



## ניתוח ועיצוב מערכות תוכנה, ניתוח ותיכון מערכות תוכנה (372-1-3801, 372-1-3401)

מטלה 1

מועד הגשת העבודה – 24.4.25 עד 23:59

שאלות ללקוח ניתן להפנות בפורום עד לתאריך 10.4.25

אין להתחיל את התרגיל לפני שהתחלקתם לזוגות בתוך השמינייה ובחרתם לכל זוג מודול - בני הזוג של המודולים לא משתנים לאורך כל הסמסטר.

שימו לב שבמידה ותתחילו את המטלה לפני שהתחלקתם לזוגות, יכול מאד להיות שבסופו של דבר תצטרכו לבצע את המטלה עבור מודול אחר.

### הקדמה

הפרויקט ילווה את הקורס ויהווה תרגול מעשי של שלבי הניתוח והעיצוב של מערכת, בדגש על ניתוח ועיצוב המערכת ועדכון המסמכים לאור שינויים בדרישות הלקוח.

הפרויקט יוגש בשלבים: לכל שלב יהיה פירוט של המטלות, ויניתן לו תאריך הגשה כפי שמפורט בסילבוס. חלק רב מן העבודות יבוצעו באמצעות כלי מידול (כגון **Visio**, **Visual Paradigm**, **Draw.io** וכלי **CASE** אחרים).

### תיאור מטלה 1

הלקוח של המערכת העלה על הכתב את הבעיות העומדות בפניו והצרכים כפי שהוא רואה אותם בחלוקה ל-4 תתי-מערכות (מודולים). **במטלה זו כל זוג בשמינייה יעבוד על מודול אחר: ספקים, מלאי, הובלות ועובדים, כאשר כל זוג עובד על מודול יחיד<sup>1</sup>**. כל זוג יגיש את המטלה בעצמו על **branch** ייעודי ויבדק באופן בלתי תלוי מהזוגות האחרים. עם זאת מומלץ לקיים דיונים בין הזוגות, לשתף קוד בתוך השמינייה, ולהשתמש באותם **coding conventions** והחלטות עיצוב.

עליכם, כמנתחי ומעצבי מערכת, לספק ללקוח ניתוח של דרישות המערכת בגישת **Object Oriented** ולממש את המודול שעליו אתם אחראים בשפת **Java**. בכל מקום בו אתם מניחים הנחה שאינה כלולה במסמך הדרישות ו/או במסמך הייזום ומשלימה אותה, אנא ציינו זאת במפורש. בכל מקרה שבו הינכם מתלבטים בין הנחות הכרחיות שמשפיעות משמעותית על קושי המימוש תפנו את השאלה לצוות הקורס דרך המודל, בפורום המיועד, כך שכלל התלמידים יקבלו את התשובה. שימו לב, התשובות בפורום מחייבות את כלל הסטודנטים - התעלמות או פספוס הנחייה הכתובה בפורומים היא על אחרייתכם ועלולה לגרור הורדה בניקוד!

המטלה תבוצע ותוגש בשני חלקים:

חלק א' (60%): ניתוח ועיצוב ראשוניים של המערכת לאור מסמך הבעיות והצרכים.

חלק ב' (40%): מימוש המודול וביצוע בדיקות היחידה - Unit Tests.

<sup>1</sup> במידה ושני זוגות יממשו את אותו מודול, ידרש זוג אחד להשלים את המודול החסר



## חלק א' - ניתוח דרישות ועיצוב ראשוני של המערכת

### 1. דרישות (25%)

1.1. נתחו את הדרישות של המודול שעליכם לממש על פי התיאור של כלל המערכת שקיבלתם. במהלך הניתוח עליכם להשלים את הדרישות שמשתמעות מהכתוב גם אם אינן מופיעות במפורש בטקסט. מומלץ להשתמש בטכניקת **ARM** על מנת להשלים את הדרישות ולסדר אותן. את הדרישות יש לפרט בטבלה מספר 1.

שימו לב, סיפור המערכת כתוב כמו שלקוח היה מספר לכם על המערכת בה הוא מעוניין. עליכם להבין ולהסיק מסיפור המערכת את הדרישות. לגבי הנחות על דרישות המערכת קראו בהמשך. אם אתם בכל זאת רואים צורך לשאול מה הייתה כוונת הלקוח בסיפור אתם מוזמנים לשאול זאת בפורום "שאלות ללקוח" עד התאריך המתאים שצויין בתחילת העבודה.

להלן דוגמה לטבלה 1: דרישות

ID	Module	Functional / Non-Functional	Description	Priority	Risk	Status
1.	HR		The system <b>MUST</b> support registration of new employees.	MH	Low	Done
2.	HR		The system <b>CAN</b> allow employees to exchange shifts.	NTH	High	InProgress
3.			....			

שימו לב, על הדרישות להיות מנוסחות בצורה של "על המערכת..." / "המערכת תאפשר...". וכדומה.

1.1.1. לכל דרישה יש להגדיר האם היא דרישה פונקציונלית או לא פונקציונלית.

1.1.2. יש לתעדף את הדרישות.

1.1.3. לבצע הערכת סיכון ולסמן באופן ברור אילו דרישות ימומשו ואילו לא ניתן לממש בשלב זה של הפרויקט (יש להסביר מדוע לא ניתן).

1.2. יש לפרט את כל השאלות הפתוחות שעליהן לא ניתן לענות על פי תיאור המערכת, כלומר שאלות שאינן הנכם אמורים לחזור ללקוח.

1.2.1. כל השאלות שאינן משפיעות על מימוש המודול בשלב זה יישארו כשאלות פתוחות ויופיעו בטבלה מספר 2.

דוגמה לטבלה 2: שאלות פתוחות

#	Topic	Issue
1.	Order frequency	How often can orders be issued due to insufficient inventory?
2.		
3.		

1.2.2. לכל השאלות שמשפיעות על המימוש בשלב זה אך אינן משנות משמעותית את קושי המימוש (הבדל של פחות מ-5 שעות עבודה בערך על הקוד) עליכם להמציא את תשובת הלקוח ולתעד אותה יחד עם השאלה בטבלה 2. (הטבלה בפורמט חופשי).

1.2.3. את כל השאלות שמשפיעות משמעותית על קושי המימוש (הבדל של יותר מ-5 שעות עבודה על הקוד) עליכם לשאול במודל, בפורום השאלות ללקוח, עד התאריך המצוין עבור שאלות מסוג זה בראש בעבודה. יש לעדכן את הטבלאות בהתאם לתשובות שיתקבלו.

## 2. תרשים מחלקות (15 %)

2.1. לאור הדרישות שתוארו במסמך הדרישות, עצבו ותארו את המודול שלכם באמצעות תרשים מחלקות. המחלקות חייבות להתחלק לשתי קבוצות (שתי שכבות): (1) מחלקות שמיועדות לאינטראקציה עם משתמש בלבד! (presentation layer) (2) מחלקות שממשות את הדרישות הפונקציונליות ללא קשר לצורת האינטראקציה עם המשתמש (domain layer).

מומלץ (אך לא חובה במטלה זו) להגדיר מחלקות שמהוות ממשק בין שתי הקבוצות (יהיה חובה במטלה הבאה).

2.2. על המחלקות לכלול שדות בלבד (כולל טיפוסים).

2.3. יש לשמור על תקינות התרשימים, לציין ריבוי (multiplicity) ותפקידי המחלקות בקשרים (Roles).

יש להגיש את התרשימים (שיפותחו באמצעות כלי תוכנה) בתוך מסמך האפיון בהמשך לדרישות. יש לדאוג שהתרשימים יהיו קריאים ברורים ועם גודל פונט סביר (לפחות 8) בדף. במידת הצורך ניתן לחלק את התרשים.

## חלק ב' - מימוש המודול (40%)

1. יש לממש את המודול שקיבלתם בשפת Java. יש לשמור על traceability בין תרשים המחלקות בסעיף 2 בחלק א' לבין המימוש: כל מחלקה וקשר שמופיעות בתרשים חייבות להופיע בקוד, במקרה של חוסר תאימות בין השניים אתם עלולים לאבד נקודות. הערה: לא כל שורת קוד\מבנה נתונים שמימשתם צריך להופיע במודל המערכת וזאת בהתאם לרמת ההפשטה שנבחרה. אין חובה לכלול בתרשים המחלקות מבני נתונים ומחלקות שירות סטטיות.

בעבודה זו אתם לא מממשים בסיס נתונים כלשהו. עליכם לשמור את כל המידע בזיכרון בעזרת מחלקות ייעודיות שיחזיקו את הנתונים הרלוונטיים.

## 2. בדיקות

יש להכין בדיקות לפי אחת משתי האופציות הבאות:

- יש לממש לפחות 10 בדיקות מסוג Unit Tests. מומלץ מאוד לממש בדיקת יחידה על שכבת התחום (domain layer). בנוסף, מומלץ לממש בדיקה כל פעם לפני שהנכם פותרים באג מסוים. הבדיקה אמורה להיכשל לפני פתרון הבאג ולהצליח אחרי פתרון הבאג. שימו לב, במקרה ולא יהיו לפחות 10 טסטים או שחלק מהם לא יעברו, ירדו לכם נקודות.
- בדיקות ידניות של שלושה תהליכים מרכזיים (end-to-end) המתאימים לדרישות פונקציונליות שהגדרתם. לכל בדיקה יש לפרט את התסריט שכולל את כל הקלטים שהמשתמש מזין למערכת אחד אחד (ערכי הקלטים ולא רק הגדרה גנרית), את הפלטים המצופים, ואת התוצאה בפועל בעת ביצוע הבדיקות. כל בדיקה חייבת להתחיל מהפעלה ראשונית של המערכת כולל אתחול המערכת במידת הצורך.
- יש לצרף הוראות הפעלה ברורות של המערכת בקובץ PDF בתיקיית docs. הוראות ההפעלה צריכות לאפשר למשתמש שאין לו כל היכרות מוקדמת עם המערכת להפעיל אותה. ההוראות יכילו פרטי משתמש + סיסמאות במידה ויש במערכת שימוש בהרשאות גישה.
- בגיש, על ה-Repository שלכם להיות במבנה הבא:

תיקיית dev: תכיל את כל ה-source code, קבצי jars, database חיצוניים וכו'.  
תיקיית docs: תכיל קובץ בשם instructions.pdf. אלו הן הוראות ההפעלה של המערכת שלכם.  
תיקיית release: תכיל קובץ בשם adss2025\_v01.jar. זהו ה-jar המקומפל שייבדק במחשב האישי שלכם.

לכל שאלה או הבהרה – יש פורום במודל למטלה .

## דגשים נוספים

1. בתרגיל זה אין database! כל הנתונים צריכים להישמר בזיכרון ולהתאפס בכל ריצה.
2. עם זאת, על המערכת להכיל נתונים לדוגמה שמאפשרים את בדיקת הפונקציונליות של המערכת שלכם באופן מלא! עליכם לצרף במסמך instructions פרטים על הנתונים הקיימים במערכת עם עלייתה.
3. מכיוון שאתם לא משתמשים בבסיס נתונים בתרגיל זה, בכדי לענות על הדרישה לעיל עליכם לכתוב פונקציה שתאתחל את המערכת במידע הנחוץ עם עלייתה. פונקציה זו צריכה להיות חיצונית ל-domain layer, כלומר עליכם לבצע את הקריאות המתאימות ל-domain layer מפונקציה זו, ולא לגשת ישיר לdata access layer.
4. יש לאפשר את עליית המערכת גם ללא הנתונים ולאפשר טעינתן לאחר מכן.