**Group 24**

|  |  |
| --- | --- |
| **Firas Khateeb** | **firask53@gmail.com** |
| **Rawad Mansour** | [**rawad\_90@hotmail.com**](mailto:rawad_90@hotmail.com) |
| **Ronen Abd-Alhak** | [**abdalhak.ronen@gmail.com**](mailto:abdalhak.ronen@gmail.com) |
| **Tameer Ghanem** | **tameer7595@outlook.com** |
| **Basel Khier** | [**basel.kh@outlook.com**](mailto:basel.kh@outlook.com) |

**Submitting date: 23/11/16**

**Time: 23:08**

שאלה 1

במהלך התכנון הייתה התלבטות לגבי שמות של נושאים.

בחרנו לאפשר לשני נושאים להיות עם אותו שם, אבל בתנאי שהם נמצאים בתחומים שונים.

מה שחייב אותנו להמציא דרך שתאפשר לנו לזהות כל נושא באופן חד ערכי. הפתרון שלהו היה להוסיף לכל נושא, בנוסף לשם של הנושא, מספר מזהה subjectID, הייחודי לכל נושא.

פתרון אחר שיכולנו להציע זה להכריז על נושא כישות חלשה של תחום, ובאופן זה הזיהוי שלה יהיה לפי שם נושא ושם תחום.

שאלה 2

(א) השתמשנו בעיקרון ההורשה. עקרון זה עזר לנו בלצמצם כפילויות, ובלתכנן מערכת שיותר פשוטה לתחזק. לדוגמה: יצרנו מחלקה בשם Person, שבה יש 3 שדות: ID, firstName, lastName. מחלקה זו הורישה להרבה מחלקות אחרות, מה שצמצם כפילויות באופן משמעותי.

(ב) לא השתמשנו בעיקרון הפולימורפיזם, כי עד עכשיו לא ראיני איך זה יבוא על ידי ביטוי בפועל. אבל אנחנו בטוחים שנשתמש בו במהלך המימוש.

שאלה 3

(א)

**1)** מודל ה- UML עזר לנו בלהשיג רמת תיאור דיי גבוהה.

למשל, ה- Sequence Diagram נתנה לנו את האפשרות לתאר איך ומתי המתודות תזומנו. אחת הדוגמאות לזה, תהליך הוספת ספר לנושא. שם אפשר לראות כמעט את כל הבדיקות הנעשות, לפני שהספר מוסף לנושא.

**2)** אחת הבעיות במודל ה- UML היא חוסר היכולת לתאר גרפית מראה של המערכת. למשל כשחשבנו על איך לתכנן את פעולת הוספת ביקורת על ספר, היינו צריכים לשרטט שרטוט מהמצאה שלנו, כדי שנוכל לדמיין את זה יותר טוב. Activity Diagram לבד לא הספיק לנו.

(ב) לא. יש אילוצים שהיה קשה לתאר אותם בעזרת UML. למשל שכאשר רוצים להתחבר ל- Server דרך Client, צריכים לדאוג שהחיבור קיים.