# קבוצה 4:

**יזהר אננייב** [Izhar\_ananiev@hotmail.com](mailto:Izhar_ananiev@hotmail.com)

**ליאור גל** [liorgal28@gmail.com](mailto:liorgal28@gmail.com)

**כפיר ביטון** [kfirbitonn@gmail.com](mailto:kfirbitonn@gmail.com)

**איתי עלמני** [itaialmani@gmail.com](mailto:itaialmani@gmail.com)

**רון בן-צבי** [benzvi.ron@gmail.com](mailto:benzvi.ron@gmail.com)

**הוגש בתאריך:** 23.11.17

1. תארו איך השתמשתם במודל Use Case בעבודתכם. הסבירו את מקומו ותרומתו של המודל לתהליך פיתוח המערכת בעזרת דוגמאות **פרטניות** (ספציפיות) מהמערכת "זר-לי".
2. תארו בפירוט איזה מרכיבים פונקציונליים ספציפיים של האופיין של מערכת "זר-לי" (כפי שמתואר במסמך "Semester Project") לא הצלחתם לבטא בעזרת מודל UC? מה מאפיין מרכיבים אלה מבחינה מהותית?
3. בתשובות 1 ו-2 תיארתם יתרונות ומגבלות שונות של מודל UC. הציעו דרכים להתגבר על המגבלות שציינתם, ונמקו למה הצעותיכם נותנות מענה למגבלות אלה.

הסבירו את תרומתה של הגישה שאתם מציעים כאן לפתרון אותן המגבלות שתיארתם תוך התייחסות ישירה למערכת "זר-לי" ובסיוע דוגמאות **פרטניות** (ספציפיות) מהמערכת.

**תשובות:**

1. במהלך ביצוע עבודתינו השתמשמנו במודל Use Case, המודל הנ"ל הינו חשוב מאוד ותורם רבות לתהליך פיתוח המערכת.

תרומתו הרבה של המודל באה לידי ביטוי בכך שתפקידו המרכזי של המודל הנ"ל הינו לתאר את הקשר שקיים בין Actors(שחקנים) שפועלים במערכת ל – Use Cases השונים.

כל Use Case מכיל סדר פעולות (Flow of events) המפרט את התנהגותו ותפקידו של כל Use Case המופיע במערכת.

מידול המערכת "זר-לי" באמצעות דיאגרמת Use Case תרם לנו רבות לצורך הבנת דרישות הלקוח בצורה הטובה ביותר לדוגמא:

1. מודל ה- Use Caseעזר לנו לראות בדיוק כמה Actors משתתפים במערכת או במילים אחרות כמה סוגי משתמשים צריכים להיות במערכת "זר-לי" (לקוח, מנהל מערכת, עובד בחנות, מומחה לניתוח סקר וכדומה).
2. כל Use Case המופיע במערכת מקושר לדרישות הרלוונטיות של הלקוח, כך למעשה לאחר סיום מידול מערכת "זר-לי" ראינו בדיוק שדרישות הלקוח טופלו לפי בקשתו.
3. מודל ה- Use Caseאפשר לנו לבצע הבנה ראשונית של המבנה הכולל של המערכת כלומר: מי הם המשתמשים, מהם הפעולות שהם יכולים לעשות (לקוח – מבצע הזמנה, עובד רשת – מזין תוצאות סקר וכדומה), מהם מאגרי המידע שנצטרך לתחזק (קטלוג מוצרים, נתוני לקוחות, נתוני הזמנות וכדומה).
4. באמצעות מודל ה-Use Case ניתן להציג תהליכים פנימיים של המערכת, לדוגמא: הפקת דוחות רבעוניים (מוצג כאליפסה ללא חץ מקשר לאחד מ- Actors במערכת).
5. כפי שכבר שנאמר בתוך כל Use Case במערכת "זר-לי" מתואר Flow of events שזוהי למעשה סדרת פעולות הן של המערכת והן של ה-Actor הרלוונטי. הדבר מסייע מאוד להבנה כיצד מבוצעות הפעולות השונות של המערכת (פתיחת חשבון, ביצוע הזמנה, תשלום באשראי וכדומה), הדבר יבוא לידי ביטוי בצורה משמעותית כאשר נגיע לשלב המימוש וכך למעשה נוודא שלא שכחנו שום דבר ולא מימשנו משהו סותר את דרישות הלקוח.
6. ישנם 2 סוגי מרכיבים פונקציונליים של המערכת אותם לא הצלחנו לממש באמצעות מודל ה–UC. הסוג הראשון הינו **דרישות המידע** של מערכת "זר-לי". דרישות מידע הן דרישות המתייחסות לישויות המידע בהן נדרשת התכנה לטפל כגון: מבני נתונים, בסיס נתונים וכדומה.

להלן דוגמאות ספציפיות עבור דרישות מידע של מערכת "זר-לי":

1. פלט - כאשר הלקוח רוצה לחפש מוצרים בקטלוג, על הנתונים המוצגים להישלף מבסיס הנתונים המתאים.
2. אחסון ושמירת נתונים – בכל יצירה של רשומה חדשה/ אובייקט חדש במערכת "זר-לי" יש לשמור את נתוניו ואת תוכנו במסד הנתונים המתאים, לדוגמא: סיום עסקה, חידוש מלאי, פתיחת חשבון לתשלום וכדומה.
3. עיבוד נתונים – בעת הפקת הדוחות הרבעוניים המערכת למעשה שולפת את הנתונים המתאימים לכל דו"ח, עלייה לעבד אותם, לבצע חישובים מסוימים ולהפיק את הדו"ח המבוקש.

שלושת התהליכים הספציפיים המתוארים הינם מרכיבים פונקציונליים של המערכת "זר-לי" אותם לא ניתן לממש באמצעות מודל ה- Use Case.

הסוג השני הינו **דרישות הממשק** של מערכת "זר-לי". דרישות ממשק הן דרישות המתייחסות לחילופי המידע או הבקרה בין התוכנה לבין ישויות חיצוניות, או בין ישויות פנימיות לבין עצמן כגון: סוגי מידע המועבר וממשקי משתמש כמו הפעלת כפתורים, תצוגות וכדומה.

בדיאגרמת ה-UC שלנו תיארנו את המקרים הכלליים שאפשר לבצע במערכת, אך לא את התהליך ברמה ספציפית. לדוגמא בתהליך ביצוע ההזמנה אנחנו יודעים מה עלינו להזין ומהו הקלט, אך אינינו יודעים מה קורה במערכת בזמן הזה ומה מתבצע על מנת שזה יקרה.

1. כפי שתיארנו בשאלה 2 קיימות מגבלות במודל ה- Use Case, כעת נציע את הדרכים שלנו בכדי להתגבר על המגבלות שתוארו:
2. בכדי להתגבר על המגבלות שתוארו במסגרת דרישות המידע של המערכת, נעזר בתרשים Class Diagram בשילוב עם מודל שלוש השכבות, מודלים אילו למעשה מתארים את מבני הנתונים של המערכת ויחסי הגומלין ביניהם.

בנוסף נציין כי בסיס הנתונים ייבנה בהתאם למבני הנתונים השונים המתוארים בתרשים ה-Class Diagram. בהתייחסות לדוגמאותינו, התרשים יכיל בתוכו בין היתר את המחלקות הבאות: עסקה, המלאים של המוצרים בחנויות, קטלוג (אוסף של מוצרים) וחשבונות של לקוחות, ולכל מחלקה יהיו המאפיינים (attributes).

על פי המחלקות האלו נוכל לבנות את הטבלאות המתאימות שיכללו בתוך מסד הנתונים שלנו. כעת, נוכל לדעת אילו פעולות, הקשורות למאפיינים, נרצה להפעיל.

בדוגמא של הפקת הדוחות, נוכל בעזרת התרשים, לדעת אילו מאפיינים נרצה לשלוף מתוך מסד הנתונים על מנת לעבדם ובעזרתם להפיק דוחות מתאימים.

1. בכדי להתגבר על המגבלות שתוארו במסגרת דרישות הממשק של המערכת, נעזר בתרשים Activity Diagram ובתרשים Sequence Diagram, מודלים אלו מתארים את התקשורת בין ה-Actors לבין המערכת.

תרשים ה-Activity יציין לנו באופן כללי את התהליך של כל UC, לדוגמא בתהליך ביצוע הזמנה נוכל לקבוע את שלבי ההזמנה: בחירת סוג המשלוח, בחירת זמני המשלוח/האיסוף, אופן ביצוע תשלום ולאחר מכן שמירת פרטי העסקה על ידי המערכת. נוכל לראות כי "עלינו שלב" ברמת הפירוט של תכנון התכנה. את הפונקציות שישויכו לכל מחלקה בתרשים המחלקות נבנה בעזרת ה-Activity Diagram.

תרשים ה-Sequence יציין לנו באופן מפורט את התהליך, הקורה במערכת, של כל פונקציה ראשית (לא פונקציות העזר-getters and setters) שמתוארת בתרשים המחלקות. לדוגמא בתהליך ביצוע הזמנה נתאר כל שלב בהזמנה בפירוט, בכך שניצור תרשים לכל שלב ושלב שיכיל בתוכו את התקשורת בין ה-Actors לבין המערכת.