**שאלות פתוחות – מטלה 1 – קבוצה 11**

**שם:** תומר גולדברג.

**שם:** אל-רז דהן

**שם:** עדן פאר

**שם:** מתן אוחיון

1. תיאור השימוש, תפקידו ומיקומו של מודל ה- Use Case:

בשלבי הראשוני של פיתוח המערכת, לאחר ביצוע ניתוח הדרישות של צורכי הארגון ולפני פיתוח המערכת עצמה, נעזר בדיאגרמת ה- Use Case לצורך מידול וסידור המערכת.

מודל זה הוא המודל הבסיסי של תהליך התכנון והפיתוח, על מודל זה נסתמך בהמשך לבניית המערכת.

מודל ה-Use Case מגדיר בצורה פונקציונאלית את מערכת המידע בה משתמשות הישויות הפועלות.

ב- Use Case תיארנו את הפעולות העיקריות שכל ישות יכולה ורשאית לבצע במערכת.

הדיאגרמה מציגה את הקשרים השונים שמתנהלים בין הישויות (קשרים הבאים לידי ביטוי בהורשה, ובהגדרת primary actor, supporting actor).

כמו כן, ניתן לראות בדיאגרמה את ההיררכיה בין ה-Use Cases הראשיים והמשניים, איזה פונקציה נובעת מאחרת, או לחילופין, תלויה בביצועה של פונקציה אחרת. (בא לידי ביטוי ב- (include, extend, generalization.

מעבר לדיאגרמה המוצגת, בחלק של ה- Flow of Events ניתן לקבל הסבר לגבי ביצוע התהליך.

דוגמאות פרטניות מהמערכת "OBL":

* לכל ישות הניגשת למערכת ישנן הרשאות שונות הקשורות לתפקידה במערכת ולכל אחת מהן יהיו פונקציות שונות במערכת, למשל:
* מנוי: השאלת ספר.
* מזכירה: הרשמה של מנוי חדש, השאלת ספר.
* מנהלת הסיפריה: שינוי הרשאות משתמשים.
* כל ישות הרוצה להשתמש במערכת עליה לבצע תחילה התחברות, לצורך כך ביצענו שימוש בהורשה ממנוי.
* קשרים בין ישויות:
* כאשר קורא מזדמן רוצה להירשם כמנוי, הספרנית מבצעת את תהליך הרישום.
* היררכיה בין Use Cases :
* כאשר מנוי מחפש ספר כלשהו הוא יכול (קשר של Extend) לצפות בתוכן העניינים של הספר.
* הספרנית יכולה לצפות בפרטי מנוי משתמש והיא יכולה (קשר של Extend) לשנות את פרטיו.

1. תארו בפירוט איזה מרכיבים פונקציונליים ספציפיים של האופיין של מערכת "OBL" (כפי שמתואר במסמך "Semester Project") לא הצלחתם לבטא בעזרת מודל UC?

* יכולים להיות מספר משתמשים שונים המחוברים למערכת בו זמנית.

אפיון- זהו תהליך פנימי של המערכת שלא מתבצע באופן ישיר על ידי אחד הActors, ולכן לא ממודל בUse Case.

* ניהול פרטי עובדים: זהו מרכיב פונקציונלי אשר מתקיים ומתוחזק ע"י מערכת נפרדת וחיצונית ולא מצאנו לכך ביטוי במודל שלנו אשר מייצג את המערכת.
* מדידת זמנים ומניית נתונים (למשל שלושה איחורים להחזרת ספר): זוהי פונקציונאליות פנימית של המערכת שלא באה לידי ביטוי בתרשים ה Use Case.

1. 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 "OBL" ובסיוע דוגמאות **פרטניות** (ספציפיות) מהמערכת.

* כל פונקציונליות המתקיימת בתור המערכת עצמה (למשל היכן נשמר המידע על הספרים/משתמשים, מדידת זמנים כמו למשל מדידת יומיים לקחת את הספר המוזמן על יד הקורא, ספירה של שלושה איחורים בהחזרת ספר) נוכל להוסיף use case שונים אשר יופיעו בתוך המערכת, אך לא יהיו מקושרים לאף actor, נוסף לכך נרשום ונשמור ייחוס של כל use case לactor שבו אנו מתכננים לבצע מימוש של use case אלה.

באופן זה, נוכל לדעת בעתיד של הפרויקט שלנו אילו דברים אנו צריכים לבצע במימוש וע"י הרישום נוכל לדעת באיזה actor צריך להתבצע מימוש פונקציונליות זו.

נוסף על כך, ע"י גישה זו לא נחסיר אף מידע פונקציונלי מהמערכת שלנו.

* בכדי להתגבר על המגבלות שתוארו במסגרת מרכיבי 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 השאלת ספר על ידי משתמש מנוי נוכל לקבוע את השלבים: בחירת ספר, בדיקה האם הספר זמין, ביצוע הזמנה על יד הספרנית, אופן הביצוע ולאחר מכן שמירת פרטי ההשאלה. נוכל לראות כי "עלינו שלב" ברמת הפירוט. 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 שמתוארת בתרשים המחלקות. לדוגמא בתהליך ביצוע הרשמה כמנוי נתאר כל שלב בהרשמה בפירוט, בכך שניצור תרשים לכל שלב ושלב שיכיל בתוכו את התקשורת בין ה-Actors לבין המערכת.