

# גרסה 3 - רכיב שמירת נתונים לאורך זמן, וחסינות מפני נפילות - 2021

גרסה 3 של מערכת המסחר עוסקת בהעשרת המערכת ברכיב **שמירת נתונים לאורך זמן** (persistence). דרישות **החסינות והתאוששות** מחייבות את היכולת לשחזר את מצב המערכת במקרה של הפעלה יזומה מחדש וכמו כן במקרים לא רצויים כגון נפילת המערכת, איבוד תקשורת וכו'. לשם כך יש לחבר בסיס נתונים למערכת. החיבור ייעשה באמצעות רכיב מקשר בין המערכת לבסיס הנתונים באמצעות ספריית **ORM** לבחירתכם.

**תפקידי צוות:** מנהל/ת גרסה המשמש גם כבודק מטעם הלקוח, ומפתחים. תחומי האחריות – [ב-מסמך המתודולוגיה](#).

## הדרישות לגרסה 3:

### 1. דרישות רמת שירות:

#### a. דרישה 6 - שמירה לאורך זמן:

המערכת שומרת את מצבה העדכני במסד נתונים חיצוני של נתונים, באופן המאפשר הפרדה בין פעולות לנתונים. ההפרדה מאפשרת שחזור מצב מערכת וניהול יעיל. יש לתמוך במסד הנתונים מרוחק (לא באותו מחשב עליו רצה המערכת).

#### b. דרישה 5 - חסינות מפני נפילת המערכת בתרחישים לא רצויים:

- המערכת צריכה להתמודד עם אובדן תקשורת בין רכיבים, ועם מערכות חיצוניות.
- המערכת צריכה להיות **חסינה** להתנהגויות לא צפויות של מערכות חיצוניות, כגון. כישלון בשליחת בקשה, ממשק לא תואם, אי מענה לבקשה וכו'.

### 2. דרישות פונקציונאליות:

#### a. דרישה 1.1 – אתחול המערכת:

- איתחול המערכת מתבצע עלפי **קובץ קונפיגורציה** חיצוני (בפורמט כלשהו) המגדיר את הפרמטרים לאתחול. למשל, הקובץ צריך להכיל מידע אודות מסד הנתונים אליו המערכת מתחברת בעת עלייתה (וכן פרטי ההתחברות הרלוונטיים למסד הנתונים), עם אילו מערכות חיצוניות להתממשק, פרטי מנהל המערכת וכו'.
- אפשרות איתחול במצב (state) המוגדר **בקובץ "מצב" חיצוני** (קובץ נוסף): לאחר האיתחול המערכת נמצאת במצב המוגדר בקובץ. קובץ מצב מכיל הנחיות לסדרה של סיפורי שימוש וארגומנטים להפעלתם (פורמט כלשהו). בשלב האתחול המערכת תקרא את הקובץ ותפעיל את סיפורי השימוש המתוארים בו, כך שבסיום האתחול המערכת תהיה במצב הנדרש.

#### b. דרישות 7 ו-8 – חיבור למערכת תשלום ואספקה. על המערכת להתקשר עם מערכות

התשלום והאספקה החיצוניות המפורטות [במסמך הנלווה](#).

#### c. תמיכה בסוג הקניה **הגשת הצעת-קנייה**, כמפורט במסמך הדרישות הכללי.

#### d. **דרישה חדשה** עבור קבוצה של שישה סטודנטים:

- מנהל המערכת יכול לצפות בהכנסה הכוללת של המערכת בחתך יומי.

## דגשים למימוש:

1. **שמירת נתונים לאורך זמן:** יש להשתמש בספריית ORM כלשהי על מנת לקשר בין המערכת לבסיס הנתונים. **מפורט בנספח למסמך זה.**
2. **איתחול מקובץ מצב:** הסבר ודוגמה מופיעים בנספח למסמך זה.
3. **הסבר לקבצי איתחול:** דף הפרויקט ב-GitHub צריך להכיל קובץ readme, המסביר כיצד לאתחל את המערכת ומפרט את הפורמטים של קובץ הקונפיגורציה ושל קובץ מצב האיתחול.
4. **בדיקות:**
  - a. יש לכתוב **בדיקות** עבור חסיונות המערכת בהתמודדות עם המערכות החיצוניות ומסד הנתונים.
  - b. יש לכתוב **בדיקות** עבור אתחול תקין: המערכת לא עולה אם האיתחול אינו תקין.
  - c. יש להשתמש **בקובץ קונפיגורציה יעודי** לצורך הבדיקות בכדי שהבדיקות לא יפנו למערכות החיצוניות האמיתיות ועל מנת שלא "יזהמו" את מסד הנתונים המשמש את המערכת.
5. **הארכיטקטורה העדכנית של המערכת תכלול את הרכיבים הבאים:**
  - רכיב **לקוח (presentation, client)** עדכני.
  - רכיב **תקשורת (communication)** ורכיב **notifications** עדכניים.
  - רכיב **שירות (service)** עדכנית.
  - רכיב **לב-מערכת (domain)** עדכני.
  - **שכבת קישור למסד נתונים (DAL).**
  - **מסד נתונים**

## תוצרי גרסה 3

1. **מודלים עדכניים:** מילון, תרחישי שימוש, ארכיטקטורה, מחלקות.
2. **מימוש.**
3. **בדיקות:** עלפי הדרישות החדשות. עבור כל הסוגים. **לא יתקבל מימוש ללא רכיב בדיקות משמעותי!**

## מצגת כיתה: בדיקות קיבול ועומס

- הגרסה הבאה של המערכת תכלול בדיקות קיבול (load) ועומס (stress) של המערכת
- הסבר/י את הבעיה של קיבול ועומס במערכות מרובות משתמשים.
  - מהן בדיקות קיבול ועומס, מהו המידע המתקבל מבדיקות אלו וכיצד ניתן להשתמש בו בכדי לשפר את המערכת.
  - הסבר/י מדוע בדיקות יחידה ובדיקות קבלה אינן מכסות מקרים אלו
  - בחר/י כלי המאפשר הגדרה והרצה של בדיקות קיבול ועומס של המערכת.
    - הדגם/י עבודה עם הכלי.
    - הסבר/י כיצד ניתן להשתמש בכלי זה בכדי ללמוד את צווארי הבקבוק במערכת (מגבלות על משאבים שונים והגורמים להם).
  - תאר/י אוסף של תרחישי בדיקה למערכת המסחר המשמשים לצורך בדיקת קיבול ועומס.

## נספח לגרסה 3

### שמירה לאורך זמן

1. ספריית ה-ORM מהווה **Data Access Layer** המפרידה בין לב-המערכת לבסיס-הנתונים. ספריות אלו מאפשרות הגדרה של התאמה בין ישויות לב-המערכת לישויות בסיס-הנתונים, ומטפלות בשמירה ואחזור של מידע ב/מ-בסיס הנתונים. הקישור בין לב-המערכת למסד-הנתונים הוא שיכבתי: מכיוון המערכת אל הנתונים ולא להיפך. כלומר, יש לתכנן ולהתאים את השמות בספריית ה-ORM למימוש הקיים. **לא יתקבל פתרון המבוסס על תכנון עצמאי של מסד נתונים.**
2. **סימון Persistency** במודל המחלקות: יש להקפיד לא לשמור במסד הנתונים ערכים ואובייקטים שהם ארעיים במהותם, לדוגמה, אין לשמור עגלת קניות של אורחים כחלק מנתוני המערכת. לצורך הדגשת ההחלטות לגבי שמירה במסד נתונים, יש לסמן במודל המחלקות מחלקות שהאובייקטים שלהן נשמרים במסד הנתונים כ-`<persistent>`. אם יש החרגה של חלק מהאובייקטים, צריך להוסיף הערה.
3. **שמירת נתונים בזיכרון (caching):** מכיוון שתקשורת לבסיס נתונים היא יקרה (משאבי מחשוב), נהוג לשמור נתונים הנמצאים בשימוש תדיר בזיכרון המערכת. לדוגמה, כאשר משתמש מנוי מתחבר למערכת, ניתן לטעון את עגלת הקניות שלו לזיכרון המערכת עד לעזיבתו. עם זאת, שמירת נתונים בזיכרון המערכת עלולה לעורר בעיות הנובעות מכפילות נתונים וחייבים לטפל בבעיות אלו.
4. **טרנסאקציות:** בכדי לשמור על עקביות הנתונים במסד הנתונים, יש פעולות שחייבים לאגד יחד כטרנסאקציה אטומית. לדוגמה, **אי הגדרת רכישה מיידית כטרנסאקציה** עלולה לייצר תיאור לא תקין של המלאי. לכן, יש לבחון פעולות המערבות שמירה במסד הנתונים, ולבחון את הצורך לאגד אותן כטרנסאקציות אטומיות.

### איתחול מקובץ מצב

אתחול המערכת מתבצע **מול שכבת השירות** וניתן לבצע באמצעותו רק פעולות חוקיות. למשל, לא ניתן לפתוח חנות מבלי לבצע חיבור למערכת. האיתחול מאפשר את כל תרחישי השימוש של הדרישות הפונקציונליות. למשל דוגמה אפשרית עבור פתיחת חנות והגדרת מנהל:

```
guest-registration(moshe,*moshe's password*,*other required information*);
```

```
guest-registration(rina,*rina's password*, *other required information*);
```

```
login(rina, *password*);
```

```
open-shop(rina, shoes, *other required information*);
```

```
appoint-manager(rina, shoes, moshe, *other required information*);
```

יש לוודא כי כלל הפעולות חוקיות ומסתיימות בהצלחה. אם אחת מהפעולות נכשלת על תהליך העדכון להיכשל ולדווח שגיאה בהתאם.