

גרסה 1 - מימוש ראשוני – 2025

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) מתקיימת. בפרט, כל אילוצי הנכונות המפורטים במסמך הדרישות הכללי נשמרים לאחר כל פעולה.
- b. דרישת הפרטיות (2) והאבטחה (3): על המערכת לשמור סיסמאות של משתנים באופן מוצפן ולדאוג למנגנון מאובטח (מבוסס tokens) להתקשרות עם המערכת לאחר החיבור.
- c. דרישת קיבול זמניות (5): על המערכת לתמוך במספר משתמשים מכל הסוגים בו זמנית ועל המערכת להיות נגישה.
- d. 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 פונקציונליות:

- a. **מערכת:** דרישות 1,2,3. בשלב זה אין חיבור למערכות חיצוניות אמיתיות, אך נדרש proxy לחיבור לצורך התממשקות עם ה-domain.
- b. **משתמשים:** מימוש כל הדרישות הפונקציונליות שפורטו בגרסה 0, **מלבד:**
 - i. דרישה 3.10 - רכישת מוצר בודד בהגרלה.
 - ii. מימוש דרישה 4.2 – שינוי מדיניות קנייה של חנות.
 - iii. **הסרת מינוי של מנהל-חנות** לטובת גרסה זו (דרישה 4.8).
 - iv. אין צורך לממש את **ההתראות** למנהלי ובעלי החנות במימוש דרישה 6.1 אך מצופה **מכם להתייחס לכך שהמנהלים ובעלי החנות צריכים לקבל התראה ולהכין את הקרקע בהתאם.**

3. **בניית מודל עבור ממשק משתמש:** מידול ממשק משתמש מבוסס web על ידי Wireframes (כפי שמוסבר בסעיף הדגשים בהמשך המסמך). יש למדל את הדף הראשי (דף הבית) של המערכת, ממנה ניתן להפעיל את תרחישי השימוש השונים. על המדול לכלול את ה-layout שעתיד להיות מוצג למשתמשים (תפריטים, כפתורים, מבנה תוכן, וכו'). בנוסף, יש למדל את התרחישים הבאים:

- a. תרחיש חיפוש מוצרים (תרחיש 2.2).
- b. תרחיש שינוי הרשאות למנהל חנות (תרחיש 4.7).

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 האפליקציה (Application Layer):** ממשק עבור סיפורי השימוש.
3. **שכבת התשתית (infrastructure layer):** המימוש הקונקרטי ל-repositories ולטכנולוגיות האחרות (מע' תשלומים \ אספקה \ authentication וכו'...)
4. **תשתית בדיקות** ומימוש בדיקות קבלה ובדיקות פיתוח (יחידה ושילוב).
5. **GitHub repository:** הכולל ניהול קוד ופרוייקט.

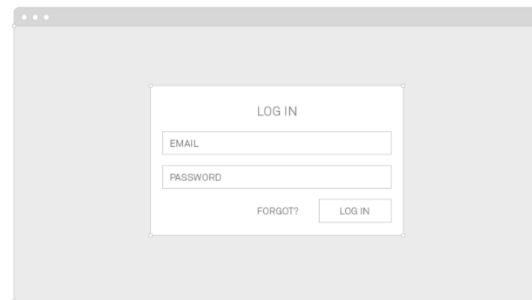
דגשים למימוש:

1. שימו לב לדגשים שנתנו במהלך הרצאת ההכנה לגרסה. דגשים אלו מהווים חלק בלתי נפרד מההנחיות.
2. **מימוש המערכת:**
 - a. מימוש דרישת רמת שירות (5) מצריך מימוש מקבילי של המערכת (דרישת הקיבול). יש לשים לב ששימוש לא זהיר במעולים ובסינכרוניזציה יפגע בדרישת הזמינות.
 - 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.
 3. **תחומי אחריות ואילוצי נכונות:** אילוצי נכונות צריך לאכוף באמצעות מבנה ופעולות המערכת, ולא להסתמך על בדיקות. יש לשים לב שפעולות מקבילות עלולות לגרום הפרה של אילוצי נכונות.
 4. **עבודה עם Github Repository:**
 - a. עליכם לנהל את הגרסאות של הקוד שלכם באמצעות github.
 - b. משימות יש להגדיר ב-github project שלכם כאשר כל משימה משוייכת ל-milestone של הגרסה הנוכחית.
 - c. שימו לב - מצופה שהcommit האחרון שלכם (לפני היום בדיקה וההגשה) יהיה משוייך ל-tag של הגרסה הנוכחית.
 - d. לכל משימה יש לייצר issue branch ו-pull request המתאימים למשימה.
 - e. יש לוודא שיש סימון מזהה של המשימה - bug, feature, ...
 - f. כל משימה פיתוח צריכה לכלול את ה-feature הנדרש ואת הבדיקות הרלוונטיות לאותו ה-feature. משימה שלא יוגדרו לה הבדיקות הרלוונטיות תוגדר כלא שלמה.
 5. **יצירת CI - Continuous Integration:**

מומלץ	להיעזר	בתוכן	שנמצא	בקישור
https://docs.github.com/en/actions/use-cases-and-examples/building-and-testing/building-and-testing-java-with-maven				

 - a. יש לעבוד עם github actions על מנת לתמוך ב-continuous integration
 - b. על ה-CI לכלול את כלל הבדיקות של כל התכולות שנוספו לגרסה.
 - c. על ה-CI לרוץ כחלק מהשלבים שמתבצעים לפני שעושים push to main.
 - d. ניתן להוסיף שלבים נוספים כראות עינכם לפי הצורך.
 6. **תשתית הבדיקות:**

- a. אחראי הבדיקות מגדיר תשתית בדיקות (מכל הסוגים) עבור כל גרסאות הפרויקט, וכן מממש את **מבחני הקבלה**. תשתית ברורה ופשוטה לניהול חיונית להמשך חלק של הפיתוח.
- שימו לב - בדיקות הפיתוח צריכות להיעשות תוך כדי פיתוח הפיצ'ר ועל ידי האדם שפיתח אותו.
 - אחראי הבדיקות אינו מצופה לכתוב את ה-unit tests והם צריכים להיכלל כחלק מהדרישה לסיום משימת פיתוח.
- b. יש לספק גם בדיקות **שליליות** (כאלו שכישלוןן הוא הצלחתן), ו**בדיקות למקבילות**. הצעות (כמובן שהבדיקות הללו לא מכילות את כל האפשרויות שנדרשות):
- שני משתמשים מנסים לקנות את המוצר האחרון במקביל.
 - בעל חנות מוחק מוצר ובמקביל משתמש אחר מנסה לקנות את המוצר.
 - שני בעלי חנות מנסים למנות את אותו המשתמש למנהל במקביל.
- c. הנחיות כלליות: **בדיקות קבלה ניגשות למערכת דרך שכבת האפליקציה בלבד**; רצוי להשתמש ב-Mocks (ע"י שימוש ב-mocking framework). תרחישי בדיקה צריכים להיות ממוקדים.
7. **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 כולל כותרת, תיבות למילוי טקסט עם כותרת לכל כל תיבה, כפתור התחברות (ואולי גם כפתור להרשמה),



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. **מודלים עדכניים**
2. **מימוש המערכת**: לא יתקבל מימוש ללא בדיקות!
3. **מודל Wireframe**