מטלה 3 – שאלות פתוחות

קבוצה 20		
תאריך הגשה: **21/06/2020**		
שם מלא	ת"ז	מייל
איילון בן סימון	312162951	aylon8@gmail.com
סער ויקטור	312392822	saarms5@gmail.com
שובל דוד	305284945	shovaldavid877@gmail.com
שירז קרוב	313599987	shirazkarov@gmail.com
גל טולדנו	205886559	gal1010100@gmail.com

שאלה 1
תארו את תהליך התכן (Design) שביצעתם עבור המערכת " MyFuel " באופן הבא:
א. בהקשר של יכולות מעקב פעילות ע"י מחלקת השיווק, תארו דילמות הנדסיות ספציפיות שעסקתם בהן בתכן של יכולות אלה. השתמשו בפורמט של תיאור Design issue כפי שמופיע בהרצאה על Design.
תארו את הדילמות, השיקולים וקבלת ההחלטות שלכם והסבירו את הפתרונות שבחרתם.

דילמה ראשונה הייתה, איך לתת את הדירוגים לכל לקוח, באיזה אלגוריתם להשתמש ואיך. תחילה חשבנו שעבור כל שילוב של סוג לקוח, שעות התידלוק וסוג הדלק שנרכש ישנו דירוג שאותו יש להציג, אך הבנו לאחר מכן, שלא זאת הייתה הכוונה. לפי הסיפור, המערכת האנליטית מנתחת את נתונים אלו ומפיקה דירוג בהתאם עבור כל לקוח, כלומר לכל לקוח צריך להיות דירוג משלו. לשם כך חשבנו על אלגוריתם שעבור כל קומבינציה של סוג לקוח, שעת תדלוק וסוג דלק, מחושב דירוג מתאים והדירוג הזה מוצג עבור כל לקוח במערכת.
דילמה נוספת שנתקלנו בה הייתה עדכון טבלת הדירוגים ב DATABASE. בעצם כל שבוע צריכים להיות מוזנים נתוני הלקוחות למערכת האנליטית עבור ניתוח והפקת דירוגים. מכיוון שהטבלה קיימת עבור מספר לקוחות מסוים, בשבוע הבא כשאר יופקו דירוגים חדשים, יכולים להתווסף לטבלה עוד לקוחות וכן, גם לדרוס ערכים קודמים, מה שהקשה על עדכון. פתרנו זאת על ידי כך שכל פעם שמופקים דירוגים, הטבלה ב DATABASE נמחקת, ונכנסים כל הדירוגים מחדש וככה בעצם התגברנו על הבעיה שבה היה צריך גם לעדכן וגם להכניס ל DATABASE.

ב. ציינו איזו מהעקרונות שנלמדו בהרצאות הקורס בנושאים: Architecture, Design, Reuse ו- Design patterns באו לידי ביטוי בתהליך התכן הכלל מערכתי שביצעתם והסבירו באיזה אופן - באמצעות דוגמאות רלבנטיות וספציפית מתוך המערכת "MyFuel".

במהלך תהליך התכן הכלל מערכתי שביצענו, עיקרון מרכזי שלמדנו הבא לידי ביטוי במערכת שלנו הינו עקרון ה Reuse. במקרים מסוימים היינו צריכים לממש פונקציונליות שכבר קיימת ומומשה על ידי מתכנתים מנוסים מאיתנו. עקרון זה בא לידי ביטוי באופן מימוש שליחת המייל בעת רכישה (דלק לחימום ביתי \ מוצרים אחרים) ושליחת קבלה למייל.
עקרון נוסף הוא Design pattern מסוג Observable. בכל רכישת דלק מתבצע עדכון המלאי באותה תחנה ובמידה והכמות עברה את רמת הסף שנקבעה, מופקת הזמנה אוטומטית על ידי המערכת. כלומר, "הצופה" הוא קונטרולר התדלוק ו"הנצפה" הוא כמות המלאי בתחנה מסוימת.
נעזרנו בהרצאות ולמדנו הרבה מן המצגת בנושא architecture שמציגה בפירוט מאילו חלקים ותתי חלקים מורכבת מערכת וכיצד יש לבנות בסיס המתאים לפרויקט. בעקבות הידע שצברנו, ראשית התחלנו בניסיונות להפוך את האב טיפוס לתוכנה שתתמוך באפשרויות מרובות, עם חלוקה נכונה לחבילות ומחלקות האחראיות על נושאים ספציפיים (מודולים). בסופו של דבר הבאנו את הקוד למצב של תקשורת קבועה בין השרת ללקוח. באמצעות עקרונות אלו שנלמדו בהרצאה הצלחנו בסופו של דבר לחבר בין כל החלקים השונים במערכת שלנו בצורה טובה ומסודרת.

שאלה 2

תארו את תהליכי הבדיקות השונים שבצעתם במהלך פיתוח הפרויקט שלכם. ציינו את מאפייני תהליכי הבדיקות שביצעתם תוך התייחסות לעקרונות שנלמדו בהרצאות בנושאי בדיקות תוכנה, ותוך מתן דוגמאות ספציפיות (לא כללית/גנרית ולא Login) שביצעתם (או לא ביצעתם) במהלך הפרויקט (ע"י תיאור מפורט של בדיקות מרכיבים ספציפיים של מערכת "MyFuel").

במהלך העבודה על הפרוייקט ביצענו בדיקות רבות שהמטרה העיקרית שלהן היא בדיקת נכונות הקוד וממשק המשתמש שנכתב בעת שילוב קטעי הקוד בין חברי הצוות השונים.
"קופסה שחורה" – בחינת התפקוד של היחידה בתוך המערכת. הסתכלנו על 2 פרמטרים:

1. נכונות הפלט / התגובה – האם הערכים חוקיים או לא.
2. מהירות התגובה.

ביצענו את הבדיקות הללו בתהליכים המורכבים מהזנת נתונים בטופס אחד ומעבר איתם ואת אופן השפעתם לטופס אחר. למשל, בתהליך רישום לקוח ושיוך רכב היינו צריכים להמשיך עם תעודת הזהות של אותו לקוח וסוג המנוי אותו בחר. במידה ובחר במנוי multiple monthly, חייבנו את נציג השיווק לשייך לאותו לקוח יותר מרכב אחד. כלומר, הפלט שלו ציפינו לאחר רישום הלקוח היה חיוב שיוך של לפחות 2 רכבים, כמו כן, שתעודת הזהות של אותו לקוח תופיע בטפסי השיוך.

"קופסה לבנה" - בחינת המבנה הפנימי של היחידה, וכעת מסתכלים על 3 פרמטרים:

1. מסלולי החישוב.
2. נכונות החישובים.
3. נכונות ההחלטות הלוגיות.

ערכנו בדיקות אלו על מנת לבדוק שפונקציות המורכבות מחישובים מסוימים עובדות כמצופה. למשל, בתהליך התדלוק, המחיר הסופי מחושב לפי הפרמטרים הבאים: סוג מנוי הלקוח, תכנית רכישה של הלקוח והאם קיים מבצע פעיל במערכת בזמן הרכישה. בדקנו כי המחיר המשוקלל באמת תואם את המחיר לאחר שקלול כל הפרמטרים.

שאלה 3
תחקור והפקת לקחים: התייחסו לאופן שבו התנהלתם לגבי 2 מרכיבים של ביצוע הפרויקט:
א. תיאום פעילויות ושיתוף בין חברי הצוות בפיתוח וגישה לניהול גרסאות: תארו את השיטה שלפיה פעלתם בהקשרים אלה, וציינו יתרונות וחסרונות שלה. יש להתייחס גם לתהליך העבודה לא להתמקד רק בכלים ואספקטים טכניים.

הפרויקט התחלק לשני שלבים.
בשלב הראשוני של הפרויקט שבו עבדנו על תכנון המערכת, השתמשנו ב visual paradigm.
את שלב התכנון ביצענו יחד כל חברי הפרויקט על מנת שתהיה הסכמה כוללת על איך המערכת תראה באופן כללי.

בשלב הבא, בו עסקנו בפיתוח המערכת היה עלינו לשתף את כתיבת הקוד באופן רציף, לשם כך עבדנו בדרך של חלוקת מטלות בין חברי הקבוצה ואיחוד הקוד של המטלות הגמורות, בעזרת העלאה ל drive ולאחר האיחוד, העלאה מחודשת של הגרסה העדכנית.
למרות ההמלצה להשתמש ב Git, בחרנו לא להשתמש בזה מכיוון ש מערכת Git אינה פשוטה לתפעול ודרשה השקעת זמן למידה על אופן הפעולה שלה, זמן יקר אותו העדפנו להשקיע בפיתוח וקידום הפרויקט.

יתרונות:
- חסך זמן למידה והתרגלות לדרכים אחרות (למשל Git).
- שלב האינטגרציה (מיזוג הקוד) היה קל ונוח למימוש בתצורה שהוצגה לעיל מפני שכל מה שהיה נדרש הוא להוריד את הקובץ אותו חבר הצוות עבד עליו שרצינו להוסיף לגרסה הנוכחית והדבקה בקוד, תוך טיפול מינורי בבעיות שנוצרו.

חסרונות:
- לא היה ניתן לשחזר גרסאות קודמות מכיוון שכל פעם החלפנו את הפרויקט ב drive בפרויקט העדכני.
- בתצורת עבודה זו, נוצרו הרבה שכפולי קוד מכיוון שכל אחד עבד על הקובץ אותו הוריד ולא היה ניתן לעקוב בזמן אמת על פונקציות משותפות לכמה חברי צוות. דבר זה טופל בזמן איחוד הקוד.

ב. שילובי קוד (אינטגרציה מערכתית - לאחר הפיתוח הראשוני) ובדיקות. ציינו באופן פרטני, בהתייחסות ספציפית לפיתוח המערכת "MyFuel", איך פעלתם בשלב זה של הפיתוח (למשל: תיאור התנהלות התהליך, אופן טיפול בבעיות, וכו').

אם היו קשיים מה הסיבה לכך? מה הייתם משנים בדיעבד בגישתכם למרכיב זה של תהליך הפיתוח מבחינת האספקטים הרלבנטיים של הנדסת תוכנה?

בתחילת המטלה השלישית ביצענו מפגש קבוצתי בו עשינו תיאום ציפיות מבחינת עיצוב המערכת, חלוקת הקוד לחבילות לפי נושאים (לפי תרשים המחלקות ב visual paradigm) וחלוקת העבודה בין חברי הצוות.
בנוסף, שמנו דגש על מספר כללים שהנחו אותנו במהלך המטלה:

- כל המחלקות בוזרו בהתאם לתצורה אותה קבענו בשלב הראשוני של המטלה ובהתאם לתרשים המחלקות.
- חילקנו את העבודה בין חברי הצוות לזוגות ויחידים לנושאים הקשורים אחד לשני כדי להקל ולבצע עבודה מסודרת כמה שניתן, ושכל חבר צוות יבין את הנושא בו עסק לעומק וידע להסביר לשאר הקבוצה.
- במידה והיה שינוי במרכיב הנוגע לכלל חברי הצוות, בוצע עדכון של חברי הצוות.
- נתנו שמות משמעותיים לפונקציות ולמשתנים על מנת שכל חבר צוות יוכל להבין ולהתמצא בקוד ללא עזרה.
- במידה ומישהו היה צריך עזרה, הוא פנה לשאר חברי הצוות והתייעץ בנוגע לבעיה שלו.

שימוש בכללים אלה הביא לכך שהיה תיאום באופן כתיבת הקוד והציפייה ההדדית בין חברי הצוות.

הקשיים בהם נתקלנו היו שבעת שלב האינטגרציה, במידה והייתה טעות קומפילציה לעיתים היה קשה למצוא את השגיאה ומאיזו סיבה היא נובעת.
שינויים שהיינו מבצעים בדיעבד הם:
עבודה עם שימוש ב Git על מנת להכיר ולהתוודע לכלי זה שכנראה ישמש אותנו בעתיד. כמו כן, ללמוד על Git עוד לפני קבלת מטלה 3 כדי להיות מוכנים ולבזבז כמה שפחות זמן על למידה והכירות.