מעבדה בעיצוב תבניות בתכנה – פרויקט סוף

חיים ינאי –

שחר קליין -

אור אליהו –

הסבר כללי

אפליקציית Sensors Tracking היא אפליקציה המנתחת את חיישני המכשיר של הפלאפון (Light, Proximity, Pressure, Position, Compass) ע"י שימוש ספריות מ"ה Android, האפליקציה יודעת להתמודד עם חיישנים שלא נמצאים במכשיר ע"י כך שהתוצאה שלהם תצבע באדום ובמסך ה-Log יופיע שגיאה בהתאם.

האפליקציה יודעת לרוץ ברקע כאשר היא יורדת מהתצוגה הראשית ע"י כך שהיא מזהה שהיא לא מופיעה יותר, פותחת service היוצר מחדש ניטור של החיישנים, ומודיע למשתמש כי האפליקציה רצה ברקע, ה-service נסגר כאשר חוזרים לאפליקציה. כדי לצאת האפליקציה קיים כפתור יציאה בצד הימני התחתון, ביציאת התוכנית האפליקציה תסגור את מעקב אחר החיישנים.

הוראות התקנה

יש להעביר את קובץ ה-apk שנמצא בתיקיית Application לפלאפון המריץ מ"ה Android,

(יש גרסאות של Android הדורשות שינוי הגדרות אבטחה כדי להתקין אפליקציה שלא מהחנות הרשמית).

Class Diagram

נמצא בתיקיית Class Diagram, מכיל שרטוט בקובץ ה-pdf וקובץ vpp הנפתח ע"י תוכנת Visual Paradigm.

Design Patterns

בפרויקט השתמשנו ב-design patterns שונים, כעת נפרט על 4 עיקריים:

* Singleton

בתבנית עיצוב זאת השתמשנו ב-2 מקרים שונים

1. AboutFragment – עבור fragment של חלון אודות, מכיוון שאין צורך ליצור אותו שוב ושוב, ביצירה הראשונה שלו יישמר reference בהתאם.
2. Logger – מחלקה זאת מטפלת בשליחת טקסט לחלון ה-log בדף הראשי, מכיוון שיש log אחד לאפליקציה (וברוב המקרים כך אמור להיות), עשינו אותו singleton.

מכיוון שבתבנית עיצוב singleton לא ניתן להעביר בפרמטרים ב-getInstance לאחר בירור באינטרנט ניתן לעקוף זאת ע"י בניית getInstance רגיל והעברת פרמטרים לאתחול בפונקציה שנייה (עבורנו init).

* Composite

מכיוון שהחיישנים ה"רגילים" של המכשיר וחיישנים המיקום מופרדים תחת implements שונים – SensorEventListener ו-LocationListener רצינו לאחד אותם "תחת אותה קורת גג" ולכן יצרנו interface בשם FragmentSensors.

מי שממש את ה-interface הזה הוא SensorEventListenerFragment ו-LocationEventListenerFragment.

במחלקת SensorEventGroup<T extends FragmentSensors> אנו ממשמשים שמירת החיישנים הממשמשים את ה-interface הזה.

* Façade

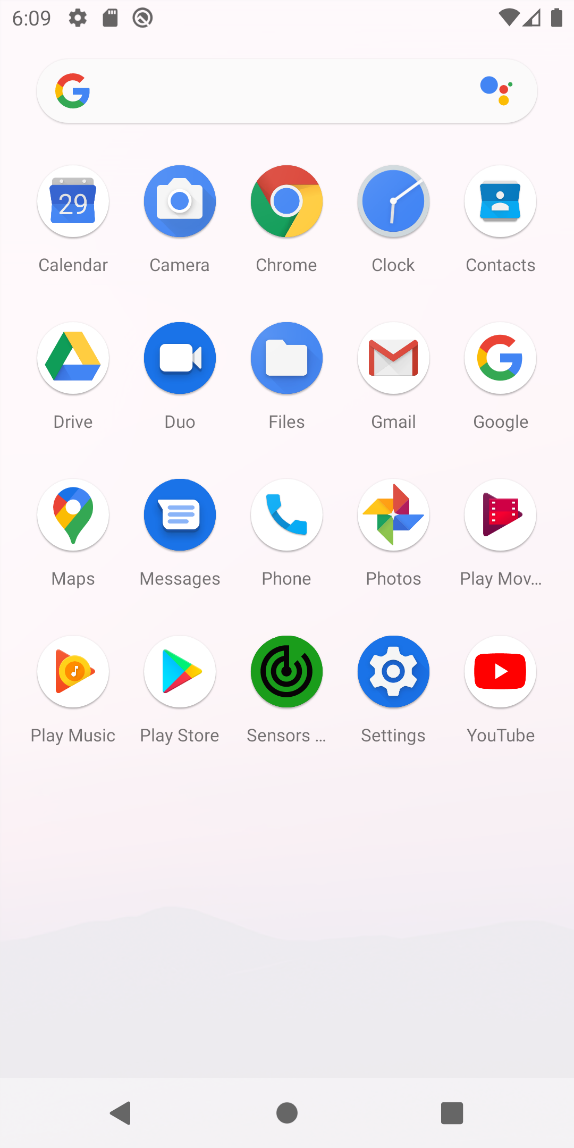
מכיוון שהחיישנים הרגילים וחיישני המיקום הם תחת מנהל חיישנים שונה של מערכת האנדרואיד היינו צריכים לבצע פעולות מורכבות שהן זהות תחת אותה הפונקציה, לכן עבור FragmentSensorsManager השתמשנו בתבנית עיצוב זאת.

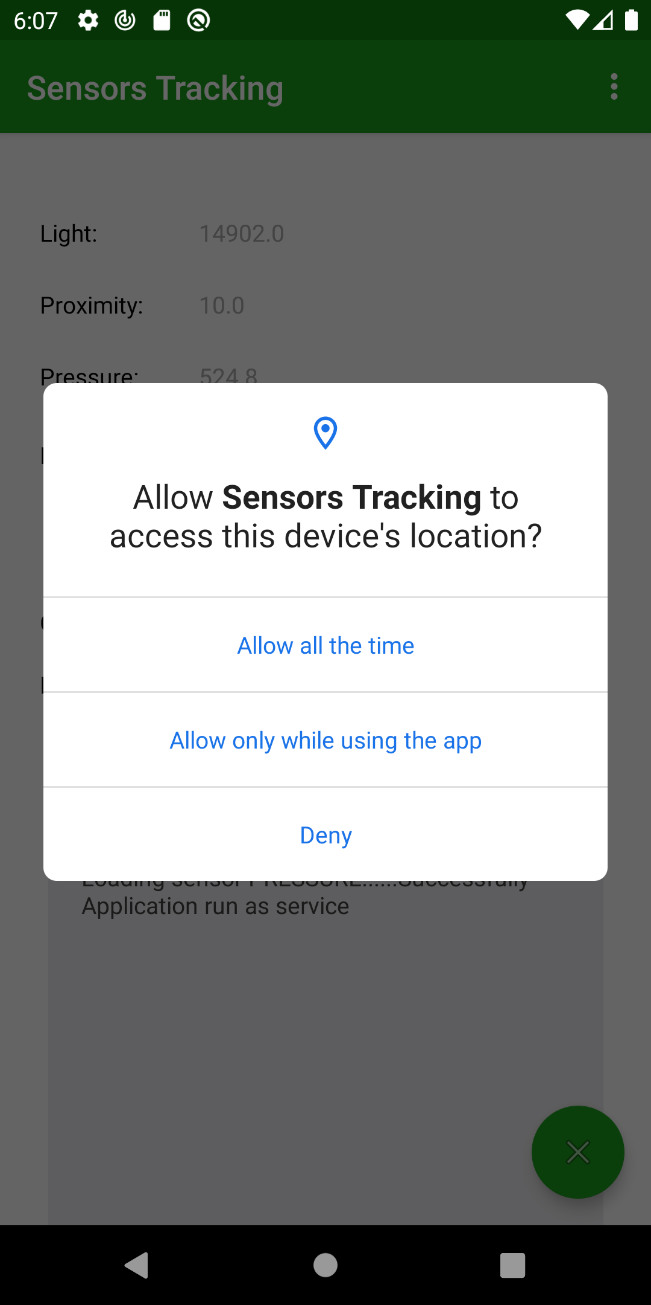
(למרות שע"פ מה שלמדנו אנחנו יצרנו את תתי המחלקות המוכלים, במחלקה זאת קיימים 2 מחלקות של מערכת אנדרואיד ועוד מחלקה שלנו).

* Data Access Object

את רשימת הסנסורים איחדנו תחת מחלקה ג'נרית(עבור משתנים המרחיבים את FragmentSensors) מסוג SensorEventGroup המממש את ה-interface SensorEventDAO.

הוראות משתמש



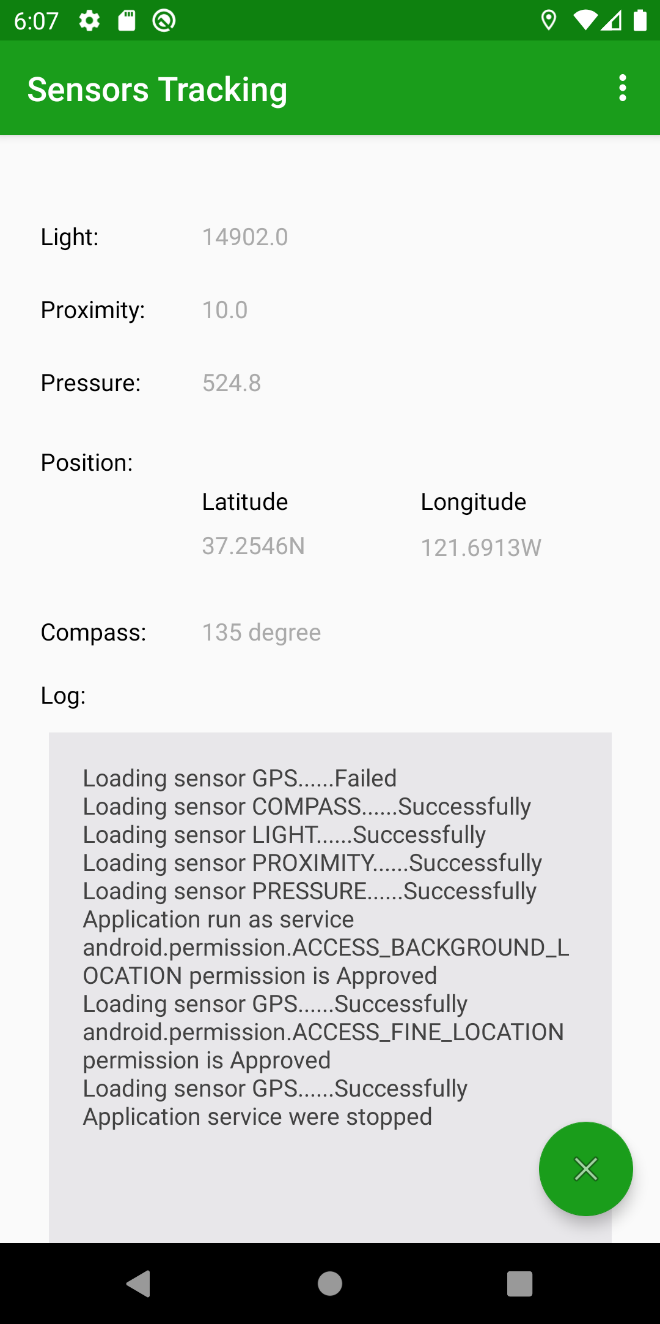
* לאחר התקנת האפליקציה, ניתן לראות במגש האפליקציות את האפליקציה עם הסמל ושמה.
* כאשר נפתח את האפליקציה,

נדרש לתת הרשאות מתאימות עבור קבלת גישה לחיישן המיקום

מכיוון שהתוכנה רצה גם ברקע יש לתת את ההרשאה "Allow all the time", כדי שבמידה והתוכנה תרוץ ברקע תהיה לה את היכולת עדיין לגשת לחיישן המיקום

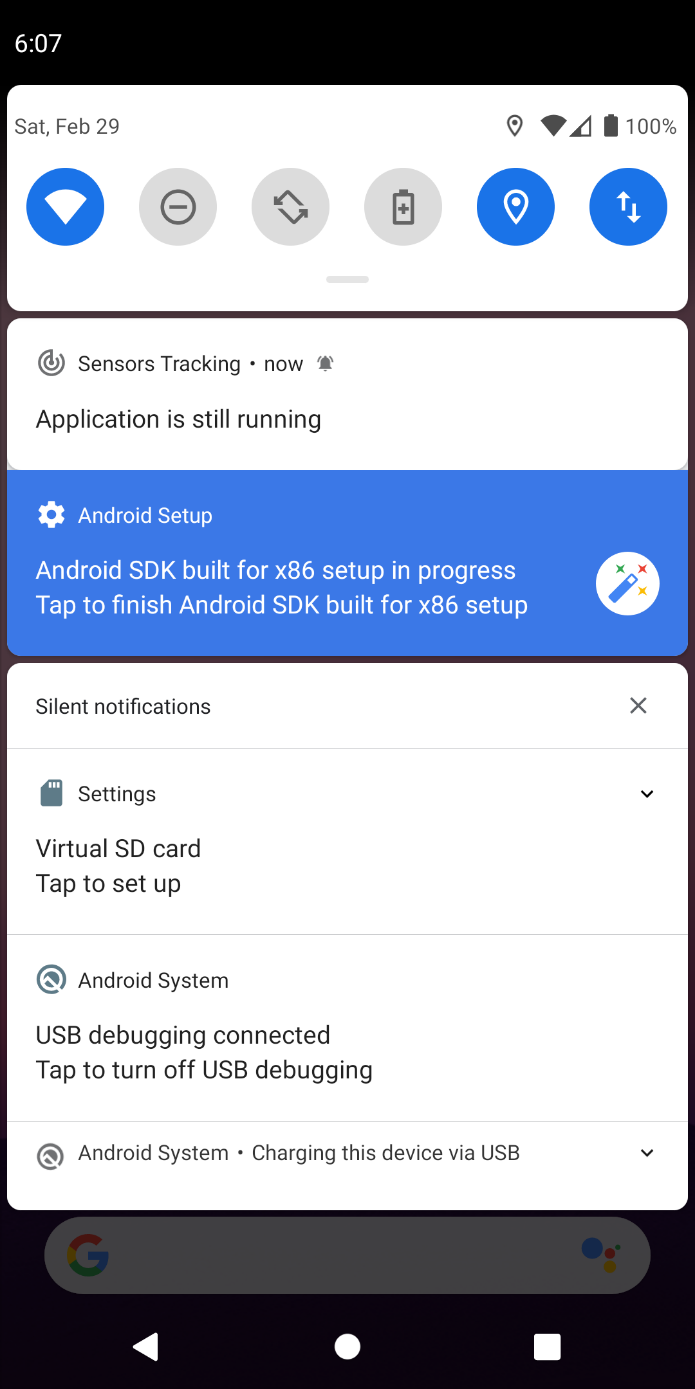
במידה ותיבחר ההרשאה "Allow only while using the app" התוכנה תבצע ניטור המיקום רק כאשר היא תפעל בתצוגה, אך כל פעם שתופעל התוכנה היא תבקש מהמשתמש גישה מחדש (עבור הרשאה קבועה).

* אחרי מתן הרשאה (במקרה שלנו "Allow all the time") ניתן לראות בתחתית המסך ב-log את טעינת החיישנים (בהתחלה נכשל GPS) לאחר קבלת ההרשאה הוא מציג כתוביות בהתאם.



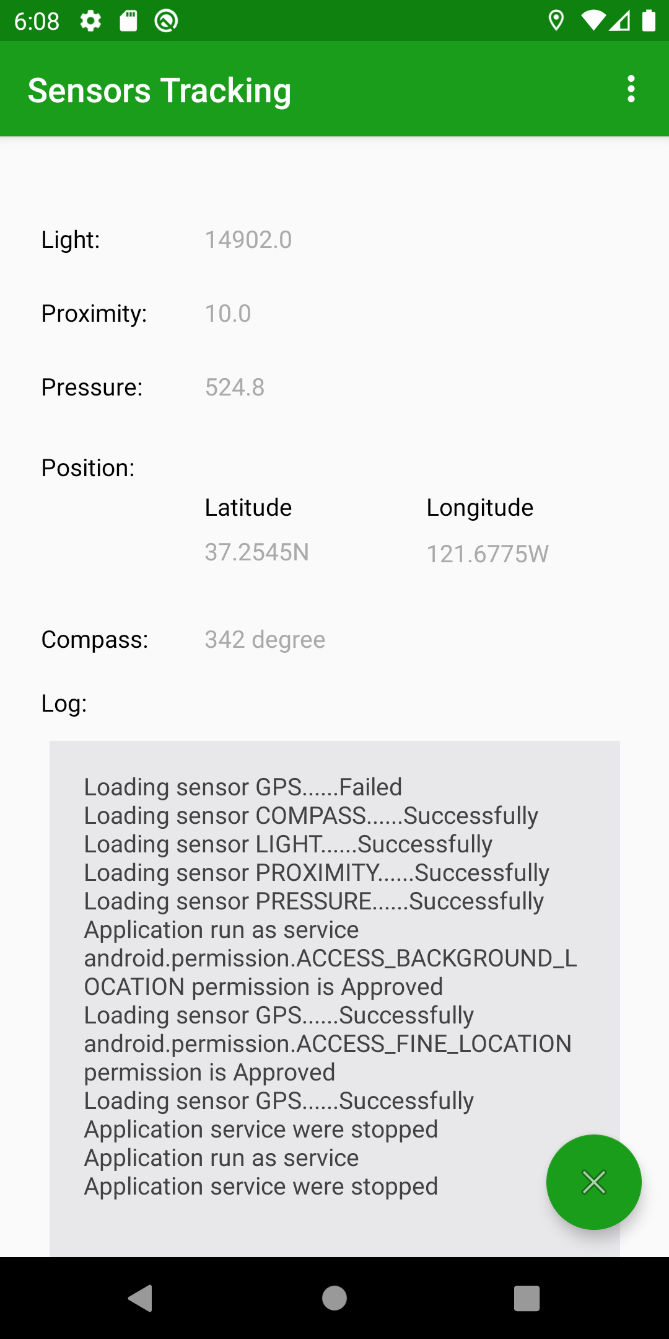
כאשר נקבל את ההרשאה הגבוהה ז"א קיבלנו עבור ריצה ברקע וריצה כאשר התוכנה מופעל לכן נקבל 2 שורות של קבלת גישה עבור ריצה ברקע וריצה כאשר האפליקציה פועלת.

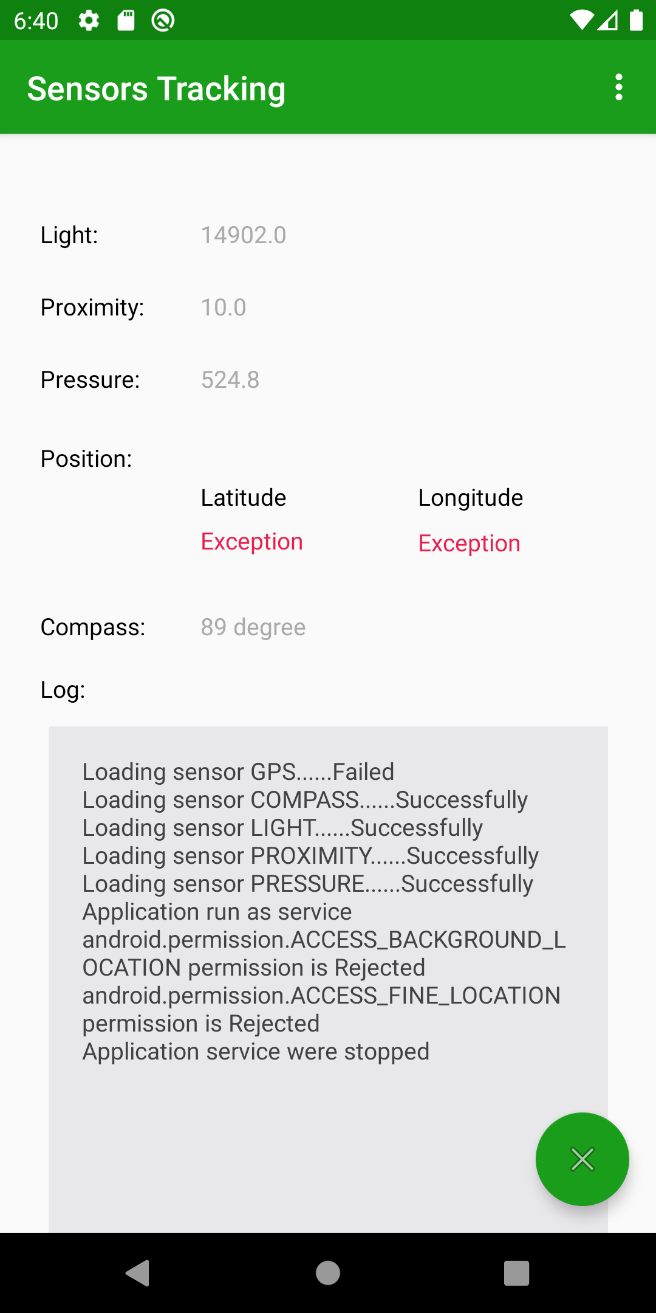
* במידה ונסתיר את האפליקציה (נוריד אתה מהתצוגה) נקבל הודעה מהאפליקציה שתודיע לנו שהאפליקציה עדיין רצה, ההודעה תשמר במגש ההתראות ולא ניתן להסיר אותה עד לפתיחת האפליקציה.



כאשר יופעל service של האפליקציה יירשם ב-log הודעות בהתאם.

(כאשר מופעלת האפליקציה ונפתח חלון של קבלת הרשאה, מ"ה מזהה זאת שהאפליקציה ירדה מהתצוגה ולכן החיישנים לא עובדים, לכן ניתן לראות בהודעות לפני הפעלת service וכיבוי כאשר מקבלים את הגישה, לכן זה לא באג).



* כאשר אחד החיישנים אינו עובד – בתצוגת המידע יהיה Exception באדום ע"פ דרישות האפליקציה.
* לאפליקציה חלון נוסף, ע"י לחיצה על התפריט בצד ימין (3 נקודות) ואז About נקבל את שמות מפתחי האפליקציה.

