

גרסה 4 – סטטיסטיקות למנהל וחיבור למערכות חיצוניות – 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 4 של מערכת המסחר עוסקת בהצגת סטטיסטיקות למנהלי המערכת אודות הפעילות השוטפת של המערכת ובהתממשקות המערכת עם מערכות חיצוניות היושבות בשרת מרוחק.

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 4

1. דרישות פונקציונאליות:

a. דרישה 6.5 – צפייה בנושא התנהלות המערכת:

- i. מנהל המערכת יכול לצפות במספר המבקרים היומי במערכת, בחתך אוכלוסיות: אורחים, משתמשים שאינם מנהלי או בעלי חנות, מנהלי חנות שאינם בעלים של אף חנות, בעלי חנות ומנהלי מערכת. הנתונים צריכים להיות מוצגים בטווח תאריכים נתון, כלומר, יש להוסיף תצוגה לממשק מנהל המערכת המאפשרת לו לבחור טווח תאריכים ולקבל את מספר המבקרים היומי עבור כל אחד מהימים בטווח התאריכים הנבחר.
- ii. כאשר מנהל המערכת צופה במספר המבקרים היומי בטווח תאריכים הכולל את היום, המידע צריך להתעדכן בזמן אמת בעת ביקור משתמשים במערכת. לצורך מימוש דרישה זו יש להשתמש ב-push notifications, בדומה להתראות זמן האמת המתקבלות בעת רכישה או מינוי מנהל.
- iii. שימו לב כי דרישת שירות 3a מחייבת זמן תגובה מהיר. נדרש למצוא פתרון שידע להתמודד ביעילות גם כאשר טווח החיפוש המבוקש מכיל חיבורים רבים למערכת. למשל, ניתוח יומן האירועים בכל בקשה להצגת נתונים הוא פתרון יקר שאינו מקיים את הדרישה.
- iv. המידע צריך להיות מוצג באופן ידידותי למשתמש, כטבלה או כגרף. ניתן להשתמש בשירות יצירת גרפים חיצוני (כגון [google chart](#)).

b. דרישות 7 ו-8 – חיבור למערכת תשלום ואספקה:

- i. על המערכת להתקשר עם מערכות התשלום והאספקה החיצוניות המפורטות [במסמך הנלווה](#).

ii. המערכת צריכה להיות **חסינה** להתנהגויות לא צפויות בעת התממשקות עם מערכות חיצוניות. דוגמאות להתנהגויות אלו הן כישלון בשליחת בקשה, קבלת פלט שאינו תואם את המנשק, אי מענה לבקשה וכו'.

2. דרישות איכות תוכנה:

a. יש לכתוב בדיקות white box המדמות טיפול **במספר בקשות במקביל**. על בדיקות אלו להתמקד בוודא קיום האילוצים וההתנהגות התקינה של המערכת, למשל, יש לוודא שאילוץ המלאי אי-שלילי יתקיים גם כאשר ישנם ניסיונות מקביליים לקנות את אותו המוצר.

3. דרישות רמת שירות:

a. **דרישה 4 – קיבול זמינות:** דרישה זו היא **חובה** עבור קבוצות של **שישה** סטודנטים **ובנוסף** (של עד 2 נקודות לציון הפרויקט) עבור קבוצות של **חמישה** סטודנטים.

i. דרישת העומס והקיבול מגדירה כי על המערכת לתמוך במספר לא מוגבל של משתמשים מכל הסוגים בו זמנית וכן במספר לא מוגבל של חנויות, מוצרים, חיפושים ורכישות (התואם את הצורך בפועל). בנוסף, על המערכת להיות זמינה ונגישה באופן תמידי (למעט פעולות סגירה יזומות). שימו לב, מדדים צריכים להיות כמותיים וניתנים לבדיקה.

ii. **היעדים (SLO)** עבור דרישות עומס וקיבול הם:

1. על המערכת לתמוך במספר לא מוגבל של משתמשים מכל הסוגים בו זמנית וכן במספר לא מוגבל של חנויות, מוצרים, חיפושים ורכישות (התואם את הצורך בפועל).
2. יש לתמוך בקצב מהיר של הצטרפות ועזיבה של משתמשים – אורחים ומנויים, ופתיחת חנויות

3. על המערכת להיות זמינה ונגישה באופן תמידי (למעט פעולות סגירה יזומות).

iii. **המדדים (SLI)** עבור עמידה ביעדים הם:

1. על המערכת להתמודד עם 100 בקשות (אירועים כגון התחברות, רכישה וכו') בו זמנית תוך עמידה בזמן תגובה של לכל היותר שניה לכל בקשה.
2. כמו כן על המערכת להיות מסוגלת לתמוך בהיקף של עד 1,000 חנויות, כאשר בכל חנות יש בממוצע 1,000 מוצרים, בהיקף של 10,000 משתמשים רשומים ובהיסטוריית של עד 1,000,000 רכישות.
3. תמיכה ב-1,000 מבקרים במערכת בכל רגע נתון.
4. המערכת איננה מפסיקה לפעול, גם כשיש אירועים לא צפויים, כמו נפילות תקשורת או קשר לרכיבים שונים (למעט פעולות סגירה יזומות).

iv. **ההסכמים (SLA)** למדדים לעיל ע"פ סדר הופעתם:

1. שסך הבקשות יהיו עם זמן תגובה החורג משניה אחת ומגיע לכל היותר לחמש שניות (ללא בקשות החורגות מחמש שניות).
2. עמידה מלאה במדדים.
3. עמידה מלאה במדדים.
4. המערכת יכולה שלא לפעול 0.05% מהזמן (כ-4.5 שעות בפרק זמן של שנה).
5. יש לכתוב ולהריץ בדיקות קיבול (load) ועומס (stress) אוטומטיות על מנת לבדוק את העמידה בהסכמים:

1. **תוכן בדיקות העומס והקיבול במערכת:** יש להגדיר אוסף תרחישים המדמה את השימוש הצפוי במערכת, כלומר, יש לחשוב על **תמהיל השימוש במערכת:** למשל, כמה מן המשתמשים יהיו מחוברים וכמה אורחים, כמה מנהלי חנות, כמה בעלי חנות, מה אחוז פעולות הרכישה מתוך סך פעולות המערכת, מהו קצב ההצטרפות, וכו'. תמהיל השימוש הצפוי צריך להיות מתואר בפירוט בראש קבוצת הבדיקות לקיבול. יש לכתוב בדיקות המדמות שימוש במערכת בהתאם לתמהיל זה.

2. **ניתוח את תוצאות הבדיקות:** יש לנתח את תוצאות הבדיקות בהתאם למדדים ולקבוע האם הסכם השירות מתקיים או לא.

רכיב הבדיקות לעומס וקיבול לא ייבדק ללא תיאור מילולי ברור של תכנון בדיקות אלו! הניקוד לא יקבע בהתאם לעמידה בהסכם השירות, אלא רק בהתאם לאופן הגדרת התרחישים וניתוח התוצאות.

מידול ומימוש: סיום הפרויקט

גרסה זו הינה הגרסה האחרונה, ובמסגרת הבדיקה של גרסה זו ייבדק הפרויקט **במלואו**. מסמך המתודולוגיה מתאר את התוצרים השונים ואופן תיאורם. התוצרים בגרסה זו מבוססים על התוצרים מגרסה 3 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. **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

2. **דו"ח גרסה:**

- פירוט ההספק לפרויקט: אילו משימות מוכנות במלואן, אילו באופן חלקי ואילו לא בוצעו.
- תיעוד ניהול הגרסה, ע"פ העקרונות המוצגים במסמך המתודולוגיה (2.1.1).
- תיקונים מגרסה 3, במידת הצורך ובהתאם להנחיות המנחה.

3. מודלים עדכניים:

- מילון מונחים עדכני.
- תרחישי שימוש עדכניים.
- ארכיטקטורה עדכנית.
- מודל מחלקות עדכני: דיאגרמה לבנה הכוללת סימון <persistent> עבור מחלקות שהאובייקטים שלהן נשמרים במסד נתונים. בשלב זה, ניתן גם להוסיף קשרי היררכיה בין מחלקות.
- מסמך דרישות לממשק המשתמש
- מודל ממשק המשתמש כתרחיש מצבים היררכי (statechart): תרחיש מרכזי, הכולל את מצב ההתחלה והמעבר למצבים היררכיים עבור תרחישי שימוש שונים. פירוט של המצבים ההיררכיים עבור תרחישי שימוש עיקריים.

4. מימוש:

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

5. בדיקות:

- בדיקות קופסה לבנה וקופסה שחורה לכלל דרישות המערכת, כולל לדרישות שנוספו בגרסה זו.
- תרחישי בדיקה (ללא מימוש) עבור הממשק הגרפי.
- על הבדיקות לעמוד בעקרונות בהם דנו במהלך הסמסטר:
 - כיסוי מרחב הקלטים
 - שימוש ב-mocks במקרים בהם חשוב לבדוק פונקציונליות של מחלקה ללא תלות במחלקות אחרות
 - שמות משמעותיים לבדיקות
 - בדיקת התנהגות אחת לכל סט
 - מחזור שלב האתחול (setup) בין הבדיקות השונות ככל הניתן
 - על בדיקות הקופסה הלבנה לוודא את המצב הפנימי של המערכת ולא רק את תקינות הערך המוחזר. לדוגמה, כאשר תהליך קניה נכשל יש לוודא כי מעבר לפלט המעיד על הכישלון מצב המערכת לא השתנה (לא ירדו מוצרים מהמלאי, לא נוספה רכישה להיסטוריה וכו')

מצגת כיתה: התאמת מערכת לשימוש בינלאומי

מערכות רבות מיועדות לשימוש בינלאומי, ויש להתאים אותן לכך. העקרונות להתאמות מהסוג הזה נקראת בינאום (internationalization), גלובליזציה (globalization) ולוקליזציה (localization). בז'רגון המקצועי הם נקראים L10N ו-G11N, I18N (המס' מייצג את כמות האותיות באמצע המילה).

1. מה מייצג כל אחד משלושת העקרונות הנ"ל ומה ההבדלים ביניהם?
2. מלבד תרגום, אילו התאמות ושינויים נוספים רצוי לבצע כחלק מתהליך התאמת מערכת לשימוש בינלאומי?

שים לב למשל, למדדים של דרישות רמת שירות.
3. איך ההתאמות הללו נתמכות? הבא/י דוגמה בסביבה/שפת תכנות לבחירתך.

מצגת כיתה נוספת עבור קבוצות של שישה סטודנטים: הערכת כיסוי קוד באמצעות מבחנים

- במסגרת הקורס שמנו דגש על חשיבות הבדיקות להבטחת איכות המערכת. אם זאת, לא מדדנו את איכות הבדיקות בהיבט כיסוי הקוד באופן פורמלי.
1. ישנן מספר מטריות להגדיר כיסוי הקוד באמצעות המבחנים, הצג/י את השיטות השונות ודון/י ביתרונות והחסרונות של כל שיטה.
 2. הסבירו כיצד ניתן להגדיר ולמדוד כיסוי קוד כאשר המערכת היא אסינכרונית.
 3. בחר/י כלי להערכת כיסוי אשר מתאים לסביבה בה אתם עובדים והדגם/י את הרצתו על המערכת שלכם.