

מטלה 2- מסמך דרישות

מגשים:

רפאל קואפיק 337614747

שי קרונפלד 322234782

משה אסקרוב 314085986

חלק א-

- שיחה עם לקוחות פוטנציאליים, נפרט על שתיים מהשיחות:

○ עובד בחברת אבטחה-

■ **אילו פונקציות אתה צריך במערכת?**

- ניהול משמרות באזור האישי כולל לראות מי פנוי להחלפה.
- צ'אט פנימי כדי להימנע מערבוב עם הודעות פרטיות.
- שיחות וידאו כי לפעמים שצריך לדבר עם עובדים בשטח.
- לראות תלושי שכר וסידור משמרות ישירות באפליקציה.

■ **מה חשוב לך בחוויית המשתמש?**

- שהכול יהיה פשוט, בלי מיליון מסכים.

■ **מה ייחשב פתרון 'טוב'?**

- תיעוד מסודר של חופשות ומשימות.
- פחות כאב ראש ופחות פניות בווטסאפ.

○ מנהל (אחמ"ש) במפעל-

■ **מה הקשיים המרכזיים בניהול עובדים?**

- העובדים לא תמיד מעדכנים נוכחות בזמן.
- הגשת משמרות וביורור זמינות למשמרות מתנהל בוואטסאפ וקשה לעקוב.
- צריך לראות מי הגיע, מי מאחר ומי ביקש חופש ואין פלטפורמה שמרכזת הכל במקום אחד.
- מעקב אחרי ביצוע משימות דרך הוואטסאפ בצורה לא מסודרת.

■ **אילו פונקציות אתם צריכים במערכת?**

- דיווח נוכחות עם מיקום.
- ניהול משמרות, לראות את זמינות העובדים בצורה מסודרת.
- צ'אט פנימי ושיחות וידאו.
- ניהול החלפת משמרות בין עובדים עם אישור אחראי בצורה מסודרת באפליקציה.
- אפשרות להגיש משימות לעובדים בצורה מסודרת ולעקוב אם בוצע או לא.

■ **מה חשוב לכם בחוויית המשתמש?**

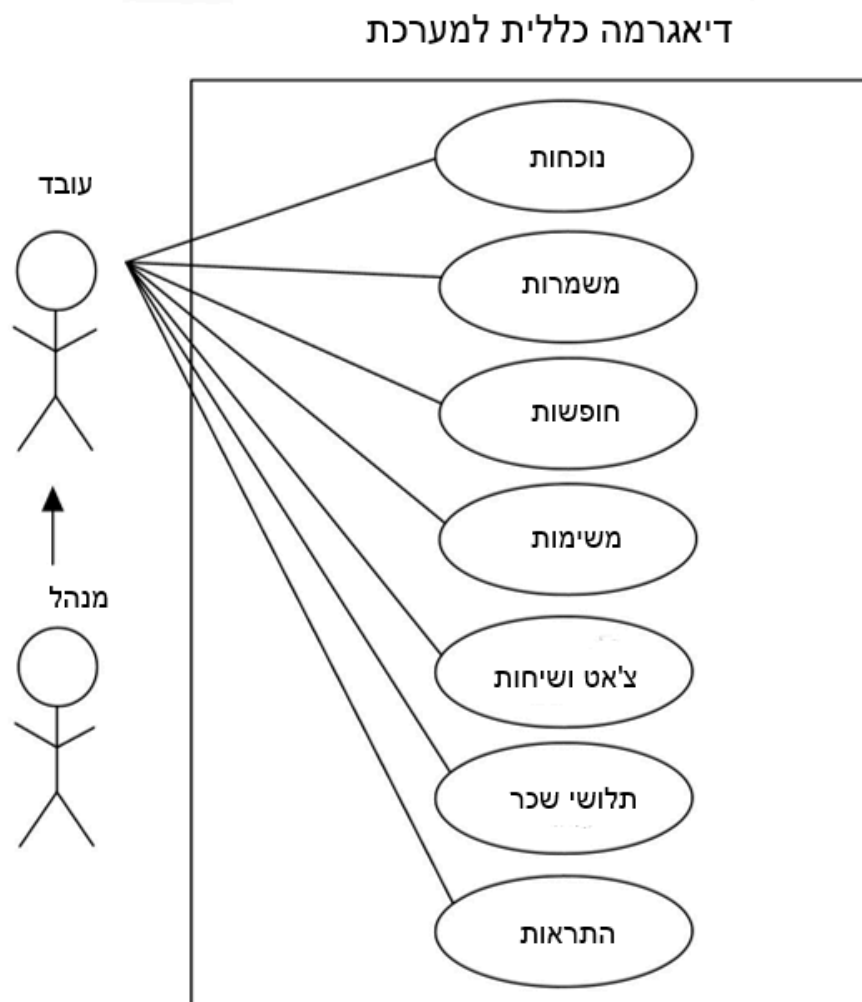
- שיכנסו, ילחצו על כפתור וזהו.
- שיהיו התראות על בקשות וחופשות.

■ **מה ייחשב פתרון 'טוב'?**

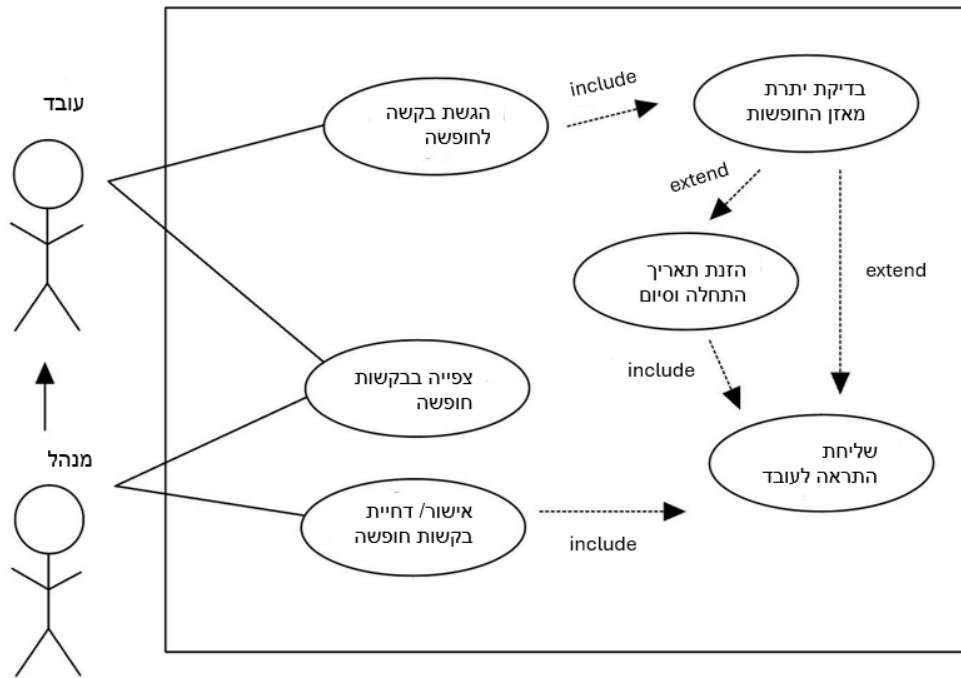
- מערכת שמציגה בזמן אמת את מצב העובדים.

- תיעוד מסודר של זמינות העובדים.
- שכל השיח עם העובדים יהיה מרוכז במקום אחד ולא יהיה מבולגן בווטסאפ.

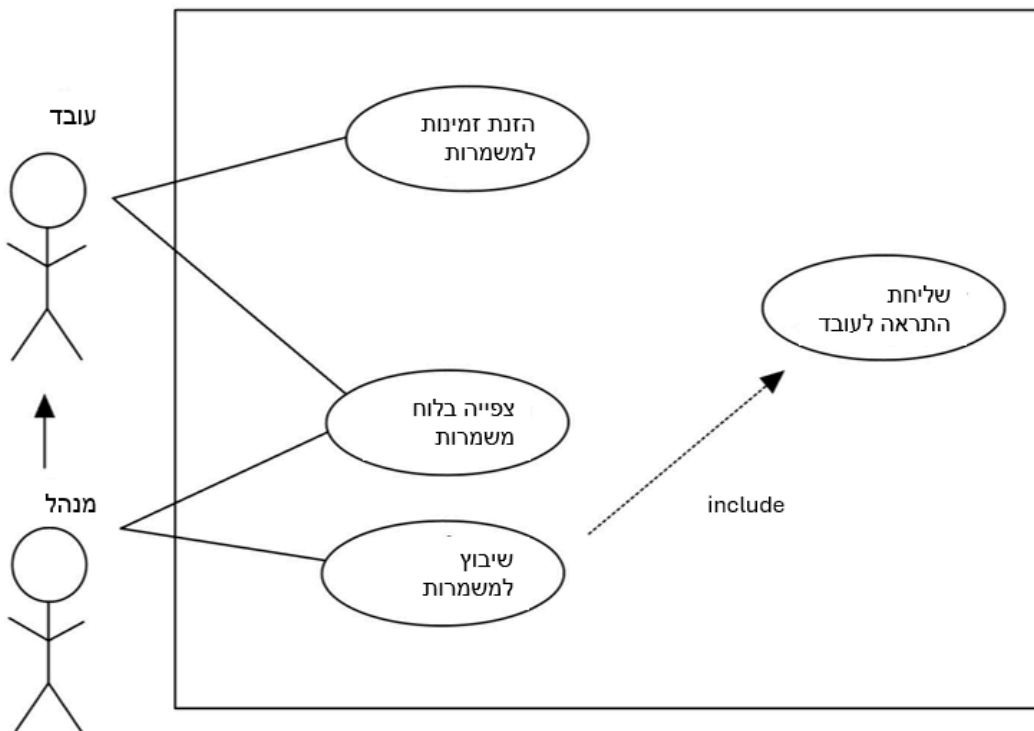
2.



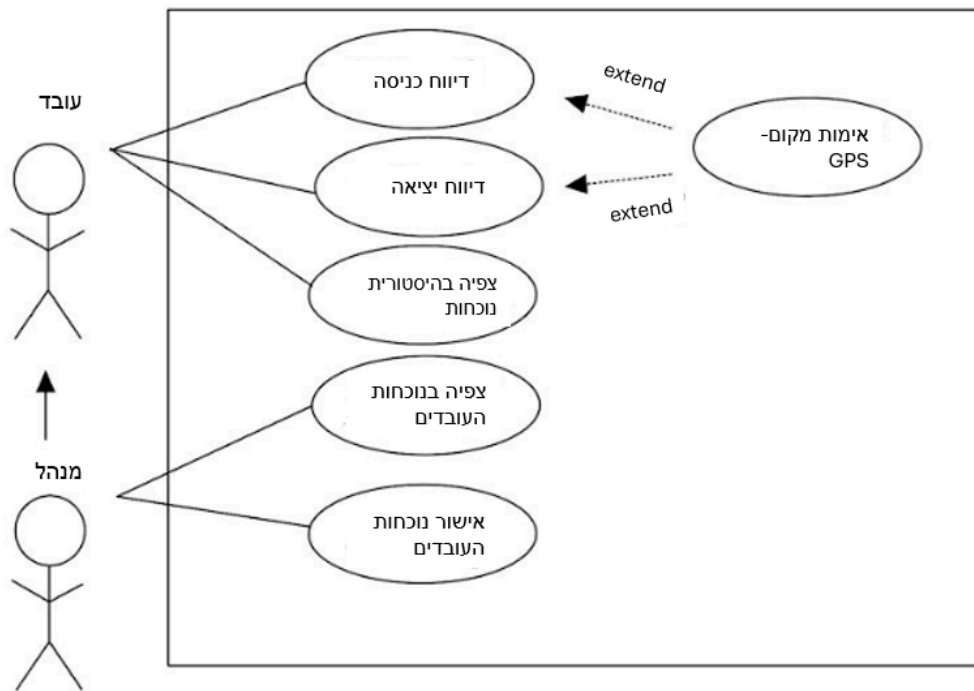
Use case : ניהול חופשות



Use case : ניהול משמרות



Use case : ניהול נוכחות



כאן הוצגו 3 דיאגרמות של 3 use cases מרכזיים. בסעיף 6 הוצגו גם הדיאגרמות של שאר התהליכים.

3. מקרה 1 ניהול בקשת חופשה :

- מטרה: העובד רוצה להגיש בקשת חופשה למנהל באופן מסודר מתוך האפליקציה.
- שחקנים מעורבים: עובד ומנהל
- תנאים מקדימים:
 - העובד מחובר למערכת.
 - פרטי העובד קיימים במסד נתונים.
 - לכל עובד מוגדרים יתרת ימי חופשה נוכחית ומספר ימי המינוס המקסימלי המותר.
- שלבי התהליך:
 1. העובד נכנס למסך "בקשות חופשה".
 2. העובד לוחץ על "הגש בקשה חדשה".
 3. המערכת מציגה טופס להזנת תאריכי התחלה וסיום החופשה וכן סיבת בקשת החופשה.
 4. העובד מזין את הפרטים ולוחץ על "שלח בקשה".
 5. המערכת מחשבת את מספר ימי החופשה המבוקשים ומשווה ליתרת הימים של העובד. אם לאחר החישוב היתרה גדולה או שווה למינימום המותר אז הבקשה יכולה להישלח.
 6. המערכת שומרת את הבקשה במסד הנתונים.
 7. המערכת שולחת התראה למנהל.
 8. המנהל רואה את הבקשה במסך הניהול.

● **תוצאה מוצלחת:**

- בקשת החופשה נשמרה במערכת בסטטוס של "ממתינה לאישור מנהל".
- אם העובד נכנס למינוס מותר, הדבר מתועד בנתוני החופשה שלו.
- המנהל רואה את הבקשה במסך הניהול, והעובד רואה אותה ברשימת הבקשות.

● **תוצאות חלופיות:**

- חסרים פרטים- המערכת מציגה הודעת שגיאה: "אנא מלא את כל השדות הנדרשים".
- אין יתרת חופשה- המערכת מציגה הודעה: "אין יתרת ימי חופשה מספקת".
- שגיאת תקשורת- המערכת מציגה: "הבקשה לא נשמרה, נסה שוב".

חלק ב-

1. מבוא:

- **מטרה:** האפליקציה WorkConnect נועדה לייעל את סביבת העבודה באמצעות שיפור התקשורת הפנים-ארגונית, ניהול משימות ויכולת ניהול נוכחות חכמה. האפליקציה מספקת כלי עבודה נוח לעובדים ולמנהלים, ומאפשרת תפעול יעיל, מהיר ומדויק של תהליכים יום-יומיים. האפליקציה כוללת את הפונקציות המרכזיות הבאות:

- דיווח כניסה/יציאה ממשמרת.
- אימות מיקום המשתמש באמצעות GPS בכניסה לשטח העבודה.
- צפייה בהיסטוריית משמרות של שלושת החודשים האחרונים ושעות עבודה.
- שליחת הודעות ותקשורת פנים-ארגונית בין עובדים ומנהלים.
- מערכת ניהול משימות עם שיוך משימות לעובדים מתאימים.
- התראות בנושאי משמרות, משימות ופעולות חשובות.
- פרופיל אישי לכל עובד הכולל מידע תעסוקתי (שכר שעות, משמרות שבועי, תלושי שכר שהתקבלו).
- הצגת תמונת מצב בזמן אמת למנהלים על עובדים שנמצאים במשמרת.

● **קהל יעד:**

- **חברות- אשר מעוניינות לייעל את מקום העבודה ולספק לעובדים ולמנהלים כלי שיעזור להם ביום יום.**
- **מנהלים- מנהלים בחברה ישתמשו באפליקציה בכדי לייעל את ניהול העובדים שלהם אם זה בכך שאפשר לצפות בנוכחות של כל עובד, להגיש משימות לעובדים והאפשרות ליצור איתם קשר ישיר במידה והם צריכים עזרה בעבודה או מכל סיבה כזו או אחרת.**
- **העובדים- העובדים ישתמשו באפליקציה בכדי לשפר את הנוחות שלהם במקום העבודה. אם זה בכך שהם יכולים להיכנס למשמרת דרך הטלפון, לעקוב אחרי משימות שיש להם לבצע במקום נוח או בזה שיש להם אפשרות לדבר עם עובדים ומעבידים אחרים מהחברה אם זה לכיף או בכדי להתייעץ קצת לגבי משימה שהוא צריך לבצע. בנוסף לעובד יהיה מקום נוח לעקוב אחרי הפרטים שלו כעובד אם זה מידע על השכר השעתי, כמה משמרות הוא כבר עשה החודש או פשוט מקום לשמור את כל התלושי שכר שהעובד קיבל עד היום.**

- **מטרת השימוש:** האפליקציה תשמש את העובדים והמנהלים ביום-יום בצורה קבועה. דוגמאות למקרים:
 - **עובד מתחיל את היום**
 - נכנס לאפליקציה < לוחץ על "כניסה למשמרת" < המערכת מבצעת GPS Verify ושומרת זמן.
 - **עובד מסיים משמרת**
 - לוחץ על "יציאה" < האפליקציה מחשבת את זמן המשמרת ומעדכנת את ההיסטוריה.
 - **למנהל יש משימה שהוא רוצה שעובדים יבצעו**
 - נכנס למסך משימות < בוחר משימה חדשה < מוסיף שם, הסבר ובוחר איזה עובדים שלו הוא רוצה שיבצעו את המשימה < מגדיר חלון זמן למשימה < לוחץ אישור < בלוח משימות מתעדכנת המשימה החדשה וכל עובד רלוונטי מקבל התראה על משימה חדשה לביצוע.
 - **מנהל מחפש משימות שלא בוצעו**
 - נכנס למסך משימות < רואה מי עדיין לא סיים < שולח תזכורת.
 - **מנהל צריך לעדכן את הצוות**
 - כותב הודעה < שולח לכל העובדים < העובדים מקבלים התראה מיידית.
 - **עובד חדש רוצה להבין את המשכורת שלו**
 - נכנס לפרופיל האישי < רואה שכר שעתי, תלושי שכר, משמרות שבוצעו החודש.
- **תחום המוצר:** האפליקציה כוללת:
 - ניהול נוכחות מלא (כניסה/יציאה).
 - אימות מיקום באמצעות GPS.
 - מערכת משימות מלאה: יצירת משימות, הקצאה לעובדים, התראות.
 - מערכת הודעות בין עובדים למנהלים.
 - שיחות וידאו בין עובדים למנהלים.
 - פרופיל אישי מלא לכל עובד.
- **גבולות ופתיחות:**
 - תמיכה רק ב-Android.
 - אין ניהול שכר מלא- רק מידע לעובד.
- **הגדרות וקיצורים:**
 - **GPS** - איתור מיקום.
 - **Application Programming Interface (API)** - מנגנון תקשורת שמאפשר לאפליקציה להתחבר למערכות אחרות ולבצע פעולות דרך ממשק חיצוני.
 - **Firestore database** - בסיס נתונים בענן מבית Firebase המשמש לאחסון מידע בזמן אמת, כולל משמרות, משימות, הודעות ופרופילים.
 - **Firestore Auth** – מערכת התחברות וניהול משתמשים.
 - **GPS API** - מאפשר שימוש ב GPS של המשתמש לאימות מיקום, והצגת מידע מיקום באפליקציה.

2. תיאור כללי:

● 2.1 צרכי משתמשים:

○ צרכי העובדים:

- אפשרות לדווח כניסה ויציאה בצורה פשוטה ומהירה, כולל אימות מיקום בכניסה למשמרת.
- לראות תלושי שכר והיסטוריית משמרות במקום מרוכז ונגיש.
- מערכת ניהול משימות ברורה ונוחה, שבה העובד רואה מה עליו לבצע ומתי.
- צ'אט פנימי ושיחות וידאו כדי לא לערב הודעות פרטיות מהוואטסאפ.
- דרך מסודרת ופשוטה להגיש ולעקוב אחרי בקשות חופשה.
- ממשק משתמש קל, מהיר, ולא עמוס.

○ צרכי המנהלים:

- שליטה מלאה בנוכחות העובדים: מי במשמרת, מי הגיע בזמן, מי מאחר ומי נעדר.
- תצוגה מרוכזת של זמינות עובדים למשמרות, ללא תלות בוואטסאפ מפוזר.
- יכולת ליצור משימות, לשייך אותן לעובדים ולעקוב אחרי ביצוען בפועל.
- שליחת הודעות והתראות לעובד בודד או לכל הצוות.
- מערכת המאפשרת להם לאשר בקשות חופשה ולנהל מעקב מסודר.
- חוויית שימוש עם מינימום צעדים: "שייכנסו, ילחצו על כפתור זהו"(מחלק א)

○ צרכים משותפים לכל המשתמשים:

- סביבת עבודה דיגיטלית מאוחדת במקום אחד, במקום שימוש בכלים מפוזרים.
- עדכונים והתראות בזמן אמת לגבי משימות, הודעות, משמרות ושינויים.
- חוויית שימוש אחידה עבור כל העובדים בחברה.
- אפשרות לתפעול יום-יומי שוטף: ניהול משמרת, תקשורת, משימות ופרופיל אישי.

● 2.2 הנחות ותלויות:

○ הנחות:

- לכל משתמש יש מכשיר Android. האפליקציה תומכת ב־Android בלבד, ולכן כל משתמש חייב להשתמש במכשיר המותאם למערכת זו.
- למשתמשים יש חיבור אינטרנט פעיל מאחר שכל הנתונים (משמרות, משימות, הודעות והתראות) מסונכרנים בזמן אמת דרך Firebase, יש צורך בחיבור אינטרנטי זמין.
- העובדים והמנהלים בעלי הרשאות מתאימות לפי Role ההרשאות מנוהלות ע"י Firebase Auth, ולכן מוגדר מראש מי עובד ומי מנהל, ומערכת ההפעלה מניחה שהרול הנכון כבר הוגדר עבור כל משתמש.

- המערכת מכילה נתונים עדכניים על העובדים לדוגמה: יתרת ימי חופשה, היסטוריית משמרות, פרטי פרופיל, זמינות למשמרות מידע זה נשמר במסד הנתונים ומניח שהוא תקין ומעודכן.
- המשתמשים יודעים לבצע שימוש בסיסי באפליקציות מאחר והממשק דורש פעולות פשוטות (ניהול משמרת, מסך בית, משימות וכו'), המערכת מניחה שהמשתמשים מיומנים בשימוש באפליקציות מובייל.
- תלויות:
 - Firebase Auth ניהול משתמשים, הרשמות, סיסמאות והרשאות גישה \. בלעדיו לא ניתן לגשת למערכת.
 - Firebase, משמש כמאגר המרכזי לכל הנתונים: משמרות, משימות, הודעות, פרופילים והתראות. המערכת תלויה ביכולת לשמור ולסנכרן נתונים בזמן אמת. במקרה של נפילה, תוצג שגיאה ותתבצע ניסיון התחברות חוזר.
 - GPS API של המכשיר נדרש לאימות מיקום בעת כניסה למשמרת. ללא גישה ל-GPS לא ניתן לבצע אימות כניסה תקין.
 - Firebase Cloud Notifications / Push Notifications משמשת לשליחת התראות על משימות, משמרות, הודעות ובקשות חופשה. המערכת תלויה ביכולת לשלוח התראות בזמן אמת.

3. דרישות מערכת:

- דרישות פונקציונליות:
 - ניהול נוכחות-
 - המערכת תאפשר כניסה/יציאה ממשמרת.
 - תשמור את זמן הכניסה, זמן היציאה ותוודא את מיקום העובד.
 - תמנע כניסה נוספת בלי יציאה.
 - ניהול משימות-
 - המנהל יוכל ליצור משימות חדשות.
 - המנהל יוכל לשייך משימה לעובדים.
 - העובד יקבל התראה כאשר יש משימה חדשה.
 - העובד יוכל לסמן משימה כבוצעה.
 - דף שנגיש לעובדים ומנהלים שייציג את כל המשימות שלא בוצעו עדיין במקום אחד.
 - דף המציג את כל המשימות שהוגשו בעבר לא משנה אם בוצעו או לא בסוף (היסטוריית משימות נשמרת לחצי שנה).
 - תקשורת-
 - המנהל יוכל לשלוח הודעות לכל העובדים או לעובד ספציפי.
 - העובדים יקבלו התראות על הודעות חדשות.
 - ההודעות יישמרו במסך ייעודי.
 - אפשרות לשיחה וידאו אם זה בין זוג אנשים או קבוצה.

○ פרופיל אישי-

- כל עובד יוכל לראות: שכר שעתי, משמרות שבוצעו החודש, תלושי שכר ופרטים אישיים על עצמו.
- המנהל יוכל גם לראות נתונים בסיסיים על העובדים שלו.

● דרישות ממשק חיצוני:

○ ממשק משתמש-

- האפליקציה רצה על Android בלבד.
- הממשק יהיה מחולק למספר מסכים עיקריים:
 - מסך ראשי (כניסה/יציאה) בו המשתמש יתחבר עם שם משתמש וסיסמה או יעשה רישום אם זה השימוש הראשון.
 - מסך בית ובו גישה לכל הפעולות של האפליקציה כמו משימות, הודעות בין עובדים, שיחות וידאו, פרופיל אישי, משמרות.
 - לכל אחת מהפעולות יהיה דף שינהל את כל התהליך לדוגמה אחרי שהמשתמש נכנס לדף הבית ובחר משמרות, הוא יעבור לדף חדש עם כל המידע על המשמרות ואפשרות לחזור לדף הבית.

○ אינטגרציה "צד שלישי"-

- Firebase Auth - לשמירה על משתמשים קיימים ורישום משתמשים חדשים.
- Firestore Database - שמירה של מידע כמו משמרות, משימות ועוד.
- GPS API - מאפשר שימוש ביכולות ה-GPS של הטלפון של הלקוח.

○ **אחסון ענן-** האפליקציה משתמשת בשירותי Firebase לצורך אחסון נתונים וסנכרון בזמן אמת. הדרישות הן:

- כל הנתונים (משמרות, משימות, הודעות, פרופילים) יישמרו ב-Firestore Database.
- הנתונים יסונכרנו בזמן אמת בין עובדים ומנהלים.
- Firestore יישמש כמאגר מרכזי יחיד.

● פיצ'רים של המערכת:

- **מידע שימושי למנהלים:** מי במשמרת עכשיו, כמה עובדים מחוברים, משימות פתוחות.
- **מנהל משימות לעובדים:** רשימת משימות, עדכוני מצב, התראות.
- **הודעות ושיחות בין עובדים ומנהלים:** הודעות בין שני אנשים או קבוצה של עובדים/מנהלים עם אפשרות לשיחות וידאו במידת הצורך.
- **מעקב נוכחות:** כל המשמרות של העובד לפי תאריך וכמה זמן עבד. בנוסף עובד לא יכול להיכנס למשמרת אם הוא לא במקום העבודה.
- **דף פרופיל שמציג פרטים על אותו עובד/מנהל:** פרטים על השכר, תלושי שכר, היסטוריית עבודה של שלושת החודשים האחרונים, פרטים אישיים, פעולות שבוצעו

כמו פתיחת משימה או ביצוע משימה.

• דרישות לא פונקציונליות:

○ **ביצועים-**

- כל פעולה באפליקציה תטען תוך עד 2 שניות.
- שמירת נתונים בענן ≥ 5 שניות.

○ **מדרגיות ואפשרויות פיתוח-**

- המערכת תוכל לתמוך במספר רב של חברות ומספר רב של משתמשים בכל חברה (עובדים ומנהלים).
- הנתונים יישמרו ב-Firestore המאפשר התרחבות.

○ **בטיחות-**

- גישה לפיצ'רים נעשית לפי Role, כך שעובד לא יוכל לבצע פעולות של מנהל.
- סיסמאות ומשתמשים מנוהלים רק ע"י Firebase.

○ **זמינות-**

- האפליקציה תהיה זמינה כל עוד Firebase פעיל.
- במקרה של נפילה, תוצג הודעת שגיאה ויתבצע ניסיון חוזר.

4. צרכים ספציפיים של משתמשי הקצה:

• 4.1 תיאור קבוצות משתמש:

○ עובדים(Primary):

- העובדים הם המשתמשים המרכזיים של המערכת, המבצעים את פעולות העבודה היומיומיות דרך האפליקציה.
- תפקידים: דיווח כניסה ויציאה מהמשמרת, קבלת משימות וסימון ביצוע, קבלת הודעות והתראות ממנהלים, צפייה בפרופיל האישי ובהיסטוריית משמרות, הגשת בקשות חופשה ומעקב אחריהן, תקשורת עם המנהלים ועם עובדים אחרים.
- מאפיינים ייחודיים: שימוש יומיומי ממוקד בפעולות אישיות, ממשק פשוט, קל לתפעול ולא עמוס, הרשאות בסיסיות בלבד כלומר ללא יכולת השפעה על עובדים אחרים, הצורך במידע מהיר וברור על המשמרת ועל המשימות.

○ מנהלים(Secondary):

- המנהלים הם משתמשי מערכת מתקדמים המחזיקים בהרשאות נוספות המאפשרות להם לנהל את העובדים ואת סביבת העבודה.
- תפקידים: מעקב אחר נוכחות העובדים בזמן אמת, יצירת משימות ושיון לעובדים, אישור או דחיית בקשות חופשה, שליחת התראות והודעות לצוות ותקשורת איתם, ניהול זמינות עובדים למשמרות.
- מאפיינים ייחודיים: רמת גישה גבוהה יותר למידע מערכת, יכולת לבצע פעולות שמשפיעות על משתמשים אחרים, צורך בתצוגת נתונים מרוכזת וברורה לקבלת החלטות, תפעול מסכים ייעודיים לניהול משמרות ומשימות

• 4.2 דרישות ייחודיות למשתמשים:

○ עובדים:

- הרשאות מיוחדות: גישה למידע אישי בלבד: משמרות, שיחות, משימות, פרופיל, פרטיים, תלויים והתראות ובקשות חופשה. אין לעובד הרשאה לצפות בנתוני עובדים אחרים או במסכי ניהול.
- תצוגות מותאמות: מסך בית המציג את המשמרת הנוכחית, המשימות האישיות וההתראות החדשות. צ'אט פנימי ושיחות וידאו, פרטים אישיים, פרופיל. תצוגה ממוקדת בפעולות האישיות של העובד בלבד.
- פונקציונליות ייחודית: דיווח כניסה/יציאה מהמשמרת, קבלת משימות וסימון ביצוען, קבלת הודעות מהמנהלים, שיחות צ'אט ווידאו עם עובדים אחרים, הגשת בקשות חופשה ומעקב אחר סטטוס, צפייה בתלויי שכר ובהיסטוריית משמרות.

○ מנהלים:

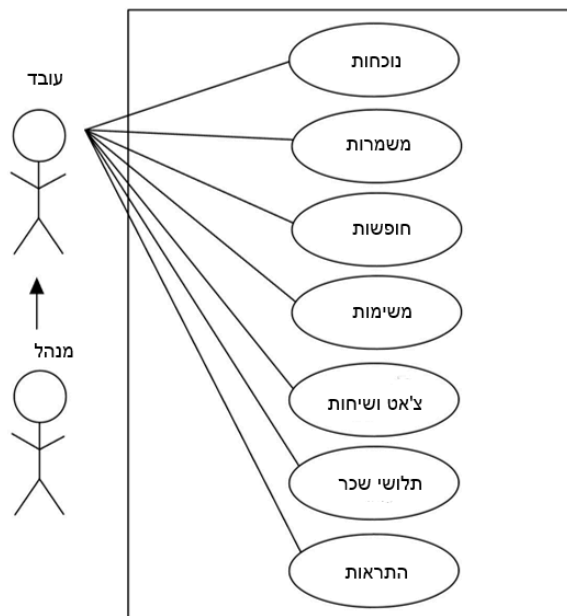
- הרשאות מיוחדות: גישה לנתוני נוכחות, זמינות ומשימות של כלל העובדים, כולל יכולת לבצע פעולות המשפיעות על משתמשים אחרים כמו אישור חופשה או מתן משימה.
- תצוגות מותאמות: Dashboard ניהולי להצגת תמונת מצב בזמן אמת, מסכי ניהול ייעודיים למשמרות, נוכחות העובדים, משימות וחופשות שאינם זמינים לעובדים.
- פונקציונליות ייחודית: יצירת משימות ושיוך לעובדים, אישור או דחיית בקשות חופשה, שליחת התראות והודעות לצוות, ניהול זמינות עובדים ושיבוץ למשמרות, וצפייה בנתוני נוכחות בזמן אמת.

5. ניהול שינויים:

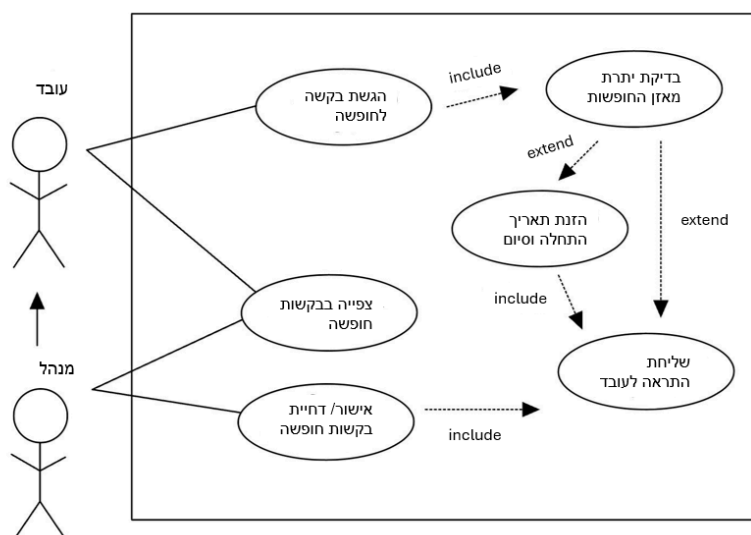
כל שינוי במסמך יתועד בטבלת היסטוריית גרסאות GitHub. לכל חבר צוות יהיו תהליכים שהוא אחראי עליהם והוא יעדכן גרסה כרצונו בלי שזה יפריע לעבודה של האחרים.

6.1 דיאגרמות use case:

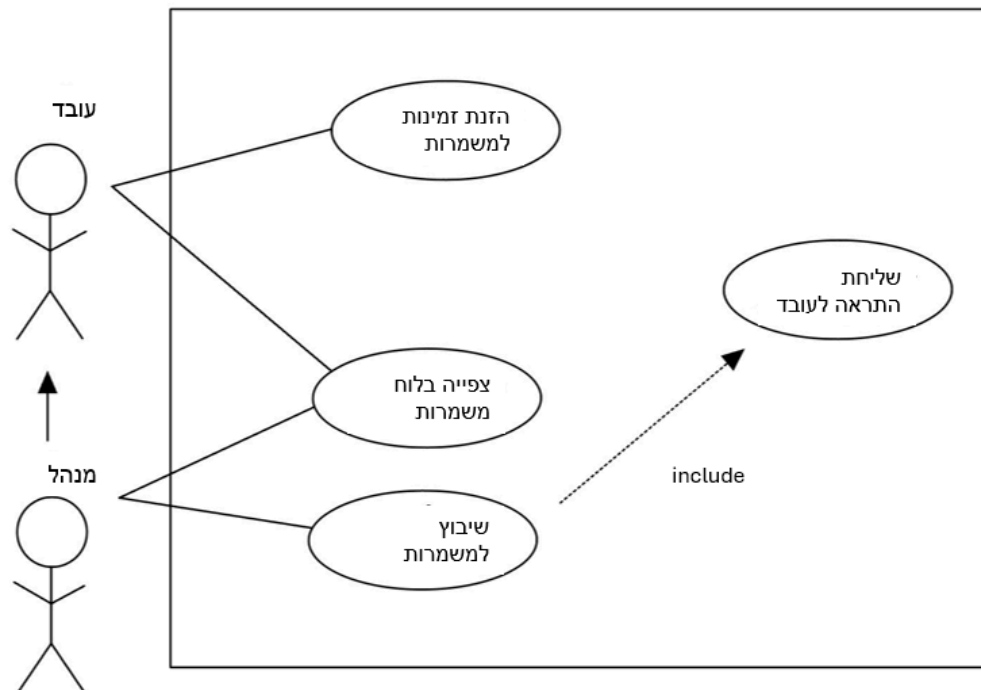
דיאגרמה כללית למערכת



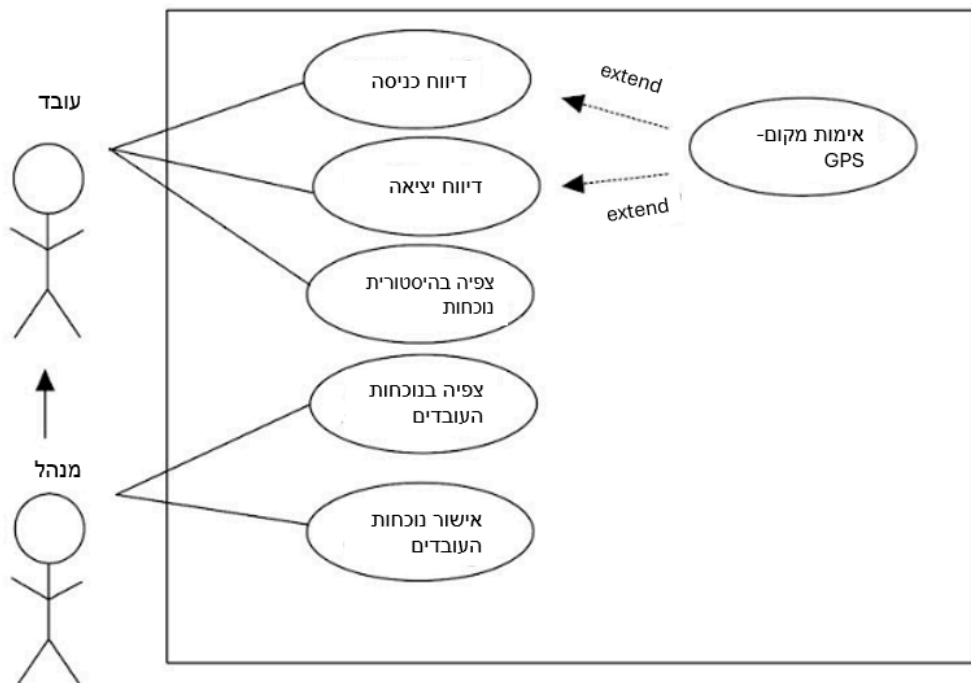
Use case : ניהול חופשות



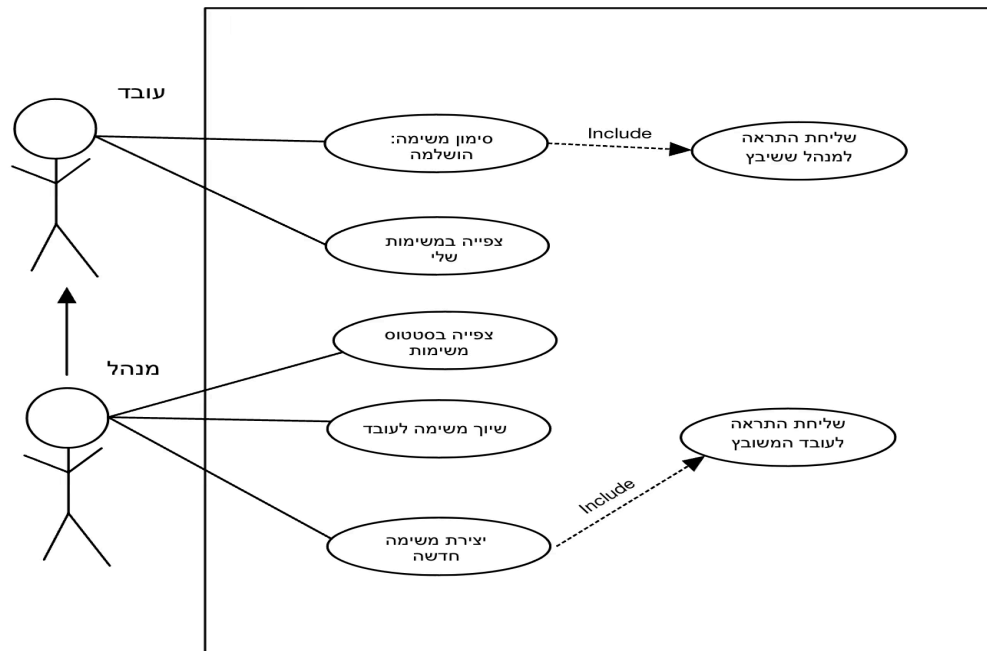
Use case : ניהול משמרות



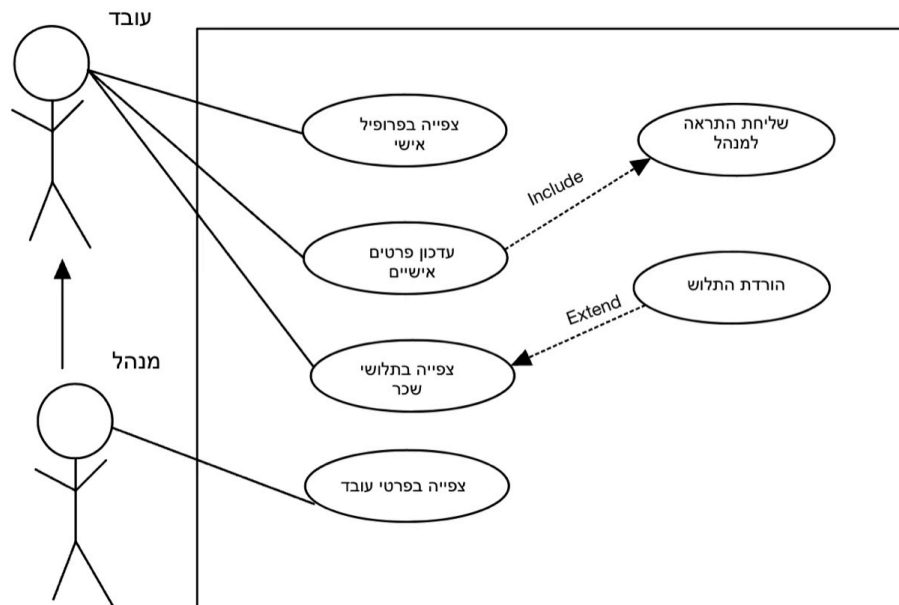
Use case : ניהול נוכחות



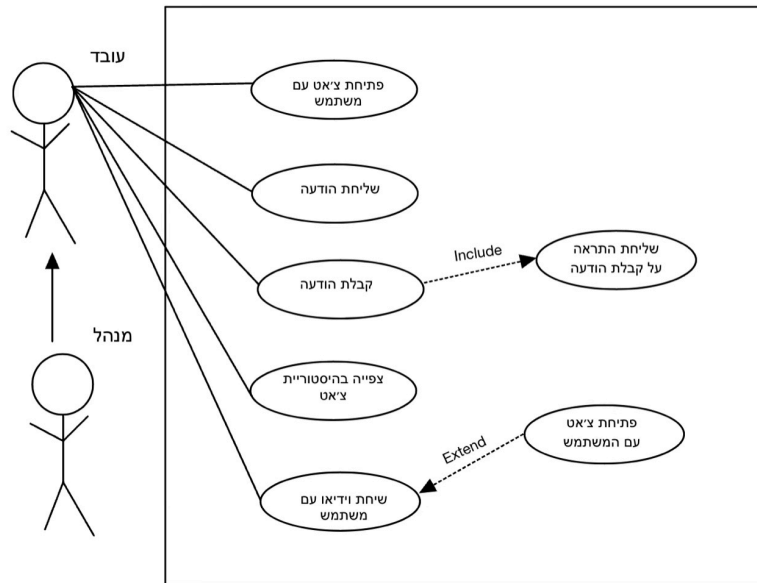
Use case: ניהול משימות



Use case: אזור אישי ותלושי שכר

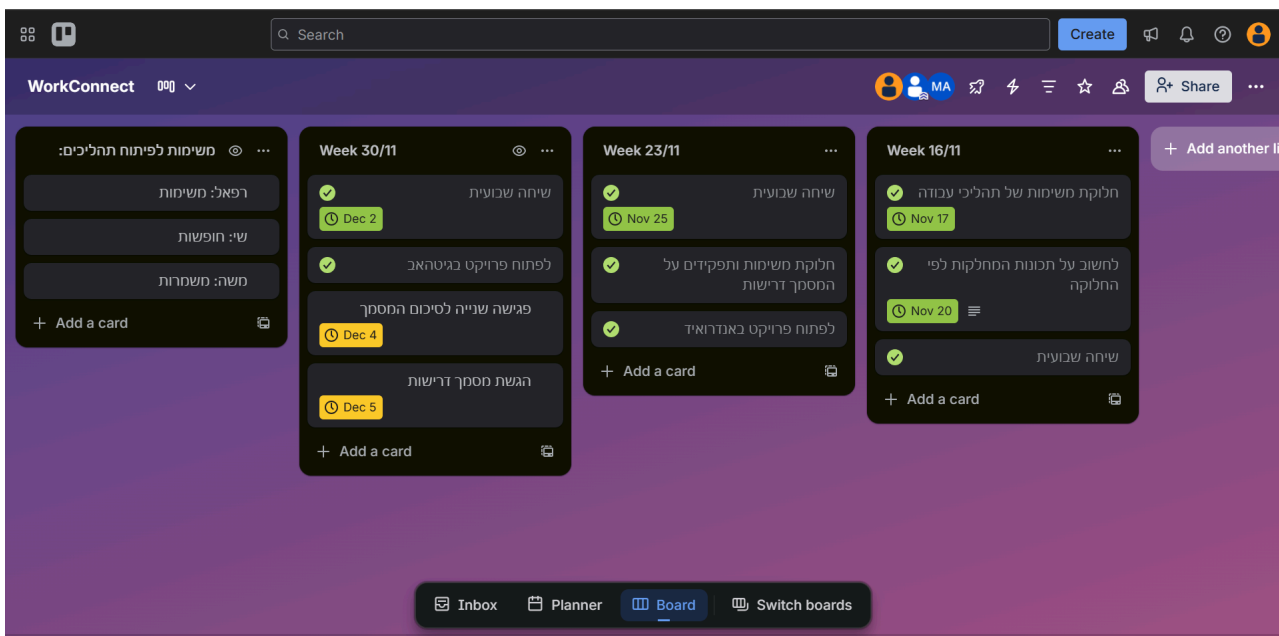


ציאט ושיחות וידיאו : User case



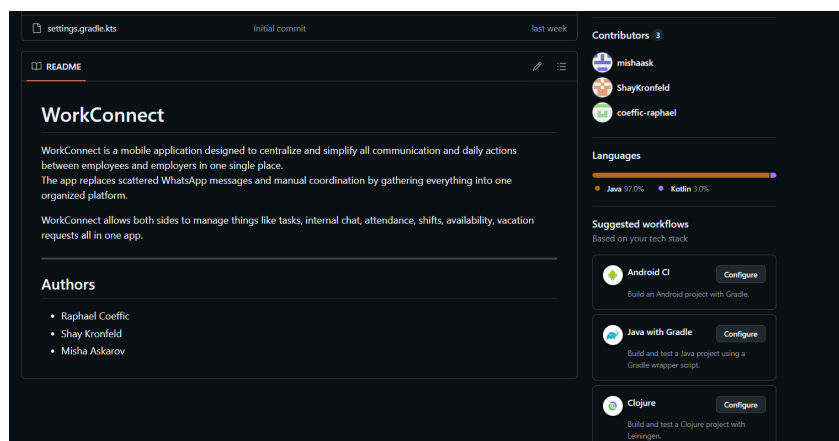
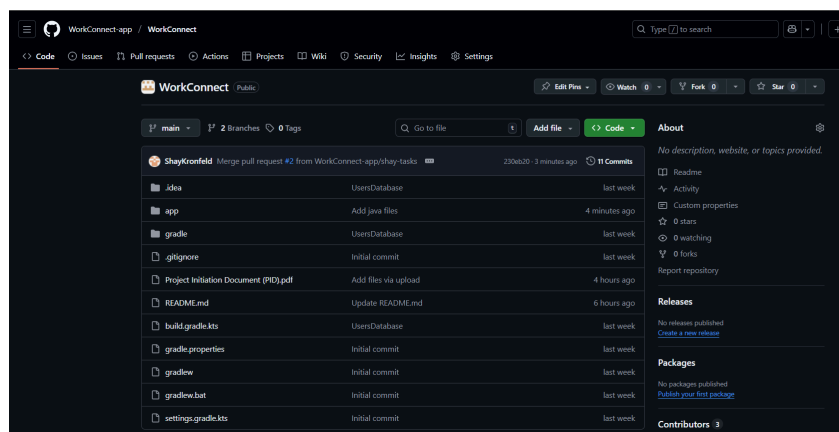
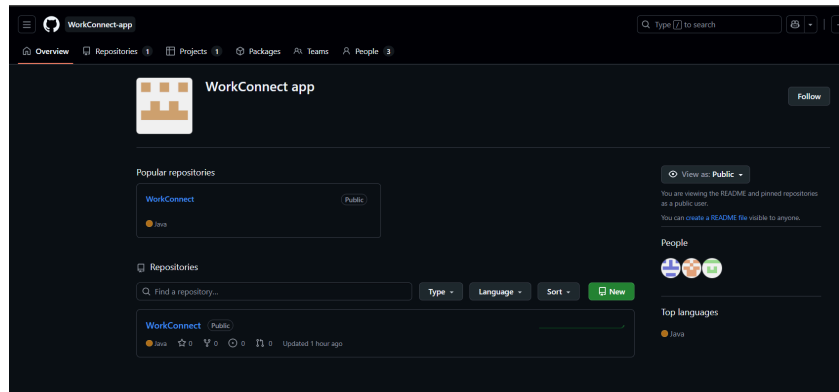
6.2 כלי ניהול משימות:

אנחנו משתמשים בכלי Trello לניהול משימות ותיעוד התקדמות הפיתוח. הצילום מציג את מצב המשימות בזמן אמת, כולל משימות שבוצעו, כאלו שנמצאות בתהליך, ותכנון שבועי.



6.3 מידע נוסף:

הפרויקט מנוהל באמצעות מאגר GitHub ייעודי, המאגד את כל קבצי המקור המסמך ייזומת המסמך דרישות יעלה לאחר הגשת מטלה זו וה-README של האפליקציה. המאגר כולל את מבנה הפרויקט, קבצים וקוד האפליקציה.



שיתוף הפיתוח מתבצע באמצעות GitHub לצורך ניהול גרסאות ושיתופי פעולה, יחד עם Trello לניהול משימות ותכנון שבועי. חברי הצוות המשתתפים בפרויקט מופעים ב-ReadMe של המאגר (בתמונה השלישית)