**1. מבוא**

**1.1. מטרה**

מסמך זה מתאר את הדרישות הפונקציונליות והלא פונקציונליות של מערכת "ZIM OCR" – מערכת לניהול מסמכי שילוח שמטרתה:

* לאפשר העלאת קבצי PDF או תמונות של מסמכי שילוח.
* לבצע עיבוד OCR על הקבצים כדי לחלץ את המידע החיוני.
* לאחסן את הנתונים במסד נתונים (PostgreSQL).
* להציג את הנתונים בדשבורד הכולל גרפים, סטטיסטיקות וטבלאות.

**1.2. היקף המערכת**

המערכת מיועדת לחברת שילוח (כגון חברת צים) לצורך ניהול, בקרה ואוטומציה של תהליך עיבוד מסמכי שילוח, כאשר המערכת:

* תאפשר העלאה ועיבוד אוטומטי של קבצים.
* תחלץ נתונים מרכזיים (כגון מספר שטר, מכולה, תאריכים, נתוני מטען, פרטי צדדים ותנאי משלוח).
* תציג דשבורד עם נתונים סטטיסטיים, גרפים והתראות.
* תעבוד במבנה MVC כאשר ה־View יהיה ב־React וה־Backend יתבסס על Spring Boot או Node.js.

**1.3. קהל יעד**

מסמך זה מיועד לצוות הפיתוח, המנהלים הטכניים והמעצבים, וכן לכל גורם מעורב בתהליך הפיתוח והפריסה של המערכת.

**1.4. הגדרות ומונחים**

* **OCR:** Optical Character Recognition – טכנולוגיה להמרת תמונות לטקסט.
* **Bill of Lading:** שטר שילוח, מסמך משפטי שמייצג הסכם בין המוסר לנמען.
* **MVP:** Minimum Viable Product – מוצר מינימלי לעמידה בדרישות הבסיסיות.
* **JWT:** JSON Web Token – אסימון לצורך אימות וביטחון בקריאות API.

**2. תיאור כללי של המערכת**

**2.1. סקירה**

המערכת תעבד קבצים (PDF/תמונות) של מסמכי שילוח, תפעיל OCR לחילוץ נתונים, תעדכן את הנתונים במסד נתונים, ותציג דשבורד מקיף עם סטטיסטיקות, גרפים והתראות.

**2.2. תכונות עיקריות**

* **העלאת קבצים:** ממשק משתמש המאפשר העלאת קבצים בצורה קלה ונוחה.
* **עיבוד OCR:** הפעלת מנוע OCR (כגון Tess4J) לחילוץ טקסט מהקובץ.
* **חילוץ נתונים ואימותם:** הפעלת אלגוריתמים (למשל, regex או NLP בסיסי) לחילוץ ואימות נתונים.
* **שמירה במסד נתונים:** אחסון הנתונים במאגר PostgreSQL.
* **דשבורד:** תצוגת גרפים, סטטיסטיקות וטבלאות המציגות את הנתונים בממשק משתמש אינטואיטיבי.
* **אבטחה:** שימוש ב־JWT לאימות קריאות API.

**3. דרישות פונקציונליות**

**3.1. העלאת מסמך ועיבוד OCR**

* **3.1.1.** המשתמש יוכל להעלות קובץ PDF או תמונה דרך ממשק משתמש (Upload Modal).
* **3.1.2.** המערכת תקבל את הקובץ ותפעיל עליו מנוע OCR.
* **3.1.3.** המערכת תחלץ נתונים רלוונטיים מהקובץ (לדוגמה: מספר שטר, מספר מכולה, שם הספינה, תאריכים, נמלים, נתוני מטען, פרטי צדדים, תנאי משלוח).
* **3.1.4.** הנתונים החולצו יאוחסנו במסד הנתונים.

**3.2. הצגת נתונים בדשבורד**

* **3.2.1.** המערכת תציג דשבורד עם נתונים סטטיסטיים כגון:
  + סך כל המסמכים.
  + חלוקה לפי סטטוס (Approved, Pending, Rejected).
  + ממוצע דירוג OCR.
* **3.2.2.** המערכת תציג טבלה עם המסמכים האחרונים (כולל מזהה, סטטוס, תאריך העלאה, OCR, נתוני מטען וכו').
* **3.2.3.** המערכת תציג גרפים (כמו גרף מגמה, גרף חלוקה לפי נמלים) להצגת הנתונים באופן ויזואלי.
* **3.2.4.** המערכת תציג התראות אם יש חריגות (כגון תאריכים לא תואמים, דירוג OCR נמוך).

**3.3. API לאבטחה והעברת נתונים**

* **3.3.1.** כל קריאת API תכלול אסימון JWT לבדיקת זהות ואימות.
* **3.3.2.** קריאות API יבוצעו בפורמט JSON.

**4. דרישות לא פונקציונליות**

**4.1. ביצועים**

* המערכת צריכה לעבד מסמכים תוך 5 שניות בממוצע (כולל OCR וחילוץ נתונים).

**4.2. זמינות ואמינות**

* המערכת צריכה להיות זמינה 99.9% מזמן פעולתה.
* יש לבצע רישום לוגים ואבחון שגיאות.

**4.3. אבטחה**

* יש להשתמש ב־JWT לכל קריאת API.
* המערכת תבצע ולידציה של קלטים למניעת מתקפות (SQL Injection, XSS וכו').

**4.4. רספונסיביות**

* ממשק המשתמש יהיה רספונסיבי, עם התאמה למובייל ולדסקטופ.

**4.5. פריסה וקונטיינריזציה**

* המערכת תיושם בצורה שניתן לארוז אותה בקונטיינרים (Docker) ולהריץ אותה בסביבת ענן או מקומית.

**5. תרשימי זרימה ולוגיקת עסקים (Flowcharts)**

**5.1. תהליך העלאת מסמך ועיבוד OCR**

**5.2. פירוק מודולים:**

* **Document Controller:**
  + מקבל בקשת העלאה.
  + קורא ל־File Upload, OCR, Data Extraction, Data Validation.
  + מעדכן את מסד הנתונים.
  + מחזיר תגובה ללקוח עם נתונים.
* **Dashboard Controller:**
  + אחראי על שליפת נתוני הדשבורד מהמסד.
  + מחשב סטטיסטיקות ומאגד נתונים.
  + מחזיר תגובת JSON ל־Frontend.

**6. סיכום**

**Model:**  
טבלת Document ב־PostgreSQL, עם כל השדות שהוגדרו.

**API Contracts:**

* POST /api/documents – לקבלת קובץ, עיבוד OCR, חילוץ ואימות נתונים.
* GET /api/dashboard – שליפת נתוני דשבורד (סטטיסטיקות, מסמכים, לוגיסטיקה).

**Controller & Business Logic:**

* Document Controller: מנהל את תהליך העלאת הקובץ, עיבוד OCR וחילוץ הנתונים.
* Dashboard Controller: אחראי על איסוף הנתונים להצגה בדשבורד.

**Flow Diagram:**  
מתאר את התהליך מהעלאת הקובץ ועד לעדכון הדשבורד.

**אבטחה:**  
כל קריאת API תכלול אסימון JWT לאימות.

**פריסה:**  
המערכת תיושם בצורה מקומית לצורך הדמו, עם אפשרות לאריזה בקונטיינרים (Docker).