

גרסה 1 - מימוש ראשוני - 2019

כל גרסאות הפרויקט מתייחסות אל ותלויות במסמך הדרישות הכללי ובמסמך 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 במסמך המתודולוגיה. במיוחד יש לשים לב לתחומי האחריות של מנהל הגרסה.

דרישות

1. כל אילוצי הנכונות מתקיימים. בנוסף יש להתייחס לאילוץ נכונות חדש:
11. בכל רגע נתון, יש מלאי אי-שלילי של כל מוצר שקיים במערכת.
2. **דרישות רמת שירות:** דרישת הפרטיות (1), דרישת עזרה כשיפור לחוויית משתמש (3b) ודרישת המעקב (7).
3. **דרישות פונקציונאליות:** כל הדרישות הפונקציונאליות שפורטו במסמך המידול (גרסה 0).
4. **שונות:**
 - a. בשלב זה, אין מגבלות על **מדיניות הקנייה**. אפשר ורצוי לחשוב על זה כעל מדיניות קנייה ריקה, שלא מגבילה את הקנייה בשום צורה.
 - b. בשלב זה יש לממש רק **סוג רכישה** יחיד של **רכישה מיידית** (נספח 2).
 - c. בשלב זה, אין הנחות בקניית מוצרים. אפשר ורצוי לחשוב על זה כעל **הנחה גלויה** בכמות 0% וללא הגבלת זמן (נספח 3).

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 ולרכיבים 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 **regression tests**, כתיבה קלה של מבחני שילוב, שילוב מבחנים, ועקיבות (traceability). לצורך זה יש בגרסה זו איש צוות האחראי על בדיקות.

1. לכל סוג מבחנים שכותבים המפתחים, יש להחליט על אופן הארגון של המבחנים.
2. עבור מבחני קבלה: יש להחליט על אופן הארגון והתלויות ביניהם. רכיב בדיקות הקבלה ייגש למערכת דרך שכבת השירות בלבד.
3. 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) ול**מבחנים שליליים** (כאלו שכישלונם הוא הצלחתם!).

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 וכתיבת בדיקות לתרחישי השימוש ולמימוש:
 - קבלה
 - יחידה
 - שילוב

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 מהשרת אל לקוח כלשהו.

- השווה/י בין תקשורת חד-כיוונית (half-duplex) לתקשורת דו-כיוונית (full-duplex) בעזרת דוגמה;
- הצג/י את הבעיות העולות מהצורך בתקשורת דו-כיוונית במסגרת פרוטוקול HTTPS;
- הצג/י את הפתרון שמספק פרוטוקול websockets;
- הדגם/י כיצד ניתן להשתמש בפרוטוקול זה בקוד.