

מטלה 2

מספר קבוצה 1G

חברי הצוות

מיכאל גינדין 316111491 mihael1023@gmail.com

אור מן 316377316 ormn1996@gmail.com

ניר פיקן 206231920 nirpikk@gmail.com

רומן קוזק 323533745 mirage164@gmail.com

אביב ונונו 313595449 avivvanun@gmail.com

תאריך הגשה : 6.12.2020

מסמך תשובות:

1. תארו את תהליך הניתוח שביצעתם למרכיב: ניהול ביקור לא מתוכנן (מזדמן).
קווים מנחים לתשובה: פרטו מה הם השאלות/הפרטים שהתייחסתם אליהם בתהליך הניתוח והחשיבה, וכן התייחסו לקשרים ולמעברים 1. ממודל Use-case למודל תהליכי המיוצג בעזרת Activity Diagram, 2. ממודל התהליכים לקראת מימושו בתוכנה.

תשובה 1:

ראשית ביצענו Use Case על מנת להבין מי השחקנים שלנו ואילו פעולות כל אחד מהם מבצע, לאחר מכן כשבנינו את ה activity diagram השתמשנו בשחקנים העיקריים (עובד הפארק והמבקר) על מנת להבין מי ה actor שמתחיל את התהליך, זיהינו כי המבקר מתחיל את התהליך אך עובד הפארק הוא מי שמבצע אותו בפועל.

על מנת לממש את התהליך של ביקור לא מתוכנן שמנו לב כי עלינו לממש פונקציונאליות המאפשרת מעקב אחר כמות האנשים בפארק. בנוסף, לא היינו בטוחים האם כדי לממש הזמנה של מבקר לא מתוכנן האם צריך להפריד את הביקורים המתוכננים והלא מתוכננים לאובייקטים שונים. לבסוף החלטנו ליצור אובייקט כללי המתאר כניסה לפארק ששני הביקורים משתמשים בו ואובייקט של הזמנה לביקור מתוכנן בלבד.

2. בהרצאה הוגדרה **Reusability** כתכונה של תוצר של תהליך הפיתוח אשר משקפת את היכולת לבצע reuse בהקשר לתוצר זה. בהתאם להגדרה זו, תארו יישום של 3 התכונות המאפשרות לכם לשלב במערכת **GoNature** שאתם מפתחים קטעי קוד ומרכיבים אחרים שלא אתם כתבתם או תכנתם.

תארו בדיוק (ובהתייחסות ספציפית) ובפירוט את התכונות המאפשרות Reuse של אותם מרכיבים אשר בחרתם לשלב במערכת שלכם, תוך התייחסות בדוגמאות ספציפיות (לא 'עקרוניות' או 'כלליות') לדרישות הפונקציונליות של המערכת שתכנתם (ההתייחסות ספציפית בהקשר זה = התייחסות למרכיבים ספציפיים מתוך התיאור המילולי הראשוני של פעולת המערכת ששאתם מפתחים מהתחלת הסמסטר. לא כולל תהליך זיהוי משתמש).
אם יש מי מ-3 התכונות הנ"ל אשר לא באה לידי ביטוי ב-reuse שביצעתם - הסבירו את הסיבה לכך.

GSON – ספרייה המשמשת ל serialization של אובייקטים.
אנו משתמשים בספרייה זו על מנת לשלוח אובייקטים שיצרנו בצורה של מחרוזת (String).
השליחה מתבצעת מצד client לצד server ולהפך.

יישום תכונות של reusability :

Availability - הספרייה היא ספריית Opensource של חברת גוגל הנגישה ב GitHub
(<https://github.com/google/gson>)

Understandability – הספרייה משתמשת במטודות פשוטות מאד ואינטואיטיביות כגון (FromJson ToJson).

Flexibility – הספרייה מסוגלת להתאים לכל סוג של אובייקט class, גם אלו שכתבנו בעצמנו. הספרייה כוללת אובייקט מסוג GsonBuilder המאפשר להתאים את עבודת ההמרה לפי דרישותינו (כתיבת null, עבודה עם פולימורפיזם ועוד..)

דרישה שreuse עוזרת לו היא ההעברת כל סוג אובייקט מהשרת ללקוח ולהפך.

לדוגמא באב טיפוס היינו צריכים לשלוח אובייקט מסוג Visitor מהשרת ללקוח

3. א. הערכה כללית:

1. מהם היתרונות של מודל UML כעזר לתהליך התכנון?

- (i) הסבירו איך מתקבלים (מתממשים) היתרונות שציינתם.
- (ii) ציינו דוגמה אחת קונקרטית ממוקדת (לא כללית ולא Login) מתוך תהליך הניתוח והתכנון שאתם בצעתם לשימוש מועיל ב-UML תוך תיאור והתייחסות ספציפית למרכיבים של מערכת "GoNature" שתכננתם ומידלתם.

היתרונות של מודל ה UML כעזר לתהליך התכנון הם :

- זיהוי בעיות לוגיות לפני פיתוח
- זיהוי שדות ומטודות נדרשות למחלקות שונות.
- זיהוי אחריות של controller-ים שונים.
- הבנת התקשורת בין רכיבים שונים בפרויקט.
- תיאום הפיתוח בין מפתחים רבים
- בניית הבסיס לפיתוח הפרויקט.

- (i) בפרויקט שלנו היתרונות שציינו מתממשים ע"י יצירת class diagram מפורט אשר דואג כי כל המפתחים יעבדו בקו מחשבה אחיד ויבינו את הצורך של כל רכיב בפרויקט. בזמן יצירת הדיאגרמות גילינו בעיות לוגיות בתכנון שלנו אשר דרשו שינוי בחלק מן המחלקות בפרויקט שמנעו תקלות בעת המימוש.
- (ii) דוגמה לכך היא שדה המחיר בהזמנה ובביקור, גילינו כי הלוגיקה שלנו מחשבת בזמן אמת את מחיר ההזמנה בהתאם למבצעים הקיימים אך לא שומרת את מחיר זה בשום מקום ולכן לא יכולנו לממש את דוח הרווחים, ללא שדה זה המחירים של הביקורים לא היו כוללים את המבצעים במועד ההזמנה.

2. ציינו קשיים הנובעים מחסרונות של UML שנתקלתם בהם. גם כאן התייחסו ספציפית לתהליך שבצעתם לפיתוח מערכת זו.

נתקלנו בבעיה זהה בתרשימי ה activity וה- sequence בתיאור של תהליך גדול (לדוגמא תהליך יצירת ההזמנה) התרשימים גדולים ומסורבלים מדי . ולכן לצורך הפשטה של התרשים יש צורך לחילוקו לתתי דיאגרמות על מנת שהתרשימים יהיו קריאים ומובנים יותר והמעקב עליהם יהיה פשוט וטבעי יותר.

חסרון כללי בתרשימי ה UML הוא חוסר היכולת להביע את כל המידע הנדרש על המערכת בתרשים בודד באופן מסודר, לכן נדרשנו להשתמש במספר תרשימים שונים.

בניית UML -ים מרובים דרשה מאיתנו לעבוד בסנכרון על התרשימים השונים, מה שהקשה על עבודתנו והאט את קצב העבודה שלנו.

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, לא כולל Login) מתוך עבודתכם.

בהתאם לבעיה שהועלתה למעלה בהקשר של דיאגרמות sequence , activity גדולים מדי, נציע את הפתרון הבא :

חלוקה של התהליך שאנו מציגים בתרשימים לתתי תהליכים, עבור כל אחד מתהליכים אלו ניתן ליצור דיאגרמה עצמאית וכך התרשימים יהיו קטנים וברורים יותר. נקח לדוגמא את תרשים "ביצוע הזמנה חדשה" ניתן היה לחלק אותו לתתי תרשימים של:

-הצגת רשימת הזמנות פנויות

-בדיקת הזמנה חדשה

-סיום התהליך (רישום ההזמנה במערכת)

בנוסף, בקשר לבעיה של זרימת האירועים ב class diagram :

ניתן להוסיף קשרים בין מחלקות המייצגים קשרים של שימוש של מחלקה אחת במטודות של השנייה. לדוגמא: מנהל ההזמנות (order controller) משתמש במטודות של ערכים של הפארק מתוך park controller.