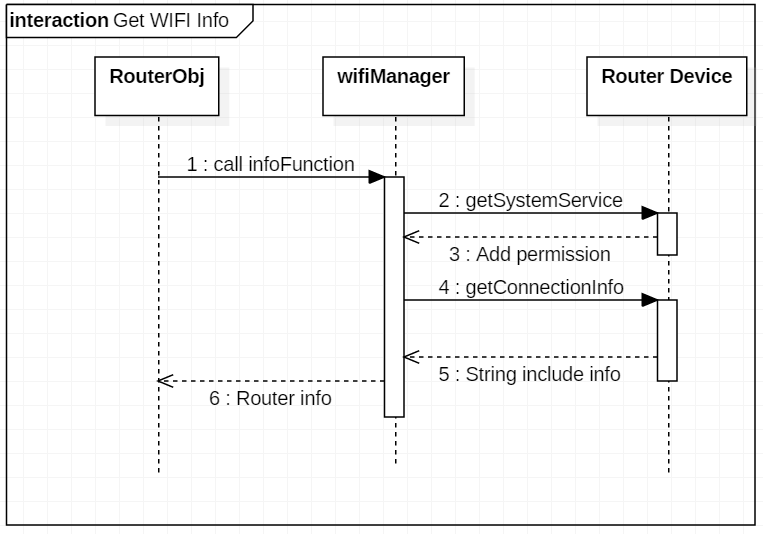
wifi.getConnectionInfo()

וכך לדוגמה נקבל את שם הנתב:

results[7] = "SSID: " + wifi.getConnectionInfo().getSSID();

wifi.getConnectionInfo().getSSID(); מחזירה לנו אובייקט מסוג String המכיל את שם הנתב.

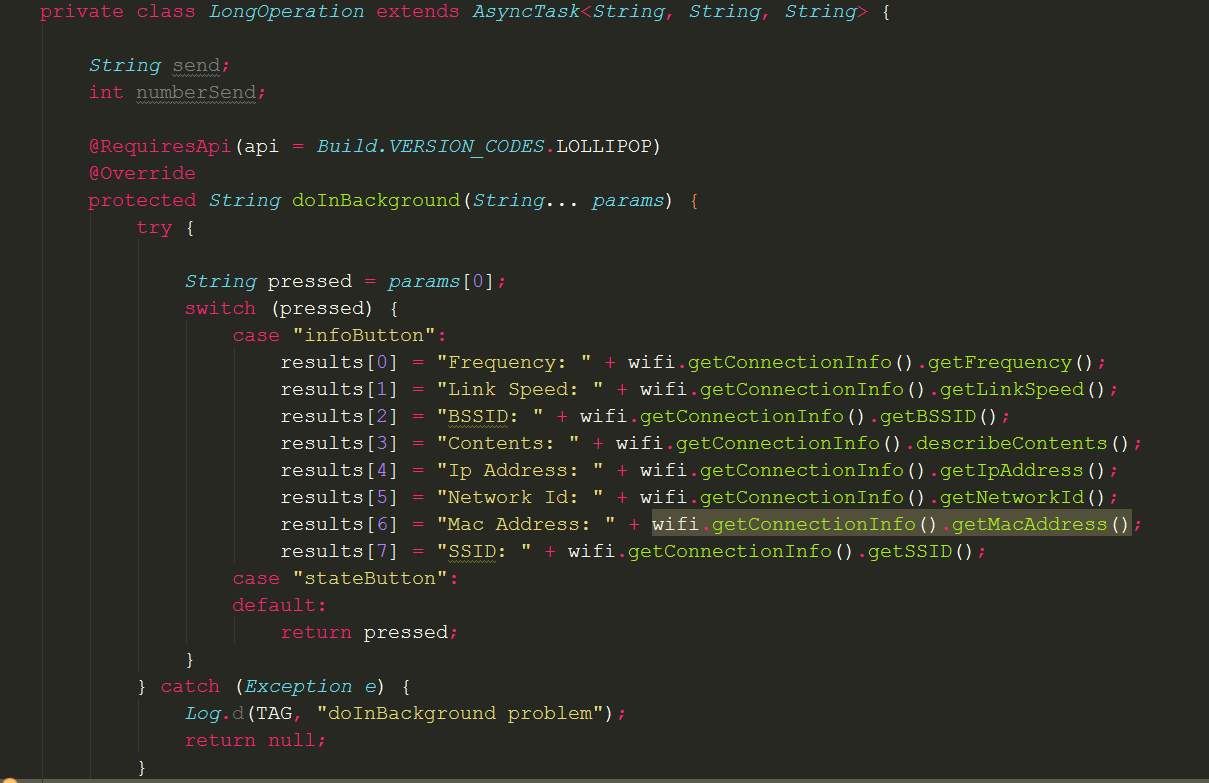
ניתן לראות בתרשים זרימה האיור למטה את דרך קבלת הנתונים מן הנתב המחובר למכשיר תוך קריאה לאובייקטים הנחוצים (ROUTER ו- WifiManager) בקוד התוכנית שלי.



* במהלך הביצוע גיליתי כי הפעולה getConnectionInfo() משביתה את המערכת כי היא מבטלת את שאר שירותי האפליקציה ולמה? הפעולה הזאת דורשת קריאת נתונים מהנתב ולכן לוקחת זמן רב ועד לסיום פעולתה שאר השירותים לא יכולים להתבצע באופן סימולטני.

## **תיאור הכלים המשמשים לפתרון**

הגעתי למסקנה כי את אותה פעולה שלוקחת זמן עיבוד רב אבצע בתהליך הפועל ברקע האפליקציה ולא בתהליך המרכזי שרץ על האפליקציה. חיפשתי גם כלי אשר מאפשר לבצע פעולה ארוכה ללא השבתת שאר שירותי המערכת. נעזרתי במחלקה Long Operation:



## **הסבר:**

## ניתן לראות כי המחלקה LongOperation יורשת ממחלקה אשר מתזמנת תהליכים ברקע האפליקציה, והיא בעצמה יודעת לבצע או לקרוא לפונקציונליות אשר מתרחשת ברקע האפליקציה ושאינה מונעת משירותים אחרים באפליקציה להתרחש. אותה שיטה שתתרחש ברקע נקראת:

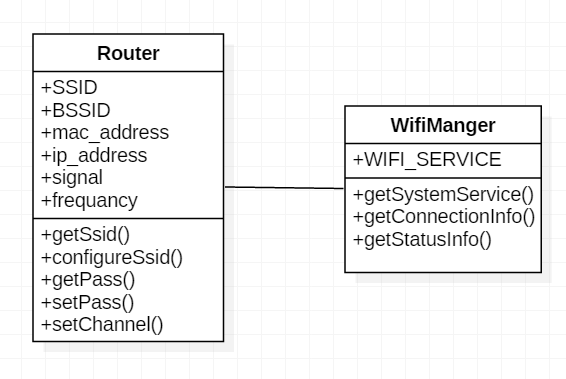
protected *String* doInBackground(*String*... *params*) {

חתימתה מאפשרת לי לקבל את מה שהמחלקה LongOpreation מקבלת. בקוד זה אני מעפיל את הבקשה לכל נתוני הנתב הנחוצים לי בפרויקט. כדוגמה:

results[4] = "Ip Address: " + wifi.getConnectionInfo().getIpAddress();

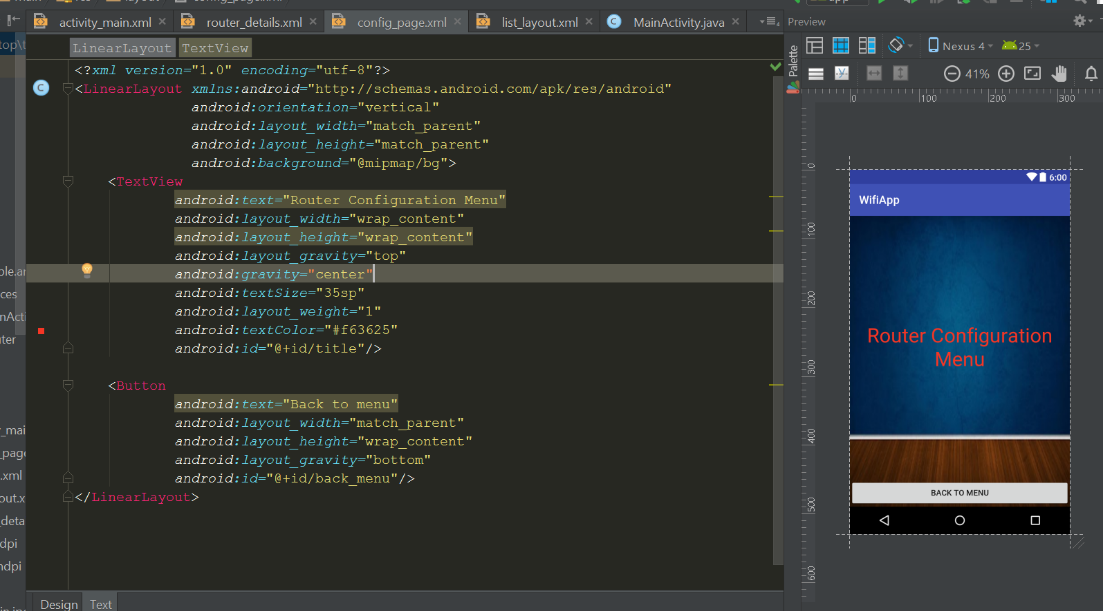
בליווי תפיסת שגיאה כלשהי הקוראת ברקע האפליקציה.

* כעיקרון על מנת לשאוב נתונים מן רכיב ה-WIFI המחובר לנתב, חשבתי על הצורך במודל המפריד את הנתונים ומאגד אותם במחלקה מסוימת. באובייקט שייצג לי את הנתב כך שאוכל להפעיל דרך האובייקט את כל השיטות הרצויות לי לאפליקציה ושכן יכיל גם את כל הנתונים כמשתנים להצגה על מסכי אפליקציה. לכן בחרתי להביא לכאן גם את הייצוג שלי לאובייקט הנקרא Router. ניתן לראות את ייצוגו בתרשים המחלקות המוצג.



דבר נוסף שמומש עת לכתיבת האב-טיפוס הוא המעבר בין מסכי האפליקציה על מנת לקבל בעתיד את השירות המבוקש באותם המסכים. אדגים את המימוש על מסך אחד:

את המעבר בין המסכים מימשתי ביצירת קובץ XML חדש עבור כל מסך כדוגמת המסך config\_page.xml שניתן לראות בתמונה וגם את המסך שנוצר מהקוד שכתוב ב XML.



על מנת להגדיר את המעבר לדף החדש, היה עלי לרשום גם בקובץ הלוגיקה של ההמסך את הדבר הבא:

final *Context* context = this;

configButton.setOnClickListener(new *View.OnClickListener*() {  
 public void onClick(*View v*) {  
 *Intent* intent = new Intent(*context*, *Router*.class);  
 startActivity(intent);  
 }  
});

בדרך זו אני משתמש באובייקט Intent שזהו אובייקט מופשט שיכול להפעיל פעולה כלשהי המבוצעת אל אותו אובייקט שבו האובייקט נמצא בscope שלו למשל השיטה startActivity המקבלת intent חדש שהוא מקושר לקובץ לוגיקה אחר שמחולל את הלוגיקה שלו במסך המקושר אליו, במקרה זה המסך הוא config\_page.xml.

את שאר המסכים עוד לא מימשתי בפרויקט ועוד דרושה עבודת מחקר נוספת על מנת להתחיל לעבוד על אימות מול הנתב והגדרות הנתב מחדש לפי צורך המשתמש.

לאחר מחקר משותף עם המנחה של הפרויקט שלי, אגש לביצוע הקוד של הצגת טווח קליטת הנתב, אימות מול הנתב, בניית ממשק להגדרות הנתב וביצוע שינויים בהגדרות הנתב.

# תכנית בדיקות

לשלב זה של האב טיפוס הבדיקות המבוצעות על האפליקציה הן:

* בדיקת תקשורת אל מול הנתב. בדיקה אם אכן מקבלים את שמו הנכון של הנתב והמידע מדויק. מתאפשר לי לבדוק את אמינות המידע בעזרת דף האינטרנט של הנתב.
* בדיקת נפילות שירותי האפליקציה: הפעלת שיטת קריאת המידע מהרכיב WIFI של המכשיר עליו יושבת האפליקציה, אינו מונע משאר השירותים של המערכת לפעול ולתפקד. הבדיקות נערכות באמצעות הסימולטור של Android Studio ובעזרת המכשיר הסלולרי החכם שלי ועוד כמה שלקחתי לצורך הפרויקט.
* בדיקת תאימות גרסאות של מערכות ההפעלה מסוג אנדרואיד: בדיקות האם מכשירים בעלי מערכות הפעלה בגרסאות שונות, תומכות באפליקציה ופונקציונליות האפליקציה עובדות בצורה מדויקת.
* בדיקות ידניות מקצה לקצה שכל הכפתורים והטקסטים באפליקציה מתפקדים בהתאמה לזרימת פעולת האפליקציה.

# סיכום \ מסקנות

בשלב זה הגדרתי לעצמי שתי שלבים עיקריים בין השלבים הנספחים והם:

* שאיבת מידע מהנתב לצורך ניתוח המידע של הנתב והצגתו על גבי מסכי האפליקציה בממשק גרפי כלשהו.
* אימות מול הנתב לצורך כניסה לשינוי הגדרות (יתכן וזה יתפצל לשלב עיקרי בעצמו) ודחיפת מידע חדש לנתב לצורך הגדרת הנתב מחדש.

בשלב האב טיפוס, ביצעתי בהצלחה קריאת נתונים מהנתב דרך רכיב התקשורת WIFI של מכשיר הסלולרי שלי ועוד כמה מחשבי לוח נוספים.

בשלב זה אני עובד על מחקר שבסופו אני מקווה להגיע להבנה, כיצד אני מקבל גישה לנתב בצורה מסוימת כך שאוכל לדחוף שינויים לנתב בכל דרך שהיא. במחקר אני שואל את השאלות הבאות: האם לכל נתב יש מערכת ספריות משלו? האם קיימים מערכות הפעלה שונות לכל נתב? כיצד ניתן להגיע לאחת מהפונקציות של הנתב? איזה דרך הכי נוחה לדחוף שינויים לנתב? שאלות אלו ועוד נחקרות בעבודת המחקר נעשית בליווי המנחה מרים אללוף תוך חיפוש במעמקי האינטרנט ובספרות. הגעתי למסקנה כי אם אצליח להזדהות מול הנתב, למעשה הצלחתי לקרוא לפונקציה של הAPI בו משתמש הנתב ואפילו לשלוח לאחת מן הפונקציות פרמטרים. במידה והצלחתי לעשות זאת, אני מאמין שהדרך לניתוח שאר הפונקציות של הנתב תהיה הרבה יותר פשוטה וכן אוכל לדחוף שינויים להגדרות הנתב.

# נספחים

## רשימת ספרות \ ביבליוגרפיה

אתר ספריות API של Android Studio

<https://developer.android.com/index.html>

אתר המסביר על חיבור לרכיב הWIFI

<https://www.tutorialspoint.com/android/android_wi_fi.htm>

אתר המדריך כיצד לבצע תהליך ברקע

<https://developer.android.com/reference/android/os/AsyncTask.html>

לינק המסביר כיצד להשתמש במחלקה LongOpreation

<http://stackoverflow.com/questions/9671546/asynctask-android-example.html>

לינק ממנו לקחתי הסברים על הנתב:

<http://etutorials.org/Networking/>

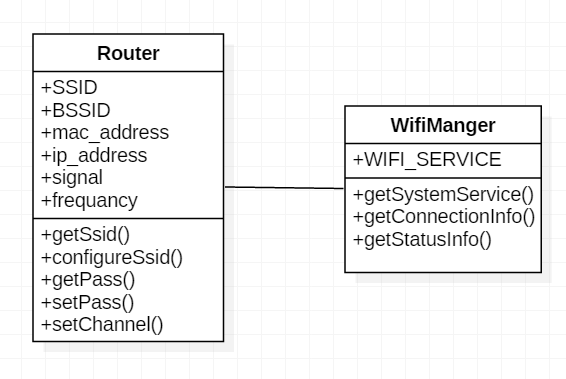
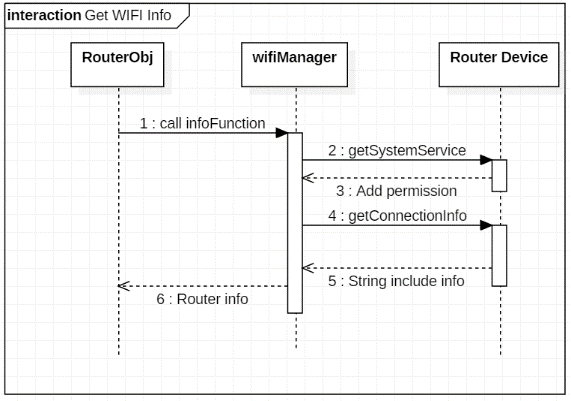
<https://en.wikipedia.org/wiki/Router_(computing)>

לינק ממנו לקחתי הסברים על מערכת ההפעלה "אנדרואיד":

<https://en.wikipedia.org/wiki/Android_(operating_system).html>

## תרשימים וטבלאות

תרשים מחלקות של קבלת מידע תרשים זרימה לבקשת מידע מהנתב



## תכנון זמני הפרויקט

|  |  |
| --- | --- |
| 4.10. | <<פגישת הכרות עם הלקוח >> |
| 8.8.16 | שיחת פתיחה על הפרויקט – פיתוח הרעיון הקודם |
| 4.12.16 | הגשת הצעה |
| 5.12.16 | שיחת פתיחה על הפרויקט – פיתוח הרעיון החדש |
| 7.12.16 | שיחה על ההצעה |
| 12.12.16 | פירוק הפרויקט לשלבים |
| 16.1.16 | הצגת שלב ראשון – שאיבת נתוני הנתב |
|  | הגשת אב טיפוס |
| זמנים אלו | שיחה על האב טיפוס |
| יקבעו בהמשך | הצגת שלב שני – הגדרת נתוני הנתב |
| עם עשיית | בניית מצגת להצגה החולשות והפתרונות לרעיון |
| הפרויקט | הקמת שרת RADIUS |
|  | הקמת אפליקציה |
|  | בדיקות תוכנה |
|  | מסירה |
|  | העברה |

## טבלת סיכונים

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **הסיכון** | **חומרה** | **מענה אפשרי** |
| 1 | אי עמידה בזמנים | 5/5 | הוספת שעות נוספות |
| 2 | פתרון לא יעיל שעלול לגרום לעיכוב בהגשת הפרויקט | 2/5 | לנסות לחפש פתרון שהומצא ולהתאימו לפרויקט ולא להמציא את הגלגל |
| 3 | חלק מהנתבים חוסמים את האפליקציה מלהתחבר | 4/5 | אנסה להתאים אימות נכון יותר לאפליקציה לצורך התחברות. |
| 4 | כל נתב מקיים ספריית פונקציות משלו | 3/5 | אעבור על קבצי השיטות של הנתב ואשלב אותם בפרויקט |
| 5 | כלים: הטלפון החכם מפסיק לעבוד וצריך להשיג חדש | 3/5 | פועל להשגת טלפון דרך המכללה או קונה בעצמי. |
| 6 | תהליך באפליקציה רומס את האפשרות להמשך שימוש ברכיבי המכשיר | 3/5 | קריאה בספרות ויצירת תהליכי רקע על מנת להתגבר על הבעיה. |
| 7 | חוסר ידע בתקשורת בתקן IEEE 802.11 | 2/5 | קריאה מרובה על רשתות WIFI בתקן IEEE 802.11 |
| 8 | שאיבת הנתונים לאפליקציה | 2/5 | מחקר על אפליקציות ששואבות נתונים משרתים. |
| 9 | חוסר ידע באנדרואיד | 2/5 | ללמוד אנדרואיד בלי שום קשר לקורס במכללה. |

## רשימת\טבלת דרישות

**טבלת דרישות (User Requirement Document)**

|  |  |
| --- | --- |
| מס' דרישה | תיאור |
| 1 | הצגת נתוני הנתב בצורה המחולקת לקטגוריות ומסכים שונים. |
| 2 | חיבור לנתב דרך השם משתמש והסיסמא של הנתב על מנת לדחוף נתונים לנתב. |
| 3 |  |
| 4 |  |
| 5 |  |