

גרסה 1 - מימוש ראשוני - 2020

כל גרסאות הפרויקט מתייחסות אל ותלויות ב**מסמך הדרישות הכללי** וב**מסמך מתודולוגיית פיתוח** הפרויקט. כל גרסה מרחיבה או משנה את הגרסה שקדמה לה.

גרסה 1 עוסקת בהקמה ובמימוש שכבות ה-domain והשירות (service) של מערכת המסחר. שכבת השירות משמשת להפעלת סיפורי השימוש. בסיום גרסה זו, הפעלת המערכת תהיה באמצעות מבחנים בלבד. גרסה 1 בונה על בסיס עבודת המידול המקדימה שנעשתה בגרסה 0. **בכל שלב יש לשמור על תאימות בין המודלים והמימוש.**

תפקידי צוות: מנהל/ת גרסה, אחראי בדיקות, ומפתחים. פירוט תחומי האחריות של התפקידים השונים נמצא ב**מסמך המתודולוגיה**. במיוחד יש לשים לב לתחומי האחריות של מנהל הגרסה.

דרישות

1. כל אילוצי הנכונות מתקיימים. בנוסף יש להתייחס לאילוצי נכונות חדש:

10. בכל רגע נתון, יש מלאי אי-שלילי של כל מוצר שקיים במערכת.

2. דרישות רמת שירות:

- a. דרישת מגרסה 0: דרישת הפרטיות (1), דרישת עזרה כשיפור לחוויית משתמש (b3).
- b. דרישת המעקב (7). לצורך מימוש דרישת המעקב עליכם לתחזק יומן אירועים (event log) ויומן שגיאות (error log) כקבצים חיצוניים למערכת.
 - i. יומן האירועים מתעד את הפניות למערכת (אילו פונקציות נקראות עם אילו ערכי משתנים)
 - ii. יומן השגיאות מתעד שגיאות של המערכת. תרחישים שליליים כגון ניסיון התחברות עם סיסמה לא נכונה או ניסיון קניה בניגוד למדיניות אינם נחשבים כשגיאות. לעומת זאת, ניסיון לפתוח קובץ שלא קיים במערכת הוא דוגמה לשגיאה שצריכה להופיע ביומן השגיאות.

יש לשים לב שאילוצי פרטיות (למשל על ידי שמירת סיסמאות ביומנים) אינם מופרים במסגרת מימוש דרישת המעקב.

3. דרישות פונקציונאליות :

- a. כל הדרישות הפונקציונליות שפורטו במסמך המידול (גרסה 0).
- b. קבוצות של שישה סטודנטים נדרשת לממש גם את דרישות 3.3 (כתיבת ביקורת על מוצר), 3.5 ו-4.9 (ניהול פניות בין קונה מנוי לבעל חנות). יש לשים לב שביקורות ופניות הן ישויות הכוללות מגוון פרטים כגון זהות המבקר/פונה, תזמון, תוכן וכו'.

מימוש

- מערכת בארכיטקטורה דו-שכבתית. המימוש נדרש להיות תואם למודל הארכיטקטורה של המערכת:
 - שכבת ה-domain: שכבה זו היא לב המערכת. היא מחולקת לרכיבים אנכיים, כפי שהוחלט במודל הארכיטקטורה של המערכת. לכל רכיב, יש להחליט על הממשק שלו לשכבת השירות לרכיבים האחרים, ואיך תיעשה פנייה למערכות חיצוניות.
 - שכבת שירות (Service Layer): משמשת ממשק למערכת ומספקת נקודות כניסה לסיפורי השימוש של הגרסה. שכבה זו צריכה לשקף את התלויות בין סיפורי השימוש. אפשר לחלק את השכבה לתתי רכיבים.

רכיב בדיקות

- בגרסה זו נקבעת התשתית לבדיקות (מכל הסוגים) כך שיאפשרו כתיבה קלה של **regression tests**, **מבחני שילוב**, **שילוב מבחנים**, **ועקיבות** (traceability). לצורך זה יש בגרסה זו איש צוות האחראי על בדיקות.
- לכל סוג מבחנים שכותבים המפתחים, יש להחליט על אופן הארגון של המבחנים.
 - יש להקדיש מחשבה **למבנה** של כל בדיקה ועל אילו חלקים ניתן לשתף בין הבדיקות השונות.
 - עבור מבחני קבלה: יש להחליט על אופן הארגון והתלויות ביניהם. רכיב בדיקות הקבלה ייגש למערכת דרך שכבת השירות בלבד.
 - אחראי הבדיקות כותב את מבחני הקבלה.

מאפיינים של המערכת

- Traceability**: ניתן לעקוב אחר שינויי דרישות, קוד, מבחנים, והקשר ביניהם.
- יש לדאוג ל-**test coverage** טוב (במגוון מדדים) ולמבחנים שליליים (כאלו שכישלונם הוא הצלחתם!), כאשר המטרה היא לשאוף לכסות את סוגי הקלטים האפשריים (נציג מכל מחלקת שקילות).

תוצרי גרסה 1:

מסמך המתודולוגיה מתאר את התוצרים השונים ואופן תיאורם. התוצרים בגרסה זו מבוססים על התוצרים מגרסה 0 ומרחיבים עליהם. יש להקפיד על תיאום בין מודלים למימוש, ולהציג באופן בולט את כל השינויים שנעשו ביחס לגרסה הקודמת.

- **דו"ח גרסה - מוצג על ידי מנהל הגרסה:**
 - פירוט ההספק לגרסה זו: אילו משימות מוכנות במלואן, אילו באופן חלקי ואילו נדחו לגרסה עתידית.
 - תיעוד ניהול הגרסה, ע"פ העקרונות המוצגים במסמך המתודולוגיה (2.1.1).
 - מנהל/ת הגרסה ת/יתבקש להדגים Traceability של דרישות-קוד-בדיקות.
- תיקונים מגרסה 0, במידת הצורך ובהתאם להנחיות המנחה.
- מודלים עדכניים:
 - מילון מונחים עדכני.
 - תרחישי שימוש עדכניים.
 - ארכיטקטורה עדכנית.

- מודל מחלקות עדכני.
 - דיאגרמה לבנה: מחלקות, קשרים, אילוצים.
 - פירוט נוסף של הדיאגרמה הלבנה - אפשר להיעזר בהצמדת טקסט.
 - מימוש
 - שכבת ה-domain באופן התואם למודל המחלקות ולמודל הארכיטקטורה.
 - שכבת השירות (service) באופן התואם את מודל הארכיטקטורה ואת תרחישי השימוש.
 - **לא יתקבל מימוש ללא בדיקות!**
 - מימוש תשתית לבדיקות אוטומטיות ממגוון סוגים וכתובת בדיקות לתרחישי השימוש ולמימוש:
 - קבלה
 - יחידה
 - שילוב
 - עמידה באילוצים: על כל אילוץ נכונות יש להיות מסוגלים להסביר ולהראות כיצד הוא נאכף בקוד וכיצד הוא נבדק.
- מנהל הגרסה אחראי על הצגת תוצרי הגרסה.

מצגת כיתה: Websockets (10 דקות)

- הגרסה הבאה של מערכת המסחר תכלול Web client שימש כמנשק גרפי (GUI) המסוגל להציג התראות למשתמש בזמן אמת. מימוש רכיב ההתראות יעשה שימוש בפרוטוקול websockets המאפשר ליזום תקשורת דו-כיוונית בין שרת ללקוחותיו.
- הסביר/י את ההבדלים בין שימוש ב-websockets (secured) לבין השימוש בפרוטוקול HTTPS.
 - כבירת מחדל, בעת ניווט לכתובת URL חדשה (למשל, על ידי ניווט לדף אחר בצד הלקוח), חיבור ה-websocket אינו נשמר, הסביר/י כיצד ניתן להתמודד עם הבעיה ולשמר חיבור websocket יחיד לכל אורך הביקור באתר.
 - הסבר מדוע תקשורת דו-כיוונית מתנגשת המבנה השכבתי של מרכיב הייצוג ומרכיב האפליקציה. הסבר כיצד ניתן לעשות זאת מבלי לשבור את מבנה השכבות.