

גרסה 1 - מימוש ראשוני – 2023

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 נבנית 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.

הדרישות לגרסה 1:

המימוש של חלק מהדרישות מגרסה 0 נדחה לגרסאות עתידיות. הייצוג של הדרישות הללו במודלים צריך להישאר.

1. דרישות רמת שירות:

- a. דרישת העקביות (1), ודרישת הפרטיות (2), מתקיימות. בפרט, כל אילוצי הנכונות המפורטים במסמך הדרישות הכללי נשמרים לאחר כל פעולה.
- b. דרישת קיבול זמינות (5): על המערכת לתמוך במספר לא חסום של משתמשים מכל הסוגים בו זמנית ועל המערכת להיות נגישה באופן תמידי.
- c. דרישת המעקב (9): יש לתחזק יומן אירועים (event log) ויומן שגיאות (error log). היומנים תמיד זמינים לצפייה. יומן האירועים מתעד את הפניות למערכת (אילו תרחישים מופעלים עם אילו פרמטרים). יומן השגיאות מתעד שגיאות של המערכת. תרחישים שליליים אינם שגיאות.

2. דרישות פונקציונאליות:

- a. מערכת: דרישות 1,3,4.
- b. משתמשים: מימוש כל הדרישות הפונקציונליות שפורטו בגרסה 0, מלבד מימוש דרישה II.4.2 – שינוי סוגי וכללי (מדיניות) קניה והנחה של חנות.
- c. קבוצות של שישה סטודנטים נדרשות למדל ולממש גם את דרישה II.3.3 – כתיבת ביקורת על מוצר.

- 3. **בניית מודל עבור ממשק משתמש:** ממשק משתמש הוא מערכת ריאקטיבית המגיבה לפעולות של המשתמש. הממשק מפעיל תרחישי שימוש ע"י מעקב אחרי סדרת הפעולות של תרחיש מצד המפעיל (Actor). נמדל את נראות ואופן ההתנהגות של ממשק המשתמש על ידי Wireframes. סעיף הדגשים בהמשך כולל הסבר לגבי Wireframe. יש למדל את התרחישים הבאים באמצעות Wireframe, תוך מתן דגש על flow מעמוד הבית לדף בו מתבצע התרחיש, והדפים אליו הם מקושרים:

- a. תרחיש חיפוש מוצרים (תרחיש II.2.2).
- b. תרחיש מינוי מנהל חנות (תרחיש II.4.6).

הנחיות למימוש:

מערכת בארכיטקטורה דו-שכבתית, בהתאם למודל הארכיטקטורה של המערכת.

1. שכבת ה-domain: לב המערכת בה ממומש המודל הלבן, והיא מחולקת לרכיבים אנכיים.
2. שכבת השירות (Service Layer): ממשק עבור סיפורי השימוש.
3. תשתית בדיקות ומימוש בדיקות קבלה ובדיקות פיתוח (יחידה ושילוב).
4. מנגנון Traceability: ניתן לעקוב אחר שינויי דרישות, קוד, מבחנים, והקשר ביניהם.

דגשים למימוש:

1. מימוש המערכת:

- a. מימוש דרישת רמת שירות (5) מצריך מימוש מקבילי של המערכת (דרישת הקיבול). יש לשים לב ששימוש לא זהיר במנעולים ובסינכרוניזציה יפגע בדרישת הזמיות.
 - b. כדאי להוסיף ממשקים (interfaces) בכל מקום שבו יש סיכוי לשינוי מימוש בעתיד. כדאי גם להגדיר ממשק עבור אוספים של ישויות שממילא כבר יש להם מזהים.
2. תחומי אחריות ואילוץ נכונות: אילוץ נכונות צריך לאכוף באמצעות מבנה ופעולות המערכת, ולא להסתמך על בדיקות. יש לשים לב שפעולות מקביליות עלולות לגרום הפרה של אילוץ נכונות.
 3. תשתית הבדיקות:

- a. אחראי הבדיקות מגדיר ומממש תשתית בדיקות (מכל הסוגים) עבור כל גרסאות הפרויקט, וכן מממש את מבחני הקבלה. תשתית ברורה ופשוטה לניהול חיונית להמשך חלק של הפיתוח.
 - b. יש לספק גם בדיקות שליליות (כאלו שכישלון הוא הצלחתן), ובדיקות למקביליות. הצעות:
 - i. שני משתמשים מנסים לקנות את המוצר האחרון במקביל.
 - ii. בעל חנות מוחק מוצר ובמקביל משתמש אחר מנסה לקנות את המוצר.
 - iii. שני בעלי חנות מנסים למנות את אותו המשתמש למנהל במקביל.
 - c. הנחיות כלליות: בדיקות קבלה ניגשות למערכת דרך שכבת השירות בלבד; רצוי להשתמש ב-Mocks (מומלץ להשתמש ב-mocking framework). תרחישי בדיקה צריכים להיות ממוקדים.
4. בניית מודל עבור ממשק משתמש: Wireframe הוא סרטוט ראשוני של ממשק המשתמש העוזר לתכנן את הנראות והפונקציונאליות של הרכיבים בו ואת הקשרים בין רכביו השונים. אנו דורשים למדל סקיצה (אפשר גם בשחור לבן) של הדפים המוזכרים להעיל ברמה המפרטת את הכפתורים, התפריטים הרלוונטיים והמידע המוצג.
- דוגמה: דף התחברות. במקרה בו קיים דף התחברות הנפרד מהמסך הראשי, דף זה צריך לכלול כותרת, תיבות למילוי טקסט עם כותרת לכל תיבה, כפתור התחברות (אולי גם כפתור להרשמה), לדוגמה:

אם יש תמונות המיועדות להיות מוצגות בדף ניתן להשאיר מלבן ריק עבורם כ-placeholder. נדרש לציין כי כפתור ההתחברות מעביר למסך הבית במקרה של התחברות מוצלחת או מציג הודעת שגיאה (שגם אותה יש למדל) במקרה של שגיאה בהתחברות. להלן דוגמה נוספת (אינטראקטיבית) לשימוש ב-wireframe לצורך מידול ממשק משתמש (שימו לב שאנו לא דורשים מידול אינטראקטיבי של הכפתורים והדפים אליהם הם מעבירים, אך כמובן שניתן ומומלץ לעשות זאת). ניתן להשתמש בכלים שונים ליצירת wireframes: דף ועט, PowerPoint, או שימוש באפליקציות ייעודיות כגון [Figma](#), [uizard](#) ו-[draw.io](#).

תוצרי גרסה 1:

מסמך המתודולוגיה מתאר את התוצרים השונים ואופן תיאורם. התוצרים בגרסה זו הם:

1. דו"ח גרסה: מוצג על ידי מנהל הגרסה ומכיל את פירוט ההספק לגרסה.

2. מודלים עדכניים

3. מימוש המערכת: לא יתקבל מימוש ללא בדיקות!

4. מודל Wireframe

מצגת כיתה: Web Application Framework (10 דקות)

הגרסה הבאה של מערכת המסחר תכלול Web Client שימש כמנשק גרפי (GUI) המסוגל להציג את הדפים השונים של האתר לצורך כך אנו ממליצים להשתמש ב-WAF על מנת להקל את בניית התשתית ל-GUI.

1. בחר/י WAF לשפת תכנות בה אתם עובדים/ות (דוגמאות ל-WAF: Spring Framework (java),

Django (python), Masonite (python), flask (python), ASP.NET (#C).

2. הסביר/י מהו WAF ומה השימושים שלו.

3. הסביר/י מהו web service.

4. בהתאם לסביבה שבחרתם/ן:

a. איזה web services היא מספקת?

b. הסביר/י איך תעזרו בסביבה הזאת בגרסה הבאה. ואילו התאמות אתם/ן תצטרכו לבצע כדי להשתמש בסביבה.

c. הדגם/י בעזרת הכלי כיצד ניתן לממש דף web השולח מידע לשכבת השירות ומקבל ממנה מידע חזרה.

d. הסביר/י כיצד ניתן לקבל התראות זמן אמת מהמערכת באמצעות ה-WAF שבחרת.