**האקתון - קורס יסודות הנדסת תוכנה**

קבוצה 15: זוהר אזולאי, יוסף חיימג'אן, יובל אביטן, גל ארזי.  
[MeisterTask](https://www.meistertask.com/app/project/wR8cpCIA/2020-21-project-19) [GitHub](https://github.com/Zohar-Azulay/CovidProject)

**ארכיטקטורה ודרישות לא־פונקציונליות**

הארכיטקטורה מקשרת בין מסדי הנתונים ושומרת על יחסי גומלין בנתונים ככה שתהיינה הגבלות על שימוש במערכת שתמנענה ביצוע פעולות לא רצויות. למשל:

* מחיקת משתמש ע"י מתנדב
* שליחת הודעת תפוצה ע"י מבקש עזרה

במסגרת ההאקתון חילקנו תפקידים בין חברי הצוות, ככה שכל אחד ואחת יקבלו תחום עליו יהיו אחראיים. החלוקה תרמה בכך שכל חבר צוות היה יכול להתרכז בחלק מסוים ומצומצם של הפרויקט.

זוהר וגל היו אחראיים על יצירת מצגת שתשמש שבלונה ואבטיפוס תצורתי ועיצובי למערכת. דבר זה סייע לנו להרכיב את השדות הרלוונטיים כאשר יצרנו את המערכת ב-Android Studio.

יוסף ויובל למדו איך לבנות את המערכת בתוכנה, למדו והרחיבו את הידע הנדרש בשפת Java.

בהמשך זוהר יצר בסיס נתונים בעזרת SQLite והעלה ל-GitHub, ככה שכל מי שעובד על המערכת יוכל להשתמש במסד הנתונים.

לבסוף, עבדנו כולם במקביל על המערכת.

**בדיקות יחידה – יתרונות וחסרונות**

לגישת פיתוח הקוד בעזרת בדיקות יחידה מספר רב של יתרונות:

* יעילות ואיכות הקוד
* קריאות ופישוט הקוד
* מהירות פיתוח וכתיבה המערכת, ללא זמן רב המושקע בניפוי תקלות (Debugging)
* שימוש חוזר בחלקי קוד
* תפעול ואחזקה עתידיים של המערכת

מנגד, לעתים היתרונות הם גם חסרונות. זמן המושקע בכתיבת בדיקות היחידה יכול לגלוש ולתפוס חלק משמעותי מזמן פיתוח הפרויקט. כאשר דרישות המערכת הן דינאמיות ומשתנות לעתים תכופות, יש צורך בהתאמה מחדש של קוד הבדיקות.  
סיבות אלה, במיוחד במימד הזמן, עלולות לגרום לעלויות גבוהות תוך כדי הפיתוח וחריגה מהתקציב ששוער בשלבים הראשונים (מסמך הייזום)

בוסס על המאמר:

**Evaluating the Effectiveness of Test-Driven Development: Advantages and Pitfalls**

מאת:

Zeba Khanam  
Mohammed Najeeb Ahsan

(Table 1, עמוד 4)

**שינויים בדרישות הפרויקט**

מתנדב – הגבלת תנועה בעקבות סגר.

במקרה של החלטת ממשלה להטיל סגר והגבלת תנועה במרחב הציבורי, המערכת תצטרך למקד את חיפוש המתנדבים לאזור בו יהיה להם את האפשרות להגיע למקומות בהם דרושה התנדבות. מצב זה יכול להוביל לזה שלא יימצאו מתנדבים למשל בשכונות בהן האוכלוסייה מבוגרת.

מבחינת הקוד, כשירצה להתנדב, המתנדב יצטרך לקבל את הבקשות הנמצאות באזורו.  
זה ידרוש מהמערכת לדעת, בעזרת מפות, לצמצם את הדרישות לרדיוס מסוים ממיקום המתנדב.

פתרון אפשרי: כפתור 'שלח חבר'. המתנדב יוכל לשלוח לאדם שהוא מכיר בקשת הצטרפות למערכת, בכדאי שיוכל לטפל בפנייה פתוחה באזור מגוריו.

מבקש עזרה – הגבלת מפגש בקפסולה.

באם יהיה ניתן לפגוש מספר מצומצם של אנשים, לכשיעלה בקשה חדשה למערכת, מבקש העזרה יוכל לקבל סיוע ממספר מועט של אנשים.

פתרון אפשרי: ליצור רשימת אנשי קשר מצומצמת (שכן / בן משפחה קרוב) אשר תקבל התראה כאשר מבקש העזרה מעלה בקשה חדשה למערכת.

בקוד יתווסף מאגר נתונים ייעודי למבקש העזרה.

מנהל אזור – מחסור במתנדבים.

בעקבות התפרצות חוזרת של הקורונה ועלייה במספר בקשות לעזרה, נוצר מחסור במתנדבים. על מנהל האזור לנתב התנדבויות מערים שכנות לצורך טיפול בבקשות פתוחות.

פתרון אפשרי: מנהל האזור יוכל לציור קשר עם מנהלי אזור שכנים, ויחד ינסו למצוא מתנדבים לכלל הבקשות.