

קבוצה מס – 19

Hana Ammar:207360561 hanaammar12345@gmail.com

Nasmat Ammar:208166371 nasmat.ammar@gmail.com

Liran Haim: 308292382 liranhaim77@gmail.com

Fadi Harb: 316239151 hfadi3011@gmail.com

natali damti: 311274542 natalidamti@gmail.com

Sapir dayan: 313905776 sapirdayan@gmail.com

שאלה 1

תארו את תהליך התכן (design) שביצעתם עבור המערכת " GoNature "
באופן הבא :

(א) יכולות **בקרת כניסה לפארק** בהקשר של ביקור מוזמן (מתוכנן) ולא מתוכנן (מזדמן).

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.

בתהליכים הראשוניים של תכנון המערכת ובנייתה נדרש מאיתנו התייחסות גדולה לעניין העיצוב (במיוחד עבור יכולת בקרת כניסה לפארק), מכיוון שהעיצוב זה דבר שמהווה חשיבות לנוחות המשתמש הקדשנו לזה לא מעט זמן על מנת שיהיה אפקטיבי, יעיל וקל לשימוש.

חשבנו על כמה רעיונות, כל אחד הציג את הרעיון/רעיונות שלו לגבי העיצוב בפני חברי הקבוצה, בחרנו בהכי יעיל והעונה לדרישות שלנו בנוגע לעיצוב, ואז חילקנו את המשימות שלנו לכמה תהליכים וכל חבר בקבוצה לקח לו תהליך אחד או יותר כדי לממש.

דילמות: מיקום של כפתורים, אם לשים את הכפתור צמוד לפעולה עצמה או למקם אותו בצדי החלון

למטה על מנת שיהיה מובן למשתמש.

בנוסף הצבעים של הכפתור על מנת שיהיה נוח למשתמש לביצוע פעולה רצויה, שיהיה בולט מספיק.

לגבי הדילמות בנוגע לעיצוב, היה לנו כמה כאלה, למשל:

הצבעים שהולכים לבחור

השדות ומיקומם המתאים ב GUI

האם להוסיף ל GUI כלשהו תמונה, ואם כן, אז איזה תמונה

עוד דילמות די גדולות שהייתה לנו במסכי ביצוע הזמנה ומסך ביקור מזדמן בהעברת נתונים המזמין,

בהתחלה לא היינו סגורים באיזה שדות להשתמש, למשל בהזנת מועד ההזמנה (תאריך ושעה),

ובביצוע ראשוני שלנו השתמשנו בשני TEXT FIELD וביצענו בדיקת נכונות קלט על שניהם,

רק לאחר דיונים והתלבטות גדולה החלטנו לשנות אותם לשדות יותר מתאימים ואפיקטיביים מבחינת המשימות

ואז שנינו ל DatePicker לבחירת ותאריך ComboBox עם טווח שעות פעילות הפארק, ואז עשינו בדיקות נכונות קלט לכל שדה.

עוד התלבטות שנתקלנו בה היא האם צריך להציג את מועד ההזמנה בביקור לא מתוכנן?,

בסוף החלטנו שעדיף להוסיף שדות אלו, אך בלי אפשרות עריכה, כי עדיף שמרכיבים עיקריים כאלה יהיו מוצגים בפני השתמש כדי לחזק נראות הממשק.

עוד התלבטות אחת שהייתה לנו, שקשורה להתחברות משתמשים למערכת למשל..

(1) אם זה מדריך שרוצה להזמין הזמנה במערכת שלנו האם במסך ההתחברות הוא צריך להזין את התפקיד שלו "כמדריך" כשדה

בחירה מתוך ComboBox שיש בו "מזמין רגיל", "מדריך", "מנוי משפחתי"

(2) אם עובד מתחבר למערכת האם הוא בוחר את התפקיד שלו מתוך ComboBox "מנהל פארק" או "מנהל מחלקה" או עובד רגיל או "נציג שירות"

או שפשוט יתחברו כרגיל עם תעודה מזהה אם זה מזמין או שם משתמש וסיסמה אם זה עובד רגיל ואז אנו כמתכנתים נספק פונקציה שמנתבת

את המשתמש הנ"ל לחלון הרלוונטי ביחס לתפקיד שלו במערכת שלנו (אחרי בדיקה ב DB)

בסוף החליט הצוות שלנו שאנחנו הולכים על אופציה

להתחברות רגילה עם מספר מזהה או שם משתמש וסיסמה והמערכת שלנו תבדוק ב Db מה התפקיד שלו ותעביר אותו למסך הרלוונטי .

(ב) ציינו אי לו מהעקרונות שנלמדו בהרצאות הקורס בנושאים Architecture : Design , Reuse , patterns Design - באו לידי ביטוי בתהליך התכנ הכלל - מערכתי שביצעתם והסבירו באיזה אופן – באמצעות דוגמאות ספציפיות (לא כלליות) מתוך המערכת שפיתחתם . לכל אחד מהעקרונות, הסבירו מה הם היתרונות המתקבלים משימוש בהם.

בפרויקט שלנו עשינו שימוש בכמה עקרונות שלמדנו בהרצאות:-

1 Divide and conquer

בפרויקט שלנו עשינו שימוש במודל ה 3 שבבות שזה חלוקה ל 3 סוגים שהם Client ,server,Gui Client התחלק ל 2 חלקים ל-Guis .
ו Controllers ב-Guis מימשנו את כל ה - Frames ואת כל מה שריצנו להציג כמסכים בפני המשתמש דאגנו שה Guis יעבירו את הפונקציונליות שהשתמש יכול לממש במערכת שלנו בצורה הכי ברורה ומובנת.

(2) השתמשנו ב- Singleton Design Pattern :-השתמשנו בתבנית סינגלטון שדרכה הגבלנו את יצירת המופע של ה-mySqlConnection למופע יחיד כפי שלמדנו בהרצאות בכך שייצרנו מופע יחיד עם בנאי פרטי שמאתחל את המשתנה שבו מוחזק חיבור ה- mySqlConnection הבנאי למשתנה זה מופעל פעם אחת בלבד שבו אנחנו בונים מופע של ה-חיבור ובכל פעם נוספת שבה אנו מנסים לבקש מופע ישלח אלינו אותו מופע שכבר קיים של החיבור.

(3) השתמשנו בנוסף בעקרון ה reuse

השימוש בעיקרון זה בה לידי ביטוי במערכת שלנו בעיקר ע"י השימוש במחלקה OCSF המבצעת תקשורת ממחשב אחד לשני.
מחלקה זו אפשרה לנו לבצע תקשורת בין מחשב השרת לבין מחשב הלקוח. הלקוח יכול לבקש מידע מן השרת ולקבל אותו באמצעות המערכת.
בנוסף לכך, הלקוח יכול לשלוח הודעות אל השרת ולהעביר מסרים.

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 על Login) שביצעתם

(או לא ביצעתם) במהלך הפרויקט (ע"י תיאור מפורט של בדיקות מרכיבים ספציפיים של מערכת GoNature ").

במהלך פתיוח התוכנה שלנו ערכנו הרבה בדיקות על מנת להבטיח נראות גבוהה של ממשקי המערכת לבנוסף יעילות גבוהה של קוד התוכנה , בכך שהמשתמשים שלנו יקבלו את מה שהם מבקשים בלי מאמץ גדול על מנת להבין מה הם צריכים לעשות ואיך הם צריכים לעשות , בתהליך הפיתוח ביצענו בדיקת יחידה בכך שבדדנו כל חלק ובדקנו שכל חלק כזה עובד באופן תקין ועצמאי , בדיקות היחידה בוצעו בדרך הזאת :-

1) אם שדה מסוים שחייב להתמלא לפני מעבר לדף הבא במערכת -קופצת הודעת שגיאה שצריך למלא אותו דוגמאות מהמערכת:-

א) אם עובד פארק רוצה להזהדות בכניסה אם השאיר שדה השם משתשם והסיסמא ריקים תופיע הודעת במפורט עבור כל שדה שצריך למלא

ב) בביצוע הזמנה -אם המטייל החסיר אחד הנתונים תפוע על המסך הודעה בהתאם לגבי כל שדה.

2) אם השדה אמור להכיל רק מספרים והמשתמש הכניס אותיות תופיע לו הודעת שגיאה שהקלט שהכניס שגוי, וצריך לתקן אותו

דוגמאות מהמערכת:-

בעריכת הזמנה הלקוח מכניס מספר הזמנה שהוא Integer אם הכניס בטעות String תופיע לו הודעה על המסך שהמספר לא חוקי

3) במידה ומשתמש במערכת מחכה למידע מה Db המערכת במקביל מצפה שיכניס לה מידע תקין אם המשתמש לא הכניס מידע כצפוי המערכת תציג לו כפלט הודעה בהתאם לדוגמא:- אם המשתמש רוצה לערוך הזמנה והוא הכניס מספר הזמנה שאין לו אותה אז תופיע לו על הודעה על המסך , "שלא קיימת הזמנה במספר כזה עבורו"

4) בדיקות שתלויות ב Db:-

א) בדיקת הרשאות למשתמש מסוים בביצוע הזמנה ,כשהמשתמש מזדהה במערכת ורוצה לבצע הזמנה כאשר הוא לוחץ על דף הזמנה בודקים מוצה Data base מה ההרשאות למשתמש זה בביצוע הזמנה - "הזמנה ליחד", "הזמנה למנוי משפחתי", "הזמנה למדריך" ובהתאם הצגנו לו בדף ביצוע ההזמנה ב Combo Box מה סוגי ההזמנות שיכול לבצע

ב) כאשר מנוי משפחתי רוצה לבצע הזמנה עם מספר המנוי שיש לו בודקים מול ה Database מה מספר האשנים במנוי ומגבילים לו מספר האנשים בהזמנה בהתאם לכך.

"קופסה שחורה":- השתמשנו בבדיקות יחידה בשיטת קופסא שחורה שבה אנחנו משווים בין הקלט להפלט המתקבל לתוצרים הצפויים ובהתאם קובעים נכונות הקלט. בדיקת תפקוד רכיב במערכת, בדיקת נכונות פלט/ תגובה, עיקרי הבדיקה היא עבור תהליכים מורכבים שנדע אם הם חוקיים או לא על פי מה שצפוי למו בפלט, לדוגמא מעבר בין מסכים שדרכו עוברות הקליינטים ממסל למסך , מצריך בדיקה שמופעלת כאשר אנחנו רוצים לבדוק אם הקלט שהוזן חוקי או לא ובהתאם אם לעבור למסך הבא או לא , לדוגמה:- כאשר המשתמש מזדהה במערכת הוא מתבקש להכניס נתונים בשדות מתאימה במסך שמופיע לו, לאחר הקלדת הנתונים הקלט עובר בדיקה אם הוא תקין המשתמש מועבר לדף הבית הרלוונטי לו, אחרת מחזירים לו הודעת שגיאה .

קופסה לבנה:-השתמשנו בבדיקת תוכנה בשיטת קופסא לבנה שמטרתה לבדוק המבנה הפנימי של הרכיב במערכת ונכונות החלקים הפנימיים של הרכיב ובדיקות פונקציונליות מורכבות, בחינת מבנה הפנימי של היחידה לפי בדיקת מסלולי החישוב, נכונות החישובים ונכונות ההחלטות הלוגיות, עשיית בדיקות אלה עזרו לנו בעיקר בבדיקות פונקציונליות כאשר קיים מסלול מורכב יותר כמו צפייה בדו"ח, הוספת מטייל חדש למערכת, עדכון פרמטרים, הוספת הנחות וכו'.

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 בשלב מימוש התוכנה, וצינו יתרונות וחסרונות שלה.
יש להתייחס גם לתהליך העבודה - לא להתמקד רק בכלים ואספקטים טכניים .

אחרי שחווינו את חווית העבודה בצוות, הבנו את משמעות הצוות ושיתוף הפעולה בין חברי הצוות על מנת לייצר מכונה שעובדת היטב ועומדת בדרישות ויעדים. עבודה בצוות היא דבר שמעודד, מייעל את העבודה, לפעמים גם מזרז את סיום העבודה והגשת המטרה בצורה יוצאת מן הכלל.

להיות בין כמה אנשים נותן את האופציה להעלאת רעיונות חדשות וכיוונים שונים שדרכם נוכל להגיע להחלטות סופיות שהן יעילות ככל האפשר. וזה בדיוק מה שהיה אצלנו: עודדנו אחד את השני, עזרנו אחד אל השני ותמיד העבודה הייתה מקבילית דבר שעשה מזה להיגמר במהירות יותר. אופן פעולת כל אחד מאתנו הייתה יותר יעילה, זה מה שהרגיש כל אחד מחברי הצוות שלנו.

ולבסוף אפשר להגיד שתמיד הרגשנו שההחלטות שלוקחים אותן היו הכי יעילות כי תמיד היו רעיונות ונקודות מבט וראות שהיא שונה מהקבוצה מעלה. נציין גם שכל אחד מאתנו עכשיו יודע דברים הרבה יותר, וזה בגלל עבודתנו כצוות שיש בה אנשים עם ידע רב וגדול אשר פיזר אותו על חברי הקבוצה. היתרונות של השיטה שלפיה פעלנו הם:

יכולת העמידה בזמנים: מכיוון שהייתה עבודה במקביליות, ועוד משהו שעזר לכך שנעמוד בזמנים הוא שבנינו לוח זמנים שהצמדנו אליו ופעלנו לפיו. חלוקת העבודה לחלקים שוב נזכיר שזה עשה עבודה מהירה יותר ויעילה, מה שגם עשה שלא נרגיש עמוסים מדי (פחות לחוצים היינו). להרגיש יותר חזקים, ללמוד שיטות חדשות, ולצבור ידע רב. להיעזר בחברי הצוות במידה ונתקעים בקושי כלשהו. חסרונות השיטה שפעלנו לפיה:

ישנם רגעים שבהם הרשינו לעצמנו לדחות את העבודה ליום שלמחרת!

(ב) **שילובי קוד (אינטגרציה מערכתית - לאחר הפיתוח הראשוני) ובדיקות. ציינו באופן פרטני, בהתייחסות ספציפית לפיתוח המערכת, "GoNature" איך פעלתם בשלב זה של הפיתוח (למשל: תיאור התנהלות התהליך ההנדסי (לא עבודת הצוות), 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?

החלטנו להתחלק לשתי תתי קבוצה, כך שחלק יעבוד על הסרבר, והחלק השני על הקליינט, והתקשורת בין שתי הקבוצות הייתה דרך whatsapp | googleDoc, היינו קובעים באופן יומי לעצמנו מטרות ומתעדכנים בכל דבר נעשה ומוודאים את תקינותו, כך שהעבודה שלנו תהיה במקביל ועם יישור קו, ואם במקרה משהו נתקל בבעיה כלשהי ולא הצליח לטפל בה, אז כל חברי הצוות חוברים לעזר ולתקן אותה, כדי שנמשיך הלאה כולנו ביחד, והשתדלנו כל הזמן לא לעבוד שניים על אותה מחלקה ואותו קוד כי לא רצינו שתקרה איזושהי התנגשות בין קטעי הקוד, כלומר היינו תמיד, נזהרים מזה ע"י כך שתמיד היינו מתעדכנים מתי עובד ועל איזה משימה.

ככה הצלחנו להתקדם ביחד לאורך כל הדרך המאתגרת, ובזכות שהשיתוף הפעולה בין חברי הצוות היה ברמה גבוהה אנחנו יכולים להגיד! Yes, We Did It.

קשיים שהיו:

פרשנו את המשימות על כל חברי הצוות בדרך יעילה. היינו תמיד בקשר לגבי המתרחש, מה נשאר לבצע ומה צריך לתקן, ואיפה אנחנו עומדים מבחינת זמן והשגת יעדים. כשנתקענו בבעיות פנינו קודם כל נסינו בכל מאמץ לפתור את הבעיה בעצמנו, ובמקרים שבהם פחות הצלחנו פנינו לצוות הקורס, או נעזרנו בסטודנטים משנים מתקדמות. הקשיים שהיו הם שילוב קטעי קוד בהתחלה הפרויקט ולדאוג שהכול עובד בצורה הכי טובה.

בשלב שילובי הקוד והבדיקות פעלנו בצורה הבאה:
עשינו שיחות זום לכל הקבוצה וקיבלנו החלטות איך כל התהליכים הולכים להראות וכיצד אנו נממש זאת, בעצם במפגש זה עשינו יישור קו בכך שכולנו מתואמים איך יש לממש זאת.
לפני שהתחלנו את מטלה 3 עשינו תיאור לתהליך וקבענו לכל חברי הקבוצה על פי מה נעבוד על מנת שיהיה סדר, הדברים שקבענו הינם:
* נתינת שמות משמעותיים למשתנים ולמחלקות, ובכך הבנה טובה יותר של הקוד.

* חלוקה ל packages- מתאימים.

o

* החלטנו על יצירת ממשק נוח וידידותי למשתמש, ובנוסף למתכנת בין ה Client- ל Server.

* באופן טיפול הבעיות החלטנו שנתיעץ כקבוצה ונקבל החלטה על פי הרוב והדרך הטובה ביותר למימוש.

דוגמה לתיאור תהליך אינטגרציה:-

כפי שהזכרנו התחלקנו בקבוצה לשתי תתי קבוצות חלק שעובד על הקליינט וחלק שעובד על הסרבר:-
הצוות קובע ישיבת צוות כדי לקבוע מה היעדים שלנו לשבוע הזה ומחלק את המטלות לקליינט וסרבר,
* הקליינט מתחיל וכשמסיים מעדכן בגוגל דוקס בצבע ירוק מה הוא עשה ומה שלח לסרבר, ובצבע צהוב מה הוא מצפה מהסרבר לעשות ולחזיר לו

* בזמן שהסרבר עובד על מה שהלקיינט ביקש, הקליינט ממשיך בעבודה שלו מסיים ומעדכן בגוגל דוקס בהתאם

* הסרבר עושה את העבודה כשהוא מסיים מעדכן בגוגל דוקס, מה שהיה בצהוב לירוק ובכך שהוא עשה מה עליו לעשות בנוסף מעדכן מה הוא מחזיר בדיוק לקליינט ובאיזה צורה

* הקליינט מקבל את העדוכנים וממשיך בהתאם

* אחד מחברי הצוות לקח אחריות ודאג לחבר בין מה שעשה הקליינט לבין מה שעשה הסרבר ובדק שהכל תקין ועידכן בגוגל דוקס מה חסר ומה צריך עוד לעשות בחלק שחייבר דברים שהיינו משנים בדיעבד בגישתנו הם:

* לבצע תיעוד תוך כדי כתיבת הקוד ולא בסיום, זאת משום שככל שעובר הזמן מפתחי הקוד (שאלה אנחנו) שוכחים איך להשתמש בקוד שאנחנו בעצמנו כתבנו. בנוסף, כתיבת הערות באופן טבעי משפרת את התקשורת בין המתכנתים העובדים על פיתוח התוכנה .

* יישום רעיונות לטובת העניין בזמן ולא לרשום הערה איפהשהו כדי לחזור לממש בעתיד, קרה שאחד מחברי הצוות העלה רעיון טוב אך לא ביצענו באותו רגע את המימוש, ולאחר זמן מה והמשך עבודה על חלק קוד אחר הרעיון נשכח ולא מומש .