**קובץ שאלות:**

|  |  |  |
| --- | --- | --- |
| **שם** | **ת.ז.** | **דוא"ל** |
| ישראל וולדמן | 316113240 | voldmanisrael@gmail.com |
| אלעד מלכה | 034726596 | Eladmalka007@gmail.com |
| סרגי קובלצ'וק | 320409576 | Sergeikov95@gmail.com |
| אביב אלמקיאס | 205840150 | Aviv454@gmail.com |
| שירן חזן | 200402568 | Shiranhazan40@gmail.com |

1. תארו את תהליך התכנון שביצעתם ליישום מודל התמחור לשירותים. פרטו מה הם ה-Design issues שהתייחסתם אליהם? מה היו החלופות ולמה נבחר הפתרון שיושם?

**תשובה:**

התכנון של מערכת התמחור קרה במהלך התכנון של ה- Entities וחלוקת המידע בין המחלקות השונות. במהלך ישיבת הצוות, בחרנו במודל שלדעתנו נתן מענה טוב יותר הן לדרישות המערכת והן להרחבה עתידית של המערכת.

הבעיות שנתקלנו בהן:

-מימוש אלגוריתמים שונים לחלוטין לתמחור בצורה הכללית ויותר, המאפשרת מימוש זהה של שאר הקוד.

-מניעה של כפילות נתונים באחוזים במודלי התמחור.

-מימוש צורת תשלום שונה לחלוטין משאר הצורות (מנוי מס' 4).

-אפשור הרחבות למערכת התמחור בצורה הקלה ביותר שלא תדרוש שינויים רבים בקוד.

המודל שבחרנו, הוא מודל בו כל סוג של תמחור הוא מחלקה אשר יורשת ממחלקת Subscription כללי. לכל תמחור קיימת פונקציה ApplyDiscount שמפעילה אלגוריתם להנחה משלה לפי הדרישה של התמחור הנ"ל, ובכך ניתן להעביר מחיר לכל תמחור שהוא ללא צורך לדעת איזה אלגוריתם הופעל.

האלגוריתמים שולפים את ההנחות שהם צריכים מטבלת ההנחות ובכך נמנעת כפילות נתונים. במידה ומנהל השיווק משנה את המחירים, הם נשמרים בהעתק של טבלת ההנחות עד לאישור של מנהל הרשת.

התשלום החודשי אינו רלוונטי לחישוב המחירים, לכן הוא נוצר בתור אמצעי תשלום. לאחר הגדרת התמחור הרביעי, תבחר שיטת תשלום חודשית עבור המשתמש. ה- ApplyDiscount מחשב את המחירים לחודש הבא וה-Pay בתשלום החודשי מסמן את ההזמנה כממתינה לתשלום בסוף החודש (אותו ניתן לבצע ידנית ע"י בחירת אמצעי תשלום אחר).

המבנה מאפשר הוספת אפשרויות תמחור נוספות פשוט ע"י ירושה מ- subscription והגדלת טבלת ההנחות במידת הצורך.

מודלים שנדחו כללו subscription אחד שאובייקטים ממנו מכילים אחוזי הנחה (מקשה על מימוש אלגוריתמי תשלום), מודל subscription דומה לנוכחי ללא טבלת הנחות (מקשה על שינוי גודל ההנחה) וכן מבנה בו ה- subscription היה מחובר לcustomer (אפשרי אך עלול להקשות על חברות המעוניינות בתמחורים שונים.

2. א. ציינו עקרון של גישת התכן :Object-Oriented Design שבו עשיתם שימוש במהלך התכן (design) של הפרויקט. תארו בדיוק ובפירוט את תרומתו של עקרון בתהליך התכן שביצעתם ובתוצאתו הסופית, תוך התייחסות ספציפית )בניגוד ל'עקרונית'( לדרישות הפונקציונליות של המערכת שתכננתם.

ב. ציינו לפחות עקרון אחד של OOD שלא בא לידי ביטוי במודלים שלכם והסבירו למה.

**תשובה:**

א. השתמשנו בעיקרון התכנות של הפרד ומשול אשר לפי עיקרןן זה מידלנו את המערכת בחלקים. עיקרון זה תורם בכך שהוא מקל על המימוש ומאפשר עבודה מסודרת ויעילה. עיקרון זה בא לידי ביטוי אצלנו בחלוקת דיאגרמת המחלקות לפי מודל שלושת השכבות.

ישות (לדוגמה אצלנו FUEL TYPE), ממשק (קשר בין actor למערכת, לדוגמא אצלנו fuel type management screen), בקר (מקשר בין ממשק לישות ואחראי לביצוע תהליכים, למשל אצלנו fuel type controller).

כבר באב טיפוס חילקנו את הפרויקט לחבילות לפי עיקרון זה.

עיקרון נוסף הוא עיקרון ההורשה שניתן לייצג בעזרת דיאגרמת המחלקות ב- UML, למשל במערכת שלנו כל סוגי הלקוחות (לדוגמא car customer) יורשים מהמחלקה customer. עיקרון זה חוסך זמן פיתוח בהמשך עקב ביטוי פשוט יותר של הישויות ושל הקשרים ביניהם וגם מונע כפילות קוד.

ב. עיקרון שאינו בא לידי ביטוי במודל זה הוא עיקרון ה- override, דריסת מתודות מסוימות במחלקות יורשות מפאת אי היכולת לייצג זאת ב-UML .

3. א. הערכה כללית:

1) מהם היתרונות של מודל UML כעזר לתהליך התכנון? ציינו דוגמה אחת קונקרטית (לא כללית) מתוך התהליך שבצעתם לשימוש כזה בUML - תוך תיאור והתייחסות ספציפית למרכיבים במערכת שתכננתם ומידלתם.

2) ציינו קשיים הנובעים מחסרונות UML שנתקלתם בהם. הציעו פתרונות. גם כאן התייחסו ספציפית לתהליך שבצעתם.

ב. ניתוח ודיון: בהתאם לניסיון שרכשתם במהלך העבודה על מטלה זו, האם שימוש במתודולוגית UML נותן מענה מלא לצורכי תהליך ה design -שביצעתם בפרויקט שלכם? הסבירו את תשובתכם תוך תיאור דוגמה ספציפית (כולל שמות של רכיבים) מתוך עבודתכם.

**תשובה:**

3. א. 1) יתרונות מודל ה-UML כעזר לתהליך התכנון הם:

- מאפשר חלוקה גרפית של המחלקות השונות.

-מאפשר יצירה של דיאגרמות ברורות וקלות להבנה אשר ניתן להעביר ולהציג לאנשים אחרים כהסבר למערכת.

- נותן אפשרות לצפות את תהליך הפיתוח ולהתכונן בהתאם, דבר שיקצר את זמן הפיתוח.

-בעזרת ACTIVITY ו- SEQUENCE ניתן להראות רצף של אירועים, כלומר, אילו מתודות מתרחשות ובאיזה סדר.

- מאפשר למדל תוך כדי שימוש בחלק מעקרונות הOOP-, מה שמבטיח תכנון ועבודה נכונה.

למשל במערכת שלנו, בעזרת שימוש בהורשה אשר מוצגת בUML- דאגנו שכל הלקוחות השונים (למשלcar customer) יורשים מישות כללית של customer, כך אנחנו עוד בשלב הdesign - דואגים לעבודה נכונה בקוד ומונעים שכפול קוד של פונקציות משותפות דבר שגורם לדיאגרמה להיות ברורה ומסודרת.

2) הקושי במידול UML הוא שלא מתאפשר מידול של ממשקי משתמש בצורה ברורה, דבר זה יכול לגרום לאי סדר בפיתוח ולאי הבנות מה שיכול להאט את הפיתוח. למשל בדיאגרמה שלנו לא הצלחנו לייצג איך חלון ה- house fuel ordering window ייראה (גודל, צבעים, סדר כפתורים וכו'), פתרון אפשרי יהיה ליצור תמונה של איך הממשק אמור להיראות ולהעלות אותה לתוכנה מה שיעזור להבנה.

ב. המידול בעזרת UML נותן מענה חלקי לצורכי פיתוח הפרויקט, ה- UML נותן מענה לחלק התיאורי של המערכת, כלומר, אילו ישויות ישנן, הקשרים ביניהן וסדר הפעולות. אך לא ניתן לראות את נראות המערכת בפועל, למשל המסך house fuel ordering window קיים במערכת ומתאר את הפעולות והקשרים הרצויים אך אי אפשר לדעת איך החלון ייראה או ייוצג בפועל (גודל, צבעים, סדר כפתורים וכו').