**קובץ שאלות – קבוצה G1**

|  |  |  |
| --- | --- | --- |
| **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 |

תאריך הגשה: 17-01-2016.

**שאלות:**

1. תארו את תהליך השיקולים וקבלת ההחלטות שלכם בתכן design)) המערכת לטיפול בסוגי הלקוחות השונים – פרטי ו"רכב חברה". תארו דילמות ספציפיות (לא כלליות) מתוך תכן המערכת שפיתחתם בהקשר זה והסבירו את הפתרונות שבחרם
2. תארו את תהליכי הבדיקות שבצעתם במהלך פיתוח הפרויקט שלכם. ציינו את המרכיבים המאפיינים של התהליכים, בהתייחס לנלמד בהרצאות, ותוך מתן דוגמאות ספציפיות שביצעתם )או לא ביצעתם) בפועל במהלך הפרויקט (בהקשר של מרכיבי המערכת הספציפית שפיתחתם).
3. תחקור והפקת לקחים: התייחסו לאופן שבו התנהלתם לגבי 2 מרכיבים של הפרויקט:
4. פיתוח צוותי משותף וניהול גרסאות: ציינו יתרונות וחסרונות של השיטה שלפיה פעלתם (יש להתייחס לתהליך - לא להתמקד רק באספקטים טכניים).
5. שילובי קוד (לאחר הפיתוח - אינטגרציה( ובדיקות. אם היו קשיים מה הסיבה לכך? מה הייתם משנים בדיעבד?

**תשובות:**

1. במהלך תכנון הפרויקט בדקנו אפשרויות שונות למימוש לקוח פרטי ולקוח חברה. בהתחלה ניסינו מודל של שני לקוחות שונים שיורשים מלקוח כללי: קלאס נפרד לכל סוג. מודל זה דרש מימוש נפרד לכל סוג לקוח. אם זאת במהלך התכנון של הפרויקט שקלנו לאחד את שני הלקוחות ללקוח אחד.

איחוד הלקוחות יוצר כמה דילמות – האדם שמבצע את התדלוק הוא לא בהכרח האדם שמשלם עליו. בנוסף, ללקוחות חברה קיימות מכוניות רבות לעומת לקוח פרטי. לבסוף חלק מהדו"חות מבקשים להפריד בין לקוח חברה ללקוח פרטי (הדו"ח של המערכת האנליטית).

החלטנו בסוף לאחד את שני הלקוחות למחלקת לקוח אחת – לקוח מכונית, ולהפריד בין לקוח פרטי ללקוח חברה על ידי מחלקה אחרת – מחלקת סוג משתמש:

כל משתמש מחובר לסוג משתמש שקובע את התפקיד שהוא משחק במערכת וכן את ההרשאות של אותו משתמש. על ידי הגדרת סוגי משתמש (סוגי לקוחות) שונים ללקוח חברה ולקוח פרטי ניתן להבדיל ביניהם במידת הצורך (למשל ביצירת דו"ח עבור סוגי לקוחות) ולאחד את הזמנות הדלק והצגתן. כל לקוח מכונית יכול להיות בעלים של כמה מכוניות (מה שמאפשר גם מימוש לקוח חברה וגם לקוחות פרטיים עם כמה מכוניות) והתשלום מתבצע עבור המשתמש הרשום בתור לקוח המכונית (עבור לקוח פרטי זה הנהג עצמו ואילו עבור לקוח חברה זה איש קשר מסוים המטפל בחיובי התשלום של החברה).

1. במהלך הפיתוח כל איש צוות לקח העתק של הפרויקט למחשב שלו. בסוף העבודה איש הצוות היה מאחד את הקבצים ששונו לתוך הפרויקט הראשי תוך יצירת העתק גיבוי. לפני האיחוד על כל איש צוות היה לבדוק את העבודה שביצע ולתקן כל שגיאה שמתגלה, ככה הובטח כי כל חלק שנוסף לפרויקט פעל בפני עצמו. כל כמה ימים אנשי הצוות בדקו כי הקטעים שהם כתבו עדיין פועלים ובכך נמנעות בעיות בנקודות החיבור במעוד מועד.

כל חלק נבדק בשיטת מלמטה למעלה בגלל צורת המימוש של הפרויקט מלמטה למעלה. עבור כל חלק נבדק כי הוא מעבד את המידע שהוא מקבל כנדרש ואילו תקינות המידע המתקבל והשימוש בו הלאה הוא באחריות איש הצוות הבא שמממש את הקטע שמעליו.

נתייחס לדוגמא לבדיקת המתודה generateCarFuelOrder המייצרת הזמת דלק חדשה ומורידה את ההזמנה מהמלאי. במהלך הביצוע המתודה קוראת ל-createCarFuelOrder המייצרת הזמנה בדטבייס וכן reduceFromStock המורידה את ההזמנה ממלאי הדלק. תכולת המתודה נבדקה בשיטת קופסת הזכוכית אך המתודות reduceFromStock ן- createCarFuelOrder לא נבדקו. במהלך הבדיקה התגלה כי reduce קורסת במהלך הריצה. האדם האחראי על reduce בדק את המתודה עם הקלט שהמתודה generate יצרה וגילה כי המתודה קרסה בגלל ש-reduce משתמשת בקלט בפורמט אחר ממה שנשלח, הבעיה תוקנה ו-generate נוסף למערכת.

1. א) ניהול הפרויקט מתבצע בקובץ אקסל של גוגל דוק המכיל כמה טבלאות לניהול, מעקב ותחזוקת הקוד: -טבלת אנטיטי העוקבת אחרי שלבי הפיתוח של האנטיטי מכתיבת האנטיטי עצמו, הבאתו, תחזוקתו ויצירתו בדטבייס עד הפונקציות הניהוליות בקליינטסייד.

* טבלת חלונות המראה את ההתקדמות בפיתוח הפונקציונליות מאחורי הפרויקט.
* טבלת changelog המראה אילו קבצים שונו, איך מתי.
* טבלת דגשים המסבירה על התכן הכללי של המערכת ושיטות העבודה (עדכון טבלאות, העלאת גרסאות וכו').

ניהול גרסאות בוצע בדרופבוקס: כל העלאת גרסה כללה תיעוד מלא של כל השינויים שבוצעו בגוגל דוק, יצירת גרסת גיבויי והעתקת רק הקבצים ששונו.

* היתרון של השיטה הוא מניעה של דריסת עבודה של אנשים אחרים כיוון שרק הקבצים ששונו מועלים ושני אנשי צוות לא עובדים על אותו קובץ. התיעוד מאפשר לגלות מאיפה נובעות השגיאות שנוצרו באיחוד והגיבוי מאפשר לתקן את הפרויקט הראשי במידת הצורך. המעקב אחרי ההקדמות בגוגל דוק מונע עבודה על אותם קבצים ומאפשר מעקב ושינוי חלוקת העבודה במידה וההתקדמות לא מספקת.
* החיסרון הוא ששני אנשים לא יכלו לעבוד על אותו קובץ וגם יש צורך לתעד כל שינויי ולדייק מאוד לעקוב אחרי השיטה כדי למנוע דריסת נתונים. נוסף לכך נדרשת לקיחה תמידית של הפרויקט העדכני למחשב, אפילו תוך כדי עבודה על חלק מסוים.

ב) בגלל שהאיחוד קרה באופן תמידי תוך כדי העבודה נדרשה רק בדיקה אחרונה עבור הגרסה האחרונה. הקשיים העיקריים נבעו מעבודה מול קבצים גדולים שהכילו קוד רב שהיוו צבר בקבוק במהלך הפיתוח. בדיעבד היינו מפצלים את הקבצים הגדולים יותר לתת קבצים כדי לשפר את מקבילות העבודה.