בס"ד תשפ"ג

**מיני פרויקט במערכות חלונות**

*תיאור כללי*

הפרויקט השנה עוסק בניהול מערכת מידע עבור חנות המציעה מוצרים מקטגוריות שונות.

נפתח אפליקציה בעלת 2 היבטים:

* ממשק עיקרי אותו מנהלים בעלי החנות המאפשר להוסיף ולעדכן מוצרים שונים ולנהל הזמנות
* ממשק עבור הקונים המעוניינים לבצע הזמנות

חשוב לציין שאנחנו מפתחים מערכת חלקית בלבד, מערכת שעונה לדרישות הקורס, במסגרת זו אנו לא יכולים לתת מענה לכל הדרישות שיכולות להיות במערכת אמיתית.

בנוסף, כיום מקובל להשתמש במערכות אינטרנטיות המאפשרות גישה אונליין, תחומים אלו לא נלמדים במסגרת הקורס.

את המערכת נבנה בשלבים במהלך הסמסטר ע"פ [ארכיטקטורת השכבות ההיררכית](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%94_%D7%A8%D7%91-%D7%A9%D7%9B%D7%91%D7%AA%D7%99%D7%AA#%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%A9%D7%9C%D7%95%D7%A9_%D7%A9%D7%9B%D7%91%D7%95%D7%AA).

נושא הארכיטקטורה נלמד בהרחבה במהלך הסמסטר.

המערכת תאפשר שמירה וגישה לנתונים של ה**מוצרים** השונים הנמכרים בחנות, וכן תנהל **הזמנות** של המוצרים.

לצורך הפשטות לא ישמרו באופן ישיר נתוני הלקוחות, מעבר לנדרש עבור ההזמנה.

בנוסף, אתם נדרשים לבחור תחום עניין שבו החנות עוסקת (למשל: חנות בגדים, חנות מוצרי חשמל או אלקטרוניקה) ולבחור קטגוריות שהמוצרים משתייכים אליהם (5 קטגוריות, ראו דוגמאות בהמשך).

הפרויקט יתבצע בשמונה שלבים. ציון הפרויקט המירבי שהתבצע לפי הדרישות בלבד - 90 נק'. בנוסף, ניתן לצבור עד 15 נק' ע"י ביצוע בונוסים. ניקוד בונוסים ייחשב רק אם בפרויקט בוצעו ועובדים כל השלבים. זאת אומרת, מי שלא יבצע שלב 6 ו-\או 7 או אחד מהם לא עובד - לא יוכל לקבל ניקוד בונוסים.

## פירוט שלבי הפרויקט

**שלב 0** - הכרת סביבת עבודה, מאגר גיט מקומי ומשותף בענן github.com - שבוע אחד (2% מהציון)

**שלב 1 - *שכבת נתונים* *(DAL)*** - שבוע אחד (4% מהציון)

* ישויות נתונים
* שמירת נתונים במערכים
* מתודות גישה לנתונים (CRUD)
* ממשק משתמש של קונסול לבדיקות שכבת נתונים

**שלב 2 - *שכבה לוגית (BL) וממשקים*** - שבועיים (10% מהציון)

* עדכון שכבת נתונים, שימוש באוספי נתונים לשמירה ולהעברה
* ישויות לוגיות
* שכבה לוגית בסיסית
* ממשק משתמש של קונסול לבדיקות שכבה לוגית

**שלב 3 - *ממשק גרפי בסיסי*** - 2 - 2.5 שבועות (20% מהציון)

* עדכון שכבות נתונים ולוגית
* שכבת תצוגה בסיסית (ממשק גרפי)

**שלב 4 - *השלמת מודל השכבות*** - חצי שבוע (10% מהציון)

* עדכון שכבת הנתונים והשכבה הלוגית לפי מודל השכבות המלא תוך שימוש בתבניות עיצוב של Singleton ו-Simple Factory Method

**שלב 5 - *ממשק גרפי מלא*** - 3 - 3.5 שבועות (20% מהציון)

* השלמת ממשק גרפי מלא
* עדכוני תצוגה ייעשו דרך binding מתוך xaml בלבד
* עדכון השכבות והממשקים לשימוש באוספי נתונים כלליים (<>IEnumerable)
* עדכון כל השכבות לעיבוד נתונים בעזרת LINQ בלבד (וללא לולאות)

**שלב 6 - *שמירת נתונים ב-XML*** - שבוע אחד (10% מהציון)

* הוספת מימוש נוסף של שכבת הנתונים עם שמירה בקבצי XML

**שלב 7 - *תהליכונים וסימולטור*** - 1.5 - 2 שבועות (14% מהציון)

* הוספת ממשק סימולציה לשכבת נתונים
* הוספת סימולטור בשכבת נתונים
* הפעלת סימולטור ממשק משתמש

## דרישות כלליות של המערכת

המערכת תאפשר שימוש ע"י "מנהל" החנות (בעלים או מוכרים), וע"י קונים בחנות.

המנהל יוכל:

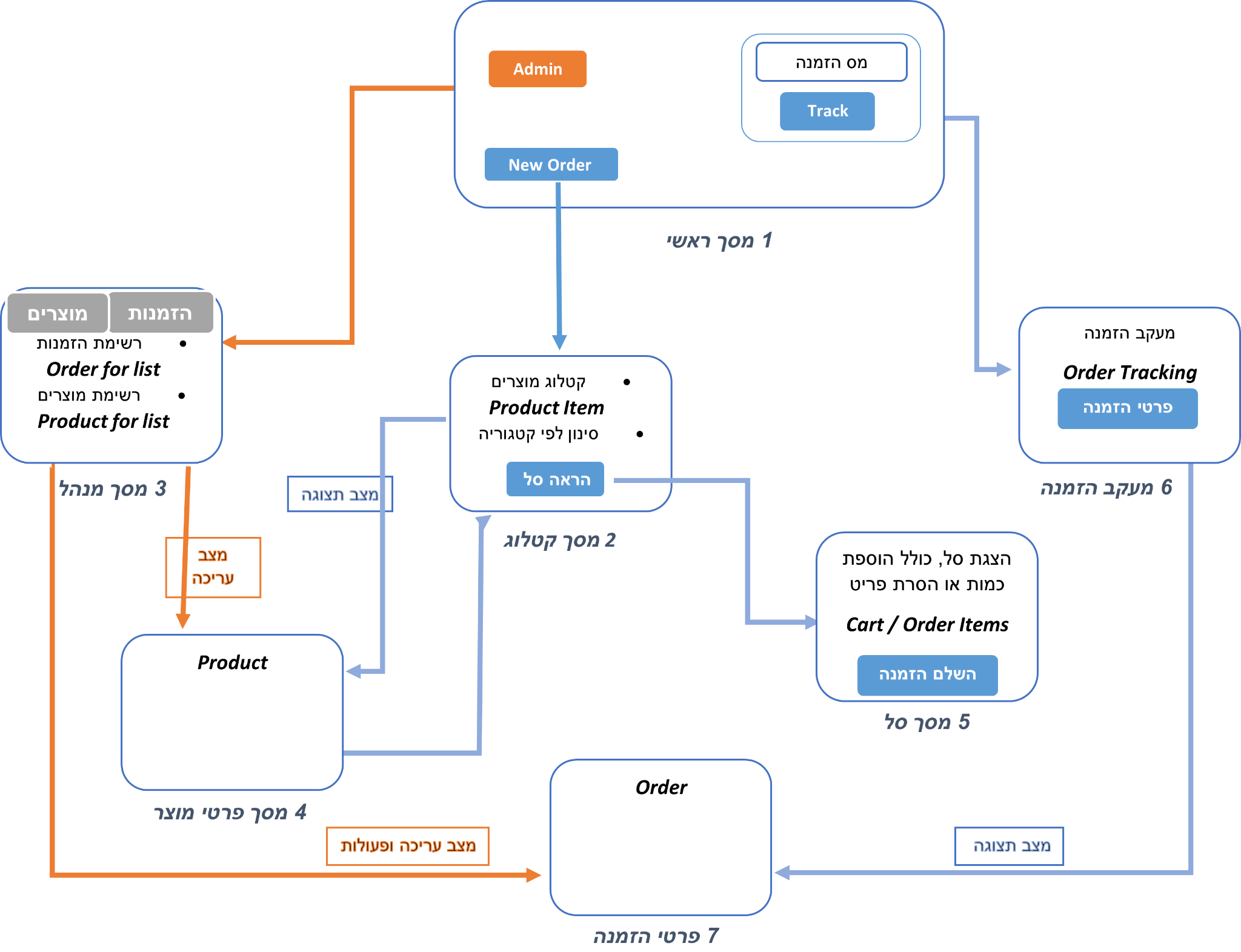
* לנהל את המוצרים:
  + לראות את רשימת המוצרים
  + להוסיף מוצר חדש
  + לעדכן נתוני מוצר קיים (תיאור\שם, מחיר, כמות במלאי, וכו')
* לנהל הזמנות קיימות
  + לראות את רשימת ההזמנות שנוצרו
  + לעדכן הזמנה קיימת (בחיים האמיתיים - לבקשת הקונה, או בתיאום עם קונה - שינוי כמות של מוצר, הורדת מוצר מהזמנה, הוספת מוצר להזמנה)
  + עדכון מצב ההזמנה (שילוח - shipping, אספקה - delivery)
  + בתצוגת ההזמנה יוכל המנהל לראות את פרטי ההזמנה (כולל רשימת מוצרים מוזמנים עם כמויות ומחירם בעת ביצוע ההזמנה)

הקונה יוכל:

* לצפות בקטלוג המוצרים (רשימת המוצרים)
* לצפות בפרטי המוצר ולהוסיף אותו לסל הקניות
* לנהל את סל הקניות (לשנות כמויות, להוריד מוצר מהסל)
* להפוך סל קניות להזמנה - ע"י אישור\ביצוע הזמנה
* לעקוב אחרי מצב ההזמנה (order tracking)

## אפיון המסכים של ממשק המשתמש

לפניכם דוגמה (לא מחייבת) של המסכים\החלונות של ממשק המשתמש במערכת:

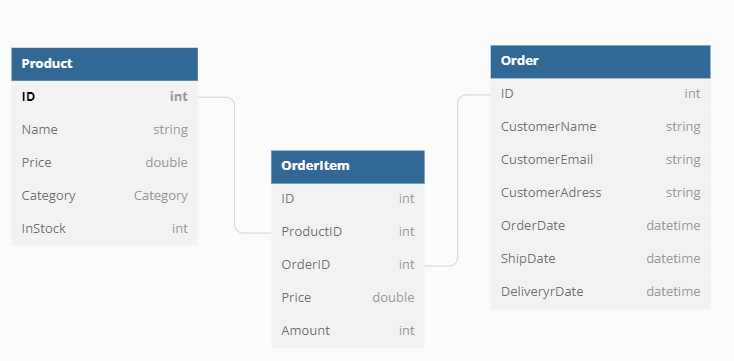
**איור 1**. תרשים מסכים של ממשק משתמש

הסטודנטים יהיו רשאים לעשות שינויים בהרכב ותסריטי זרימה בין המסכים\החלונות ע"פ - הבנתם של תפקוד החנות וע"פ הגיון בריא. הסטודנטים יצטרכו להסביר את החלטותיהם ואת ההיגיון שלהם בשלב ההגנה של פרויקט.

## 

## ישויות הנתונים

לפניכם דוגמה (**לא מחייבת**) להגדרת הישויות ([מקור](https://dbdiagram.io/d/61adb58b8c901501c0e1ebdc)):

**איור 2**. תרשים ישויות נתונים

נגדיר שלוש ישויות נתונים : ***מוצר***, ***הזמנה*** ו***פריט בהזמנה*** כדי לציין אלו מוצרים כלולים בהזמנה, כלומר הזמנה אחת יכולה לכלול מוצרים שונים, וכן מספר פריטים מאותו מוצר.

הערה : הישות המקשרת כוללת גם את המחיר של הפריט בהזמנה שנקבע בעת ההזמנה מתוך המחיר הנוכחי של המוצר. שימו לב שבמקרה והמחיר של המוצר משתנה זה לא משפיע על המחיר של הפריט בהזמנות קיימות.

רשימת הישויות והנתונים שבהם (הרשימה הזו מחייבת, אך ניתן להרחיבה כרצון וצרכי הסטודנטים):

* ***מוצר***
  + מספר מזהה ייחודי (כמו מספר הברקוד של המוצר)
  + שם המוצר
  + קטגוריה
  + מחיר
  + כמות במלאי
* ***פריט בהזמנה***
  + מספר מזהה ייחודי (מספר רץ אוטומטי)
  + מספר מזהה של הזמנה
  + מספר מזהה של מוצר
  + מחיר ליחידה
  + כמות
* ***הזמנה***
  + מספר מזהה ייחודי (מספר רץ אוטומטי)
  + שם הלקוח המזמין
  + כתובת דוא"ל
  + כתובת למשלוח
  + תאריך יצירת הזמנה
  + תאריך משלוח
  + תאריך מסירה

**נ.ב.** במקרה של אי התאמה בין תרשים לבין הרשימה לעיל - הרשימה היא זו שקובעת

**דוגמאות**:

עבור חנות העוסקת בביגוד ומוצרי אופנה

| **מס' מזהה** | **שם המוצר** | **קטגוריה** | **מחיר** | **כמות במלאי** |
| --- | --- | --- | --- | --- |
| 54206823 | שמלת פרחי סתיו | שמלות | 150 | 15 |
| 15661345 | מכנס שחור אלגנט | מכנסיים | 100 | 12 |

עבור חנות העוסקת במוצרי אלקטרוניקה

| **מס' מזהה** | **שם המוצר** | **קטגוריה** | **מחיר** | **כמות במלאי** |
| --- | --- | --- | --- | --- |
| 92548335 | iPhone 14 | טלפונים ניידים | 3000 | 15 |
| 92263548 | מסך סמסונג 21" | מסכים | 850 | 12 |

כמובן שאפשר להציע תחומים נוספים.

טיפוסי הנתונים בישויות יהיו:

* מספרים מזהים (ID) - מסוג int
* שמות, תיאורים, כתובות - מסוג string
* עבור קטגוריית מוצר תוגדר אנומרציה (enum)
* כמויות - מסוג int
* מחירים - מסוג double
* תאריכים - מסוג DateTime

במימוש שכבת הנתונים עבור ישות פריט בהזמנה בנוסף למתודות גישה בסיסיות של CRUD יהיו גם:

* מתודת בקשה\קריאה ע"פ שני מזהים (ID) - מזהה מוצר ומזהה הזמנה, המתודה תחזיר את אובייקט של ***פריט בהזמנה*** המתאים
* מתודת בקשה\קריאה של רשימת\מערך פרטי הזמנה ע"פ מספר מזהה של הזמנה

במימוש שכבת הנתונים במערכים\רשימות אובייקטים במחלקה DataSource:

* גודלי המערכים באתחול יהיו:
  + עד 50 מוצרים
  + עד 100 הזמנות
  + עד 200 פריטי הזמנה
* באתחול ערכי המערכים יוספו אליהם כמויות אובייקטים כדלהלן:
  + לפחות 10 מוצרים
  + לפחות 20 הזמנות
  + לפחות 40 פריטי הזמנה (1 עד 4 להזמנה בודדת)
* המוצרים שנוספו:
  + בכל המוצרים יהיה מספר מזהה שיוגרל או יוגדר ידנית עבור כל מוצר, עם לפחות 6 ספרות, וידאו שמספר זהה לא יוקצה לשניים או יותר מוצרים
  + כל המוצרים יהיו מקטגוריות שונות (כמובן שיכולים להיות כמה מוצרים באותה קטגוריה)
  + בכל המוצרים יוגדר שם המוצר ומחירו (תהיו הגיוניים)
  + בכל המוצרים תוגדר כמות מלאי, כ-5% מהמוצרים לא יהיו במלאי (זאת אומרת כמות=0)
* ההזמנות שנוספו:
  + בכולן יהיו מספר מזהה (רץ - בעזרת מחלקה פנימית Config), שם הלקוח עם הכתובת וכתובת דוא"ל שלו (הזמנות שונות יכולות לשתף אותו לקוח, אך כמובן יהיו מספר לקוחות)
  + כל התאריכים יהיו **לפני** הזמן של הפעלת התכנית (DateTime.Now)
  + לכולם יהיה תאריך יצירת הזמנה
  + לכ-80% מההזמנות יהיה תאריך משלוח שחייב להיות אחרי זמן יצירת הזמנה
  + לכ-60% מההזמנות **שנשלחו** יהיה תאריך מסירה
* פריטי ההזמנות שנוספו:
  + בכולן יהיו מספר מזהה (רץ - בעזרת מחלקה פנימית Config)
  + מזהי מוצר והזמנה יהיו קיימים במוצרים ובהזמנות!
  + לכל הזמנה יהיה 1 עד 4 פריטי הזמנה
  + כמויות פריט מסוים בהזמנה ייגרלו אך יהיו הגיוניים!
  + מחיר פריט יהיה זהה למחיר המוצר המתאים

## 

## שכבה לוגית - תיאור כללי

לפי האפליקציה שראינו ב-[אפיון מסכים של ממשק משתמש](#kix.vgxqzepvdd8k) נגדיר ישויות לוגיות **ראשיות**:

* עבור המנהל - ישויות **מוצר** (**Product**) ו**הזמנה** (**Order**) לניהול
* עבור הקונה - ישות **סל קניות** (**Cart**)

עבור הישויות האלה נזדקק לפעולות רבות שיאפשרו לְתַפְעֵל את המערכת. לכן ניצור עבורן ממשקים שיכילו הצהרות של המתודות עבור הפעולות שנצטרך. את הממשקים האלה נממש גם בשכבה זו.

הישויות האלה תהיינה ישויות מורכבות (לא שטוחות - זאת אומרת לא PDS). למשל ישות של הזמנה תכיל **בתוכה** רשימת פרטי הזמנה.

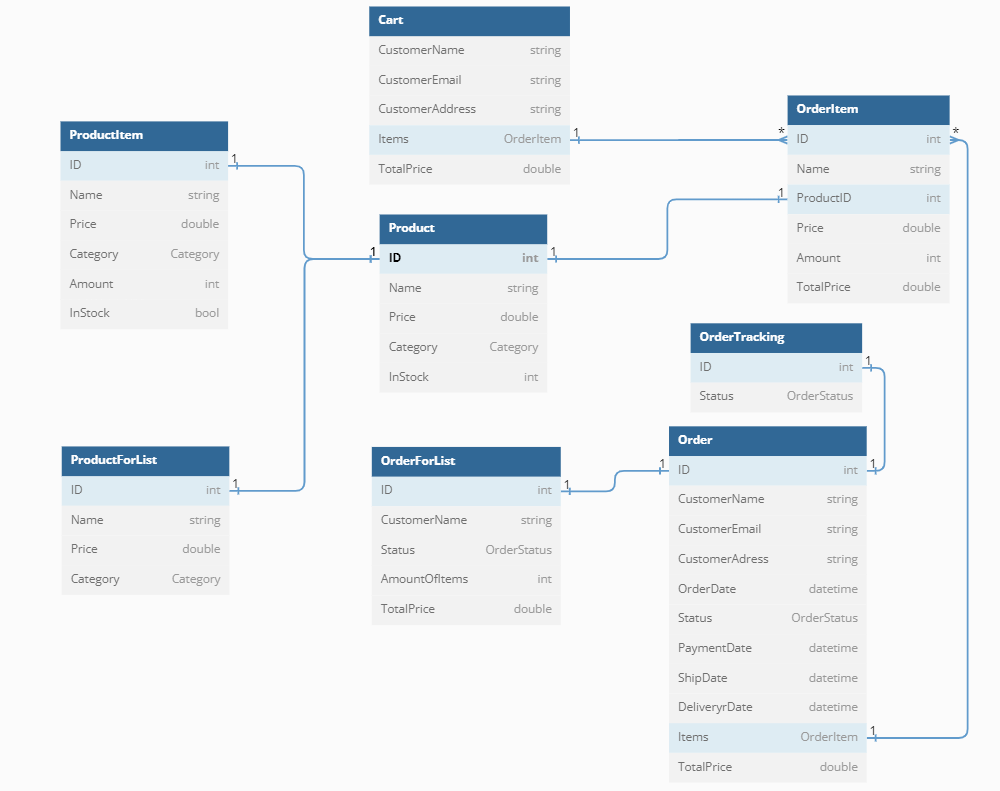
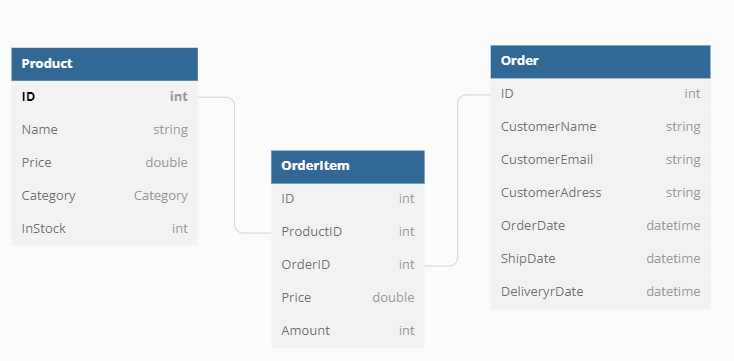
בנוסף נצטרך *ישויות עזר* נוספות עבור הצרכים של האפליקציה שאנחנו בונים. למשל עבור עבודה עם קטלוג מוצרים, נצטרך ישות עבור מוצר ברשימה (שיכיל רק מידע חלקי על המוצר - לפי מה שצריך להציג ברשימה). כמו כן, נצטרך ישות של פריט הזמנה שיכיל מידע עבור תצוגה בסל קניות או בהזמנה, שיכיל את המידע שבישות הנתונים של פריט הזמנה, אך בנוסף יכיל במפורש גם את שם המוצר (כי זה מה שהייתם רוצים לראות בתצוגת סל קניות או בתצוגת הזמנה בשביל הקונה). אנחנו לא נתאר כאן את כל הסיבות של כל הישויות אלא נסתפק בדוגמאות שמנינו. תפקידכם גם להקדיש מחשבה ולהבין את צרכי מנהל המערכת (או בעל\מוכר בחנות) ושל קונה בחנות.

השכבה הלוגית תהיה אחראית על:

* שלמות והתאמה של כל הנתונים המאוחסנים בשכבת הנתונים
* בדיקת תקינות והתאמה נתוני קלט
* בדיקת התאמה נתוני קלט ע"פ מידע בשכבת הנתונים
* הכנת אובייקטים של ישויות לוגיות לבקשת האפליקציה (ע"י ממשק המשתמש):
  + הנתונים הדרושים יושגו בעזרת "**שליחת" בקשות לשכבת הנתונים (ע"י זימון מתודות של שכבת הנתונים)**
  + אם יש צורך - השכבה "תשלח" לשכבת הנתונים כמה בקשות שצריך עבור הכנת האובייקט
  + הנתונים החסרים בשכבת הנתונים ייחשבו לפי הדרישות ואלגוריתמים מתאימים (למשל מחיר כולל של פריט הזמנה ייחשב ע"י כפל של מחיר המוצר בהזמנה בכמות המוזמנת, ומחיר כולל של סל קניות\הזמנה ייחשב כסכום של מחיר כולל של כל פרטי ההזמנה)
* ביצוע פעולות נדרשות על הישויות הראשיות - לפי דרישות המערכת
  + תידרשנה כל מיני פעולות, למשל עבור הזמנה קיימת, המנהל יוּכַל לבצע "שילוח" (shipping) ו"אספקה" (delivery) של ההזמנה
  + ביצוע הפעולה יתממש ע"י עדכון כל הנתונים המתאימים בשכבת הנתונים (כמו למשל הוספת מוצר, עדכון תאריך-זמן של שילוח, וכו'), ולפעמים גם ע"י עדכון אובייקט של ישות לוגית והחזרתו לאפליקציה (כמו למשל הוספה או שינוי כמות של פריט הזמנה בסל קניות)
  + פעולה תהיה מוצהרת בממשק מתאים
  + פעולה תמומש בהתאם לדרישות במחלקת המימוש של השכבה הלוגית
    - חלק מהדרישות תורחבנה בהמשך המסמך הזה, **וחלק תוסיפו בעצמכם ע"פ הבנתכם את תפקוד החנות וע"פ הגיון בריא)**
  + פעולה תבצע את כל הבדיקות הנדרשות של נתוני קלט
  + פעולה תבצע את כל החישובים הנדרשים
  + פעולה תבדוק את כל מה שצריך בעזרת שליחת בקשות לשכבת הנתונים ובדיקת הנתונים שהתקבלו
  + פעולה תבדוק את חוקיות והתאמה בין כל הנתונים בהתאם לצורכי ודרישות הפעולה, למשל עבור הקונה - גם בהוספת פריט לסל קניות וגם ב"שליחת" סל קניות:
    - בדיקה שהמוצר קיים
    - בדיקה שיש מלאי מספיק של המוצר
    - לקיחת מחיר המוצר עבור פריט מנתוני המוצר בעת הוספתו לסל קניות
    - חישוב מחיר כולל של סל קניות
  + פעולה תשלח בקשות עדכון\הוספה\מחיקה לכל הישויות המתאימות בשכבת הנתונים על מנת לבצע את כל השינויים הנדרשים בנתונים בצורה שתשמור על שלמות ונכונות הנתונים

## ישויות שכבה לוגית (aka[[1]](#footnote-0) ישויות לוגיות)

לפניכם דוגמה (**לא מחייבת**) להגדרת הישויות הלוגיות ([מקור](https://dbdiagram.io/d/6329a06d0911f91ba5f262d8)):

**איור 3**. תרשים ישויות לוגיות

במבט נוסף (לזה שנתנו בתיאור כללי של שכבה לוגית), השכבה הלוגית אוספת ומעבדת את המידע מישויות שכבת הנתונים, בהתאם לדרישות וצרכים של המערכת.

ישויות לוגיות מוגדרות כך שייארזו מידע עבור הצרכים של שכבת התצוגה (החלונות השונים). בשכבת התצוגה, לא מתבצע שום עיבוד של מידע (למעט בדיקות פורמט של קלט). כמו כן כשתתחילו לבנות את שכבת התצוגה, ייתכן שתרצו לשנות\לעדכן את הישויות הלוגיות שבניתם קודם לכן שיספקו את המידע החסר לתצוגה. שימו לב שישויות לוגיות יכולות להיות מורכבות בניגוד לישויות שכבת הנתונים שמוגדרות שטוחות (למשל להכיל אובייקטים של ישויות לוגיות אחרות או אוספי אובייקטים של ישויות לוגיות אחרות).

לפניכם רשימת הישויות המופיעה [בתרשים](#kix.xgy4q7ynwict), וקישורן למסכים ותצוגות אפשריות, כפי שמופיע [אפיון המסכים של ממשק המשתמש](#kix.vgxqzepvdd8k) והנתונים שבהם (הרשימה הזו מחייבת, אך ניתן להרחיבה כרצון וצרכי הסטודנטים):

* ישות לוגית ראשית של מוצר (Product) - עבור מסכים של פרטי מוצר (לניהול) ופעולות על מוצר, שתכיל את:
  + מזהה (מוצר)
  + שם
  + מחיר
  + קטגוריה
  + כמות במלאי
* ישות לוגית ראשית של הזמנה (Order) - עבור מסך פרטי הזמנה ופעולות על הזמנה, שתכיל את:
  + מזהה (הזמנה)
  + שם קונה
  + כתובת דוא"ל (מייל) של קונה
  + כתובת של קונה
  + מצב הזמנה
  + תאריך אישור\ביצוע הזמנה
  + תאריך שילוח
  + תאריך אספקה
  + רשימת פרטי הזמה (מסוג OrderItem)
  + מחיר כולל של הזמנה
* ישות לוגית ראשית של סל הקניות Cart - עבור מסך ניהול סל הקניות ואישור הזמנה, שתכיל את:
  + שם קונה
  + כתובת דוא"ל (מייל) של קונה
  + כתובת של קונה
  + רשימת פרטי הזמנה (מסוג OrderItem)
  + מחיר כולל של סל הזמנה
* ישות עזר של פריט בהזמנה (מייצג שורה בהזמנה) OrderItem - עבור רשימת פריטים במסך סל קניות ובמסך פרטי הזמנה, שתכיל את:
  + מזהה פריט הזמנה
  + מזהה מוצר
  + שם מוצר
  + מחיר מוצר
  + כמות פריטים של מוצר בסל\הזמנה
  + מחיר כולל של פריט (לפי מחיר מוצר וכמותו בהזמנה\סל)
* ישות עזר של פריט מוצר (המייצג מוצר עבור הקטלוג) ProductItem - עבור מסך הקטלוג - עם רשימת המוצרים שמוצגים לקונה, שתכיל את:
  + מזהה (מוצר)
  + שם מוצר
  + מחיר מוצר
  + קטגוריה
  + זמינות (**האם** במלאי)
  + כמות ב***סל הקניות*** של הקונה
* ישות עזר של מוצר ברשימה ProductForList - עבור מסך רשימת המוצרים, שתכיל את:
  + מזהה (מוצר)
  + שם מוצר
  + מחיר מוצר
  + קטגוריה
* ישות עזר של הזמנה ברשימה OrderForList - עבור מסך רשימת הזמנות, שתכיל את:
  + מזהה (הזמנה)
  + שם קונה
  + מצב הזמנה
  + כמות פריטים
  + מחיר כולל
* ישות עזר של מעקב הזמנה OrderTracking - עבור מסך מעקב הזמנה, שתכיל את:
  + מזהה (הזמנה)
  + מצב הזמנה
  + רשימה של צמדים (תאריך, תיאור התקדמות חבילה)

בכל התנגשות בין התרשים והרשימה הנ"ל - הרשימה היא זו שקובעת.

**הערה**: מומלץ ליצור תחילה את הישויות והמחלקות הדומות לאלו שכבר מוגדרות בשכבת הנתונים (Product, Order, OrderItem) , ולאחר מכן את הישויות המורכבות.

שימו לב להבדל בין ישויות נתונים לישויות לוגיות "מקבילות".

למשל, ישות OrderItem בשכבת הנתונים בנוסף למזהה ייחודי, מכילה מזהים של הזמנה ומוצר, מחיר שנקבע למוצר בהזמנה וכמות יחידות מאותו מוצר - בשכבת הנתונים זוהי ישות מקשרת בין הזמנה למוצר שנוסף כפריט בהזמנה. ישות לוגית לא מכילה מזהה הזמנה - כי היא בעצם תהיה מוכלת בתוך ישות של הזמנה ישירות (בצורת רשימת פרטי הזמנה), כמו כן היא תכיל נתונים נוספים - כמו שם מוצר ומחיר כולל של הפריט (לפי כמות).

בישויות לוגיות יוגדרו האנומרציות הדרושות לייצוג הערכים של שדות כמו:

* קטגוריה של מוצר (אנומרציה מקבילה לזו שבשכבת הנתונים)
* מצב הזמנה (הזמנה מאושרת, נשלחה, סופקה ללקוח)

**שימו לב**: אין להשתמש בטיפוסים של ישויות נתונים בתוך ישויות לוגיות (גם לא באנומרציות).

בישויות לוגיות יוגדרו חריגות ע"פ הנדרש (ראו בהמשך במימוש שכבה לוגית)

## 

## פעולות הנדרשות מהשכבה הלוגית ומימושן

עליכם להמציא את הפעולות הנדרשות עבור השכבה הלוגית ע"פ הפירוט ב-[דרישות כלליות של המערכת](#kix.abtz9yozaws9). אנחנו נמנה כאן את הפעולות הבסיסיות העיקריות, אך עליכם לעשות חשיבה ולהשלים את הרשימה בהתאם.

**פעולות על מוצר:**

* **בקשת רשימת מוצרים** (עבור מסך מנהל)
  + תבקש רשימת מוצרים משכבת הנתונים
  + תבנה על בסיס הנתונים רשימת מוצרים מטיפוס ***ProductForList*** (**ישות לוגית**)
  + **תחזיר** את הרשימה שנבנתה
* **בקשת קטלוג מוצרים** (עבור מסך קטלוג)
  + תבקש רשימת מוצרים משכבת הנתונים
  + תבנה על בסיס הנתונים רשימת מוצרים מטיפוס ***ProductItem***(**ישות לוגית**)
  + **תחזיר** את הרשימה שנבנתה
* **בקשת פרטי מוצר** (עבור מסך מנהל ועבור)
  + תקבל מזהה מוצר
  + אם המזהה הוא מספר חיובי - תבצע ניסיון בקשת מוצר משכבת נתונים
  + תבנה אובייקט ***מוצר Product*** (**ישות לוגית**) על בסיס הנתונים שהתקבלו וחישוב מידע חסר
  + **תחזיר** אובייקט מוצר שנבנה
  + תזרוק חריגה מתאימה משלה בקשת מוצר נכשלה (מוצר לא קיים בשכבת נתונים - תפיסת חריגה)
* **בקשת פרטי מוצר** (עבור מסך קונה - מהקטלוג)
  + תקבל מזהה מוצר ואובייקט של סל קניות של הקונה
  + אם המזהה הוא מספר חיובי - תבצע ניסיון בקשת מוצר משכבת נתונים
  + תבנה אובייקט ***פריט מוצר ProductItem*** (**ישות לוגית**) על בסיס הנתונים שהתקבלו וחישוב מידע חסר
  + **תחזיר** אובייקט מוצר שנבנה
  + תזרוק חריגה מתאימה משלה בקשת מוצר נכשלה (מוצר לא קיים בשכבת נתונים - תפיסת חריגה)
* **הוספת מוצר** (עבור מסך מנהל)
  + תקבל (בפרמטרים) אובייקט של מוצר (ישות לוגית)
  + תבדוק תקינות מזהה ( שהוא מספר חיובי), שם (מחרוזת לא ריקה), מחיר (שהוא מספר חיובי), כמות במלאי (אינה שלילית)
  + אם הנתונים תקינים - תבצע ניסיון בקשת הוספה לשכבת הנתונים
  + תזרוק חריגה מתאימה משלה עם הוספת מוצר נכשלה (עקב כפילות מזהה מוצר בשכבת נתונים - תפיסת חריגה, או חוסר תקינות הנתונים כנ"ל)
* **מחיקת מוצר** (עבור מסך מנהל)
  + תקבל מזהה מוצר
  + תבדוק שהמוצר לא מופיע באף הזמנה
  + אם הבדיקה הצליחה - תבצע ניסיון בקשת מחיקה משכבת הנתונים
  + תזרוק חריגה מתאימה משלה עם מוצר מופיע בהזמנות או עם לא קיים מוצר כזה כלל
* **עדכון נתוני מוצר**(עבור מסך מנהל)
  + תקבל אובייקט של ***מוצר*** (שכולל את העדכון)
  + תבדוק את הנתונים (כמו בהוספה)
  + אם הבדיקה הצליחה - תְּבַצֵּעַ ניסיון בקשת עדכון לשכבת הנתונים
  + תזרוק חריגה מתאימה משלה עם עדכון מוצר נכשל (עקב היעדרות מזהה מוצר בשכבת נתונים - תפיסת חריגה, או חוסר תקינות הנתונים כנ"ל)

**פעולות על סל קניות (הכל עבור מסכי קונה בלבד):**

* **הוספת מוצר** ל***סל קניות*** (עבור מסך קטלוג, מסך פרטי מוצר)
  + תקבל אובייקט של ***סל קניות*** ומזהה מוצר
  + אם מוצר לא קיים ב***סל קניות***:
    - תבדוק האם המוצר קיים ויש במלאי
    - יוסיף לסל לפי המחיר הנוכחי של המוצר
    - תגדיר מחיר כולל של פריט כמחיר המוצר הנ"ל
    - תעדכן את המחיר הכולל של ***סל קניות***
  + אם המוצר כבר מופיע בסל קניות:
    - תבדוק האם יש מספיק במלאי של המוצר
    - תגדיל כמות ב-1, תעדכן מחיר כולל של פריט ומחיר כולל של ***סל קניות***
  + **תחזיר** אובייקט ***סל קניות*** (**ישות לוגית**) מעודכן
  + תזרוק חריגה מתאימה במקרה של בעיה כלשהי (לפי הבדיקות כנ"ל)
  + **נ.ב.** בפעולה זו אין בקשות עדכון\הוספה\מחיקה לשכבת נתונים
* **עדכון כמות של מוצר** ב***סל קניות*** (עבור מסך סל קניות)
  + תקבל אובייקט של ***סל קניות***, מזהה מוצר, כמות חדשה
  + תבדוק האם כמות השתנתה:
    - אם הכמות גדלה - תפעל בדומה להוספת מוצר ל***סל קניות*** שכבר קיים ב***סל קניות*** כנ"ל
    - אם הכמות קטנה - תקטין את הכמות בהתאם ןתעדכן מחיר כולל של פריט ושל ***סל קניות***
    - אם הכמות נהייתה 0 - תִּמְחַק את הפריט המתאים מהסל ותעדכן מחיר כולל של ***סל קניות***
  + **תחזיר** אובייקט ***סל קניות*** (**ישות לוגית**) מעודכן
  + תזרוק חריגה מתאימה במקרה של בעיה כלשהי (לפי הבדיקות כנ"ל)
  + **נ.ב.** בפעולה זו אין בקשות עדכון\הוספה\מחיקה לשכבת נתונים
* **אישור סל להזמנה \ ביצוע הזמנה** (עבור מסך ***סל קניות*** או מסך השלמת הזמנה)
  + תקבל ***סל קניות*** (שהפעם כולל את פרטי הקונה - שם, כתובת דוא"ל, כתובת)
  + תבדוק את תקינות כל הנתונים (כל המוצרים קיימים, כמויות חיוביות, יש מספיק במלאי, שם וכתובת קונה לא ריקים, כתובת דוא"ל ריקה או לפי פורמט חוקי)
  + תיצור אובייקט של ***הזמנה*** (**ישות נתונים**) על בסיס הנתונים בסל
    - כל התאריכים "מאופסים" למעט תאריך יצירת הזמנה שהוא "**עכשיו**" (Now)
  + תְּבַצֵּעַ ניסיון בקשת הוספה של הזמנה (**ישות נתונים**) שנוצרה לשכבת הנתונים ותקבל בחזרה מספר הזמנה
  + תבנה אובייקטים של ***פריט בהזמנה*** (**ישות נתונים**) על פי הנתונים מהסל ומספר ההזמה הנ"ל ותבצע ניסיונות בקשת הוספת פריט הזמנה
  + באישור\ביצוע ההזמנה, המוצרים שבהזמנה צריכים לרדת מהמלאי, לכן: תְּבַצֵּעַ ניסיון בקשות מוצרים מתאימים משכבת הנתונים ובקשות עדכון מוצרים אלה לאחר עדכון המלאי
  + תזרוק חריגה מתאימה במקרה של בעיה כלשהי (לפי הבדיקות כנ"ל)

**פעולות על הזמנה:**

* **בקשת רשימת הזמנות** (מסך מנהל)
  + תבקש רשימת הזמנות משכבת הנתונים
  + תבנה על בסיס הנתונים רשימת הזמנות מטיפוס ***OrderForList*** (**ישות לוגית**)
  + **תחזיר** את הרשימה שנבנתה
* **בקשת פרטי הזמנה** (עבור מסך מנהל ומסך קונה)
  + תקבל מזהה הזמנה
  + אם המזהה הוא מספר חיובי - תבצע ניסיון בקשת הזמנה משכבת נתונים
  + תְּבַצֵּעַ ניסיון בקשת פריטי הזמנה משכבת נתונים
  + תבנה אובייקט ***הזמנה*** (**ישות לוגית**) על בסיס הנתונים שהתקבלו וחישוב מידע חסר
  + **תחזיר** אובייקט הזמנה שנבנה
  + תזרוק חריגה מתאימה משלה בקשת מוצר נכשלה (מוצר לא קיים בשכבת נתונים - תפיסת חריגה)
* **עדכון שילוח הזמנה** (מסך ניהול הזמנה של מנהל)
  + תקבל מספר הזמנה
  + תבדוק האם הזמנה קיימת (בשכבת נתונים) ועוד לא נשלחה
  + תעדכן את תאריך הַשִּלּוּחַ בהזמנה (ב**ישות נתונים** וגם ב**ישות לוגית**)
  + לבצע ניסיון עדכון הזמנה לשכבת נתונים
  + **תחזיר** אובייקט הזמנה מעודכן של **ישות לוגית**
  + תזרוק חריגה מתאימה במקרה של בעיה כלשהי (לפי הבדיקות והניסיונות כנ"ל)
* **עדכון אספקת הזמנה** (מסך ניהול הזמנה של מנהל)
  + תקבל מספר הזמנה
  + תבדוק האם הזמנה קיימת (בשכבת נתונים), כבר נשלחו אך עוד לא סופקה
  + תעדכן את תאריך האספקה ב***הזמנה*** (ב**ישות נתונים** וגם ב**ישות לוגית**)
  + לבצע ניסיון עדכון הזמנה לשכבת נתונים
  + **תחזיר** אובייקט ***הזמנה*** (**ישות לוגית**) מעודכן
  + תזרוק חריגה מתאימה במקרה של בעיה כלשהי (לפי הבדיקות והניסיונות כנ"ל)
* **מעקב הזמנה** (מסך ניהול הזמנה של מנהל)
  + תקבל מספר הזמנה
  + תבדוק האם הזמנה קיימת (בשכבת נתונים)
  + תחזיר מופע של ***מעקב הזמנה OrderTracking*** (ישות לוגית) שנוסף למזהה הזמנה ומצב נוכחי שלה, יכיל רשימה של צמדים תאריך ותיאור מילולי של מצב - על כל שינוי במצב (ההזמנה נוצרה, ההזמנה שֻלְּחָה, ההזמנה סופקה)
  + **תחזיר** את האובייקט הנ"ל
  + תזרוק חריגה מתאימה במקרה של בעיה כלשהי (לפי הבדיקות והניסיונות כנ"ל)
* **לבונוס** - עדכון הזמנה (עבור מסך מנהל)
  + יאפשר הוספה \ הורדה \ שינוי כמות של מוצר בהזמנה ע"י המנהל (שימו לב מתי מותר לעשות את זה!)
  + אין יותר פירוט (כי זה לבונוס) - אך ניקוד הבונוס יינתן (בפרויקט הסופי) רק במקרה של השלמת כל הפונקציונליות (כולל בשכבת התצוגה) בצורה מלאה.

**תהליך קניה:**

לצורך הבנה של תהליך קניה ושימוש בסל קניות, נתאר את התהליך הן מבחינת מעבר בין חלונות\מסכים והן מבחינת שימוש במופע של סל קניות.

1. מופע סל הקניות נוצר בשכבת התצוגה בעת כניסת הקונה למערכת, הסל מתרוקן\נמחק כאשר הקונה יוצא מהמערכת (בדומה לכניסה לאתר קניות ויציאה מהאתר). המופע של סל יישמר לצד או בתוך האובייקט של חלון הקונה
2. **כניסת קונה למערכת**

כאשר קונה נכנס למערכת, הוא מקבל מסך של קטלוג - רשימת מוצרים. הקונה יכול להיכנס לפרטי כל מוצר - ושם יוצגו בין היתר: זמינות המוצר במלאי (כן או לא), כמות פרטי מוצר בסל הקניות שלו, אפשרות להוסיף פריט לסל ולהוריד פריט מהסל. בכניסה למערכת אין פריטים בסל ולכן בכל המוצרים כמות בסל תהיה אפס.

1. **ניהול הסל**

בחלון הקונה עם הקטלוג, יוצג כפתור "סל קניות" (אפשר גם להציג את כמות הפריטים בסל אם יש כאלה). בלחיצה על הכפתור נפתח חלון ניהול סל קניות ובו מופיעים:

* רשימת פרטי הזמנה - לפי מוצר: שם מוצר, כמות פריטים, מחיר פריט, מחיר כל הפריטים, כפתור הוספת פריט מוצר, כפתור הורדת פריט מוצר, כפתור מחיקת מוצר מהסל
* כפתור ריקון הסל
* כפתור סיום\ביצוע הזמנה (זאת אומרת - שליחת\יצירת הזמנה על בסיס נתוני הסל)
* כפתור חזרה לקטלוג מוצרים (אם בחרתם לעבוד בשיטה של Single Page Application עבור כל קונה) או סגירת חלון סל קניות (אם בחרתם לפתוח חלון נפרד לכל קונה ולכל פעילות)

כמו כן, בחלון ניהול הסל יופיע סה"כ מחיר כל הפריטים בסל הקניות שיתעדכן בכל שינוי בסל.

1. **הוספת מוצר או פריט מוצר לסל**

בקשת הוספה יכולה להתבצע ממסך פרטי מוצר. בקשת הוספת פריט של מוצר שכבר מופיע בסל יכולה להתבצע ממסך ניהול סל קניות. בהוספת מוצר לסל או הגדלת כמות פריטי מוצר בסל, שכבת התצוגה "תשלח" בקשה לשכבה הלוגית, ובבקשה ייכללו מופע של סל קניות ומספר מוצר (מזהה מוצר). השכבה הלוגית תבדוק האם אפשר להוסיף לסל את המוצר או פריט נוסף של המוצר ותחזיר מופע מעודכן של קניות (בהתאם) והמידע המועדכן יוצג לקונה. אם קרתה תקלה כלשהי בשכבה הלוגית - היא תזרוק חריגה מתאימה, ושכבת התצוגה תודיע לקונה בדרך כלשהי על כישלון הבקשה שלו ועל סיבתה. **נ.ב.** לא מתבצע עדכון מלאי המוצר בשינויים בסל קניות.

1. **הורדת מוצר או פריט מוצר לסל**

בקשת הורדה יכולה להתבצע ממסך פרטי מוצר או ממסך ניהול סל קניות. בהורדת מוצר מסל או הקטנת כמות פריטי מוצר בסל, שכבת התצוגה "תשלח" בקשה לשכבה הלוגית, ובבקשה ייכללו מופע של סל קניות ומספר מוצר (מזהה מוצר). השכבה הלוגית תחזיר מופע מעודכן של קניות (בהתאם) והמידע המועדכן יוצג לקונה. אם קרתה תקלה כלשהי בשכבה הלוגית - היא תזרוק חריגה מתאימה, ושכבת התצוגה תודיע לקונה בדרך כלשהי על כישלון הבקשה שלו ועל סיבתה. **נ.ב.** לא מתבצע עדכון מלאי המוצר בשינויים בסל קניות.

1. **שליחת\יצירת הזמנה**

בסיום ההזמנה הקונה נכנס למסך סל קניות ומבקש לשלוח את ההזמנה למערכת. ייפתח חלון השלמת פרטי הזמנה (שם הלקוח וכו'), ולאחר השלמת הנתונים תִּשַּלַח בקשה לשכבה הלוגית יחד עם מופע של סל קניות ופרטי הוספה נוספים כנ"ל. השכבה הלוגית תבדוק את חוקיות ההזמנה (כולל בדיקה מחדש של מלאיי המוצרים שבסל וכו'), ואם הכל תקין - תיצור את ההזמנה ואת פרטי ההזמנה בהתאם. אם קרתה תקלה כלשהי בשכבה הלוגית - היא תזרוק חריגה מתאימה, ושכבת התצוגה תודיע לקונה בדרך כלשהי על כישלון הבקשה שלו ועל סיבתה. בסיום מוצלח של שליחת ההזמנה יְאֻפּס סל הקניות של הקונה על מנת לאפשר לבצע קניה חדשה.

## שכבת התצוגה

נתחיל להגדיר את שכבת התצוגה.

**בשלב 3** נגדיר ממשק בסיסי עבור המנהל ובשלבים הבאים תייצרו את יתר הממשק הגרפי .

לצורך הפשטות, החלון הראשי יוביל לאפשרויות השונות כמתואר ב-[**איור 1**. תרשים מסכים של ממשק משתמש](#bjrbo0vczais) ("1 מסך ראשי").

תחילה ניצור ממשק גרפי שיאפשר למנהל לנהל את מלאי המוצרים בחנות.

לצורך זאת נוסיף 2 חלונות :

* חלון עבור הצגת רשימת המוצרים, בחלון זה נוכל לסנן את המוצרים לפי ה**קטגוריה**.

חלון זה ישתמש בישות הלוגית **ProductForList**.

* חלון עבור הוספת או עריכת פרטים עבור מוצר.

חלון זה ישתמש בישות הלוגית הראשית **מוצר** (**Product**)

החלון ייפתח מתוך חלון הצגת רשימת המוצרים בשני מצבים אפשריים :

1. מצב הוספת מוצר חדש

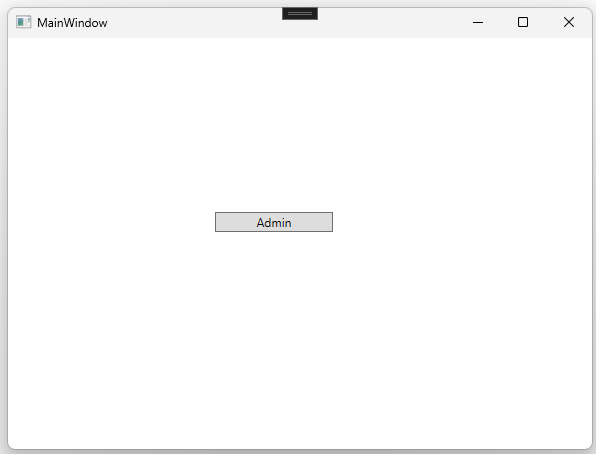
ייפתח חלון שיאפשר הכנסת כל הפרטים הנדרשים על מוצר (כגון : שם, מחיר, קטגוריה)

1. מצב "פעולות" שיציג את הפרטים הנוכחיים עבור מוצר קיים ויאפשר עדכון ופעולות. כרגע אין פעולות נוספות מלבד עדכון אבל בהמשך ייתכן ויהיו. (מצב זה ייפתח על ידי לחיצה כפולה על שורת תיאור מוצר מסוים בחלון הרשימה,.

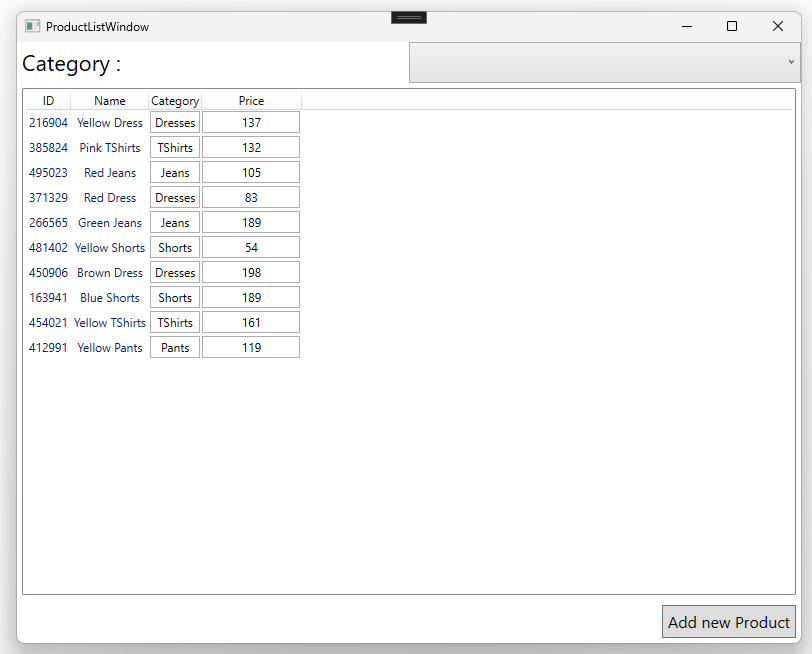
**שימו לב : בשלב 3 אין להוסיף חלונות נוספים, אלא רק בהמשך לאחר שיילמדו מנגנונים נוספים.**

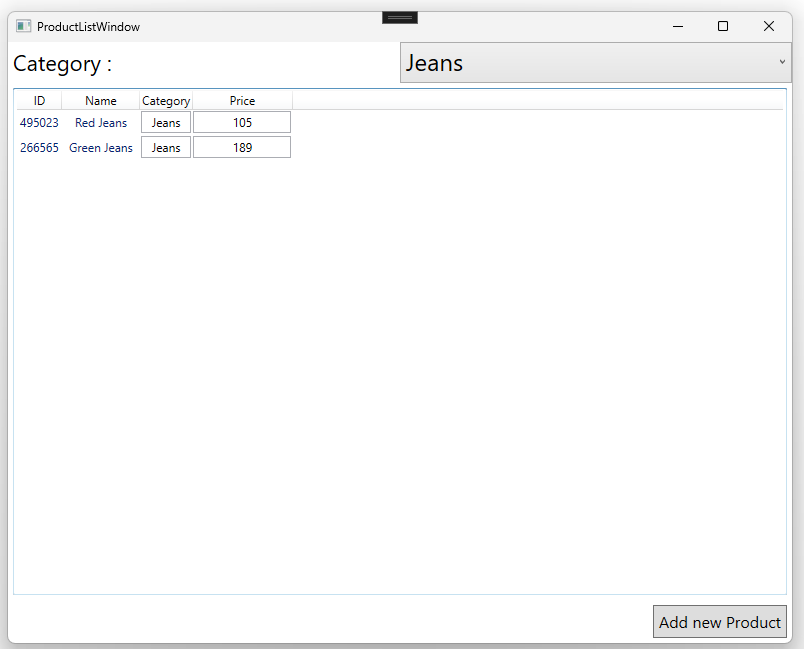
נשלים את פעולות עיצוב החלונות לפי השלבים המתוארים בהנחיות המפורטות של שלב 3, כאשר כל פעם שמוזכרת הישות הראשית *BoEntity* תהיה הכוונה ליישות Product וכל פעם שמדובר על חלון תצוגת רשימה הכוונה שתוצג רשימה של הישות ProductForList.

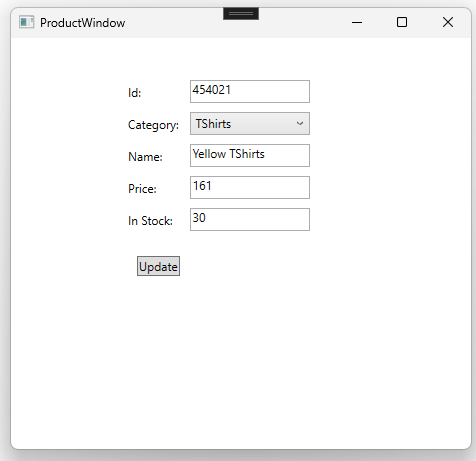
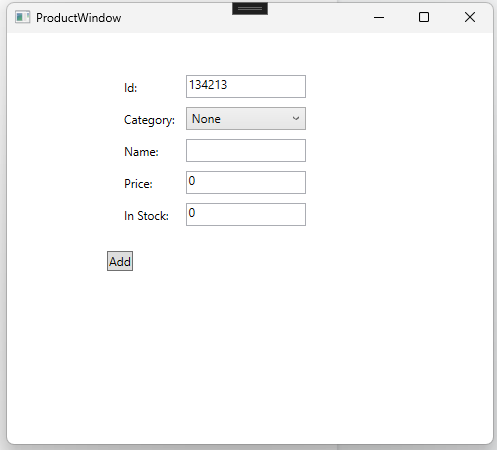
דוגמת הרצה :

חלון ראשי

חלון תצוגת רשימת מוצרים



חלון תצוגת רשימת מוצרים - רשימה מסוננת



חלון מוצר - מצב הוספה חלון מוצר - מצב עדכון

## השלמת שכבת התצוגה (שלב 5)

כאמור, המערכת שלנו תאפשר **ניהול** מרכז המידע של החברה המספקת את שירות הזמנת מוצרים

וכן אפשרות **לבצע הזמנות** **ולעקוב** אחרי המצב שלהם.

**נפרט להלן את התצוגות השונות הקיימות במערכת.**

**עבור כל תצוגה, את התצוגות המסתעפות ממנה.**

* **תצוגת מסך כניסה:**
  + תצוגת מנהל
  + תצוגת הזמנה חדשה
  + תצוגת מעקב אחר הזמנות
  + (בונוס: ביצוע הפעולות כמשתמש רשום)
* **תצוגת מנהל :**
  + תצוגת רשימת המוצרים **ProductForList**
    - מעבר לחלון פרטי מוצר (אפשרות עריכה)
    - מעבר להוספת מוצר
  + תצוגת רשימת הזמנות **OrderForList**
    - מעבר לחלון פרטי הזמנה **Order**

יש לאפשר ביצוע [עדכון פרטי הזמנה](#a43eya8vedzs) (שימו לב שניתן לבצע את כל הפעולות המותרות בהזמנה לפי ההנחיות בקישור לעיל)

* **חלון מוצר:**
  + חלון המציג את פרטי המוצר.

שימו לב: יש לאפשר מצב תצוגה בלבד, ומצב עריכה.

* **חלון הזמנה:**
  + חלון המציג את פרטי ההזמנה.

שימו לב: יש לאפשר מצב תצוגה בלבד, ומצב עריכה.

* **יצירת הזמנה חדשה:**
  + הצגת הקטלוג ע"י **ProductItem**
    - יש לאפשר סינון ע"פ קטגוריה ( filter )
    - יש לאפשר מצב תצוגה של רשימה "מקובצת"(grouping) ע"פ קטגוריה
    - יש לאפשר תצוגה של חלון מוצר **ProductItem** (ללא אפשרות לשינויים) מתוך הקטלוג
    - מעבר לתצוגת סל קניות, עבור אישור הזמנה.
* **תצוגת סל קניות:**
  + הצגת הפריטים העדכניים בסל הקניות
  + יש לאפשר שינויים בסל הקניות (שינוי כמות, הסרת פריט)
  + אישור סל הקניות, השלמת ההזמנה.
* **מעקב הזמנה:**
  + הצגת **מעקב הזמנה OrderTracking**
  + מעבר להצגת פרטי ההזמנה **Order** לתצוגה בלבד (ללא אפשרות לשינויים)
* **בונוס: משתמש רשום:**
  + המערכת תאפשר **לשמור משתמש רשום**, שעבורו ה-Cart נשמר גם אם יצא ללא ביצוע הזמנה
  + במקרה שההזמנה לא הושלמה, לא תיווצר ישות הזמנה
  + עבור הבונוס הזה, יש להוסיף בשכבת נתונים ישויות משתמשים,
  + יש לחשוב על איך ובעזרת אלו ישויות ינוהל סל קניות ופריטים בסל של כל לקוח רשום, וכו'
  + תינתן אפשרות להירשם ולהיכנס למערכת כלקוח
  + במידה והמשתמש יצא מחלון ההזמנה ללא השלמת ההזמנה, הסל קניות נשמר עבורו, ויוצג בפעם הבאה שייכנס לחלון ביצוע הזמנה
  + תתאפשר הצגת הזמנות של הלקוח בלבד
  + יש לשים לב האם נדרשות עוד הגדרות וישויות על מנת לממש את הבונוס הזה
  + תתאפשר גם טיפול בלקוח מזדמן - כפי שזה עובד ללא הבונוס הזה (עם סל קניות השמור בשכבת התצוגה בלבד)

***נ.ב. אנחנו מזהירים שמדובר בתוספת כבדה כבר ברמה תכנונית***

**הערות כלליות:**

* תצוגת פרטי הישות תהיה ע"פ הישות של BO (מוצר, הזמנה…)
* תצוגת הרשימות תהיה ע"פ ישות לרשימה מתאימה של BO (מוצר לרשימה, וכו')

## שמירת נתונים בקבצי XML

כפי שכתוב בפירוט שלב 6, יש לשמור קובץ **xml** נפרד עבור כל ישות הנתונים (DO). כלומר 3 קבצי **xml**, המקבילים לרשימות ב-DataSource שמהווים את מקור הנתונים ב-DalList.

**שמירת נתוני הקונפיגורציה בקובץ config.xml:**

ה"מספרים הרצים" הנדרשים לחלק מהישויות. הנתונים שיתווספו לקובץ הם מספר רץ עבור המפתח הייחודי של הזמנה ושל פריט ההזמנה (ועבור ישויות נוספות שתוסיפו לבונוסים למי שיוסיף בהם מזהה אוטומטי שהוא מספר רץ).

מבחינת המימוש, **מומלץ** לבחור למימוש LINQ-to-XML מלא את הישות של מוצר. אך אם בחרתם ישות אחרת - זה יתקבל.

## סימולטור (שלב 7)

אנחנו רוצים להוסיף תהליך של הדמיה (סימולציה) למערכת שלנו, זאת כדי שנוכל לראות כיצד המערכת שלנו מתפקדת לאורך זמן.

בשלב זה נוסיף סימולטור שתפקידו יהיה לעדכן את סטטוס ההזמנה ואת השדות הרלוונטים הנלווים בצורה אוטומטית. בכל צעד של הסימולציה תיבחר ההזמנה הבאה למשלוח או לאספקה ע"פ סדר עדיפות כדלהלן, עד שאין יותר הזמנות שעומדות בקריטריונים לשליחה. אנו נראה את מספר ההזמנה שבהדמיה כרגע ואת שינוי סטטוס ההזמנה בתצוגה של חלון הסימולטור באופן מיידי.

העדיפות על פיה יעבוד הסימולטור: עובר על כל ההזמנות שנוצרו (אבל עדיין לא נשלחו) או נשלחו (אבל עדיין לא סופקו). לטיפול ההדמיה כל פעם נבחרת לקידום ההזמנה בה השינוי האחרון בסטטוס הוא הישן ביותר.

**בשכבת התצוגה PL:**

* מבחינה עיצובית - הפעלת הסימולטור תתבצע בחלון הראשי על ידי כפתור שמפעיל את הסימולציה
* על מנת לעקוב אחר פעולת הסימולטור ב**חלון הסימולטור**, נוסף לשעון, בכל רגע נתון יופיעו
  + מספר ההזמנה המטופלת עכשיו
  + סטטוס ישן של הזמנה וזמן תחילת הטיפול
  + סטטוס עתידי של ההזמנה וזמן "משוער" של סיום הטיפול (ראו בהמשך)
  + לבונוס: progress bar של התקדמות ההזמנה בטיפול
* לבונוס: מצב ההזמנות יתעדכן אוטומטית בחלונות הבאים (אם ו**כאשר** הם פתוחים):
  + **חלון OrderTracking**
  + **חלון רשימת ההזמנות** של המנהל
    - מי שמשתמש בחלון הזה ב-ObservableCollection, יוכל לבצע את העדכונים "יחידנית" עבור ההזמנה הרלוונטית בלי לבקש מחדש את כל הרשימה משכבת ה-BL
  + **חלון ניהול ההזמנה** של המנהל

**בשכבה לוגית BL:**

* הוסיפו בממשק הישות הראשית של ההזמנה ובמימוש שלו מתודת בחירת ההזמנה הבאה לטיפול
  + המתודה לא תקבל פרמטרים
  + המתודה תחזיר את מספר ההזמנה הכי "ישנה" ע"פ המתואר בסדר העדיפויות של בחירת הזמנה לטיפול בתיאור הכללי של הסימולטור לעיל
  + אם לא נמצאה הזמנה מתאימה - יוחזר null (אם אתם מחזירים מספר הזמנה - הסוג המוחזר יהיה מתאפס - **?int**).
* לבונוס:
  + רשימת הזמנות תמוין לפי מספר הזמנה או לפי לפי קריטריון העדיפות של הסימולטור כנ"ל ((בשילוב של שינוי מתאים שכבת PL)
    - דאגו למנוע העתקת קודים ולממש את הדרישה המתאימה לעיל ואת המיון לפי סדר העדיפויות לוך שימוש במתודת עזר משותפת
  + בממשק ובמימוש הישות הראשית של הזמנה:
    - יוגדר ב-BO דלגט שיקבל בפרמטרים סוג אירוע (הוספה, עדכון, מחיקה - אם עשיתם) ונתונים נלווים מתאימים לפי החלטתכם (למשל - אובייקט הזמנה מתאים או מספר הזמנה וכו')
    - סוג האירוע יכול להיות מספר או enum חדש שיוגדר ב-BO
    - יוגדר אירוע (event delegate) אחד עבור כל סוגי האירועים של הזמנה הכנ"ל
    - יוגדרו מתודת הרשמה (Register) ומתודת ביטול הרשמה (Unregister) לאירוע הזה
    - **הוסיפו זימון האירוע** בכל המתודות שמוסיפות\מעדכנות\מוחקות הזמנה - **לאחר השלמה מוצלחת** של הפעולה על ההזמנה (כולל פניות מתאימות ל-DAL)

1. **aka** - also known as … - **… ידוע גם כ** [↑](#footnote-ref-0)