

# **תיק פרויקט הנדסת תוכנה**

**בית ספר: אילון**

**אפליקציית EZWay**

**שם תלמיד: רום גרנות**

**ת.ז. 325020790**

**שם המנחה: אבי טויל**

**שם החלופה: טלפונים חכמים**

**תאריך הגשה: 31.3.2021**

## מבוא

### תוכן עניינים

2	מבוא
2	תוכן עניינים
5	רקע לפרויקט
5	תהליך המחקר
6	אתגרים מרכזיים
7	הצגת פתרונות לבעיה
9	הארכיטקטורה של הפרויקט
9	הצגת הפתרון המוצע והסיבות לבחירתו
10	ארכיטקטורה של פתרון מוצע בפורמט Top – down level design
11	מחלקות הפרויקט
12	מסכים
13	מתאמים
14	מחלקות בסיסיות
15	דיאלוגים
16	מטפלים
17	מבני נתונים שנעשה בהם שימוש
17	זרימת מידע בין המחלקות השונות בשלבי הניווט
19	תיאור האלגוריתם הראשי ותיאור אלגוריתמים עיקריים
21	הצגת מקרה עבור הפונקציות העיקריות בפרויקט
22	טכנולוגיות שנעשה בהן שימוש בפרויקט
22	ממשקים למערכות אחרות
23	תרשים UML
24	מדריך למשתמש
24	תרשים מסכים (Screen flow diagram)
25	מסך התחברות
26	מסך הרשמה
29	מסך מפה
30	מסך חנות
31	מסך הנחות
32	מסך ביקורות

33.....	מסך תפריט ראשי
34.....	מסך פרופיל
36.....	מסך הגדרות
37.....	מסך הגדרות מפה
38.....	מסך מקומות מועדפים
39.....	<b>בסיס הנתונים</b>
39.....	Shared Preferences
40.....	Realtime Database
45.....	Storage
47.....	Authentication
48.....	Cloud Messaging
49.....	<b>סיכום אישי / רפלקציה</b>
49.....	איך הייתה עבורי העבודה על הפרויקט?
49.....	מה קיבלתי מהעבודה?
50.....	אילו כלים אני לוקח איתי להמשך?
50.....	מה הם הקשיים והאתגרים שעמדתי בפניהם?
51.....	מה המסקנות שלי מהעבודה?
51.....	מה הייתי עושה אחרת לו הייתי מתחיל היום?
51.....	מה אם היה קורה אחרת, העבודה הייתה יותר יעילה עבורי?
52.....	<b>מדריך למפתח</b>
52.....	AndroidManifest
54.....	<b>Activites</b>
54.....	Class CriticsActivity
58.....	Class DiscountActivity
61.....	Class FavoritePlacesActivity
65.....	Class LoginActivity
68.....	Class MapActivity
75.....	Class MapSettingsActivity
78.....	Class MenuActivity
81.....	Class ProfileActivity
85.....	Class SettingsActivity
88.....	Class ShopActivity
91.....	Class SignUpActivity
95.....	<b>Adapters</b>
95.....	Class DiscountRecyclerViewAdapter

98.....	Class FavShopsAdapter
101.....	Class MyAdapter
103.....	Class MyAdapter.MyViewHolder
104.....	Class RecyclerView
107.....	Class RecyclerView.MyViewHolder
109.....	Class ReviewAdapter
112.....	<b>Base Classes</b>
112.....	Class BeaconInfo
114.....	Class Discount
117.....	Class MyDate
119.....	Class Navigation
124.....	Class Position
127.....	Class Review
130.....	Class Store
134.....	Class User
137.....	<b>Dialogs</b>
137.....	Class ConfirmPasswordDialog
140.....	Class DiscountDialog
142.....	Class MyDatePickerDialog
144.....	Class ReviewDialog
147.....	<b>Handlers</b>
147.....	Class BottomSheetHandler
150.....	Class BroadcastReceiverHandler
151.....	Class CustomToastHandler
152.....	Class MessageHandlerService
154.....	<b>ביבליוגרפיה</b>

## רקע לפרויקט

בתור ילד שטייל יחסית הרבה בעולם, לא פעם ולא פעמיים שמתי לב שכאשר משפחתי ואני נכנסים למקומות כמו שווקים וקניונים גדולים בחוץ לארץ, אנחנו לעיתים הולכים לאיבוד בין החנויות השונות כאשר אנחנו מחפשים חנות מסוימת, או לא מבינים הנחיות של כיצד להגיע לחנות שניתנו על ידי אדם מקומי. לכן, לאחר חשיבה ארוכה על איזה סוג של אפליקציה אני רוצה לפתח ולהגיש כפרויקט הגמר שלי בהנדסת תוכנה, מזכרתי בטיולים הרבים האלה והחלטתי לעשות אפליקציה המבוססת על ניווט במקומות כאלה.

האפליקציה EZWay שפיתחתי נותנת בעצם מענה לבעיה שתיארתי, מפני שהיא נותנת למשתמש דרך לנווט בקלות בקניון עם עיצוב פשוט למדי שכל אדם בכל גיל יכול לתפעל, כך שלמשל גם אנשים מבוגרים אשר צריכים לקנות את התרופות שלהם ולא זוכרים את הדרך לבית מרקחת שבקניון או תיירים שרוצים לנווט בקניון או בשוק ביעד שהם מטיילים בו יוכלו לעשות זאת ביתר קלות במקום לבקש עזרה ממקומיים שעלולים לבלבל אותם או להבין לא נכון את המידע שנמסר.

## תהליך המחקר

תחילה, כאשר המנחה שלי אישר את רעיון האפליקציה התחלתי לחפש בגוגל אם קיימות אלטרנטיבות שונות שהרעיון שלהם דומה לרעיון האפליקציה שלי ואכן מצאתי אחת כזאת. היא הייתה נראית, מהתמונות שמצאתי, לא מובנת לשימוש ולא ממוקדת ברעיון שאני חיפשתי לפתור. לכן, ניסיתי לקחת את האלמנטים הטובים מהאפליקציה (בעיקר מבחינת עיצוב) ולשפר אותם על מנת ליצור אפליקציה טובה יותר שמותאמת לצרכים ולדרישות שרציתי שיבואו לידי ביטוי בפרויקט שלי.

לאחר זמן מה שעבדתי על הפרויקט, בעיקר על שלד האפליקציה ועל בסיס הנתונים התחלתי את החלק בו אני מתעסק עם הניווט. המנחה שלי הביא לי את החיישנים וגם העביר לי אפליקציה מאוד בסיסית שרק מדפיסה למסך את מרחק המשתמש מהחיישן.

חקרתי על החיישנים ובדקתי את המגבלות שלהן ועשיתי מחקר על הקוד של האפליקציה שהמנחה שלי רשם על מנת להבין כיצד להשתמש בחיישנים ומה כל פקודה עושה. במחקר באינטרנט על אודות החיישנים מצאתי פרטים רבים כגון כיצד החיישנים מתנהגים ועל איזה API הם עובדים ואיך הסביבה המקיפה את החיישנים משפיעה עליהם.

גיליתי כי יש עדיפות להשתמש בחיישנים למרחקים בינוניים 5-10 מטרים וכי יש חשיבות לסביבה בה ממוקמים אותם חיישנים, לסוג האוויר, למהירות שליחת אותות בין הטלפון לחיישן וכדומה.

## אתגרים מרכזיים

אתגרים מרכזיים שהיו בזמן הפרויקט היו בעיקר העבודה עם החיישנים ולמידה כיצד להשתמש בהם נכון. כאשר התחלתי לעבוד עם החיישנים גיליתי שיש להם מידת סטיה ואי דיוק גדול בהרבה ממה שחשבתי (בגלל הסיבות שפורטו **בתהליך המחקר** כגון סוג אוויר, מרחק בין משתמש לחיישן וכו').

לכן, רציתי לנסות להשתמש בחיישנים נוספים שיש למנחה שלי בבית הספר על מנת לבדוק אם אלה יעבדו בצורה יותר אופטימלית אך בשל מגבלות הקורונה בית הספר בעת שלב הניסויים של הניווט היה סגור ונאלצתי להשתמש בחיישנים שניתנו לי. דבר זה גרר מחקר רציני שבו ביצעתי ניסויים רבים שעירבו בדיקת מרחקים שונים בין חיישנים ומיקומם ובדיקת אלגוריתמים שונים על מנת לראות מה עובד בצורה הטובה ביותר עבור הניווט.

תחילה רציתי לנסות להשתמש בשיטה הדומה לזו המשתמשים ב-GPS (שימוש במיקום החיישנים והרדיוס של המשתמש מהחיישן על מנת למקם אותו על המפה, עליה יפורט בהמשך **בהצגת פתרונות לבעיה**). אך, הדבר יתברר לבלתי אפשרי בגלל האי דיוקים של החיישנים בעוד שהמתמטיקה העומדת מאחורי האלגוריתם והקוד שרשמתי דרש לדיוק במידע של החיישנים.

לכן, נאלצתי לחשוב על דרך אחרת לניווט. לאחר שהמנחה שלי ואני חשבנו על אלגוריתם ניווט חדש, גם כאן התברר כי האי דיוקים באפליקציה גורמים לכך שבמרחקים מסוימים היא לא מדויקת כלל וכי היא גם "מתעכבת מאחורה" (כלומר, לוקח זמן לשינוי של המרחקים המתקבלים מן החיישנים להשפיע בצורה ניכרת) בגלל שצריך "לרכך רעשים" של החיישנים ולכן ההתקדמות במציאות הרבה יותר מהירה מההתקדמות באפליקציה.

יתר על כן, בהתחלה הייתה לי מוטיבציה רבה לעבוד על האפליקציה וסיימתי חלקים נרחבים מהאפליקציה עוד לפני שהתחילה שנת הלימודים של כיתת י"ב. אבל, את החלק החשוב והעיקרי ביותר באפליקציה שהוא הניווט השארתי לסוף. הניווט יתברר כקשה בהרבה ממה שחשבתי וזה בעיקר בגלל שלא היה לי דרך לראות את המנחה שלי ולשאול אותו שאלות מסוימות שיותר קל לשאול כאשר מראים את האפליקציה עובד. בנוסף, את אי הדיוק ומידת הסטיה לא חזיתי מראש וחשבתי שהדיוק יהיה יותר טוב. בגלל זה היו פעמים רבות בעת כתיבת הקוד שבהם לא ידעתי כיצד לפתור את הבעיה בגלל המגבלה של הטכנולוגיה איתה אני עובד. לכן, רמת המוטיבציה שלי לעומת ההתחלה ירדה, אך ניסיתי להחזיק במוטיבציה כמה שיותר מפני שידעתי שרעיון האפליקציה הוא טוב ושביצוע נכון שלו גם יכול להיות התחלה מוצלחת לעתיד של פרויקטים מיוחדים וזה הוא גם בסיס טוב כיצד לנהל פרויקט גם כאשר אין מוטיבציה כמו פעם ובסופו של דבר התמדתי והגעתי לתוצאה שאני מרוצה ממנה.

הצורך שהפרויקט עונה עליו הוא פתרון לבעיה שדיברתי עליה לעיל (סוגיית הניווט). הוא מספק חווית משתמש כך שכל אחד יכול להבין ולהתמצא במקום שהוא נמצא בו, ובנוסף לקבל הנחות ומבצעים שיעודדו את המשתמש להיעזר באפליקציה

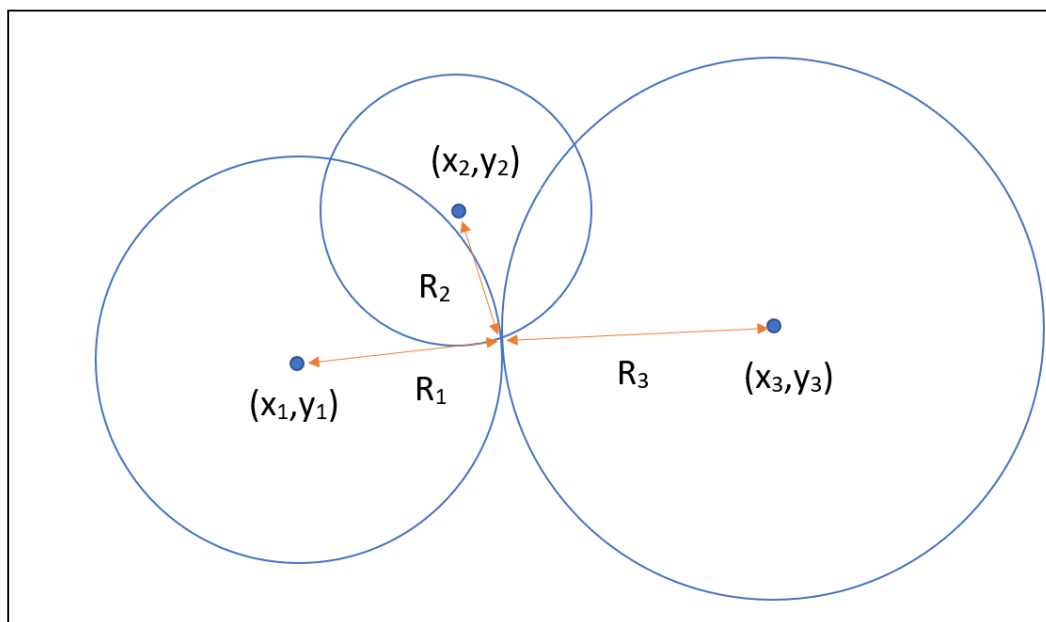
## הצגת פתרונות לבעיה

אתאר כאן את הפתרון לבעיה אשר בסופו של דבר לא צלח, ובהמשך בתיאור האלגוריתם הראשי, ותיאור האלגוריתמים עיקריים את הפתרון הסופי.

הפתרון שחשבנו שיהיה המועיל ביותר לבעית הניווט הוא שימוש בחיישני מרחק על מנת למקם משתמש במפה. על ידי כך ניתן להציג את המפה למשתמש תוך כדי שהוא מנווט בשוק או בקניון עם התקדמותו. חיישני המרחק האלו שולחים דרך ה - Bluetooth אותות אשר הטלפון מנסה לקלוט כל פרקי זמן קבועים שמוגדרים באפליקציה.

תחילה, על מנת לבדוק את גבולות יכולת החיישנים ורמת דיוקם בניתי אפליקציה משנית שמציגה נתונים למסך הטלפון בזמן אמת בטבלאות על מנת לאמת נתונים שאמורים להיות מוצגים בניווט ובזכות כך ששינתי בניסויים את מרחקי החיישנים הגעתי למרחקים וביצועים הטובים ביותר שניתן להגיע אליהם עם החיישנים.

כפי שנאמר באתגריים המרכזיים, תחילה חשבתי על פתרון הבעיה באמצעות שיטה הדומה ל - GPS אשר משתמשת במיקום החיישנים על המפה ועל המרחק מהמשתמש לחיישן. בעצם, היא לוקחת את מיקומם של שלושת החיישנים הקרובים ביותר ובזכות הרדיוסים שהם המרחק בין המשתמש לחיישנים השונים, מייצרת 3 מעגלים שנחתכים לכדי נקודת חיתוך משותפת אחת. כאשר בודקים את חיתוכם זה עם זה, אמורים להגיע למיקום אחד של המשתמש שאמור להיות פחות או יותר מדויק ולמקם אותו על המפה. נציג את הרעיון באופן ויזואלי:



בתמונה למעלה ניתן לראות 3 מעגלים כאשר כל מעגל מייצג חיישן ומרחקו מהמשתמש. כל מרכז מעגל מייצג מיקום של החיישן במפה. בכל קריאה מהחיישנים המרחק מהמשתמש מכל חיישן נשמר והם משמשים כרדיוס המעגל.

אם החיישנים היו מדויקים מספיק אז היה ניתן לראות 3 מעגלים כאשר הם חותכים אחד את השני ויוצרים נקודת חיתוך משותפת. אחת שאמורה לייצג את המשתמש ואת מיקומו על המפה כפי שמתואר בתמונה. בנוסף, רעיון זה אמור לעבוד היטב בגלל שהרדיוסים משתנים בהתאם למיקום המשתמש בעוד שמרכזי המעגלים שונים זה מזה ולכן יוצר חיתוך יחיד בין כל המעגלים.

אך, בגלל שהחיישנים לא היו מדויקים מספיק, נוצר מצב בו המעגלים והרדיוס שלהם קטנים מדי (או גדולים מדי) בשל אי הדיוק הרב שהיה לחיישנים. מכך נבע שהיו מספר נקודות חיתוך של המעגלים ולא היה ניתן ליצור חיתוך אחד של המעגלים. כלומר, לא היה ניתן לממש את הרעיון. לכן, היה צורך לשנות את רעיון פתרון הניווט ליחסי מרחקים בין החיישנים, ופירוט לכך יהיה בהמשך בתיאור האלגוריתם הראשי, ותיאור האלגוריתמים עיקריים.



## הארכיטקטורה של הפרויקט

### הצגת הפתרון המוצע והסיבות לבחירתו

הפתרון הסופי, כפי שהתחלתי לפרט עליו בהצגת פתרונות לבעיה הוא שימוש בחיישני מרחק על מנת למקם את המשתמש במפה דו – ממדית. באמצעות כך שהאפליקציה תבחר את שני החיישנים הקרובים ביותר ותחשב את יחסי המרחקים בין החיישנים, היא תדע למקם את המשתמש בין החיישנים והמשתמש יוכל לנווט.

הסיבה שבחרנו דווקא בחיישנים אלה היא מכיוון שקניון או שוק הם מקומות גדולים יחסית עם מרחקים גדולים במסדרונות. חיישנים אלו הם חיישנים שמיועדים למרחקים בינוניים (עשרה עד חמישה עשר מטר מרחק בין חיישן לחיישן, בשביל דיוק מרבי) ולכן השימוש בחיישנים הוא מינימלי. כך, מספר החיישנים המשומשים עבור האפליקציה הוא מינימלי ככל הניתן.

האלגוריתם והרעיון נמצאים בצורה מפורטת יותר בתיאור האלגוריתם הראשי, ותיאור אלגוריתמים עיקריים. הרעיון הוא שבחירת החיישנים הקרובים אמורים לייצג בצורה כמה שיותר אמינה את המיקום האמיתי של המשתמש (מפני שככל שהמרחק גדל בין החיישן למשתמש גם רמת האמינות גדלה). מציאת היחסים עוזרת להבין כיצד למקם על מפה את המשתמש. אם במציאת היחסים המשתמש קרוב יותר לחיישן מסוים מהחיישן השני הקרוב ביותר, המשתמש ימוקם קרוב יותר לחיישן המסוים.

יתר על כן, הרעיון מסתמך על יחסי מרחקים ולא המרחקים עצמם. לכן, הוא יתאים גם למקומות קטנים וגם למקומות גדולים (הכוונה היא מרחקי חיישנים שונים), כל עוד יש אכן דיוק יחסי בחיישנים ונתוני המרחקים ויחסם המתקבל. כלומר, הפתרון שהאלגוריתמים מציעים הוא פתרון גנרי למרחקים ולא מגביל את המרחקים אלא רק דורש שתוצאות היחסים בין המרחקים יהיו הגיוניים ותואמים לניווט.

## ארכיטקטורה של פתרון מוצע בפורמט Top – down level design

באפליקציה, לפי הפתרון שבסופו של דבר בחרנו בו, אנו צריכים ליצור ניווט בעזרת כך שנתקשר עם החיישנים ונשתמש במידע המתקבל מהם, ננתח את המידע הזה ובעזרתו נמקם את המשתמש על המפה בכל קריאה חדשה מהחיישנים.

את הפתרון המוצע פירקתי לגורמים ולתת בעיות לפי פורמט Top – down approach. תחילה, פישטתי את העבודה מהחיישנים והתעסקתי קודם בקוד שמבא את מיקום המשתמש.

אחרי שסיימתי עם חלק זה, עבדתי על החלק בו אני קולט נתונים מהחיישנים עצמם "ומרכז רעשים". לאחר מכן שילבתי בין החלקים (אגב, זו הסיבה שפתרון ההצלבה של שלושה מעגלים שהוסבר למעלה התברר בניסויים כלא אידיאלי – מפני שבהתחלה כשהתעסקתי בקוד התאורטי שרק קולט נתונים פקטיביים הוא היה מדויק ורק לאחר מכן התברר כי הנתונים לא מדויקים מספיק על מנת לעשות שימוש ברעיון זה).

בחלק של ניבוי ומיקום המשתמש פירקתי את המשימות של ניווט למספר חלקים שהם בעצם תת – בעיות לפתרון הכללי והם נמצאים בתוך פעולה אחת ראשית בקוד. החלקים הם:

- קבלת מערך שמכיל עצמים (כאשר כל עצם מכיל בתוכו את מרחק החיישן מהמשתמש והמספר הסידורי של החיישן) ומחיקת ערכים לא רלוונטיים ממנו (כגון מרחקי 0 אשר מסמנים כי אין קלט מחיישנים מסויימים).

- מציאת שני החיישנים הקרובים ביותר למשתמש

- מציאת היחס שנוצר בין מרחק המשתמש לחיישן הקרוב ביותר למרחק בין שני החיישנים.

- חישוב המרחק בין המשתמש לחיישן הקרוב ביותר בעזרת היחס שמצאנו, והפיכת המרחק למרחק בפיקסלים.

- מציאת המיקום של החיישן הקרוב ביותר.

- מציאת הכיוון ממנו צריכים להחסיר / להוסיף את המרחק שמצאנו ממיקום החיישן (כלומר, אם המשתמש נמצא בצד ימין של החיישן הקרוב ביותר, נרצה להוסיף ממיקום החיישן את המרחק שמצאנו על מנת להגיע למיקום המשתמש, לכן נצטרך לדעת כיוון).

- שינוי המיקום של החיישן בעזרת המרחק והכיוון והחזרת המיקום החדש שכעת מנובא להיות מיקום המשתמש.

כל פעולה כאן היא תת פעולה אשר נקראת מפעולה ראשית אחת וגם לחלק מתת הפעולות יש תתי פעולות אשר מקלות על סידור הקוד והבנתו.

לדוגמה, בעת קריאה לתת פעולה אשר מוצאת את שני החיישנים הקרובים ביותר, קוראים לפונקציה שלוקחת מהעצמים המוזנים רק את מרחקיהם על מנת למצוא את המרחקים הקצרים וכך מפשטים את המורכבות של הקוד בפורמט Top – down approach.

בנוסף, חוץ מהניווט, פירקתי את כל הפרויקט שלי לתת נושאים ובעיות לפתירה בשלבים. תחילה יצרתי שלד של עיצוב אשר מצאתי כראוי עבור האפליקציה ובכך יצרתי מעין עטיפה שעליה התכוונתי לבנות את רכיבי הפרויקט העיקריים על מנת להקל על העבודה ולא להתעסק בעיצוב ובניווט בין המסך השונים בזמן, למשל, כתיבת קוד "לשליפת נתונים" מבסיס הנתונים. לאחר מכן עברתי לדברים כגון הרשמה והתחברות למשתמש וגם לשמירת נתונים בסיסית בטלפון.

בהמשך, כאשר הגעתי לחלק של הזנת ושליפת נתונים מבסיס הנתונים. חילקתי זאת למספר חלקים. תחילה, חשבתי על רעיון כיצד הנתונים ישמרו בשרת. אחר כך, על פי אותו שלד פירקתי את ההזנה והשליפה לתת בעיות.

לדוגמה, בהתחלה כאשר הזנתי פרטי חנות לאפליקציה דאגתי שיהיו נתונים בסיסיים כמו מיקום החנות במפה אך רק כאשר הגעתי לחלק הניווט באפליקציה שיניתי את הנתונים כך שיתאימו לנתונים הרלוונטיים עבור הניווט וכך עשיתי לכל חנות. כלומר, פירקתי את הבעיה לתת בעיות ופתרתי כל בעיה בנפרד ולבסוף חיברתי בין הפתרונות לפתרון שיוצר את האפליקציה שלי.

## מחלקות הפרויקט

את מחלקות הפרויקט חילקתי ל – 5 תיקיות על מנת לארגן אותם לפי סוג. חמש התיקיות הן:

- מסכים (Activities).
- מתאמים (Adapters).
- מחלקות בסיסיות (Base Classes).
- דיאלוגים (Dialogs).
- מטפלים (Handlers).

תיעוד מלא של המחלקות נמצא ב – JavaDoc המצורף בהמשך, פה ניתן תיאור כללי של המחלקות.

כעת נפרט על כל אחד מהתיקיות והמחלקות הנמצאות בהן (לפי סדר לקסיקוגרפי).

## מסכים

כל ממשקי המסכים ועיצובם מפורטים בהמשך בתפקידי המסכים.

**LoginActivity** – המסך הראשוני כאשר המשתמש נכנס לאפליקציה בפעם הראשונה או כאשר הוא מתנתק מהמשתמש. במסך זה, המשתמש צריך להזין את פרטי המשתמש על מנת להתחבר בשדות המתאימים. יש אפשרות למטה, שאם המשתמש לא רשום הוא יכול לעבור למסך ההירשמות וליצור משתמש לאפליקציה.

**SignUpActivity** - מסך הירשמות לאפליקציה. קיימים שדות חובה של שם משתמש, אימייל וסיסמה ושני שדות אופציונליים של מגדר ותאריך לידה, ובנוסף יש כפתור הירשמות שלאחר מכן יוביל למסך המפה.

**MapActivity** – במסך זה יופיע מפה של המקום. אם נבחר ניווט לחנות כלשהיא יופיעו חצים שיעזרו לנווט לחנות ומרחק המשתמש מהחנות. בנוסף, יש בתחתית מסך שלושה כפתורים אשר כול אחד מהם יש לו פעולות שונות (אחד הוא המקום ממנו בוחרים חנות לנווט אליה, השני מראה אופציות על החנות כמו לדוגמה להסתכל על הביקורות על החנות, לקרוא הודות החנות או לבדוק הנחות מהחנות והשלישי מעביר למסך התפריט הראשי). מחלקה זו גם מממשת ממשק בשם BeaconConsumer אשר יש בה פעולות למימוש על מנת לחבר בין החיישנים לפרויקט.

**MapSettingActivity** – מסך בו יש אפשרות לשנות את המרחק מהחנות למשתמש מיחידת המרחק מטרים לרגל או ההפך על מנת שהמשתמש יהיה עם יחידות המרחק המוכרות והעדיפות עליו.

**MenuActivity** – במסך זה יש אפשרות לראות את פרטי המשתמש וגם לגשת למסך אשר שם ניתן לשנות את פרטי המשתמש. בנוסף, גם יש כפתורים למעבר למסכים של הגדרות, הגדרות למפה, מקומות מועדפים ואפשרות להתנתק.

**ProfileActivity** – במסך זה יש אפשרות לשים תמונת פרופיל וגם ניתן לשנות את המייל את השם ואת הסיסמה.

**SettingsActivity** – במסך זה יש אפשרות לשנות את הגדרות האפליקציה היותר כלליות. לדוגמה, שפת האפליקציה (אנגלית או עברית), שינוי למצב לילה ואפשרות גם לקבל התראות מהאפליקציה על הנחות מיוחדות וכו'.

**ShopActivity** – מסך בו ניתן לראות תיאור קצר על החנות שניתן ע"י החנות עצמה, אפשרות להתקשר לחנות, שעות קבלה ודירוג ממוצע שנתנו המשתמשים השונים על החנות ומספר האנשים שנתנו דירוג זה.

**CriticsActivity** – מסך הביקורות של החנות, כאן ניתן לראות ביקורות של משתמשים שונים על החנות אליה המשתמש מנווט וגם לרשום ביקורת עליה.

**DiscountActivity** – מסך שבו ניתן לראות מבצעים והנחות של החנות אליה המשתמש מנווט. ההנחות הן גם כלליות (כלומר פונות לכל המשתמשים) וחלק גם פרטיות (לדוגמה, אם יש למשתמש יום הולדת הוא יקבל הנחה).

**FavoritePlacesActivity** – מסך שבו ניתן להוסיף מקומות וכך בפעם הבאה שהמשתמש יבוא לקניון יוכל להיכנס למסך זה וללחוץ ללא חיפוש על החנות המועדפת עליו ובכך להגיע לחנות הזו ביתר קלות.

## מְתָאָמִים

**DiscountRecyclerAdapter** – מחלקה זו יורשת ממחלקה בשם MyAdapter והשדה המרכזי בה הוא שדה אשר שומר את ההנחות שיש למשתמש בחנות אליה הוא מנווט. בנוסף, מחלקה זו מכילה מתודות שמסייעות בהצגות הנתונים בצורה מסודרת למסך.

**FavShopsAdapter** – מחלקה זו יורשת ממחלקה בשם MyAdapter והשדה המרכזי בה הוא שדה אשר שומר את החנויות אשר המשתמש סימן כחנויות מועדפות. בנוסף, מחלקה זו מכילה מתודות שמסייעות בהצגות הנתונים בצורה מסודרת למסך.

**MyAdapter** – מחלקה זו יורשת ממחלקה בשם RecyclerView.Adapter אשר עוזרת להציג למסך בצורת רשימת פריטים את כל הנתונים אשר שמורים בה. המחלקה הזו היא אבסטרקטית. במחלקה זו נוספו שני שדות שהם המסך בו המשתמש נמצא ושדה בוליאני שאומר האם המשתמש מעדיף מסך בהיר או כהה.

**RecyclerAdapter** – מחלקה זו יורשת ממחלקה בשם RecyclerView.Adapter אשר עוזרת להציג למסך בצורת רשימת פריטים את כל הנתונים אשר שמורים בה. בנוסף,

המחלקה מממשת ממשק בשם Filterable (על מנת לסנן את הרשימה) מחלקה זו משמשת עבור הצגת רשימת פריטים של חנויות כאשר תוך כדי המשתמש יכול לרשום שם של חנות ולכן יש לסנן ולהוריד חנויות אשר שמן לא דומה לחנות אותה המשתמש מחפש.

**ReviewAdapter** – מחלקה זו יורשת ממחלקה בשם MyAdapter והשדה המרכזי בה הוא שדה אשר שומר את הביקורות שיש על חנות זו. בנוסף, מחלקה זו מכילה מתודות שמסייעות בהצגות הנתונים בצורה מסודרת למסך.

## מחלקות בסיסיות

**BeaconInfo** - מחלקה זו היא מחלקה המייצגת קריאה אחת מהחיישן. השדות אשר יש במחלקה זו הם המספר הסיידורי של החיישן והמרחק שנקלט בקריאה זו.

**Discount** – מחלקה זו מייצגת הנחה של משתמש בחנות כלשהיא. שדות המחלקה הן מספר זהות ייחודי של ההנחה, כותרת, שם החנות אשר בה יש את ההנחה, המספר זהות הייחודי של המשתמש אשר הוא בעל ההנחה, תאריך תפוגה של ההנחה ומחרוזת שמכילה את תוכן הודעת ההנחה.

**MyDate** – מחלקה זו מייצגת תאריך. שדות המחלקה הן היום, החודש והשנה שבתאריך.

**Navigation** – מחלקה זו היא המחלקה הראשית לניווט. היא מטפלת בכול החלק של הניווט באפליקציה. יש פה מתודות אשר משתמשות בנתונים כמו קריאות מחיישנים ובעזרת אלגוריתמים היא עושה פעולות כמו להחזיר את מיקום המשתמש וכו'.

**Position** – מחלקה זו מייצגת את המיקום של המשתמש במפה, למחלקה זו 3 שדות שמייצגות קואורדינטות ובכך ניתן למקם את המשתמש על מפה.

**Review** – מחלקה זו מייצגת ביקורת על החנות אליה מנווט המשתמש. שדות המחלקה הן מספר הזהות הייחודי של הביקורת, מספר הזהות הייחודי של החנות, שם המשתמש שכתב את הביקורת, תאריך כתיבת הביקורת, מספר הכוכבים שהמשתמש נתן, תיאור של הביקורת ושם החנות.

**Store** – מחלקה זו מייצגת חנות במפה. שדות המחלקה הן מספר הזהות ייחודי של החנות, שם החנות, מספר טלפון של החנות, שעות פתיחה וסגירה, מספר הכוכבים הממוצע שניתנו לחנות, מספר הפעמים שניתנו דירוג לחנות, תמונה של לוגו החנות, תיאור של החנות ומיקום החנות במפה.

**User** – מחלקה זו מייצגת משתמש. שדות המחלקה הן מספר הזהות ייחודי של החנות, תאריך לידה ומגדר (אופציונלי), שם המשתמש ודוא"ל.

## דיאלוגים

**ConfirmPasswordDialog** – מחלקה זו יורשת מהמחלקה Dialog. שדות המחלקה הם layout של הדיאלוג ושל המסך, שדה של הדיאלוג עצמו, שדה של המסך ושדות בוליאניים (לדוגמה, האם לשנות סיסמה או האם לשנות דוא"ל). במחלקה זו יש מתודת פתיחת דיאלוג שבו המשתמש מזין את הסיסמה פעמיים על מנת לאשר שינוי של פרטים עליו כמו שינוי סיסמה או דוא"ל. שאר המתודות הן מתודות שמטפלות בעדכון הסיסמה או הדוא"ל בהתאמה.

**DiscountDialog** – מחלקה זו יורשת מהמחלקה Dialog. שדות המחלקה הם המסך שממנו נקרא הדיאלוג וגם שדה של הנחה. בקריאה לפעולה הבונה של המחלקה, הדיאלוג מוצג במסך ובו ניתן לראות את כל פרטי ההנחה הרלוונטיים ללקוח כמו כותרת ההנחה, שם החנות, תיאור תוכן ההנחה ותאריך תפוגה.

**MyDatePickerDialog** – מחלקה זו יורשת מהמחלקה DialogFragment ומממשק ממשק בשם DatePickerDialog.OnDateSetListener אשר מקשיבה האם נבחר תאריך. התאריך מייצג תאריך הלידה כך שהמשתמש יקבל הצעות שיותר מתאימות לגילו של המשתמש. שדות המחלקה הם התאריך הנוכחי, על מנת לבדוק שהמשתמש לא בחר תאריך שגדול יותר מהיום ובכך יכול להטעות את המערכת, והשדה השני הוא המסך ממנו נקרא הדיאלוג. במחלקה יש מתודה אשר גורמת לדיאלוג להיות מוצג ועוד מתודה שבודקת את תקינות התאריך שנבחר.

**ReviewDialog** - מחלקה זו יורשת מהמחלקה Dialog. שדות המחלקה הן החנות שעליה נכתבה הביקורת, המסך שממנו נקרא הדיאלוג, התאריך של ביקורת החנות, עצם שעוזר לדירוג החנות, הפניה לבסיס נתונים על מנת לשמור את הביקורת שהולכת להיכתב ושם המשתמש. במחלקה יש מתודות אשר מציגות את הדיאלוג, בודקות את תקינות הביקורת שנכתבה ומשנות את בסיס הנתונים כך שאם המשתמש כבר רשם ביקורת בעבר על החנות היא תמחק ובמקומה תישמר הביקורת החדשה.

## מטפלים

**BottomSheetHandler** – מחלקה זו יורשת ממחלקה בשם BottomSheetDialog. שדות המחלקה הן דיאלוג, המסך ממנו נקרא הדיאלוג, עצם של RecyclerView, מיקום המשתמש, החנות אליה מנווטים, ושדה בוליאני של מצב המסך (בהיר או כהה). מחלקה זו בעצם מטפלת בשני סוגי הדיאלוגים שיש באפליקציה שהם מסוג BottomSheetDialog. יש אחד הדיאלוגים נועד עבור חיפוש חנות מבין רשימת החנויות והדיאלוג השני הוא דיאלוג אשר מציג את האפשרויות השונות לאחר בחירת החנות. במחלקה זו מתודות שונות שעוזרות לסנן ולהבדיל בין הדיאלוגים השונים ולהתנהג בהתאם.

**BroadcastReceiverHandler** – מחלקה זו יורשת ממחלקה בשם BroadcastReceiver. בגלל שהיא יורשת ממחלקה זו היא צריכה לממש מתודה כאשר מתקבל מידע על ה-Broadcast הרצוי. באפליקציה זו, כאשר אחוז הסוללה משתנה, אם האפליקציה פתוחה ואחוז הסוללה פחות מ-30, תוצג הודעה אשר קוראת למשתמש להתנתק מהאפליקציה מפני שהיא עושה שימוש רב בסוללה בעת ניווט.

**CustomToastHandler** – מחלקה זו יורשת ממחלקה בשם Toast. וכאשר נקראת הפעולה הבונה נוצר Toast שמעוצב בצורה שונה וייחודית לאפליקציה. ה-Toast נוצר על מנת להציג הודעות כמו ההודעה שמקבלים מהמחלקה של CustomToastHandler או כמו שמקבלים מהמחלקה MessageHandlerService.

**MessageHandlerService** – מחלקה זו יורשת ממחלקה בשם FirebaseMessagingService והיא שירות שמופעל כאשר מקבלים מבסיס הנתונים הודעה המיועדת למשתמש, אם המשתמש מחובר לאפליקציה בזמן שליחת ההודעה ייוצר עצם מסוג CustomToastHandler ויציג את ההודעה שנשלחה, אך הוא יציג את ההודעה רק אם המשתמש איפשר זאת בהגדרות.



## מבני נתונים שנעשה בהם שימוש

- רשימת מערך (ArrayList) – רשימת מערך הוא מבנה נתונים שיש בו שימוש רב בפרויקט. הוא משמש על מנת לשמור נתונים מבסיס הנתונים כמו רשימת מערך של ביקורות לאפליקציה, הנחות ומבצעים וחנויות מועדפות. בנוסף, כאשר רוצים לסנן את העצמים עם מרחקי 0 במערך נשתמש במבנה נתונים זה (פירוט בהמשך בתיאור האלגוריתם הראשי ותיאור אלגוריתמים עיקריים).
- מפת גיבוב (HashMap) – מבנה נתונים זה משמש בפרויקט על מנת להעלות תמונה לבסיס הנתונים. בסיס הנתונים, כפי שיפורט בהמשך, שומר מידע באמצעות שיטת מפתח – ערך. לכן, ניתן להשתמש בעצם של HashMap ובפעולה מובנית על מנת לשמור את התמונה בבסיס הנתונים.

## זרימת מידע בין המחלקות השונות בשלבי הניווט

זרימת המידע מתחילה בכך שהחיישני מרחק משדרים אותות באמצעות Bluetooth לפלאפון. במסך MapActivity יצרנו הפניה לעצם בשם beaconManager שיושם ממחלקה בשם BeaconManager שהיא מחלקה של API אשר פתוח לשימוש על ידי כולם, והוא גם משמש עבור החיישנים של הפרויקט בשם AltBeacon. אנו מגדירים, בנוסף, גם עצם מסוג Navigation שנקרא navigation על מנת לשמור על 3 קריאות אחרונות של כול חיישן בכך שהוא יוצר מערך ID – ממדי בצורת מטריצה ששומר עצמים מסוג BeaconInfo. אנו מאתחלים את העצם beaconManager ומזינים לו מידע שימושי עבור השימוש בו (לדוגמה, הפניה לאפליקציה המשתמשת בחיישן, הזמן בין קליטת קריאות מהחיישן ועוד).

לאחר שכל אלה הוגדרו, והמשתמש אישר שהוא מסכים לחיבור הBluetooth שמתרחש, הקלטים מהחיישנים מתחילים להתקבל במבנה הנתונים Collection של עצמים מסוג Beacon. עבור כל עצם כזה אנו קוראים לפעולה סטטית שנמצאת בNavigation אשר מקבלת עצם מסוג Beacon ומחזירה עצם מסוג BeaconInfo, כלומר שומרת את הפרטים שחשובים לפרויקט מבחינתנו, והם המרחק של החיישן מהטלפון, והמספר הסידורי של החיישן.

כעת, כל פעם שאנו מקבלים חזרה עצם מסוג BeaconInfo, אנו שומרים אותו בnavigation בתור קריאה אחרונה ומחליפים במידת הצורך קריאות ישנות. לאחר מכן קוראים לפעולה סטטית בNavigation אשר לוקחת את המטריצה, עושה חישובים מתמטיים ומחזירה מערך חד – ממדי של עצמים מסוג BeaconInfo שהם ממוצעים של הקריאות האחרונות של כול חיישן בנפרד ושומרים מערך זה.

חשוב לציין, כי בהתחלה כאשר מגדירים את שדות במסך, אנו יצרנו גם עצם של Position בשם pos והגדרנו אותו להיות נקודה שרירותית על המפה. כעת, לאחר שקיבלנו את המערך החד – ממדי, אנו קוראים לפעולה סטטית נוספת בNavigation בשם getPosition שהיא לוקחת את המערך החד – ממדי של עצמים מסוג BeaconInfo.

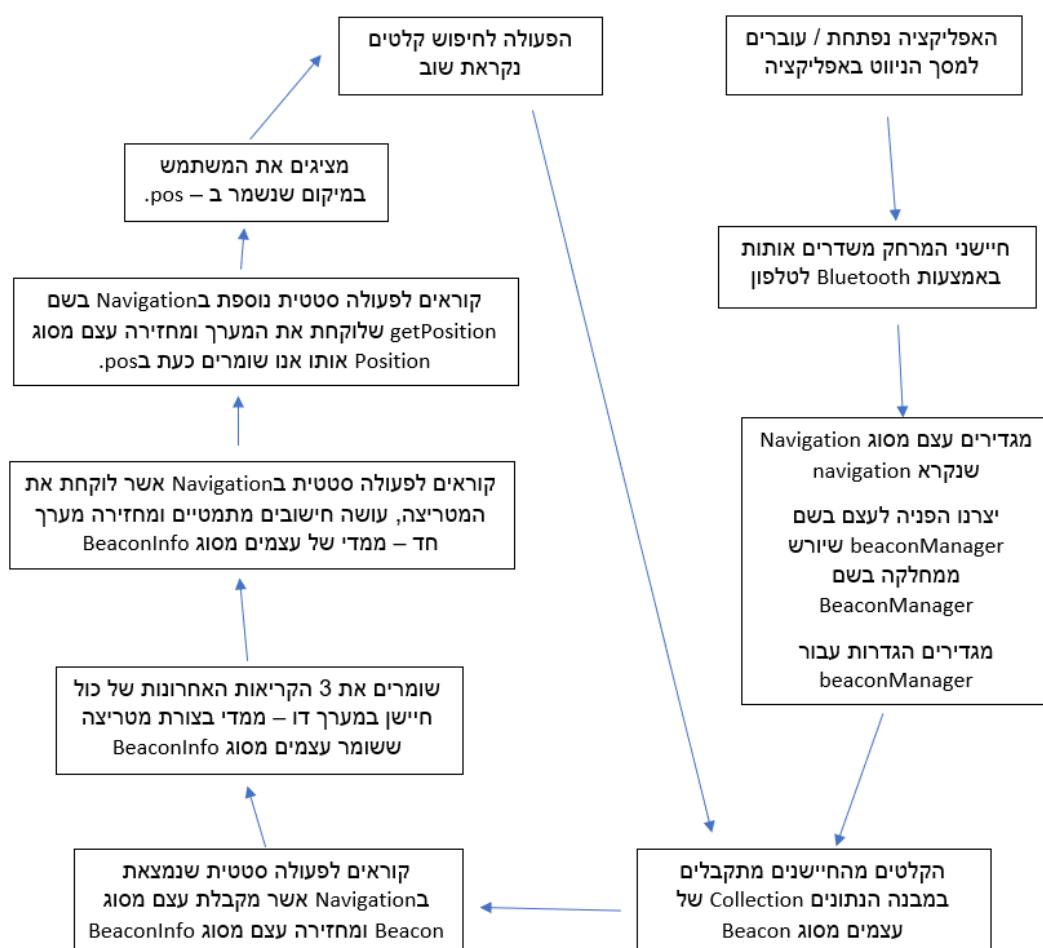
הפעולה עושה חישובים מתמטיים ומתחשבת ביחסים בין מרחקי החיישנים ומחזירה עצם מסוג Position אותו אנו שומרים כעת ב-pos. לאחר שעשינו כל זה, אנו בודקים אם נבחרה חנות לנווט אליה או לא. אם כן נבחרה חנות אנו מחשבים את מרחק ה-pos, כלומר מיקום המשתמש ביחס למיקום החנות במטרים או ברגל, תלוי ביחידות המידה שהמשתמש בחר ומציגים זאת במסך.

בנוסף, יש קריאה לפעולה סטטית אשר מקבלת את מיקום החנות ואת מיקום המשתמש ומחזירה את הכיוון אשר אליו צריך המשתמש לנווט על מנת להגיע לחנות.

אנו מעדכנים את המפה כך שהיא תציג את המשתמש באמצע המסך והחיצים עבור ניווט מוצגים בהתאם לחנות אשר נבחרה. בכך, תם תהליך ניווט של קריאה בודדת, תהליך זה חוזר על עצמו חלילה עד שהמשתמש מפסיק לנווט.

כל המידע שתואר למעלה מתרחש בכמה מחלקות כאשר כל מחלקה מטפלת בחלק אחר ובכך נוצרת זרימת מידע בין המחלקות.

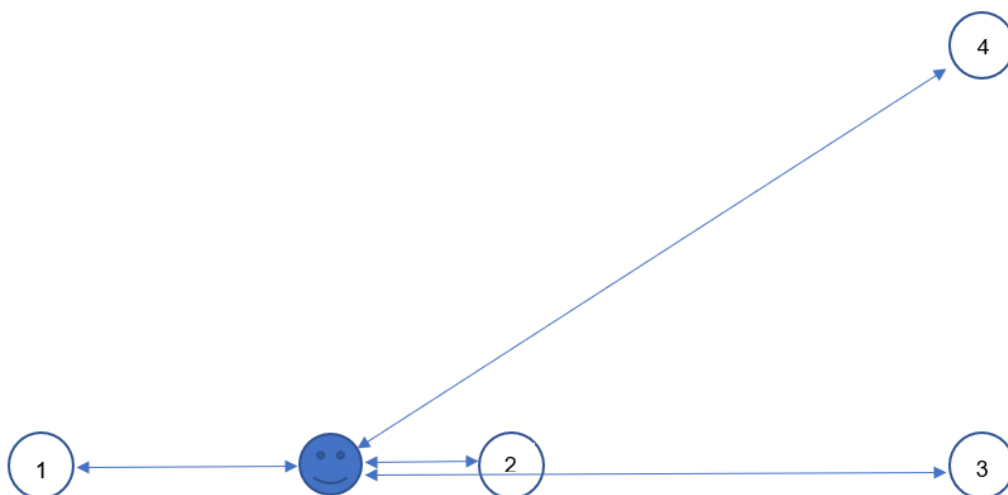
## בתרשים:



## תיאור האלגוריתם הראשי ותיאור אלגוריתמים עיקריים

לפני שנדבר על האלגוריתם עצמו, אסביר תחילה על הרעיון שאני והמנחה שלי חשבנו על כיצד למקם את המשתמש על המפה באמצעות לפחות שני חיישנים:

לאחר מספר של קריאות וחישובים מתמטיים, נשמור מערך שבו יש את המרחקים הממוצעים של כל חיישן מהטלפון של המשתמש. ממערך זה אנו בעצם צריכים רק את שני המרחקים הקצרים ביותר מפני שהם יתארו לנו היכן נמצא המשתמש בין שני חיישנים באופן המדויק ביותר וכך נוכל למקם אותו על המפה, לכן לאחר שנתמקד בשני המרחקים הקצרים ביותר, נמצא את היחס בין המרחק הקצר ביותר לסכום זוג המרחקים הקצרים וכעת נכפיל סכום זה בקבוע כלשהוא שהוחלט מראש שהוא המרחק בפיסקלים בין החיישנים במפה (במקרה שלנו, קבוע האפליקציה הוא 600 פיקסלים של המפה שהיא בגודל 2000 \* 2000). לאחר שעשינו פעולה זו, בעצם קיבלנו את המרחק של המשתמש (בפיסקלים) מהחיישן הקצר. לדוגמה:



אנו משערים כי המשתמש פחות או יותר בקו הישר שבין החיישנים, וגם אם לא, הסטייה של המרחקים לא אמורה להיות גדולה כאשר עושים יחסים בין המרחקים (בנוסף אנחנו מניחים כי יתקינו את החיישנים באמצע המסדרונות לדוגמה בתקרות או רצפות ובכך יחס המרחקים יסתדר עם הקוד). בדוגמה שהבאתי למעלה, ניתן לראות כי המשתמש ממוקם בין החיישן הראשון לשני ולכן המרחקים הקטנים ביותר יהיו המרחקים של החיישן הראשון והשני. כעת, נחלק את המרחק הקצר ביותר (בין המשתמש לחיישן 2) לסכום המרחקים של המשתמש מהחיישנים ונקבל יחס כפי שרשמנו לעיל, את היחס הזה נכפיל ב – 600 ונקבל את המרחק בין המשתמש לחיישן 2. לאחר מכן, בעזרת קוד נמצא היכן ממוקם המשתמש ביחס לחיישן הקרוב ביותר (מימין, משמאל, מעליו או מתחתיו). בעזרת מיקום החיישן השני נוסף או נחסר משיעורי הקואורדינטות של מיקום החיישן הזה את המרחק שחישבנו קודם ובכך נקבל פחות או יותר את מיקום המשתמש על מפה.

דבר נוסף שחשוב לציין הוא שהמשתמש במפה זז רק בקווים ישרים ובפניות של 90 מעלות וזאת על מנת לפשט את האלגוריתם, כי גם לרוב מסדרונות בקניונים ושווקים הם 90 מעלות אחד מהשני.

האלגוריתם שאני הולך לתאר אותו כעת הוא האלגוריתם שרשום במחלקה Navigation בפעולה getPosition. לפני כן, נדבר על המתרחש לאחר שמתקבל קריאה מהחיישן.

כפי שהוסבר בזרימת מידע בין המחלקות השונות בשלבי הניווט, בשדה בשם navigation שהוא עצם מסוג Navigation, שומרים את 3 הקריאות האחרונות מכל חיישן במערך דו – ממדי ששומר עצמים מסוג BeaconInfo. לאחר מכן, אנו קוראים לפעולה סטטית ב – Navigation בשם `getAverageDis` שלוקחת את המערך הדו – ממדי שנזכר שהוא בצורת מטריצה אך לא חייב שכל המקומות בו יהיו עם עצמים (חלק מהמקומות יכולים להיות Null), ויוצרת מערך חד – ממדי של BeaconInfo שמייצג את המרחקים הממוצעים של 3 הקריאות האחרונות. הפעולה סוכמת את סך המרחקים ומחלקת במספר הקריאות שיש לאותו חיישן (מספר העצמים שאינם Null אשר נע בין 0 ל – 3), ובכך מקבלת מרחק ממוצע של אותו חיישן ואז יוצרת עצם חדש מסוג BeaconInfo אותו היא שומרת במקום המתאים במערך שהוגדר בתחילת הפעולה. כך היא עושה לכל חיישן. אם אין קריאות אחרונות לחיישן (כלומר מספר העצמים שאינם Null הם 0) אז העצם שישמר במערך החד – ממדי יהיה עצם עם מרחק 0.

לאחר שהמערך החד – ממדי הזה נשמר בעצם בשם `avgInfo`, קוראים לפעולה הסטטית `getPosition` שרשומה ב-Navigation וכארגומנט, מעבירים לה את `avgInfo`. בפעולה זו רשום האלגוריתם הראשי על מנת למצוא מיקום משתמש במפה והוא מחולק לתת – פעולות וחישובים:

(מעתה והלאה כל פעולה שאני אדבר עליה היא פעולה סטטית במחלקה Navigation אלא אם צוין אחרת)

- תחילה, יש קריאה לפעולה בשם `removeEmptyInfo` אשר לוקחת כארגומנט את המערך החד – ממדי ומחזירה מערך חד – ממדי של עצמי BeaconInfo ללא מרחקי 0 (שהסיבה לכך שיכולים להיות עצמים עם מרחקי 0 הוסברה למעלה) בכך שהיא יוצרת ArrayList של BeaconInfo ועוברת על כל עצם במערך, ורק אם המרחק גדול מ – 0 העצם נשמר ב – ArrayList, ולאחר מכן יש קריאה לפעולה בשם `toArray` שלוקחת ArrayList ומחזירה מערך חד – ממדי.
- לאחר מכן, יש קריאה לפעולה בשם `getClosestBeacons` שמחזירה מערך חד – ממדי בגודל 2 של עצמים מסוג BeaconInfo ששומרת בתוכה את שני העצמים עם המרחקים הכי קטנים (במיקום ה – 0 ממוקם החיישן הקרוב יותר).
- לוקחים את שני המרחקים הללו שנמצאים בעצמים שבמערך החד – ממדי בגודל 2 ומחלקים את המרחק הקצר ביותר בסכום זוג המרחקים ובכך מקבלים יחס של איפה נמצא המשתמש בין שני החיישנים כפי שהוסבר ברעיון למעלה.
- מכפילים יחס זה בקבוע האפליקציה על מנת לקבל את המרחק בפיקסלים בין המשתמש לחיישן הקרוב ביותר.
- ניצור עצם מסוג Position בשם `pos` ונקרא לפעולה בשם `indexToPosition` שלוקחת כארגומנט את המספר הסידורי של החיישן הקרוב ביותר למשתמש (ע"י כך שניקח

את המספר הסידורי של העצם במקום ה-0 (במערך). הפעולה מחזירה את המיקום של החיישן במפה והוא נשמר ב – pos.

- לאחר מכן, יש קריאה לפעולה בשם `getDirection` שמקבלת את המערך שמכיל את זוג החיישנים הקרובים ביותר ומשתמשת במספר הסידורי של החיישנים על מנת להחליט לאיזה כיוון נוטה המשתמש. בדוגמה שהראיתי למעלה, המשתמש קרוב יותר לחיישן השני מאשר לראשון ולכן הכיוון יהיה שמאלה (כלומר, משמאל לחיישן הקרוב ביותר שהוא 2 במקרה זה). כיוון מוצג ע"י מספר כך שהמספר 0 מייצג צפון, 1 מזרח, 2 דרום, 3 מערב.
- כעת יש קריאה לפעולה אחרונה בשם `add` שהיא פעולה שלא מחזירה כלום. פעולה זו לוקחת כארגומנט את הכיוון שמצאנו, pos (כזכור, המיקום של החיישן הקרוב ביותר), והמרחק בין החיישן הקרוב ביותר למשתמש. הפעולה בודקת מהו הכיוון שהאלגוריתם בחר, וקוראת לפעולה של pos ומחסרת או מוסיפה אופקית או אנכית את המרחק שבין החיישן הקרוב ביותר למשתמש ובכך משנה את המיקום למיקום המשתמש במפה.
- הפעולה `getPosition` מחזירה את pos.

ובכך תם תהליך חישוב מיקום לאחר קריאה אחת נוספת של החיישנים, תהליך זה חוזר חלילה בכל קריאה חדשה מהחיישנים.

## **הצגת מקרה עבור הפונקציות העיקריות בפרויקט**

כפי שכבר ידוע עד עכשיו, הניווט הוא החלק העיקרי בפרויקט. ניתן לראות שהמשתמש נכנס לאפליקציה, אם הוא כבר רשום הוא עובר למסך המפה שם הוא יכול לראות היכן הוא נמצא על המפה. אם יבחר לנווט לחנות מסוימת יוצג בפניו אומדן של מרחקו מהחנות וחץ שיכוון אותו לכיוון החנות.

כאשר המשתמש יתחיל להתקדם / להתרחק מכיוון החנות ישתנה מיקומו על המפה בהתאם וכך האומדן מרחק ישתנה וגם לעיתים ישתנה החץ המוביל לכיוון החנות.

רק לאחר שהמשתמש בחר לנווט לחנות מסוימת תהיה לו אפשרות לפתוח דיאלוג ששם ניתן לראות 3 אפשרויות חדשות למסכים שבהם המשתמש יכול לבחור והם: מסך החנות, מסך ביקורות ומסך הנחות.

במסך החנות יוכל המשתמש לראות פרטים אודות החנות ולהתקשר אליהם במקרה הצורך, במסך הביקורות יוכל המשתמש לראות ביקורות של משתמשים שרשמו על החנות והדירוג שלהם לחנות וגם לרשום ביקורת ובמסך ההנחות המשתמש יוכל לראות הנחות שיש לו בחנות אליה הוא מנווט. לדוגמה אם יש לו הנחה על יום הולדת בחנות יופיע שם כרטיסיה אשר המשתמש יוכל לבחור ולהראות לעובד בחנות ובצד למטה יש אפשרות למחוק את

הכרטיסיה של ההנחה. האפשרות למחוק היא אפשרות של המוכר לאחר שהוא נותן את ההנחה למשתמש.

## **טכנולוגיות שנעשה בהן שימוש בפרויקט**

החיישנים שמשמשים עבור הפרויקט הם חיישנים של accent systems בשם IBKS Plus. חיישנים אלה. לפי אתר החברה יכולים לקלוט אותות עד מרחק של 100 מטר ויכולים להיות בשימוש עד 104 חודשים. החיישנים מדייקים רב במרחקים של כ – 10 מטר אך יעילותם לפרויקט שכזה מאוד גבוהה, גם בשל המרחק בו הם יכולים לקלוט באופן יעיל, מפני שקניונים ושווקים הם מקומות גדולים יחסית עם מסדרונות שמשתרכים על כמה עשרות מטרים ולכן שימוש בחיישני מרחק שיכולים לקלוט ממרחקים כאלה מצמצם את מספר החיישנים הנחוץ ובכך מצמצם עלויות עבור הצד המנהל את הקניון או השוק.

המנחה שלי הביא לי שלושה חיישנים לעבוד איתם כך שיהיה אפשרי למקם במפה את המשתמש אך יכולים להיות מספר לא מוגבל של חיישנים על מנת למקם ולהשתמש בניווט של הפרויקט.

החיישנים משדרים בעזרת אותות Bluetooth ובעזרת חישובים מתמטיים שנקבעו בממשקים שפתוחים לשימוש על ידי כולם בשם AltBeacon. החיישנים, כפי שהוסבר לעיל, ממוקמים לרוב במרחק של כ – 10 עד 15 מטרים זה מזה ובכך ניתן להשתמש באלגוריתם שתואר קודם לכן על מנת למקם את המשתמש.

## **ממשקים למערכות אחרות**

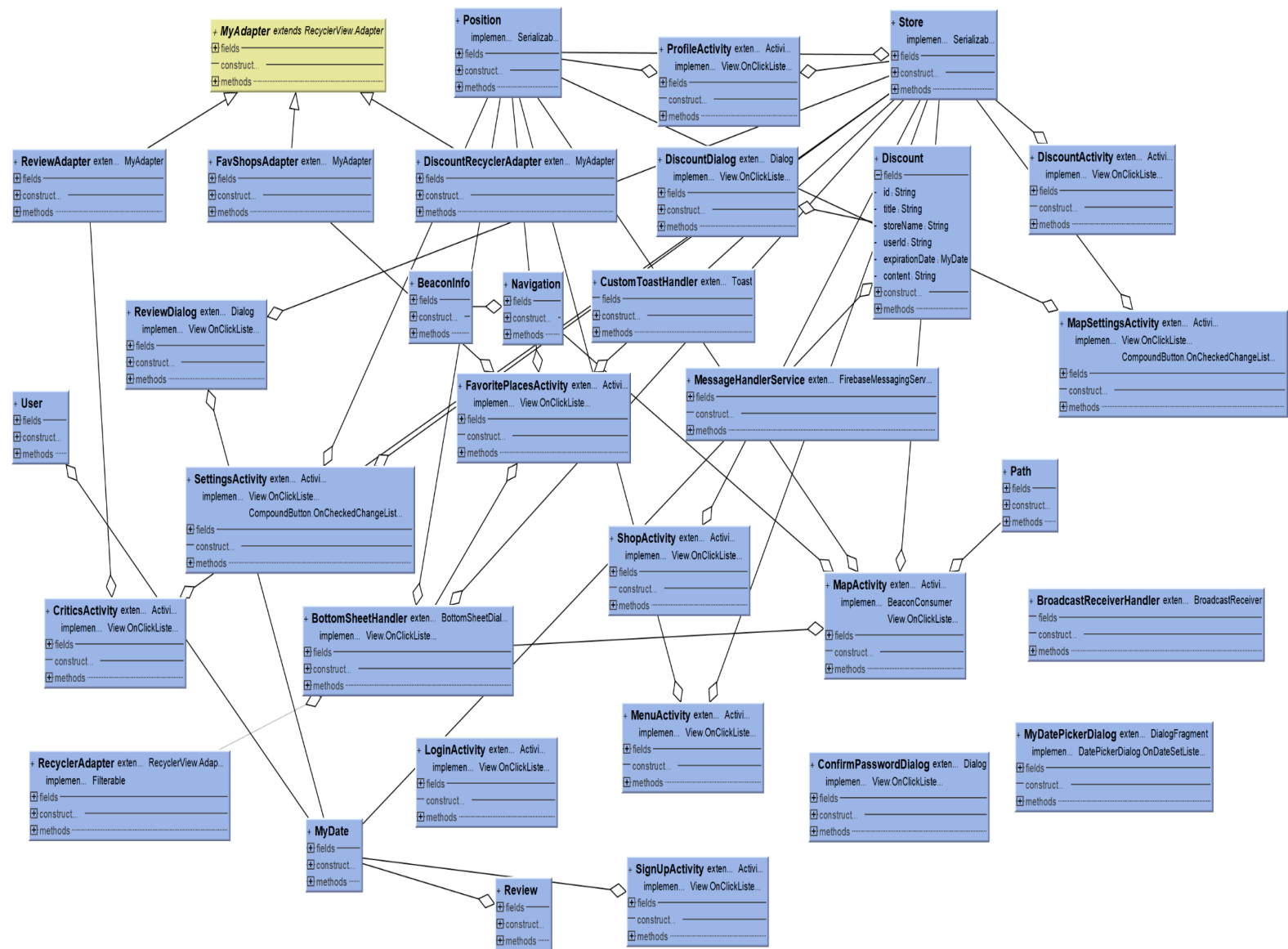
הממשק שהשתמשתי בו בפרויקט נקרא AltBeacon. הוא ממשק שמקשר בין החיישנים המשושמים בפרויקט לבין האפליקציה. הוא לקוח מספריה של Android שמספקת ממשק לאינטראקציה עם Beacons (חיישנים למיניהם).

בפרויקט, אנו משתמשים באימפלמנטציות מסוימות של ה – AltBeacon API ומתאימים אותם לצרכים של הפרויקט שלנו.

הממשק הזה משמש על מנת להתחבר לחיישנים של accent systems ובפרויקט אנו משתמשים בו על מנת לחשב אומדן של מרחקים ובכך לנסות לפתור את בעיית הניווט באפליקציה.

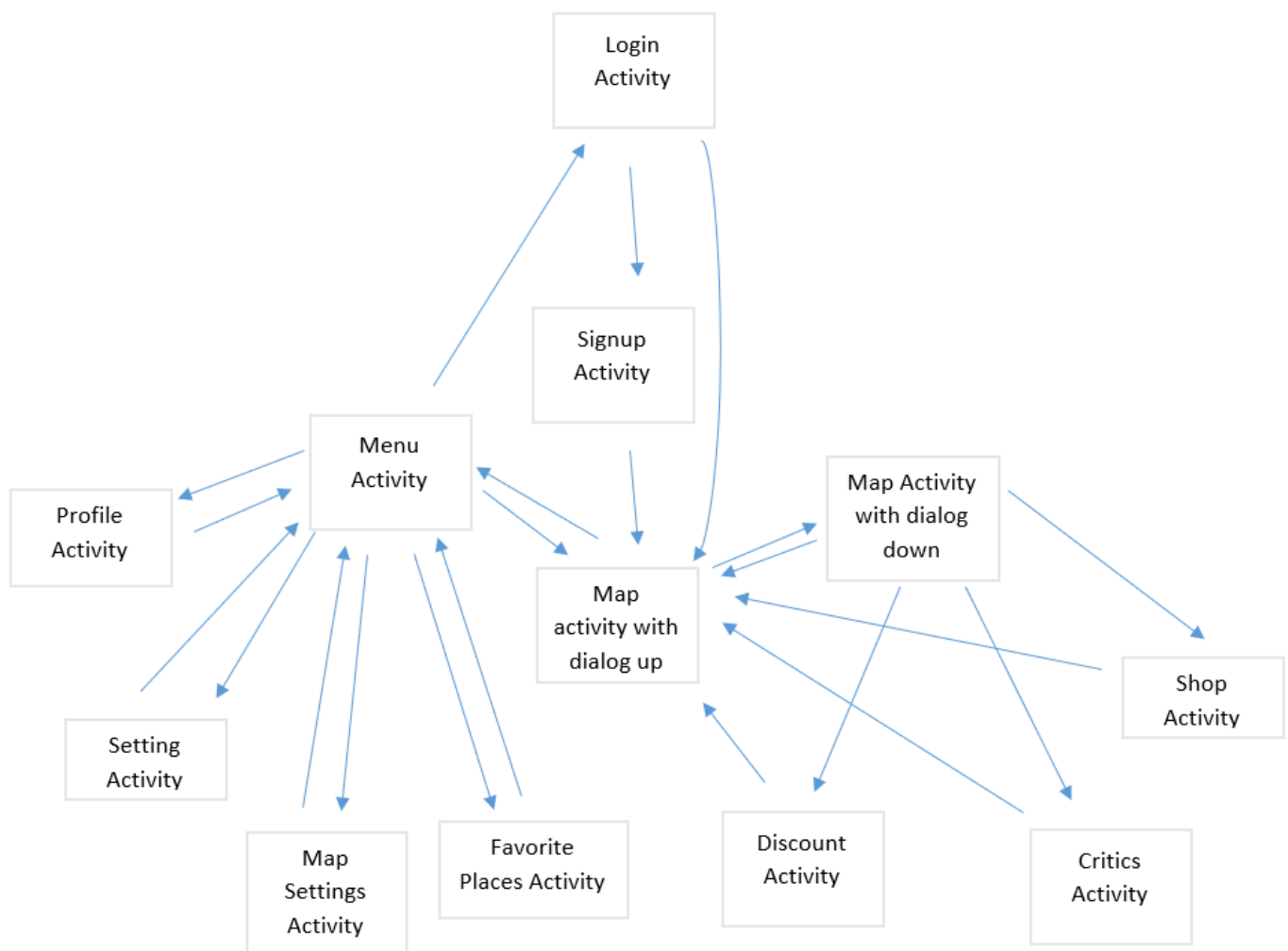
בממשק ניתן לראות פעולות רבות שאפשר להשתמש בהם, אך, הפעולות העיקריות שאנו היינו צריכים לפרויקט הן רק קליטה של המערך Beacons שמשתתפים בפרויקט ועליהם לבצע פעולות מתמטיות וחישוביות שיצרנו. בממשק, כפי שתואר למעלה, **בזרימת מידע בין המחלקות השונות בשלבי הניווט** הגדרנו כי כל 100 מאיות השנייה האפליקציה תחפש קריאות מהחיישנים של מרחק.

## תרשים UML



## מדריך למשתמש

### תרשים מסכים (Screen flow diagram)





## מסך התחברות

### הסברים:

במסך זה יש אפשרות להיכנס לאפליקציה בעזרת השדות של הזנת אימייל וסיסמה. קיימת אפשרות גם על ידי לחיצה על סימן העין בשדה הסיסמה לראות את הסיסמה שהשתמש הזין במידה וירצה. לחיצה נוספת תסתיר בחזרה את הסיסמה.

לאחר שהשתמש הזין אימייל וסיסמה בתנאי שהם עומדים בתנאים של אימייל תקין וסיסמה מעל 6 תווים, הוא יכול ללחוץ על כפתור ההתחברות **LOG IN** על מנת להיכנס לאפליקציה.

במידה והשתמש אינו רשום הוא יכול ללחוץ על הכפתור **Sign Up** על מנת לעבור למסך ההרשמה לאפליקציה.

במידה והשתמש כן הזין אימייל וסיסמה אך הפניה לשרת זיהתה כי יש שגיאה באימייל או בסיסמה תוצג ההודעה:

במידה ולא הזין פרטים תוצג לו ההודעה הזאת:

## מסך הרשמה

### הסברים:

כפתור חזרה למסך ההתחברות

קיימים 4 שדות להזנת נתונים. הראשון הוא שם המשתמש אשר אין עליו הגבלות. השניים הנוספים הם הסיסמה של המשתמש שצריכה להיות באורך של 6 תווים לפחות והשדה רביעי הוא אימייל שצריך להיות אימייל תקין (סיומת נכונה וכו').

לחיצה על כפתור העין תראה את הסיסמה.  
לחיצה נוספת תסתיר אותה.

יש אפשרות לבחור תאריך לידה על ידי לחיצה על הכפתור **DATE OF BIRTH** כך שיפתח דיאלוג מסוג MyDatePickerDialog על מנת לבחור תאריך.

ניתנת אפשרות להזין מגדר על ידי המשתמש בלחיצה על אחד הכפתורים העגולים.

כפתור הרשמה לאפליקציה. אם ההרשמה מוצלחת, המשתמש יעבור למסך המפה.

## המשך הסברים:

אם המשתמש הזין אימייל לא תקין (בלי תווים לפני ה- @ / אין @ בכלל / אין נקודה אחרי ה- @) תוצג לו ההודעה הבאה:



אם המשתמש לא הזין אימייל תוצג לו ההודעה הבאה:



אם המשתמש לא הזין שם משתמש תוצג לו ההודעה הבאה:



אם המשתמש הזין סיסמה שאינה לפחות 6 תווים תוצג ההודעה הבאה:





אם הסיסמאות לא תואמות תוצג ההודעה הבאה:

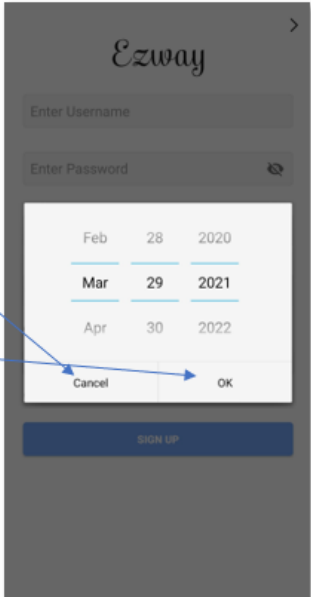
אם האימייל כבר נמצא בשימוש על ידי משתמש אחר תוצג ההודעה הבאה:

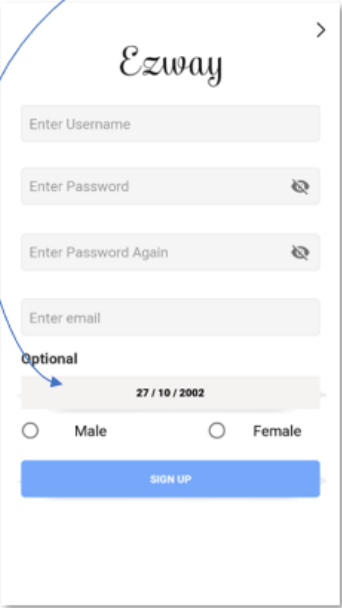
## המשך הסברים:

דיאלוג בחירת התאריך  
שנפתח לאחר לחיצה על  
**DATE OF BIRTH** כפתור

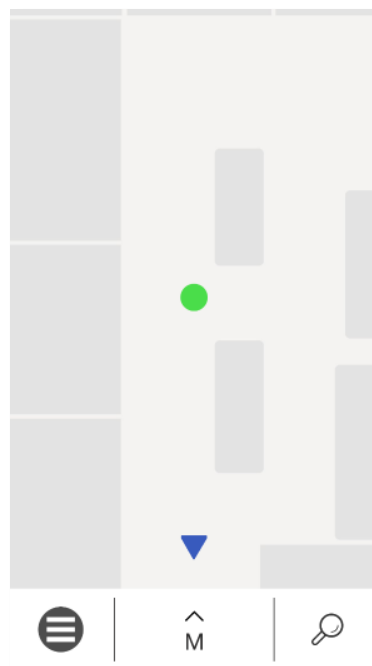
כפתור לסגירת הדיאלוג  
כאשר לא נבחר אף תאריך

כפתור לסגירת הדיאלוג כאשר  
נבחר התאריך שממוקם באמצע  
לאחר בחירת תאריך, על  
הכפתור יהיה התאריך שנבחר  
כך:





## מסך מפה

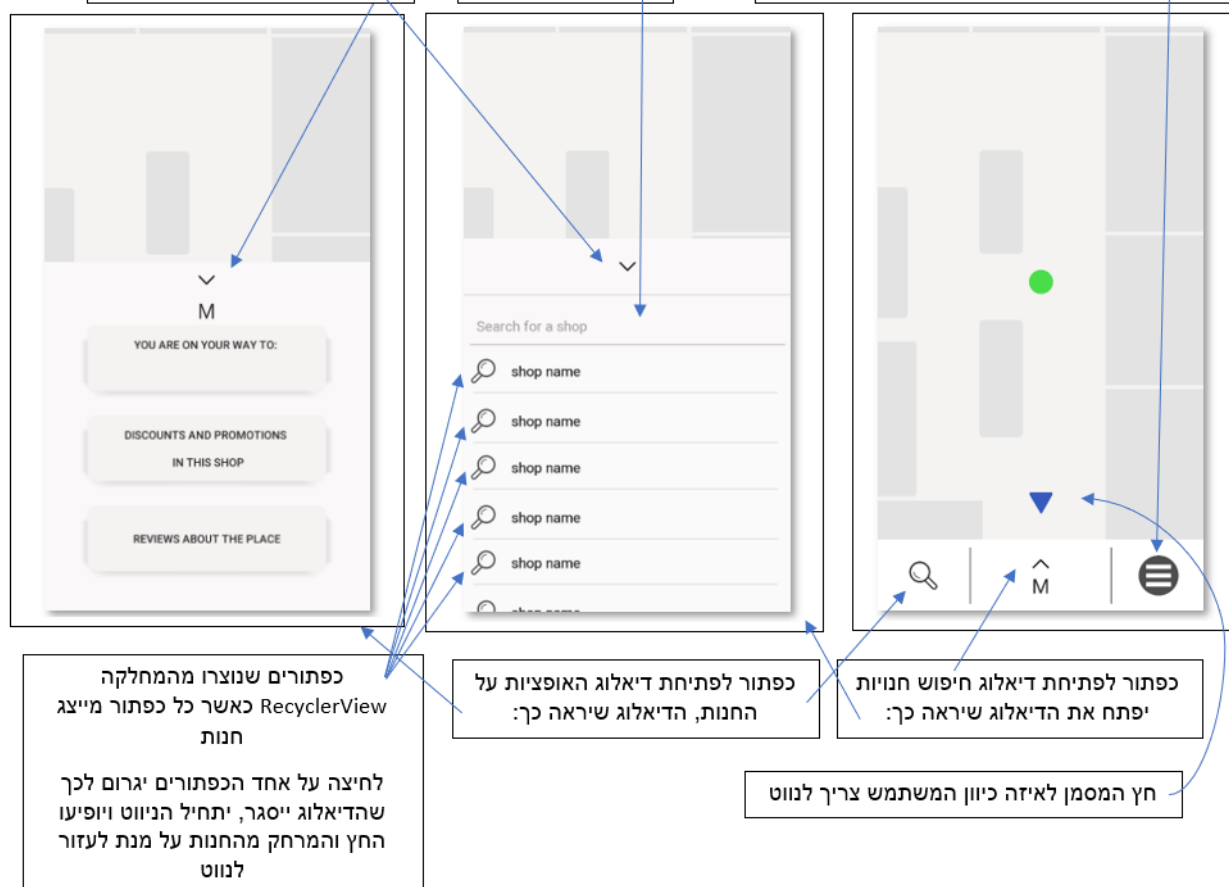


## הסברים:

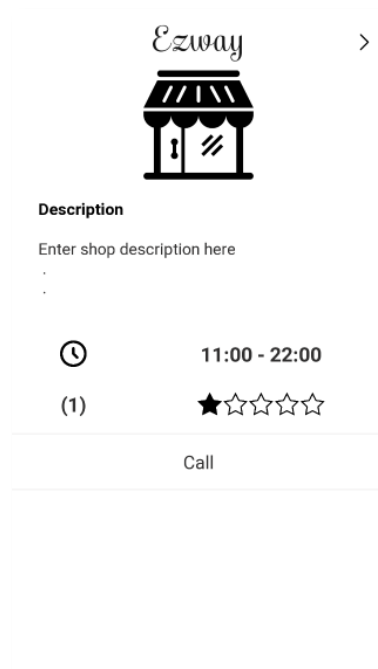
**כפתור להורדת הדיאלוג בחזרה**

שדה לחיפוש חנות

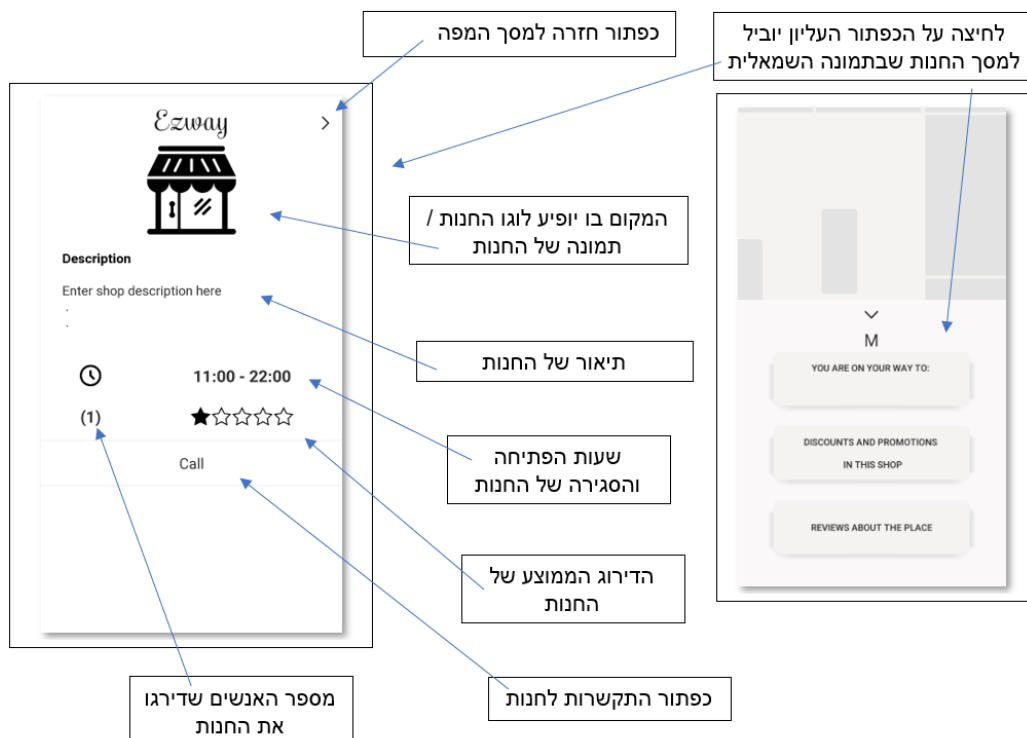
כפתור מעבר למסך תפריט ראשי (Menu Activity)



## מסך חנות



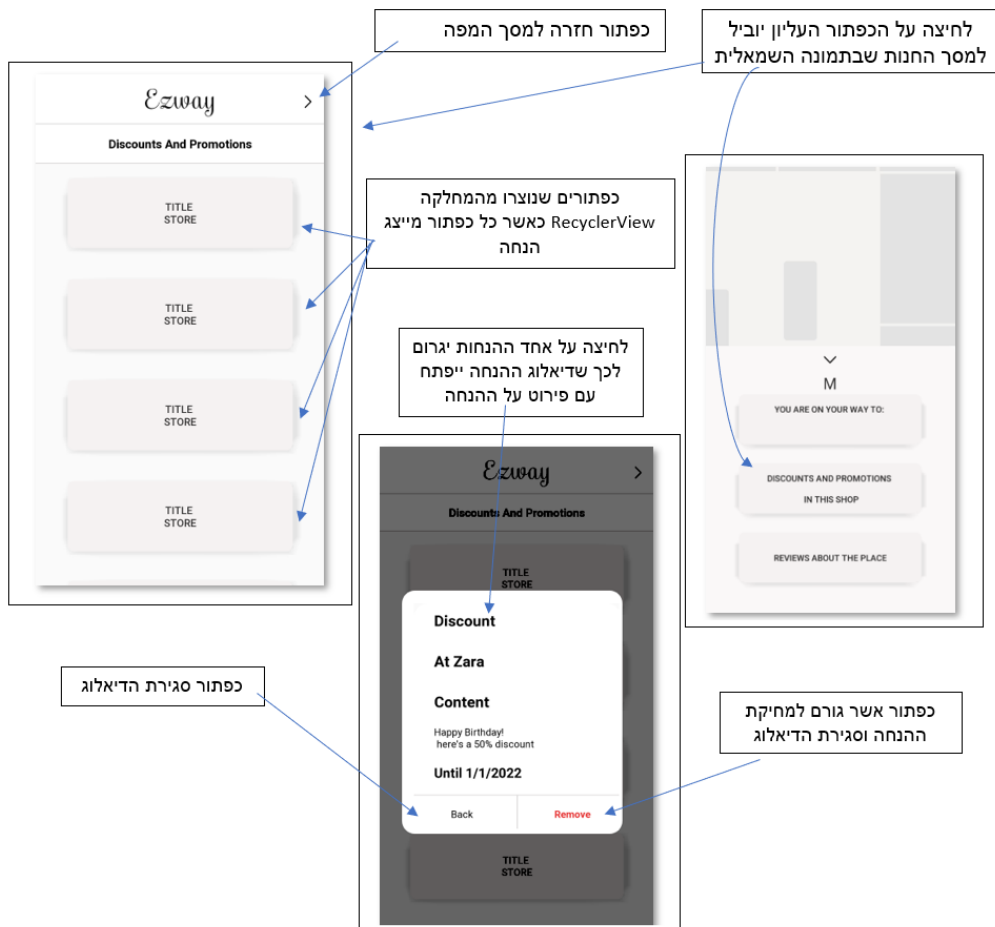
### הסברים:



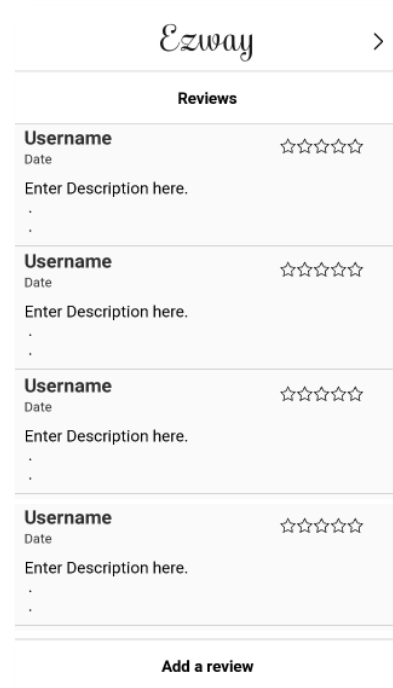
## מסך הנחות



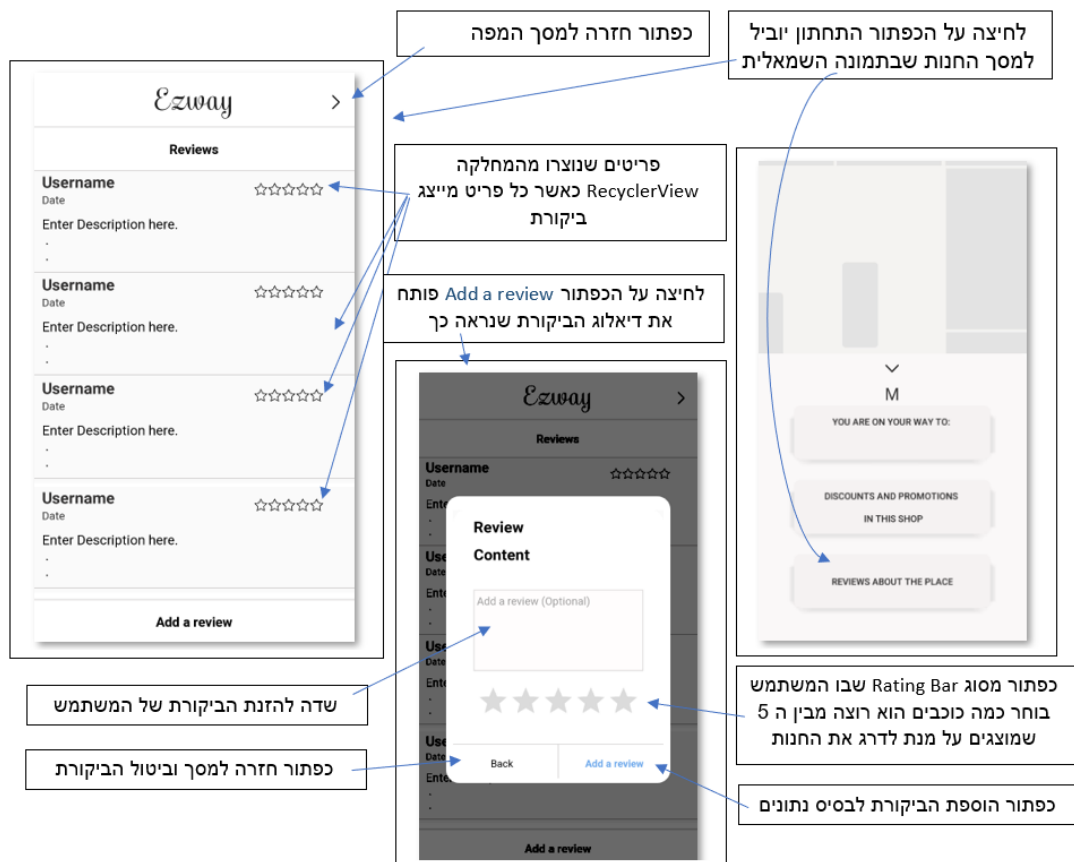
### הסברים:



## מסך ביקורות

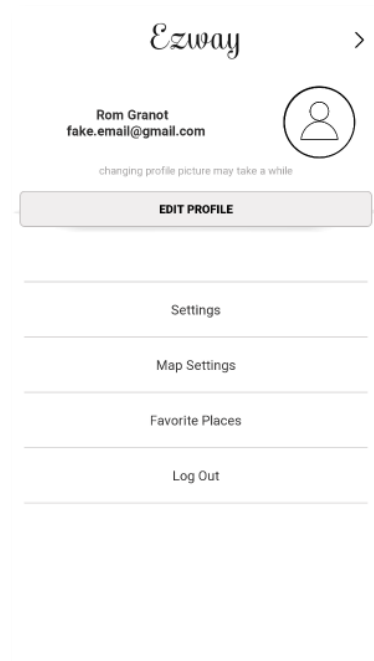


### הסברים:

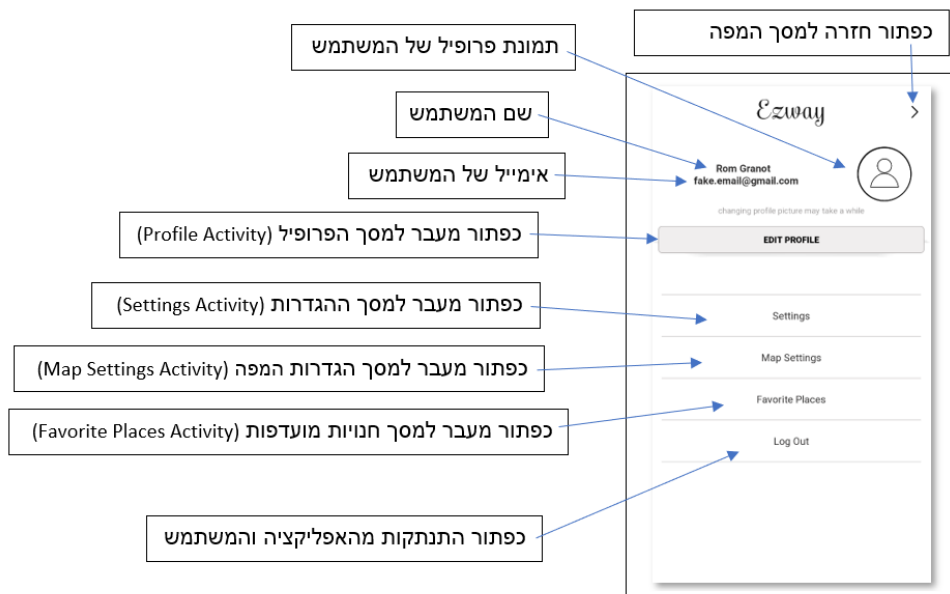




## מסך תפריט ראשי




### הסברים:



## מסך פרופיל

Cancel
Edit Profile
Done



changing profile picture may take a while

Change Profile Photo

Username

Email

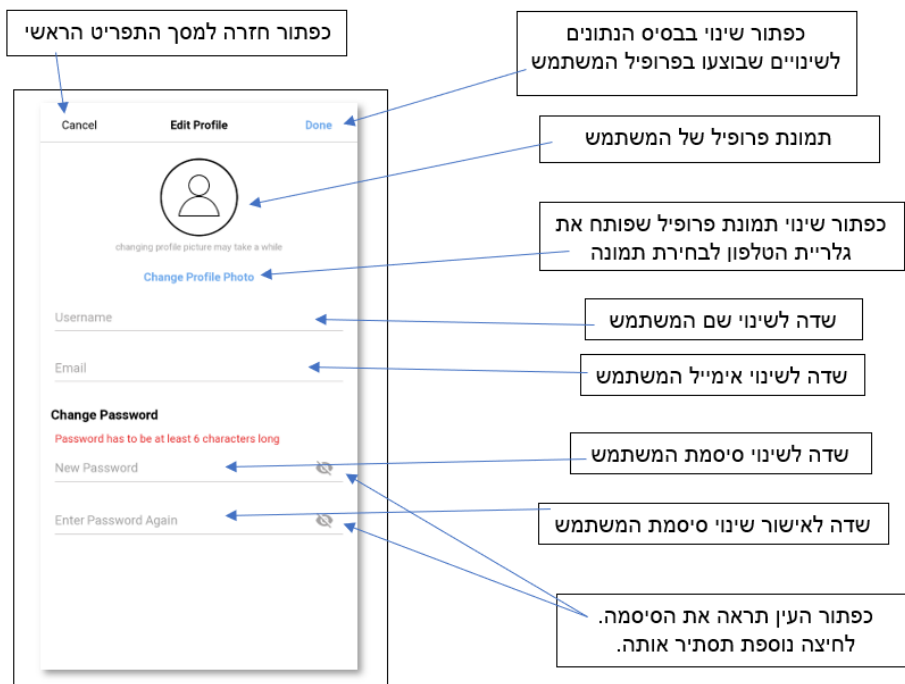
**Change Password**

Password has to be at least 6 characters long

New Password

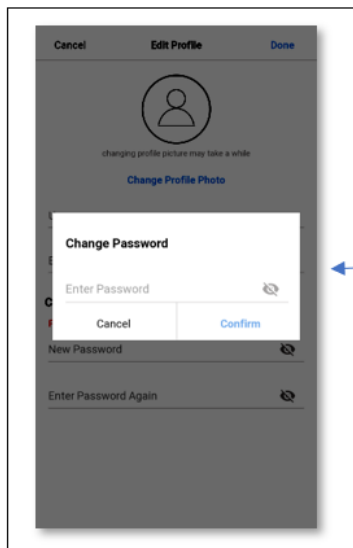
Enter Password Again

### הסברים:



## המשך הסברים:

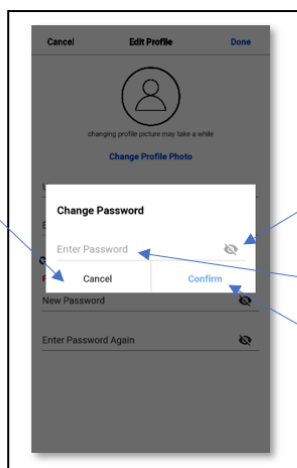
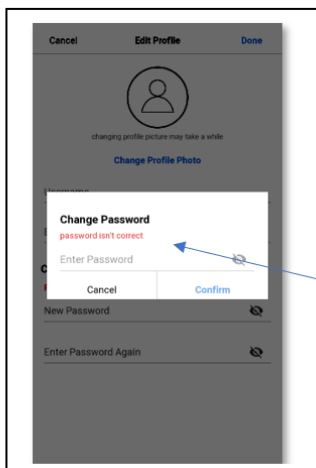
כאשר נכנסים למסך הפרופיל, יש אפשרות לבחור תמונה מהגלריה ולהעלות אותה לבסיס הנתונים בתור תמונת פרופיל. אם רוצים לשנות שם משתמש, שוב, כמו בהרשמה אין הגבלה כל עוד השם משתמש לא ריק. אם רוצים לשנות אימייל, האימייל צריך להיות תקין, לא ריק וגם לא קיים בבסיס נתונים. בנוסף, אם רוצים לשנות סיסמה אז צריך להזין את הסיסמה החדשה פעמיים וגם לוודא שהסיסמה היא עם יותר מ- 6 תווים.



אם רוצים לעשות שינוי כל שהוא בבסיס נתונים של שינוי סיסמה, אימייל (אך לא שינוי תמונה או שם משתמש) אז כאשר המשתמש לוחץ על הכפתור **Done** בצד ימין למעלה יפתח לו דיאלוג והמסך יראה כך:

בדיאלוג זה יש בקשה להזין את הסיסמה הישנה שוב מפני שבסיס הנתונים, firebase מבקש ריענון של סיסמה בזמן האחרון (כמה דקות אחרונות) על מנת לשנות פרטים כמו אימייל או סיסמה.

כפתור הסוגר את הדיאלוג ללא שום פניות לבסיס נתונים



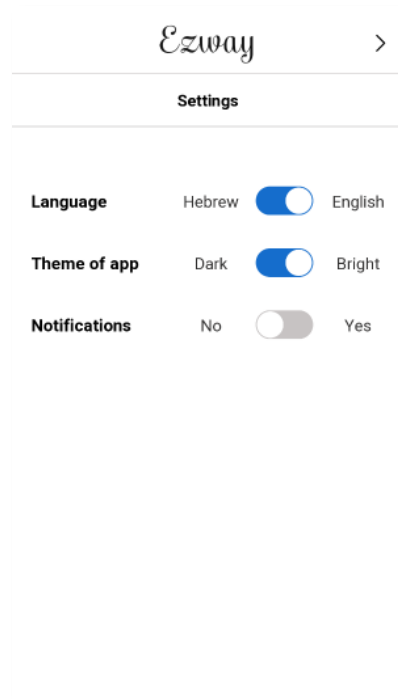
כפתור העין תראה את הסיסמה. לחיצה נוספת תסתיר אותה.

שדה להזנת סיסמה

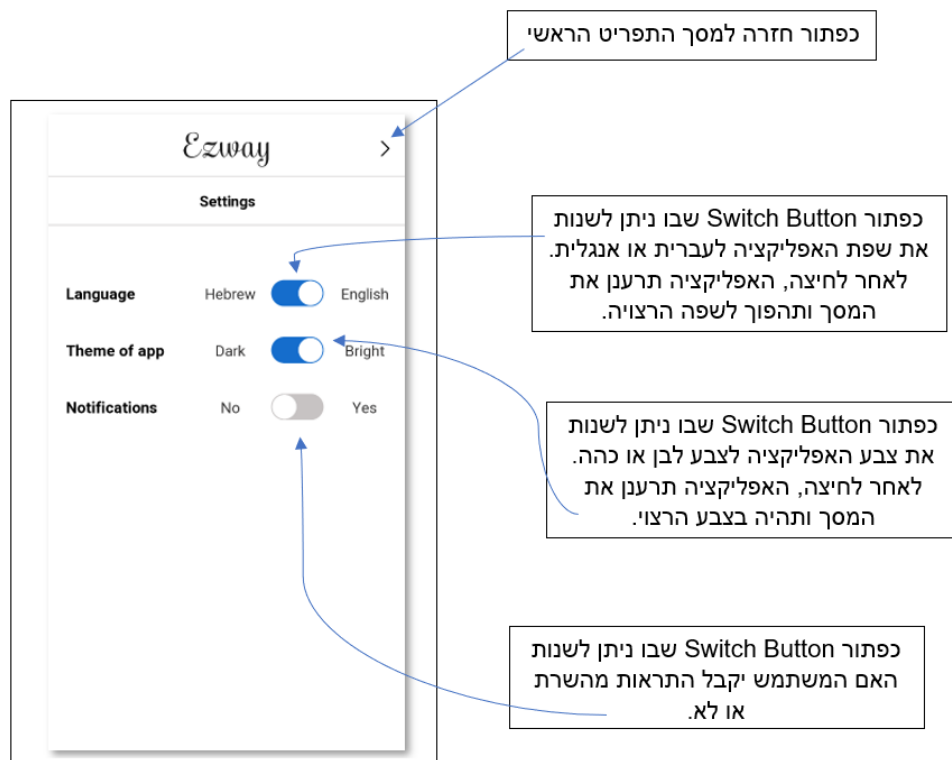
כפתור לבדיקת הסיסמה

הסיסמה נבדקת על ידי בסיס הנתונים. אם היא נמצאת מתאימה לסיסמת המשתמש השמורה שם, הדיאלוג נסגר וניתן ללחוץ על כפתור **Done** אם הסיסמה אינה מתאימה, תוצג ההודעה הבאה:

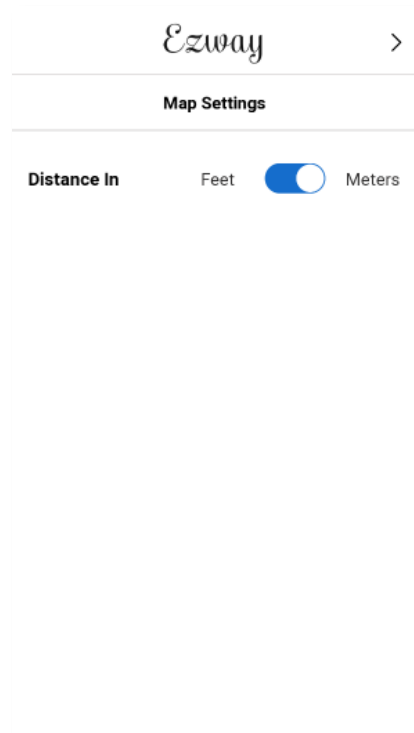
## מסך הגדרות



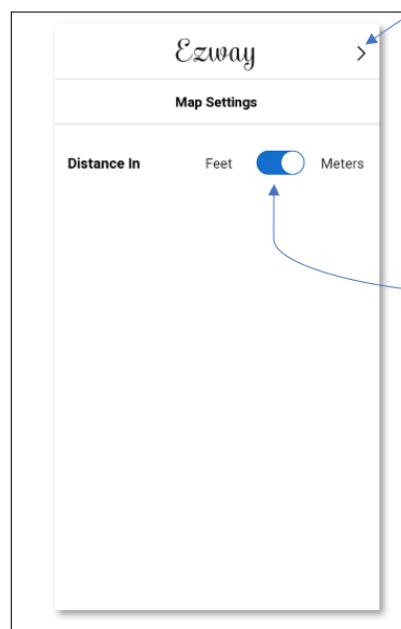
הסברים:



## מסך הגדרות מפה



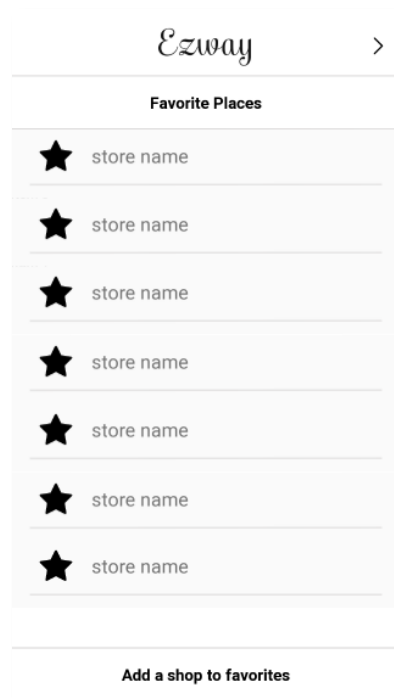
### הסברים:



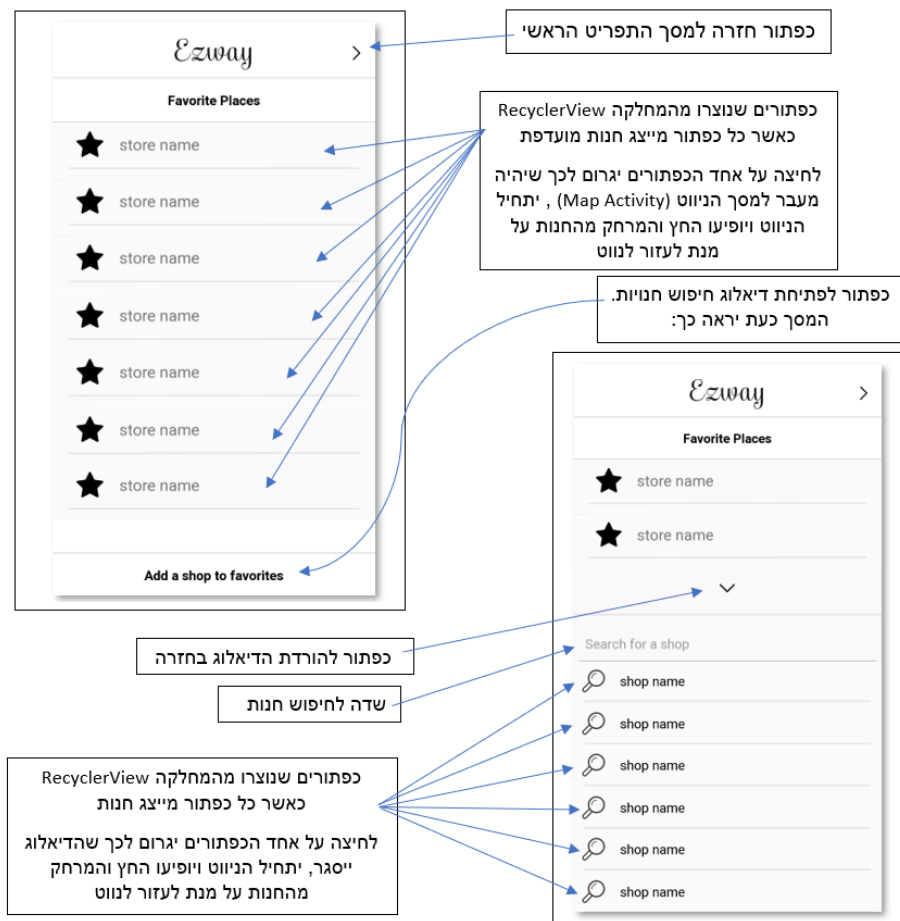
כפתור חזרה למסך התפריט הראשי

כפתור Switch Button שבו ניתן לשנות האם המרחק מהחנות יוצג במידת המרחק מטר או רגל.

## מסך מקומות מועדפים



### הסברים:



## בסיס הנתונים

בסיסי הנתונים שהפרויקט עושה בהם שימוש הם Firebase – I Shared Preferences.

### Shared Preferences

בסיס נתונים זה שומר את המידע בצורת זוגות של מפתח – ערך. הנתונים נשמר בתוך הטלפון בקובץ שמכיל את הזוגות האלה. בעזרת ה- Shared Preferences ניתן לשמור נתונים ולגשת אליהם בשלב מאוחר יותר ללא חיבור לאינטרנט ובכך ניתן לפנות לנתונים בזמן סינכרוני. בבסיס נתונים זה לרוב שומרים מידע פשוט ולא רב שעדיף שיהיה שמור על הטלפון ולא בצד שרת.

בפרויקט, נשתמש ב- Shared Preferences עבור ההגדרות והגדרות המפה שהמשתמש בוחר באפליקציה. נתונים אלה הם נשמרים בצורה בוליאנית.

לדוגמה, נשמור בבסיס הנתונים זוג מפתח – ערך כאשר הערך הוא בוליאני שמסמל האם המשתמש רוצה מסכים בהירים או לא. אם הוא רוצה מסך בהיר, כאשר נשלף מידע זה נוכל לדעת שכאשר האפליקציה טוענת את המסך נבחר בצבע הבהיר ולא הכהה עבור המסך (דבר זה מתרחש מאוד מהר וזו גם אחת הסיבות שבגללן נרצה לשמור את המידע בבסיס נתונים שניתן לשלוח בזמן סינכרוני).

כך נעשה לכל האפשרויות שקיימות בהגדרות והגדרות המפה והן:

- שפת האפליקציה (true – אנגלית, false – עברית)
- צבע הרקע (true – בהיר, false – כהה)
- קבלת הודעות (true – כן לקבל, false – לא לקבל)
- יחידת מרחק (true – מטר, false – רגל)

נשמור את המידע בבסיס הנתונים בכך שניצור הפניה לקובץ שדובר עליו מקודם. לאחר מכן, ניתן ליצור עצם אשר מזין נתונים לתוך הקובץ. בשביל לשלוח, נשתמש בהפניה שהזכרנו מקודם ונשלוח בעזרת השם ששמרנו איתנו.

כאשר שולפים צריך גם לתת ערך דיפולטי (default), שאצלי באפליקציה הוא true, למקרה ובו אין ערך שנמצא בקובץ. לכן, כאשר בפעם הראשונה שהמשתמש נכנס לאפליקציה, כל ההגדרות מופעלות – כלומר, האפליקציה באנגלית, במצב בהיר, מאשרת קבלת הודעות ויחידת המרחק היא במטרים.

**בפיירבייס אני משתמש ב – 4 מהאפשרויות אשר פיירבייס תומכת בהם ומעודדת להשתמש:**

- Realtime Database
- Storage
- Authentication
- Cloud Messaging

נבחן כל אחד מהם בנפרד:

## Realtime Database

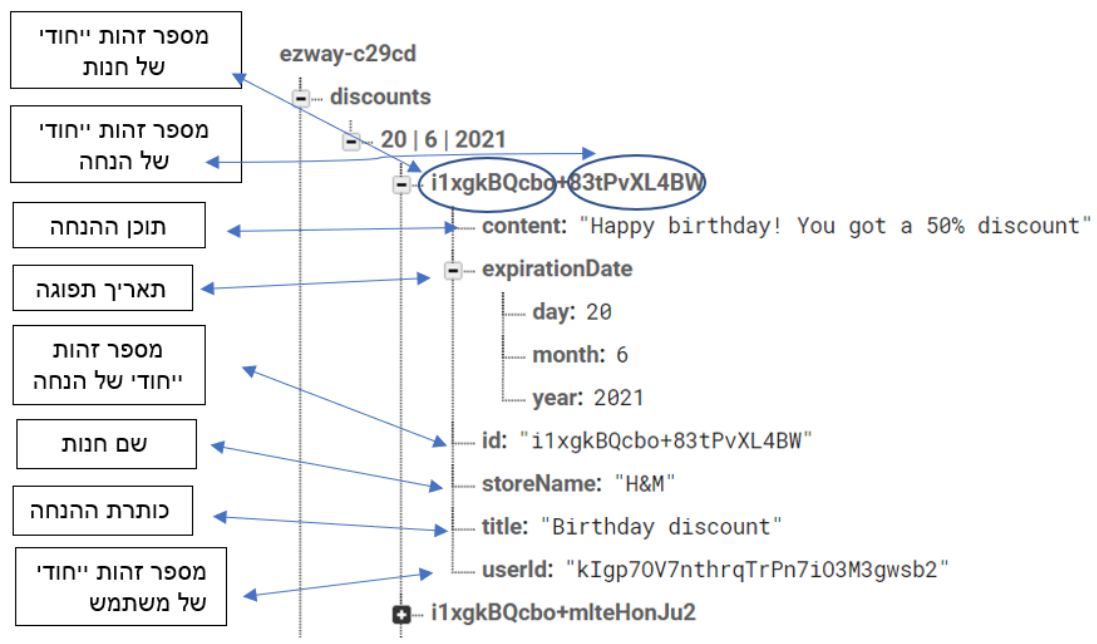
חלק מבסיס הנתונים של פיירבייס אשר שומר את הנתונים שלו בצורת JSON, ובכך בעצם שומר כל מידע באמצעות זוגות של מפתח – ערך. כאן בפרויקט נשמרים 4 סוגים עיקריים של מידע: ההנחות, הביקורות, החנויות והמשתמשים.

**(המשך עמוד הבא)**

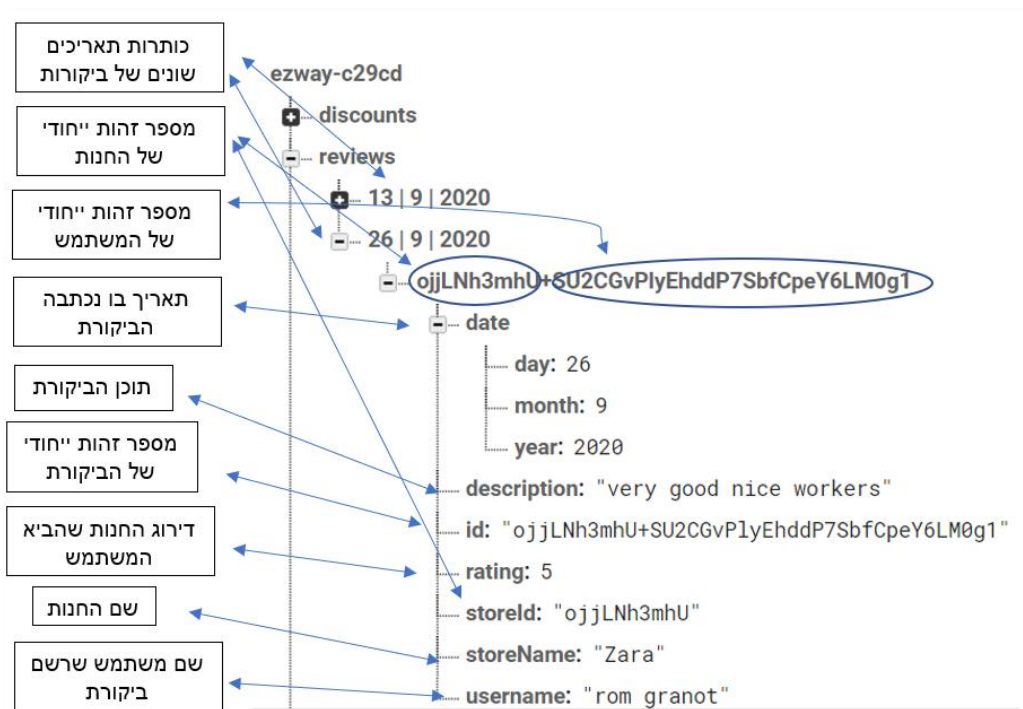


**ההנחות** שמורות מתחת לכותרת התאריכים בהם תוקפם יפוג, וכל הנחה שמורה בשם שהוא מספר הזהות הייחודי של החנות בה יש את ההנחה בצירוף פלוס ואז מספר זהות ייחודי של ההנחה עצמה.

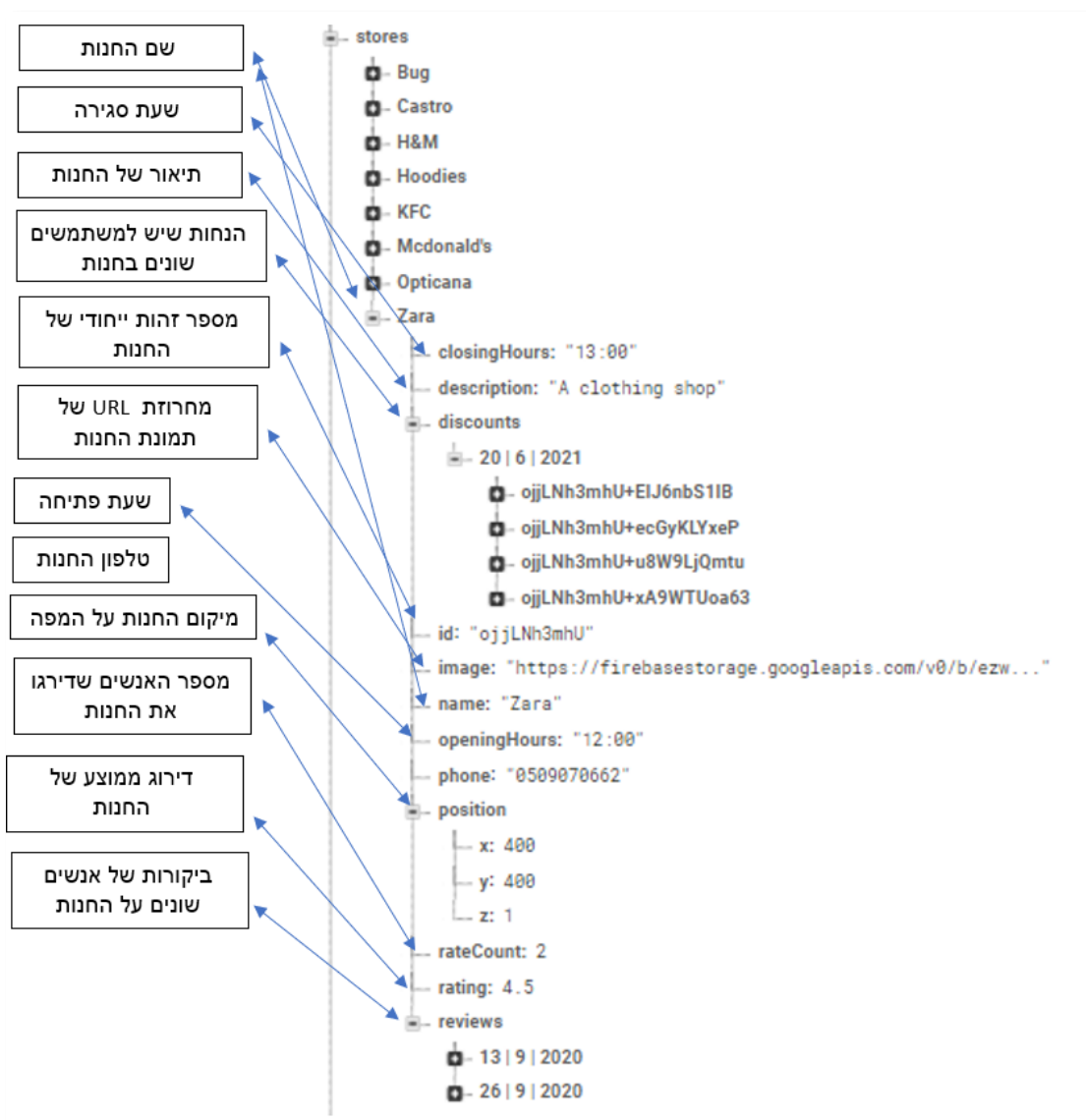
מתחת לכל הנחה יש את כל השדות אשר דיברנו עליהם במחלקה Discount, שהם תוכן ההנחה, תאריך תפוגה, מספר זהות ייחודי, שם החנות, כותרת ההנחה ומספר זהות ייחודי של המשתמש.



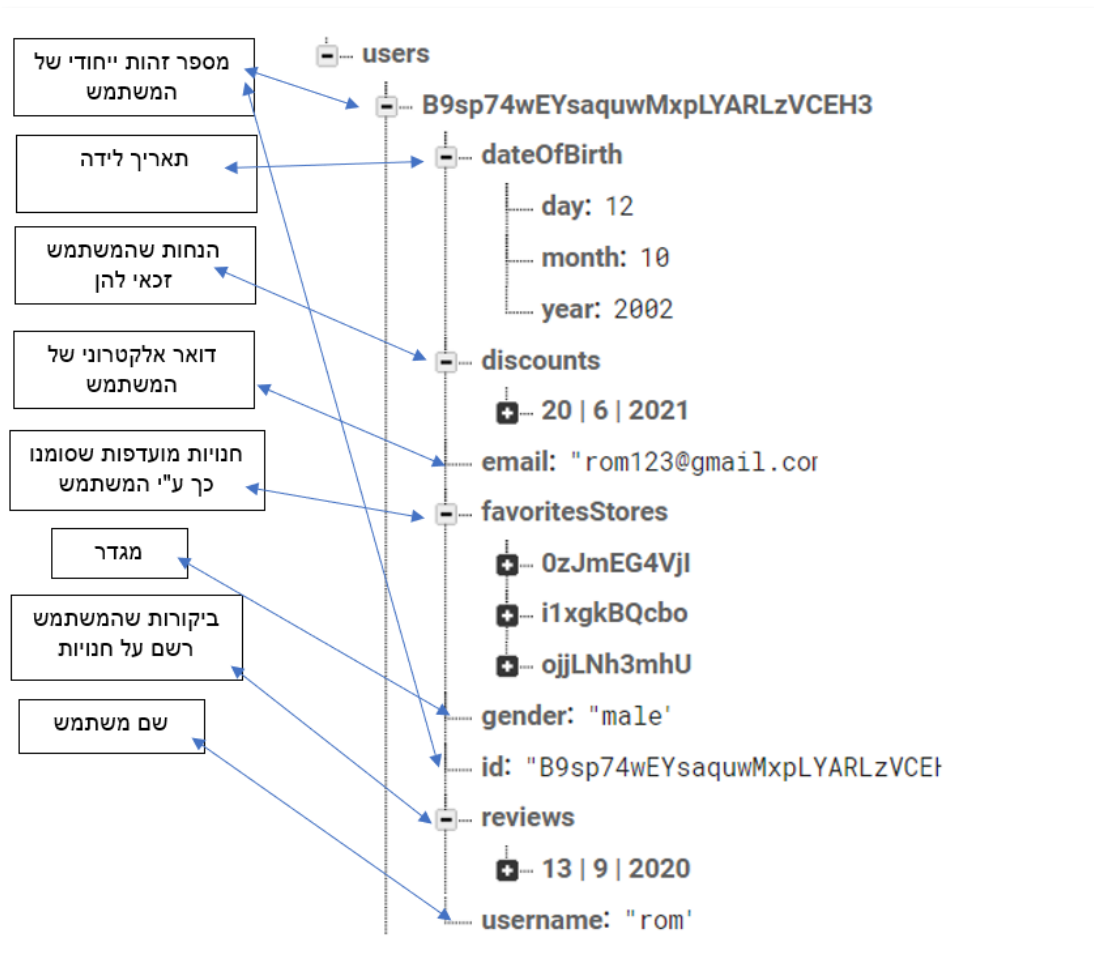
**הביקורות** שמורות מתחת לכותרת התאריכים בהם הביקורות נכתבו, וכל ביקורת שמורה בשם שהוא מספר הזהות הייחודי של החנות בה יש את הביקורת בצירוף פלוס ואז מספר זהות ייחודי של המשתמש שרשם את הביקורת. מתחת לכל ביקורת יש את כל השדות אשר דיברנו עליהם במחלקה Review, שהם תוכן הביקורת, תאריך שבו נכתבה הביקורת, מספר זהות ייחודי של הביקורת, מספר הכוכבים שהמשתמש הביא לחנות, שם החנות, שם המשתמש ומספר זהות ייחודי של החנות.



**החנויות** שמורות מתחת לכותרת של חנויות, וכל חנות שמורה תחת השם שלה. מתחת לכל חנות יש את כל השדות אשר דיברנו עליהם במחלקה Store, שהם תיאור החנות, מספר זהות ייחודי של החנות, תמונת החנות, שם החנות, שעות פתיחה וסגירה, טלפון החנות, מיקום החנות במפה, מספר אנשים שדירגו את החנות, דירוג ממוצע של החנות, ביקורות של החנות והנחות של משתמשים בחנות.



**המשתמשים** שמורים מתחת לכותרת של המשתמשים, וכל משתמש שמור תחת מספר הזהות הייחודי שלו. מתחת לכל משתמש יש את השדות שדיברנו עליהם במחלקה User, שהם מספר זהות הייחודי של המשתמש, שם משתמש, דוא"ל, תאריך לידה אם הוא הזין אחד, מגדר אם הוא הזין אחד, ביקורות לחנויות, הנחות מחנויות וגם חנויות מועדפות.












כאשר רוצים לעשות פניה כלשהיא לשרת ורוצים לגשת לנתונים שנמצא בדאטאבייס יוצרים עצם מסוג `DatabaseReference` שמגדירים אותו להיות הרפרנס של ה-Instance הספציפי של הדאטאבייס באפליקציה. לאחר מכן, ניתן לשלוף עצמים מבסיס הנתונים בקלות יחסית בכך שמקבלים את הנתונים כאובייקטים של מחלקות שמוגדרות אצלנו ולכן הדבר מקל על התהליך של הצגת הנתונים.


## Storage

חלק זה של שמירת הנתונים בעצם נועד עבור שמירת תמונות בשיטה נוחה, זריזה וקלה לשימוש ולארגון שפירבייס יצרו. ב - Storage בפרויקט אנחנו שומרים שני סוגי תמונות: תמונה של חנות באזור בו ממוקמת החנות ותמונת פרופיל של המשתמשים השונים באפליקציה. מסיבה זו, יצרתי שתי תיקיות, כל אחת בשם המתאים לסוגי התמונות שדיברנו עליהן (Shops Logos, Profile Images).

**בתיקייה של Shops Logos יש את הלוגואים ותמונות החנות של כל חנות בנפרד, ושמות התמונות שמורים בשם שהוא שם החנות בשרשור של נקודה ואז המספר זהות הייחודי של החנות.**

<input type="checkbox"/>	Name	Size	Type	Last modified	
<input type="checkbox"/>	 Bug.u5SQL4yBh3	13.8 KB	image/png	Apr 11, 2020	
<input type="checkbox"/>	 Castro.mM6EMpzWH	12.78 KB	image/png	Apr 11, 2020	
<input type="checkbox"/>	 Erroca.uoPvdUa9Ae	4.45 KB	image/jpeg	Apr 11, 2020	
<input type="checkbox"/>	 H&M.i1xgkBQcbo	34.77 KB	image/png	Apr 11, 2020	
<input type="checkbox"/>	 Hoodies.PCefE7JfK2	45.26 KB	image/jpeg	Apr 11, 2020	
<input type="checkbox"/>	 KFC.mSQNCst5Is	37.14 KB	image/png	Apr 11, 2020	
<input type="checkbox"/>	 Mcdonald's.0zJmEG4VJl	45.41 KB	image/png	Apr 11, 2020	
<input type="checkbox"/>	 Opticana.TgCuxvKupQ	55.49 KB	image/jpeg	Apr 11, 2020	
<input type="checkbox"/>	 Zara.ojLh3mhU	59.72 KB	image/png	Apr 11, 2020	

Zara.ojLh3mhU



Name  
[Zara.ojLh3mhU](#)

Size  
61,151 bytes

Type  
image/png

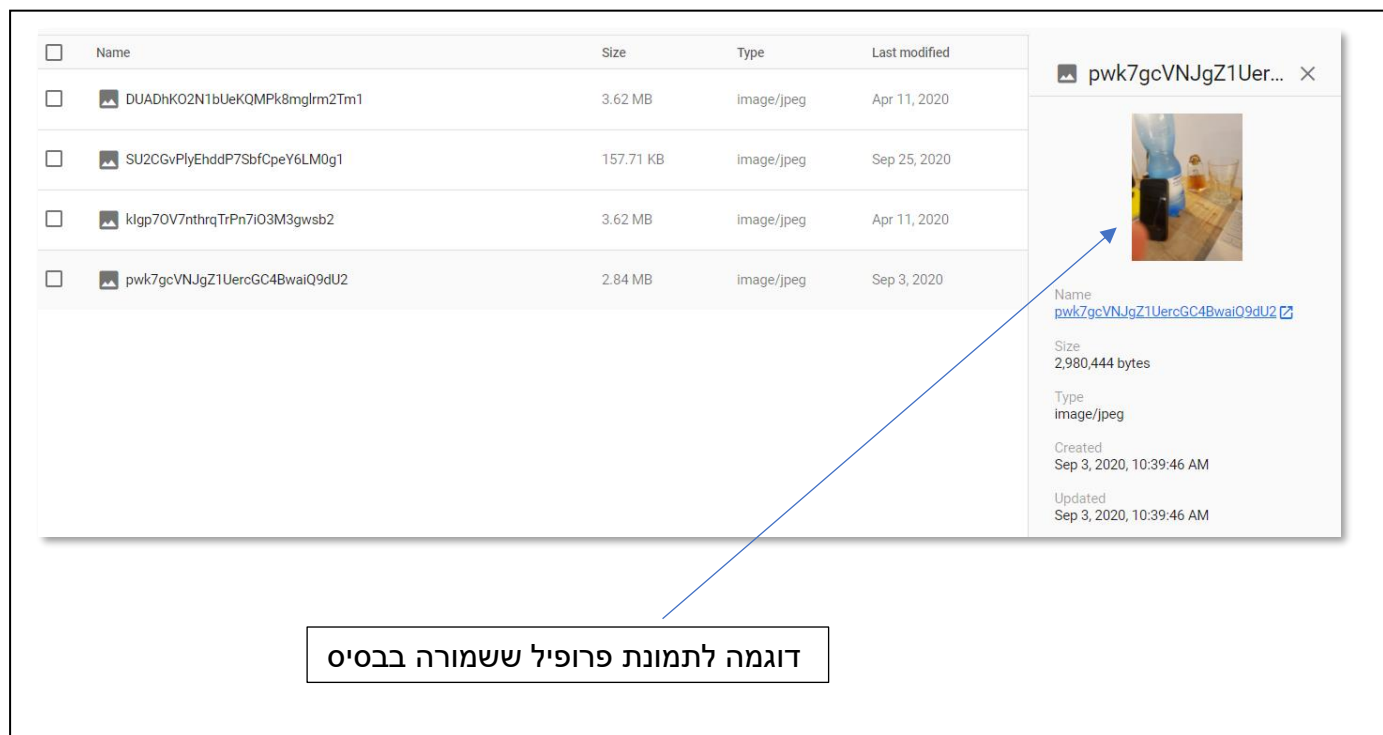
Created  
Apr 11, 2020, 8:51:49 PM

Updated  
Apr 11, 2020, 8:51:49 PM

File location


Other metadata

**בתיקייה של Profile Images** יש את תמונות הפרופיל של המשתמשים השונים אשר הזינו תמונת פרופיל, שם התמונה שמור תחת מספר הזהות הייחודי של המשתמש.



Name	Size	Type	Last modified
DUAdhK02N1bUeKQMPk8mglrm2Tm1	3.62 MB	image/jpeg	Apr 11, 2020
SU2CGvPlyEhddP7SbfCpeY6LM0g1	157.71 KB	image/jpeg	Sep 25, 2020
klgp7OV7nthrqTrPn7iO3M3gwsb2	3.62 MB	image/jpeg	Apr 11, 2020
pwk7gcVNJgZ1UercGC4BwaiO9dU2	2.84 MB	image/jpeg	Sep 3, 2020

pwk7gcVNJgZ1Uer... ×



Name  
[pwk7gcVNJgZ1UercGC4BwaiO9dU2](#)

Size  
2,980,444 bytes

Type  
image/jpeg

Created  
Sep 3, 2020, 10:39:46 AM

Updated  
Sep 3, 2020, 10:39:46 AM

דוגמה לתמונת פרופיל ששמורה בבסיס

כאשר רוצים לשלוח מה – Storage תמונה מגדירים שדה בתוך ה – Realtime Database כקישור לתמונה שב Storage ולכן אנו מבצעים את אותם פעולות כמו שמבצעים עבור שליפת נתונים אחרים מה – Realtime Database. כאשר רוצים לשמור תמונה ב – Storage. לדוגמה כאשר המשתמש מעדכן את תמונת הפרופיל שלו או אפילו מזין בפעם הראשונה תמונת פרופיל, יוצרים עצם מסוג StorageReference שמוגדר להיות הרפרנס של ה – Instance הספציפי של הסטוראג' באפליקציה וכאשר המשתמש בוחר תמונה מהגלריה שהיא תשמש כתמונת פרופיל, ניתן ביתר קלות לייצר Uri שבעזרתו שומרים תמונות ב – Storage ולאחר מכן מעלים את התמונה לשם, אם התמונה אכן נקלטה (יודעים זאת בעזרת מאזין) אז גם מעדכנים או מוסיפים את השדה של תמונת פרופיל בדאטאבייס עבור המשתמש הספציפי ששינה את תמונת הפרופיל שלו לקישור של התמונה ב – Storage.

## Authentication

חלק מבסיס הנתונים של הפיירבייס. מערכת אוטומטית, מהירה ובטוחה להרשמה לפרויקטים בעזרת דוא"ל וסיסמה. בנוסף, הם הוסיפו מערכת בטוחה ומהירה לשינוי סיסמה ודוא"ל ולכן אין צורך לשמור את הסיסמה ובכך האפליקציה הופכת לבטוחה הרבה יותר. כאשר המשתמש נרשם לאפליקציה, אנו יוצרים עצם מסוג FirebaseAuth ששומר Instance ספציפי של ה – Authentication באפליקציה. לעצם זה יש פעולות רבות שנרשמו על ידי מפתחי firebase ואחת מהן היא פעולה פשוטה למדי שבה פשוט מזינים את הדוא"ל והסיסמה אשר רוצים להירשם איתה ותשמר ב – Authentication. הדוא"ל צריך להיות תקין וגם הסיסמה צריכה להכיל יותר מ – 6 תווים לפי הדרישות המינימליות של firebase. אני מצאתי דרישות מינימליות אלה כמספיקות. לאחר שהמשתמש נרשם, נשמר בזיכרון של הטלפון ההרשמה שלו ולכן בפעמים הבאות כאשר יכנס לאפליקציה, כבר בפתיחת האפליקציה ניצור עצם מסוג FirebaseAuth, אם עצם זה ריק סימן שהמשתמש לא נרשם או לא התחבר לאפליקציה דרך הטלפון הנוכחי, ואם העצם לא ריק סימן שהמשתמש כן נרשם או התחבר דרך הטלפון הנוכחי ולכן יעבור ישר למסך הניווט ללא צורך להירשם מחדש שוב. כל המשתמשים חייבים להיות רשומים ב – Authentication והם יראו שם כרשימה כך:

Search by email address, phone number, or user UID					Add user		
Identifier	Providers	Created	Signed In	User UID			
rom123@gmail.com	✉	Sep 12, 2020	Nov 26, 2020	B9sp74wEYsaquwMxpLYARLzVCE...			
rom.granot@gmail.com	✉	Sep 25, 2020	Nov 26, 2020	SU2CGvPlyEhddP7SbfCpeY6LM0g1			
rom.granot123@gmail.com	✉	Mar 4, 2020	Mar 6, 2020	ao3PGXaZEAhgvNPh1kd3zDnaCtj2			
ofir@gmail.com	✉	Apr 9, 2020	Apr 15, 2020	klgp70V7nthrqTrPn7iO3M3gwsb2			
guycat@gmail.com	✉	Apr 15, 2020	Apr 16, 2020	pwk7gcVNJgZ1UercGC4BwaiQ9d...			
Rows per page: 50					1 – 5 of 5	<	>

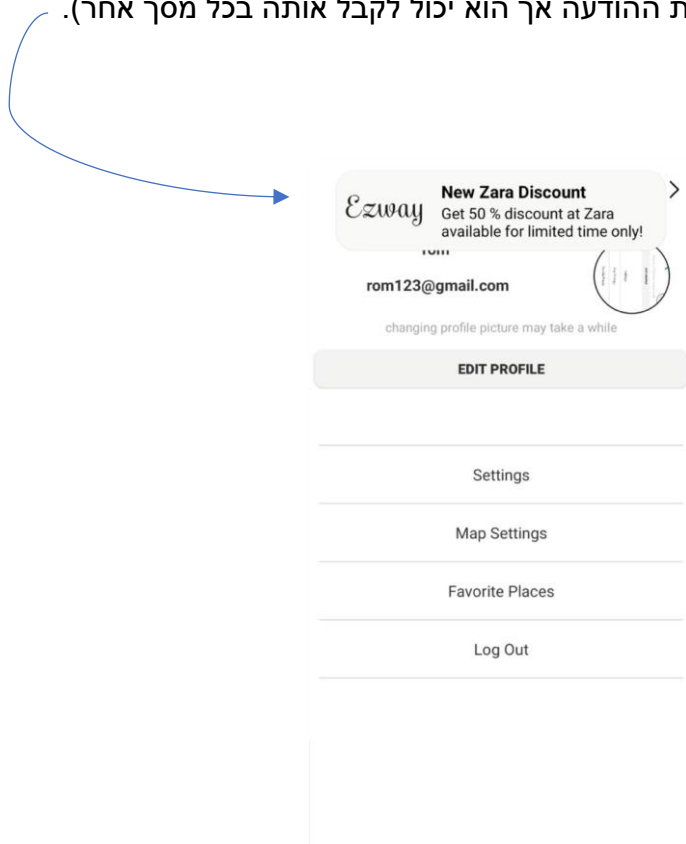
(כפי שניתן לראות אין שום איזכור לסיסמה או דרך לגשת אליה ולכן השימוש ב – Authentication בטוח מאוד בצד השרת וגם בצד לקוח).

כאשר רוצים לשנות פרטים של משתמש, ה – Authentication דורש שהסיסמה נשלחה למערכת במספר דקות בודדות אחרונות לפני שינוי הפרטים על מנת לאשר שהאדם שמשנה את הפרטים הוא אכן המשתמש, ובעזרת פעולות מובנות של פיירבייס, ניתן לשנות את הפרטים ב – Authentication.

## Cloud Messaging

חלק זה של הבסיס נתונים נועד לשליחת הודעות למשתמש על ידי צד השרת. כאן בצד שרת, ניתן ליצור הודעה שתוצג למשתמשים ספציפיים (על ידי שינוי בהגדרות ההודעה דרך האתר של פיירבייס) או לסך המשתמשים של האפליקציה. כאשר ההודעה נשלחת, אם האפליקציה כבויה או שהמשתמש מחוץ לה היא מוצגת כמו ההודעה רגילה, ואם המשתמש בתוך האפליקציה אז ההודעה מגיעה למחלקה MessageHandlerService שהרחבתי עליה למעלה.

ההודעה שתתקבל בזכות המחלקה MessageHandlerService – CustomToastHandler תראה כך (בדוגמה כאן ניתן לראות שהמשתמש היה במסך התפריט כאשר קיבל את ההודעה אך הוא יכול לקבל אותה בכל מסך אחר).





## סיכום אישי / רפלקציה

### איך הייתה עבורי העבודה על הפרויקט?

בשבילי, מהרגע שעליתי לכיתה יא והתחלנו ללמוד אנדרואיד סטודיו ופיתוח אפליקציות הייתי מאוד להוט להתחיל לעבוד, לפתח ולשבת כל היום ולרשום קוד ולעצב את האפליקציה שתהיה על הצד הטוב ביותר שלה ואכן במשך חודשים רבים, מבערך פברואר 2020 עד יולי 2020 עבדתי על האפליקציה שעות רבות כל יום ואף עקפתי את כל כיתתי בהרבה מבחינת החומר שהיה צריך להקיף ולסיים לאותה תקופה, כך שהגיע מצב שביוולי סיימתי באופן עקרוני כמעט הכול חוץ מהחלק של הניווט. כלומר, שליפת נתונים מהשרת, הצגת נתונים, סידור וארגון נתונים, עיצוב האפליקציה ומעבר בין מסכים וכו' היה גמור כבר בתחילת שנת הלימודים החדשה בספטמבר 2020. העבודה על שליפת הנתונים מהשרת הייתה בעיקר מייגעת אך בכל פעם שמשהו הצליח בסוף לעבוד הסיפוק גרם לרצון להמשיך עוד ועוד על מנת לשפר עוד יותר את האפליקציה. עיצוב האפליקציה גם הוא היה חלק יחסית גדול מזמני, לפחות בהתחלה ואת רוב ההשראה לקחתי מאפליקציות אחרות ומאינסטגרם וניסיתי לשלב גם את הסגנון עיצוב שלי על מנת שזה יהיה בכל זאת ייחודי ולא העתקה. ארגון הנתונים והצגתם גם היו קלים יחסית בזכות המנחה שלי שישב עם כל תלמיד והגיע איתו לצורה בה יהיה יעיל ביותר לסדר את הנתונים במבנה הנתונים שאנו משתמשים בו (firebase) ולכן לאחר שליפת הנתונים היה די קל לדאוג לעניינים אלה, ואז היה צריך להתחיל לעבוד על הניווט, בהתחלה התרגשתי נורא וחייתי זמן רב בשביל לעבוד עם החיישנים, ואכן, ניצלתי זמן רב כדי לראות איך החיישנים שהמורה הביא לי הגיבו ממרחקים קרובים יחסית לבינוניים (מטר עד 4 מטר) והתחלתי לשחק עם הקוד של החיישנים ולראות כיצד הם יגיבו וגם קראתי באינטרנט על הפעולות של החיישנים, לאחר זמן מה של עבודה נתקלתי בקשיים רבים בניווט ובגלל הייאוש שאין לי בדרך קלה להראות למורה שלי את הבעיות כי נסגרו בתי הספר והתחיל סגר חדש היה לי קשה להמשיך במשך מספר חודשים, אך לאחר שכן ניסיתי להמשיך כל פעם קצת הגעתי לפריצת דרך באמצעות שיחות עם המנחה לשיפור קוד ודרכים לפנות לבעיית הניווט ובכך הניווט עבד חלקית בפעם הראשונה ולאחר שינויים רבים וניסויים הגעתי לניווט שאני מרוצה ממנו וגם סיימתי קטעי קוד נוספים שהייתי צריך לסיים על מנת לגמור את הפרויקט ולאחר מכן הייתה הקלה גדולה. העבודה על הפרויקט הייתה לא קלה ויש הרבה מה ללמוד ממנה.

### מה קיבלתי מהעבודה?

מהעבודה הזאת קיבלתי דברים רבים, כמו תכנון זמן נכון יותר, תכנון קוד ומחלקות בתכנות מונחה עצמים טוב יותר שהשתפר עם הזמן, מפני שלאחר חודשים כשחזרתי לקוד שרשמתי בעבר שמתי לב לפרטים רבים שיכולתי לרשום בצורה מסודרת יותר ויותר יעילה וקריאה ולכן העבודה עזרה להבין באמת על לשים תשומת

לב בדברים אלה. דבר נוסף שקיבלתי מהעבודה הוא פרספקטיבה על זמן ועל פרויקטים. עד כה עסקתי בדברים ופרויקטים בתכנות באופן עצמאי שנמשכו ימים אחדים או שפרשתי מהם לאחר ימים אחדים, כאן הפרויקט לקח קצת יותר משנה והשתמש במשאבים רבים וזמן רב מימים רבים, ולכן מעבודה זו אני לוקח את ההבנה שפרויקט גדול שכזה צריך לתכנן היטב, שאי אפשר לפרוש מפרויקט שכזה שיש לי בו מחויבות לימודית באמצע ולהבין שהוא לוקח זמן ולחלק אותו לתת – בעיות כפי שהמנחה שלי עודד תמיד לעשות.

## אילו כלים אני לוקח איתי להמשך?

כלים שאני לוקח להמשך איתי הם ניסיון בעבודה עם API שכתב גורם חיצוני וקריאת מסמכים על מנת להבין את הקוד ואת הרעיון מאחוריו, למדתי לחפש באינטרנט פתרונות בצורה יעילה יותר ולהבין איזה חלקים כדאי לקרוא בעיוניות ואיזה חלקים לא, וגם איזה חלקים בקוד שאני לוקח ממנו עזרה אני צריך ואיזה הוא קוד סרק או קוד שלא תואם לקוד שלי לכן אצטרך לשנות אותו או לא להשתמש בו כלל. בנוסף, בזכות כך שהיו ימים שהייתי צריך לשבת ולחשוב על דרכים מתמטיות לפתור בעיה וגם לחשוב על דרכים לרשום בקוד את הדרכים המתמטיות האלה למדתי לייעל את עצמי ואת דרך החשיבה המתמטית שלי.

## מה הם הקשיים והאתגרים שעמדתי בפניהם?

קשיים ואתגרים שעמדתי בפניהם היו בהתחלה של הפרויקט מבחינת עיצוב האפליקציה מפני שאני נורא אוהב אפליקציות שנראות יפה לעין ולפי דעתי אפליקציה שנראית טוב נותנת חווית משתמש ומושכת משתמשים רבים ורציתי לנסות להביע זאת בפרויקט שלי אך אני אדם לא יצירתי במיוחד מבחינת עיצוב ולכן היה קשה לי לשלב עיצוב באפליקציה, וגם כאשר לקחתי אלמנטים שונים מאפליקציות אחרות היה לי קושי לשלב זאת בצורה שאני דמיינתי בראש. לאחר שעברתי את שלב העיצוב ועברתי לשלב התכנות והקוד היה לי לא מעט קשיים בשליפת והזנת נתונים לשרת, בעיקר בכל מה שמדובר בשינוי פרטים והעלאת תמונה לשרת וכו' מפני שדברים אלו נעשו לבד והרבה לפני שחבריי לכיתה עשו אותם ולכן הייתי צריך להתמודד עם בעיות אלה לבד בזמן שהמורה עזר לתלמידים פחות מתקדמים. לאחר שסיימתי זאת, וכשהגעתי לחלק של הניווט באפליקציה, החלק העיקרי והחשוב ביותר מפני שכל האפליקציה שלי מבוססת על כך נתקלתי בקשיים רבים מפני שהחיישנים שהמורה סיפק לי לא היו מדויקים במיוחד ותחילה הרעיון שלי למיקום המשתמש היה באמצעות חיתוך מעגלים ואכן לאחר קוד רב שרובו היה מתמטיקה בה אני יצרתי מחלקת משוואות מעגל ועשיתי שימוש בחשבון מטריצות כדי למצוא בקלות את חיתוך המעגלים בעזרת המשוואות, הגעתי למבוי סתום מפני שהחיישנים הראו מרחקים כך שלעולם לא היה חיתוך בין 3 מעגלים בנקודה אחת ולכן לא היה אפשרות למקם בקוד. לכן, אני והמנחה שלי חשבנו יחד על הרעיון הנוכחי, שהוא שהמשתמש ינוע בקווים ישרים לאורך הקווים של החיישנים ובכך להשתמש ביחסי

מרחקים הקצרים ביותר, אך, גם לאחר שעברנו לרעיון זה היה קושי רב מפני שהיחסים בין החיישנים היו לא מדויקים והמיקום של המשתמש היה לא טוב אך גם לא היה אפשרות להחליף את החיישנים מפני שבית הספר היה סגור והתחיל סגר לכן נאלצתי להשתמש במה שיש ולהגיע לתוצאות הטובות ביותר.

## מה המסקנות שלי מהעבודה?

המסקנות שלי מהעבודה על הפרויקט הם שהוא היה קשה בהרבה ממה שחשבתי ולכן צריך לפעמים לא להקל ראש על פרויקטים שאתה לוקח ולהבין שהם דורשים זמן, חשיבה ותכנון. בנוסף, העבודה פתחה בפניי עולם חדש מבחינת הבנת קוד לעומק והסתכלות על קוד מלפני זמן רב שכתבת אותו וכעת אני מבין יותר טוב שלרשום דוקומנטציה לקוד ולרשום אותו יעיל קריא ומסודר זה דבר חשוב מאוד, אולי הכי חשוב אפילו יותר מזה שהקוד יעבוד במאה האחוזים.

## מה הייתי עושה אחרת לו הייתי מתחיל היום?

לו הייתי מתחיל היום הייתי משקיע פחות זמן בדברים כמו עיצוב, או משתמש בכלים לעצב שמצאתי עם הזמן ביותר יעילות וקלות כעת שאני שולט בהם. בנוסף הייתי עושה לעצמי תכנון של כל המחלקות ולהשתדל לעשות זאת כמה שיותר מפורט ולפרק בעיה לכמה שיותר שכבות כך שאופן הטיפול בה יהיה ממוקד. כאשר הייתה לי בעיה גדולה אז לא ידעתי מאיפה להתחיל ולכן לרוב נתקעתי ולא עשיתי כלום ולכן זה דבר שהייתי עושה אחרת. בנוסף, פעמים רבות התעצלתי לקרוא דוקומנטציה של קוד שאחרים רשמו או שאנדרואיד רשמו בעצמם, העדפתי לראות סרטונים שלא מנמקים באופן מלא מה כל דבר בקוד עושה ולמה הוא עושה אותנו ולכן התעכבתי רבות וזה דבר שהייתי גם כן עושה אחרת.

## מה אם היה קורה אחרת, העבודה הייתה יותר יעילה עבורי?

אני מניח שאם לא היה קורונה העבודה מול החיישנים הייתה הרבה יותר יעילה ומהירה אך מצד שני אם לא היה קורונה לא הייתי מסיים את רוב הפרויקט חוץ מהניווט מוקדם כפי שסיימתי. בנוסף, היה לי קשיים אישיים ובחירות בחיים שעשיתי שהקשו עלי בעת העבודה על הפרויקט, לדוגמה: למדתי באוניברסיטה קורס במתמטיקה שלקח את רוב זמני על מנת ללמוד את החומר של הקורס והיה לי דד – ליין של מטלות להגיש ולכן לא היה לי זמן לעבוד על האפליקציה כמו לפני כן, דבר שהאט את קצב ההתקדמות, בעיקר בחלק המשמעותי – הניווט.

## מדריך למפתח

נציג כל אחד מקבצי הפרויקט:

### AndroidManifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.ezway">

    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.MEDIA_CONTENT_CONTROL"
        tools:ignore="ProtectedPermissions" />
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
        <activity
            android:name=".Activities.CriticsActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.DiscountActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.ShopActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.FavoritePlacesActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.MapActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.SignUpActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.SettingsActivity"
            android:screenOrientation="portrait" />
        <activity
            android:name=".Activities.ProfileActivity"
            android:screenOrientation="portrait" />
        <activity
```

```

        android:name=".Activities.MenuActivity"
        android:screenOrientation="portrait" />
    <activity
        android:name=".Activities.MapSettingsActivity"
        android:screenOrientation="portrait" />
    <activity
        android:name=".Activities.LoginActivity"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <service
        android:name="org.altbeacon.beacon.service.BeaconService"
        tools:node="replace">
        <meta-data
            android:name="longScanForcingEnabled"
            android:value="true" />
    </service>
    <service android:name=".Handlers.MessageHandlerService"
        android:exported="false">
        <intent-filter>
            <action android:name="com.google.firebase.MESSAGING_EVENT"
        />
        </intent-filter>
    </service>
</application>
</manifest>

```

עבור קבצי הקוד ב – Java חילקתי את המחלקות ל – 4 תיקיות לשם נוחות וחלוקה לתפקידים כפי שפירטתי קודם לכן, אציג זאת שוב בטבלה:

Handlers	Dialogs	BaseClasses	Adapters	Activities
BottomSheetHandler	ConfirmPasswordDialog	BeaconInfo	DiscountRecyclerViewAdapter	CriticsActivity
BroadcastReceiver	DiscountDialog	Discount	FavShopsAdapter	DiscountActivity
CustomToastHandler	MyDatePickerDialog	MyDate	MyAdapter	FavoritePlacesActivity
MessageHandlerService	ReviewDialog	Navigation	RecyclerViewAdapter	LoginActivity
		Position	ReviewAdapter	MapActivity
		Review		MapSettingsActivity

		Store		MenuActivity
		User		ProfileActivity
				SettingsActivity
				ShopActivity
				SignUpActivity

נעבור תחילה על כל תיקיה בסדר א"ב באנגלית:

## Activites

### Class CriticsActivity

- `Java.lang.Object`
  - `android.content.Context`
    - `android.content.ContextWrapper`
      - `android.view.ContextThemeWrapper`
        - `android.app.Activity`
          - `com.example.ezway.Activities.CriticsActivity`

```
public class CriticsActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

#### • Field Summary

Fields	
Modifier and Type	Field and Description
<code>private com.google.firebase.database.DatabaseReference</code>	<code>databaseRef</code>
<code>private com.google.firebase.auth.FirebaseAuth</code>	<code>mAuth</code>
<code>private android.content.SharedPreferences</code>	<code>preferences</code>
<code>static java.lang.String</code>	<code>PREFS_NAME</code>
<code>private ReviewAdapter</code>	<code>recyclerAdapter</code>
<code>private java.util.List&lt;Review&gt;</code>	<code>reviews</code>

```
private Store store
static java.lang.String THEME_PREF
```

- **Constructor Summary**

### Constructors

#### Constructor and Description

**CriticsActivity** ()

- **Method Summary**

### All Methods Instance Methods Concrete Methods

#### Modifier and Type

#### Method and Description

private void	<b>changeBackground</b> (int color) Change the background color of views in layout according to the color passed in.
java.util.List<Review>	<b>getReviews</b> ()
private void	<b>loadLocale</b> () Loads the current language last that is saved on phone.
void	<b>onClick</b> (android.view.View v)
protected void	<b>onCreate</b> (android.os.Bundle savedInstanceState)
private void	<b>setLocale</b> (java.lang.String language) Sets the current language of the app to the language passed in.
private void	<b>setUp</b> () Gets all reviews of the shop the user is navigating to.

- **Field Detail**

- **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

- **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

- **preferences**

```
private android.content.SharedPreferences preferences
```

- **store**

```
private Store store
```

- **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

- **databaseRef**

```
private com.google.firebase.database.DatabaseReference
databaseRef
```

- **reviews**

```
private java.util.List<Review> reviews
```

- **recyclerAdapter**

```
private ReviewAdapter recyclerAdapter
```

- **Constructor Detail**

- **CriticsActivity**

```
public CriticsActivity()
```

- **Method Detail**

- **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

- **setUp**

```
private void setUp()
```

Gets all reviews of the shop the user is navigating to.

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **onClick**

```
public void onClick(android.view.View v)
```



## רום גרנות – אפליקציית EZWAY

Specified by:

onClick in interface android.view.View.OnClickListener

- **getReviews**

```
public java.util.List<Review> getReviews()
```

## Class DiscountActivity

- **Java.lang.Object**
  - android.content.Context**
    - android.content.ContextWrapper**
      - android.view.ContextThemeWrapper**
        - android.app.Activity**
          - com.example.ezway.Activities.DiscountActivity**

```
public class DiscountActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • Field Summary

#### Fields

Modifier and Type	Field and Description
private com.google.firebase.database.DatabaseReference	<b>databaseRef</b>
private java.util.List<Discount>	<b>discounts</b>
private com.google.firebase.auth.FirebaseAuth	<b>mAuth</b>
private android.content.SharedPreferences	<b>preferences</b>
static java.lang.String	<b>PREFS_NAME</b>
private Store	<b>store</b>
static java.lang.String	<b>THEME_PREF</b>

### • Constructor Summary

#### Constructors

Constructor and Description
<b>DiscountActivity()</b>

### • Method Summary

#### All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
private void	<b>changeBackground</b> (int color) Change the background color of views in layout according to the color passed in.
private void	<b>loadLocale</b> () Loads the current language last that is saved on phone.
void	<b>onClick</b> (android.view.View v)

protected void	<b>onCreate</b> (android.os.Bundle savedInstanceState)
private void	<b>setLocale</b> (java.lang.String language) Sets the current language of the app to the language passed in.
private void	<b>setUp</b> () Gets all discount of the shop the user is navigating to.

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **preferences**

```
private android.content.SharedPreferences preferences
```

### • **store**

```
private Store store
```

### • **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

### • **databaseRef**

```
private com.google.firebase.database.DatabaseReference  
databaseRef
```

### • **discounts**

```
private java.util.List<Discount> discounts
```

## • **Constructor Detail**

### • **DiscountActivity**

```
public DiscountActivity()
```

## • **Method Detail**

### • **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

### • **setUp**

```
private void setUp()
```

Gets all discount of the shop the user is navigating to.

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

## Class FavoritePlacesActivity

- `Java.lang.Object`
  - `android.content.Context`
    - `android.content.ContextWrapper`
      - `android.view.ContextThemeWrapper`
        - `android.app.Activity`
          - `com.example.ezway.Activities.FavoritePlacesActivity`

```
public class FavoritePlacesActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • Field Summary

#### Fields

Modifier and Type	Field and Description
private <code>FavShopsAdapter</code>	<code>adapter</code>
private <code>com.google.firebase.database.DatabaseReference</code>	<code>databaseRef</code>
private <code>java.util.List&lt;java.lang.String&gt;</code>	<code>favStores</code>
private <code>BottomSheetHandler</code>	<code>handler</code>
private <code>androidx.recyclerview.widget.ItemTouchHelper.SimpleCallback</code>	<code>itemTouchHelperCallBack</code> An instance of a call back in order to listen to swipe.
private <code>com.google.firebase.auth.FirebaseAuth</code>	<code>mAuth</code>
private <code>Position</code>	<code>pos</code>
private <code>android.content.SharedPreferences</code>	<code>preferences</code>
static <code>java.lang.String</code>	<code>PREFS_NAME</code>
private <code>java.lang.String[]</code>	<code>shops</code>
private <code>Store</code>	<code>store</code>
private <code>java.lang.String</code>	<code>storeId</code>
static <code>java.lang.String</code>	<code>THEME_PREF</code>

### • Constructor Summary

#### Constructors

##### Constructor and Description

`FavoritePlacesActivity()`

### • Method Summary

**All Methods** Instance Methods Concrete Methods

Modifier and Type	Method and Description
void	<b>addFavorite</b> (Store store) Adds the store to the user's favorite places list.
private void	<b>changeBackground</b> (int color) Change the background color of views in layout according to the color passed in.
private void	<b>loadLocale</b> () Loads the current language last that is saved on phone.
void	<b>onClick</b> (android.view.View v)
protected void	<b>onCreate</b> (android.os.Bundle savedInstanceState)
private void	<b>removeFavorite</b> (androidx.recyclerview.widget.RecyclerView.ViewHolder viewHolder) Removes the store from the user's favorite places.
private void	<b>setLocale</b> (java.lang.String language) Sets the current language of the app to the language passed in.
private void	<b>setUp</b> () Gets all the favorite places of the user.

• **Field Detail**

• **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

• **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

• **preferences**

```
private android.content.SharedPreferences preferences
```

• **handler**

```
private BottomSheetHandler handler
```

• **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

• **databaseRef**

```
private com.google.firebase.database.DatabaseReference  
databaseRef
```

• **adapter**

```
private FavShopsAdapter adapter
```

• **favStores**

```
private java.util.List<java.lang.String> favStores
```

- **store**

```
private Store store
```

- **pos**

```
private Position pos
```

- **shops**

```
private java.lang.String[] shops
```

- **storeId**

```
private java.lang.String storeId
```

- **itemTouchHelperCallBack**

```
private androidx.recyclerview.widget.ItemTouchHelper.SimpleCallback itemTouchHelperCallBack
```

An instance of a call back in order to listen to swipe.

- **Constructor Detail**

- **FavoritePlacesActivity**

```
public FavoritePlacesActivity()
```

- **Method Detail**

- **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **setUp**

```
private void setUp()
```

Gets all the favorite places of the user.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

- **addFavorite**

```
public void addFavorite(Store store)
```

Adds the store to the user's favorite places list.

**Parameters:**  
store

- **removeFavorite**

```
private void removeFavorite(androidx.recyclerview.widget  
.RecyclerView.ViewHolder viewHolder)
```

Removes the store from the user's favorite places.

**Parameters:**  
viewHolder



## Class LoginActivity

- **Java.lang.Object**
  - android.content.Context**
    - android.content.ContextWrapper**
      - android.view.ContextThemeWrapper**
        - android.app.Activity**
          - com.example.ezway.Activities.LoginActivity**

```
public class LoginActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • **Field Summary**

#### Fields

Modifier and Type	Field and Description
private android.widget.EditText	<b>etEmail</b>
private android.widget.EditText	<b>etPassword</b>
private com.google.firebase.auth.FirebaseAuth	<b>mAuth</b>
static java.lang.String	<b>PREFS_NAME</b>
static java.lang.String	<b>THEME_PREF</b>

### • **Constructor Summary**

#### Constructors

##### Constructor and Description

**LoginActivity()**

### • **Method Summary**

#### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
private void	<b>changeBackground</b> (int color) Change the background color of views in layout according to the color passed in.
private void	<b>loadLocale</b> () Loads the current language last that is saved on phone.
void	<b>onClick</b> (android.view.View v)
protected void	<b>onCreate</b> (android.os.Bundle savedInstanceState)
void	<b>onStart</b> ()

private void	<b>setLocale</b> (java.lang.String language) Sets the current language of the app to the language passed in.
private void	<b>SignIn</b> (java.lang.String email, java.lang.String password) Verifies the email and password and signs in.

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **etEmail**

```
private android.widget.EditText etEmail
```

### • **etPassword**

```
private android.widget.EditText etPassword
```

### • **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

## • **Constructor Detail**

### • **LoginActivity**

```
public LoginActivity()
```

## • **Method Detail**

### • **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

### • **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

### • **SignIn**

```
private void SignIn(java.lang.String email, java.lang.String password)
```

Verifies the email and password and signs in.

**Parameters:**

email – Email passed in by the user.

password – Password passed in by the user.

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**

color

- **onStart**

```
public void onStart()
```

**Overrides:**

onStart in class android.app.Activity

## Class MapActivity

- **Java.lang.Object**

**android.content.Context**

**android.content.ContextWrapper**

**android.view.ContextThemeWrapper**

**android.app.Activity**

**com.example.ezway.Activities.MapActivity**

```
public class MapActivity
extends android.app.Activity
implements org.altbeacon.beacon.BeaconConsumer,
android.view.View.OnClickListener
```

- **Field Summary**

### Fields

Modifier and Type	Field and Description
private org.altbeacon.beacon.BeaconManager	<b>beaconManager</b>
private android.graphics.Bitmap	<b>bitmap</b>
private android.content.BroadcastReceiver	<b>br</b>
private static int	<b>DOWN</b>
<b>BottomSheetHandler</b>	<b>handler</b>
private boolean	<b>isEnglish</b>
static java.lang.String	<b>LANG_PREF</b>
private static int	<b>LEFT</b>
private android.widget.ImageView	<b>mapImageView</b>
private <b>Navigation</b>	<b>navigation</b>
private static int	<b>PERMISSION_REQUEST_FINE_LOCATION</b>
private <b>Position</b>	<b>pos</b>
private android.content.SharedPreferences	<b>preferences</b>
static java.lang.String	<b>PREFS_NAME</b>
private static int	<b>RIGHT</b>
private java.lang.String[]	<b>shops</b>
private int	<b>showedHeight</b>
private int	<b>showedWidth</b>
private <b>Store</b>	<b>store</b>
private static java.lang.String	<b>TAG</b>
static java.lang.String	<b>THEME_PREF</b>

private static int

UP

- Constructor Summary**

### Constructors

#### Constructor and Description

**MapActivity()**

- Method Summary**

### All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
private void	<b>broadcastReceiverSetup()</b> Initializes the broadcast receiver and its settings.
private void	<b>changeBackground(int color)</b> Change the background color of views in layout according to the color passed in.
private void	<b>changeLanguage(boolean langPref)</b> Changes the language of a certain view.
private void	<b>checkForFineLocationGrant()</b> Checks if the user has granted permission to bluetooth connection to the app.
private void	<b>displayFineLocationMessage()</b> Displays an alert dialog when the user does not allow bluetooth connection to the app.
private int	<b>getDirectionToShop(Position user, Position shop)</b> Calculates which direction the user needs to go.
private int	<b>getDistanceFromShop()</b> Calculates the distance of the user from the shop.
<b>Position</b>	<b>getPos()</b>
<b>Store</b>	<b>getStore()</b>
private void	<b>handleBeaconsInput(java.util.Collection&lt;org.altbeacon.beacon.Beacon&gt; beacons)</b> Updates the last call of each detected beacon.
private void	<b>loadLocale()</b> Loads the current language last saved on phone.
void	<b>navigate(Store store)</b>
void	<b>onBeaconServiceConnect()</b> Called when the beacon service is running and ready to be used through BeaconManager.
void	<b>onClick(android.view.View view)</b>
protected void	<b>onCreate(android.os.Bundle savedInstanceState)</b>

void	<b>onRequestPermissionsResult</b> (int requestCode, java.lang.String[] permissions, int[] grantResults) checks of the permission of the app for bluetooth is valid.
protected void	<b>onStop</b> ()
private void	<b>setBeaconSettings</b> () initializes the beacon manager and its settings.
private void	<b>setLocale</b> (java.lang.String language) sets the current language to the language passed in.
void	<b>setStore</b> (Store store)
private void	<b>showAnimation</b> (int direction) Shows the direction the user needs to go towards.
void	<b>unbindManager</b> ()
void	<b>updateMap</b> () Updates the map according to the position of the user and the shop he's navigating to.

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **LANG\_PREF**

```
public static final java.lang.String LANG_PREF
```

### • **TAG**

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

### • **PERMISSION\_REQUEST\_FINE\_LOCATION**

```
private static final int  
PERMISSION_REQUEST_FINE_LOCATION
```

### • **preferences**

```
private android.content.SharedPreferences preferences
```

### • **handler**

```
public BottomSheetHandler handler
```

### • **beaconManager**

```
private org.altbeacon.beacon.BeaconManager beaconManager
```

### • **bitmap**

```
private android.graphics.Bitmap bitmap
```

### • **mapImageView**

```
private android.widget.ImageView mapImageView
```

- shops

```
private java.lang.String[] shops
```

- pos

```
private Position pos
```

- store

```
private Store store
```

- navigation

```
private Navigation navigation
```

- isEnglish

```
private boolean isEnglish
```

- br

```
private android.content.BroadcastReceiver br
```

- UP

```
private static final int UP
```

- RIGHT

```
private static final int RIGHT
```

- DOWN

```
private static final int DOWN
```

- LEFT

```
private static final int LEFT
```

- showedWidth

```
private int showedWidth
```

- showedHeight

```
private int showedHeight
```

- Constructor Detail**

- MapActivity

```
public MapActivity()
```

- Method Detail**

- onCreate

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

Overrides:

onCreate in class android.app.Activity

- **broadcastReceiverSetup**

```
private void broadcastReceiverSetup()
```

Initializes the broadcast receiver and its settings.

- **checkForFineLocationGrant**

```
private void checkForFineLocationGrant()
```

Checks if the user has granted permission to bluetooth connection to the app.

- **onRequestPermissionsResult**

```
public void onRequestPermissionsResult(int requestCode,
    java.lang.String[] permissions, int[] grantResults)
```

checks of the permission of the app for bluetooth is valid.

**Overrides:**

onRequestPermissionsResult in class android.app.Activity

**Parameters:**

requestCode

permissions

grantResults

- **displayFineLocationMessage**

```
private void displayFineLocationMessage()
```

Displays an alert dialog when the user doesn't allow bluetooth connection to the app.

- **setBeaconSettings**

```
private void setBeaconSettings()
```

initializes the beacon manager and its settings.

- **onBeaconServiceConnect**

```
public void onBeaconServiceConnect()
```

Called when the beacon service is running and ready to be used through BeaconManager.

**Specified by:**

onBeaconServiceConnect in interface org.altbeacon.  
.beacon.BeaconConsumer

- **handleBeaconsInput**

```
private void handleBeaconsInput(java.util.Collection<org
    .altbeacon.beacon.Beacon> beacons)
```

Updates the last call of each detected beacon.

**Parameters:**

beacons - A collection of the beacons whose signals are detected.



- **showAnimation**

```
private void showAnimation(int direction)
```

Shows the direction the user needs to go towards.

**Parameters:**

direction - An int value representing the direction.

- **getDirectionToShop**

```
private int getDirectionToShop(Position user,
    Position shop)
```

Calculates which direction the user needs to go.

**Parameters:**

user - User's position

shop - Shop's position

**Returns:**

An int value representing the direction.

- **getDistanceFromShop**

```
private int getDistanceFromShop()
```

Calculates the distance of the user from the shop.

**Returns:**

An int value representing distance.

- **updateMap**

```
public void updateMap()
```

Updates the map according to the position of the user and the shop he's navigating to.

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

sets the current language to the language passed in.

- **changeLanguage**

```
private void changeLanguage(boolean langPref)
```

Changes the language of a certain view.

**Parameters:**

langPref - Preferred language of the user.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **onClick**

```
public void onClick(android.view.View view)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

- **unbindManager**

```
public void unbindManager()
```

- **getStore**

```
public Store getStore()
```

- **setStore**

```
public void setStore(Store store)
```

- **getPos**

```
public Position getPos()
```

- **navigate**

```
public void navigate(Store store)
```

- **onStop**

```
protected void onStop()
```

**Overrides:**  
onStop in class android.app.Activity

## Class MapSettingsActivity

- **Java.lang.Object**
  - android.content.Context**
    - android.content.ContextWrapper**
      - android.view.ContextThemeWrapper**
        - android.app.Activity**
          - com.example.ezway.Activities.MapSettingsActivity**

```
public class MapSettingsActivity
extends android.app.Activity
implements android.view.View.OnClickListener,
android.widget.CompoundButton.OnCheckedChangeListener
```

### • Field Summary

#### Fields

Modifier and Type	Field and Description
static java.lang.String	<b>DIS_PREF</b>
private <b>Position</b>	<b>pos</b>
private android.content.SharedPreferences	<b>preferences</b>
static java.lang.String	<b>PREFS_NAME</b>
private <b>Store</b>	<b>store</b>
static java.lang.String	<b>THEME_PREF</b>

### • Constructor Summary

#### Constructors

Constructor and Description
<b>MapSettingsActivity</b> ()

### • Method Summary

#### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
private void	<b>changeBackground</b> (int color) Change the background color of views in layout according to the color passed in.
private void	<b>loadLocale</b> () Loads the current language last that is saved on phone.
void	<b>onCheckedChanged</b> (android.widget.CompoundButton compoundButton, boolean b)

void	<b>onClick</b> (android.view.View v)
protected void	<b>onCreate</b> (android.os.Bundle savedInstanceState)
private void	<b>setLocale</b> (java.lang.String language) Sets the current language of the app to the language passed in.

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **DIS\_PREF**

```
public static final java.lang.String DIS_PREF
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **preferences**

```
private android.content.SharedPreferences preferences
```

### • **store**

```
private Store store
```

### • **pos**

```
private Position pos
```

## • **Constructor Detail**

### • **MapSettingsActivity**

```
public MapSettingsActivity()
```

## • **Method Detail**

### • **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

### • **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

### • **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

- **onCheckedChanged**

- ```
public void onCheckedChanged(android.widget.CompoundButt  
on compoundButton, boolean b)
```

**Specified by:**  
onCheckedChanged in interface android.widget  
.CompoundButton.OnCheckedChangeListener

## Class MenuActivity

- `Java.lang.Object`
  - `android.content.Context`
    - `android.content.ContextWrapper`
      - `android.view.ContextThemeWrapper`
        - `android.app.Activity`
          - `com.example.ezway.Activities.MenuActivity`

```
public class MenuActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • Field Summary

#### Fields

| Modifier and Type                                      | Field and Description    |
|--------------------------------------------------------|--------------------------|
| private com.google.firebase.database.DatabaseReference | <code>databaseRef</code> |
| private com.google.firebase.auth.FirebaseAuth          | <code>mAuth</code>       |
| private <code>Position</code>                          | <code>pos</code>         |
| static java.lang.String                                | <code>PREFS_NAME</code>  |
| private <code>Store</code>                             | <code>store</code>       |
| static java.lang.String                                | <code>THEME_PREF</code>  |

### • Constructor Summary

#### Constructors

##### Constructor and Description

`MenuActivity()`

### • Method Summary

#### All Methods Instance Methods Concrete Methods

| Modifier and Type | Method and Description                                                                                                       |
|-------------------|------------------------------------------------------------------------------------------------------------------------------|
| private void      | <code>changeBackground(int color)</code><br>Change the background color of views in layout according to the color passed in. |
| private void      | <code>loadLocale()</code><br>Loads the current language last that is saved on phone.                                         |
| void              | <code>onClick(android.view.View v)</code>                                                                                    |
| protected void    | <code>onCreate(android.os.Bundle savedInstanceState)</code>                                                                  |

```
private void
```

**setLocale** (java.lang.String language)  
Sets the current language of the app to the language passed in.

```
private void
```

**setUp** ()  
Get the user information from the database.

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

### • **databaseRef**

```
private com.google.firebase.database.DatabaseReference  
databaseRef
```

### • **store**

```
private Store store
```

### • **pos**

```
private Position pos
```

## • **Constructor Detail**

### • **MenuActivity**

```
public MenuActivity()
```

## • **Method Detail**

### • **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

### • **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

### • **setLocale**

```
private void setLocale (java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**

color

- **setUp**

```
private void setUp()
```

Get the user information from the database.

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**

onClick in interface android.view.View.OnClickListener



## Class ProfileActivity

- **Java.lang.Object**
  - android.content.Context**
    - android.content.ContextWrapper**
      - android.view.ContextThemeWrapper**
        - android.app.Activity**
          - com.example.ezway.Activities.ProfileActivity**

```
public class ProfileActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • **Field Summary**

#### **Fields**

| Modifier and Type                                      | Field and Description   |
|--------------------------------------------------------|-------------------------|
| private java.lang.String                               | <b>currentEmail</b>     |
| private java.lang.String                               | <b>currentUsername</b>  |
| private com.google.firebase.database.DatabaseReference | <b>databaseRef</b>      |
| private android.widget.EditText                        | <b>etEmail</b>          |
| private android.widget.EditText                        | <b>etNewPassword</b>    |
| private android.widget.EditText                        | <b>etNewPasswordCon</b> |
| private android.widget.EditText                        | <b>etUsername</b>       |
| private com.google.firebase.auth.FirebaseAuth          | <b>mAuth</b>            |
| static int                                             | <b>PICK_IMAGE</b>       |
| private <b>Position</b>                                | <b>pos</b>              |
| private android.content.SharedPreferences              | <b>preferences</b>      |
| static java.lang.String                                | <b>PREFS_NAME</b>       |
| private com.google.firebase.storage.StorageReference   | <b>storageReference</b> |
| private <b>Store</b>                                   | <b>store</b>            |
| static java.lang.String                                | <b>THEME_PREF</b>       |

### • **Constructor Summary**

#### **Constructors**

##### Constructor and Description

**ProfileActivity** ()

### • **Method Summary**

**All Methods** Instance Methods Concrete Methods

| Modifier and Type | Method and Description                                                                                                            |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| private void      | <b>changeBackground</b> (int color)<br>Change the background color of views in layout according to the color passed in.           |
| private void      | <b>loadLocale</b> ()<br>Loads the current language last that is saved on phone.                                                   |
| void              | <b>onActivityResult</b> (int requestCode, int resultCode, android.content.Intent data)<br>Gets called after a picture was chosen. |
| void              | <b>onClick</b> (android.view.View v)                                                                                              |
| protected void    | <b>onCreate</b> (android.os.Bundle savedInstanceState)                                                                            |
| private void      | <b>setLocale</b> (java.lang.String language)<br>Sets the current language of the app to the language passed in.                   |
| private void      | <b>setUp</b> ()<br>Get the user information from the database.                                                                    |
| private void      | <b>uploadPhoto</b> (android.net.Uri uri)<br>Uploads profile picture of the user to the database.                                  |
| private void      | <b>validate</b> ()<br>Validates the input of email and password of the user and changes done by the user.                         |

• **Field Detail**

• **PICK\_IMAGE**

```
public static final int PICK_IMAGE
```

• **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

• **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

• **preferences**

```
private android.content.SharedPreferences preferences
```

• **etUsername**

```
private android.widget.EditText etUsername
```

• **etEmail**

```
private android.widget.EditText etEmail
```

• **etNewPassword**

```
private android.widget.EditText etNewPassword
```

- **etNewPasswordCon**

```
private android.widget.EditText etNewPasswordCon
```

- **currentEmail**

```
private java.lang.String currentEmail
```

- **currentUsername**

```
private java.lang.String currentUsername
```

- **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

- **databaseRef**

```
private com.google.firebase.database.DatabaseReference  
databaseRef
```

- **storageReference**

```
private com.google.firebase.storage.StorageReference  
storageReference
```

- **store**

```
private Store store
```

- **pos**

```
private Position pos
```

- **Constructor Detail**

- **ProfileActivity**

```
public ProfileActivity()
```

- **Method Detail**

- **onCreate**

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

**Overrides:**

onCreate in class android.app.Activity

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **setUp**

```
private void setUp()
```

Get the user information from the database.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

- **validate**

```
private void validate()
```

Validates the input of email and password of the user and changes done by the user.

- **onActivityResult**

- ```
public void onActivityResult(int requestCode,
int resultCode, android.content.Intent data)
```

Gets called after a picture was chosen. The method gets the uri and uploads the uri to the database.

**Overrides:**  
onActivityResult in class android.app.Activity

**Parameters:**  
requestCode

resultCode

data

- **uploadPhoto**

```
private void uploadPhoto(android.net.Uri uri)
```

Uploads profile picture of the user to the database.

**Parameters:**  
uri - Uri of the picture in the Storage database.

## Class SettingsActivity

- `Java.lang.Object`
  - `android.content.Context`
    - `android.content.ContextWrapper`
      - `android.view.ContextThemeWrapper`
        - `android.app.Activity`
          - `com.example.ezway.Activities.SettingsActivity`

```
public class SettingsActivity
extends android.app.Activity
implements android.view.View.OnClickListener,
android.widget.CompoundButton.OnCheckedChangeListener
```

### • **Field Summary**

#### Fields

Modifier and Type	Field and Description
static java.lang.String	<code>APP_LANG</code>
static java.lang.String	<code>LANG_PREF</code>
static java.lang.String	<code>NOTF_PREF</code>
private <code>Position</code>	<code>pos</code>
private android.content.SharedPreferences	<code>preferences</code>
static java.lang.String	<code>PREFS_NAME</code>
private <code>Store</code>	<code>store</code>
static java.lang.String	<code>THEME_PREF</code>

### • **Constructor Summary**

#### Constructors

Constructor and Description
<code>SettingsActivity()</code>

### • **Method Summary**

#### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
private void	<code>changeBackground(int color)</code> Change the background color of views in layout according to the color passed in.
void	<code>loadLocale()</code> Loads the current language last that is saved on phone.

void	<b>onCheckedChanged</b> (android.widget.CompoundButton compoundButton, boolean b)
void	<b>onClick</b> (android.view.View v)
protected void	<b>onCreate</b> (android.os.Bundle savedInstanceState)
private void	<b>setLocale</b> (java.lang.String language) Sets the current language of the app to the language passed in.

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **LANG\_PREF**

```
public static final java.lang.String LANG_PREF
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **NOTF\_PREF**

```
public static final java.lang.String NOTF_PREF
```

### • **APP\_LANG**

```
public static final java.lang.String APP_LANG
```

### • **preferences**

```
private android.content.SharedPreferences preferences
```

### • **store**

```
private Store store
```

### • **pos**

```
private Position pos
```

## • **Constructor Detail**

### • **SettingsActivity**

```
public SettingsActivity()
```

## • **Method Detail**

### • **onCreate**

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

**Overrides:**

onCreate in class android.app.Activity

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**  
color

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **loadLocale**

```
public void loadLocale()
```

Loads the current language last that is saved on phone.

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

- **onCheckedChanged**

- ```
public void onCheckedChanged(android.widget.CompoundButton compoundButton, boolean b)
```

**Specified by:**  
onCheckedChanged in interface android.widget  
.CompoundButton.OnCheckedChangeListener

## Class ShopActivity

- **Java.lang.Object**
  - android.content.Context**
    - android.content.ContextWrapper**
      - android.view.ContextThemeWrapper**
        - android.app.Activity**
          - com.example.ezway.Activities.ShopActivity**

```
public class ShopActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • **Field Summary**

#### Fields

| Modifier and Type             | Field and Description          |
|-------------------------------|--------------------------------|
| private java.lang.String      | <a href="#">phoneNumOfShop</a> |
| static java.lang.String       | <a href="#">PREFS_NAME</a>     |
| private static int            | <a href="#">REQUEST_CALL</a>   |
| private <a href="#">Store</a> | <a href="#">store</a>          |
| static java.lang.String       | <a href="#">THEME_PREF</a>     |

### • **Constructor Summary**

#### Constructors

| Constructor and Description    |
|--------------------------------|
| <a href="#">ShopActivity()</a> |

### • **Method Summary**

#### All MethodsStatic MethodsInstance MethodsConcrete Methods

| Modifier and Type | Method and Description                                                                                                           |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------|
| private void      | <a href="#">changeBackground</a> (int color)<br>Change the background color of views in layout according to the color passed in. |
| static int        | <a href="#">getResId</a> (java.lang.String resName, java.lang.Class<?> c)                                                        |
| private void      | <a href="#">loadLocale</a> ()<br>Loads the current language last that is saved on phone.                                         |
| private void      | <a href="#">makePhoneCall</a> ()<br>Calls the shop                                                                               |



|                |                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| void           | <b>onClick</b> (android.view.View v)                                                                                                        |
| protected void | <b>onCreate</b> (android.os.Bundle savedInstanceState)                                                                                      |
| void           | <b>onRequestPermissionsResult</b> (int requestCode, java.lang.String[] permissions, int[] grantResults)<br>Gets permission to call the shop |
| private void   | <b>setLocale</b> (java.lang.String language)<br>Sets the current language of the app to the language passed in.                             |
| private void   | <b>setUp</b> ()<br>Change text according to the shop and initialize fields.                                                                 |

## • **Field Detail**

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

### • **REQUEST\_CALL**

```
private static final int REQUEST_CALL
```

### • **store**

```
private Store store
```

### • **phoneNumOfShop**

```
private java.lang.String phoneNumOfShop
```

## • **Constructor Detail**

### • **ShopActivity**

```
public ShopActivity()
```

## • **Method Detail**

### • **onCreate**

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

**Overrides:**

onCreate in class android.app.Activity

### • **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **getResId**

- ```
public static int getResId(java.lang.String resName,  
java.lang.Class<?> c)
```

**Parameters:**

resName - The resource's name.

c - The class of the resource.

**Returns:**

resource Id of the resource name passed in.

- **setUp**

```
private void setUp()
```

Change text according to the shop and initialize fields.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**

color

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**

onClick in interface android.view.View.OnClickListener

- **makePhoneCall**

```
private void makePhoneCall()
```

Calls the shop

- **onRequestPermissionsResult**

- ```
public void onRequestPermissionsResult(int requestCode,  
@NonNull java.lang.String[] permissions, @NonNull  
int[] grantResults)
```

Gets permission to call the shop

**Overrides:**

onRequestPermissionsResult in class android.app.Activity

**Parameters:**

requestCode

permissions

grantResults

## Class SignUpActivity

- `Java.lang.Object`

`android.content.Context`

`android.content.ContextWrapper`

`android.view.ContextThemeWrapper`

`android.app.Activity`

`com.example.ezway.Activities.SignUpActivity`

```
public class SignUpActivity
extends android.app.Activity
implements android.view.View.OnClickListener
```

### • **Field Summary**

| <b>Fields</b>                                          |                            |
|--------------------------------------------------------|----------------------------|
| Modifier and Type                                      | Field and Description      |
| private android.widget.Button                          | <code>btnDatePicker</code> |
| private com.google.firebase.database.DatabaseReference | <code>database</code>      |
| private android.widget.EditText                        | <code>etEmail</code>       |
| private android.widget.EditText                        | <code>etPassword</code>    |
| private android.widget.EditText                        | <code>etPasswordCon</code> |
| private android.widget.EditText                        | <code>etUsername</code>    |
| private com.google.firebase.auth.FirebaseAuth          | <code>mAuth</code>         |
| private com.google.firebase.auth.FirebaseUser          | <code>mUser</code>         |
| private <code>MyDate</code>                            | <code>pickedDate</code>    |
| static java.lang.String                                | <code>PREFS_NAME</code>    |
| private android.widget.RadioGroup                      | <code>radioGroup</code>    |
| static java.lang.String                                | <code>THEME_PREF</code>    |

### • **Constructor Summary**

| <b>Constructors</b>           |  |
|-------------------------------|--|
| Constructor and Description   |  |
| <code>SignUpActivity()</code> |  |

### • **Method Summary**

| All Methods              | Instance Methods                                                                                                        | Concrete Methods |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------|------------------|
| Modifier and Type        | Method and Description                                                                                                  |                  |
| private void             | <b>changeBackground</b> (int color)<br>Change the background color of views in layout according to the color passed in. |                  |
| private void             | <b>createAccount</b> (java.lang.String email)<br>Creates an account in the database.                                    |                  |
| private java.lang.String | <b>decideGender</b> ()<br>Gets the radio button input from the user about their gender.                                 |                  |
| private void             | <b>loadLocale</b> ()<br>Loads the current language last that is saved on phone.                                         |                  |
| void                     | <b>onClick</b> (android.view.View v)                                                                                    |                  |
| protected void           | <b>onCreate</b> (android.os.Bundle savedInstanceState)                                                                  |                  |
| int                      | <b>onRadioButtonClicked</b> (android.view.View view)                                                                    |                  |
| private void             | <b>setLocale</b> (java.lang.String language)<br>Sets the current language of the app to the language passed in.         |                  |
| void                     | <b>setPickedDate</b> (int year, int month, int day)                                                                     |                  |
| private void             | <b>showDatePicker</b> ()                                                                                                |                  |
| private void             | <b>validate</b> ()<br>Validates the fields that are required to be filled by the user.                                  |                  |

#### • **Field Detail**

##### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

##### • **THEME\_PREF**

```
public static final java.lang.String THEME_PREF
```

##### • **etUsername**

```
private android.widget.EditText etUsername
```

##### • **etPassword**

```
private android.widget.EditText etPassword
```

##### • **etPasswordCon**

```
private android.widget.EditText etPasswordCon
```

##### • **etEmail**

```
private android.widget.EditText etEmail
```

- **btnDatePicker**

```
private android.widget.Button btnDatePicker
```

- **radioGroup**

```
private android.widget.RadioGroup radioGroup
```

- **database**

```
private com.google.firebase.database.DatabaseReference  
database
```

- **mAuth**

```
private com.google.firebase.auth.FirebaseAuth mAuth
```

- **mUser**

```
private com.google.firebase.auth.FirebaseUser mUser
```

- **pickedDate**

```
private MyDate pickedDate
```

- **Constructor Detail**

- **SignUpActivity**

```
public SignUpActivity()
```

- **Method Detail**

- **onCreate**

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

**Overrides:**  
onCreate in class android.app.Activity

- **onClick**

```
public void onClick(android.view.View v)
```

**Specified by:**  
onClick in interface android.view.View.OnClickListener

- **loadLocale**

```
private void loadLocale()
```

Loads the current language last that is saved on phone.

- **setLocale**

```
private void setLocale(java.lang.String language)
```

Sets the current language of the app to the language passed in.

- **validate**

```
private void validate()
```

Validates the fields that are required to be filled by the user.

- **createAccount**

```
private void createAccount(java.lang.String email)
```

Creates an account in the database.

**Parameters:**

email – Email passed in by the user.

- **decideGender**

```
private java.lang.String decideGender()
```

Gets the radio button input from the user about their gender.

**Returns:**

String that represents the user's chosen gender.

- **changeBackground**

```
private void changeBackground(int color)
```

Change the background color of views in layout according to the color passed in.

**Parameters:**

color

- **showDatePicker**

```
private void showDatePicker()
```

- **setPickedDate**

- ```
public void setPickedDate(int year, int month, int day)
```

- **onRadioButtonClicked**

```
public int onRadioButtonClicked(android.view.View view)
```

## Adapters

### Class DiscountRecyclerViewAdapter

- `Java.lang.Object`

`androidx.recyclerview.widget.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

`com.example.ezway.Adapters.MyAdapter`

`com.example.ezway.Adapters.DiscountRecyclerViewAdapter`

```
public class DiscountRecyclerViewAdapter
extends MyAdapter
```

#### • **Nested Class Summary**

- Nested classes/interfaces inherited from class `com.example.ezway.Adapters.MyAdapter`

`MyAdapter.MyViewHolder`

#### • **Field Summary**

##### Fields

Modifier and Type	Field and Description
<code>private java.util.List&lt;Discount&gt;</code>	<code>discounts</code>

- Fields inherited from class `com.example.ezway.Adapters.MyAdapter`

`activity, themePref`

#### • **Constructor Summary**

##### Constructors

##### Constructor and Description

`DiscountRecyclerViewAdapter`(`java.util.List<Discount>` discounts, `android.app.Activity` activity, `boolean` themePref)  
Constructor

#### • **Method Summary**

##### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
<code>int</code>	<code>getItemCount()</code>
<code>void</code>	<code>onBindViewHolder(MyAdapter.MyViewHolder holder, int position)</code>

	Sets up the preview about the discount in the view holder.
<b>MyAdapter.MyViewHolder</b>	<b>onCreateViewHolder</b> (android.view.ViewGroup parent, int viewType) Changes the theme of the view holder according to the user's preferred theme.

### • **Field Detail**

#### • **discounts**

```
private java.util.List<Discount> discounts
```

### • **Constructor Detail**

#### • **DiscountRecyclerViewAdapter**

- ```
public DiscountRecyclerViewAdapter(java.util.List<Discount> discounts, android.app.Activity activity, boolean themePref)
```

Constructor

**Parameters:**

discounts - Array list of discounts of the user.

activity - Activity from where the adapter is being called from.

themePref - Theme preferred by the user.

### • **Method Detail**

#### • **onCreateViewHolder**

- ```
@NonNull
```
- ```
public MyAdapter.MyViewHolder onCreateViewHolder(@NonNull android.view.ViewGroup parent, int viewType)
```

Changes the theme of the view holder according to the user's preferred theme.

**Specified by:**

`onCreateViewHolder` in class `MyAdapter`

**Parameters:**

parent

viewType

**Returns:**

An instance of the view holder that holds information about the discount.

#### • **onBindViewHolder**

- ```
public void onBindViewHolder(@NonNull MyAdapter.MyViewHolder holder, int position)
```



Sets up the preview about the discount in the view holder.

**Specified by:**

`onBindViewHolder` in class `MyAdapter`

**Parameters:**

`holder`

`position`

- **getItemCount**

```
public int getItemCount()
```

**Specified by:**

`getItemCount` in class `MyAdapter`

## Class FavShopsAdapter

- `Java.lang.Object`

`androidx.recyclerview.widget.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

`com.example.ezway.Adapters.MyAdapter`

`com.example.ezway.Adapters.FavShopsAdapter`

```
public class FavShopsAdapter
extends MyAdapter
```

### • Nested Class Summary

- Nested classes/interfaces inherited from class `com.example.ezway.Adapters.MyAdapter`

`MyAdapter.MyViewHolder`

### • Field Summary

#### Fields

##### Modifier and Type

##### Field and Description

`private java.util.List<java.lang.String>` **stores**

- Fields inherited from class `com.example.ezway.Adapters.MyAdapter`

`activity, themePref`

### • Constructor Summary

#### Constructors

##### Constructor and Description

**FavShopsAdapter** (`java.util.List<java.lang.String>` stores, `android.app.Activity` activity, `boolean` themePref)  
Constructor

### • Method Summary

#### All Methods Instance Methods Concrete Methods

##### Modifier and Type

##### Method and Description

`int`

**getItemCount()**

`void`

**onBindViewHolder** (`MyAdapter.MyViewHolder` holder, `int` position)  
Sets up the view holder and gets the store from the user's favorite places list.

## MyAdapter.MyViewHolder

**onCreateViewHolder** (android.view.ViewGroup parent, int viewType)  
Changes the theme of the view holder according to the user's preferred theme.

### • Field Detail

#### • stores

```
private java.util.List<java.lang.String> stores
```

### • Constructor Detail

#### • FavShopsAdapter

- ```
public FavShopsAdapter(java.util.List<java.lang.String> stores, android.app.Activity activity, boolean themePref)
```

Constructor

Parameters:

stores

activity

themePref

### • Method Detail

#### • onCreateViewHolder

- @NonNull
- ```
public MyAdapter.MyViewHolder onCreateViewHolder(@NonNull android.view.ViewGroup parent, int viewType)
```

Changes the theme of the view holder according to the user's preferred theme.

Specified by:

`onCreateViewHolder` in class `MyAdapter`

Parameters:

parent

viewType

Returns:

An instance of the view holder that holds the store name.

#### • onBindViewHolder

- ```
public void onBindViewHolder(@NonNull MyAdapter.MyViewHolder holder, int position)
```

Sets up the view holder and gets the store from the user's favorite places list.

Specified by:

`onBindViewHolder` in class `MyAdapter`

**Parameters:**

`holder`

`position`

- **`getItemCount`**

`public int getItemCount()`

**Specified by:**

`getItemCount` in class `MyAdapter`

## Class MyAdapter

- `Java.lang.Object`

`androidx.recyclerview.widget.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

`com.example.ezway.Adapters.MyAdapter`

- Direct Known Subclasses:

`DiscountRecyclerViewAdapter`, `FavShopsAdapter`, `ReviewAdapter`

```
public abstract class MyAdapter
extends androidx.recyclerview.widget.RecyclerView.Adapter<MyAdapter
.MyViewHolder>
```

### • Nested Class Summary

#### Nested Classes

##### Modifier and Type

##### Class and Description

class

`MyAdapter.MyViewHolder`

### • Field Summary

#### Fields

##### Modifier and Type

##### Field and Description

protected android.app.Activity

`activity`

protected boolean

`themePref`

### • Constructor Summary

#### Constructors

##### Constructor and Description

`MyAdapter()`

### • Method Summary

#### All MethodsInstance MethodsAbstract Methods

##### Modifier and Type

##### Method and Description

abstract int

`getItemCount()`

abstract void

`onBindViewHolder(MyAdapter.MyViewHolder holder, int position)`

abstract `MyAdapter.MyViewHolder`

`onCreateViewHolder(android.view.ViewGroup parent, int viewType)`

### • Field Detail

- **activity**

`protected android.app.Activity activity`

- **themePref**

`protected boolean themePref`

- **Constructor Detail**

- **MyAdapter**

`public MyAdapter()`

- **Method Detail**

- **onCreateViewHolder**

- `@NonNull`  
`Public abstract MyAdapter.MyViewHolder`  
`onCreateViewHolder(@NonNull android.view.ViewGroup`  
`parent, int viewType)`

**Specified by:**

`onCreateViewHolder` in class `androidx.recyclerview.widget`  
`.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

- **onBindViewHolder**

- `public abstract void onBindViewHolder(@NonNull`  
`MyAdapter.MyViewHolder holder, int position)`

**Specified by:**

`onBindViewHolder` in class `androidx.recyclerview.widget`  
`.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

- **getItemCount**

`public abstract int getItemCount()`

**Specified by:**

`getItemCount` in class `androidx.recyclerview.widget`  
`.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

## Class MyAdapter.MyViewHolder

- `Java.lang.Object`

`androidx.recyclerview.widget.RecyclerView.ViewHolder`

`com.example.ezway.Adapters.MyAdapter.MyViewHolder`

- Enclosing class:

`MyAdapter`

```
public class MyAdapter.MyViewHolder
    extends androidx.recyclerview.widget.RecyclerView.ViewHolder
```

### • *Field Summary*

- Fields inherited from class `androidx.recyclerview.widget.RecyclerView.ViewHolder`

`itemView`

### • *Constructor Summary*

#### Constructors

##### Constructor and Description

`MyViewHolder` (`android.view.View itemView`)

### • *Constructor Detail*

- `MyViewHolder`

- `public MyViewHolder(@NonNull android.view.View itemView)`

## Class RecyclerView.Adapter

- `java.lang.Object`

`androidx.recyclerview.widget.RecyclerView.Adapter<RecyclerView.ViewHolder>`

`com.example.ezway.Adapters.RecyclerViewAdapter`

- All Implemented Interfaces:

`android.widget.Filterable`

```
public class RecyclerViewAdapter
extends androidx.recyclerview.widget.RecyclerView.Adapter
<RecyclerView.ViewHolder> implements android.widget.Filterable
```

### • Nested Class Summary

#### Nested Classes

| Modifier and Type       | Class and Description                       |
|-------------------------|---------------------------------------------|
| (package private) class | <code>RecyclerViewAdapter.ViewHolder</code> |

### • Field Summary

#### Fields

| Modifier and Type                                           | Field and Description            |
|-------------------------------------------------------------|----------------------------------|
| private <code>android.app.Activity</code>                   | <code>activity</code>            |
| private <code>java.util.List&lt;java.lang.String&gt;</code> | <code>allSuggestions</code>      |
| private <code>android.widget.Filter</code>                  | <code>filter</code>              |
| private <code>java.util.List&lt;java.lang.String&gt;</code> | <code>suggestionsFiltered</code> |
| private <code>boolean</code>                                | <code>themePref</code>           |

### • Constructor Summary

#### Constructors

##### Constructor and Description

`RecyclerViewAdapter`(`java.util.List<java.lang.String>` shops, `boolean` themePref, `android.app.Activity` activity)  
Constructor

### • Method Summary

#### All MethodsInstance MethodsConcrete Methods

| Modifier and Type                  | Method and Description                            |
|------------------------------------|---------------------------------------------------|
| <code>android.widget.Filter</code> | <code>getFilter()</code>                          |
| <code>int</code>                   | <code>getItemCount()</code>                       |
| <code>void</code>                  | <code>onBindViewHolder(RecyclerViewAdapter</code> |



**.MyViewHolder holder, int position)**  
Sets up the view holder's name shown to the user to be a shop's name.

**RecyclerView.MyViewHolder**

**onCreateViewHolder**(android.view.ViewGroup parent, int viewType)  
Changes the theme of the view holder according to the user's preferred theme.

## • **Field Detail**

### • **activity**

```
private android.app.Activity activity
```

### • **themePref**

```
private boolean themePref
```

### • **suggestionsFiltered**

```
private java.util.List<java.lang.String>
suggestionsFiltered
```

### • **allSuggestions**

```
private java.util.List<java.lang.String> allSuggestions
```

### • **filter**

```
private android.widget.Filter filter
```

## • **Constructor Detail**

### • **RecyclerViewAdapter**

- `public RecyclerViewAdapter(java.util.List<java.lang.String> shops, boolean themePref, android.app.Activity activity)`

Constructor

**Parameters:**

shops - Array of shops.

themePref - Theme preferred by the user.

activity - Activity where the recycler adapter is being used.

## • **Method Detail**

### • **onCreateViewHolder**

- `@NonNull`
- `public RecyclerViewAdapter.MyViewHolder onCreateViewHolder(@NonNull android.view.ViewGroup parent, int viewType)`

Changes the theme of the view holder according to the user's preferred theme.

**Specified by:**

onCreateViewHolder in class androidx.recyclerview.widget  
.RecyclerView.Adapter<RecyclerView.Adapter.MyViewHolder>

**Parameters:**

parent

viewType

**Returns:**

An instance of the view holder that holds information  
about the discount.

- **onBindViewHolder**

- `public void onBindViewHolder(@NonNull RecyclerView.Adapter  
.MyViewHolder holder, int position)`

Sets up the view holder's name shown to the user to be a shop's name.

**Specified by:**

onBindViewHolder in class androidx.recyclerview.widget  
.RecyclerView.Adapter<RecyclerView.Adapter.MyViewHolder>

**Parameters:**

holder

position

- **getItemCount**

`public int getItemCount()`

**Specified by:**

getItemCount in class androidx.recyclerview.widget  
.RecyclerView.Adapter<RecyclerView.Adapter.MyViewHolder>

- **getFilter**

`public android.widget.Filter getFilter()`

**Specified by:**

getFilter in interface android.widget.Filterable

## Class RecyclerView.MyViewHolder

- **Java.lang.Object**  
**androidx.recyclerview.widget.RecyclerView.ViewHolder**  
**com.example.ezway.Adapters.RecyclerView.MyViewHolder**
- **All Implemented Interfaces:**  
**android.view.View.OnClickListener**  
**Enclosing class:**  
**RecyclerView**  
**class RecyclerView.MyViewHolder**  
**extends androidx.recyclerview.widget.RecyclerView.ViewHolder**  
**implements android.view.View.OnClickListener**

### • **Field Summary**

#### **Fields**

| Modifier and Type                                | Field and Description |
|--------------------------------------------------|-----------------------|
| private <b>Store</b>                             | <b>store</b>          |
| (package private) <b>android.widget.TextView</b> | <b>textView</b>       |

### • **Constructor Summary**

#### **Constructors**

##### **Constructor and Description**

**MyViewHolder**(**android.view.View** itemView)  
 Constructor

### • **Method Summary**

#### **All Methods****Instance Methods****Concrete Methods**

| Modifier and Type | Method and Description                          |
|-------------------|-------------------------------------------------|
| void              | <b>onClick</b> ( <b>android.view.View</b> view) |

### • **Field Detail**

#### • **textView**

**android.widget.TextView** textView

#### • **store**

private **Store** store

### • **Constructor Detail**

- **MyViewHolder**

- `public MyViewHolder(@NonNull android.view.View itemView)`

Constructor

**Parameters:**  
itemView

- **Method Detail**

- **onClick**

`public void onClick(android.view.View view)`

**Specified by:**  
onClick in interface android.view.View.OnClickListener

## Class ReviewAdapter

- `Java.lang.Object`

`androidx.recyclerview.widget.RecyclerView.Adapter<MyAdapter.MyViewHolder>`

`com.example.ezway.Adapters.MyAdapter`

`com.example.ezway.Adapters.ReviewAdapter`

```
public class ReviewAdapter
extends MyAdapter
```

### • Nested Class Summary

- Nested classes/interfaces inherited from class `com.example.ezway.Adapters.MyAdapter`

`MyAdapter.MyViewHolder`

### • Field Summary

#### Fields

| Modifier and Type                                 | Field and Description |
|---------------------------------------------------|-----------------------|
| <code>private java.util.List&lt;Review&gt;</code> | <code>reviews</code>  |

- Fields inherited from class `com.example.ezway.Adapters.MyAdapter`

`activity, themePref`

### • Constructor Summary

#### Constructors

##### Constructor and Description

`ReviewAdapter`(`java.util.List<Review>` reviews, `android.app.Activity` activity, `boolean` themePref)  
Constructor

### • Method Summary

#### All MethodsStatic MethodsInstance MethodsConcrete Methods

| Modifier and Type       | Method and Description                                                                                  |
|-------------------------|---------------------------------------------------------------------------------------------------------|
| <code>int</code>        | <code>getItemCount()</code>                                                                             |
| <code>static int</code> | <code>getResId</code> ( <code>java.lang.String</code> resName, <code>java.lang.Class&lt;?&gt;</code> c) |
| <code>void</code>       | <code>onBindViewHolder</code> ( <code>MyAdapter.MyViewHolder</code> holder, <code>int</code> position)  |

Sets up the reviews of the stores in each view holder.

**MyAdapter.MyViewHolder**

**onCreateViewHolder** (android.view.ViewGroup parent, int viewType)  
Changes the theme of the view holder according to the user's preferred theme.

## • **Field Detail**

### • **reviews**

```
private java.util.List<Review> reviews
```

## • **Constructor Detail**

### • **ReviewAdapter**

- `public ReviewAdapter(java.util.List<Review> reviews, android.app.Activity activity, boolean themePref)`

Constructor

**Parameters:**  
reviews

activity

themePref - Theme preferred by the user.

## • **Method Detail**

### • **getResId**

- `public static int getResId(java.lang.String resName, java.lang.Class<?> c)`

**Parameters:**  
resName - The resource's name.

c - The class of the resource.

**Returns:**  
resource Id of the resource name passed in.

### • **onCreateViewHolder**

- `@NonNull public MyAdapter.MyViewHolder onCreateViewHolder(@NonNull android.view.ViewGroup parent, int viewType)`

Changes the theme of the view holder according to the user's preferred theme.

**Specified by:**  
`onCreateViewHolder` in class `MyAdapter`

**Parameters:**  
parent

viewType

**Returns:**

An instance of the view holder that holds information about the discount.

- **onBindViewHolder**

- ```
public void onBindViewHolder(@NonNull MyAdapter  
.MyViewHolder holder, int position)
```

Sets up the reviews of the stores in each view holder.

**Specified by:**

`onBindViewHolder` in class `MyAdapter`

**Parameters:**

`holder`

`position` - The position of the review in the list.

- **getItemCount**

```
public int getItemCount()
```

**Specified by:**

`getItemCount` in class `MyAdapter`

## Base Classes

### Class BeaconInfo

- `Java.lang.Object`  
`com.example.ezway.BaseClasses.BeaconInfo`

```
public class BeaconInfo
extends java.lang.Object
```

#### • Field Summary

##### Fields

Modifier and Type	Field and Description
private double	<code>distance</code>
private int	<code>index</code>

#### • Constructor Summary

##### Constructors

###### Constructor and Description

```
BeaconInfo(BeaconInfo info)  
BeaconInfo(int index, double distance)
```

#### • Method Summary

##### All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
double	<code>getDistance()</code>
int	<code>getIndex()</code>
void	<code>setDistance(double distance)</code>
void	<code>setIndex(int index)</code>
java.lang.String	<code>toString()</code>

#### • Field Detail

##### • index

```
private int index
```

##### • distance

```
private double distance
```

#### • Constructor Detail



- **BeaconInfo**

- `public BeaconInfo(int index, double distance)`

- **BeaconInfo**

`public BeaconInfo(BeaconInfo info)`

- **Method Detail**

- **getIndex**

`public int getIndex()`

- **getDistance**

`public double getDistance()`

- **setIndex**

`public void setIndex(int index)`

- **setDistance**

`public void setDistance(double distance)`

- **toString**

`public java.lang.String toString()`

**Overrides:**

`toString` in class `java.lang.Object`

## Class Discount

- **Java.lang.Object**  
**com.example.ezway.BaseClasses.Discount**

```
public class Discount
extends java.lang.Object
```

### • **Field Summary**

#### **Fields**

Modifier and Type	Field and Description
private java.lang.String	<b>content</b>
private <b>MyDate</b>	<b>expirationDate</b>
private java.lang.String	<b>id</b>
private java.lang.String	<b>storeName</b>
private java.lang.String	<b>title</b>
private java.lang.String	<b>userId</b>

### • **Constructor Summary**

#### **Constructors**

##### **Constructor and Description**

**Discount()**

**Discount**(java.lang.String id, java.lang.String title, java.lang.String storeName, java.lang.String userId, **MyDate** expirationDate, java.lang.String content)

### • **Method Summary**

#### **All Methods****Instance Methods****Concrete Methods**

Modifier and Type	Method and Description
java.lang.String	<b>getContent()</b>
<b>MyDate</b>	<b>getExpirationDate()</b>
java.lang.String	<b>getId()</b>
java.lang.String	<b>getStoreName()</b>
java.lang.String	<b>getTitle()</b>
java.lang.String	<b>getUserId()</b>
void	<b>setContent</b> (java.lang.String content)
void	<b>setExpirationDate</b> ( <b>MyDate</b> expirationDate)
void	<b>setId</b> (java.lang.String id)

void	<code>setStoreName(java.lang.String storeName)</code>
void	<code>setTitle(java.lang.String title)</code>
void	<code>setUserId(java.lang.String userId)</code>
java.lang.String	<code>toString()</code>

- **Field Detail**

- **id**

```
private java.lang.String id
```

- **title**

```
private java.lang.String title
```

- **storeName**

```
private java.lang.String storeName
```

- **userId**

```
private java.lang.String userId
```

- **expirationDate**

```
private MyDate expirationDate
```

- **content**

```
private java.lang.String content
```

- **Constructor Detail**

- **Discount**

- `public Discount(java.lang.String id, java.lang.String title, java.lang.String storeName, java.lang.String userId, MyDate expirationDate, java.lang.String content)`

- **Discount**

```
public Discount()
```

- **Method Detail**

- **getId**

```
public java.lang.String getId()
```

- **setId**

```
public void setId(java.lang.String id)
```

- **getTitle**

```
public java.lang.String getTitle()
```

- **setTitle**

```
public void setTitle(java.lang.String title)
```

- **getStoreName**

```
public java.lang.String getStoreName()
```

- **setStoreName**

```
public void setStoreName(java.lang.String storeName)
```

- **getUserId**

```
public java.lang.String getUserId()
```

- **setUserId**

```
public void setUserId(java.lang.String userId)
```

- **getExpirationDate**

```
public MyDate getExpirationDate()
```

- **setExpirationDate**

```
public void setExpirationDate(MyDate expirationDate)
```

- **getContent**

```
public java.lang.String getContent()
```

- **setContent**

```
public void setContent(java.lang.String content)
```

- **toString**

- @NonNull  
public java.lang.String toString()

**Overrides:**

toString in class java.lang.Object

## Class MyDate

- **Java.lang.Object**  
**com.example.ezway.BaseClasses.MyDate**

```
public class MyDate
extends java.lang.Object
```

### • **Field Summary**

#### Fields

Modifier and Type	Field and Description
private int	<b>day</b>
private int	<b>month</b>
private int	<b>year</b>

### • **Constructor Summary**

#### Constructors

##### Constructor and Description

**MyDate**()

**MyDate**(int day, int month, int year)

### • **Method Summary**

#### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
boolean	<b>equals</b> (java.lang.Object obj)
int	<b>getDay</b> ()
int	<b>getMonth</b> ()
int	<b>getYear</b> ()
void	<b>setDay</b> (int day)
void	<b>setMonth</b> (int month)
void	<b>setYear</b> (int year)
java.lang.String	<b>toString</b> ()
java.lang.String	<b>toString2</b> ()

### • **Field Detail**

#### • **year**

private int year

- **month**

```
private int month
```

- **day**

```
private int day
```

#### • **Constructor Detail**

- **MyDate**

```
public MyDate()
```

- **MyDate**

- `public MyDate(int day, int month, int year)`

#### • **Method Detail**

- **getYear**

```
public int getYear()
```

- **setYear**

```
public void setYear(int year)
```

- **getMonth**

```
public int getMonth()
```

- **setMonth**

```
public void setMonth(int month)
```

- **getDay**

```
public int getDay()
```

- **setDay**

```
public void setDay(int day)
```

- **toString**

- `@NonNull`  
`public java.lang.String toString()`

**Overrides:**  
`toString` in class `java.lang.Object`

- **toString2**

```
public java.lang.String toString2()
```

- **equals**

- `public boolean equals(@Nullable java.lang.Object obj)`

**Overrides:**  
`equals` in class `java.lang.Object`

## Class Navigation

- `Java.lang.Object`  
`com.example.ezway.BaseClasses.Navigation`

```
public class Navigation
extends java.lang.Object
```

### • **Field Summary**

#### Fields

Modifier and Type	Field and Description
private int[]	<code>indices</code>
private <code>BeaconInfo</code> [][]	<code>lastThreeReceived</code>
private static int	<code>NUMBER_OF_BEACONS</code>
private static int	<code>PIXEL_DISTANCE_STANDARD</code>

### • **Constructor Summary**

#### Constructors

##### Constructor and Description

`Navigation()`  
Constructor

### • **Method Summary**

#### All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
private static void	<code>add</code> (int direction, <code>Position</code> pos, int pixelDis) Changes the position passed in by the two other parameters.
void	<code>changeData</code> ( <code>BeaconInfo</code> info) Updates the information stored of the beacons.
static <code>BeaconInfo</code>	<code>convertToInfo</code> (org.altbeacon.beacon. <code>Beacon</code> beacon) Converts the Beacon object passed in into a <code>BeaconInfo</code>
private static int	<code>findIndexByName</code> (java.lang.String beaconName)
static <code>BeaconInfo</code> [][]	<code>getAverageDis</code> ( <code>BeaconInfo</code> [][] infos) Calculates the average distance of the information stored.
private static double[]	<code>getAvgDisAndCount</code> ( <code>BeaconInfo</code> [][] infos)

	Calculates the average distance of a specific beacon.
private static <b>BeaconInfo</b> []	<b>getClosestBeacons</b> ( <b>BeaconInfo</b> [] infos) Calculates and returns 2 closest beacons.
private static double[]	<b>getClosestTwoDistances</b> (double[] distances) Calculates and returns 2 closest distances.
private static int	<b>getDirection</b> ( <b>BeaconInfo</b> [] closeBeacons) Gets the 2 closest beacon array and calculates the direction the user needs to navigate to.
<b>BeaconInfo</b> [][]	<b>getInfos</b> ()
static <b>Position</b>	<b>getPosition</b> ( <b>BeaconInfo</b> [] infos) Estimates the user's position by calculations.
static <b>Position</b>	<b>indexToPosition</b> (int beaconIndex)
private static <b>BeaconInfo</b> []	<b>removeEmptyInfo</b> ( <b>BeaconInfo</b> [] infos) Gets an array and returns an array without 0 distance beacon infos.
private static <b>BeaconInfo</b> []	<b>toArray</b> (java.util.ArrayList< <b>BeaconInfo</b> > list) Gets an array list and returns an array.

## • **Field Detail**

### • **PIXEL\_DISTANCE\_STANDARD**

```
private static final int PIXEL_DISTANCE_STANDARD
```

### • **NUMBER\_OF\_BEACONS**

```
private static final int NUMBER_OF_BEACONS
```

### • **lastThreeReceived**

```
private BeaconInfo[][] lastThreeReceived
```

### • **indices**

```
private int[] indices
```

## • **Constructor Detail**

### • **Navigation**

```
public Navigation()
```

Constructor

## • **Method Detail**

### • **getPosition**

```
public static Position getPosition(BeaconInfo[] infos)
```



Estimates the user's position by calculations.

**Parameters:**

infos - An array of BeaconInfo objects.

**Returns:**

An instance of the estimated position of the user.

- **removeEmptyInfo**

```
private static BeaconInfo[] removeEmptyInfo(BeaconInfo[] infos)
```

Gets an array and returns an array without 0 distance beacon infos.

**Parameters:**

infos - An array of BeaconInfo objects.

**Returns:**

A modified array of BeaconInfo objects.

- **toArray**

```
private static BeaconInfo[] toArray(java.util.ArrayList<BeaconInfo> list)
```

Gets an array list and returns an array.

**Parameters:**

list - List that holds BeaconInfo objects in order (By distance).

**Returns:**

An array of BeaconInfo objects.

- **getDirection**

```
private static int getDirection(BeaconInfo[] closeBeacons)
```

Gets the 2 closest beacon array and calculates the direction the user needs to navigate to.

**Parameters:**

closeBeacons - An array of the 2 closest beacons.

**Returns:**

An int that represents direction.

- **add**

- ```
private static void add(int direction, Position pos, int pixelDis)
```

Changes the position passed in by the two other parameters.

**Parameters:**

direction - The direction from the closest beacon to the user.

pos - The position of the closest beacon.

pixelDis - The distance calculated between the beacon and the user.

- **indexToPosition**

```
public static Position indexToPosition(int beaconIndex)
```

**Parameters:**

beaconIndex - The beacon's index in its name.

**Returns:**

A beacon position by using the index passed in.

- **getClosestBeacons**

```
private static BeaconInfo[] getClosestBeacons(BeaconInfo[] infos)
```

Calculates and returns 2 closest beacons.

**Parameters:**

infos - Array of beacon infos.

**Returns:**

returns 2 closest beacons.

- **getClosestTwoDistances**

```
private static double[] getClosestTwoDistances(double[] distances)
```

Calculates and returns 2 closest distances.

**Parameters:**

distances - An array of the two closest beacons' distances.

**Returns:**

returns 2 closest distances.

- **convertToInfo**

```
public static BeaconInfo convertToInfo(org.altbeacon.beacon.Beacon beacon)
```

Converts the Beacon object passed in into a BeaconInfo object.

**Parameters:**

beacon - A Beacon object.

**Returns:**

An instance of a BeaconInfo that represents a beacon.

- **findIndexByName**

```
private static int findIndexByName(java.lang.String beaconName)
```

**Parameters:**

beaconName - Name of the beacon.

**Returns:**

The serial number of the beacon.

- **changeData**

```
public void changeData(BeaconInfo info)
```

Updates the information stored of the beacons.

**Parameters:**

info – BeaconInfo object that holds information about the beacon.

- **getAverageDis**

```
public static BeaconInfo[] getAverageDis(BeaconInfo[][]  
infos)
```

Calculates the average distance of the information stored.

**Parameters:**

infos – matrix of beacon infos.

**Returns:**

An array that stores average calculation of the distances of each beacon.

- **getAvgDisAndCount**

```
private static double[] getAvgDisAndCount(BeaconInfo[]  
infos)
```

Calculates the average distance of a specific beacon.

**Parameters:**

infos – Array of beacon infos.

**Returns:**

An average calculation of an array.

- **getInfos**

```
public BeaconInfo[][] getInfos()
```

## Class Position

- **Java.lang.Object**  
**com.example.ezway.BaseClasses.Position**
- **All Implemented Interfaces:**  
java.io.Serializable

```
public class Position
extends java.lang.Object
implements java.io.Serializable
```

### • Field Summary

#### Fields

| Modifier and Type | Field and Description |
|-------------------|-----------------------|
| private int       | <b>x</b>              |
| private int       | <b>y</b>              |
| private int       | <b>z</b>              |

### • Constructor Summary

#### Constructors

##### Constructor and Description

```
Position()
Position(int x, int y, int z)
```

### • Method Summary

#### All MethodsInstance MethodsConcrete Methods

| Modifier and Type | Method and Description |
|-------------------|------------------------|
| int               | <b>getX()</b>          |
| int               | <b>getY()</b>          |
| int               | <b>getZ()</b>          |
| void              | <b>moveDown()</b>      |
| void              | <b>moveLeft()</b>      |
| void              | <b>moveRight()</b>     |
| void              | <b>moveUp()</b>        |
| void              | <b>setX(int x)</b>     |
| void              | <b>setY(int y)</b>     |
| void              | <b>setZ(int z)</b>     |

java.lang.String

toString()

#### • **Field Detail**

- **x**

private int x

- **y**

private int y

- **z**

private int z

#### • **Constructor Detail**

- **Position**

- public Position(int x, int y, int z)

- **Position**

public Position()

#### • **Method Detail**

- **moveUp**

public void moveUp()

- **moveDown**

public void moveDown()

- **moveLeft**

public void moveLeft()

- **moveRight**

public void moveRight()

- **getX**

public int getX()

- **setX**

public void setX(int x)

- **getY**

public int getY()

- **setY**

public void setY(int y)

- **getZ**

```
public int getZ()
```

- **setZ**

```
public void setZ(int z)
```

- **toString**

```
public java.lang.String toString()
```

**Overrides:**

toString in class java.lang.Object

## Class Review

- **Java.lang.Object**  
**com.example.ezway.BaseClasses.Review**

```
public class Review
extends java.lang.Object
```

### • **Field Summary**

#### **Fields**

| Modifier and Type        | Field and Description |
|--------------------------|-----------------------|
| private <b>MyDate</b>    | <b>date</b>           |
| private java.lang.String | <b>description</b>    |
| private java.lang.String | <b>id</b>             |
| private double           | <b>rating</b>         |
| private java.lang.String | <b>storeId</b>        |
| private java.lang.String | <b>storeName</b>      |
| private java.lang.String | <b>username</b>       |

### • **Constructor Summary**

#### **Constructors**

##### Constructor and Description

**Review**()

**Review**(java.lang.String id, java.lang.String storeId, java.lang.String username, **MyDate** date, double rating, java.lang.String description, java.lang.String storeName)

### • **Method Summary**

#### **All Methods**Instance MethodsConcrete Methods

| Modifier and Type | Method and Description               |
|-------------------|--------------------------------------|
| <b>MyDate</b>     | <b>getDate</b> ()                    |
| java.lang.String  | <b>getDescription</b> ()             |
| java.lang.String  | <b>getId</b> ()                      |
| double            | <b>getRating</b> ()                  |
| java.lang.String  | <b>getStoreId</b> ()                 |
| java.lang.String  | <b>getStoreName</b> ()               |
| java.lang.String  | <b>getUsername</b> ()                |
| void              | <b>setDate</b> ( <b>MyDate</b> date) |

|      |                                                         |
|------|---------------------------------------------------------|
| void | <b>setDescription</b> (java.lang.String description)    |
| void | <b>setId</b> (java.lang.String id)                      |
| void | <b>setRating</b> (double rating)                        |
| void | <b>setStoreId</b> (java.lang.String storeId)            |
| void | <b>setStoreName</b> (java.lang.String storeName)        |
| void | <b b="" setusername<="">(java.lang.String username)</b> |

#### • **Field Detail**

##### • **id**

```
private java.lang.String id
```

##### • **storeId**

```
private java.lang.String storeId
```

##### • **username**

```
private java.lang.String username
```

##### • **date**

```
private MyDate date
```

##### • **rating**

```
private double rating
```

##### • **description**

```
private java.lang.String description
```

##### • **storeName**

```
private java.lang.String storeName
```

#### • **Constructor Detail**

##### • **Review**

```
public Review(java.lang.String id, java.lang.String storeId, java.lang.String username, MyDate date, double rating, java.lang.String description, java.lang.String storeName)
```

##### • **Review**

```
public Review()
```

#### • **Method Detail**

##### • **getId**

```
public java.lang.String getId()
```



- **setId**

```
public void setId(java.lang.String id)
```

- **getStoreId**

```
public java.lang.String getStoreId()
```

- **setStoreId**

```
public void setStoreId(java.lang.String storeId)
```

- **getUsername**

```
public java.lang.String getUsername()
```

- **setUsername**

```
public void setUsername(java.lang.String username)
```

- **getDate**

```
public MyDate getDate()
```

- **setDate**

```
public void setDate(MyDate date)
```

- **getRating**

```
public double getRating()
```

- **setRating**

```
public void setRating(double rating)
```

- **getDescription**

```
public java.lang.String getDescription()
```

- **setDescription**

```
public void setDescription(java.lang.String description)
```

- **getStoreName**

```
public java.lang.String getStoreName()
```

- **setStoreName**

```
public void setStoreName(java.lang.String storeName)
```

## Class Store

- **Java.lang.Object**  
**com.example.ezway.BaseClasses.Store**

- **All Implemented Interfaces:**  
`java.io.Serializable`

```
public class Store
extends java.lang.Object
implements java.io.Serializable
```

### • **Field Summary**

#### Fields

| Modifier and Type             | Field and Description     |
|-------------------------------|---------------------------|
| private java.lang.String      | <code>closingHours</code> |
| private java.lang.String      | <code>description</code>  |
| private java.lang.String      | <code>id</code>           |
| private java.lang.String      | <code>image</code>        |
| private java.lang.String      | <code>name</code>         |
| private java.lang.String      | <code>openingHours</code> |
| private java.lang.String      | <code>phone</code>        |
| private <code>Position</code> | <code>position</code>     |
| private int                   | <code>rateCount</code>    |
| private double                | <code>rating</code>       |

### • **Constructor Summary**

#### Constructors

##### Constructor and Description

`Store()`

`Store`(java.lang.String id, java.lang.String name, java.lang.String phone, java.lang.String openingHours, java.lang.String closingHours, double rating, int rateCount, java.lang.String image, java.lang.String description, `Position` position)

### • **Method Summary**

#### All MethodsInstance MethodsConcrete Methods

| Modifier and Type | Method and Description         |
|-------------------|--------------------------------|
| java.lang.String  | <code>getClosingHours()</code> |
| java.lang.String  | <code>getDescription()</code>  |
| java.lang.String  | <code>getId()</code>           |

|                  |                                                        |
|------------------|--------------------------------------------------------|
| java.lang.String | <code>getImage()</code>                                |
| java.lang.String | <code>getName()</code>                                 |
| java.lang.String | <code>getOpeningHours()</code>                         |
| java.lang.String | <code>getPhone()</code>                                |
| <b>Position</b>  | <code>getPosition()</code>                             |
| int              | <code>getRateCount()</code>                            |
| double           | <code>getRating()</code>                               |
| void             | <code>setClosingHours(java.lang.String closingH</code> |
| void             | <code>setDescription(java.lang.String descripti</code> |
| void             | <code>setImage(java.lang.String img)</code>            |
| void             | <code>setName(java.lang.String name)</code>            |
| void             | <code>setOpeningHours(java.lang.String openingH</code> |
| void             | <code>setPhone(java.lang.String phone)</code>          |
| void             | <code>setPosition(Position position)</code>            |
| void             | <code>setRateCount(int rateCount)</code>               |
| void             | <code>setRating(double rating)</code>                  |
| java.lang.String | <code>toString()</code>                                |

#### • **Field Detail**

##### • **id**

```
private java.lang.String id
```

##### • **name**

```
private java.lang.String name
```

##### • **phone**

```
private java.lang.String phone
```

##### • **openingHours**

```
private java.lang.String openingHours
```

##### • **closingHours**

```
private java.lang.String closingHours
```

##### • **rating**

```
private double rating
```

##### • **rateCount**

```
private int rateCount
```

##### • **image**

```
private java.lang.String image
```

- **description**

```
private java.lang.String description
```

- **position**

```
private Position position
```

- **Constructor Detail**

- **Store**

- ```
public Store(java.lang.String id, java.lang.String name, java.lang.String phone, java.lang.String openingHours, java.lang.String closingHours, double rating, int rateCount, java.lang.String image, java.lang.String description, Position position)
```

- **Store**

```
public Store()
```

- **Method Detail**

- **getId**

```
public java.lang.String getId()
```

- **getName**

```
public java.lang.String getName()
```

- **setName**

```
public void setName(java.lang.String name)
```

- **getPhone**

```
public java.lang.String getPhone()
```

- **setPhone**

```
public void setPhone(java.lang.String phone)
```

- **getOpeningHours**

```
public java.lang.String getOpeningHours()
```

- **setOpeningHours**

```
public void setOpeningHours(java.lang.String openingHours)
```

- **getClosingHours**

```
public java.lang.String getClosingHours()
```

- **setClosingHours**

```
public void setClosingHours(java.lang.String closingHours)
```

- **getRating**

```
public double getRating()
```

- **setRating**

```
public void setRating(double rating)
```

- **getRateCount**

```
public int getRateCount()
```

- **setRateCount**

```
public void setRateCount(int rateCount)
```

- **getImage**

```
public java.lang.String getImage()
```

- **setImage**

```
public void setImage(java.lang.String img)
```

- **getDescription**

```
public java.lang.String getDescription()
```

- **setDescription**

```
public void setDescription(java.lang.String description)
```

- **getPosition**

```
public Position getPosition()
```

- **setPosition**

```
public void setPosition(Position position)
```

- **toString**

- @NonNull  
public java.lang.String toString()

**Overrides:**  
toString in class java.lang.Object

## Class User

- **Java.lang.Object**  
**com.example.ezway.BaseClasses.User**

```
public class User
extends java.lang.Object
```

### • Field Summary

#### Fields

Modifier and Type	Field and Description
private <b>MyDate</b>	<b>dateOfBirth</b>
private java.lang.String	<b>email</b>
private java.lang.String	<b>gender</b>
private java.lang.String	<b>id</b>
private java.lang.String	<b>username</b>

### • Constructor Summary

#### Constructors

##### Constructor and Description

**User** ()

**User**(java.lang.String id, **MyDate** dateOfBirth, java.lang.String gender, java.lang.String username, java.lang.String email)

### • Method Summary

#### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
<b>MyDate</b>	<b>getDateOfBirth</b> ()
java.lang.String	<b>getEmail</b> ()
java.lang.String	<b>getGender</b> ()
java.lang.String	<b b="" getid<=""> ()</b>
java.lang.String	<b b="" getusername<=""> ()</b>
void	<b b="" setdateofbirth<=""> (<b>MyDate</b> dateOfBirth)</b>
void	<b b="" setemail<=""> (java.lang.String email)</b>
void	<b b="" setgender<=""> (java.lang.String gender)</b>
void	<b b="" setid<=""> (java.lang.String id)</b>
void	<b b="" setusername<=""> (java.lang.String username)</b>

- **Field Detail**

- **id**

```
private java.lang.String id
```

- **dateOfBirth**

```
private MyDate dateOfBirth
```

- **gender**

```
private java.lang.String gender
```

- **username**

```
private java.lang.String username
```

- **email**

```
private java.lang.String email
```

- **Constructor Detail**

- **User**

```
public User(java.lang.String id, MyDate dateOfBirth,
java.lang.String gender, java.lang.String username,
java.lang.String email)
```

- **User**

```
public User()
```

- **Method Detail**

- **getId**

```
public java.lang.String getId()
```

- **setId**

```
public void setId(java.lang.String id)
```

- **getDateOfBirth**

```
public MyDate getDateOfBirth()
```

- **setDateOfBirth**

```
public void setDateOfBirth(MyDate dateOfBirth)
```

- **getGender**

```
public java.lang.String getGender()
```

- **setGender**

```
public void setGender(java.lang.String gender)
```

- **getUsername**

```
public java.lang.String getUsername()
```

- **setUsername**

```
public void setUsername(java.lang.String username)
```

- **getEmail**

```
public java.lang.String getEmail()
```

- **setEmail**

```
public void setEmail(java.lang.String email)
```



## Dialogs

### Class ConfirmPasswordDialog

- `Java.lang.Object`  
`android.app.Dialog`  
`com.example.ezway.Dialogs.ConfirmPasswordDialog`

```
public class ConfirmPasswordDialog
extends android.app.Dialog
implements android.view.View.OnClickListener
```

#### • *Field Summary*

##### Fields

Modifier and Type	Field and Description
<code>private android.app.Activity</code>	<code>activity</code>
<code>private android.app.Dialog</code>	<code>dialog</code>
<code>private android.widget.LinearLayout</code>	<code>layout</code>
<code>private android.widget.LinearLayout</code>	<code>parentLayout</code>
<code>private boolean</code>	<code>updateEmail</code>
<code>private boolean</code>	<code>updatePassword</code>

#### • *Constructor Summary*

##### Constructors

##### Constructor and Description

**ConfirmPasswordDialog**(`android.content.Context context`, `android.app.Activity activity`, `boolean updateEmail`, `boolean updatePassword`, `boolean themePref`)  
 Constructor

#### • *Method Summary*

##### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
<code>private void</code>	<b>confirm</b> ( <code>java.lang.String password</code> , <code>boolean updateEmail</code> , <code>boolean updatePassword</code> ) Gets the password passed in by the user and checks if the password is the user's password and if it is, it updates the user's email and password to the new ones.
<code>private void</code>	<b>handleUpdateEmail</b> ( <code>java.lang.String email</code> ) Updates the user's email.

private void	<b>handleUpdatePassword</b> (java.lang.String password) Updates the user's password.
void	<b>onClick</b> (android.view.View view)

## • **Field Detail**

### • **layout**

```
private android.widget.LinearLayout layout
```

### • **parentLayout**

```
private android.widget.LinearLayout parentLayout
```

### • **dialog**

```
private android.app.Dialog dialog
```

### • **activity**

```
private android.app.Activity activity
```

### • **updateEmail**

```
private boolean updateEmail
```

### • **updatePassword**

```
private boolean updatePassword
```

## • **Constructor Detail**

### • **ConfirmPasswordDialog**

- `public ConfirmPasswordDialog(@NonNull android.content.Context context, android.app.Activity sactivity, boolean updateEmail, boolean updatePassword, boolean themePref)`

Constructor

**Parameters:**  
context

activity

updateEmail - A Boolean variable to know if the email was updated.

updatePassword - A Boolean variable to know if the password was updated.

themePref - Theme preferred by the user.

## • **Method Detail**

### • **onClick**

```
public void onClick(android.view.View view)
```

Specified by:

onClick in interface android.view.View.OnClickListener

- **confirm**

- ```
private void confirm(java.lang.String password,
boolean updateEmail, boolean updatePassword)
```

Gets the password passed in by the user and checks if the password is the user's password and if it is, it updates the user's email and password to the new ones.

**Parameters:**

password - Password for confirmation.

updateEmail - A Boolean variable to know if the email was updated.

updatePassword - A Boolean variable to know if the password was updated.

- **handleUpdatePassword**

```
private void handleUpdatePassword(java.lang.String
password)
```

Updates the user's password.

**Parameters:**

password - New password passed in by the user.

- **handleUpdateEmail**

```
private void handleUpdateEmail(java.lang.String email)
```

Updates the user's email.

**Parameters:**

email - New email passed in by the user.

## Class DiscountDialog

- `Java.lang.Object`

`android.app.Dialog`

`com.example.ezway.Dialogs.DiscountDialog`

```
public class DiscountDialog
extends android.app.Dialog
implements android.view.View.OnClickListener
```

### • Field Summary

#### Fields

| Modifier and Type                         | Field and Description |
|-------------------------------------------|-----------------------|
| private <code>android.app.Activity</code> | <code>activity</code> |
| private <code>Discount</code>             | <code>discount</code> |

### • Constructor Summary

#### Constructors

##### Constructor and Description

`DiscountDialog`(`android.content.Context` context, `boolean` themePref, `Discount` discount)  
Constructor

### • Method Summary

#### All MethodsInstance MethodsConcrete Methods

| Modifier and Type | Method and Description                                      |
|-------------------|-------------------------------------------------------------|
| <code>void</code> | <code>onClick</code> ( <code>android.view.View</code> view) |

### • Field Detail

#### • activity

`private android.app.Activity activity`

#### • discount

`private Discount discount`

### • Constructor Detail

#### • DiscountDialog

- `public DiscountDialog(@NonNull android.content.Context context, boolean themePref, Discount discount)`

### Constructor

#### Parameters:

context

themePref - Theme preferred by the user.

Discount - Discount Object chosen to be viewed by the user.

- **Method Detail**

- **onClick**

```
public void onClick(android.view.View view)
```

#### Specified by:

onClick in interface android.view.View.OnClickListener

## Class MyDatePickerDialog

- `Java.lang.Object`
  - `android.app.Fragment`
    - `android.app.DialogFragment`
      - `com.example.ezway.Dialogs.MyDatePickerDialog`

```
public class MyDatePickerDialog
extends android.app.DialogFragment
implements android.app.DatePickerDialog.OnDateSetListener
```

### • *Field Summary*

#### Fields

| Modifier and Type            | Field and Description |
|------------------------------|-----------------------|
| private android.app.Activity | <code>activity</code> |
| private java.util.Date       | <code>today</code>    |

### • *Constructor Summary*

#### Constructors

| Constructor and Description       |
|-----------------------------------|
| <code>MyDatePickerDialog()</code> |

### • *Method Summary*

#### All Methods Instance Methods Concrete Methods

| Modifier and Type  | Method and Description                                                                                                |
|--------------------|-----------------------------------------------------------------------------------------------------------------------|
| android.app.Dialog | <code>onCreateDialog</code> (android.os.Bundle savedInstanceState)<br>Creates a date-pick dialog.                     |
| void               | <code>onDateSet</code> (android.widget.DatePicker view, int year, int month, int day)<br>Updates the activity's date. |
| void               | <code>setActivity</code> (android.app.Activity activity)                                                              |

### • *Field Detail*

#### • `today`

```
private java.util.Date today
```

#### • `activity`

```
private android.app.Activity activity
```

- **Constructor Detail**

- **MyDatePickerDialog**

```
public MyDatePickerDialog()
```

- **Method Detail**

- **onCreateDialog**

```
public android.app.Dialog onCreateDialog(android.os.Bundle savedInstanceState)
```

Creates a date-pick dialog.

**Overrides:**

onCreateDialog in class android.app.DialogFragment

**Parameters:**

savedInstanceState

**Returns:**

- **setActivity**

```
public void setActivity(android.app.Activity activity)
```

- **onDateSet**

- ```
public void onDateSet(android.widget.DatePicker view, int year, month, int day)
```

Updates the activity's date.

**Specified by:**

onDateSet in interface android.app.DatePickerDialog.OnDateSetListener

**Parameters:**

view

year

month

day

## Class ReviewDialog

- **Java.lang.Object**
  - android.app.Dialog**
    - com.example.ezway.Dialog.ReviewDialog**

```
public class ReviewDialog
extends android.app.Dialog
implements android.view.View.OnClickListener
```

### • **Field Summary**

#### **Fields**

Modifier and Type	Field and Description
private android.app.Activity	<b>activity</b>
private com.google.firebase.database.DatabaseReference	<b>databaseRef</b>
private <b>MyDate</b>	<b>date</b>
private java.lang.String	<b>myUser</b>
private android.widget.RatingBar	<b>ratingBar</b>
private <b>Store</b>	<b>store</b>
private com.google.firebase.auth.FirebaseUser	<b>user</b>

### • **Constructor Summary**

#### **Constructors**

##### **Constructor and Description**

**ReviewDialog**(android.content.Context context, boolean themePref, **Store** store)  
Constructor

### • **Method Summary**

#### **All Methods** **Instance Methods** **Concrete Methods**

Modifier and Type	Method and Description
private void	<b>changeStoreStats</b> (java.lang.String review Text) Checks if the user has already reviewed the shop and if he did, it removes the old review and updates to a new review.
private void	<b>initialize</b> () Sets up the review's initial form.
void	<b>onClick</b> (android.view.View view)
private void	<b>removeReview</b> ( <b>Review</b> review) Removes the review from the database.



private void	<b>sendToFirebase</b> (java.lang.String reviewText) Uploads the review to the database.
private void	<b>validate</b> (java.lang.String reviewText) Validates the review by the user and makes sure It is under 80 characters in length.

## • **Field Detail**

### • **databaseRef**

```
private com.google.firebase.database.DatabaseReference
databaseRef
```

### • **user**

```
private com.google.firebase.auth.FirebaseUser user
```

### • **myUser**

```
private java.lang.String myUser
```

### • **store**

```
private Store store
```

### • **activity**

```
private android.app.Activity activity
```

### • **date**

```
private MyDate date
```

### • **ratingBar**

```
private android.widget.RatingBar ratingBar
```

## • **Constructor Detail**

### • **ReviewDialog**

```
public ReviewDialog(@NonNull android.content.Context
context, themePref, Store store)
```

Constructor

**Parameters:**  
context

themePref - Theme preferred by the user.

store - Object of the store the user is navigating to.

## • **Method Detail**

### • **initialize**

```
private void initialize()
```

Sets up the review's initial form.

- **onClick**

```
public void onClick(android.view.View view)
```

**Specified by:**

onClick in interface android.view.View.OnClickListener

- **validate**

```
private void validate(java.lang.String reviewText)
```

Validates the review by the user and makes sure it is under 80 characters in length.

**Parameters:**

reviewText - Text of the review written by the user.

- **changeStoreStats**

```
private void changeStoreStats(java.lang.String reviewText)
```

Checks if the user has already reviewed the shop and if he did, it removes the old review and updates to a new review. If the user has never reviewed the shop, a new review will be uploaded to the database.

**Parameters:**

reviewText - Text of the review written by the user.

- **removeReview**

```
private void removeReview(Review review)
```

Removes the review from the database.

**Parameters:**

review - Object of the review written by the user.

- **sendToFireBase**

```
private void sendToFireBase(java.lang.String reviewText)
```

Uploads the review to the database.

**Parameters:**

reviewText - Text of the review written by the user.

## Handlers

### Class BottomSheetHandler

- `Java.lang.Object`
  - `android.app.Dialog`
    - `androidx.appcompat.app.AppCompatActivity`
      - `com.google.android.material.bottomsheet.BottomSheetDialog`
        - `com.example.ezway.Handlers.BottomSheetHandler`

```
public class BottomSheetHandler
extends com.google.android.material.bottomsheet.BottomSheetDialog
implements android.view.View.OnClickListener
```

#### • Field Summary

##### Fields

Modifier and Type	Field and Description
private <code>android.app.Activity</code>	<code>activity</code>
private <code>com.google.android.material.bottomsheet.BottomSheetDialog</code>	<code>bottomSheetDialog</code>
private <code>Position</code>	<code>pos</code>
private <code>RecyclerViewAdapter</code>	<code>recyclerAdapter</code>
private <code>Store</code>	<code>store</code>
boolean	<code>themePref</code>

#### • Constructor Summary

##### Constructors

##### Constructor and Description

`BottomSheetHandler`(`android.content.Context context`, `android.app.Activity activity`, `boolean themePref`)  
Constructor

#### • Method Summary

##### All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
private void	<code>changeLanguage</code> ( <code>android.view.View view</code> , <code>boolean langPref</code> ) Change the language of the dialog according to the user's preferred language.
void	<code>closeDialog</code> ()

void	<b>loadSearchDialog</b> (java.lang.String[] shops) Loads the search dialog with the shops's name.
void	<b>onClick</b> (android.view.View view)
void	<b>openSearchDialog</b> () Opens and shows the search dialog.
void	<b>openShopDialog</b> (boolean langPref) Opens up the shops dialog.

## - Field Detail**

### - themePref**

```
public boolean themePref
```

### - store**

```
private Store store
```

### - pos**

```
private Position pos
```

### - recyclerView**

```
private RecyclerView recyclerView
```

### - activity**

```
private android.app.Activity activity
```

### - bottomSheetDialog**

```
private com.google.android.material.bottomsheet.BottomSheetDialog bottomSheetDialog
```

## - Constructor Detail**

### - BottomSheetHandler**

- public BottomSheetHandler(@NonNull android.content.Context context, android.app.Activity activity, boolean themePref)

Constructor

**Parameters:**  
context

activity - Activity the user is opening the dialog from.

themePref - Theme preferred by the user.

## - Method Detail**

### - openSearchDialog**

```
public void openSearchDialog()
```

Opens and shows the search dialog.

- **closeDialog**

```
public void closeDialog()
```

- **loadSearchDialog**

```
public void loadSearchDialog(java.lang.String[] shops)
```

Loads the search dialog with the shops's name.

**Parameters:**

shops – Array of shops.

- **openShopDialog**

```
public void openShopDialog(boolean langPref)
```

Opens up the shops dialog.

**Parameters:**

langPref – Language preferred by the user.

- **changeLanguage**

- ```
private void changeLanguage(android.view.View view,  
boolean langPref)
```

Change the language of the dialog according to the user's preferred language.

**Parameters:**

view

langPref – Language preferred by the user.

- **onClick**

```
public void onClick(android.view.View view)
```

**Specified by:**

onClick in interface android.view.View.OnClickListener

## Class BroadcastReceiverHandler

- `Java.lang.Object`

`android.content.BroadcastReceiver`

`com.example.ezway.Handlers.BroadcastReceiverHandler`

```
public class BroadcastReceiverHandler
extends android.content.BroadcastReceiver
```

### • Constructor Summary

#### Constructors

##### Constructor and Description

`BroadcastReceiverHandler ()`

### • Method Summary

#### All Methods Instance Methods Concrete Methods

##### Modifier and Type

`void`

##### Method and Description

`onReceive` (`android.content.Context context`, `android.content.Intent intent`)  
Checks if the phone's battery is lower than 30, and if it is the app shows a notification recommending the user to not use the app due to its battery consuming percentage.

### • Constructor Detail

#### • BroadcastReceiverHandler

```
public BroadcastReceiverHandler()
```

### • Method Detail

#### • onReceive

- `public void onReceive(android.content.Context context, android.content.Intent intent)`

Checks if the phone's battery is lower than 30, and if it is the app shows a notification recommending the user to not use the app due to its battery consuming percentage.

##### Specified by:

`onReceive` in class `android.content.BroadcastReceiver`

##### Parameters:

`context`

`intent`

## Class CustomToastHandler

- `Java.lang.Object`

`android.widget.Toast`

`com.example.ezway.Handlers.CustomToastHandler`

```
public class CustomToastHandler
extends android.widget.Toast
```

### • **Field Summary**

- **Fields inherited from class `android.widget.Toast`**

`LENGTH_LONG`, `LENGTH_SHORT`

### • **Constructor Summary**

#### Constructors

##### Constructor and Description

**CustomToastHandler**(`android.content.Context context`, `java.lang.String title`, `java.lang.String body`)

Constructor Build a notification with the title and body passed in.

### • **Method Summary**

#### All MethodsInstance MethodsConcrete Methods

##### Modifier and Type

##### Method and Description

`void`

**show()**

### • **Constructor Detail**

#### • **CustomToastHandler**

- `public CustomToastHandler(android.content.Context context, java.lang.String title, java.lang.String body)`

Constructor - Build a notification with the title and body passed in.

##### Parameters:

`context`

`title` - The message's title.

`body` - The message's body.

### • **Method Detail**

- **show**

```
public void show()

Overrides:
show in class android.widget.Toast
```

## Class MessageHandlerService

- **Java.lang.Object**
  - android.content.Context**
    - android.content.ContextWrapper**
      - android.app.Service**
        - com.google.firebase.iid.zzp**
          - com.google.firebase.messaging.FirebaseMessagingService**
            - com.example.ezway.Handlers.MessageHandlerService**
- **All Implemented Interfaces:**
  - android.content.ComponentCallbacks**,
  - android.content.ComponentCallbacks2**

```
public class MessageHandlerService
extends com.google.firebase.messaging.FirebaseMessagingService
```

### • **Field Summary**

#### **Fields**

| Modifier and Type               | Field and Description |
|---------------------------------|-----------------------|
| static java.lang.String         | <b>NOTE_PREF</b>      |
| static java.lang.String         | <b>PREFS_NAME</b>     |
| private static java.lang.String | <b>TAG</b>            |

### • **Constructor Summary**

#### **Constructors**

##### Constructor and Description

**MessageHandlerService()**

### • **Method Summary**

#### **All Methods** **Instance Methods** **Concrete Methods**

| Modifier and Type | Method and Description                                                                                                                          |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| private void      | <b>displayNotification</b> (java.lang.String body, java.lang.String title)<br>shows the message from the database on screen with a notification |
| void              | <b>onMessageReceived</b> (com.google.firebase                                                                                                   |



```
.messaging.RemoteMessage remoteMessage)
Gets the message from the database.
```

## • **Field Detail**

### • **TAG**

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

### • **PREFS\_NAME**

```
public static final java.lang.String PREFS_NAME
```

### • **NOTF\_PREF**

```
public static final java.lang.String NOTF_PREF
```

## • **Constructor Detail**

### • **MessageHandlerService**

```
public MessageHandlerService()
```

## • **Method Detail**

### • **onMessageReceived**

```
public void onMessageReceived(com.google.firebase.messaging.RemoteMessage remoteMessage)
```

Gets the message from the database.

#### **Overrides:**

onMessageReceived in class com.google.firebase.messaging.FirebaseMessagingService

#### **Parameters:**

remoteMessage – The message received from the database.

### • **displayNotification**

- ```
private void displayNotification(java.lang.String body, java.lang.String title)
```

shows the message from the database on screen with a notification

#### **Parameters:**

body – The message's body.

title – The message's title.

## ביבליוגרפיה

Accent systems, what is a Beacon? –

<https://accent-systems.com/support/faq/what-is-a-beacon/?v=88588bacf0da>

AltBeacon API - <https://github.com/AltBeacon>

Beacon class JavaDoc - <https://altbeacon.github.io/android-beacon-library/javadoc/org/altbeacon/beacon/BeaconManager.html>

David G. young, How Far Can You Go (2020) -

<http://www.davidyoungtech.com/2020/05/15/how-far-can-you-go>

What are region Monitoring and Ranging -

<https://community.estimote.com/hc/en-us/articles/203356607-What-are-region-Monitoring-and-Ranging->

What is the difference between foregroundScanPeriod and

backgroundScanPeriod - <https://bbs.aprbrother.com/t/what-is-difference-between-foregroundscanperiod-and-backgroundscanperiod/138>