

# גרסה 0 - מידול המערכת – 2020

כל הגרסאות של הפרויקט מתייחסות ותלויות במסמך הדרישות הכללי ובמתודולוגיית פיתוח הפרויקט. כל גרסה כוללת חלק מהדרישות במסמך הכללי, עם או בלי שינויים, ואולי דרישות נוספות. משימת המידול הראשוני (גרסה 0) מניחה את התשתית לגרסה 1 ובעצם לכל המערכת ומהווה את הבסיס לפרויקט. על כן יש לה חשיבות מכרעת על המשך הפרויקט. הגרסה כוללת:

- בניית מודלים – use cases, class models, system sequence diagrams, עבור הדרישות בגרסה,

- דיאגרמת ארכיטקטורה של רכיבי המימוש,
- תכנון בדיקות לקוח (acceptance tests).

**תפקידי צוות:** מנהל גרסה ומפתחים. חשוב להגדיר בכל גרסה במהלך הסמסטר מנהל גרסה שונה. כל חברי הצוות צריכים להיות שותפים מלאים במשימת המידול.

## הדרישות בגרסה 0 (מתוך מסמך הדרישות הכללי):

1. כל אילוצי הנכונות מתקיימים.
2. דרישות רמת שירות: דרישת הפרטיות (1), ודרישת עזרה בטעויות כשיפור לחוויית משתמש (3b), מתקיימות.
3. דרישות פונקציונליות: מידול תרחישי השימוש הבאים, מתוך מסמך הדרישות הכללי:
  - א. מערכת: 1.1
  - ב. קונה אורח: 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8
  - ג. קונה מנוי: 3.1, 3.2, 3.7
  - ד. מוכר בעל חנות: 4.1, 4.3, 4.5, 4.6, 4.7, 4.10
  - ה. מוכר מנהל חנות: 5.1
  - ו. מנהל מערכת: 6.4
  - ז. מערכת גביית כספים (7): התקשרות למערכת גבייה חיצונית.
  - ח. מערכת אספקת מוצרים (8): התקשרות למערכת חיצונית.מומלץ להסתפק ברכישה מיידית (נספח: 2.1), ובהנחה גלויה (נספח: 4.1).

## תוצרי גרסה 0:

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

- א. מילון מונחים.
- ב. תרחישי שימוש:
  - תרחישים מורכבים, כגון דרישה 2.8, יש להפריד לתתי-תרחישים. עבור קונה מנוי, אין צורך לחזור על התרחישים שכבר תוארו עבורו כקונה אורח. רק לתאר את ההבדלים.
  - בנוסף לתיאור המילולי של התרחישים יש לבטא את האינטראקציה בין המערכת לגורמים חיצוניים באמצעות System sequence diagram.

- כל תרחיש שימוש צריך להיות מלווה בתרחישי קבלה חיוביים ושלייליים. אלו יהוו את הבסיס למבחני קבלה עבור המימוש.
- ג. **דיאגרמת ארכיטקטורה:** יש לשים לב מהם הרכיבים במערכת, לשים לב למנשק שכל מרכיב חושף, ולשים לב לצורך בשכבת שירות כמנשק כולל עבור הדרישות הפונקציונליות.
- ד. **מודל מחלקות:** דיאגרמת מחלקות לבנה **עשירה**, המציגה מחלקות (ללא היררכיה), קשרים ביניהן ואילוצים המוטלים עליהם. על הדיאגרמות לשקף את הרכיבים.
  - לכל מחלקה, צריכים להיות מסוגלים להסביר עבור איזו דרישה/ות נוספה, את ייעודה, ולהדגים אובייקטים.
  - עבור כל קשר, צריך להיות מסוגל להצדיק את קיומו כקשר סטטי במימוש, ולהראות כיווניות, אם ישנה.
  - הדיאגרמה חייבת לכלול את כל האילוצים המתבקשים ממסמך הדרישות של גרסה זו, שבהמשך ימומשו בקוד.
- ה. **קיום אילוצי נכונות:** עבור כל אילוץ נכונות, צריכים להראות כיצד יובטח קיומו: על ידי הבנייה, או על ידי תרחישי קבלה.

## מצגת כיתה – מערכת לניהול פרויקטים

**בכל גרסה יהיה על סטודנט אחד מהצוות להציג מצגת אישית.**

אנו ממליצים של מערכת GitHub ככלי לניהול הפרויקט והקוד.

נתח/י כיצד האלמנטים הבאים ניתנים לניהול באמצעות הכלי לניהול פרויקטים (GitHub):

- הגדרת משימות
- תיעדופן ותזמון
- ניהול הצוות וחלוקת משימות
- מעקב אחרי התקדמות הגרסה
- מיפוי (מעקב) בין הדרישות השונות לקוד המתאים ולבדיקות (traceability).
- ניהול הקוד (code review, pull requests, etc.).
- ניהול תיעוד הפרויקט (כולל המודלים).

הדגם/י את אפשרויות הניהול הנ"ל עבור הפרויקט.

**הערה:** ניתן לבחור פלטפורמה חלופית לניהול הפרויקט בהתייעצות עם ובאישור המנחה.