

מטלה 2- מסמך דרישות

מגישים:

רפאל קוואפיק 337614747
שי קרוןפלד 322234782
משה אסקרוב 314085986

חלק א-

- שיחה עם לקוחות פוטנציאליים, נפרט על שתיים מהשיחות:

- עובד בחברת אבטחה-

■ **אילו פונקציות אתה צריך במערכת?**

- ניהול משמרות באזרה האישי כולל לראות מי פנוי להחלפה.
- יצאט פנימי כדי להימנע מערובב עם הודעות פרטיות.
- שיחות וידאו כי לפעמים צריך לדבר עם עובדים בשטח.
- לראות תלושי שכיר וסידור משמרות ישירות באפליקציה.

■ **מה חשוב לך בחווית המשתמש?**

- שהcoil יהיה פשוט, בלי מיליון מסכים.

■ **מה אתה חושב פתרון 'טוב'?**

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

- מנהל (אחים"ש) בפועל-

■ **מה הקשיים המרכזיים בניהול עובדים?**

- העובדים לא תמיד מעודכנים נכון בזמן.
- הגשת משמרות ובירור זמינות למשמרות מתנהל בוואטסאפ וקשה לעקב.
- צריך לראות מי הגיע, מי אחר ומילא ביקש חופש ואין פלטפורמה שמרכזת הכל במקום אחד.
- מעקב אחריו ביצוע משימות דרך הוואטסאפ בצורה לא מסודרת.

■ **אילו פונקציות אתם צריכים במערכת?**

- דיווח נכון עם מיקום.
- ניהול משמרות, לראות את זמינות העובדים בצורה מסודרת.
- יצאט פנימי ושיחות וידאו.
- ניהול החלפת משמרות בין עובדים עם אישור אחראי בצורה מסודרת באפליקציה.
- אפשרות להגיש משימות לעובדים בצורה מסודרת ולעקב אם בוצע או לא.

■ **מה חשוב לך בחווית המשתמש?**

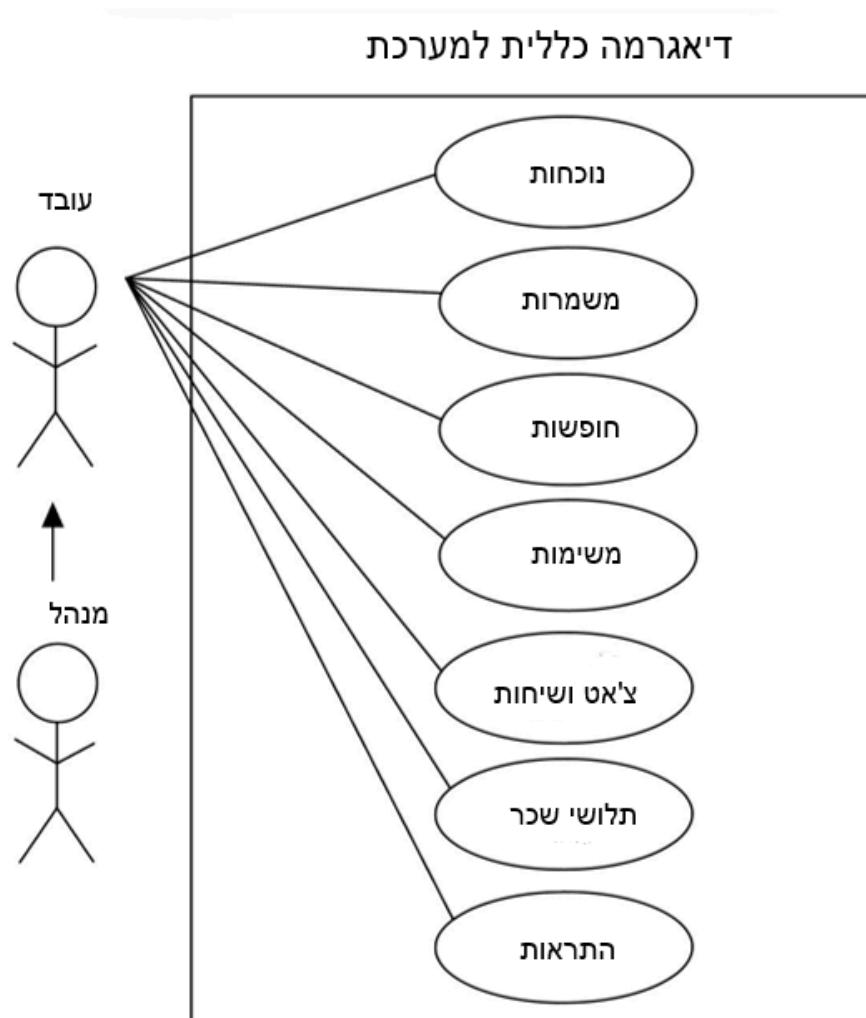
- שיוכנו, ילחצו על כפתור זהה.
- שייהיו התראות על בקשות וחופשות.

■ **מה אתה חושב פתרון 'טוב'?**

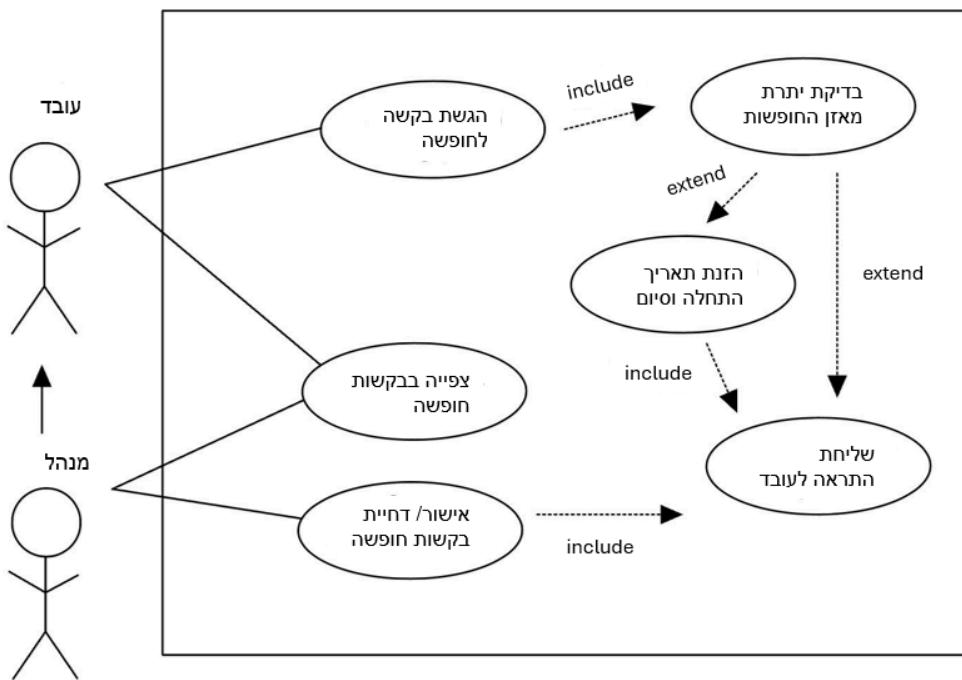
- מערכת שמצויה בזמן אמת את מצב העובדים.

- תיעוד מסודר של זמינות העובדים.
- iscal השיח עם העובדים יהיה מרכז במקום אחד ולא יהיה מבולגן בווטסאף.

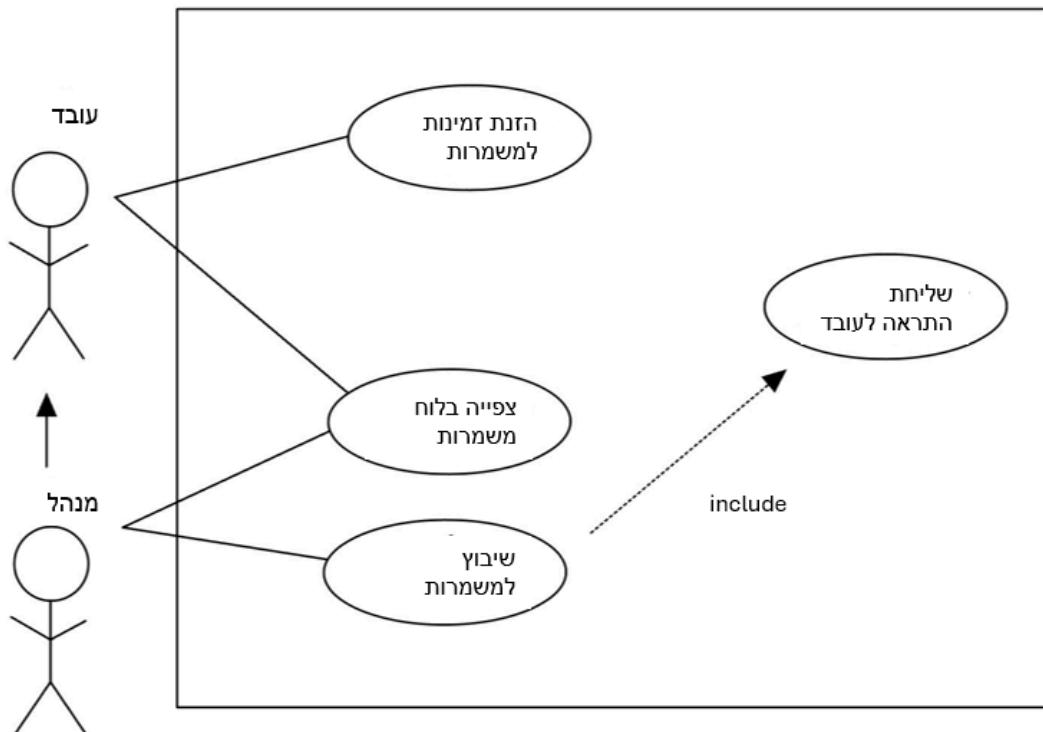
.2



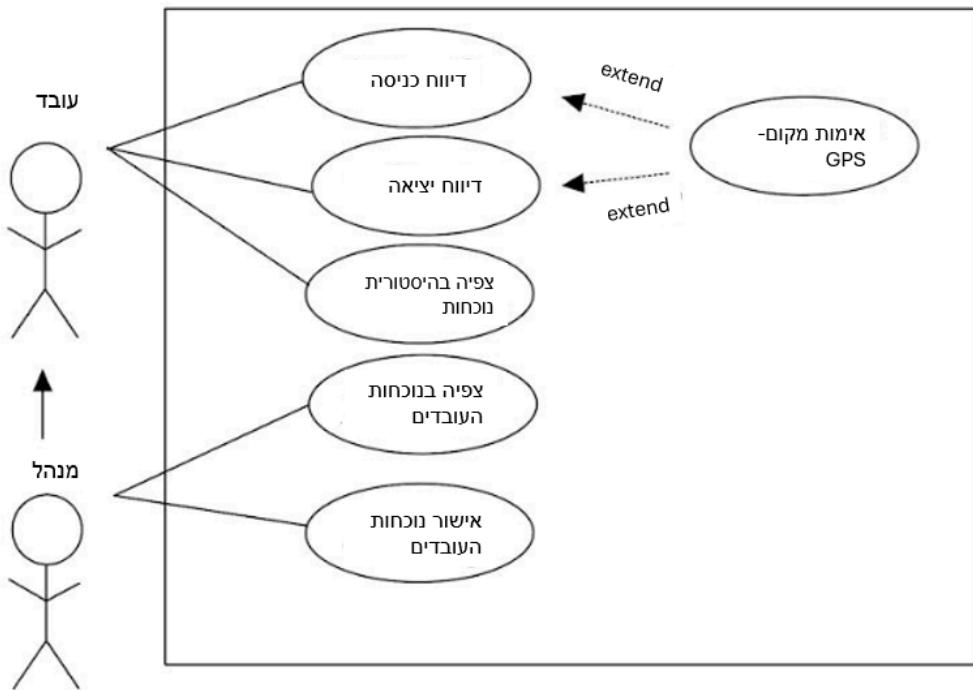
ניהול חופשות : Use case



ניהול משמרות : Use case



Use case : ניהול נוכחות



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

3. מקרה 1 ניהול בקשה חופשה :

- מטרה: העובד רוצה להגיש בקשה חופשה למנהל באופן מסודר מתוך האפליקציה.
- חקנים מעורבים: עובד ומנהל
- תנאים מקדים:
 - העובד מחובר למערכת.
 - פרטי העובד קיימים במסד נתונים.
 - לכל עובד מוגדרים יתרת ימי חופשה נוכחית ומספר ימי המילוק המקסימלי המותר.
- שלבי התהליך:
 1. העובד נכנס למסך "בקשות חופשה".
 2. העובד לוחץ על "הגשת בקשה חדשה".
 3. המערכת מציגה טופס להזנת תאריכי התחלת וסיום החופשה וכן סיבת בקשה 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 יתרה גדולה או שווה למינימום המותר אז הבקשת יכולת להישלח.
 6. המערכת שומרת את הבקשת במסד הנתונים.
 7. המערכת שולחת התראה למנהל.
 8. המנהל רואה את הבקשת במסך הניהול.

• **תוצאה מוצחת:**

- בקשה החופשה נשמרת במערכת בסטטוס של "מתינה לאישור מנהל".
- אם העובד נכנס למינוס מוחר, הדבר מתועד בתנאי החופשה שלו.
- המנהל רואה את הבקשה במסמך הניהול, והעובד רואה אותה בראשית הבקשות.

• **תוצאות חלופיות:**

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

חלק ב-

1. **מבוא:**

מטרה: האפליקציה WorkConnect נועדה ליעיל את סביבת העבודה באמצעות שיפור התקשורת הפנים-ארגוני, ניהול משימות ויכולת ניהול נכחות חכמה. האפליקציה מספקת כלי עבודה נוח לעובדים ולמנהלים, ומאפשרת תפעולiesel, מהיר ומדויק של תהליכי יום-יומיים. האפליקציה כוללת את הfonקציות המרכזיות הבאות:

- דיווח כניסה/יציאה ממוגנת.
- אימוט מיקום המשתמש באמצעות GPS בכניסה לשטח העבודה.
- צפייה בהיסטורית משמרות של שלושת החודשים האחרונים ו朔ות העבודה.
- שליחת הודעות ותקשורת פנים-ארגוני בין עובדים ומנהלים.
- מערכת ניהול משימות עם שירור משימות לעובדים מתאימים.
- התראות בנושאי משמרות, משימות ופעולות חשובות.
- פרופיל אישי לכל עובד כולל מידע תעסוקתי (שכר שנתי, משמרות שבוצעו, תלוishi שכר שהתקבלו).
- הצגת תמונות מצב בזמן אמת למנהל על עובדים שנמצאים בשירות.

• **קהל יעד:**

- **חברות-** אשר מעוניינות ליעיל את מקום העבודה ולספק לעובדים ולמנהלים כלי שיעזר להם ביום יום.
- **מנהליפ-** מנהלים בחברה ישתמשו באפליקציה כדי ליעיל את ניהול העובדים שלהם אם זה בכך שאפשר לצפות בתנוכות של כל עובד, להציג משימות לעובדים והאפשרות ליצור איתם קשר ישיר במידה והם צריכים עזרה בעבודה או מכל סיבה כזו או אחרת.

- **העובדים-** העובדים ישמשו באפליקציה כדי לשפר את הנוכחות שלהם במקום העבודה. אם זה בכך שהם יכולים להיכנס לשירות דרך הטלפון, לעקוב אחריו משימות שיש להם לבצע במקום נוח אוanza שיש להם אפשרות לדבר עם עובדים ובעלי תפקידים אחרים מהחברה אם זה לכיף או כדי להתייעץ קצר לגבי משימה שהוא צריך לבצע. בנוסף לעבוד יהיה מקום נוח לעקוב אחרי הפרטים שלו כעובד אם זה מידע על השכר השנתי, כמה משמרות הוא כבר עשה החודש או פשוט מקום לשמר את כל התלוishi שכר שהעובד קיבל עד היום.

- **מטרת השימוש:** האפליקציה תשמש את העובדים והמנהלים ביום-יום בקרה קבועה. דוגמאות למקירם:
 - **עובד מתחיל את היום**
נכns לאפליקציה < לוחץ על "כניסה למשמרת" > המערכת מבצעת GPS Verify ומשמרת זמן.
 - **עובד מסיים משמרת**
לוחץ על "יציאה" > האפליקציה מחשבת את זמן המשמרת וمعدכנת את ההיסטוריה.
 - **מנהל ישמשימה שהוא רוצה שעובדים יבצעו**
נכns למסך משימות > בוחר משימה חדשה > מוסיף שם, הסבר ובוחר איזה עובדים שלו הוא רוצה שיבצעו את המשימה > מגדר חלון זמן למשימה > לוחץ אישור > בלוט משימות מתעדכנת המשימה החדשה וכל עובד רלוונטי מקבל התראה על משימה חדשה לביצוע.
 - **מנהל מחפש משימות שלא בוצעו**
נכns למסך משימות > רואה מי עדין לא סימ > שלוח תזכורת.
 - **מנהל צריך לעדכן את הוצאות**
כותב הודעה > שלוח לכל העובדים > העובדים מקבלים התראה מידית.
 - **עובד חדש רוצה להבין את המשכורת שלו**
נכns לפרופיל האישי > רואה שכר שעה, תלושי שכר, משמרות שביצעו החודש.
 - **תחומי המוצר:** האפליקציה כוללת:
 - ניהול נוכחות מלא (כניסה/יציאה).
 - אימות מקום באמצעות GPS.
 - מערכת משימות מלאה: יצירת משימות, הקצאה לעובדים, התראות.
 - מערכת הודעות בין עובדים למנהל.
 - שיחות וידאו בין עובדים למנהל.
 - פרופיל אישי מלא לכל עובד.
 - **גבולות ופותוחות:**
 - תמיכה רק ב-Android .
 - אין ניהול שכר מלא - רק מידע לעובד.
 - **הגדרות וKİצרים:**
 - GPS - איתור מיקום.
 - Application Programming Interface (API) - מנגנון תקשורת שמאפשר לאפליקציה להתחבר למערכות אחרות ולבצע פעולות דרך משקח חיצוני.
 - Firestore database - בסיס נתונים בענן מבית Firebase המשמש לאחסן מידע בזמן, כולל משמרות, משימות, הודעות ופרופילים.
 - Firebase Auth – מערכת התחרבות וניהול משתמשים.
 - GPS API - מאפשר שימוש ב GPS של המשתמש לאימות מיקום, והציגת מידע מיקום באפליקציה.

2. תיאור כללי:

• 2.1 צרכים משתמשים:

◦ צרכים העובדים:

- אפשרות לדוח כניסה ויציאה בצורה פשוטה ומהירה, כולל אימות מיקום בכניסה למשמרת.
- לראות תלוishi שכר והיסטוריית משמרות במקום מרכז וגיש.
- מערכת ניהול משימות ברורה ונוחה, שבה העובד רואה מה עלי לבצע ומתי.
- צ'אט פנימי ושיחות ידאו כדי לא לערב הודעות פרטיות מהוואטסאפ.
- דרך מסודרת ופשוטה להגish ולעקב אחריו בקשות חופשה.
- ממשק משתמש קל, מהיר, ולא עמוס.

◦ צרכים המנהליים:

- שליטה מלאה בפעילויות העובדים: מי במשמרת, מי הגיע בזמן, מי אחר ומי נעדר.
- תצוגה מרכזית של זמינות העובדים למשמרות, ללא תלות בוואטסאפ מפוזר.
- יכולה ליצור משימות, לשידר אותן לעובדים ולעקב אחריו ביצוען בפועל.
- שליחת הודעות והתראות לעובד בודד או לכל הצוות.
- מערכת המאפשרת להם לאשר בקשות חופשה ולנהל מעקב מסודר.
- חווית שימוש עם מינימום צעדים: "שייכנסו", ילחצו על כפתור זהה" (מחלק א)

◦ צרכים משתפים לכל המשתמשים:

- סביבת עבודה דיגיטלית מאוחדת במקום אחד, במקום שימוש בכלים מפוזרים.
- עדכוניים והתראות בזמן אמיתי לגבי משימות, הודעות, משמרות ושינויים.
- חווית שימוש אחידה עבור כל העובדים בחברה.
- אפשרות לתפעול יומיומי שוטף: ניהול משמרות, 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 לאחר שכל הנתונים (משמרות, משימות, הודעות והתראות) מסונכרנים בזמן אמיתי דרך Firebase, יש צורך בחיבור אינטרנט זמין.
- העובדים והמנהלים בעלי הרשאות מתאימות לפי Role מההרשאות מנוהלות ע"י Firebase Auth, ולכן מוגדר מראש מי עובד ומניין, ומערכת הפעלה מניצה שהרול הנוכחי כבר הוגדר עבור כל משתמש.

- המערכת מכילה נתונים עדכניים על העובדים
 - לדוגמה: יתרת ימי חופשה, היסטורית משמרות, פרטי פרופיל, זמינות למשמרות מידע זה נשמר בסיסי באפליקציות תקין ומעודכן.
 - המשתמשים יודעים לבצע שימוש בסיסי באפליקציות לאחר והמשק דרוש פעולות פשוטות (ניהול משמרות, מסך בית, משימות וכו'), המערכת מניחה שהמשתמשים מ信ונים בשימוש באפליקציות מובייל.
- תלוויות:
- Firebase Auth ניהול משתמשים, הרשאות, סיסמות והרשאות גישה לא. בלבדיו לא ניתן לגשת למערכת.
- Firebase, משמש כאגר המרכזி לכל הנתונים: משמרות, משימות, הודיעות, פרופילים והתראות. המערכת תלויה ביכולת לשמר ולסנן נתונים בזמן אמת. במקרה של נפילה, תציג שגיאה ותבצע ניסיוןהתחברות חוזרת.
- API GPS של המכשיר נדרש לאימות מקום בעת כניסה למשמרת. ללא גישה ל-GPS לא ניתן לבצע אימות כניסה תקין.
- Firebase Cloud Notifications / Push Notifications משמשות לשילוח התראות על משימות, משמרות, הודיעות וביקשות חופשה. המערכת תלויה ביכולת לשולח התראות בזמן אמת.

3. דרישות מערכת:

- דרישות פונקציונליות:
 - ניהול כניסה-
 - המערכת תאפשר כניסה/יציאה ממשמרת.
 - תשמור את זמן הכניסה, זמן היציאה ותזודד את מקום העובד.
 - תמנע כניסה נוספת בלי יציאה.
 - ניהול משימות-
 - המנהל יוכל ליצור משימות חדשות.
 - המנהל יוכל לשிரט משימה לעובדים.
 - העובד יקבל התראה כאשר יש משימה חדשה.
 - העובד יוכל לסמן משימה כבוצע.
 - דף שגיגש לעובדים ומנהלים שיציג את כל המשימות שלא בוצעו עדין במקום אחד.
 - דף המציג את כל המשימות שהוגשו בעבר לא משנה אם בוצעו או לא בסוף (ההיסטוריה משימות נשמרת לחצי שנה).
 - תקשורת-
 - המנהל יוכל לשולח הודיעות לכל העובדים או לעובד ספציפי.
 - העובדים יקבלו התראות על הודיעות חדשות.
 - הודיעות ישמרו מסך ייעודי.
 - אפשרות לשיחה וידאו אם זה בין זוג אנשים או קבוצה.

- **פרופיל אישי.**

- כל עובד יוכל לראותו: שכר שנתי, 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.
- המשתק יהיה מחולק למספר מסכימים עיקריים:
 - מסך הראשי (כניסה/יציאה) בו המשתמש יתחבר עם שם משתמש וסיסמה או יעשה רישום אם זה השימוש הראשון.
 - מסך בית שבו ניתן לגישה לכל הפעולות של האפליקציה כמו משימות, הודעות בין עובדים, שיחות וידאו, פרופיל אישי, משמרות.
 - לכל אחת מהפעולות יהיה דף שיניה את כל התהילה לדוגמה אחרי שהמשתמש נכנס לדף הבית ובחירה, הוא יעבור לדף חדש עם כל המידע על המשמרות ואפשרות לחזור לדף הבית.

- **אינטגרציה "צד שלישי".**-

- Firebase Auth - לשימירה על משתמשים קיימים ורישום משתמש חדשים.
- Firestore Database - שימירה של מידע כמו משמרות, משימות ועוד.
- GPS API - מאפשר שימוש ביכולות ה-GPS של הטלפון של הלוקה.

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

- כל הנתונים (משמרות, משימות, הודעות, פרופילים) ישמרו ב-Firestore.Database.
- הנתונים יסונרנו בזמן אמת בין עובדים ומנהלים.
- Firestore ישמש כמאגר מרכזי יחיד.

- **פיצרים של המערכת:**

- **מידע שימושי למנהלים:** מי במשמרת עציו, 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 במידה 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, היסטוריות עבודה של שלושת החודשים האחרונים, פרטיים אישיים, פעולות שבוצעו

כמו פתיחת משימה או ביצוע משימה.

- **דרישות לא פונקציונליות:**

- **ביטחונית-**

- כל פעולה באפליקציה תטען תוך עד 2 שניות.

- שמירת נתונים בענין ≥ 5 שניות.

- **דרגות ואפשרויות פיתוח-**

- המערכת תוכל לתמוך במספר רב של חברות ומספר רב של משתמשים בכל חברה (עובדים ומנהלים).

- הנתונים ישמרו ב-[Firestore](#) המאפשר התרחבות.

- **בטיחות-**

- גישה לפיצ'רים נעשית לפי Role, כך שעובד לא יוכל לבצע פעולות של מנהל.

- סיסמות ומשתמשים מנהלים רק ע"י [Firebase](#).

- **זמן-**

- האפליקציה תהיה זמינה כל עוד [Firebase](#) פעיל.

- במקרה של נפילה, תציג הודעה 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 מהשירות, קבלת משימות וסימון ביצוע, קבלת הודעות והתראות מנהלים, צפיה בפרופיל האישי ובהיסטוריה של משתמש, הגשת בקשות חופשה ומעקב אחריה, תקשורת עם המנהלים ועם עובדים אחרים.

- **מאפיינים ייחודיים:** שימוש יומיומי ממוקד בפעולות אישיות, ממשק פשוט, קל לתפעול ולא עמוס, הרשאות בסיסיות בלבד ככלומר ללא יכולת השפעה על עובדים אחרים, הצורך במידע מהיר וברור על השירות ועל המשימות.

- **מנהלים(Secondary):**

- המנהלים הם משתמשי מערכת מתקדמים המחזיקים בהרשאות נוספות

- האפשרות להם לנהל את העובדים ואת סביבת העבודה.

- **תפקידים:** מעקב אחר נוכחות העובדים בזמן אמת, יצירת משימות ושיכון 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 מדויקים וברורה לקבלת החלטות, תפעול מסכים "יעודים" לניהול משמרות ומשימות

• 4.2 דרישות ייחודיות למשתמשים:

○ עובדים:

- הרשות מיוחדות: גישה למידע אישי בלבד: משמרות, שיחות, מושימות, פרופיל, פרטיים, תלושים והתראות ובקשות חופשה. אין לעובד הרשות לצפות בנוטנו, עובדים אחרים או בסיסי ניהול.
- תצוגות מותאמות: מסך בית המציג את המשמרת הנווכחית, המשימות האישיות וההתראות החדשנות. צ'אט פנימי ושיחות וידאו, פרטיים אישיים, פרופיל. תצוגה ממוקדת בפעולות האישיות של העובד בלבד.
- פונקציונליות ייחודית: דיווח כניסה/יציאה מהמשמרת, קבלת מושימות וסימון ביצוען, קבלת הודעות מהמנהלים, שיחות צ'אט ווידיאו עם עובדים אחרים, הגשת בקשות חופשה ומעקב אחר סטטוס, צפיה בתלושי שכר ובהיסטוריית משמרות.

○ מנהליים:

- הרשות מיוחדות: גישה לנוטנו וכוחות, זמינות ומשימות של כל העובדים, כולל יכולת לבצע פעולות המשפיעות על משתמשים אחרים כמו אישור חופשה או מתן משימה.
- תצוגות מותאמות: Dashboard ניהול להציג תמונה מצב בזמן אמת, בסיסי ניהול יעודיים למשמרות, נוכחות העובדים, משימות וחופשות שאינן זמינים לעובדים.
- פונקציונליות ייחודית: יצירת מושימות ושיקון לעובדים, אישור או דחית בקשות חופשה, שליחת התראות והודעות לצוות, ניהול זמינות עובדים ושיבוץ למשמרות, צפיה בנוטנו וכוחות בזמן אמת.

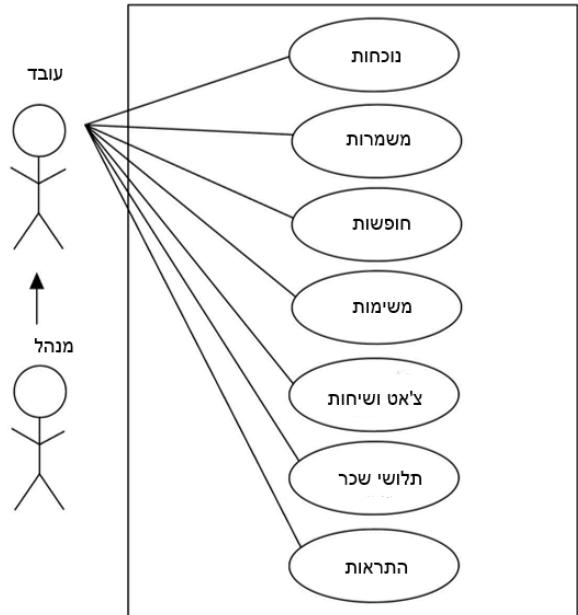
5. ניהול שינויים:

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

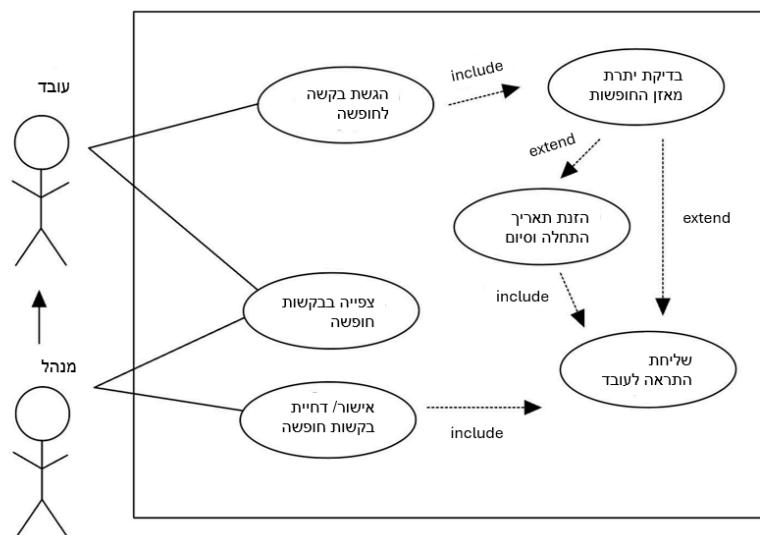
6. נספחים:

6.1 דיאגרמות use case

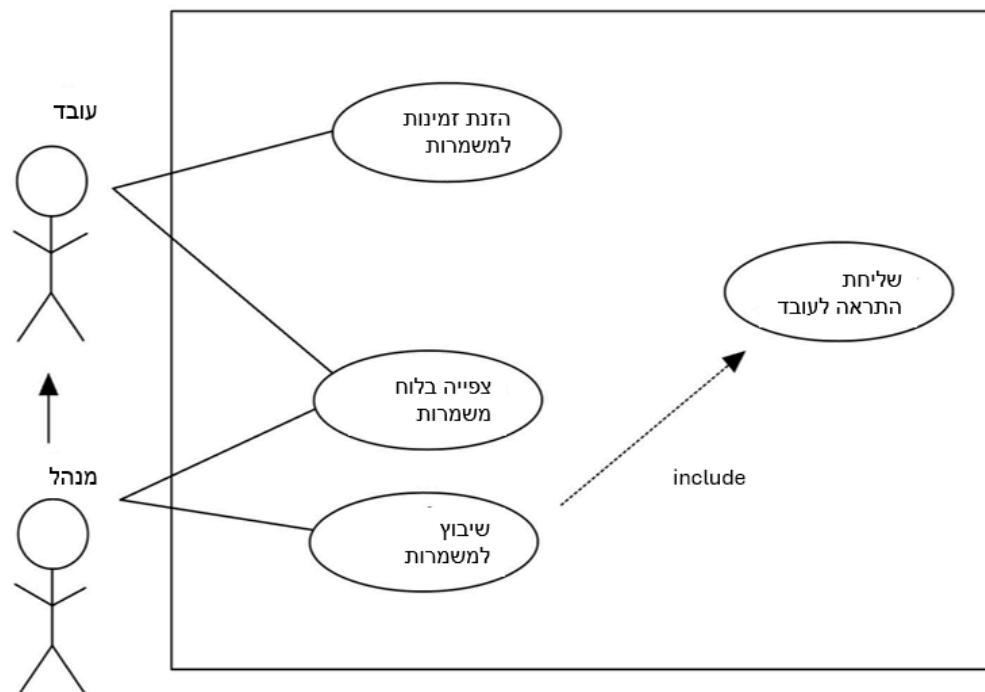
דיאגרמה כללית למערכת



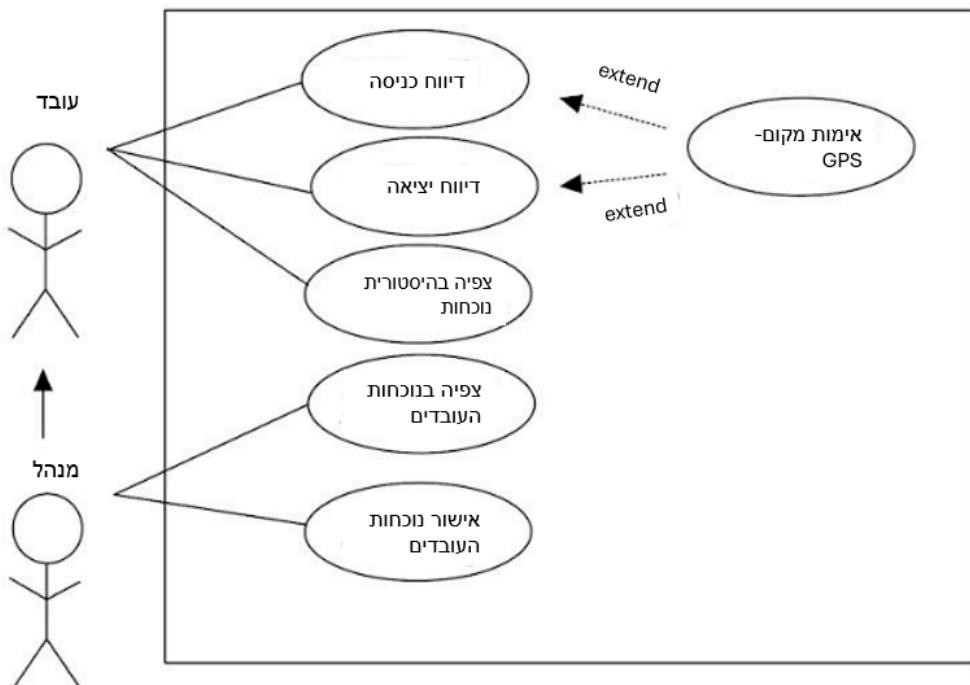
nihal Chofshot : Use case



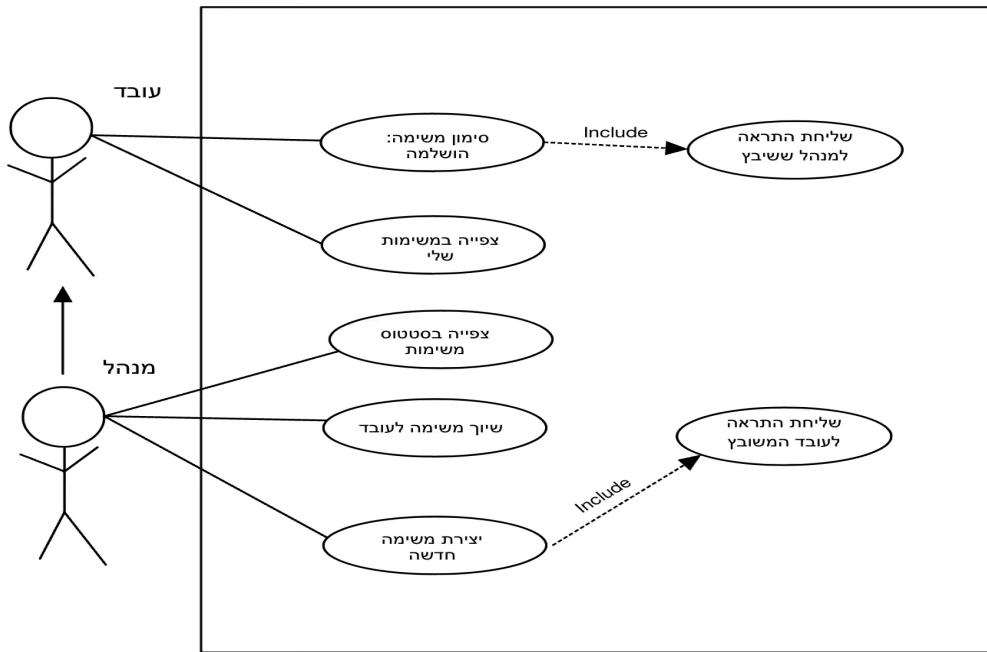
ניהול משמרות : Use case



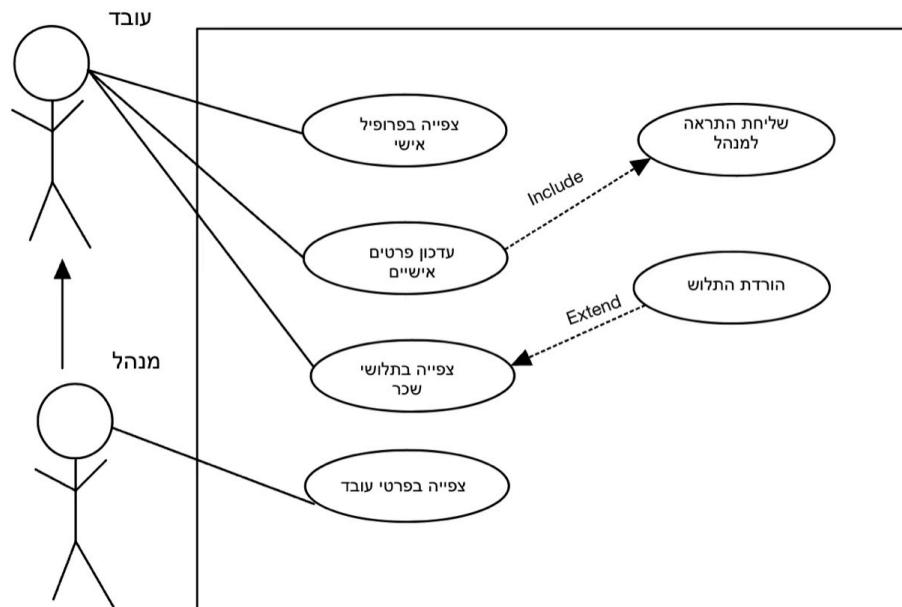
ניהול נוכחות : Use case



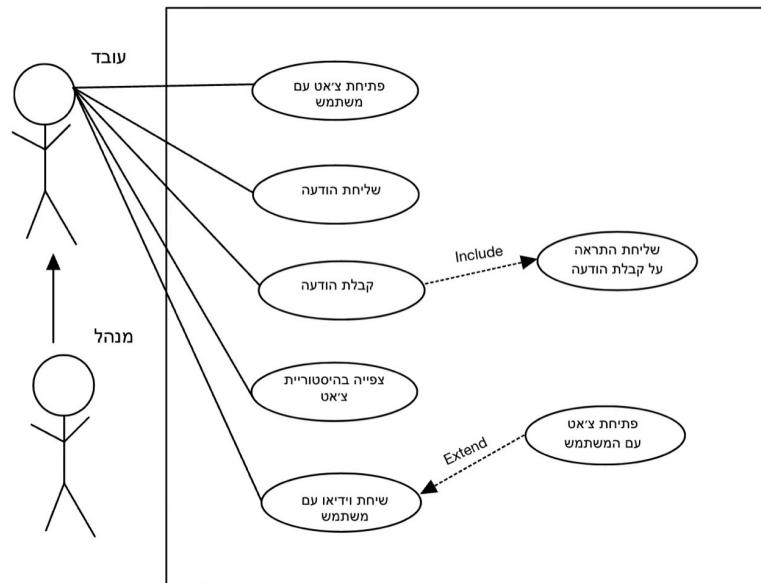
ניהול משימות: Use case



אזור אישי ותלוšíי שכר: Use case

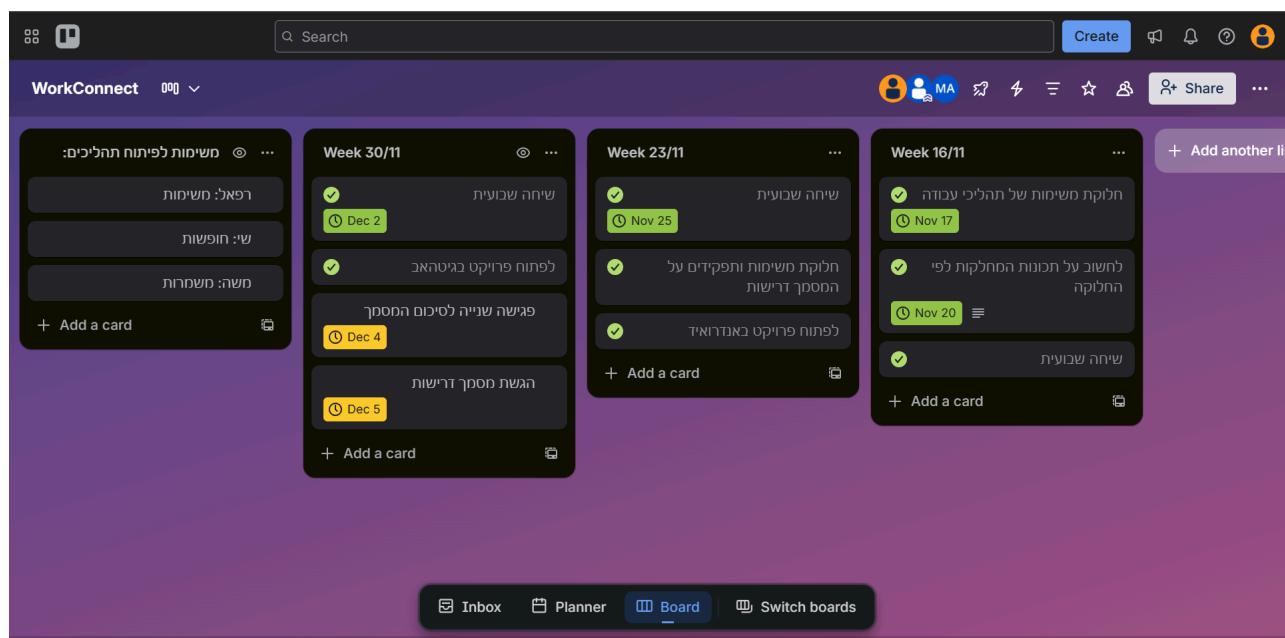


צ'אט ושיחות וידיאו : User case :



2. כל ניהול משימות:

אנחנו משתמשים בכל Trello לניהול משימות וטייעוד התקדמות הפיתוח. הצלום מציג את מצב המשימות בזמןאמת, כולל משימות שבוצעו, כאלה שנמצאות בתהילר, וככונן שבועי.



6.3 מידע נוסף:

הפרויקט מנוהל באמצעות מאגר GitHub יי'ודי, המאגד את כל קבצי המקור המסמך ייזמות המסמך דרישות עליה לאחר הגשת מטלה זו וה- README של האפליקציה. המאגר כולל את מבנה הפרויקט, קבצים וקוד האפליקציה.

The figure consists of three vertically stacked screenshots of a GitHub repository page for 'WorkConnect-app'.
The top screenshot shows the repository overview, featuring a thumbnail of the app icon, repository statistics (1 star, 0 forks, 0 issues), and a link to the README file.
The middle screenshot shows the code commit history for the 'main' branch, listing several commits from 'ShayKronfeld' made within the last week, including merges and initial commits for files like 'idea', 'app', 'gradle', and 'README.md'.
The bottom screenshot displays the contents of the 'README' file, which describes WorkConnect as a mobile application for communication and daily actions between employees and employers. It also lists authors: Raphael Coeffic, Shay Kronfeld, and Misha Askarov.

שיתופי הפיתוח מתבצע באמצעות GitHub לצורכי ניהול גרסאות ושיתופי פעולה, יחד עם SoLo Trello לניהול משימות ותוכנן שבועי. חברי הצוות המשתתפים בפרויקט מופיעים בREADME של המאגר (בתמונה השלישייה)