

גרסה 3 - רכיב שמירת נתונים לאורך זמן, וחסינות מפני נפילות - 2020

תזכורת: גרסאות הפרויקט מתייחסות אל ותלויות [במסמך הדרישות הכללי](#) וב [מסמך מתודולוגיית פיתוח](#) פרויקט. כל גרסה מרחיבה או משנה את הגרסה שקדמה לה. בכל שלב יש לשמור על תאימות בין המודלים והמימוש.

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 (Data-Access-Layer (DAL) המאפשר [אבסטרקציה דו כיוונית](#).

תפקידי צוות: מנהל/ת גרסה המשמש גם כבודק מטעם הלקוח, ומפתחים. פירוט תחומי האחריות של התפקידים השונים נמצא במסמך המתודולוגיה. במיוחד יש לשים לב לתחומי האחריות של מנהל הגרסה.

דרישות נוספות עבור גרסה 3

1. דרישות רמת שירות:

a. דרישה 6 - שמירה לאורך זמן:

המערכת שומרת את מצבה העדכני במסד נתונים חיצוני של נתוני המערכת. שמירת נתונים באופן זה מאפשרת הפרדה בין פעולות המערכת לנתונים. היתרונות של הפרדה זו הן שחזור מצב המערכת וכן ניהול משאבים יעיל.

- i. **רכיב DAL:** הקישור בין המערכת (להלן **לב-המערכת**, שהוא שכבת ה-Domain) לבין מסד הנתונים מתבצע באמצעות רכיב שכבתי מקשר, DAL, המבצע את התרגום בין מושגי לב-המערכת למושגי מסד הנתונים.
- ii. **סימון Persistency:** יש להקפיד לא לשמור במסד הנתונים ערכים ואובייקטים שהם ארעיים במהותם, לדוגמה, אין לשמור עגלת קניות של אורחים כחלק מנתוני המערכת. לצורך הבהרת החלטות לגבי שמירה במסד נתונים, יש לסמן בדיאגרמת המחלקות הלבנה מחלקות שהאובייקטים שלהן נשמרים במסד הנתונים כ- <persistent>. אם יש החרגה של חלק מהאובייקטים, צריך להוסיף הערה.
- iii. **מסד נתונים מרוחק (לא באותו מחשב כמו השרת):** בשנים כתיקון, הדרישה היא לתמוך בקישור המערכת למסד נתונים מרוחק. השנה, עקב הווירטואליות שנכפתה עלינו, תמיכה במסד נתונים מרוחק מזכה בציון בונוס של כ-10% ממשקל הגרסה.
- iv. **שמירת נתונים בזיכרון (caching):** מכיוון שתקשורת לבסיס נתונים היא יקרה (משאבי מחשב), 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.

v. [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.

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

- i. המערכת צריכה להתמודד עם אובדן תקשורת למסד נתונים.
 - ii. המערכת צריכה להתמודד עם החלפה של מערכות חיצוניות ואובדן תקשורת למערכות אלו.
- תבנית העיצוב Proxy עשויה לעזור בפתרון בעיות אלו.

2. דרישות פונקציונאליות:

a. שינוי דרישה 1.1 – אתחול המערכת:

בשלב האתחול המערכת יכולה לקבל קובץ חיצוני (בפורמט לבחירתכם) המגדיר מצב (state) בו המערכת צריכה להיות לאחר האתחול. הקובץ יכול הנחיות לסדרה של סיפורי שימוש וארגומנטים רלוונטיים להפעלתם. בשלב האתחול המערכת תקרא את הקובץ ותפעיל את סיפורי השימוש המתוארים בו, כך שבסיום האתחול המערכת תהיה במצב הנדרש. למשל:

```
guest-registration(moshe);
guest-registration(rina);
enter-market(rina);
login(rina);
open-shop(rina, shoes);
appoint-manager(rina, shoes, moshe);
```

b. דרישה 4.4 – הסרת מינוי של בעל חנות: יש לשים לב שדרישה זו מחייבת הסרת כל המינויים של בעל החנות ה"מפוטר", וכל המנויים המעורבים חייבים לקבל הודעות על שינוי מעמדם.

הערה: לפותח חנות אין ממנה שיכול להסיר את מינויו.

כמובן שהסרת מינוי משמעה שינוי מצב של מנוי, והשינוי צריך להיות משודר (dispatched) לכל יישות התלויה במצב זה, למשל חנויות שצריכות לדעת את מצב בעלי התפקידים בחנות.

c. שינוי דרישה 4.3 – בעל חנות רשאי למנות מנוי של המערכת כבעל חנות תחת הסכם מינוי, כמפורט להלן:

- הוספת בעל חנות טעונה אישור של כל בעלי החנות באותו זמן. בעל החנות מקבל את המעמד בחנות רק אם כל בעלי החנות האחרים מאשרים את המינוי החדש. כלומר, מינוי בעל חנות מחייב הסכם מינוי.
- תהליך אישור הסכם מנוי כולל משלוח הודעות לכל הבעלים ברגע המינוי, וקבלת תשובה חיובית.

- לכל מינוי בעל חנות יש בעלים אשר מינה אותו. שרשרת מינויי בעלי החנות אינה מעגלית עבור חנות נתונה. כלומר, לא תיתכן טרנסאקציה שבה בוב חתום על הסכם מינוי עבור אליס, שחתומה על הסכם מינוי עבור בוב, לאותה חנות.
הערה: יש לשים לב שהוספת בעל חנות הוא תהליך עם מצב משתנה, שרק בסופו המועמד הופך לבעל החנות. לכן, חייבים לאפשר **יישות המתארת את הסכם המינוי** ומצבו ברגע נתון. יש לדון בשאלת השמירה (persistence) של הסכם המינוי שעדיין לא אושר במלואו.

d. דרישה נוספת עבור קבוצה של שישה סטודנטים: דרישה חדשה:

- i. מנהל המערכת יוכל לצפות בסכום ההכנסות הכולל של מערכת המסחר מכלל החנויות באותו יום
- ii. מנהל המערכת יכול לבחור לצפות בהכנסות של כל אחד מהימים מרגע אתחול המערכת.

מידול ומימוש

ארכיטקטורה: הארכיטקטורה העדכנית של המערכת תכלול לפחות את הרכיבים השכבתיים הבאים:

- רכיב לקוח (presentation layer) עדכני.
- רכיב תקשורת (communication layer) עדכני.
- שכבת שירות (service layer) עדכנית.
- שכבת לב-מערכת (domain) עדכנית.
- שכבת קישור למסד נתונים (DAL).
- מסד נתונים

הסבר לגבי מימוש ותוכן רכיב ה-DAL: שכבת ה-DAL מגדירה את ההתאמה בין ישויות לב-המערכת לישויות בסיס הנתונים, ומטפלת בשמירה ואחזור של מידע ב/מ-בסיס הנתונים. לכל מחלקה מסוג persistent, מתאימים את הטבלה (או טבלאות) עבורה. מימוש המיפוי מתבצע על ידי הוספת שיטות עבור *insert, update, find* של האובייקטים שלה בבסיס הנתונים.
הקישור בין לב-המערכת למסד נתונים הוא מכיוון המערכת אל הנתונים ולא להיפך. כלומר, יש לתכנן ולהתאים את שכבת הנתונים למימוש הקיים. לא יתקבל פתרון המבוסס על תכנון עצמאי של מסד נתונים, שאינו מבוסס על תרגום ישויות המערכת ליישויות מסד הנתונים. לצורך מימוש רכיב DAL יש להשתמש בספריה קיימת התואמת את שפת המימוש של לב-המערכת.

תוצרי גרסה 3:

מסמך המתודולוגיה מתאר את התוצרים השונים ואופן תיאורם. התוצרים בגרסה זו מבוססים על התוצרים מגרסה 2 ומרחיבים עליהם. יש להקפיד על תיאום מוחלט בין מודלים למימוש, ולהציג באופן בולט את כל השינויים וההוספות שנעשו ביחס לגרסה הקודמת. מנהל הגרסה אחראי על הצגת תוצרי הגרסה.

1. דו"ח גרסה:

- a. פירוט ההספק לגרסה זו: אילו משימות מוכנות במלואן, אילו באופן חלקי ואילו נדחו לגרסה עתידית.
- b. תיעוד ניהול הגרסה, ע"פ העקרונות המוצגים במסמך המתודולוגיה (2.1.1).
2. תיקונים מגרסה 2, במידת הצורך ובהתאם להנחיות המנחה.

3. מודלים עדכניים:

- a. מילון מונחים עדכני.
- b. תרחישי שימוש עדכניים.
- c. ארכיטקטורה עדכנית.
- d. מודל מחלקות עדכני: דיאגרמה לבנה הכוללת סימון <persistent> עבור מחלקות שהאובייקטים שלהן נשמרים במסד נתונים.
- e. מסמך דרישות לממשק המשתמש
- f. מודל ממשק המשתמש כתרשים מצבים היררכי (statechart)
4. מימוש:

- a. רכיבי לב-מערכת (domain) ושירות (service) באופן התואם למודל המחלקות, למודל הארכיטקטורה, ולתרחישי השימוש.
- b. רכיב לקוח (presentation) באופן התואם את ה-statechart היררכי, ואת מסמך הדרישות שהכנתם.
- c. מימוש מפורש לתמיכה בהתראות זמן אמת, על פי תבנית העיצוב Observer.
- d. מימוש רכיב שמירת הנתונים לאורך זמן (כולל DAL ומסד נתונים).

5. בדיקות: הוספה ועדכון בדיקות לפי הדרישות החדשות. עבור כל הסוגים, כולל טסטים לממשק הגרפי. 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 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.
 - בחר כלי המאפשר הגדרה והרצה של בדיקות קיבול ועומס של המערכת.
 - הדגם עבודה עם הכלי.
 - תאר אוסף של תרחישי בדיקה למערכת המסחר המשמשים לצורך בדיקת קיבול ועומס.