מכון טכנולוגי חולון · הפקולטה למדעים · פרויקט גמר

Smart Shade

מערכת אוטומציה וניהול תחזוקה לקמפוס — מסמך פיתוח ובסיס מדעי

סטודנט

תומר בראל

מנחה

נתנאל זוהר

תואר

B.Sc. מדעי המחשב

תאריך

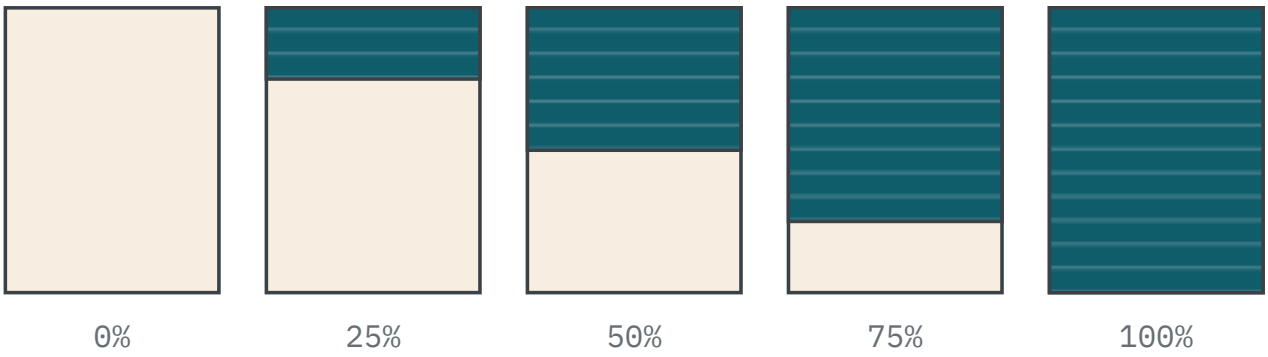
ספטמבר 2026

מסמך ייזום

MaintControl — מארק ישראל

מאגר קוד

github.com/Tomer98/My-Shade-Project



איור 1. חמשת מצבי הסגירה הבדידים שהמערכת מייצרת. הציון הרציף שמחושב מהחיישנים מוצמד תמיד לאחד מהם — ראו סעיף 5.1.

תוכן העניינים

1. מבוא: רקע, הגדרת הבעיה ומונחי מפתח
2. מטרות הפרויקט ודרישות פונקציונליות
3. דרישות לא-פונקציונליות
4. ארכיטקטורה ועיצוב המערכת
5. אלגוריתמים ובסיס מדעי
6. בדיקות ואימות
7. סקירת הפתרון והמימוש
8. המלצות להמשך פיתוח
9. מקורות

1 מבוא

1.1 רקע

תחזוקה מונעת במתקנים גדולים — קמפוס אקדמי, בית חולים, מרכז לוגיסטי — מתנהלת ברובה על נייר, בגיליונות אלקטרוניים ובשיחות טלפון. הידע המקצועי נשמר בראשם של עובדים ותיקים, לוחות הזמנים מנוהלים בזיכרון, והמנהל לומד על תקלה כשמישהו מתלונן.

במקביל, מערכות בקרת מבנה (BMS) שמפעילות תריסים, מיזוג ותאורה פועלות כמערכת נפרדת לחלוטין ממערכת התחזוקה. כאשר מנוע תריס מפסיק להגיב, מערכת הבקרה יודעת זאת — אבל אין לה דרך להפוך את הידיעה הזו לעבודה שמישהו יבצע. הפער בין שתי המערכות הוא שממלא את יומן התקלות.

1.2 הגדרת הבעיה

מסמך הייזום MaintControl מגדיר שני חסרים משלימים:

- **בשטח.** לעובד התחזוקה אין פלטפורמה שאומרת לו מה לעשות, באיזה סדר, והאם הנוהל עודכן מאז הביקור הקודם. תיעוד הביצוע, אם קיים, נעשה בדיעבד ומהזיכרון.
- **בחדר הבקרה.** למנהל אין מדד אמין לאיזו תחזוקה בוצעה בפועל, מי עמוס ומי פנוי, ואיזה ציוד חוזר ונכשל. ההחלטות מתקבלות על סמך תחושה.

לשני אלה מתווסף חסר שלישי שהמסמך רומז עליו אך אינו מגדיר: **אין לולאה סגורה**. גם כשמערכת אוטומטית מזהה כשל, אף אחד לא הופך אותו למשימה.

1.3 הצעת הערך

Smart Shade מאחדת את שלושת החסרים בפלטפורמה אחת. היא מפעילה את הציוד לפי תנאי סביבה נמדדים, מנהלת את עבודת התחזוקה עליו, ומחברת ביניהם: **כשל שמדווח בשטח יוצר התראה מיידית, וסבב תחזוקה שהסתיים בהצלחה קובע את עצמו מחדש למועד הבא.** המערכת אינה עוקבת אחרי העבודה בלבד — היא גם יוצרת אותה.

1.4 מונחי מפתח

מונח	הגדרה במסמך זה
אזור (Area)	יחידה פיזית נשלטת — חדר, כיתה או מרחב — בעלת חיישנים, תריס, מיקום על המפה וקואורדינטות GPS.
מצב סגירה	מיקום התריס כאחוז, 0–100. המערכת מייצרת חמישה ערכים בלבד: 0, 25, 50, 75, 100.
ציון גולמי	תוצאת הממוצע המשוקלל של הטמפרטורה ועוצמת האור, בתחום רציף [0,1], לפני ההצמדה.
משימה (Mission)	עבודת תחזוקה מחזורית המשויכת לאזור, לעובד ולעיתים לפריט ציוד, ומורכבת מתת-משימות.
תת-משימה	פריט בודד בצ׳ק-ליסט. מצביה: ממתין, בוצע, נכשל. כשל מחייב הערה מילולית.
שוכר (Tenant)	חברה. כל רשומה במערכת שייכת לשוכר אחד, וההפרדה נאכפת בשאילתה ולא בממשק.
היסטרזיס	אזור מת סביב הערך הנוכחי שבתוכו המערכת אינה מגיבה, כדי למנוע ריצוד סביב סף.

2 מטרות הפרויקט ודרישות פונקציונליות

2.1 מטרות

1. לספק לעובד השטח כלי נייד יחיד שמגדיר את עבודת היום, מנווט אליה ומתעד אותה במקום.
2. לספק למנהל תמונת מצב בזמן אמת של הסביבה, של העומס ושל מצב הציוד.
3. לשלוט בציוד ההצללה אוטומטית לפי מדידות, בלי התערבות אנושית שגרתית.
4. לסגור את הלולאה: להפוך כשל שדווח לעבודה מתוזמנת, והצלחה לסבב הבא.
5. לשרת מספר חברות במקביל תוך בידוד נתונים מלא.

2.2 בעלי עניין

בעל עניין	צורך עיקרי	כיצד המערכת עונה
עובד תחזוקה	לדעת מה לעשות והיכן	אפליקציית אנדרואיד עם משימות היום, צ׳ק-ליסט, ניווט ומצלמה
מתכנן	לשבץ עבודה נכון	דירוג מוצע של עובדים לפי אזור, זמינות ועומס פתוח
מנהל מתקן	שליטה ובקרה	מפת קמפוס, טלמטריה חיה, דוחות וייצוא
משתמשי המבנה	נוחות תרמית וויזואלית	בקרת הצללה אוטומטית לפי חום ואור
מנהל מערכת	שליטה בגישה	אישור נרשמים, קביעת תפקידים, בידוד בין חברות

2.3 דרישות פונקציונליות

מזהה	תחום	דרישה
`FR-01`	גישה	משתמש חדש נרשם עצמאית, בוחר חברה, ונוצר במצב `pending` ללא יכולת כניסה.
`FR-02`	גישה	מנהל מאשר או דוחה נרשם וקובע לו תפקיד ואזור עבודה.
`FR-03`	גישה	התחברות מחזירה JWT חתום המכיל מזהה משתמש, תפקיד ומזהה חברה.
`FR-04`	גישה	איפוס סיסמה בדוא״ל באמצעות טוקן חד-פעמי בעל תוקף מוגבל.
`FR-05`	גישה	שלושה תפקידים — מנהל, מתכנן, עובד תחזוקה — עם הרשאות נבדלות הנאכפות בשרת.
`FR-06`	ניטור	מפת קמפוס המציגה את כל האזורים ואת מצבם הנוכחי.
`FR-07`	ניטור	לוח אזור עם טמפרטורה, עוצמת אור ומצב תריס, המתעדכן בדחיפה ולא בתשאול.
`FR-08`	ניטור	נקודת קצה לקליטת מדידות מחומרה, נגישה ללא אסימון (חומרת שדה).
`FR-09`	ניטור	היסטוריית מדידות מוצגת בגרפים לאורך זמן.

מזהה	תחום	דרישה
FR-10	אוטומציה	חישוב מחזורי של מצב התריס הרצוי מתוך הטמפרטורה ועוצמת האור.
FR-11	אוטומציה	כללי חירום גוברים על החישוב: גשם או סערה, חום קיצוני, קור קיצוני.
FR-12	אוטומציה	שינוי זניח אינו מייצר פעולה — מנגנון היסטרזיס.
FR-13	אוטומציה	שילוב נתוני מזג אוויר חיצוניים כמקור משלים למדידות המקומיות.
FR-14	אוטומציה	דריסה ידנית של המצב האוטומטי על ידי משתמש מורשה.
FR-15	תחזוקה	יצירת משימה עם אזור, עובד, ציוד, תדירות בימים ורשימת תת-משימות.
FR-16	תחזוקה	בשיבוץ מוצגת רשימת עובדים מדורגת לפי התאמת אזור, זמינות ועומס פתוח.
FR-17	תחזוקה	עובד רואה את משימות היום שלו בלבד.
FR-18	תחזוקה	סימון תת-משימה כבוצעה או כנכשלה; כשל מחייב הערה מילולית.
FR-19	תחזוקה	צירוף תיעוד ויזואלי לתת-משימה, הנשמר באחסון אובייקטים עצמי.
FR-20	תחזוקה	ניווט מהמיקום הנוכחי לקואורדינטות האזור.
FR-21	תחזוקה	סיום מוצלח של כל תת-המשימות סוגר את המשימה ויוצר את המופע הבא בהתאם לתדירות, עם העתקת הצ׳ק-ליסט.
FR-22	תחזוקה	כשל בתת-משימה מסמן את המשימה ככושלת, מונע קביעה מחדש, ומייצר התראה בעדיפות גבוהה הכוללת אזור, זמן, משימה, תת-משימה והערת העובד.
FR-23	נכסים	ניהול ציוד לפי אזור עם סטטוס שירות; משימה יכולה להתמקד בפריט ציוד.
FR-24	ידע	עובדים כותבים מדריכי נוהל; מדריך מתפרסם רק לאחר אישור מנהל.
FR-25	ידע	דירוג מדריכים על ידי משתמשים ומיון לפי דירוג ממוצע.
FR-26	דיווח	דוחות: אחוז השלמה, עומס לפי עובד ולפי אזור, מצב ציוד, ותת-המשימות הנכשלות ביותר.
FR-27	דיווח	ייצוא נתוני דוח לקובץ CSV.
FR-28	ריבוי שוכרים	כל רשומה משויכת לחברה; משתמש אינו יכול לגשת לנתוני חברה אחרת בשום מסלול.
FR-29	ממשק	ממשק דו-לשוני, אנגלית ועברית, כולל היפוך פריסה מלא לימין-לשמאל.
FR-30	ממשק	אפליקציית אנדרואיד מותקנת עם גישה למצלמה ול-GPS המקוריים של המכשיר.

3 דרישות לא-פונקציונליות

מזהה	קטגוריה	דרישה ואופן המימוש
NFR-01	ביצועים	מחזור ההערכה רץ כל 5 שניות לכל האזורים. עדכון נדחף ללקוחות ב-WebSocket ולא בתשאול מחזורי, כדי שמספר הלקוחות לא ייתרגם לעומס לינארי על מסד הנתונים.
NFR-02	ביצועים	מחזור שאינו מייצר שינוי אינו כותב למסד. הכתיבה מותנית במעבר סף ההיסטרזיס.
NFR-03	אבטחה	סיסמאות נשמרות כ-hash עם bcrypt בלבד. אסימון JWT חתום; כל נתיב מוגן עובר אימות בצד השרת. אין הסתמכות על הסתרה בממשק.
NFR-04	אבטחה	סטטוס המשתמש נבדק **אחרי** אימות הסיסמה, כדי שהודעת השגיאה לא תסגיר אילו שמות משתמש קיימים (מניעת מניית משתמשים).
NFR-05	אבטחה	כל שאילתה מפורמטרת. אין שרשור מחרוזות לתוך SQL.
NFR-06	בידוד	מזהה החברה נלקח מן האסימון החתום ולא מגוף הבקשה, ונכנס כתנאי `WHERE` לכל שאילתה. בקשה עם מזהה זר מחזירה תוצאה ריקה ולא שגיאת הרשאה.
NFR-07	התרחבות	ה-API חסר-מצב, ולכן ניתן להריץ מספר עותקים מאחורי מאזן עומסים. חיבורי המסד מנוהלים ב-pool. שדות מפתח מאונדקסים.
NFR-08	עמידות	שירות מזג האוויר החיצוני מוגן בשלוש שכבות: ניסיון חוזר עם השהיה מעריכית, דילוג מיידי על שגיאות 4xx, ומטמון בעל תוקף המשרת את הערך האחרון. כשל בשירות אינו מפיל את המערכת.
NFR-09	ניידות	המערכת ארוזה בקונטיינרים. הפריסה בפועל היא על AWS, אך אינה תלויה בשירות ייחודי לספק, ולכן ניתנת להעברה.
NFR-10	נגישות	תמיכה מלאה בימין-לשמאל, לא תרגום טקסט בלבד: כיוונים, יישור וסדר רכיבים מתהפכים.
NFR-11	תחזוקתיות	101 בדיקות אוטומטיות. כל דחיפה מפעילה אינטגרציה רציפה הכוללת את שתי הסוויטות ובנייה מלאה של הלקוח.
NFR-12	תאימות	אותו בסיס קוד משרת דפדפן ואנדרואיד. גישה לחומרה נעשית דרך תוספים מקוריים עם נפילה חזרה ל-API של הדפדפן.

4 ארכיטקטורה ועיצוב המערכת

4.1 מבנה שכבות

המערכת בנויה בארבע שכבות עם תלות חד-כיוונית: כל שכבה מכירה רק את זו שמתחתיה. שכבת הבקרים אחראית על HTTP בלבד — פענוח בקשה, אימות קלט והחזרת תשובה — והלוגיקה העסקית יושבת בשכבת השירותים כפונקציות טהורות. ההפרדה הזו היא שמאפשרת לבדוק את אלגוריתם ההחלטה בלי להרים שרת ובלי מסד נתונים.

לקוח
Presentation

`i18n EN/HE` `Capacitor → Android` `Recharts` `Socket.io client` `Vite` `React 19`

▼ REST · WebSocket ▲

בקרים ונתיבים
controllers · 49 endpoints 10

`scheduler` `reports` `alerts` `guides` `equipment` `missions` `sensors` `areas` `users` `auth`

▼ ▲

שירותים
Domain logic

`emailService` `storageService` `weatherService` `scheduler` `decisionEngine`

▼ ▲

נתונים ואחסון
Persistence

`SQL migrations` `AWS S3` `connection pool` `MySQL 8 — 13 tables`

איור 2. שכבות המערכת. חומרת החיישנים כותבת ישירות לשכבת הבקרים בנתיב ציבורי; שירות מזג האוויר נצרך בשכבת השירותים.

4.2 מבנה הנתונים

הסכימה כוללת 13 טבלאות במסד יחסי מנורמל. הגרעין הוא היחס `companies → users / areas → missions → mission_subtasks`, כאשר `company_id` חוזר בכל טבלת תוכן ומשמש כמפתח הבידוד.

טבלה	תפקיד ויחסים עיקריים
`companies`	השוכר. שורש ההפרדה הלוגית של כל הנתונים.
`users`	משתמש עם תפקיד, סטטוס אישור, אזור עבודה, התמחות וזמינות.
`areas`	אזור פיזי: מיקום על המפה באחוזים, קואורדינטות GPS, מצב תריס ומדידה אחרונה.
`work_areas`	אזורי עבודה מוגדרים לשיוך עובדים.

טבלה	תפקיד ויחסים עיקריים
`equipment`	נכס פיזי המשויך לאזור, עם סטטוס שירות.
`logs`	רשומת מדידה והחלטה לאורך זמן. מקור נתוני הגרפים.
`alerts`	התראה עם עדיפות; נוצרת ידנית או אוטומטית מכשל בתת-משימה.
`missions`	משימת תחזוקה: אזור, עובד, ציוד, מועד, תדירות בימים וסטטוס.
`mission_subtasks`	פריט צ׳ק-ליסט עם סטטוס, הערה וקישור לתיעוד ויזואלי.
`guides`	מדריך נוהל עם מצב אישור.
`guide_ratings`	דירוג משתמש למדריך. מקור הדירוג הממוצע.
`schedules`	כללי תזמון קבועים לאוטומציה.
`password_resets`	טוקן איפוס חד-פעמי עם חותמת תפוגה.

החלטת עיצוב. ניהול הגרסאות של הסכימה מופרד לשניים: `schema.sql` בונה מסד חדש מאפס, ומיגרציות ממוספרות מקדמות מסד קיים. ערבוב השניים הוא מקור נפוץ לאובדן נתונים בפריסה, משום שקובץ בנייה מלא פותח במחיקת המסד.

4.3 ממשקים טכנולוגיים

- 49 — **REST** נקודות קצה על פני עשרה נתבים, בעיצוב מונחה-משאבים וחסר-מצב.
- **WebSocket** — ערוץ דו-כיווני לדחיפת עדכוני מצב. הבחירה בדחיפה על פני תשאול נובעת מכך שתשאול בתדירות של 5 שניות מייצר עומס פרופורציוני למספר הלקוחות גם כשאין שינוי.
- **קליטת חיישנים** — נתיב ציבורי המקבל מדידה מחומרת שדה. הנתיב אינו דורש אסימון במכוון, שכן בקרים משובצים אינם מנהלים מחזור חיי אסימון.
- **מזג אוויר** — צריכת API חיצוני, מוגן כמתואר ב-NFR-08.
- **אחסון אובייקטים** — תמונות תיעוד נשמרות ב-S3; במסד נשמר המצביע בלבד.
- **דוא״ל** — SMTP לשליחת קישורי איפוס.
- **גשר מקורי** — Capacitor חושף מצלמה ו-GPS; שכבת עטיפה אחת נופלת חזרה ל-API של הדפדפן כשרצים בווב.

4.4 ממשק המשתמש

שני קהלים, שני דפוסי שימוש. המנהל עובד מול מסך רחב וסורק מידע; העובד עובד מול טלפון, לעיתים בעמידה, ומבצע פעולה אחת בכל רגע. הממשק בנוי כרספונסיבי אחד ולא כשני מוצרים, אך סדר העדיפויות משתנה: במסך צר, פעולת הצ׳ק-ליסט והמצלמה נמצאות בהישג אגודל.

המסכים: כניסה · הרשמה · איפוס סיסמה · מפת קמפוס · לוח אזור · גרפים · משימות · פרטי משימה · בסיס ידע · ציוד · דוחות · התראות · ניהול משתמשים · תזמון.

5 אלגוריתמים ובסיס מדעי

הבסיס המדעי של הפרויקט נשען על קורס **אלגוריתמים ומבני נתונים**: נרמול וקוונטיזציה של ערכים רציפים, בחירת מבנה נתונים לפי סיבוכיות הפעולה הנדרשת, הרכבת יחס סדר ממספר קריטריונים, וניתוח סיבוכיות זמן.

5.1 אלגוריתם ההחלטה

הבעיה: להמיר שתי מדידות פיזיקליות בעלות יחידות שונות — מעלות צלזיוס ולוקס — להחלטה בדידה אחת. הפתרון הוא שלושה שלבים.

שלב א׳ — כללי חירום

שלושה תנאי סף נבדקים לפני כל חישוב, ומחזירים סגירה מלאה מיידית:

```
if (condition ∈ {Storm, Rain, Heavy Rain})  → CLOSE (1.0)
if (T ≥ 35°C)                               → CLOSE (1.0)
if (T ≤ 10°C)                               → CLOSE (1.0)
```

קור קיצוני גורם לסגירה מאותו טעם כמו חום קיצוני: התריס משמש כשכבת בידוד ומצמצם איבוד חום. זהו כלל תחום, לא חריגה מהאלגוריתם.

שלב ב׳ — נרמול וממוצע משוקלל

כל מדידה מנורמלת לתחום [0,1] מול טווח התייחסות, ואז משוקללת. הנרמול הוא מה שמאפשר לחבר גדלים שאינם ברי-השוואה:

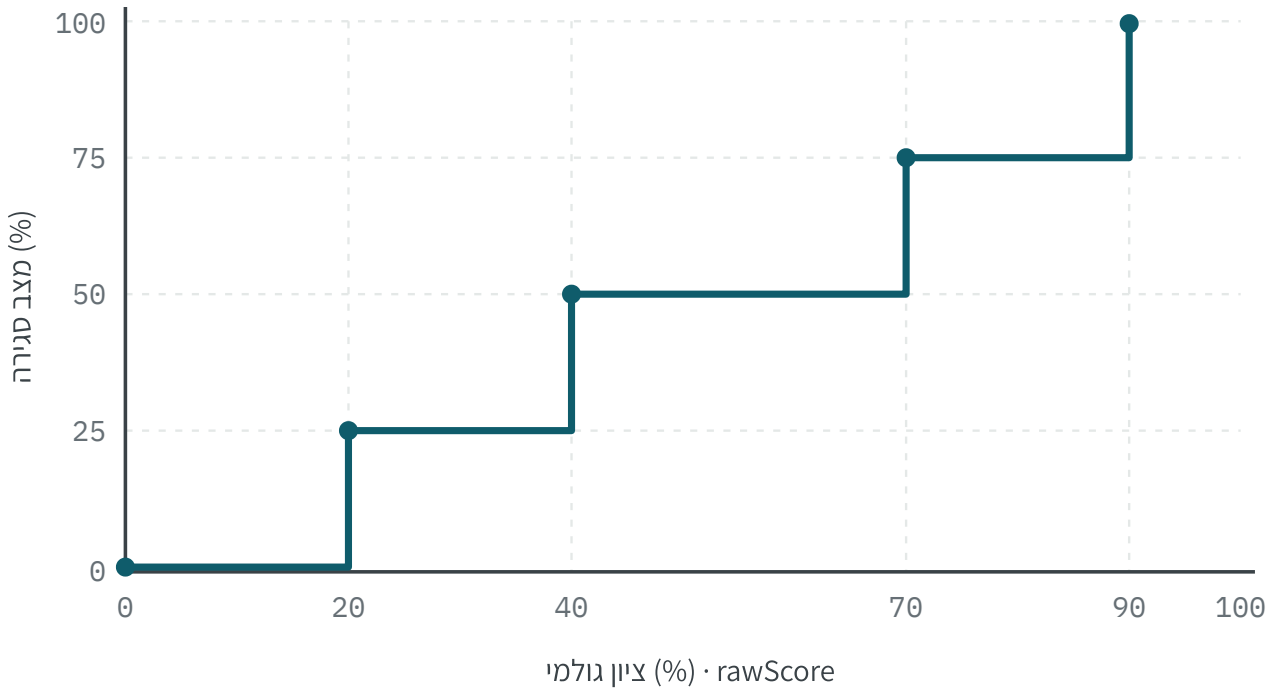
```
normT = clamp( (T − 20) / (35 − 20), 0, 1 )
normL = clamp( L / 80000, 0, 1 )

rawScore = 0.6 · normT + 0.4 · normL
```

הטווח 20–35°C נבחר כתחום שבו הטמפרטורה הופכת ממנעימה למטרידה; מתחת ל-20°C תרומת החום לציון היא אפס. הקבוע 80,000 לוקס מייצג שמיים בהירים בשיא היום.
המשקלים 0.6 ו-0.4 מבטאים החלטת תחום: **עומס תרמי מטריד יותר מסנוור**, אך סנוור לבדו מספיק כדי להצדיק סגירה חלקית.

שלב ג׳ — קוונטיזציה

הציון הרציף מוצמד לאחד מחמישה מצבים, משום שתריס פיזי אינו נע באחוז בודד ואינו אמור לנוע ברזולוציה שהעין אינה מבחינה בה:



איור 3. פונקציית ההצמדה. הספים אינם אחידים: הרצועה 40–70 היא הרחבה ביותר, כלומר סגירה חצי היא ברירת המחדל לתנאי ביניים, בעוד סגירה מלאה שמורה לציון מעל 90 בלבד.

5.2 היסטרזיס — מניעת ריצוד

ערך נמדד המתנודד סביב סף גורם למערכת תגובה נאיבית לעבור הלוך ושוב בכל מחזור. בבקרה זו תופעה ידועה (chattering), והיא שוחקת חומרה, מייצרת רעש ביומן, ומפריעה למשתמשים. הפתרון הוא אזור מת סביב הערך הנוכחי:

```
diff = |targetPosition − currentPosition|

if (diff < 5 && state == newState && reason == prevReason)
        → skip the cycle entirely (no write, no broadcast)
```

שלושת התנאים נדרשים **יחד**, וזו נקודה עדינה: אילו נבדק רק ההפרש המספרי, מעבר למצב חירום שאינו משנה את המיקום — למשל כניסת גשם כשהתריס כבר סגור — היה נבלע. הבדיקה המשולשת מבטיחה שכל שינוי במצב או בסיבה יעבור, גם אם התנועה אפסית.

5.3 דירוג עובדים לשיבוץ

בעת שיבוץ, המערכת מסדרת את העובדים לפי שלושה קריטריונים בסדר עדיפות יורד: התאמת אזור, זמינות, ואז עומס פתוח. זהו יחס סדר מורכב שנבנה כשרשרת השוואות — הקריטריון הבא נבחן רק כאשר הקודם שווה.

החלק המעניין הוא **התאמת האזור**. הניסיון הראשון היה השוואת תת-מחרוזת, והוא נכשל: אזור העבודה `"Building 216"` לא זוהה כמתאים לחדר `"Classroom 216"`, למרות שמדובר באותו מקום. הפתרון הוא טוקניזציה:

```
tokenize(s)  = lowercase → split on non-alphanumeric
             → drop tokens of length < 2
             → drop generic words {building, room, floor, area, zone, ...}

worksInArea(area, room):
    A ← tokenize(area)
    R ← new Set( tokenize(room) )     // hash set
    return ∃ t ∈ A such that t ∈ R
```

`"Building 216"` ו- `"Classroom 216"` חולקים את הטוקן `216` ולכן מתאימים. סינון המילים הגנריות הוא שמונע התאמות שווא — בלעדיו, **כל** חדר בבניין כלשהו היה מתאים לכל עובד שרשום ל"בניין" כלשהו.

בחירת מבנה הנתונים: טוקני החדר נטענים ל- `Set` ולא נשארים כמערך. בדיקת שייכות למערך היא O(|R|) ומייצרת O(|A|·|R|) לכל השוואה; ה-Set מוריד את הבדיקה ל-O(1) ממוצע ואת ההשוואה כולה ל-O(|A| + |R|).

5.4 עמידות: השהיה מעריכית ומטמון

שירות מזג האוויר החיצוני אינו אמין, ומחזור ההערכה תלוי בו. שלוש שכבות מגנות עליו:

1. **ניסיון חוזר עם השהיה מעריכית.** ההשהיה גדלה בין ניסיונות. הרציונל הוא שכשל בשירות נובע לרוב מעומס רגעי, וניסיונות חוזרים בקצב קבוע מחריפים אותו.
2. **דילוג על 4xx.** שגיאת לקוח — מפתח שגוי, פרמטר לא חוקי — לא תיפתר בניסיון חוזר. זיהויה חוסך את מלוא זמן ההשהיה.
3. **מטמון עם תוקף.** הערך התקין האחרון נשמר ומשמש כגיבוי. מדידת מזג אוויר בת דקה אינה מדויקת פחות באופן משמעותי, ועדיף להשתמש בה מאשר להפסיק את מחזור ההערכה.

5.5 סיבוכיות

פעולה	סיבוכיות	הערה
חישוב החלטה לאזור בודד	O(1)	מספר קבוע של פעולות חשבון והשוואות
הצמדה למדרגה	O(1)	ארבע השוואות לכל היותר
מחזור הערכה מלא	O(n)	n = מספר האזורים
טוקניזציה	O(L)	L = אורך המחרוזת
בדיקת התאמת אזור	O(|A| + |R|)	בזכות ה-Set; היה O(|A|·|R|) עם מערך

פעולה	סיבוכיות	הערה
מיון מועמדים לשיבוץ	`O(k log k · L)`	k = מועמדים. ראו הערה למטה
שליפה לפי מפתח מאונדקס	`O(log n)`	עץ B+ של המנוע

> **שיפור ידוע.** במיון המועמדים, פונקציית ההשוואה מבצעת טוקניזציה מחדש בכל קריאה, ולכן הגורם L מוכפל ב-k log k. חישוב מוקדם של דגל ההתאמה פעם אחת לכל מועמד (decorate–sort–undecorate) יוריד זאת ל-O(k·L + k log k). בהיקף הנוכחי — עשרות עובדים — ההבדל אינו מורגש, ולכן השיפור לא בוצע; הוא נרשם כהמלצה בסעיף 8.

5.6 דפוסים ארכיטקטוניים

- **הפרדת שכבות** עם תלות חד-כיוונית, כמתואר בסעיף 4.1.
- **פונקציה טהורה** לליבת ההחלטה: אותו קלט מחזיר תמיד אותו פלט, ללא תופעות לוואי. זהו התנאי שהופך את האלגוריתם לבר-בדיקה בלי תשתית.
- **REST חסר-מצב** — מצב הסשן נישא באסימון, ולכן ניתן להריץ מספר עותקים של השרת.
- **פרסום-מנוי** ב-WebSocket להפצת עדכונים.
- **ריבוי שוכרים בסכימה משותפת** — עמודת מפתח אחת בכל טבלה במקום מסד נפרד לכל לקוח.

6 בדיקות ואימות

6.1 אסטרטגיה

הבדיקות בנויות כפירמידה: בסיס רחב של בדיקות יחידה מהירות על הלוגיקה הטהורה, ומעליו בדיקות אינטגרציה שמפעילות את שכבת ה-HTTP מקצה לקצה מול מסד ממוקק. ההמקאה של המסד היא החלטה מכוונת — היא מאפשרת להריץ את כל הסוויטה בסביבת אינטגרציה רציפה נקייה, בלי להרים MySQL ובלי תלות במצב נתונים.

6.2 הסוויטות

סוויטה	כלים	מה נבדק
`decisionEngine.test.js`	Jest	ליבת האלגוריתם: כללי החירום, הנרמול, גבולות ההצמדה, וערכי קצה משני צדי כל סף.
`weatherService.test.js`	Jest	ניסיון חוזר, דילוג על 4xx, ונפילה למטמון בכשל.
`api.integration.test.js`	Jest · Supertest	אימות, הרשאות לפי תפקיד, ניהול אזורים והתראות.
`missions.integration.test.js`	Jest · Supertest	מחזור חיי משימה מלא: יצירה, צ׳ק-ליסט, סגירה ויצירת המופע הבא, וכשל המייצר התראה.
`signup.equipment.reports.test.js`	Jest · Supertest	הרשמה ואישור, ניהול ציוד, דוחות, ובידוד בין חברות.
`staffRanking.test.js`	Vitest	טוקניזציה, סינון מילים גנריות, וסדר הדירוג בשלושת הקריטריונים.

סך הכול: 101 בדיקות אוטומטיות — 89 בשרת ו-12 בלקוח.

6.3 אינטגרציה רציפה

כל דחיפה ובקשת מיזוג מפעילות שתי משימות מקבילות: סוויטת השרת, וסוויטת הלקוח יחד עם בנייה מלאה. הבנייה נכללת במכוון — היא תופסת שגיאות ייבוא ותחביר שבדיקות יחידה על מודולים בודדים אינן נוגעות בהן.

6.4 אימות ידני

בנוסף לאוטומציה בוצע מסלול אימות ידני מלא מול המערכת החיה, ובו כל אחד מ-30 הדרישות הפונקציונליות. המסלול איתר פגמים שהבדיקות האוטומטיות לא כיסו — בעיקר בשכבת התצוגה ובאינטגרציה עם יכולות המכשיר. הבולט שבהם היה כשל בדירוג העובדים (סעיף 5.3), שהתגלה רק בשימוש בפועל והביא לכתיבת המודול המשותף והבדיקות שלו.

> **מסקנה מתודולוגית.** כמעט כל תקלה משמעותית בפרויקט התגלתה מבדיקת **התוצאה**, ולא מכך שפקודה דיווחה על הצלחה. פקודה שהסתיימה בקוד 0 אינה עדות לכך שהיא עשתה את מה שהתכוונו.

7 סקירת הפתרון והמימוש

7.1 היקף

רכיב	היקף
נקודות קצה REST	49
בקרים · שירותים	10 · 5
טבלאות במסד	13
רכיבי לקוח	21
בדיקות אוטומטיות	101
שפות ממשק	אנגלית, עברית (RTL מלא)
פלטפורמות	דפדפן, אנדרואיד

7.2 מחסנית טכנולוגית

לקוח: React 19 · Vite · Socket.io client · Recharts · Axios · Capacitor 8.
שרת: Node.js · Express 5 · Socket.io · node-cron · JWT · bcrypt · Multer · Nodemailer.
נתונים ותשתית: MySQL 8 · Docker · AWS EC2 · AWS S3 · Vercel.
איכות: Jest · Supertest · Vitest · GitHub Actions.

7.3 פריסה

השרת ומסד הנתונים ארוזים בקונטיינרים ורצים על מכונת EC2; הלקוח נבנה סטטית ומוגש מ-Vercel; תמונות התיעוד נשמרות ב-S3. התעבורה בין הלקוח לשרת עוברת ב-HTTPS. הפריסה אינה משתמשת בשירות מנוהל ייחודי לספק, ולכן ניתנת להעברה לספק אחר בהחלפת תצורה בלבד.

7.4 מה נמסר מעבר לדרישה

מסמך הייזום הגדיר מערכת לניהול משימות תחזוקה. המימוש מספק אותה במלואה, ומרחיב בשלושה כיוונים שהמסמך מבקש אך אינו מגדיר: **בקרה בזמן אמת** על הציוד עצמו, **סגירת הלולאה** בין כשל בשטח לעבודה מתוזמנת, ו**ריבוי שוכרים** שהופך את המערכת ממופע יחיד למוצר. חלק ניכר מהקוד קיים לא כדי שהמערכת תעבוד, אלא כדי שתהיה ניתנת להפעלה: פריסה, בידוד, עמידות בפני שירות חיצוני נופל, וסוויטת בדיקות שהופכת שינוי לבטוח.

8 המלצות להמשך פיתוח

8.1 חיזוי תקלות

המערכת כבר אוגרת את חומר הגלם: היסטוריית מדידות, סטטוס ציוד, ורישום של כל תת-משימה שנכשלה עם הסבר מילולי. על בסיס זה ניתן לבנות מודל שמדרג ציוד לפי הסתברות כשל ומקדים משימת תחזוקה לפני התקלה במקום אחריה. זהו המעבר מתחזוקה מונעת מבוססת-לוח-זמנים לתחזוקה חזויה מבוססת-נתונים, וזו התרומה המשמעותית ביותר שניתן להוסיף.

8.2 חיבור לחומרה אמיתית

כיום החיישנים מיוצגים בסימולטור הכותב לנתיב הציבורי. חיבור לבקרי מבנה בפרוטוקול תעשייתי — BACnet או Modbus — יחליף את הסימולטור בלי לשנות את שכבת השירותים, משום שגבול המערכת כבר מוגדר בנקודת הקצה הזו.

8.3 עבודה ללא קליטה

עובד בחדר מכונות או במרתף עלול לאבד קליטה בדיוק בזמן הביצוע. אחסון מקומי של המשימות וסנכרון מאוחר, עם פתרון התנגשויות, יהפוך את האפליקציה לשמישה בכל מקום במתקן.

8.4 שיפורים ממוקדים

- ייעול מיון המועמדים לשיבוץ באמצעות חישוב מוקדם, כמתואר בסעיף 5.5.
- הפיכת משקלי האלגוריתם לניתנים לכיוונון לכל אזור, כדי להתאים לכיווני חזית שונים.
- התראות דחיפה למכשיר במקום הסתמכות על פתיחת האפליקציה.
- חתימה דיגיטלית על ביצוע משימה, לצרכים רגולטוריים.

8.5 סיכום

הפרויקט הגדיר בעיה בעלת שני צדדים — חוסר הכוונה בשטח וחוסר נראות בחדר הבקרה — וסיפק מערכת אחת שמטפלת בשניהם ומחברת ביניהם. מה שהחל כדרישה למעקב אחר משימות הפך למערכת שגם מבצעת עבודת תחזוקה וגם יוצרת אותה.

9 מקורות

9.1 ספרות מקצועית

מקור	שימוש במסמך
Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. — **Introduction to Algorithms**	ניתוח סיבוכיות זמן, בחירת מבנה נתונים לפי סיבוכיות הפעולה, ויחסי סדר מורכבים — סעיפים 5.3 ו-5.5.
Fielding, R. T. (2000) — **Architectural Styles and the Design of Network-based Software Architectures**	עקרונות REST והמשמעות של ממשק חסר-מצב — סעיפים 4.3 ו-5.6.
Codd, E. F. (1970) — **A Relational Model of Data for Large Shared Data Banks**	המודל היחסי והנרמול שבבסיס סכימת 13 הטבלאות — סעיף 4.2.
OWASP — **Top 10 Web Application Security Risks**	שאילתות מפורמטרות, אחסון סיסמאות, ומניעת מניית משתמשים — סעיף 3 (NFR-03 עד NFR-06).

9.2 תקנים ופרוטוקולים

תקן	רלוונטיות
ANSI/ASHRAE Standard 135 — **BACnet**	פרוטוקול התקשורת המקובל לבקרת מבנה; יעד האינטגרציה המומלץ בסעיף 8.2.
Modbus Application Protocol Specification — Modbus Organization	פרוטוקול תעשייתי חלופי לחיבור בקרים.
ISO/IEC 14543-3 — **KNX**	תקן אירופי לאוטומציית מבנה.
EN 17037 — **Daylight in Buildings**	קריטריונים לתאורת יום ולבקרת סנוור; הרקע התקני לאיזון שבין טמפרטורה לעוצמת אור בסעיף 5.1.

9.3 מערכות מסחריות שנסקרו

ההשוואה שבבסיס סעיף 1.1 נסמכת על תיעוד המוצר הרשמי של היצרנים: **מערכות ניהול תחזוקה** — IBM Maximo Application Suite, SAP Plant Maintenance, Fiix, UpKeep, Limble. **מערכות בקרת מבנה** — Siemens Desigo CC, Johnson Controls Metasys, Honeywell EBI. **בקרת הצללה ייעודית** — Somfy animeo, Warema, Lutron. **פלטפורמות קוד פתוח** — Home Assistant, openHAB, Node-RED.

יכולות המוצרים המסחריים משתנות בין גרסאות ובין מהדורות רישוי. ההשוואה מתייחסת ליכולות הליבה כפי שהן מתוארות בתיעוד הרשמי, ולא לתצורה ספציפית אצל לקוח.