|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Email** |
| Taimor fares | 308137355 | [Fares.taimor@gmail.com](mailto:Fares.taimor@gmail.com) |
| Miras Safadi | 315891549 | [safadimiras@gmail.com](mailto:safadimiras@gmail.com) |
| Rani Hassan | 315060103 | [ranihassan8@gmail.com](mailto:ranihassan8@gmail.com) |
| Fares Jaraisy | 208451682 | [jaraisy.fares@gmail.com](mailto:jaraisy.fares@gmail.com) |
| Date Of Submission | 30/12/2018 | |

**שאלה 1:**

**תארו את תהליך התכנון שביצעתם לתהליך השאלה. פרטו מה הם הדילמות התכנוניות שהתייחסתם אליהם? מה היו החלופות ומה השיקולים לבחירת הפתרון שיושם ? בתשובה התייחסו לעקרונות שנלמדו בהרצאות :Pattern-** **Design, וReuse ,Design ,Architecture !!**

תיאור תהליך התכנון :

על מנת להשאיל ספר יש להגיע לספרנית עם הספר הרצוי כדי שהספרנית תוכל להשאיל אותו למנוי.

ולכן כשלב א' על המנוי לחפש את הספר בספריה דרך מערכת OBL שלנו ולבדוק האם ישנם עותקים זמינים שניתן להשאיל אותם או שישנו צורך לזמין את הספר הנ"ל,

נתאר כאן בקצרה תהליך של השאלה שכולל אי המצאתו של הספר בספרייה (שזה בעצן תהליך השאלה והזמנה):

1. מנוי מתחבר למערכת ומכניס שם משתמש וסיסמא.
2. מערכת בודקת את תקינות הנתונים מול ה-DB במקרה של אי נכונות מוצגת הודעה שגיאה מתאימה, בנוסף מערכת בודקת שסטטוס המנוי הינו פעיל שמאפשר לו שימוש במערכת (כלומר לא מוקפא ולא חסום), אם כלל הבדיקות עוברות בהצלחה המנוי מתחבר למערכת, כלומר נכנס לדף האישי שלו.
3. המנוי מחפש את הספר הרצוי בתיבת החיפוש (לפי שם, מחבר, נושא או תיאור חופשי)
4. המערכת מחפשת ב-DB את הספר הרצוי ובמקרה של תקינות קלט, מוצגים כלל הפרטים על הספר הרצוי. אם הספר לא נמצא מופיע כפתור שמאפשר הזמנה של הספר הנ"ל.
5. לצורך הדוגמא שלנו, הספר אינו זמין ולכן מוצג כפתור הזמנה, והמנוי לוחץ ומזמין ספר.
6. המערכת שומרת את הבקשה של ההזמנה ומנהלת את ההזמנות הכלליות, ובנוסף מופיעה ההזמנה בדף ההזמנות של המנוי על מנת שיוכל לערוך אותה לפי צורכו האישי (לבטל למשל).
7. ההזמנה מגיעה ומטופלת ע"י הספרנית, ועם הגעת הספר הנדרש נשלחת הודעה אוטומטית למנוי שיש להגיע תוך יומיים ולהשאיל את הספר.
8. המנוי מקבל הודעה ומגיע להשאיל את הספר.
9. המנוי מציג את מספר ההזמנה לספרנית וע"פ זה הספרנית מבצעת את תהליך ההשאלה עבור הספר הרצוי.
10. הספרנית מתחברת למערכת ונכנסת לחלון "השאלה" וכמניסה את הפרטים הנחוצים, מספר קטלוגי עבור הספר שהוא ייחודי ואת מספר המנוי שגם הוא מפתח, ואת תאריך ההחזרה ושעת ההחזרה ולוחצת על השאלה.
11. המערכת בודקת את כלל הנתונים שאכן נכונים ואפשריים, ולפי הצורך מציגה הודעה מתאימה, למשל עבור ספר מבוקש אם ההשאלה ליותר משלושה ימים אז היא מציגה הודעת שגיאה, או ספר רגיל מעל לשבועיים.
12. אם כלל הבדיקות עברו בהצלחה, ההשאלה נשמרת במערכת ובכרטיס הקורא של הסטודנט, כולל כלל המידע החיוני כגון שם ספרנית תאריך השאלה תאריך החזרה ומספר מנוי.
13. כלל המידע מתעדכן ונשמר ב-DB והמערכת כעשת מנהלת אתת ההשאלה.

אחת הדילמות שמעסיקה כל מתכנת בערך היא שכפול מידע, ובעקבות זה השתמשנו ב- Design Pattern :

Abstraction Occurrence Pattern: לעיתים אנו נתקלים בבעיה שישנם אובייקטים החולקים מידע משותף, אבל בנוסף לכך לכל אובייקט יש מידע מיוחד עבורו. בכדי לפתור בעיה זו השתמשנו ב- AOP , שאומר : ניצור אובייקט משותף שבו שומרים את המידע הזהה ובשאר האובייקטים השונים שומרים את המידע הייחודי, ובכך מונעים את שכפול המידע.

למשל עבור Book an CopyOfBook and BookInBorrow.

עוד דילמה שנתקלנו בה היא השימוש במתודות או מחלקות שהן כבר קיימות מוגדרות וממומשות בג'אווה, ולשם כך נשתמש ב-reuse אשר מאפשר לנו שימוש חוזר במה שכבר כתוב מוכן ונבדק מבלי לבזבז זמן יקר וחשוב בלכתוב את אותו הדבר שוב (אין סיבה להמציא את הגלגל מחדש).

**שאלה 2:**

ציינו אחד מהתכונות המאפשרות שימוש חוזר )Reuse ,)שקיומה מאפשר לכם לשלב קטעי קוד ומרכיבים אחרים שלא אתם כתבתם או תכננתם. תארו בדיוק )ובהתייחסות ספציפית( ובפירוט את התכונות המאפשרות של אותם מרכיבים אשר בחרתם לשלב במערכת שלכם, תוך התייחסות בדוגמאות ספציפיות )לא 'עקרוניות' או 'כלליות'( לדרישות הפונקציונליות של המערכת שתכננתם )התייחסות ספציפית בהקשר זה = התייחסות למרכיבים ספציפיים מתוך התיאור המילולי הראשוני של פעולת המערכת ששאתם מפתחים מהתחלת הסמסטר(. ציינו לפחות אחת מתוך קבוצת התכונות הנ"ל אשר לא באה לידי ביטוי בתכן שלכם - והסבירו את הסיבה לכך.

העקרון Reuse המאפשר בצורה מיטבית שילוב קטעי קוד ומרכיבים שלא כתבנו או תכננו הוא עקרון ההורשה.

ניתן להגיד שעקרון זה הוא מהחשובים ביותר, העקרון כידעו מאפשר למחלקה B להרחיב את מחלקה A כלומר לרשת את כול הפעולות והתכונות שלה ובנוסף להגדיר תכונות ומתודות ייחודית רק עבור המחלקה עצמה (מחלקה B).

המערכת שלנו מבסוות בעיקר על ממשק בין אדם למחשב שרוב הפעולות מתבצעות דרך מחלקות GUI, ולכן כאן נתייחס למחלקות אלה.

מחלקת GUI מרחיבה ויורשת ממחלקת Application, כלומר כל מחלקה של GUI צריכה הרחיב את המחלקה Application, כידוע המחלקה Application מוגדרת בשפה עצמה (כלומר כתובה ומוכנה)והיא מחלקה אבסטרקטית, שמכילה את המתודה בעיקרית start() שכול מחלקה שיורשת את המחלקה Application צריכה לממש את המתודב start().

ניתן לתת דוגמא עבור מחלקות ה-GUI של ה CS-(Client Server), ששתיהן מרחיבות את המחלקה Application אבל כל אחת ממשת את המתודה start() בצורה שונה.

גם ניתן להתייחס למחלקות שהרחבנו אותם ב-OCSF , כדי לממש CS.

העקון שלא בא עד עכשיו לידי ביטוי בקוד ובתכנון שלנו הוא עקרון הפולימורפזים, העקרון שמפעיל מתודה מסויימת רק בזמן ריצה כדי לדעת את סוג האובייקט, עקרון זה הינו חשוב מאוד, משום שהוא נותן לנו את האפשרות להפעיל את המתודה הנכונה על סוג טיפוס מסויים מבלי לשכפל קוד, ולכתוב מתודות רבות.

**שאלה 3:**

**א. הערכה כללית:**

**1. מהם היתרונות של מודל UML כעזר לתהליך התכנון?**

יתרונות של UML:

1. שפה קלה ללימוד, היא יותר מוחשית משפות אחרות (שאינן OOP), היא יותר ויזואלית מאחרות, הגישה שלה היא כמו הגישה היום יומית, לכל אובייקט (מחלקה) יש תכונות (אטרביוטים) שמייחדים אותו ופעולות (מתודות) שהוא יכול לבצע.
2. שפה שמאפשרת תיאור מופשט של מפרטי מערכת, כך שניתן להגיע לדיוק רב כי הסשפה היא מבוססת תרשימים, כך שניתן לתאר הכול בעזרת התרשימים.
3. כלל מתכנתי OOP מכירים את הדיאגרמות של UML וזה יכול להוות שפה משותפת חדה וברורה שמאפשרת תקשורת פנימית מיטבית בין כלל המתכנתים.

**2. הסבירו איך מתקבלים (מתממשים)היתרונות שציינתם.**

העיקרונות מתממשים כאשר מבצעים את הדיאגרמות שמהוות את הבסיס של OOP, מהדיאגרמה הראשונה ועד לאחרונה ניתן לראות את השיפור וההבנה שחלים אצל הקבוצה, הם מאפשרות לנו הבנה מעמיקה ומסודרת של כלל המערכת מנוקדת מבטו של הלקוח.

שלבי התכנון הם השלבים החשובים ביותר לדעתנו, אם בהתחלה נבין ונפנים את הדרישות כפי שהם מפורטות במסמך הדרישות, שלב הפיתוח יהי קל עבורנו.

הדבר התממש כאשר חלקנו את העבודה בין הצוות,(חלוקת העבודה הייתה בכל תת משימה כלומר בכול תרשים חלקנו את העבודה בינינו לא היה מצב שסטודנט אחד לא היה מעורב בתהליך כלשהו).

**3. ציינו דוגמה אחת קונקרטית (לא כללית) מתוך התהליך שאתם בצעתם לשימוש מועיל ב- UML תוך תיאור והתייחסות ספציפית למרכיבים של מערכת “OBL” שתכננתם ומידלתם.**

המידול היה בצורה מפורטת מדויקת ופשוטה, חלקנו את המערכת שלנו לחלקים שעבור כל חלק תארנו את הרצוי בדיאגרמה, וכל דיאגרמה הבהירה לנו דברים מוסתרים חשובים, למשל:

1. מידול תרחישים ופעולות וציון השחקים שבמערכת באצמעות UC. כלומר הבנו מה התרחישים האפשריים ועבור כל תרחיש מי השחקן שמבצע את זה.
2. מידול אובייקטים באמצעות class, כלומר הבנו איזה אובייקטים נכללים במערכת ומה כול אובייקט מבצע ועושה (מה שתארנו בשאלה קודמת). עבור ה-class חלקנו אותו ל-3 קבוצות:
3. Entity
4. Controller
5. Boundary
6. תיאור הpackages של כלל המערכת ע"י package diagram, שבו ציינו בכל חבילה איזה אובייקטים נכללים ומה הקשר בין חבילות שונות.
7. מידול התנהגות המערכת עבור כל פעולה שתארנו ב-UC בעזרת sequence diagram.
8. מידול התנהגות או זרימה של המערכת עבור כל דרישה/פעולה שתארנו ב-UC בעזרת activity diagram.

**ב. ניתוח ודיון:**

**בהתאם לניסיון שרכשתם במהלך העבודה על מטלה זו, תארו אפשרויות לשינויים ושיפורים במתודולוגית UML אשר נותנים מענה לחסרונות שנתקלתם בהם במהלך ה-design שביצעתם בפרויקט שלכם. הסבירו את תשובתכם תוך תיאור דוגמה ספציפית (כולל שמות של רכיבים) מתוך עבודתכם.**

בהתאם לנסיון שרכשנו במהלך העבודה הזאת, היינו מציעים לשפר:

1. נתקלנו פעמים רבות בצורך לשנות את דיאגרמת ה-class במהלך בניית ה-sequence, הרבה פעמים מצאנו את עצמנו מוסיפים מתודות לאובייקטים או אפילו לעיתים אובייקטים שלמים.

היינו רוצים שתהיה לנו אפשרות לעדכון אוטומטי כלומר אם נוסיף אובייקט חדש בsequence אז שיתווסף אוטומטית ל-class, דבר זה מקל רבות על העבודה במקביל ובצוות.

למשל: עבור הצגת תוכן עניינים של ספר, כאשר התחלנו לבצע את ה-sequence שמנו לב שאין לנו מחלקות להצגת פרטי ספר על מנת להתשמש במתודות של הצגת תוכן עניינים, ולכן הצטרכנו לחזור אחורה ולעדכן בהתאם, וכמובן דבר זה מבזבז את הזמן בצורה גבוהה.

1. היינו מציעים גם שברגע שמוסיפים גם אובייקט ב-sequence וכאשר אנו רוצים לבצע פעולה בין אובייקט A ל-B אז שיפתחו עבורנו רשימת המתודות האפשריות, לפי דיאגרמת ה-class שבנינו אותה, כך ניתן להבין יותר את סדר המתודות ואת היכולות שלנו.

למשל: בעת הוספת אובייקט ספר, וכאשר רוצים להוציא מהקו חיים שלו פעולה שהוא מבצע אז שתפתח עבורנו רשימת המתודות האפשריות.

1. בעבודת צוות מתחלקת העבודה בין אנשי הצוות , וכל אחד עובד במחשב האישי שלו, בסוף כאשר רצינו למזג את כלל העבודות נתקלנו בבעיות רבות במיזוג, הממשק של התוכנה אינו מופשט וקל היינו מתארים. יש תוכנות שהן יותר ידודיות למשתמש מאשר VP.