

נקודות שעלו במפגש עם שחף:

- ☐ להחליט אילו פעולות תקשורת צריכות להיות ממומשות באמצעות webSockets (למשל נוטיפיקציות).
- ☐ לממש נוטיפיקציות באמצעות webSockets.
- ☐ לוודא שיש לנו התייחסות בקוד למערכות חיצוניות (על אף שאין API):
- ☐ איך מתייחסים לביצוע תשלום בקוד (מערכת תשלומים)?
- ☐ איך מתייחסים לביצוע משלוח בקוד (מערכת אספקה)?
- ☐ לרשום מה הן פעולות שאורח יכול לבצע.
- ☐ לתמוך בביצוע פעולות אורח (מימוש בקוד).
- ☐ בבדיקות קבלה - לא אמורים להכיר את שכבת הדומיין (למשל לא מכירים מחלקה שנקראת User), אלא רק את ה-API של ה-Service Layer.
- ☐ בדיקות הקבלה צריכות לקרוא ישירות ל-API (להוריד אותן שכבה אחת).
- ☐ לוודא שבזמן ההרשמה המשתמש יתבקש להזין פרטי תשלום וכתובת (להוסיף שדות)?
- ☐ לאפשר עדכון של פרטי התשלום והכתובת של משתמש.
- ☐ לבצע בדיקות חוקיות לפונקציות:
- ☐ חוקיות של קלט - למשל: האם rank הוא מספר בין 1 ל-5?
- ☐ חוקיות של הוספת אובייקט למערכת - למשל: האם שדה מסוים צריך להיות ייחודי במערכת?
- ☐ חוקיות של הסרת אובייקט במערכת - למשל: האם האובייקט המוסר קיים בכלל?
- ☐ חוקיות של הרשאה - למשל: האם המשתמש רשאי לבצע את הפעולה?
- ☐ הפונקציה addReview לא נדרשת למימוש כרגע. האם לוותר על תיקון שלה? - החלטה. משאירים תקול. לא עושים טסטים לפונקציה הזו.
- ☐ להוסיף סקריפט שבהרצת המערכת מאתחל אותה ואת בסיס הנתונים.
- ☐ לעדכן את דיאגרמת הרצף של UC 1.1.
- ☐ להתייחס ב-UC 1.1 לעובדה שיש סקריפט שרץ באתחול המערכת (?)
- ☐ תרחיש שימוש 2.8:
- ☐ כתוב שהמשתמש צריך להזין פרטי תשלום וכתובת - לתקן.
- ☐ מתי נועלים את המוצרים עבור המשתמש? או שלא נועלים ובודקים שוב בסוף התהליך שהם עדיין זמינים? ומה אם הם לא זמינים?
- ☐ להסביר מה פירוש "אם הם אינם רלוונטיים".
- ☐ לתקן את דיאגרמת הרצף בהתאם.
- ☐ בפוסט קונדישנס העגלה הופכת להיות הזמנה (הערה של שגיא).
- ☐ להתייחס **במלל** למקרים המתוארים בבולט הבא:
- ☐ להתייחס **בקוד** למקרים הבאים:
- ☐ פעולת התשלום הצליחה ופעולת האספקה נכשלה.
- ☐ פעולת התשלום נכשלה ופעולת האספקה הצליחה.

☐ שתי הפועלות נכשלו.

- ☐ בבדיקות הקבלה - להוסיף בדיקות של קלט לא חוקי וטיפול נכון במקרים של כישלון.
- ☐ לשנות בתרשים המחלקות דברים שקשורים ל-workers, role, store - להעניף את השדה workers מ-store
- ☐ טסטים שחייבים להיות: ביצוע רכישה, וביצוע רכישה שנכשלה (בדיקות קבלה).
- ☐ לטפל במדיניות קניה (מה זה אומר?).
- ☐ לכתוב במלל את ה-flow של תהליך רכישה כולל מקרים של שגיאה.
- ☐ לכתוב את תהליך הקניה במלל באופן ברור כולל התייחסות לשגיאות.
- ☐ צריך לעשות mock למערכות החיצוניות ולבדוק מקרים של הצלחה ושגיאה.
- ☐ בבדיקות הקבלה עובדים רק עם אובייקטים פרימיטיביים.
- ☐ סדר עדיפויות: 1. סרבר. 2. מדיניות קניה והנחות. 3. ממשק גרפי.