

אפיון מערכת CaregiveFlow

תאריך: 19/09/2025

מטרה: מערכת לניהול סוכנויות והשמות בתחום הסיעוד בישראל - ניהול עובדים זרים, מעסיקים, שיבוצים, דרכונים, ויזות, ביטוחים, דמי תאגיד, היתרי העסקה, ודוחות/ייצוא רגולטוריים (BAFI/PIBA).

עקרונות מנחים: - לוגיקה עסקית בצד ה-PHP; ה-DB משמש **אך ורק** לאחסון/שליפה/עדכון/מחיקה (+ אינדקסים לביצועים). - שמירה על סגנון קוד קיים בפרויקט MVC, תבניות views, עזרי `helpers.php`, שימוש ב-PDO וכו'. - RTL מלא; טקסטים בעברית; עקביות UI/UX.

(1) ארכיטקטורה וטכנולוגיות

- **Stack:** PHP 8.x, MariaDB/MySQL, Apache/XAMPP, Bootstrap 5 (RTL), Font Awesome, Vanilla JS.
- **מבנה פרויקט (תקני):**
- `public/` - נקודת כניסה (`index.php`), קבצי CSS/JS/Assets.
- `src/controllers/` - Controllers לפי מודולים.
- `src/models/` - Models (ORM ידני/שכבת DAO ב-PDO).
- `src/services/` - שירותים (Business Services, Builders לייצוא קבצים).
- `views/<module>/` - תצוגות (`index.php`, `form.php`, `show.php`, `view.php`, `partials`).
- `sql/` - סקריפטים/דאמפים.
- `docs/` - מסמכי אפיון/מיפויים.
- **Routing:** פרמטר `r` ב-GET (לדוג' `?r=employees/index`).
- **DB Access:** שכבה דקה ב-PDO; כל השאילות פרמטריות; ללא לוגיקה ב-DB מלבד אינדקסים/מפתחות זרים.

(2) מודולים עיקריים

להלן המודולים כפי שהוגדרו/בפיתוח, עם עקרונות שדות ומסכים. השדות מתייחסים ל-DB הקיים (ע"פ `caregiveflow.sql`) בתוספת שדות עומק ל-UI.

2.1 עובדים (Employees)

- **מסכים:** רשימה, יצירה/עריכה, כרטיס עובד מרובה טאבים, תצוגה.
- **שדות ליבה (דוגמה):** `id`, `first_name_he`, `last_name_he`, `gender`, `date_of_birth`, `country_code`, `phone`, `email`, `address_city_code`, `address_street`, `is_active`.
- ועוד.
- **תכונות:**
- טאבים בכרטיס: פרטי עובד | דרכונים | ויזות | ביטוחים | שיבוצים/מעסיקים.
- עמודת "מעסיק נוכחי" ברשימה.
- צביעה ירוק/אדום לתאריכי תוקף (see Passports/Visas/Insurance).
- פעולה בשורה: שיבוץ למעסיק (פותח `placements/form` עם עובד מסומן).

2.2 מעסיקים (Employers)

- **שדות ליבה:** id, employer_name, id_number, phone, email, address_city_code, is_active, ועוד.
- **תכונות:** דרכונים למעסיק (כמו לעובד), דמי תאגיד, היתרי העסקה.

2.3 שיבוצים (Placements)

- קשר N:1 בין עובד-מעסיק לאורך זמן.
- **שדות:** id, employee_id, employer_id, start_date, end_date, (nullable), status_code, notes.
- **כללים:** שיבוץ נוכחי = end_date IS NULL או מאוחר מהיום.

2.4 דרכונים (Passports)

- **לשני ישויות:** Employees ו-Employers.
- **שדות:** id, owner_type, ('employee'/employer'), owner_id, passport_number, is_primary, expiry_date, issue_date, country_code, passport_type_code, status_code, בטבלה).
- **UI:** רשימה/טופס/תצוגה אחד-לאחד בסגנון קיים; צביעת expiry_date (אדום אם פג, ירוק אם תקף).

2.5 ויזות (Visas)

- **שדות מוצעים:** id, employee_id, visa_type_code, reference_number, submit_date, (תאריך הגשה), issue_date, expiry_date, status_code, notes, (מספר סימוכין).
- **UI:** מבנה זהה לדרכונים (אייקונים/partial/קוד).

2.6 ביטוחים רפואיים (Insurances)

- **שדות מוצעים:** id, employee_id, provider_name, policy_number, coverage_type, expiry_date, issue_date, status_code, notes, (ערכים בעברית: "בריאות בסיסי", "בריאות מלא", "תאונות", "נסיעות", "אחר").
- **UI:** זהה למודול ויזות/דרכונים (טבלאות, אייקונים, partials).

2.7 דמי תאגיד (Employer Fees)

- **מטרה:** מעקב גבייה חודשי לפי מעסיק.
- **שדות:** id, employer_id, period_year, period_month, amount, due_date, paid_date, (nullable), status_code, ('שולם'/'לא שולם'), notes.
- **דוחות:** ספירת מעסיקים שלא שילמו עבור תקופה נתונה (קריטריון לפי חודש/שנה).

2.8 היתרי העסקה למעסיק (Employer Permits)

- **שדות:** id, employer_id, permit_type_code, status_code, issue_date, expiry_date, notes.
- **UI:** כמו דרכונים/ביטוחים; הצגת status ו"סוג היתר" ללא קווים ריקים.

2.9 מערכת קבצים (Files)

- **מטרה:** צירוף מסמכים לכל מודול.
- **טבלאות:** `files` + טבלת קוד לסוגי קובץ (ערכים ראשוניים: דרכון, ת.ז, אסמכתא תשלום דמי תאגיד, היתר מעסיק, ויזה, מסמכי ביטוח).
- **UI:** בכל `edit.php` / `form.php` של מודול – אזור העלאה + בחירת "סוג קובץ" מהרשימה ב-DB; בתצוגה `show.php` – רשימת קבצים.

(3) ייצוא רגולטורי ודוחות

3.1 BAFI (מת"ש)

- **Builder Service:** `services/BafiFileBuilder.php` (בדומה ל-`PibaXmlBuilder.php`).
- **פורמט:** קובץ טקסט רוחב-קבוע (Fixed Width) עם שורת בקרה + רשומות עובדים.
- **חישובים:** "סה"כ רשומות בקובץ" מחושב בצד ה-PHP (לא מתוך DB), לא כולל שורת הבקרה.
- **מיפוי:** מתוחזק ב-Excel (עמודות: Field Name, Position, Type [N/EN/HE], Length, Alignment [L/R], Table, Column, Transform/Format [e.g. `zfill`], `YYYYMMDD`], הערות).
- **UI:** מודל ייצוא עם פרמטרים (record type / mana type / status code וכו'), היסטוריית ייצוא עם קישור להורדה.

3.2 PIBA (XML)

- **Builder Service:** `services/PibaXmlBuilder.php`.
- **ולידציות:** פורמט תאריכים, קודים תקינים, סכימות (XSD אם זמין / אימות מבני).

3.3 דוחות נוספים

- **Exports History:** מסך היסטוריה (Pagination, סיכומים, יחס הצלחה, טיפול בשגיאות).
- **Health-Checks:** `tools/healthcheck.php` + SQL לבדיקות עקביות (ID ריק, תאריכי תוקף חריגים, רשומות בלי קוד סוג וכו').

UI/UX (4)

- RTL, Bootstrap 5, עקביות אייקונים (Font Awesome).
- **כרטיס עובד חדש** עם טאבים (ללא שבירת קוד קיים).
- **טבלאות:** עיצוב אחיד, צבעי תוקף אדום/ירוק, כפתורי פעולה עם Tooltips אופציונליים.
- **ניווט:** כפתור "חזרה לעובד" בתתי-מסכים.
- **מודלים:** ייצוא BAFI, העלאת קבצים.
- **הנחיות RTL:** יישור שדות זה לצד זה (`col` / `row`), לא טור אחד בלבד (תיקון ב-`views/employees/edit.php`).

(5) אבטחה והרשאות

- ששן מאובטח; התחברות + תפקידים (מנהל/רכז/צפייה).

- ולידציה דו-שלבית בצד שרת + צד לקוח; בריחה (e()) בכל יציאה ל-HTML.
- מגבלות העלאת קבצים: סוגי mime, גודל, שמות קבצים יוניקיים, שמירה מחוץ ל-`public` או עם hash + הגנת גישה.

(6) ביצועים ואינדקסים (DB-Only)

- אינדקסים מוצעים:
- `employees(is_active)`, `employees(country_code)`
- `employee_passports(owner_id, owner_type, is_primary)`
- `employee_passports(expiry_date)`
- `visas(employee_id, expiry_date, status_code)`
- `insurances(employee_id, expiry_date, status_code)`
- `placements(employee_id, employer_id, start_date, end_date)`
- `employer_fees(employer_id, period_year, period_month, paid_date)`
- `files(owner_type, owner_id, file_type_code)`

(7) כללי פיתוח וקוד

- **הצמדות לסגנון קיים:** מבנה תיקיות, שמות קבצים/פונקציות, הערות דו-לשוניות קצרות בקוד.
- **Controllers:** שימוש ב-`Model::find()`, `Model::count()`, `Model::all()`; בלי לפגוע בפונקציות קיימות.
- **Views:** שמירה על partials קיימים; שימוש ב-helpers לרכיבים חוזרים; טיפול ב-old values בעת שגיאה; שמירת בחירות ב-`<select>`.

(8) מיפוי נתונים (תקציר)

- **קודי מדינות/ערים/סטטוסים** – מטבלאות קוד קיימות בפרויקט (לאומיות משתמשת ב-`countries`).
- **פורמטים:**
- תאריכים UI: `DD/MM/YYYY`
- תאריכים ייצוא: `YYYYMMDD`
- מספרים קבועי רוחב: `zfill(length)`
- **ניקוי וטרנספורמציות:** `trim, upper/normalize` ל-EN; בדיקה ל-HE.

(9) Roadmap קצר (המשך)

1. מערכת קבצים כללית לכל המודולים (DB + UI ב-`employees`, אח"כ הרחבה לשאר).
2. כרטיס עובד עם טאבים – מיגרציה הדרגתית, שמירת קוד קיים.
3. **Employer Fees** – מסך סיכום חודשי: "מעסיקים שלא שילמו" לפי תקופה.
4. **Employer Permits** – השלמת UI אחד-לאחד (תיקון שדות ריקים).
5. **BAFI Builder ב-PHP** – העברת החישובים משכבת DB לשירות; תיקון "סה"כ רשומות".
6. **PIBA XML** – אימות מבני; שמירת היסטוריית קבצים.

10) נספח: עקרונות בדיקות

- Unit (Services: Builders, Validators)
- Integration (Controllers + Models + Views)
- נתוני דמה ב-SQL ל-seed.

11) הצעות לשיפור העבודה המשותפת

- להשתמש ב-Canvas עבור **קבצי קוד/מסמכים ארוכים**, קובץ לכל רכיב (Controller/Model/View/Service) כדי לאפשר הורדה/השוואה.
- לכל Patch: לציין **קבצים מושפעים**, סיכום שינוי, והדגשות אחד-לאחד מול המקור.
- לשמור **Excel Mapping** יחיד כמקור-אמת ל-BAFI/PIBA; לשלב עמודת "Implemented?".
- להפעיל **בדיקת תקינות** לפני ייצוא (כפתור "בדוק לפני ייצוא").
- לשמור **מסמכי מסכים** (Screenspec) קצרים לכל מודול (שדה, מקור, ולידציה, הודעות שגיאה).

סיום

המסמך מסכם את האפיון הנוכחי של CaregiveFlow בהתאם לשיחות והקבצים העדכניים. במידת הצורך אעדכן כאן סעיפים נקודתיים (שדות/מסכים/חוקים) לפי קבצי הפרויקט ששלחת.

<div/>