

קבוצה 20		
תאריך כתיבת המטלה: 29/03/2020-02/04/2020		
שם מלא	ת"ז	מייל
איילון בן סימון	312162951	aylon8@gmail.com
סער ויקטור	312392822	saarms5@gmail.com
שובל דוד	305284945	shovaldavid877@gmail.com
שירז קרוב	313599987	shirazkarov@gmail.com
גל טולדנו	205886559	gal1010100@gmail.com

1. תרומתו של המודל שיצרנו תורם לתהליך הפיתוח הכולל של המערכת ע"י כך שהוא מתאר בצורה חזותית תרחישי שימוש לדוגמא המקרה הבא: order execution examination. אם נתבונן ב-use case הנ"ל אנו נבחין שיש לנו שני תנאים בשביל מקרה זה (yes/no) ונדע להתייחס אליהם ולדרכי הטיפול בהם בהתאם בהמשך פיתוח הפרויקט ובפרט עבודה על הקוד. כמו כן, המודל נחוץ למידול קשרי הגומלין בין התוכנה ולבין מפעיליה ובכך לדעת בין אילו בעלי תפקידים שונים נוכל לקשר פעולות שונות במערכת בהמשך פיתוח הפרויקט כלומר ניתן להסתכל על הדוגמא הבאה: customer system registration. אם נתבונן ב-use case הנ"ל אנו נבחין שיש לנו case משותף אשר גם הלקוח וגם הנציג שיווק חולקים אותו. בנוסף, המודל משמש כלי להגדרת ממשקים נחוצים למערכת אשר יהוו בסיס לניתוח פשוט ויעיל יותר בהמשך שלבי הפיתוח, למשל במקרה: track order status הוא בעצם ממשק שמשמש את הלקוח אחרי ההזמנה שלו. כמו כן, חיישן ה-nfc מהווה ממשק ללקוחות בעלי דלקן ותומך בתדלוק מהיר.

2. המרכיבים הפונקציונליים של המערכת שלא ניתן לבטא בעזרת מודל use case הם:

- מודל ה-use case אינו מתאר את הדרך מימוש של כל מקרה כלומר אפילו שפעולה מצוינת בדיאגרמה לא ידוע כיצד ממשים אותה לדוגמא: כאשר לקוח עוקב אחרי הזמנה (track order status), לא מתואר אופן המימוש של המעקב אחר ההזמנה כלומר לא ידוע איזה פונקציה תתמוך בפעולה זו.
- מודל ה-use case אכן מתאר קשרי גומלין אך במידה ופעולה קשורה לשניים או יותר actors (בעלי תפקידים) לא ניתן לדעת איזה מבין בעלי התפקידים מבצע את הפעולה ואיזה actors קשורים אליה באופן עקיף, לדוגמה במקרה customer system registration לא ניתן להבחין מי יוזם את הפעולה ומי קשור אליה באופן עקיף.
- במודל ה-use case לא ניתן לראות את המסרים המועברים ע"י המערכת בין ה-actors השונים (בעלי תפקידים) כלומר לא ניתן לראות מתי המערכת מבצעת פעולות שונות ומתי בעלי התפקידים מבצעים פעולות שונות במערכת, לדוגמה במקרה שהזמנה בוצעה, המערכת גם משנה סטטוס וגם מודיעה למנהל בהודעת אימייל ובמסרון שההזמנה בוצעה, דבר שאיננו רואים במודל ה-use case.
- פעולות ישירות של המערכת כלומר אין בעל תפקיד אשר מבצע פעולות אלו לדוגמא בתרגיל שלנו המערכת מאפשרת מתן שירותים ללקוחות פרטיים וכן לחברות אך הנ"ל אינו מיוצג בתרשים use case.

הסיבה לאי הצלחה של מודל ה-use case להראות את הבעיות שהצגנו היא אופן מימוש דיאגרמת ה-use case אשר לא מתארת בשלמות את הנקודות שתיארנו לעיל.

המאפיין של מרכיבים אלו הוא שהם אינם דברים טכניים אלא דברים כלליים, כלומר נדרשת הסתכלות פנימה לפרטים והבנה עמוקה יותר של המערכת/דברים שניתן לבצע בעזרתה. פרטים אשר ימומשו בשלבים מתקדמים יותר בעזרת שאר הדיאגרמות.

3. בתשובה מספר 2 ציינו מגבלות של מודל ה-use case, הדרכים שאנו מציעים להתגבר על המגבלות שציינו הם:

- שימוש במודל class diagram יפתור לנו את הבעיה הזו מכיוון שבאמצעות מודל זה ניתן לתאר את דרך המימוש של כל מקרה, המחלקות שבמודל אמורות לספק את כל הפונקציונליות המערכתית. במודל מתוארות הפונקציות וכך אנו מתגברים על חוסר הידיעה של אופן המימוש עבור כל פעולה. עבור ה-class לקוח תהיה רשומה למשל הפונקציה track my order וכך נדע היכן למקם את המתודה במהלך כתיבת הקוד.
- שימוש במודל sequence diagram יפתור לנו את הבעיה הזו מכיוון שהתרשים ממדל את נקודת המבט הדינמית של המערכת כמו כן, הוא מציין את סדר המשלוח של המסרים השונים. כלומר עבור פעולה מסוימת בתרשים רואים איזה ישות מתחילה את הפעולה ואת תהליך הזרימה המלא של כלל המקרה בדוגמה שהצגנו במידה והיינו מציגים את ה-case של customer system registration בתרשים sequence היינו רואים כיצד נציג השיווק רשם למערכת את הלקוח שלב אחר שלב.
- שימוש במודל activity diagram יפתור לנו את הבעיה הזו מכיוון שבמודל זה ניתן לראות את המסרים המועברים ע"י המערכת לבעלי התפקידים השונים ולהיפך. לדוגמה במקרה שהזמנה בוצעה, נוכל לראות בתרשים את אופן שינוי הסטטוס ושליחת הודעה (פעולות המבוצעות ע"י המערכת) מתאימה לבעלי התפקידים הרלוונטיים.
- שימוש במודל class diagram יפתור לנו את הבעיה הזו מכיוון שבמודל יש class של system שבו נוכל לראות את המתודות השונות אשר באמצעותן נוכל לבצע את הפעולות השונות.