# קבוצה 4:

**יזהר אננייב** [Izhar\_ananiev@hotmail.com](mailto:Izhar_ananiev@hotmail.com)

**ליאור גל** [liorgal28@gmail.com](mailto:liorgal28@gmail.com)

**כפיר ביטון** [kfirbitonn@gmail.com](mailto:kfirbitonn@gmail.com)

**איתי עלמני** [itaialmani@gmail.com](mailto:itaialmani@gmail.com)

**רון בן-צבי** [benzvi.ron@gmail.com](mailto:benzvi.ron@gmail.com)

**הוגש בתאריך:** 17/12/2017

1. **תארו את תהליך התכנון שביצעתם לתהליך ביטול הזמנה ע"י הלקוח. פרטו מה הם הדילמות התכנוניות שהתייחסתם אליהם? מה היו החלופות ומה השיקולים לבחירת הפתרון שיושם? בתשובה התייחסו לעקרונות שנלמדו בהרצאות: ,Reuse ,Design ,Architectureו Design .patterns**

כאשר לקוח רוצה לבטל הזמנה, עליו להזין את מספר ההזמנה.

לאחר ההזנה, ראשית המערכת תבדוק האם ההזמנה קיימת במסד הנתונים, על ידי תקשורת עם ה-server.

בנוסף המערכת תבדוק האם ההזמנה המצוינת נוצרה על ידי המשתמש המבקש לבטלה – המערכת תשלוף את פרטי המשתמש מה-Context, ותשווה את פרטי המשתמש המקושר להזמנה למשתמש הקיים (Architecture Design). המחלקה Context מכילה את המידע הנעשה בו שימוש במספר מחלקות שונות, של כל Client. כך פתרנו את הדילמה כיצד לקשר בין הלקוח לבין ההזמנה שלו על מנת שלא ייווצר מצב שלקוח ביטל הזמנה של לקוח אחר.

כעת, המערכת תבדוק את ההפרש בין זמן ביטול ההזמנה לבין זמן המשלוח המבוקש ותבצע החזר כספי ללקוח לפיו הפרש הזמנים, על פי סיפור המערכת. הפיצוי מתבצע על ידי הוספה של הסכום המתאים אל חשבון התשלום של הלקוח. ניגש לחשבון הלקוח על ידי שליפת פרטי המשתמש הנוכחי ממחלקת ה-Context.

בשלב זה, המערכת תשנה את סטטוס ההזמנה ל-"canceled". על מנת להפיק דוחות של הזמנות, ולהציג בהם גם את ההזמנות המבוטלות, לא נמחק הזמנות שבוטלו מן המסד, אלא נשנה את סטטוס ההזמנה.

לבסוף, נעדכן את הנתונים השונים במסד הנתונים, ונציג הודעה בהתאם.

* 1. **ציינו עקרון של גישת התכן : Object-Oriented Design שאפשר לכם לשלב קטעי קוד ומרכיבים אחרים שלא אתם כתבתם או תכננתם. תארו בדיוק (ובהתייחסות ספציפית) ובפירוט את התכונות של אותם מרכיבים אשר אפשרו לכם לשלבם במערכת שלכם, תוך התייחסות בדוגמאות ספציפיות (לא 'עקרוניות)' לדרישות הפונקציונליות של המערכת שתכננתם (התייחסות ספציפית בהקשר זה = התייחסות למרכיבים ספציפיים מתוך התיאור המילולי הראשוני של המערכת ששאתם מפתחים מהתחלת הסמסטר).**

\* בתשובה לשאלתנו אנו מתייחסים לכתיבת האב-טיפוס.

בתיאור המילולי של המערכת אנו נדרשים לאפשר למספר משתמשים שונים להיות מחוברים בו זמנית למערכת, בנוסף נאמר כי המערכת תהיה בעלת אופי תפעולי מבוזר כך שניתן יהיה לעבוד מתחנות עבודה בו זמנית על ידי משתמשים שונים.

על מנת לממש את הדרישות הללו השתמשנו בעקרון ההורשה שהוא חלק מרכזי בגישת התכן Object – Oriented Design.

עקרון זה בא לידי ביטוי על ידי הגדרת מחלקות חדשות היורשות מהמחלקות הכלליות : AbstractServer, AbstractClient.

על ידי ביצוע הורשה זו שילבנו למעשה קטעי קוד קיימים (מתודות שונות) שהיו נחוצים לנו לצורך עבודה נכונה עם מסד הנתונים שלנו.

* 1. **ציינו לפחות עקרון אחד של OOD שלא בא לידי ביטוי במודלים שלכם - והסבירו למה.**

\*בתשובה זו התייחסנו לתרשימי ה- UML

העקרון שלא בא לידי בטוי בתרשימי ה – UML שלנו הוא עקרון הכימוס.

עקרון הכימוס הינו עקרון הסתרת מידע, עקרון זה מאפשר הסתרת כל המרכיבים הפנימיים של האוביקט ומאפשר גישה

* 1. **הערכה כללית:**
     1. **מהם היתרונות של מודל UML כעזר לתהליך התכנון? ציינו דוגמה אחת קונקרטית (לא כללית) מתוך התהליך שאתם בצעתם לשימוש כזה ב- UMLתוך תיאור והתייחסות ספציפית למרכיבים של מערכת "זר-לי" שתכננתם ומידלתם.**
     2. **ציינו קשיים הנובעים מחסרונות של UML שנתקלתם בהם. גם כאן התייחסו ספציפית לתהליך שבצעתם בפרויקט זה.**
  2. **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) מתוך עבודתכם.**