



# פרויקט גמר מה"ט

לתואר הנדסאי במחלקה: תוכנה

שם הפרויקט: מערכת ניהול מוצרים

שם הסטודנט/ים: סאמי חיחי/יוסף חיר

ת.ז: 207417874/318741444

מקום ביצוע העבודה : אורט בראודה - כרמיאל

תאריך הגשה: 09/09/2021

שם המנחה : שאדי עסאקלה

חתימת ראש המחלקה

חתימת המנחה

\_\_\_\_\_  
תאריך

\_\_\_\_\_  
תאריך

**הצהרת סטודנט**

שם הסטודנט: \_\_\_\_\_ סאמי חי'חי' \_\_\_\_\_ ת.ז. \_\_\_\_\_ 318741444

שם המכללה בה לומד הסטודנט: \_\_\_\_\_ אורט בראודה הנדסאים- כרמיאל \_\_\_\_\_

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.  
פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.  
פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.  
מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.  
אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.



חתימת הסטודנט: \_\_\_\_\_ תאריך: \_\_\_\_\_ 01/09/2021

**אישור המנחה האישי**

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנחה: \_\_\_\_\_ חתימה: \_\_\_\_\_ תאריך: \_\_\_\_\_

**אישור ראש המגמה**

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר.

שם ראש המגמה: \_\_\_\_\_ חתימה: \_\_\_\_\_ תאריך: \_\_\_\_\_

**הצהרת סטודנט**

שם הסטודנט: \_\_\_\_\_ יוסף חיר \_\_\_\_\_ ת.ז. 207417874

שם המכללה בה לומד הסטודנט: \_\_\_\_\_ אורט בראודה הנדסאים- כרמיאל \_\_\_\_\_

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.  
פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.  
פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.  
מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.  
אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.

חתימת הסטודנט: \_\_\_\_\_  תאריך: 01/09/2021

**אישור המנחה האישי**

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנחה: \_\_\_\_\_ חתימה: \_\_\_\_\_ תאריך: \_\_\_\_\_

**אישור ראש המגמה**

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר.

שם ראש המגמה: \_\_\_\_\_ חתימה \_\_\_\_\_ תאריך: \_\_\_\_\_

חוזר מנהל מה"ט 15-4-55 – נספח מס' 5) הצעה לפרויקט גמר)

תאריך: 12.1.2021

לכבוד  
יחידת הפרויקטים  
מה"ט

**הצעה לפרויקט גמר**

\* יש להדפיס את כל הנתונים הנדרשים

**א. פרטי הסטודנטים**

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים
יוסף חיר	207417874	פקיעין	0542077433	2021
סאמי חיחי	318741444	פקיעין	0545472527	2021

שם המכללה: \_\_\_\_\_ אורט בראודה

סמל המכללה: 271015

מסלול הכשרה: הנדסאים / מכנאים מוסמכים (מחק את המיותר)

מגמת לימוד: \_\_\_\_\_ הנדסאי תוכנה

מקום ביצוע הפרויקט: \_\_\_\_\_ אורט בראודה

**ב. פרטי המנחה האישי**

שם המנחה *	כתובת	טלפון נייד	תואר	מקום עבודה / תפקיד
ד"ר שאדי עסאקלה	מגאר	0505487741	PhD	רמ"ח

\* עבור מנחה אישי חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט

\_\_\_\_\_  
חתימת הגורם המקצועי מטעם מה"ט

\_\_\_\_\_  
חתימת המנחה

\_\_\_\_\_  
חתימת הסטודנט

## 1 שם הפרויקט

תוכנת ניהול מוצרים במלאי.

## 2. רקע

### 2.1. תיאור ורקע כללי

תוכנת ניהול מוצרים במלאי ניהול המלאי וכל עסקאות הרכש הוא החלק בתוכנה המאפשרת על כל יציאה וכניסה מהמלאי של כל המוצרים, ועל המשתמש לבחור את הקטגוריה מתאימה לכל מוצר. תוכנת ניהול המוצרים מאפשרת למשתמש:

- א. עדכון נתונים של מוצר במלאי.
- ב. הוספת מוצרים למלאי.
- ג. הוצאת מוצרים מהמלאי על המשתמש לבצע הרשמה לתוכנה לפני כניסה לתוכנה, ועל מנת לבצע הזמנה ללקוח על המשתמש להוסיף את הלקוח לרשימת הלקוחות וכל הזמנה נשמרת ברשימת ההזמנות. התוכנה מציגה את הרווח היומי היא מבצעת חישוב ההכנסות מינוס ההוצאות.

התוכנה יודעת לחשב אוטומטי את כמות המוצרים אם כמות המוצרים קרובה להסתיים מוציאה הודעה מתאימה, וגם מוציאה הודעה מתאימה אם חסר מוצרים במלאי, והתוכנה מאפשרת למשתמש לבצע חיפוש מוצר מהמלאי, ויודעת גם לחשב כמה מוצרים נמכרו בשבוע. התוכנה יודעת לבצע תרגום לשפה מסוימת לפי בחירה של המשתמש לדוגמא: מאנגלית לעברית. בנוסף, המערכת צריכה לעשות חישובים אוטומטיים. למשל מוצרים שמגיעים הכי הרבה לאחסון במלאי, מה הצפי של ציוד שנגמר. חישוב מקום אחסון לפי כמויות על המדפים והמערכת תדע לפצל את המחסן בהתאם וכו'..

### 2.2. מטרות המערכת

1. מערכת קלה להתאמה לצרכי העסק.
2. המערכת מאפשרת לנהל את המוצרים במלאי בצורה יעילה.
3. המערכת יודעת לנהל את הלקוחות.
4. המערכת יודעת לנהל את הזמנות של הלקוחות.
5. המערכת יודעת לנהל את כל הקטגוריות של המוצרים.
6. המערכת יודעת לחשב את הרווח היומי.
7. המערכת יודעת לחשב כמה מוצרים נמכרו בשבוע.
8. המערכת יודעת לנהל את המוצרים על המדפ

### 3. סקירת מצב קיים בשוק

דוגמה לשתי תוכנות קיימים בשוק:

5. ABC Inventory:

זוהי תוכנה פיתוחה של חברת Almyta Control System, בנויה באופן פשוט וקל להבנה נועדה במיוחד לעסקים קטנים עד בינוניים.

הגרסה החינמית של התוכנות פותחת בפני הלקוח כמעט את כל האפשרויות – למעט תמיכה טכנית וסנכרון בין עמדות עבודה (מחשבים) שונות – של הגרסה המלאה אשר בתשלום.

התוכנה כוללת מאפיינים לוגיסטיים החשובים לכל עסק ועוזרת לכם לנהל מכירות, שינויים במלאי ובמחסנים, פרטי משלוח וייצוא\ייבוא וניהול חשבונות וחשבונות.

התוכנה החינמית פשוטה, בסיסית ונוחה לשימוש ואינה כוללת מאפיינים ופונקציות מתקדמות או עיצוב חדשני ומלהיב, היא פשוט מאוד הבסיס של תוכנות ניהול מלאי.

### 2. Delivrd:

זוהי תוכנה מבוססת ענן לניהול מלאי ומכירות בעלת שתי גרסאות: גרסה חינמית בסיסית וגרסה בתשלום בעלת אפשרויות נרחבות יותר.

אחסון המידע בענן ויכולת השיתוף בין מכשירים שונים ומספר עמדות הוא נוח ויעיל ומעניק לתוכנה זו את היתרון המשמעותי ביותר.

לעסקים קטנים בעלי מיקוד מצומצם ומספר מוצרים מוגבל – פתרון מעולה לחברות קטנות וממוקדות המציעות מוצרים לשימוש ייעודי.

התוכנה עוזרת לכם לנהל את המחסן שלכם וגם את המכירות, רווח והפסד, הוצאות.

### 4. מה הפרויקט מחדש/משפר

בנוסף לניהול המלאי המערכת תעזור בסידור המוצרים בתוך המחסן של החנות. המערכת תדע לזהות את סוגי המוצרים ולחלק אותם לפי קטגוריות על המדפים. המערכת תדע לזהות משקלים ותכונות נוספות בכדי לאפשר חלוקה נכונה על המדפים. כלומר, לכל מדף יהיה מידע אילו מוצרים הוא יוכל להכיל.

## 5. דרישות

### 5.1. דרישות מערכת

סביבת מערכת: ג'אווה(eclipse), MySQL,

מערכת הפעלה : windows

זיכרון : מינימום 4GB .

כמות משתמשים: התוכנה מיועדת לבעל העסק (אחד).

### 5.2. דרישות פונקציונאליות

זיהוי	נוסח דרישה	סיווג
5	המערכת מאפשרת למשתמש לבצע רישום לתוכנה	FR
2	על המשתמש להכניס את הפרטים האישיים שלו כדי לבצע רישום	NFR
0	המערכת מאפשרת למשתמש לבצע כניסה לתוכנה.	FR
4	על המשתמש לרשום שם משתמש וסיסמה כדי לכנס לתוכנה.	NFR
1	המערכת מאפשרת למשתמש להוסיף מוצרים למלאי.	FR
8	על מנת להוסיף מוצר למלאי על המשתמש להכניס את הנתונים של המוצר.	NFR
0	המערכת מאפשרת למשתמש לבצע עדכון למוצר במלאי.	FR
7	על מנת לעדכן מוצר במלאי על המשתמש לשנות את הנתונים של המוצר.	NFR
9	המערכת מאפשרת למשתמש לבצע מחיקת מוצר מהמלאי.	FR
5.	על מנת למחוק מוצר מהמלאי על המשתמש לבצע מחיקה של מוצר מהמלאי.	NFR
55	המערכת מאפשרת למשתמש לבצע חיפוש של קטגוריה במלאי.	FR
52	על מנת לבצע חיפוש על המשתמש להכניס את שם הקטגוריה.	NFR
50	המערכת מאפשרת למשתמש להוסיף קטגוריות חדשות.	FR
54	על מנת להכניס קטגוריה חדשה על המשתמש להכניס שם הקטגוריה החדשה.	NFR
51	המערכת מאפשרת למשתמש לבצע מחיקה של קטגוריות.	FR
58	על מנת לבצע מחיקה של קטגוריה על המשתמש לבצע מחיקה של קטגוריה.	NFR
50	המערכת מאפשרת להציג את כל הקטגוריות של המוצרים במלאי.	FR

57	המערכת מאפשרת להציג את המוצרים במלאי.	FR
59	המערכת מאפשרת להכניס פרטי לקוח לרשימת הלקוחות.	FR
2.	על מנת לבצע הכנסת לקוחות לרשימה על המשתמש להכניס פרטי הלקוח.	NFR
25	המערכת מאפשרת להציג את רשימת הלקוחות	FR
22	המערכת מאפשרת למשתמש לבצע הזמנה ללקוח.	FR
20	המשתמש יכול להוסיף מוצרים להזמנה של לקוח.	NFR
24	המערכת מאפשרת להציג את הכנסות של העסק	FR
21	המערכת מאפשרת להציג את הוצאות של העסק	FR
28	המערכת מאפשרת הצגת הרווח של העסק	FR
27	המשתמש יכול לבצע הכנסת מוצרים באמצעות רשימת המוצרים למלאי	NFR
28	המערכת מאפשרת למשתמש לשנות את השפה	FR

## 6. בעיות צפויות במהלך הפיתוח ופתרונות

### 6.1. תיאור הבעיות

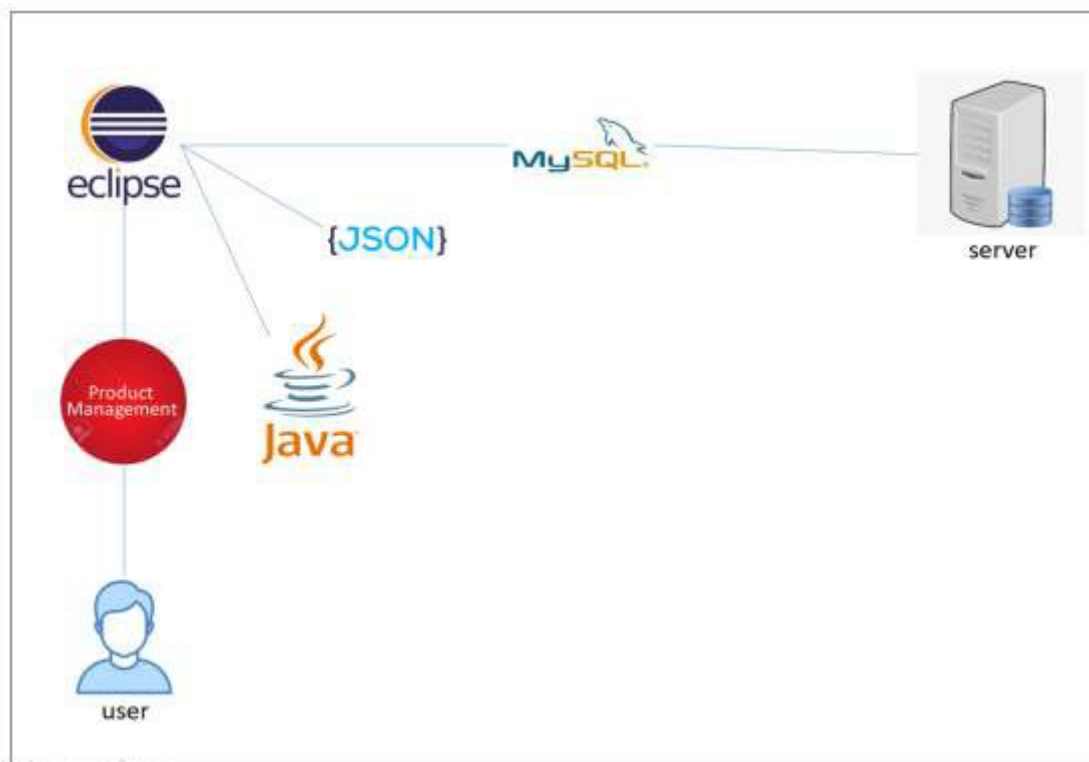
### 6.2. פתרונות אפשריים

בעיה	פתרונות אפשריים
משתמשים לא קיימים	-מוציא הודעה מתאימה. -צריך לרשם לתוכנה.
סיסמה שגויה	מוציא הודעה מתאימה(סיסמה שגויה).
הכנסת מוצר לקטגוריה לא קיימת	-מוציא הודעה מתאימה. (אין קטגוריה כזאת.) -הוספת קטגוריה זו לרשימת הקטגוריות.
מחיקת מוצר לא קיים	-מוציא הודעה מתאימה. (נמחק המוצר מהמלאי).
כמות מוצר קרובה להסתיים	-מוציא הודעה מתאימה. (הכמות קרובה להסתיים אם אתה רוצה להזמין). -הזמנת מוצרים למלאי.



הוספת קטגוריה קיימת	-מוציא הודעה מתאימה. (קטגוריה קיימת)
הוספת לקוח קיים ברשימת הלקוחות	-מוציא הודעה מתאימה. (לקוח זה קיים).
הזמנת מוצר לא קיים	-מוציא הודעה מתאימה. (מוצר זה לא קיים).
הכנסת כמות של מוצר מסוים שווה ל אפס או פחות	-הודעת שגיאה(צריך להכניס מספר גדול מאפס). - הכנסת כמות של המוצר גדולה מאפס.
הכנסת מחיר של מוצר מסוים שווה לאפס או פחות.	-הודעת שגיאה(צריך להכניס מספר גדול מאפס). -הכנסת מחיר של המוצר גדול מאפס.

## 7. פתרון טכנולוגי נבחר 7.1. טופולוגית הפתרון



הטיפולוגיה מציגה את תרשים זרימת הנתונים המשתמשים מחוברים לתוכנה (management products) התוכנה מוצגת על ידי eclipse בשפת ג'אווה .  
MySQL מחובר לשרת לשמירת נתונים וצריך לקשר בין MySQL ל eclipse כדי להעביר הנתונים לבסיס הנתונים.  
JSON בשביל לבצע תרגום למילים צריך לקשר קובץ JSON ל Eclipse.

### 7.2.1.1 טכנולוגית בשימוש

mySql - לשמירת הנתונים בבסיס הנתונים.

Eclipse - פיתוח התוכנה

JSON - בשביל לבצע תרגום למילים.

### 1.2 שפות הפיתוח

Java כדי לפתח את התוכנה צריך להשתמש בשפת תכנות ואנחנו בחרנו ב java וכך מאפשרת למשתמש לבצע פעולות לצרכי המשתמש.

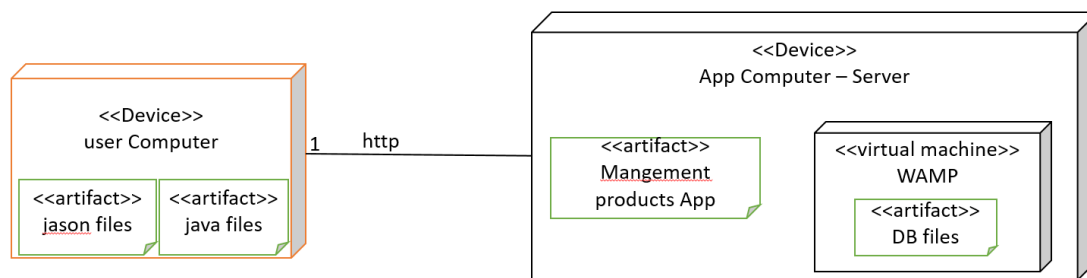
### 1.3 תיאור הארכיטקטורה הנבחרת

ריכוזית: מחשב אחד מרכזי הכל.

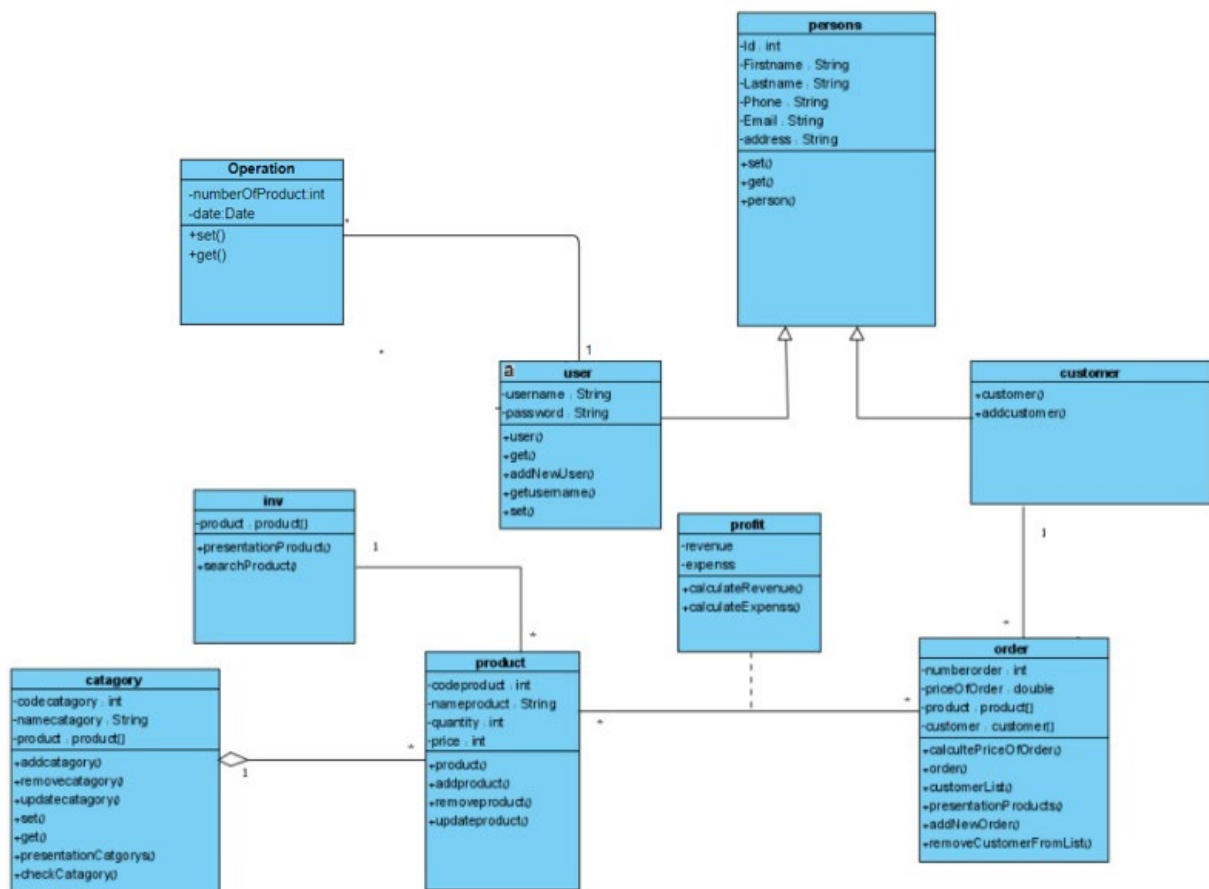
בחרנו את הארכיטקטורה זה כי:

-עלות נמוכה.

-אבטחה גבוהה.



### 1.4 חלוקה לתוכניות ומודולים



\***מחלקת משתמש:** המשתמש יורש ממחלקת אדם והמשתמש יכול לבצע כמה פעולות (כל פעם המשתמש מוכר מוצר המערכת שומרת כמה מוצרים נמכרו בטבלת הפעולות בבסיס הנתונים).

\***מחלקת אדם:** מחלקת אדם יש בה תכונות לאדם ויש מחלקת לקוח ומחלקת משתמשים יורשים ממנה.

\***מחלקת לקוח:** לקוח יורש מהאדם את התכונות שלו ויש לו פעולה שמוסיפה לקוח חדש לרשימת הלקוחות.

\***מחלקת קטגוריה:** קטגוריה מקושרת עם מוצר מקשר יחיד לרבים שבכל קטגוריה יש בה כמה מוצרים. ויש בה כמה פעולות מחיקה ועדכון קטגוריות ואם רוצים להוסיף קטגוריה חדשה יש פעולה שבודקת אם קטגוריה זו קיימת

\***מחלקת מוצר:** מוצר מקושר עם קטגוריה מקשר יחיד לרבים ויש בו פעולות שעושות מחיקה ועדכון למוצר והוספת מוצר.

\***מחלקת המלאי:** היא מקושרת עם מוצר מקשר יחיד לרבים שבמלאי יש הרבה מוצרים.

\***מחלקת הזמנה:** הזמנה מקושרת עם לקוח מקשר יחיד לרבים שלכל לקוח יש כמה הזמנות ויש בה כמה פעולות הוספת המנה חדשה, חישוב מחיר ההזמנה.

## 1.5 סביבת השרת

ייעשה שימוש בשרת מקומי localhost

## 1.6 ממשק משתמש

Design:

Form-> מספק תבניות עיצוב

דף כניסה לתוכנה.

The image shows a web browser window titled "Mangement products - Login". The page has a light gray background. At the top center, the word "login" is displayed in a large, dark gray font. Below it is a square placeholder with a black 'X' inside. Further down, there are two input fields. The first is labeled "User name:" and the second is labeled "password:". Below the password field, there are two buttons: one labeled "sign up" and another labeled "Login", both in blue text. The window has standard minimize, maximize, and close buttons in the top right corner.

דף הרשמה לתוכנה.

Mangement products - Sign up

Sign up

Full name :

Address:

p.number:

User name:

password:

☒ accept terms

sign up

הצגת מלאי אחרי שהמשתמש בצע כניסה.

Mangement products - Inventory

Inventory

catagory

customer

order

profit

Add Product

Inventory

Product name

Quantity

Price

Catagory

Action

milk

200

5

Milky

Remove

Edit

לחיצה על כפתור "Add Product" נפתח דף של הוספת מוצר.

Mangement products - Add product

**Product**

**Product Name:**

**Quantity:**

**price:**

**Catagory:**

דף הקטגוריה.

Mangement products - Catagory

[Inventory](#) [catagory](#) [customer](#) [order](#) [profit](#)

**catagory**

**Add Catagory:**

**catagory list:**

id	catagory	Action
1	Milky	<input type="button" value="Remove"/> <input type="button" value="Edit"/>

דף הוספת הלקוחות לרשימת הלקוחות בדף ההזמנות.

Mangement products - Customer

Inventory

catagory

customer

order

profit

customer

Full name :

Email:

Address:

P.Number:

Clear

Add customer

דף הזמנה.

Mangement products - Order

Inventory

catagory

customer

order

profit

Order

Customer list

id	Name	Action
1	yosef khier	<div>Remove</div> <div>Edit</div>

Product name:

Quantity:

Order

Order list

Product name	Quantity	Price	Catagory	Action
milk	200	5	Milky	<div>Remove</div> <div>Edit</div>

Price total: 1000

רווח יומי (הכנסות והוצאות).

Mangement products - Profit

Inventorycategorycustomerorderprofit

Profit

Revenue

Order id	OrderPrice
1	2000

Total: 2000

Expenses

Product id	Quantity	Product price
4	100	1500

Total: 1500

Daily profit Total: 500

## 1.7. ממשקים למערכות אחרות

### 1.\*. Import javax.swing.\*

ספריית java swing בנויה על גבי ערכת הכלים של AWT, זאת ערכת כלים GUI ישנה יותר תלויה בפלטפורמה ברכיבי התכנות של Java GUI כמו כפתור תיבת טקסט וכו.

### 2.\*. import java.util.\*

ספריית Java Swing בנויה על גבי ערכת הכלים של Java Abstract חבילת java.util מכילה את מסגרת האוספים (arraylist), שיעורי איסוף מדור קודם, מודל אירוע, מתקני תאריך ושעה, ושיעורי שירות שונים.

S

### 3.\*. Import java.util.event.\*

חבילת java.awt.event מגדירה מחלקות וממשקים המשמשים לטיפול באירועים ב-AWT וב-Swing. חברי החבילה הזו מתחלקים לשלוש קטגוריות: Events.1.3

השיעורים עם שמות המסתיימים ב"אירוע" מייצגים סוגים ספציפיים של אירועים, שנוצרו על ידי AWT או על ידי אחד ממרכיבי AWT או Swing.

### 2.3. Listener

הממשקים בחבילה זו הם כולם מאזינים לאירועים שמותיהם מסתיימים ב"Listener", ממשקים אלה מגדירים את השיטות שיש ליישם על ידי כל אובייקט שרוצה לקבל הודעה כאשר מתרחש אירוע מסוים.

### :Adapters.3.3

כל אחד מהשיעורים עם השם המסתיים ב"Adapters" מספק הטמעה ללא אופציה לממשק מאזין לאירועים המגדיר יותר משיטה אחת.

### **:Import java.awt.\*.4**

Java AWT הוא API לפיתוח GUI או יישומים מבוססי חלונות בג'אווה, רכיבי Java AWT תלויים בפלטפורמה כלומר רכיבים מוצגים על פי תצוגת מערכת ההפעלה. AWT הוא כבד משקל, כלומר רכיביו משתמשים במשאבי מערכת ההפעלה. חבילת java.awt מספקת שיעורים ל- AWT API כגון TextField, Label, TextArea, RadioButton, CheckBox, Choice, List וכו'.

### **:Import java.sql.\*.5**

מספק ממשק API לגישה ולעיבוד נתונים המאוחסנים במקור נתונים (בדרך כלל בסיס נתונים יחסי) באמצעות שפת התכנות Java. הוא ממשק ה-API של JDBC מיועד בעיקר להעברת הצהרות SQL למסד נתונים, הוא מספק קריאה וכתובה של נתונים מכל מקור נתונים בפורמט טבלאי.

### **חבילות תוכנה**

#### **:JDBC\***

JDBC מייצג קישוריות מסדי נתונים של Java. JDBC הוא API ל- Java לחיבור וביצוע השאילתה עם מסד הנתונים. זהו חלק מ-Java. ה-API של JDBC משתמש במנהלי התקנים של JDBC כדי להתחבר למסד הנתונים. אנו יכולים להשתמש ב-JDBC API כדי לגשת לנתונים טבלאיים המאוחסנים בכל בסיס נתונים התייחסותי. בעזרת JDBC API אנו יכולים לשמור, לעדכן, למחוק ולהביא נתונים ממסד הנתונים.

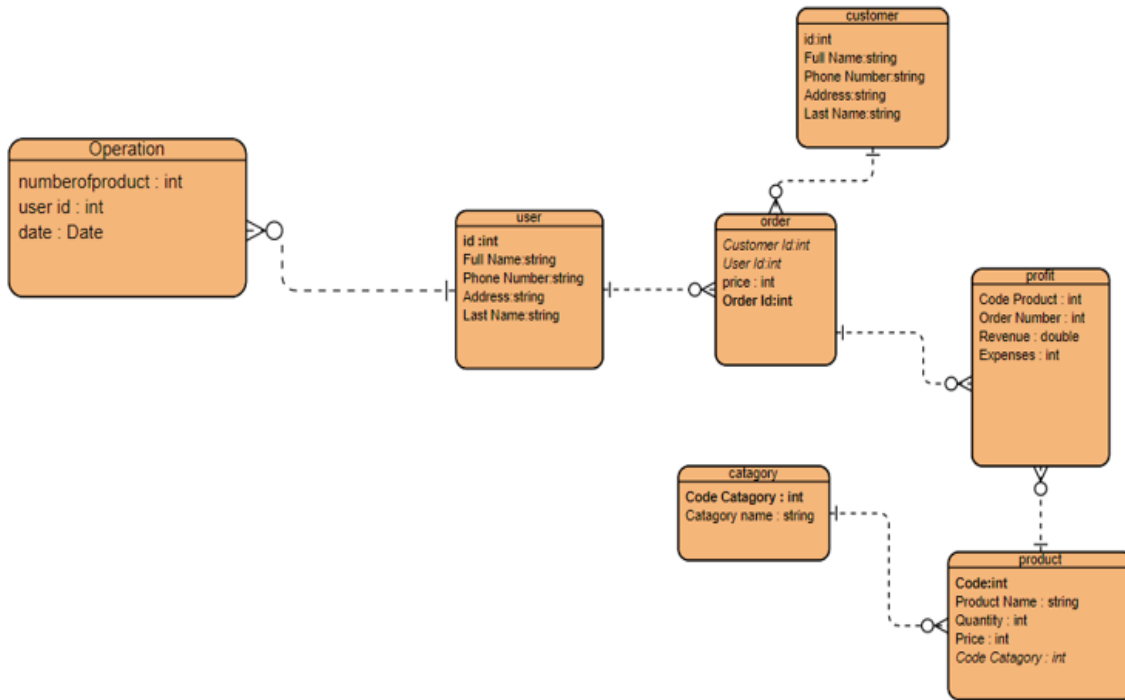
## **2. מבני נתונים וארגון קבצים**

### **2.1. מבני נתונים**

1. שימוש במערכים לנתונים הנאספים מבסיס הנתונים.
2. שימוש ב- array list להכנסת הנתונים לבסיס הנתונים.



## 2.2. שיטות אחסון

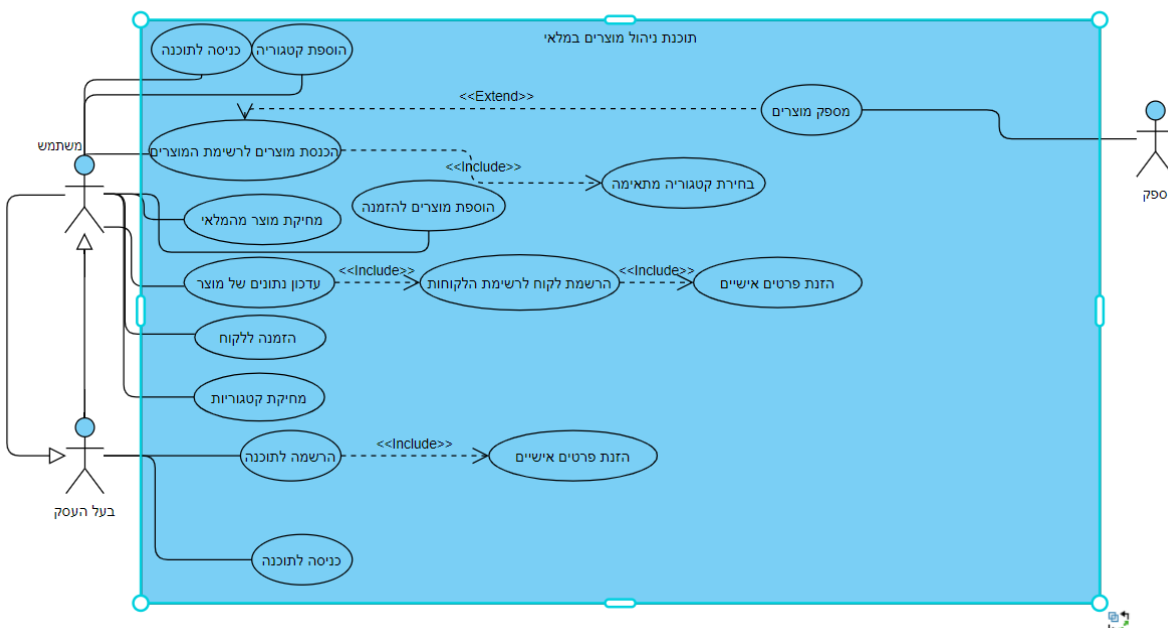


## 2.3. מנגנוני התאוששות

\*טיפול בחריגות (Exceptions Handling) הוא מנגנון המשלב בתוכנית מחשב קוד שייעודו להתריע על שגיאות המתרחשות בזמן ריצת התוכנית ולטפל בהן. מנגנון זה מתריע ומטפל בחריגה מפעולתה התקינה של התוכנית שהתרחשה בזמן ריצה.

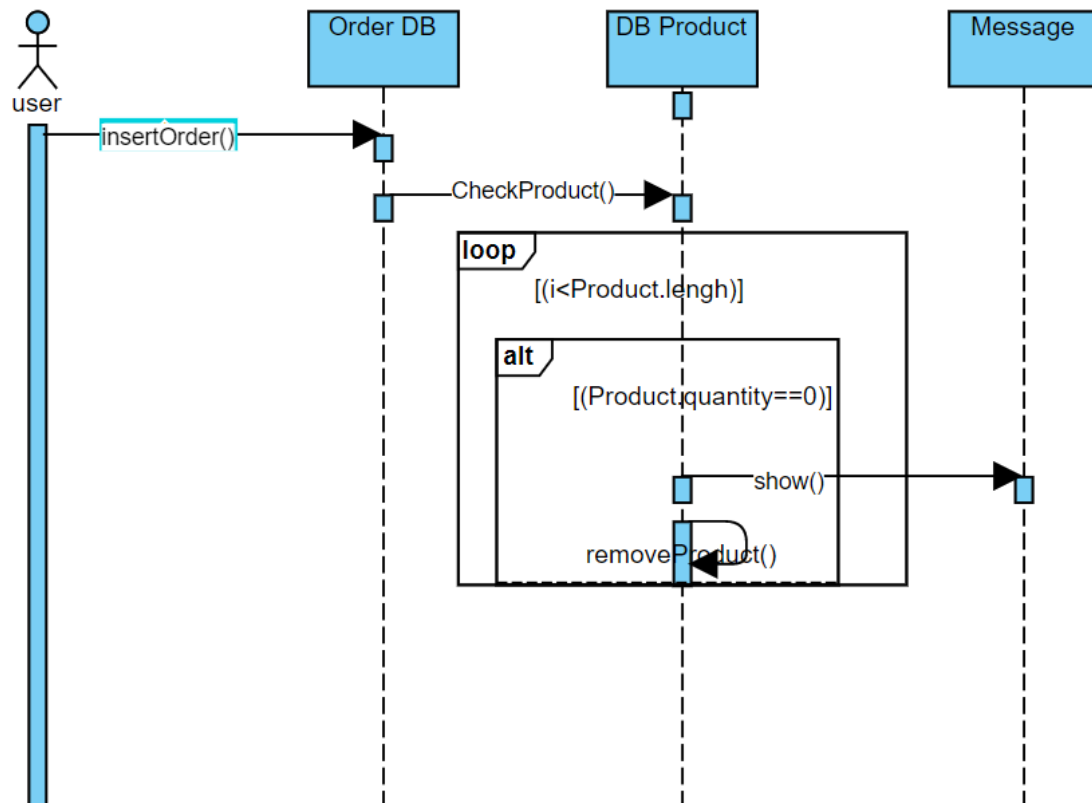
## 3. תרשימי מערכת

### Use Case .3.1

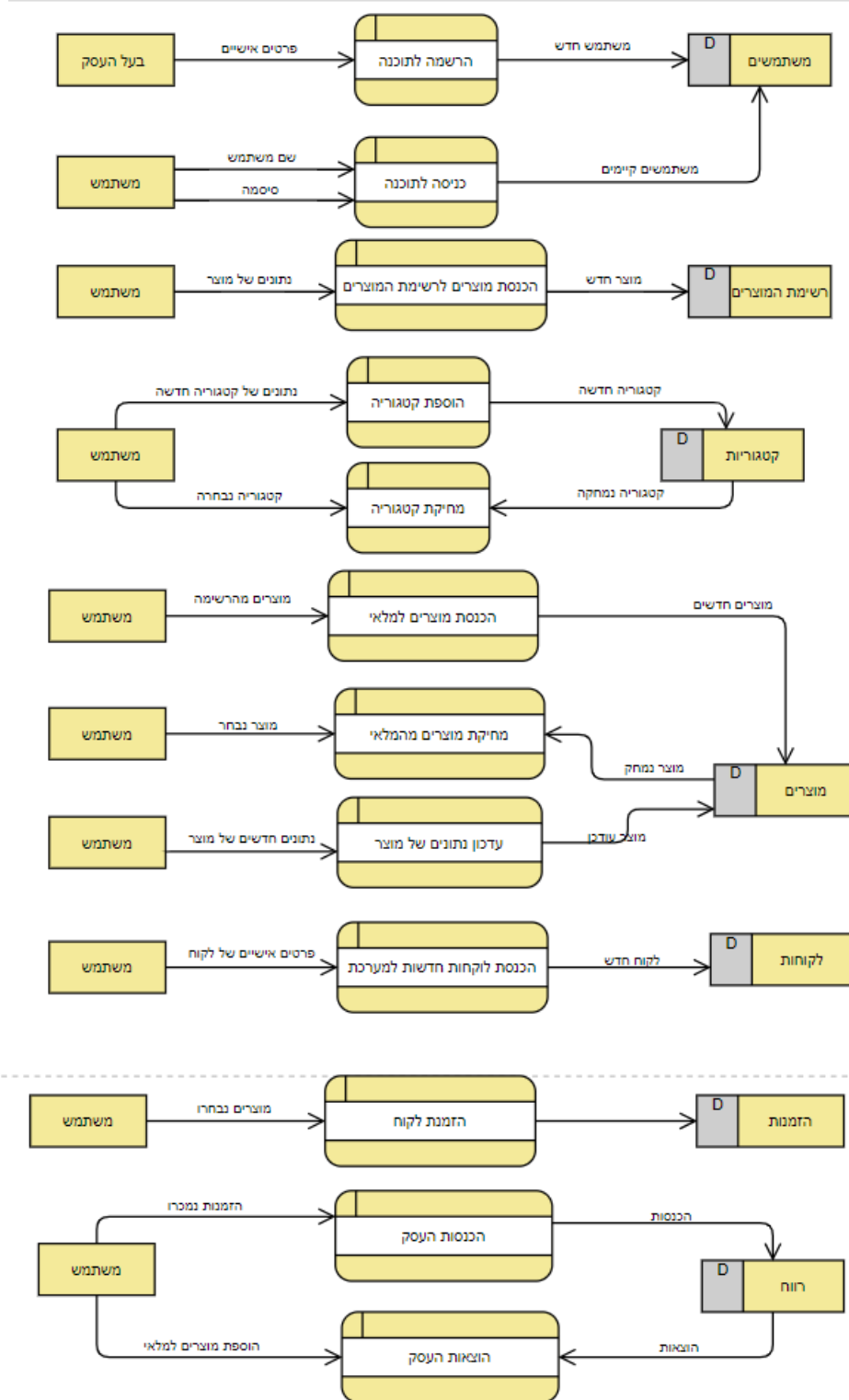


### Sequence.3.2

תרשים לתיאור רצף מסרים בין אובייקטים עבור פונקציות מרכזיות של המערכת.



### Data Flow .3.3



## 4. המרכיב האלגוריתמי

### 4.1. תיאור האלגוריתם

אלגוריתם כמות מוצר מסוים מגיעה לאפס:  
המערכת יודעת לדעת אם כמות מוצר מסוים מגיעה לאפס, כל פעם המשתמש מבצע הזמנה אז המערכת עוברת על כל המוצרים בטבלת המוצר בבסיס הנתונים ובודקת את כמות כל מוצר במלאי אם יש כמות של מוצר מגיעה לאפס אז מוציאה הודעה מתאימה למשתמש ומוחקת את המוצר מהמלאי.

### 4.2. איסוף מידע

מידע הנאסף של רווח היומי:

המידע הנאסף של הכנסות צריך לשמור את המחיר של הזמנה בכל פעם שהמשתמש בצע הזמנה וצריך לשמור את המידע הנאסף של הוצאות ולשמור את המחיר של המוצר בכל פעם שהמשתמש הוסיף מוצר למלאי ולכן הרווח היומי נעשה בעזרת הכנסות והוצאות.

## 5. אבטחת מידע

### 1. כניסה למערכת:

כדי לבצע פעולות במערכת צריך לבצע login.

### 2. קוד אימות:

קוד אימות לכניסה למערכת על מנת לבצע כניסה למערכת על המשתמש להזין קוד אימות נשלח למייל (מייל של המשתמש) כך מונעים מפריצות.

## 6. משאבים נדרשים

### 6.1. שעות וחלוקת עבודה

- כל חבר בצוות נדרש להקדיש לפחות 300 שעות.
- יוסף אחראי על העיצוב סאמי אחראי על DB ושנינו אחראים על כתיבת קוד.

### 6.2. ציוד נדרש

מחשב

### 6.3. תוכנות נדרשות

Eclipse java

### 6.4. ידע חדש

1. עיצוב מתקדם בג'אווה.
2. אלגוריתם ניהול מוצרים במלאי.
3. איך אפשר לפתח תוכנה.
4. עבודה בצוות.
5. קשר בין java ו Jason

### 6.5. ספרות ומקורות מידע

\*קישור הדרכה איך אפשר לפתח תוכנה:

<https://www.youtube.com/watch?v=VpwLw42vrwU&t=188s>

**\*נעזרנו בספר בשם:**

Head First Java

**\*קישור הדרכה עבודה על java ב designer:**

<https://www.youtube.com/watch?v=4SzG7wOkP98>

**7. תוכנית עבודה**

אחראי	
יוסף + סאמי	עבודה על הצעת פרויקט
יוסף	בניית ה database
יוסף	עיצוב לתוכנה (גרסה ראשונה)
סאמי	בניית ממשק משתמש
סאמי	בניית דף הרשמה למשתמש
סאמי + יוסף	בדיקות
יוסף	בניית דף ניהול מוצרים
סאמי	בניית רשימת הקטגוריות
יוסף + סאמי	בדיקות
סאמי	בניית דף ניהול לוקחות
יוסף	בניית דף הזמנות
סאמי	בניית דף הרווח
יוסף + סאמי	בדיקות

עיצוב לתוכנה (גרסה שנייה)	יוסף + סאמי
ספר פרויקט	יוסף + סאמי
תיקון הפרויקט אחרי ההגנה	יוסף + סאמי

**8. בדיקות**  
**8.1 Full Flow**

תהליך	בדיקה	פלט צפוי לבדיקה
רישום	הכנסת פרטי משתמש קיים.	הודעה שגויה
רישום	הכנסת פרטי משתמש חדש.	נוצר משתמש חדש בטבלת המשתמשים בבסיס הנתונים.
כניסה	הכנסת סיסמה שגויה או שם משתמש שגוי	הודעת שגיאה
כניסה	הכנסת שם משתמש קיים וסיסמה נכונה.	כניסה לתוכנה
הוספת מוצר	הכנסת מוצר קיים.	הודעת שגיאה(מוצר קיים).
הוספת מוצר	הזנת כמות או מחיר של מוצר שווה לאפס או פחות מאפס.	הודעת שגיאה.
הוספת מוצר	הכנסת מוצר חדש.	הוספת מוצר חדש בטבלת מוצר בבסיס הנתונים.
מחיקת מוצר	מחיקת מוצר לא קיים.	הודעה(המוצר לא נמצא במלאי).
מחיקת מוצר	מחיקת מוצר קיים.	מחיקת מוצר מטבלת מוצר בבסיס הנתונים
עדכון מוצר	הזנת כמות או מחיר של מוצר שווה לאפס או פחות מאפס.	הודעת שגיאה.
הכנסת קטגוריה	הכנסת קטגוריה קיימת.	הודעת שגיאה.
הכנסת קטגוריה	הכנסת קטגוריה לא קיימת.	הוספת קטגוריה לטבלת הקטגוריות בבסיס הנתונים.

הכנסת לקוח	הכנסת לקוח קיים.	הודעת שגיאה.
הכנסת לקוח	הכנסת לקוח לא קיים.	הוספת לקוח לטבלת הלוקחות בבסיס הנתונים.
הזמנה	הוספת הזמנה ריקה.	הודעת שגיאה.
הזמנה	הוספת הזמנה מכילה מוצרים.	הוספת הזמנה לטבלת ההזמנות בבסיס הנתונים.

## 8.2 Unit Test

רכיב	בדיקה	פלט צפוי
משתמש	-קליטת נתונים תקין -בדיקת נתונים של המשתמש באמצעות בסיס הנתונים(אם המשתמש קיים).	-הוצאת הודעה. -הכנסת נתונים של המשתמש לטבלת המשתמש בבסיס הנתונים.
מוצר	-קליטת נתונים תקין. - בדיקה אם המוצר קיים במלאי.	-הוצאת הודעה. -הכנסת נתונים של מוצר לטבלת המוצר בבסיס הנתונים.
קטגוריה	-קליטת נתונים תקין. -בדיקה אם הקטגוריה קיימת.	-הוצאת הודעה. -הכנסת נתונים של קטגוריה לטבלת המוצר בבסיס הנתונים.

## 9. בקרת גרסאות

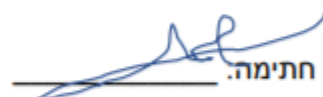
גרסה	תיאור
גרסה ראשונה	הרשמה וכניסה לתוכנה
גרסה שנייה	הכנסה עדכון ומחיקת מוצר ובחירת קטגוריה
גרסה שלישית	הרשמת לקוחות והזמנות לקוח
גרסה רביעית	רווח יומי(הכנסות והוצאות)
גרסה חמישית	אפשר לשנות את השפה של התוכנה.

\_\_\_\_\_  
חתימת המנחה

  
\_\_\_\_\_  
חתימת הסטודנט

ג. הערות ראש המגמה במכללה  
על פי חוות דעת הוועדה המכללית. ההצעה מאושרת. יש להתייחס לכל ההערות.  
וועדת הוראה: יו"ר ד"ר שאדי עסאקלה רמ"ח, חבר קופפר מורן – מנחה ומרצה + חאמד אוסנת  
– מנחה ומרצה.

**ד. אישור ראש המגמה**

  
שם: \_\_\_\_\_ ד"ר שאדי עסאקלה חתימה: \_\_\_\_\_  
תאריך: 24.05.2021 \_\_\_\_\_

**ג. הערות הגורם המקצועי מטעם מה"ט**

\_\_\_\_\_  
\_\_\_\_\_

**ד. אישור הגורם המקצועי מטעם מה"ט**

שם: \_\_\_\_\_ חתימה: \_\_\_\_\_ תאריך: \_\_\_\_\_



## תוכן עניינים

<b>28</b>	<b>תיאור הפרויקט</b>
28	שם הפרויקט
28	תיאור ורקע כללי
28	מטרת הפרויקט
28	ציוד חומרה ותוכנה הנדרש לביצוע הפרויקט
28	מטרת העבודה, הצגת הבעיה וניתוח מערכת
28	הגדרת המוצר הסופי
29	סקירה כללית של המצב הקיים בשטח
29	הפרויקט מחדש/משפר
29	פתרונות אפשריים
29	פתרון נבחר, נימוקים לבחירותינו
30	אלגוריתמיקה
30	מבני נתונים וארגון קבצים
30	אמצעי אחסון
30	חלוקת תפקידים
31	מה יקנה הפרויקט לסטודנט
31	שלבים בפיתוח הפרויקט
<b>33</b>	<b>מבוא</b>
33	הקדמה
34	סביבת הפיתוח
35	Java language שפת ג'אווה
36	MYSQL
37	Scene builder מהו?
37	האם כדאי JavaFx ללמוד?
38	Eclipse
<b>39</b>	<b>מסכי התוכנה</b>
39	עמוד ההרשמה למערכת Sign up
40	עמוד הכניסה למערכת Login
41	עמוד המלאי Inventory
42	עמוד הקטגוריות Categories
43	עמוד הלקוח Customer
44	סטטיסטיקות של העסק Statistics page
45	רשימת ההזמנות orders list
46	עמוד הזמנה order
47	עמוד המדפים shelves page
48	עמוד ההוצאות Expenses
<b>49</b>	<b>תרשים הסופי של המחלקות</b>
49	הסבר על כל מחלקה
<b>51</b>	<b>הסבר על כל הפונקציות</b>
51	הסבר הפונקציות במחלקת Conditions
52	הסבר הפונקציות במחלקת Field
53	הסבר הפונקציות במחלקת DatabaseManager
56	הסבר הפונקציות במחלקת AnimationClass
56	הסבר הפונקציות במחלקת RowTable
57	הסבר הפונקציות במחלקת SwitchingPages
57	הסבר הפונקציות במחלקת Message
57	הסבר הפונקציות במחלקת DateClass

58	.....	LogoutClass	הסבר הפונקציות במחלקת
58	.....	Category	הסבר הפונקציות במחלקת
58	.....	Customer	הסבר הפונקציות במחלקת
58	.....	Expenses	הסבר הפונקציות במחלקת
58	.....	Manager	הסבר הפונקציות במחלקת
58	.....	Order	הסבר הפונקציות במחלקת
59	.....	Product	הסבר הפונקציות במחלקת
60	.....	Shelf	הסבר הפונקציות במחלקת
60	.....	User	הסבר הפונקציות במחלקת
<b>61</b>	.....	<b>FXML</b>	<b>קוד קבצי</b>
62	.....	Login.fxml	קובץ
65	.....	SignUp.fxml	קובץ
69	.....	inventory.fxml	קובץ
73	.....	addProduct.fxml	קובץ
75	.....	EditProductPage.fxml	קובץ
77	.....	CategoryPage.fxml	קובץ
81	.....	addCategoryPage.fxml	קובץ
82	.....	EditCategoryPage.fxml	קובץ
83	.....	CustomerPage.fxml	קובץ
88	.....	CustomerList.fxml	קובץ
90	.....	EditCustomerPage.fxml	קובץ
92	.....	OrderPage.fxml	קובץ
97	.....	Shopping_cart.fxml	קובץ
100	.....	OrderListPage.fxml	קובץ
102	.....	viewOrder.fxml	קובץ
105	.....	ShelvesPage.fxml	קובץ
109	.....	ShelvesPage.fxml	קובץ
112	.....	statistics.fxml	קובץ
123	.....	productsInCategories.fxml	קובץ
125	.....	Best-selling-page.fxml	קובץ
127	.....	Best-selling-page-top-3.fxml	קובץ
129	.....	Best-selling-page-top-3.fxml	קובץ
131	.....	OrderInMonth.fxml	קובץ
133	.....	ProductsInTheCategory.fxml	קובץ
135	.....	Expenses.fxml	קובץ
<b>140</b>	.....	<b>Java</b>	<b>קוד קבצי</b>
141	.....	SignupController.java	קובץ
144	.....	Controller.java	קובץ
146	.....	InventoryController.java	קובץ
149	.....	AddProductController.java	קובץ
151	.....	EditProductController.java	קובץ
153	.....	CategoryController.java	קובץ
156	.....	AddCategoryPageController.java	קובץ
157	.....	EditCategoryPageController.java	קובץ
158	.....	CustomerController.java	קובץ
160	.....	CustomerListController.java	קובץ
162	.....	orderController.java	קובץ
166	.....	ShoppingCartController.java	קובץ
171	.....	viewOrderController.java	קובץ
176	.....	OrderListController.java	קובץ
178	.....	ShelvesController.java	קובץ
181	.....	AddShelfController.java	קובץ
182	.....	EditShelfController.java	קובץ

183	.....ViewShelfController.java	קובץ
186	.....AddProductToShelf.java	קובץ
188	.....statisticsController.java	קובץ
191	.....bestSellingProductController.java	קובץ
193	.....bestSellingTop3Controller.java	קובץ
195	.....productsInCategoriesController.java	קובץ
196	.....OrderInMonthController.java	קובץ
198	.....OrderNumberController.java	קובץ
200	.....ProductInTheCategoryController.java	קובץ
202	.....ExpensesController.java	קובץ
206	.....DatabaseManager.java	קובץ
215	.....Message.java	קובץ
216	.....Field.java	קובץ
219	.....emptyFiels.java	קובץ
220	.....DateClass.java	קובץ
221	.....Person.java	קובץ
222	.....User.java	קובץ
223	.....Manager.java	קובץ
224	.....Customer.java	קובץ
226	.....Category.java	קובץ
228	.....Product.java	קובץ
232	.....Shelf.java	קובץ
234	.....Order.java	קובץ
236	.....Water.java	קובץ
236	.....Electrical.java	קובץ
237	.....Conditions.java	קובץ
242	.....RowTable.java	קובץ
244	.....SwitchingPages.java	קובץ
245	.....LogoutClass.java	קובץ
247	.....PdfFile.java	קובץ
<b>250</b>	.....ביביליוגרפיה	

## **תיאור הפרויקט**

### **שם הפרויקט**

מערכת ניהול מוצרים.

### **תיאור ורקע כללי**

יש במערכת טופס הרשמה למשתמש שהוא רושם את הפרטים שלו והם נשמרים בבסיס הנתונים, אחרי ההרשמה הוא יכול לנכנס למערכת עם שם משתמש וסיסמה. המערכת מאפשרת למשתמש להוסיף לעדכן ולמחוק קטגוריות וגם הוספת מוצרים למלאי ועל המשתמש לבחור את הקטגוריה המתאימה לכל מוצר והתוכנה. התוכנה יכולה לעבור על טבלת המוצרים ולשים קו אזהרה על כל מוצר שיש בו כמות מתחת לעשר ואם יש מוצרים שהכמות שלהם שווה לאפס מדליקה את השורה בטבלה באדום. בתוכנה יש גם טופס להוספת לקוחות שרוצים להזמין מוצרים לבסיס הנתונים (טבלת הלקוחות). המערכת גם יודעת לחלק את המוצרים שבתוך המלאי למדפים לפי המשקל למשל יש משקל מקסימלי שמשקל כמות המוצרים על המדף לא יכול לעבור הגבול המקסימלי וכל פעם מוסיפים מוצר למדף כמות המוצרים במדף יורדת בהתאם. המערכת גם יודעת לחשב משכורת יומית של העובד וגם יודעת לחשב הוצאות חשמל ומים של העסק בחודש הנוכחי. המערכת יודעת לחשב את הרווח, הוצאות והכנסות של העסק המוצרים הכי נמכרים, שלושה המוצרים הנמכרים ביותר, כמה מוצרים יש בעסק, מספר לקוחות ומספר מדפים וכמות הזמנות, מספר המוצרים בכל קטגוריה, מספר ההזמנות של הלקוחות, ומציגה את כל ההזמנות שבוצעו בחודש הנוכחי.

### **מטרת הפרויקט**

- מטרת הפרויקט – לבנות מערכת קלה לניהול מוצרים בעסק.
- הפרויקט יכלול בסיס נתונים לפרטי המשתמשים ונתוני כל המוצרים שנמצאים במלאי ופרטי הלקוחות עם ההזמנות שלהם.
- הפרויקט מציג סטטיסטיקות לעסק ויודעת לעשות את זה באופן אוטומטי.
- הפרויקט אמור לשלב את הטכנולוגיות אשר למדנו במהלך השלוש שנים: אלגוריתמיקה, ניהול בסיס נתונים, ושליטה בשפות תכנות שונות.
- למידת דברים חדשים תוך כדי ביצוע העבודה והתמודדות מול הבעיות באופן עצמאי.

### **ציוד חומרה ותוכנה הנדרש לביצוע הפרויקט**

Eclipse, MySQL, מחשב.

### **מטרת העבודה, הצגת הבעיה וניתוח מערכת**

ברצוננו ללמוד לפתח חשיבה אלגוריתמית, למצוא פתרונות אוריינלים לבעיות ולכתוב קוד בצורה הכי יעילה שיש.

### **הגדרת המוצר הסופי**

מערכת ניהול מוצרים (חלוקת מוצרים על המדפים וניהול מלאי וניהול הזמנות).

## סקירה כללית של המצב הקיים בשטח

דוגמא לשתי תוכנות קיימות בשוק :

### **ABC Inventory**

תוכנה בנויה באופן פשוט וקל להבנה נועדה לעסקים קטנים.  
התוכנה כוללת מאפיינים לוגיסטיים החשובים לכל עסק ועוזרת לנהל שינויים במלאי  
ובמחסנים פרטי משלוח וניהול חשבונות.

### **Delivrd**

תוכנה מבוססת על ניהול מלאי ומכירות .  
התוכנה עוזרת לנהל את המחסן וגם את המכירות רווח הפסד והוצאות והיא מיועדת  
לעסקים קטנים עם מספר מוצרים מוגבל.

## הפרויקט מחדש/משפר

בנוסף לניהול המלאי המערכת תעזור בסידור המוצרים על המדפים בתוך העסק .  
המערכת יודעת לזהות את סוגי המוצרים ולחלק אותם לפי משקל ולכל מדף יש משקל מקסימלי  
ומאפשרת גם חלוקה נכונה של המוצרים.

## פתרונות אפשריים

- חיפוש מידע על כל מה שצריך באינטרנט.
- מחלקים את הבעיה לתת משימות כך שיהיה קל למצוא את הפתרון.
- קורסים ללימוד פיתוח תוכנות ב JavaFX באינטרנט.
- תמיכת המנחה שלנו.

## פתרון נבחר, נימוקים לבחירותינו

חיפוש על מידע דרך האינטרנט יש באינטרנט כיום מאגר מידע חינוכי שניגש בקלות ויש קורסים  
ללימוד שפות שונות בחינם.

## אלגוריתמיקה

### **אלגוריתם המוצרים הכי נמכרים בחודש הנוכחי:**

באלגוריתם הזה המערכת יודעת לחשב את כל המוצרים הנמכרים בחודש הנוכחי ולסדר אותם בתוך טבלה מהכי נמכר להכי פחות.

כדי לחשב את המוצרים הכי נמכרים בהתחלה אנחנו עוברים על כל ההזמנות ובודקים תאריך של ההזמנה אם היא בוצעה בחודש הנוכחי ובודקים אם היא בוצעה בשנה הנוכחית.

Order ID	Customer id	Order date
1	207418874	12/8/2021
2	318741444	12/9/2020
3	231222213	10/9/2121

במידה ויש את ההזמנות המופיעות בתוך הטבלה אז המערכת מתייחסת רק להזמנה מספר 3 כי היא בוצעה החודש הנוכחי, אחר כך המערכת ניגשת למוצרים בתוך ההזמנה ומחשבת כל מוצר מופיע בהזמנה כמה פעמים המוצר נמכר בכל ההזמנות שנמצאים בחודש והשנה הנוכחית (בעצם לוקחים את כמות המוצרים מאותו סוג) וכל פעם המערכת ניגשת להזמנה אחרת בודקת אם כבר בדקה את המוצר כי לא צריך לעבור עליו עוד פעם ובסוף המערכת עושה מיון לכל המוצרים מהכי נמכר להכי פחות ומסדרת אותם בתוך טבלה.

### **מבני נתונים וארגון קבצים**

ארגון קבצים ומאגר המידע של המערכת יש כמה טבלאות במסד הנתונים שנמצא בא נתונים שקשורים לטבלאות, יש 10 טבלאות במסד הנתונים:

1. Shopping Cart - סל הקניות של הלקוח.
2. Customer – טבלת הלקוח נשמר בא את כל הפרטים על הלקוח.
3. User – טבלת המשתמשים כל המשתמשים שרוצים להשתמש בתוכנה צריך לבצע רישום וכל המשתמשים נשמרים תוך הטבלה.
4. Manager – טבלת המנהל נשמרים כל הפרטים של המנהל.
5. Product – כל המוצרים המוכנסים למלאי נשמרים בתוך הטבלה Product.
6. Shelf – כל הנתונים של המדפים נשמרים בתוך Shelf.
7. Orders – כל הנתונים של הזמנות נשמרים תוך הטבלה orders.
8. order\_product – בטבלה הזו נשמרים כל המוצרים השייכים להזמנות.
9. shelf\_product – בטבלה הזו נשמרים כל המוצרים השייכים למדפים.
10. Category – בטבלה הזו נשמרים כל הקטגוריות.
11. Expenses – בטבלה הזו נשמרים כל ההוצאות של המים וחשמל בעסק.

בפרויקט שלנו השתמשנו בשלושה שאילתות select, insert, update.

### **אמצעי אחסון**

זיכרון נייד, בסיס נתונים (MySQL).

### **חלוקת תפקידים**

עבודה משותפת ביחד עם חשיבה על כל הבעיות ביחד כך שנהיה מעודכנים בכל שינוי.

## מה יקנה הפרויקט לסטודנט

- פיתוח חשיבה אלגוריתמית.
- להתחזק בשפת java.
- עבודה מול בסיס נתונים.
- עבודה בצוות.
- רכישת ניסיון בפיתוח תוכנה.

## שלבים בפיתוח הפרויקט

שלבי פיתוח הפרויקט :

1. בהתחלה בנינו את ההצעה .
2. אחרי בניית ההצעה התחלנו ללמוד כיצד לעבוד ב JavaFx . ( איך אפשר להשתמש ב controller , איך אפשר לשלוח אובייקטים בין המחלקות...).
3. בניית בסיס נתונים ראשוני כדי שנוכל לעבוד על מערכת לטווח ארוך.
4. בניית תוכנה ראשונית שדורשת מהמשתמש לבצע הרשמה לתוכנה ואם עבר בהצלחה יפתח את החלון הראשון בתוכנה.
5. אחר כך התחלנו ללמוד כיצד אפשר לעבוד מול הטבלה ב JavaFx ותוך כדי הלמידה בנינו את הדף של המלאי והקטגוריה שמאפשרת למשתמש להכניס או לערוך או למחוק מוצר מהמלאי וכל מוצר שייך לקטגוריה מסוימת.
6. כשהגענו לשלב הזה התחלנו לשלוט ולהרגיש את ה JavaFx אחר כך בנינו הדף של ניהול לקוחות.
7. בניית דף של המדפים וכיצד אפשר לחלק את המוצרים על המדפים לפי משקל בצורה יעילה.
8. אחר כך מאפשרת למשתמשים להוסיף מוצרים לסל הקניות ולבצע הזמנה , ולכל ההזמנה יופיע סך כל במחיר של ההזמנה.
9. אחר כך עושים אפשר למשתמש לערוך את ההזמנה ולקחת (אפשר למחוק את המוצר מההזמנה אז מחזירים את המוצר למדף באופן אוטומטי , אפשר לערוך כמות המוצר בהזמנה) במקביל צריך לערוך את כמות המוצר במדף לפי כמות המוצר הנמצאת במדף).
10. בניית דף סטטיסטיקות של העסק ( הכנסות , הוצאות ורווח , מספר המוצרים הכי נמכרים בחודש הנוכחי , מספר ההזמנות של כל הלקוחות , הצגת ההזמנות בחודש הנוכחי , מספר המוצרים בקטגוריה , מספר המדפים , סך כל מוצרים במלאי , סך כל מוצרים במדפים , מספר מוצרים במלאי , מספר מוצרים במדפים).
11. בניית דף הוצאות כללי בדף הזה אפשר לשלם לכל עובד בכל יום וכל יום מתווסף למשכורת הכללית של עובד ומוסיפה את ההוצאות .
12. הוספת צבעים לשורות בטבלה (תלוי בנתונים בטבלה) למשל : לטבלת המוצרים בודק אם כמות המוצר שווה לאפס אם כן משנה צבע השורה לאדום אם לא בודק אם כמות המוצרים פחות מ 10 משנה צבע של השורה לכתום.

13. בשלב האחרון בכתיבת קוד הוספנו אפשרות למנהל לחלק משכרות יומית לעובדים ולהקליד הוצאות מים וחשמל של העסק בחודש

13. בשלב האחרון של הפרויקט כתיבת ספר הפרויקט.



**הקדמה**

ספר זה נכתב על ידי סאמי חיחי ויוסף חיר, סטודנטים של מכללת אורט בראודה מחלקת הנדסת תוכנה מסלול הנדסאים (משולב). כמו כל סטודנט בתום סיום הלימודים עושים פרויקט מעשי שמשקף מה שנלמד במהלך הלימודים. בהתחלה היה קשה לבחור את נושא הפרויקט או באיזה שפה נעשה אותו. אחרי שעלה לנו בראש רעיון בניית תוכנת ניהול מוצרים בעסק שאחראית על ניהול העסק העבודה התחילה לזרום לאט לאט.

בכל שלב של בנייה היינו נתקלים בבעיות שונות לפעמים לקח לנו זמן לפתור אותם ולפעמים פתרנו אותן ישר דרך חשיבה משותפת. המכללה תרמה לנו הרבה מידע שהועבר בדרך הכי טובה, עם מרצים מקצוענים ומנוסים בשפות תכנות שונות. במהלך כתיבת הספר אנחנו מנסים להסביר את בסיס בניית המערכת ואיך משתמשים בה. ולא לשכוח להגיד תודה לכל מי שעזר לנו בדרך להקים את הפרויקט בדרך הכי טובה ויעילה שיש.

## סביבת הפיתוח

השתמשנו בשלוש סביבות לפיתוח הפרויקט:

**Eclipse** : השתמשנו ב Eclipse כי היא תוכנה נוחה וקלה לשימוש למשל היא מציגה לך איפה הטעויות באיזה שורה ומה הטעות.  
גם השתמשנו בה כדי לנהל את המערכת ולקשר בין המערכת למסד הנתונים.

**MySQL** : השתמשנו ב MySQL כדי לשמור את נתוני המשתמשים, מוצרים, קטגוריות, לקוחות, הזמנות וגם שנוכל לעדכן נתונים ואפשר להשתמש בכמה שאלות להכנסת ועדכון ומחיקת נתונים ממערכת.

**Scene builder** : השתמשנו ב scene builder כדי לסדר את העמודים שלנו בתוך המערכת וגם היא עוזרת לנו לכתוב קוד בצורה יותר מסודרת למשל עם העתקת פונקציה לכל כפתור.

אחרי שדיברנו על סביבות הפיתוח שהשתמשנו בהם נצטרך עכשיו לדון בכל אחת בנפרד ולהוסיף קצת מידע על כל צפה בנפרד.

## Java language שפת ג'אווה

ג'אווה באנגלית Java: היא שפת תכנות מונחית עצמים אשר פותחה בחברת סאן מיקרוסיסטמס) כיום חברת-בת של אורקל (על ידי צוות בראשות ג'יימס גוסלינג בשנת 1991 והיא אחת משפות התכנות הנפוצות ביותר הנמצאות בשימוש כיום השפה הוצגה לראשונה בשנת 1995, והיא מהווה את אחד מרכיבי הליבה של פלטפורמת התוכנה ג'אווה. התחביר של השפה מבוסס במידה רבה על התחביר של ++C, אך כולל הרחבות רבות במטרה לאפשר תמיכה מובנית בתהליכונים, בינלאומיות, אבטחה ועבודה בסביבת האינטרנט ותכונות נוספות. לרוב עוברות תכניות ג'אווה הידור ל- Java bytecode, שפת ביניים דמוית שפת מכונה, שאותה מריצה מכונה וירטואלית (Java Virtual Machine JVM). הודות לכך התוכנית יכולה לרוץ על כל מחשב ועל כל מערכת הפעלה המריצים JVM, החל מטלפונים סלולריים ועד למחשבי-על.

### **עקרונות:**

ג'אווה היא שפה בעלת טיפוסיות סטטית חזקה, כלומר לכל ביטוי בשפה מתאים טיפוס יחיד, תקינות ביטויים נבדקת בזמן ההידור, במידת האפשר. כאשר אין זה אפשרי, תתבצע בדיקה בזמן ריצה.

ג'אווה תומכת בתכנות מונחה עצמים. עם זאת ג'אווה עדיין כוללת, מטעמי יעילות, גם "טיפוסים פרימיטיביים", שאינם אובייקטים. משום כך (ומסיבות נוספות) אינה נחשבת שפה מונחית עצמים "טהורה" כמו Smalltalk או Ruby עם זאת, לכל טיפוס פרימיטיבי בג'אווה יש מחלקה עוטפת, אליה וממנה מתבצעת המרה אוטומטית (Boxing). למשל, לטיפוס הפרימיטיבי int קיימת מחלקה עוטפת Integer.

שפת ג'אווה כוללת גם ניהול זיכרון אוטומטי המתכנת פטור מן ההכרח לשחרר זיכרון המוקצה לאובייקט ברגע שאין עוד משתנים המצביעים עליו. במקום זאת, סביבת זמן הריצה כוללת מנגנון "איסוף זבל" המבצע זאת אוטומטית.

ג'אווה מאפשרת ירושה יחידה בלבד, וזאת על מנת למנוע בעיות דו-משמעות הנוצרות מירושת יהלום. כדי לאפשר גמישות דומה לזו של ירושה מרובה, קיים בג'אווה מנגנון ממשק (interface) המגדיר רשימה של מתודות, המהווה חוזה עם המתכנת. כך כל מחלקה יכולה להרחיב (extends) לכל היותר מחלקה אחת, ולממש (implements) מספר בלתי מוגבל של ממשקים. על המחלקה לממש במפורש כל מתודה בכל ממשק כזה, וכך לא קיימת בעיה של כפל משמעות.

### **ביצועים:**

בראשית דרכה סבלה ג'אווה מביצועים גרועים, היות שמנועי זמן הריצה בוססו על מפרש בלבד. אולם, עם התפתחות השפה, הוחל שימוש בטכניקה הקרויה Just In Time, JIT - אשר בבסיסה עמד העיקרון של הידור דינמי של קוד המכונה הווירטואלי לקוד המכונה הספציפי בזמן הריצה. על ידי כך מתקבלים ביצועים טובים יותר, ותאורטית אף גבוהים יותר משפת ++C במצבים מסוימים (היות שההידור מתבצע תוך כדי ריצת התוכנה, יש למהדר ידע מדויק יותר על ריצת התוכנה, מה שמסייע לביצוע אופטימיזציות טובות יותר לקוד).

גישה אחרת לנושא הביצועים מפרידה את עיקרון המכונה הווירטואלית מהשפה, ומהדרת את קוד המקור ישירות לקוד מכונה ספציפי, שרץ ככל תוכנה אחרת במערכת. עם זאת, גם בהידור כזה ישנו חיסרון לעומת שפות כמו ++C שעוצבו במיוחד לאפשר ריצה מהירה. לדוגמה, כל קריאה לשיטה בג'אווה כוללת חיפוש בזמן ריצה של השיטה המתאימה, בניגוד לשפת ++C בה קריאה לשיטה שאינה מוגדרת כווירטואלית שקולה לקריאה לפונקציה רגילה. בנוסף לכך ג'אווה נודעת בצריכת זיכרון גבוהה בהשוואה לקוד דומה שנכתב בשפה כגון שפת ++C, זאת עקב טעינה של ספריות ניהול של המכונה הווירטואלית. צריכת זיכרון גבוהה זו באה לידי ביטוי בעיקר במערכות הפעלה מסוג UNIX.

## MYSQL

MySQL הוא מסד נתונים יחסי, רב נימי ורב משתמשים מבוסס שפת SQL. (Structured Query Language) נקרא ע"ש My בתו של מונטי וידניוס, אחד מהכותבים המקוריים. התוכנה פותחה במקור על ידי החברה השוודית MySQL AB כיום היא בבעלות חברת אורקל. התוכנה היא חלק מ LAMP-אוסף תוכנות תשתית פופולריות שעומדות בבסיסם של אתרים רבים, כגון ויקיפדיה. מערכות ניהול תוכן רבות כגון וורדפרס ודרופל (משתמשות בה כבסיס נתונים).

### **שימושים:**

MySQL נפוצה ביישומי רשת כמו ויקיפדיה ויישומים רבים נוספים הדורשים בסיס נתונים MySQL. נפוצה באופן דומה ל PHP-אשר לרוב משולבות יחד ואף זכו לכינוי *Dynamic Duo* הזוג הדינמי. קל למצוא הקשרים רבים לשילוב זה החל בספרים ועד לאתרים ברשת MySQL. הוא בסיס נתונים קל ללימוד ולשימוש באופן יחסי לבסיסי נתונים אחרים.

### **פלטפורמות:**

MySQL יכולה לפעול על מספר רב של פלטפורמות, AIX, BSDi, FreeBSD, HP-UX, לינוקס Mac, OS X, NetBSD, Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows vista, Windows 7, OpenVMS גם גרסה למערכת ברשת.

## מהו? Scene builder

בונה סצינות הוא כלי פריסה חזותי המאפשר למשתמשים לעצב במהירות ממשקי משתמש של יישומי JavaFX, ללא קידוד. התוצאה היא קובץ FXML שאותו ניתן לשלב עם Java פרויקט על ידי קישור ממשק המשתמש ללוגיקה של היישום.

## האם כדאי JavaFx ללמוד?

כן, אם אתה רוצה לעשות דברים אשר JavaFX הוא טוב מאוד ב JavaFX. אינה רק אלטרנטיבה לאתר אינטרנט, אלא אלטרנטיבה לדרכים אחרות לבניית יישומי שולחן עבודה / מובייל של GUI ביסודו של דבר, מה שאתה מקבל הוא ערכת כלים של ממשק משתמש אשר ניתן להשוואה ל WPF / XAML -של מיקרוסופט.

JavaFX כלול באורקל JDK 8 פועלות על OS X, Linux x86 ו Windows. -המקום הטוב ביותר ללמוד עוד על כך הוא התייעוד של (x. Oracle Client Technologies JavaFX) הופץ כערכת הורדה ופיתוח נפרדת והטמיע שפה חדשה בשם JavaFXסקריפט שהיה שונה מ Java.

## Eclipse

Eclipse היא פלטפורמת תוכנה פתוחה וחופשית לפיתוח יישומי צד לקוח עשירים Eclipse. מנוהלת על ידי Eclipse Foundation, מוסד ללא כוונת רווח המתווה את כיווני ההתפתחות של הפלטפורמה. כלי פיתוח רבים מפותחים כיום על-גבי Eclipse וחלקם מנוהלים כתתי פרויקטים במסגרת Eclipse Foundation. על הפלטפורמה מפותחים מוצרים מסחריים רבים וכן פרויקטים פתוחים וחופשיים שונים לדוגמה Vuse.

עם זאת, השימוש הרווח ביותר ב־Eclipse הוא ככלי פיתוח ליישומים בשפת Eclipse Java. מספקת סביבת פיתוח מתקדמת לשפה זו, והיא בעצמה משמשת לפיתוח פלטפורמת Eclipse, הכתובה גם היא ב־Java. בנוסף על כך, ל־Eclipse יש מספר רב של פרויקטים משניים, המספקים סביבות פיתוח לשפות אחרות כגון C++, פתרון ורבות אחרות. ההשערה הרווחת היא שהשם "Eclipse" נבחר בגלל התחרות עם חברת סאן מיקרוסיסטמס, מפתחת שפת "Java. סאן" פירושו שמש ו־"Eclipse" פירושו ליקוי חמה.

## מסכי התוכנה

**עמוד ההרשמה למערכת Sign up**

Management products - Sign up

Sign up

Login

## Sign up

First Name

Last Name

Email

I.D

User name

Password

☐ I agree to the tos

Sign up

בעמוד הזה המשתמש מזין את הפרטים שלו בשדות ואז לוחץ על כפתור ההרשמה .  
**בדיקות קלט אחרי לחיצה על כפתור sign up:**

- המערכת בודקת אם המשתמש מילא את כל השדות .
- בדיקת שם ושם משפחה אם הם תקינים.
- בדיקה אם המייל תקין לפי הדרישות.
- בדיקת ת.ז אם מכילה תשע ספרות.
- בדיקת שם משתמש וסיסמה אם תקינים .
- המערכת בודקת אם שם המשתמש קיים או לא.

## עמוד הכניסה למערכת Login

Management products - Login


Sign up Login

### Login

User name

Password

Login



Management inventory system

- יש כפתור signup שמעביר לעמוד ההרשמה.  
המשתמש שרוצה להשתמש בתוכנה הוא צריך לבצע רישום לתוכנה.  
**בדיקות קלט אחרי לחיצה על כפתור הכניסה:**
- בדיקה אם השדות ריקות.
  - בדיקה אם המנהל רוצה להכניס למערכת.
  - בדיקה אם שם המשתמש קיים.
  - בדיקה אם הסיסמה נכונה.
- ויש כפתור login מעביר המשתמש למערכת אם כל הבדיקות עברו בהצלחה.



## עמוד המלאי Inventory

Management Products - Login

Inventory

Category

Customer

Order

Shelves

Statistics

Logout

### Products List

Search

Remove Edit Add

Product id	Product name	Category	Price		Weight (KG)	Quantity
			Supplier price (₪)	Client price (₪)		
1	Box set 23	Boxes and vests	190.0	250.0	5.0	203
2	Stainless steel	Boxes and vests	240.0	300.0	7.0	66
9	Insulated set 6	Screwdrivers ...	60.0	90.0	1.0	3
2222	cutter	Flyers	70.0	100.0	1.0	50
22	flyers 3 pcs	Flyers	35.0	50.0	1.5	9
122	Allen mini	Wrench	25.0	50.0	1.0	1
55	TORX key	Wrench	12.0	25.0	0.8	48
4	Swedish key 6	Wrench	10.0	20.0	0.4	21
3	Box set 127	Boxes and vests	300.0	400.0	6.0	2
23	Bits + Adapters	Screwdrivers ...	180.0	300.0	12.0	120
1212	30W recharge...	Flashlights	230.0	300.0	3.0	0
2323	30W recharge...	Flashlights	230.0	300.0	3.0	0

יש כפתורים שמעבירים המשתמש בין העמודים.

### כפתור Add:

לחיצה על כפתור הזה פותחת חלון קטן להוספת מוצר אם כל הפרטים שהמשתמש הזין תקינים אז מתווסף מוצר למלאי.

### כפתור Remove:

המשתמש מסמן את המוצר שהוא רוצה למחוק מהטבלה ואז לוחץ על כפתור המחיקה.

### כפתור Edit:

לחיצה על כפתור הזה פותחת חלון קטן לעדכון מוצר אם כל הפרטים שהמשתמש הזין תקינים אז מעדכן את המוצר הנבחר מהמלאי.

- אם כמות של המוצר פחות מ 10 מסמן את השורה בצבע כתום (אזהרה) אם כמות של המוצר שווה לאפס מסמן את השורה באדום והשאר בצבע אפור.

## עמוד הקטגוריות Categories

Category id	Category name
51	clothing and footwea
49	Screwdrivers and bit
46	Boxes and vests
50	Rinsing and vacuumin
47	Wrench
48	Flyers
52	safety
54	gardening tools
55	Flashlights

יש כפתורים שמעבירים המשתמש בין העמודים.

### כפתור Add:

המשתמש רושם את הפרטים של הקטגוריה שרוצה להכניס ואז המערכת בודקת אם השדות ריקות ואם כל הפרטים שהזין הם תקינים ובודקת גם אם קטגוריה קיימת כבר בטבלת הקטגוריות.

### כפתור Remove:

המשתמש מסמן את הקטגוריה שהוא רוצה למחוק מהטבלה ואז לוחץ על כפתור המחיקה.

### כפתור Edit:

המשתמש מסמן את הקטגוריה שהוא רוצה לעדכן מהטבלה ואז לוחץ על כפתור העדכון ואז הוא מעדכן מה שהוא רוצה .

- אם אין מוצרים בקטגוריה מסמן את הקטגוריה בצבע אדום והשאר בצבע אפור .

## עמוד הלקוח Customer

Management Product - Customer

Inventory  
Category  
Customer  
Order  
Shelves  
Statistics  
Expenses

Logout

**Customer details**

[Customer list](#)

יש כפתורים שמעבירים המשתמש בין העמודים.

### כפתור clear:

תפקיד הכפתור הוא לנקות את השדות.

### כפתור add:

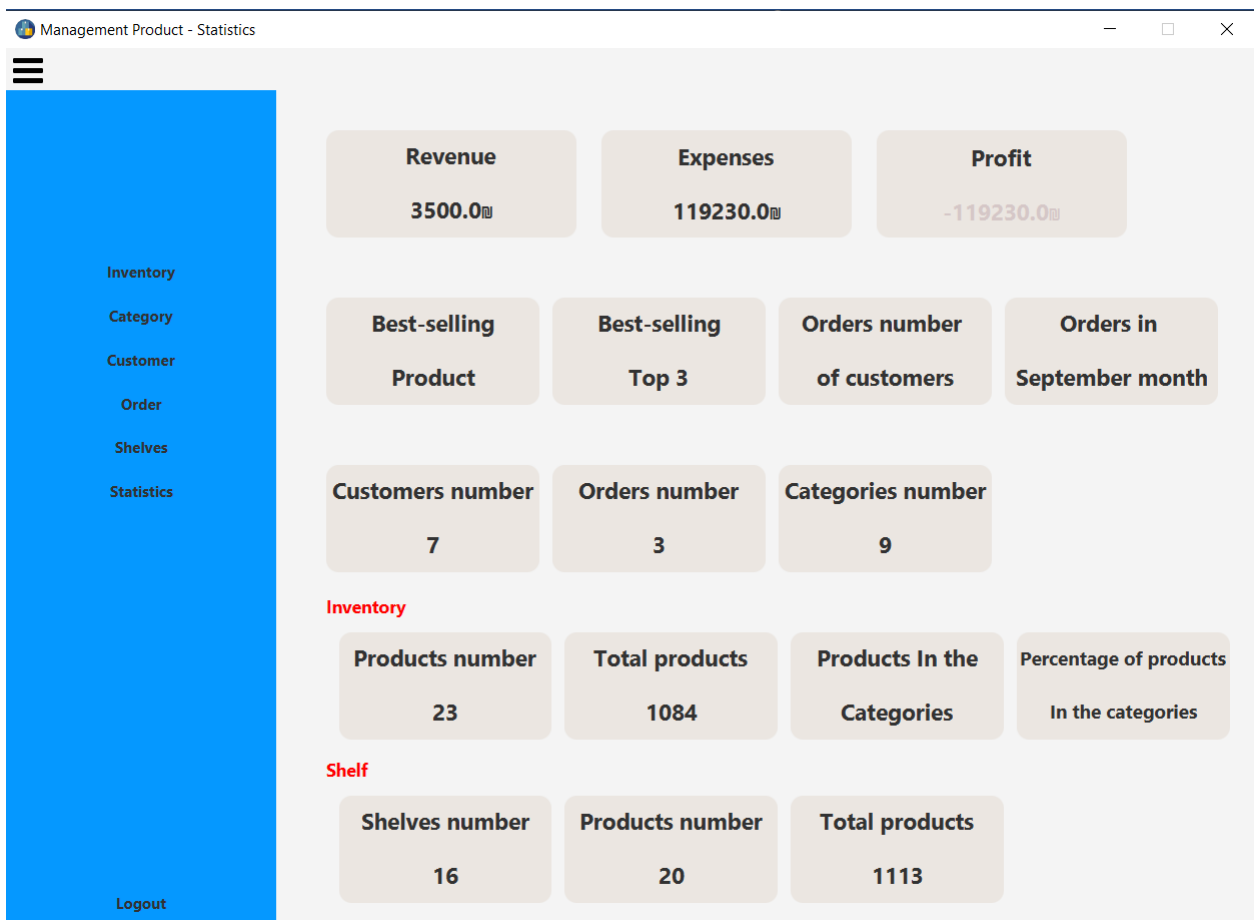
אחרי הלחיצה על כפתור ההוספה יש בדיקות תקינות קלט:

- המערכת בודקת אם המשתמש מילא את כל השדות .
- בדיקה אם המשתמש הזין שם חוקי.
- בדיקת ת.ז אם מכילה רק תשע ספרות.
- בדיקת אם הטלפון, מייל, כתובת חוקיים.
- בדיקה אם הלקוח כבר קיים בטבלת הלקוחות.

### כפתור customer list:

מעביר את המשתמש לטבלת הלקוחות הקיימים במערכת.

## סטטיסטיקות של העסק Statistics page



עמוד הסטטיסטיקות של העסק.

**Revenue:** הכנסות העסק ממכירת מוצרים.

**Expenses:** הוצאות העסק מחשמל ומים והסחורה ומשכר העובדים.

**Profit:** רווח של העסק.

**Best-selling product:** המוצרים שנמכרו הכי הרבה בעסק.

**Best-selling top 3:** שלושת המוצרים שנמכרו הכי הרבה בעסק ומציגה אותם בתוך גרף תורים.

**Orders number of customers:** מספר ההזמנות של הלקוחות.

**Orders in august month:** מציג ההזמנות בחודש הנוכחי.

**Customer's numbers:** מספר לקוחות.

**Orders number:** מספר הזמנות בעסק.

**Categories number:** מספר קטגוריות.

**Products number:** מספר המוצרים בעסק.

**Total products:** סה"כ מוצרים בעסק.

**Products in the category:** מציג מספר המוצרים בכל קטגוריה.

**Shelves number:** מספר המדפים.

**Products number:** מספר המוצרים בתוך המדף.

**Total products:** סה"כ מוצרים על המדף.

**Percentage of products in category:** מציג את אחוז המוצרים של כל קטגוריה בעסק ומציג אותם

בתוך עוגה.

## רשימת ההזמנות orders list

Management products - Orders list

### Orders List

Search

Remove View order

Order id	Customer id	Total price	Order date
21	122222222	984.0	24/8/2021
20	122222222	21.0	23/8/2021

Back

בעמוד הזה המערכת מציגה את טבלת ההזמנות בעסק.

### כפתור remove:

על המשתמש לבחור הזמנה מרשימת ההזמנות כדי לבטל את ההזמנה אם מחק המשתמש מחק את ההזמנה מחזיר את כל המוצרים בתוך ההזמנה למלאי.

### כפתור view order:

מציג כל המוצרים בתוך ההזמנה .

### כפתור back:

חוזר לעמוד ההזמנות.

## עמוד הזמנה order

Management products - Order

Customers list

Full name	I.D	Email	Telephone	Address
yosef	123333333	yosefkh@gmail.com	0542071234	yosef
samo	122222222	yosef@gmail.com	0542077322	peqiin
sahar	123412312	sahar1@gmail.com	0545472527	yanoh
yosef	987654321	yoseff@gmail.com	0502123995	nehf
tamer	987654322	tamer@gmail.com	0546515818	osfia
alex	344433332	alex@gmail.com	0503076722	haifa

Add

Customer name :  I.D :

Products list

Product id	Shelf id	Product name	Category	Price	Quantity
2	10	Stainless steel	Boxes and vests	300.0	0
1212	10	30W rechargeable pr...	Flashlights	300.0	2
122	10	Allen mini	Wrench	50.0	0
9	222	Insulated set 6	Screwdrivers and bit	90.0	46
1	1	Box set 23	Boxes and vests	250.0	1
9	1	Insulated set 6	Screwdrivers and bit	90.0	32

Orders

Shopping cart

Add

הדף הזה מאפשר למשתמש להוסיף מוצרים לסל הקניות , אם המשתמש רוצה להוסיף מוצרים לסל הקניות על המשתמש לבחור לקוח מתוך רשימת הלקוחות אחר כך המערכת מאפשרת למשתמש להוסיף מוצרים לסל הקניות.

אם המשתמש רוצה לעבור לדף אחר או לחץ על X ויש מוצרים בסל הקניות המערכת בודקת אם יש מוצרים בסל הקניות אם כן המערכת מציגה למשתמש הודעה מתאימה אם יוצא מהדף או עובר לדף אחר המערכת מחזירה באופן אוטומטי את כל המוצרים בסל הקניות למדפים.

משמעות הכפתורים בדף הזה:

1.Add: הוספת מוצרים לסל הקניות.

2.Shopping cart: עובר לס הקניות.

3.Orders: עובר לדף ההזמנות.

- אם כמות של המוצר פחות מ 10 מסמן את השורה בצבע כתום (אזהרה) אם כמות של המוצר שווה לאפס מסמן את השורה באדום והשאר בצבע אפור.

## עמוד המדפים shelves page

Shelf number	Maximum weight (KG)
10	115.0
222	46.0
1	440.0
5	150.0
123	390.0
1233	220.5
333	100.9
300	300.5
301	209.9
200	200.9
3421	100.0
43221	122.5
4321	120.8
1298	223.7

העמוד הזה מציג את המדפים בתוך העסק.

### כפתורים:

1. Remove - מחיקת מדף אם המדף מכיל מוצרים המשתמש לא יכול למחוק.
2. Edit - עדכון פרטים של מדף המשתמש יכול לערוך את המשקל המקסימלי של העסק המערכת בודקת אם המדף מכיל מוצרים אם כן בודק אם המשקל המקסימלי החדש מתאים לפי מה שמכיל המדף ממוצרים.
3. Add - הוספת מדף לטבלת המדפים.
4. View shelf - מציג את המוצרים בתוך המדף שהמשתמש בחר אותו.
5. Add product - הוספת מוצרים למדף שהמשתמש בחר.

- אם משקל המוצרים בתוך המדפים מעל 90% ביחס למשקל המקסימלי מסמן את המדף בצבע אדום ושאר בצע אפור.

## עמוד ההוצאות Expenses

Management products - Expenses

Inventory

Category

Customer

Order

Shelves

Statistics

Expenses

Logout

Expenses

Customers salary

Update Salary

Full name	User name	Salary day	General Salary	Update salary date
yosef khier	usr123333	0.0	850.0	-
yosefkh khier	usr123	0.0	200.0	-
sami hihi	usr1	0.0	555.0	-

Current month

Electrical expenses

Update Show

Water expenses

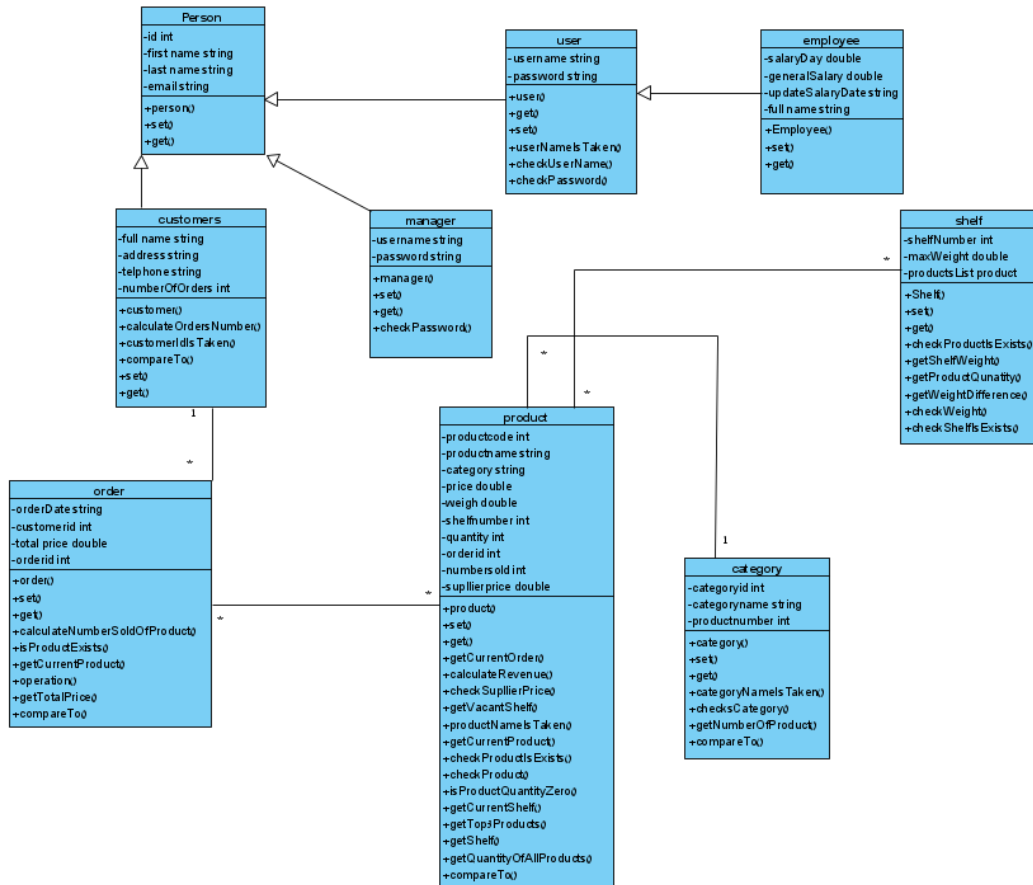
Update Show

העמוד הזה מציג את הפרטים של העובדים.

Update salary - חישוב משכורת היומית של עובד שהמנהל בחר והמנהל יכול לעדכן את המשכורת היומית של העובד.  
יש גם הוצאות מים וחשמל המנהל יכול להוסיף הוצאות מים וחשמל בעסק בכל יום ואם באותו חודש המשתמש רוצה לעדכן את סוג ההוצאה הוא מעדכן מה שהזין בהתחלה .



## תרשים הסופי של המחלקות



## הסבר על כל מחלקה

### Person

במחלקה הזאת יש כמה תכונות משותפות לכמה מחלקות שיורשות מהמחלקה הזאת כמו לקוח משתמש מנהל.

### User

המחלקה הזאת יורשת ממחלקה person שיש להם כמה תכונות משותפות ובנוסף יש שם משתמש וסיסמה.

יש פונקציה שבודקת אם שם המשתמש קיים כדי להירשם ושתי פונקציות בודקות אם שם המשתמש והסיסמה תואמים למה יש במערכת.

### Customers

המחלקה הזאת יורשת ממחלקת person ויש לה עוד כמה תכונות ומחלקת הלקוח מקושרת למחלקת ההזמנה בקשר יחיד לרבים.

יש פונקציה שסופרת כמה הזמנות ללקוח ויש פונקציה בודקת אם הלקוח קיים ברשימת הלקוחות.

### Manager

מחלקת המנהל יורשת ממחלקת person ויש בה פונקציה שבודקת אם הסיסמה תקינה של המנהל.

### Employee

מחלקת העובד יורשת ממחלקת משתמש ובנוסף יש עוד תכונות שקשורות אך ורק לעובד.

### **:Product**

מחלקת המוצר יש לה כמה תכונות והיא מקושרת לכמה מחלקות שונות יש לכל מוצר קטגוריה אחת ולכל קטגוריה מכילה כמה מוצרים וגם מחלקת המוצר מקושרת למחלקת המדף ולהזמנה בקשר רבים לרבים.

### **:Shelf**

מחלקת המדף יש לה כמה תכונות שקשורים למדף ומחלקת המדף מקושרת למחלקת המוצר בקשר רבים לרבים.

### **:Category**

מחלקת הקטגוריה יש לה כמה תכונות שקשורות לקטגוריה ומחלקה הזו מקושרת עם מחלקת מוצר רבים לרבים כי לכל קטגוריה יש כמה מוצרים וכל מוצר שייך לקטגוריה מסוימת.

### **:Order**

מחלקת ההזמנה יש לה תכונות שקשורות להזמנה ומחלקת ההזמנה מקושרת למחלקת הלקוח בקשר יחיד לרבים כל לקוח יכול להזמין יותר מהזמנה אחת ולכל הזמנה שייכת ללקוח אחד.

## הסבר על כל הפונקציות

### הסבר הפונקציות במחלקת Conditions

Public static Boolean isContainsUsrInField (String str)

הפונקציה הזו בודקת אם ומחרוזת מכילה ב שלושה איברים הראשונים ב usr .

Public static Boolean isContainsDigits (String str)

הפונקציה הזו בודקת אם המחרוזת מכילה רק מספרים מאפס עד תשע.

Public static Boolean isContainsDigitsWithLine (String str)

הפונקציה הזו בודקת אם המחרוזת מכילה מספרים מאפס עד תשע או קו עליון.

Public static Boolean isFloatNumber (TextField field)

הפונקציה הזו בודקת אם המחרוזת ה התקבלה מכילה מספר עם שארית או מספר בלי שארית.

Public static Double get2DigitsAfterPoint (String str)

הפונקציה הזו מחזירה את שתי הספרות אחרי הנקודה במספר עם שארית.

Public static Boolean isEmailValid (String str)

הפונקציה הזו בודקת אם המחרוזת שהתקבלה מתאימה לתנאי המייל ( חייב שתהיה הסיומת @gmail.com או @walla.com ).

Public static Boolean isContainsCharacter (String str)

הפונקציה הזו בודקת אם המחרוזת שהתקבלה מכילה תווים מ a-z או A-Z.

Public static Boolean isFullNameValid (TextField field)

הפונקציה הזו בודקת אם המחרוזת שהתקבלה מכילה תווים מ a-z או A-Z או רווח .

Public static Boolean isTelephoneNumberValid (String str)

הפונקציה הזו בודקת אם המחרוזת שהתקבלה מתאימה לתנאי מספר הטלפון לדוגמה : אפשר להזין מספר טלפון בגודל 10 או 12 ו(0542077433,054207-7433) וחייב שיהיה שלושה תווים הראשונים שווים ל (04 או 03 או 07).

Public static Boolean isPhoneNumberValid (String str)

הפונקציה הזו בודקת אם המחרוזת שהתקבלה מתאימה לתנאי מספר פלאפון נייד לדוגמה : אפשר להזין מספר טלפון בגודל 9 או 11 ו(049971234,04-997-1234) וחייב שיהיה השני תווים הראשונים שווים ל (054 או 057 או 054 או 052 או 050).

Public static Boolean isAreaCodeValid (String str, String [] str)

הפונקציה הזו בודקת אם תחילת מספר הטלפון או הפלאפון תקינים.

Public static Boolean isAddressValid (String str)

הפונקציה הזו בודקת אם המחרוזת מכילה מ a-z או A-Z או רווח או קו עליון.

Public static Boolean isIdPersonValid (String str)

הפונקציה הזו בודקת אם המחרוזת מתאימה לתנאי תעודת הזהות של הבן אדם.

Public static Boolean checkManagerStatus (Button button)

הפונקציה הזו בודקת אם הסטטוס של המנהל שווה ל false אם כן מעלים את הכפתור ואם הסטטוס של המנהל שווה ל true הכפתור מופיע.

Public static Boolean removeLastIndexInString (TextField textField)

הפונקציה הזו מוחקת את התו האחרון במחרוזת.

Public static <T> ObservableList<T> getItemssSearch (ObservableList<T> list, String str)

הפונקציה הגנרית הזו היא מיועדת לשדה של החיפוש מחזירה את כל האובייקטים שמתאימים למחרוזת שהתקבלה .

Public static <T> Boolean checkStringInSearch (T object, String str)

הפונקציה הגנרית הזו היא מיועדת לשדה של החיפוש והיא בודקת אם האובייקט מתאים למחרוזת שהתקבלה לדוגמה : אם המחרוזת שהתקבלה app האובייקט מטיפוס מוצר אז מחזירה אם שם המוצר או קוד מוצר שווה למחרוזת אם כן מחזיר true אם לא מחזירה false והפונקציה הזו עובדת על כמה אובייקטים שונים.

## הסבר הפונקציות במחלקת Field

Public static Boolean checkEmptyFields (TextField field1, TextField field2)

הפונקציה הזאת בודקת אם תוכן שתי השדות לא ריק אם ריק הפונקציה מחזירה false אם לא ריק הפונקציה מחזירה true .

Public static Boolean checkEmptyFields (TextField field1)

הפונקציה הזאת בודקת אם תוכן השדה לא ריק אם ריק הפונקציה מחזירה false אם לא ריק הפונקציה מחזירה true .

Public static Boolean checkEmptyFields (TextField field1, TextField field2, TextField field3, TextField field4, TextField field5)

הפונקציה הזאת בודקת אם תוכן חמשה השדות לא ריק אם ריק הפונקציה מחזירה false אם לא ריק הפונקציה מחזירה true .

Public static Boolean checkEmptyFields (TextField field1, TextField field2, TextField field3, TextField field4)

הפונקציה הזאת בודקת אם תוכן ארבע השדות לא ריק אם ריק הפונקציה מחזירה false אם לא ריק הפונקציה מחזירה true .

Public static Boolean checkEmptyFields (TextField field1, TextField field2, TextField field3, TextField field4, TextField field5, TextField field6)

הפונקציה הזאת בודקת אם תוכן שש השדות לא ריק אם ריק הפונקציה מחזירה false אם לא ריק הפונקציה מחזירה true .

Public static Boolean checkIdField (TextField field, Label label, String color)

הפונקציה הזאת אחראית על השדה תעודת הזהות של בן אדם והיא בודקת אם גודל הספרות בתוך השדה שווה ל 9 ובודקת אם כל התווים מ 0-9 אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static void changeBorderColor (TextField field1, TextField field2, TextField field3, TextField field4, TextField field5)

הפונקציה הזאת משנה צבע של כל השדות הריקות לאדום והשאר לצבע שחור.

Public static Boolean checkUserNameValid (TextField field, Label label)

הפונקציה הזאת אחראית על השדה שם משתמש והיא בודקת אם השדה מתחיל ב usr אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkAddressField (TextField field, Label label)

הפונקציה הזאת אחראית על השדה הכתובת והיא בודקת אם השדה תקין אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkEmailField (TextField field, Label label, String color)

הפונקציה הזאת אחראית על השדה המייל והיא בודקת אם הסימנים הסימנים של המייל תקינים אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkPhoneNumberField (TextField field, Label label)

הפונקציה הזאת אחראית על השדה מספר טלפון והיא בודקת אם זה מספר טלפון או מספר פלאפון אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkNameField (TextField field, Label label)

הפונקציה הזאת אחראית על השדה שם ו שם משפחה והפונקציה בודקת אם בשדות מכילים תווים מ a-z או A-Z אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkNumberField (TextField field, Label label)

הפונקציה הזאת אחראית על כל השדות שיש בהם רק מספרים והפונקציה בודקת אם כל התוכן של השדה מכיל רק תווים מ 0 – 9 אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkNumberField (TextField field, Label label)

הפונקציה הזאת אחראית על השדה שם המלא והפונקציה בודקת אם השדה השם המלא תקין אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkPasswordField (TextField field, Label label)

הפונקציה הזאת אחראית על הסיסמה והפונקציה בודקת אם הגדול של התוכן גדול מ 6 אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static Boolean checkPasswordField (TextField field, Label label)

הפונקציה הזאת אחראית על השדות שאפשר להכניס בה מספר עם שארית לדוגמא : מחיר ספק וכו' אם הכל עובר בהצלחה מחזיר true ולא משנה את הצבע ואם לא עבר בהצלחה מחזיר false ומשנה את הצבע של השדה לאדום ומציג label מתאים לסוג השגיאה.

Public static void changeBorderColor (TextField field, String color)

הפונקציה הזאת משנה צבע המסגרת של השדה לפי הצבע שהתקבל בפונקציה.

## הסבר הפונקציות במחלקת DatabaseManager

Public static Connection getconn ()

הפונקציה הזאת אחראית על התחברות ל database.

Public static void addCategory (Category category)

הפונקציה הזו להוספת קטגוריה לרשימת הקטגוריות במסד הנתונים.

Public static void addProduct (Product product)

הפונקציה הזו להוספת מוצר לרשימת המוצרים במסד הנתונים.

Public static void addProductToShelf (Product product, Shelf shelf, int quantity)  
הפונקציה הזו להוספת מוצר למדף מסוים במסד הנתונים.

Public static void addShelf (Shelf shelf)  
הפונקציה הזו להוספת מדף לרשימת המדפים במסד הנתונים.

Public static void addUser (User user)  
הפונקציה הזו להוספת משתמש חדש לרשימת המשתמשים במסד הנתונים.

Public static void addCustomer (Customer customer)  
הפונקציה הזו להוספת לקוח חדש לרשימת הלוקחות במסד הנתונים.

Public static void insertProductToShoppingCart (Product product, Shelf shelf)  
הפונקציה הזו מוסיפה את המוצר לסל הקניות.

Public static void insertProductToOrder (Product product, Order order)  
הפונקציה הזו מוסיפה את המוצר להזמנה במסד הנתונים.

Public static void addExpenses (Expenses exp)  
הפונקציה הזו להוספת הוצאת מים או חשמל לרשימת ההוצאות במסד הנתונים.

Public static void addProductToOrder (int orderNumber)  
הפונקציה הזו להוספת המוצרים להזמנה ספציפית במסד הנתונים.

Public static void removeOrderQuery (Order order, String tableName)  
הפונקציה הזו למחיקת הזמנה מטבלת ההזמנות במסד הנתונים.

Public static void deleteCategory (Category category)  
הפונקציה הזו למחיקת קטגוריה מטבלת הקטגוריות במסד הנתונים.

Public static void removeShelf (Shelf shelf)  
הפונקציה הזו למחיקת מדף מטבלת המדפים במסד הנתונים.

Public static void removeCustomer (Customer customer)  
הפונקציה הזו למחיקת לקוח מטבלת הלוקחות במסד הנתונים.

Public static void deleteProductInShelf (String tableName, Product product, Shelf shelf)  
הפונקציה הזו למחיקת מוצר מהמדף ממסד הנתונים.

Public static void deleteProduct (Product product, String tableName)  
הפונקציה הזו למחיקת מוצר מטבלת המוצרים ממסד הנתונים.

Public static void removeShoppingCart ()  
הפונקציה הזו למחיקת כל המוצרים בסל הקניות במסד הנתונים.

Public static ObservableList<Product> getProductFromShoppingCart ()  
הפונקציה הזו מחזירה את כל המוצרים בתוך סל הקניות במסד הנתונים.

Public static ObservableList<Product> getProductsInShelf (Shelf shelf)  
הפונקציה הזו מחזירה את כל המוצרים בתוך מדף מסוים במסד הנתונים.

Public static ObservableList<Order> getOrders ()  
הפונקציה הזו מחזירה את כל הזמנות במסד הנתונים.

Public static ObservableList<Category> getCategorys ()  
הפונקציה הזו מחזירה את כל הקטגוריות במסד הנתונים.

Public static ObservableList<User> getUsers ()  
הפונקציה הזו מחזירה את כל המשתמשים במסד הנתונים.

Public static ObservableList<User> getEmployees ()  
הפונקציה הזו מחזירה את כל העובדים במסד הנתונים.

Public static Manager getManager ()  
הפונקציה הזו מחזירה את פרטים של המנהל במסד הנתונים.

Public static ObservableList<Product> getProducts ()  
הפונקציה הזו מחזירה את כל המוצרים הנמצאים במלאי ממסד הנתונים.

Public static int getProductQunatityInOrder (Product product, Order order)  
הפונקציה הזו מחזירה את כמות של מוצר בהזמנה מסוימת ממסד הנתונים.

Public static int getProductQuantityInShelf (String tableName, Product product, Shelf shelf)  
הפונקציה הזו מחזירה את כמות של מוצר במדף מסוים ממסד הנתונים.

Public static ObservableList<Product> getProductsOfOrders ()  
הפונקציה הזו מחזירה את כל המוצרים בהזמנות במסד הנתונים.

Public static ObservableList<Product> getProductOfOrder (Order order)  
הפונקציה הזו מחזירה את כל המוצרים בהזמנה ספציפית המסד בנתונים.

Public static ObservableList<Product> getProductsInShelves ()  
הפונקציה הזו מחזירה את כל המוצרים בכל המדפים במסד הנתונים.

Public static int getProductQuantity (String tableName, int id) ()  
הפונקציה הזו מחזירה את כל כמות של מוצר מסוים במלאי במסד הנתונים.

Public static int getOrderCode (Order order) ()  
הפונקציה הזו מחזירה את קוד ההזמנה האחרון שנכנס למסד הנתונים.

Public static ObservableList<Expenses> getExpenss ()  
הפונקציה הזו מחזירה את כל ההוצאות של מים וחשמל ממסד הנתונים.

Public static ObservableList<Expenses> getExpensesType (String description)  
הפונקציה הזו מחזירה את כל ההוצאות מסוג מים או חשמל ממסד הנתונים.

Public static void updateProductInShelf (String tabelName, Product product, Shelf shelf, int quantity)  
הפונקציה הזו מעדכנת כמות המוצר בתוך המדף לפי מספר מדף וקוד מוצר במסד הנתונים.

Public static void updateProduct (String tableName, Product product, int quantity)

הפונקציה הזו מעדכנת כמות מוצר מטבלה מסוימת (טבלת המוצר או טבלת סל הקניות) במסד הנתונים.

Public static void updateProductInOrder (Product product, Order order, Shelf shelf, int quantity)

הפונקציה הזו מעדכנת כמות המוצר בהזמנה במסד הנתונים.

Public static void updateProductCategory (Category newCategory, Category oldCategory)

הפונקציה הזו מעדכנת שם הקטגוריה לפי שם הקטגוריה הישן במסד הנתונים.

Public static void updateCustomer (Customer customer)

הפונקציה הזו מעדכנת פרטים של לקוח מסוים מטבלת הלקוחות במסד הנתונים.

Public static void updateProduct (Product product)

הפונקציה הזו מעדכנת נתונים של מוצר מסוים מטבלת המוצרים במסד הנתונים.

Public static void updateShelf (Shelf shelf)

הפונקציה הזו מעדכנת נתונים של מדף מסוים מטבלת המדפים במסד הנתונים.

Public static void updateTotalPriceOfOrder (Order order, Double totalPrice)

הפונקציה הזו מעדכנת סך כל מחיר של הזמנה במסד בנתונים.

Public static void editCategory (Category category)

הפונקציה הזו מעדכנת הנתונים של הקטגוריה במסד הנתונים במסד בנתונים.

Public static void editEmployee (Employee employee)

הפונקציה הזו מעדכנת הנתונים של העובד (משכורת היומית ומשכורת כללית וגם תאריך עדכון המשכורת) בטבלת המשתמש במסד בנתונים.

Public static void editExpenses (Expenses exp)

הפונקציה הזו מעדכנת סוג ההוצאה (הוצאת חשמל או מים) בטבלת ההוצאות במסד בנתונים.

Public static void editManagerStatus (Manager manager)

הפונקציה הזו מעדכנת סטטוס של המנהל אם הוא במצב (online או offline) במסד בנתונים.

### הסבר הפונקציות במחלקת AnimationClass

Public void flashingLabel (Label label, Double second, Color color1, Color color2)

הפונקציה הזו אחראית על הבהוב ה label לפי הצבעים ה התקבלו בפונקציה.

Public void sideAnimation (AnchorPane pane)

הפונקציה הזו מיועדת לתפריט שאם התפריט מופיע אז לסגור אותו אם התפריט מופיע אז לפתוח את התפריט.

Public void openSide (AnchorPane pane)

הפונקציה הזו מיועדת לפתיחת התפריט.

Public void closeSide (AnchorPane pane)

הפונקציה הזו מיועדת לסגירת התפריט.

### הסבר הפונקציות במחלקת RowTable

Public <T> void changeRowsColor (TableView<T> table)



הפונקציה הגנרית הזאת מיועדת לשינוי צבע של שורה לטבלאות שונות כגון : לטבלת המוצרים אם כמות המוצרים שווה לאפס מדליק את השורה מצבע אדום ואם פחות מעשר מדליק את השורה בצבע כתום , לטבלת הקטגוריה אם קטגוריה ריקה לא שייך לה מוצרים מדליק את השורה באדום , לטבלת המדף אם משקל המדף מגיע ל 90% או יותר ביחס למשקל המקסימלי מדליק את השורה בצבע אדום.

### הסבר הפונקציות במחלקת SwitchingPages

Public static void switchPage (Event event, String pageName, String title)  
הפונקציה הזו מעבירה לחלון אחר לפי שם הטבלה ה התקבל בפונקציה ומגדיר לחלון את הכותרת.

Public static void openWindowInCenter (String pageName, Event event, String title)  
הפונקציה הזו מעבירה לחלון אחר לפי שם הטבלה ה התקבל בפונקציה ומגדיר לחלון את הכותרת ופותח את החלון החדש במרכז המסך.

Public static void exitStage (Event event)  
הפונקציה הזו סוגרת את החלון (exit).

### הסבר הפונקציות במחלקת Message

Public static void showMessageWithOKButton (AlertType type, String contentText, String headerText)  
הפונקציה הזו מציגה הודעה עם כפתור אחד (ok button).

Public static buttonType showMessageWithYesNoButtons (AlertType type, String contentText, String headerText)  
הפונקציה הזו מציגה הודעה עם שני כפתורים (no button ו ok button) ומחזירה את ה button type (no או yes) .

Public int showMessageWithTextField (String headerText, String contentText)  
הפונקציה הזו מציגה הודעה עם שני כפתורים (cancel button ו ok button) ושדה ומחזירה מספר מה שהוקלד בתוך השדה אם הקלט לא תקין מחזירה -1 .

### הסבר הפונקציות במחלקת DateClass

Public String getDateToday ()  
הפונקציה הזו מחזירה את התאריך הנוכחי כמחרוזת.

Public int getCurrentYear ()  
הפונקציה הזו מחזירה את השנה הנוכחית כמספר.

Public int getCurrentMonth ()  
הפונקציה הזו מחזירה את החודש הנוכחי כמספר.

Public int getMonth (String orderDate)  
הפונקציה הזו מקבלת תאריך כמחרוזת ומחזירה את החודש כמספר בתאריך ה התקבל בפונקציה.

Public int getYear (String orderDate)  
הפונקציה הזו מקבלת תאריך כמחרוזת ומחזירה את השנה כמספר בתאריך ה התקבל בפונקציה.

Public int getMonth (int month)

הפונקציה הזו מקבלת חודש כמספר ומחזירה את החודש כמחרוזת.

### הסבר הפונקציות במחלקת LogoutClass

Public static void logoutWindow ()

הפונקציה הזו בודקת אם סל קניות מכיל מוצרים אם כן מציג הודעה מתאימה למשתמש שאם יצא מהחלון המערכת מחזירה את המוצרים אוטומטי למדפים.

Public static void returnProductsToShelves ()

הפונקציה הזו מחזירה את כל המוצרים בסל הקניות למדפים.

### הסבר הפונקציות במחלקת Category

Public int getNumberOfProduct (Category category)

הפונקציה הזו מקבלת קטגוריה ומחזירה מספר המוצרים בתוך הקטגוריה.

Public Boolean checksCategory (Category category)

הפונקציה הזו בודקת אם הקטגוריה מכילה מוצרים אם הקטגוריה מכילה מוצרים מחזירה true אם הקטגוריה לא מכילה מוצרים מחזירה false.

Public Boolean categoryNamelsTaken (Category category, ObservableList<Category> categoryList)

הפונקציה הזו בודקת אם הקטגוריה שהתקבלה בפונקציה קיימת ברשימת הקטגוריות.

### הסבר הפונקציות במחלקת Customer

Public void calculateOrdersNumber (Customer customer)

הפונקציה הזו יודעת לחשב את מספר ההזמנות שהלקוח הזמין.

Public Boolean customerIdsTaken (Customer customer, ObservableList<Customer> customerList)

הפונקציה הזו בודקת אם הלקוח שהתקבל בפונקציה קיים ברשימת הלקוחות.

### הסבר הפונקציות במחלקת Expenses

Public Expenses isExpensesPaid (ObservableList<Expenses> expList)

הפונקציה הזו בודקת אם סוג ההוצאה כבר שולם בחודש הנוכחי אם סוג ההוצאה כבר שולם הפונקציה מחזירה את האובייקט אם לא מחזירה null .

### הסבר הפונקציות במחלקת Manager

Public Boolean checkPassword (Manager m, String password)

הפונקציה הזו אם הסיסמה של המנהל נכונה אם הסיסמה נכונה מחזירה true אם לא נכונה הפונקציה מחזירה false.

### הסבר הפונקציות במחלקת Order

Public int calculateNumberSoldOfProduct (Product product)

הפונקציה הזו מקבלת אובייקט מסוג מוצר ומחזירה כמה המוצר ה התקבל נמכר בחודש הנוכחי .

Public Boolean isProductExists (Product p)

הפונקציה הזו מוצר ובודקת אם המוצר ה התקבל בפונקציה קיים בהזמנה .

Public Double getTotalPrice (ObservableList<Product> productsList)

הפונקציה הזו מחשבת את סך כל מחיר המוצרים בתוך ההזמנה הפונקציה מחשבת את כמות המוצר כפול מחיר המוצר.

Public Product getCurrentProduct (Product product)

הפונקציה הזו מחזירה את המוצר ה התקבל לפי קוד המוצר.

## הסבר הפונקציות במחלקת Product

Public Order getCurrentOrder (int orderCode)

הפונקציה הזו מחזירה את ההזמנה לפי קוד ההזמנה.

Public Double calculateRevenue ()

הפונקציה הזו יודעת לחשב את הרווח של העסק לפי סכום כל ההזמנות .

Public Boolean checkSupplierPrice (Product product)

הפונקציה הזו בודקת אם מחיר ספק של המוצר גדול ממחיר לקוח של המוצר הפונקציה מחזירה false אם מחיר לקוח של המוצר גדול ממחיר ספק של המוצר הפונקציה מחזירה true .

Public Shelf getVacantShelf ()

הפונקציה הזו בודקת אם יש מדף פנוי למוצר אם יש מדף פנוי הפונקציה מחזירה את המדף הפנוי אם אין מדף פנוי הפונקציה מחזירה false.

Public Boolean productNamesTaken (Product product, ObservableList<Product> productList)

הפונקציה הזו בודקת אם שם מוצר קיים ברשימת המוצרים אם שם המוצר קיים הפונקציה מחזירה true אם שם מוצר לא קיים הפונקציה מחזירה false.

Public Boolean checkProductsExists (ObservableList<Product> productList)

הפונקציה הזו בודקת אם קוד מוצר קיים ברשימת המוצרים אם קוד המוצר קיים הפונקציה מחזירה true אם קוד מוצר לא קיים הפונקציה מחזירה false.

Public Boolean checkProduct (int quantity, int productQuantity)

הפונקציה הזו כמות הנבחרת יותר גדולה מהכמות במלאי.

Public Boolean isProductQuantityZero (int quantity)

הפונקציה הזו בודקת אפ כמות המוצר הגיעה לאפס אם כן הפונקציה מחזירה false אם לא שווה לאפס הפונקציה מחזירה true

Public Shelf getShelf (int shelfNumber)

הפונקציה הזו מחזירה את המדף לפי קוד מדף.

Public ObservableList<Product> getTop3Products (ObservableList<Product> list)

הפונקציה הזו מחזירה את שלושה המוצרים הכי נמכרים.

Public int getQuantityOfAllProducts (ObservableList<Product> productList)

הפונקציה מחזירה כמות כל המוצרים ברשימת המוצרים.

## הסבר הפונקציות במחלקת Shelf

Public int checkProductsExists (Product product)

הפונקציה בודקת אם המוצר קיים במדף אם כן קיים הפונקציה מחזירה את כמות המוצר אם לא קיים הפונקציה מחזירה 1-.

Public Double getShelfWeight ()

הפונקציה הזו מחזירה את המשקל של כל המוצרים בתוך המדף.

Public int getProductQuantity (Product product)

הפונקציה הזו מחזירה כמה מוצרים אפשר להכניס למדף.

Public Double getWeightDifference ()

הפונקציה הזו מחזירה את ההפרש בין המשקל המקסימלי שאפשר להכניס למדף למשקל כל המוצרים בתוך המדף.

Public Boolean checkShelfExists (ObservableList<Shelf> shelfList)

הפונקציה הזו בודקת אם המדף קיים לפי מספר מדף אם המדף קיים הפונקציה מחזירה false אם לא קיים הפונקציה מחזירה true.

Public Boolean checkWeight (Product product, int quantity)

הפונקציה הזו בודקת אם משקל כל המוצרים בתוך המדף ומשקל המוצר ה התקבל פחות שווה למשקל המקסימלי של המדף.

## הסבר הפונקציות במחלקת User

Public Boolean usernamesTaken (ObservableList<User> userList, User user)

הפונקציה הזו אם המשתמש קיים ברשימת המשתמשים אם כן קיים.

Public Boolean checkUserName (ObservableList<User> userList, String userName)

הפונקציה הזו בודקת אם שם המשתמש קיים אם קיים מחזיר את האובייקט של המשתמש אם לא קיים הפונקציה מחזירה null .

Public Boolean checkPassword (User user, String Password)

הפונקציה הזו בודקת אם הסיסמה של המשתמש ה התקבל בפונקציה נכונה אם כן הפונקציה מחזירה true אם לא תקינה הפונקציה מחזירה false.

# **קוד קבצי FXML של התוכנה**

**(נכתב בסביבת פיתוח Eclipse)**

## קובץ Login.fxml

קובץ עיצוב שמכיל את העיצוב של דף הכניסה.

```
<?xml version="1.0" encoding="UTF-8"?>
<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.text.Font?>
<BorderPane id="background" fx:id="Loginpage" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="560.0"
prefWidth="670.0" styleClass="background" stylesheets="@application.css"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.Controller">
    <right>
        <AnchorPane prefHeight="560.0" prefWidth="336.0" BorderPane.alignment="CENTER">
            <children>
                <TextField id="username" fx:id="userNameField" focusTraversable="false"
layoutX="104.0" layoutY="213.0" prefHeight="27.0" prefWidth="156.0" promptText="User
name" style="-fx-background-color: transparent; -fx-border-color: BLUE; -fx-border-
width: 0px 0px 2px 0px;" stylesheets="@../../../../hi/src/application/application.css">
                    <cursor>
                        <Cursor fx:constant="TEXT" />
                    </cursor>
                </TextField>
                <Label layoutX="142.0" layoutY="145.0"
stylesheets="@../../../../java%20effects/src/application/application.css" text="Login"
textFill="#059bff">
                    <font>
                        <Font name="System Bold" size="22.0" />
                    </font>
                </Label>
                <PasswordField fx:id="passwordField" focusTraversable="false"
layoutX="104.0" layoutY="286.0" prefHeight="27.0" prefWidth="156.0"
promptText="Password" style="-fx-background-color: transparent; -fx-border-color:
blue; -fx-border-width: 0px 0px 2px 0px;" />
                <Label fx:id="SignUpbtn" alignment="CENTER" layoutX="195.0"
layoutY="15.0" onMouseReleased="#SignUpbtn" prefHeight="28.0" prefWidth="82.0"
style="-fx-background-color: grey; -fx-background-radius: 20px;" text="Sign up"
textFill="WHITE">
                    <cursor>
                        <Cursor fx:constant="HAND" />
                    </cursor>
                    <font>
                        <Font name="System Bold" size="12.0" />
                    </font>
                    <padding>
                        <Insets right="30.0" />
                    </padding>
                </Label>
                <Label alignment="CENTER" layoutX="246.0" layoutY="15.0"
prefHeight="28.0" prefWidth="75.0" style="-fx-background-color: #0598ff; -fx-
background-radius: 20px;" text="Login" textFill="WHITE">
                    <cursor>
                        <Cursor fx:constant="HAND" />
                    </cursor>
                </Label>
            </children>
        </AnchorPane>
    </right>
</BorderPane>
```

```

        </cursor>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Label>
    <FontAwesomeIcon fill="#0598ff" glyphName="USER" layoutX="84.0"
layoutY="236.0" scaleX="1.5" scaleY="1.5" />
    <FontAwesomeIcon fill="#0598ff" glyphName="LOCK" layoutX="85.0"
layoutY="310.0" scaleX="1.5" scaleY="1.5" />
    <Label fx:id="errtxt" layoutX="84.0" layoutY="184.0" prefHeight="18.0"
prefWidth="252.0" textFill="#f50808">
        <font>
            <Font name="System Bold" size="12.0" />
        </font></Label>
    <Label id="buttonsShadow" alignment="CENTER" focusTraversable="false"
layoutX="107.0" layoutY="366.0" onMouseReleased="#Loginbutton" prefHeight="36.0"
prefWidth="128.0" style="-fx-background-color: #0598ff; -fx-background-radius: 5px;">
        <graphic>
            <Label text="Login" textFill="WHITE">
                <font>
                    <Font name="System Bold" size="14.0" />
                </font>
                <graphic>
                    <Label>
                        <graphic>
                            <FontAwesomeIcon fill="WHITE" glyphName="KEY"
size="1.5em" textOrigin="BOTTOM" />
                        </graphic>
                    </Label>
                </padding>
                <Insets right="10.0" />
            </padding>
        </Label>
    </graphic>
</Label>
</graphic>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Label>
<Label fx:id="userNameLabel" layoutX="82.0" layoutY="261.0"
prefHeight="18.0" prefWidth="252.0" textFill="#f50808">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
</Label>
<Label fx:id="passwordLabel" layoutX="82.0" layoutY="331.0"
prefHeight="18.0" prefWidth="252.0" textFill="#f50808">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
</Label>
</children>
<cursor>
    <Cursor fx:constant="DEFAULT" />
</cursor>
</AnchorPane>
</right>
<center>
    <AnchorPane id="panelRight" disable="true" prefHeight="470.0" prefWidth="339.0"
style="-fx-background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Label layoutX="51.0" layoutY="328.0" text="Management inventory"
textFill="#130000">

```

```

        <font>
            <Font name="Lucida Calligraphy Italic" size="18.0" />
        </font>
    </Label>
    <Label layoutX="133.0" layoutY="364.0" text="system" textFill="#130000">
        <font>
            <Font name="Lucida Calligraphy Italic" size="18.0" />
        </font>
    </Label>
    <ImageView fitHeight="235.0" fitWidth="297.0" layoutX="29.0"
layoutY="99.0" pickOnBounds="true" preserveRatio="true">
        <image>
            <Image url="file:images/inventoryImage.png" />
        </image>
    </ImageView>
</children>
</AnchorPane>
</center>
</BorderPane>

```



קובץ עיצוב שמכיל את העיצוב של דף ההרשמה.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.CheckBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.Blend?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.text.Font?>

<BorderPane fx:id="LoginPage" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="560.0" prefWidth="670.0"
stylesheets="@../../../../java%20effects/src/application/application.css"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.SignupController">
    <right>
        <AnchorPane prefHeight="560.0" prefWidth="335.0" BorderPane.alignment="CENTER">
            <children>
                <TextField id="username" fx:id="userField" focusTraversable="false"
layoutX="98.0" layoutY="355.0" onKeyTyped="#userNameTyped" prefHeight="27.0"
prefWidth="156.0" promptText="User name" style="-fx-background-color: transparent; -fx-border-color: BLUE; -fx-border-width: 0px 0px 2px 0px;"
stylesheets="@../../../../hi/src/application/application.css">
                    <cursor>
                        <Cursor fx:constant="TEXT" />
                    </cursor>
                </TextField>
                <Label layoutX="128.0" layoutY="57.0"
stylesheets="@../../../../java%20effects/src/application/application.css" text="Sign up"
textFill="#059bff">
                    <font>
                        <Font name="System Bold" size="22.0" />
                    </font>
                </Label>
                <Button focusTraversable="false" layoutX="117.0" layoutY="515.0"
mnemonicParsing="false" onAction="#SignUpbutton" prefHeight="32.0" prefWidth="118.0"
style="-fx-background-color: #0598ff; -fx-background-radius: 5px;"
stylesheets="@../../../../java%20effects/src/application/application.css" text="Sign up"
textFill="WHITE">
                    <cursor>
                        <Cursor fx:constant="HAND" />
                    </cursor>
                    <effect>
                        <Blend />
                    </effect>
                    <font>
                        <Font name="System Bold" size="12.0" />
                    </font>
                </Button>
                <PasswordField fx:id="passwordField" focusTraversable="false"
layoutX="98.0" layoutY="415.0" onKeyTyped="#passwordTyped" prefHeight="27.0"
```

```

prefWidth="156.0" promptText="Password" style="-fx-background-color: transparent; -
fx-border-color: blue; -fx-border-width: 0px 0px 2px 0px;" />
    <TextField id="username" fx:id="firstNameField" focusTraversable="false"
layoutX="98.0" layoutY="115.0" onKeyTyped="#firstNameTyped" prefHeight="27.0"
prefWidth="156.0" promptText="First Name" style="-fx-background-color: transparent; -
fx-border-color: BLUE; -fx-border-width: 0px 0px 2px 0px;"
stylesheets="@../src/application/application.css">
        <cursor>
            <Cursor fx:constant="TEXT" />
        </cursor>
    </TextField>
    <TextField id="username" fx:id="lastNameField" focusTraversable="false"
layoutX="98.0" layoutY="175.0" onKeyTyped="#lastNameTyped" prefHeight="27.0"
prefWidth="156.0" promptText="Last Name" style="-fx-background-color: transparent; -
fx-border-color: BLUE; -fx-border-width: 0px 0px 2px 0px;"
stylesheets="@../src/application/application.css">
        <cursor>
            <Cursor fx:constant="TEXT" />
        </cursor>
    </TextField>
    <TextField id="username" fx:id="emailField" focusTraversable="false"
layoutX="98.0" layoutY="235.0" prefHeight="27.0" prefWidth="156.0" promptText="Email"
style="-fx-background-color: transparent; -fx-border-color: BLUE; -fx-border-width:
0px 0px 2px 0px;" stylesheets="@../src/application/application.css">
        <cursor>
            <Cursor fx:constant="TEXT" />
        </cursor>
    </TextField>
    <TextField id="username" fx:id="idField" focusTraversable="false"
layoutX="98.0" layoutY="295.0" onKeyTyped="#idTyped" prefHeight="27.0"
prefWidth="156.0" promptText="I.D" style="-fx-background-color: transparent; -fx-
border-color: BLUE; -fx-border-width: 0px 0px 2px 0px;"
stylesheets="@../src/application/application.css">
        <cursor>
            <Cursor fx:constant="TEXT" />
        </cursor>
    </TextField>
    <CheckBox fx:id="checkbox1" cache="true" focusTraversable="false"
layoutX="98.0" layoutY="480.0" mnemonicParsing="false" prefHeight="20.0"
prefWidth="131.0" stylesheets="@application.css" text="I agree to the tos" />
    <Label fx:id="SignUpbtn1" alignment="CENTER" focusTraversable="false"
layoutX="239.0" layoutY="15.0" onMouseReleased="#loginbtn" prefHeight="28.0"
prefWidth="82.0" style="-fx-background-color: grey; -fx-background-radius: 20px;"
text="Login" textFill="WHITE">
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </padding>
    <Insets left="25.0" />
</padding>
<font>
    <Font name="System Bold" size="12.0" />
</font>
</Label>
<Label alignment="CENTER" focusTraversable="false" layoutX="195.0"
layoutY="15.0" prefHeight="28.0" prefWidth="75.0" style="-fx-background-color:
#0598ff; -fx-background-radius: 20px;" text="Sign up" textFill="WHITE">
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
<font>
    <Font name="System Bold" size="12.0" />
</font>

```

```

        </Label>
        <Label fx:id="errtxt" layoutX="96.0" layoutY="92.0" prefHeight="18.0"
prefWidth="238.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="firstNameLabel" layoutX="97.0" layoutY="158.0"
prefHeight="18.0" prefWidth="237.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="lastNameLabel" layoutX="96.0" layoutY="216.0"
prefHeight="18.0" prefWidth="237.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="emailLabel" layoutX="98.0" layoutY="280.0"
prefHeight="18.0" prefWidth="237.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="idLabel" layoutX="97.0" layoutY="336.0" prefHeight="18.0"
prefWidth="237.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="userNameLabel" layoutX="98.0" layoutY="396.0"
prefHeight="18.0" prefWidth="237.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="passwordLabel" layoutX="98.0" layoutY="453.0"
prefHeight="18.0" prefWidth="237.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
    </children>
    <cursor>
        <Cursor fx:constant="DEFAULT" />
    </cursor>
</AnchorPane>
</right>
<center>
    <AnchorPane id="panelRight" disable="true" prefHeight="560.0" prefWidth="977.0"
style="-fx-background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Label layoutX="51.0" layoutY="328.0" text="Management inventory"
textFill="#130000">
                <font>
                    <Font name="Lucida Calligraphy Italic" size="18.0" />
                </font>
            </Label>
            <Label layoutX="133.0" layoutY="364.0" text="system" textFill="#130000">
                <font>
                    <Font name="Lucida Calligraphy Italic" size="18.0" />
                </font>
            </Label>
        </children>
    </AnchorPane>
</center>

```

```
        </Label>
        <ImageView fitHeight="235.0" fitWidth="297.0" layoutX="29.0"
layoutY="99.0" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="file:images/inventoryImage.png" />
            </image>
        </ImageView>
    </children>
</AnchorPane>
</center>
</BorderPane>
```

## קובץ inventory.fxml

קובץ עיצוב שמכיל את העיצוב של דף המלאי.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.InventoryController">
    <center>
        <VBox alignment="CENTER_LEFT" prefHeight="700.0" prefWidth="770.0"
BorderPane.alignment="CENTER_LEFT">
            <children>
                <Label text="Products List" textFill="#ff0505">
                    <VBox.margin>
                        <Insets left="20.0" top="100.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="24.0" />
                    </font>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="148.0" prefWidth="714.0"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets left="20.0" top="20.0" />
                    </VBox.margin>
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets right="50.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button mnemonicParsing="false" onAction="#removeProduct"
styleClass="removeBtn" stylesheets="@application.css" text="Remove">
                            <HBox.margin>
                                <Insets right="10.0" />
                            </HBox.margin>
                            <cursor>
                                <Cursor fx:constant="HAND" />
                            </cursor>
                            <font>
```

```

        <Font name="System Bold" size="12.0" />
    </font>
</Button>
    <Button focusTraversable="false" mnemonicParsing="false"
onAction="#editProductBtn" styleClass="editBtn" stylesheets="@application.css"
text="Edit" textFill="#171719">
    <HBox.margin>
        <Insets right="10.0" />
    </HBox.margin>
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button mnemonicParsing="false" onAction="#addProductToInventory"
prefHeight="23.0" prefWidth="78.0" styleClass="addBtn" stylesheets="@application.css"
text="Add" HBox.hgrow="ALWAYS">
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
    <HBox.margin>
        <Insets right="20.0" />
    </HBox.margin>
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
</Button>
</children>
</HBox>
    <TableView fx:id="productsTable" cache="true" editable="true"
prefHeight="509.0" prefWidth="669.0" VBox.vgrow="ALWAYS">
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
    <VBox.margin>
        <Insets bottom="100.0" left="20.0" right="20.0" top="5.0" />
    </VBox.margin>
    <columns>
        <TableColumn fx:id="productIdCol" prefWidth="51.199949622154236"
text="Product id" />
        <TableColumn fx:id="productNameCol" minWidth="1.60003662109375"
prefWidth="234.3999786376953" text="Product name" />
        <TableColumn fx:id="categoryCol" minWidth="0.0" prefWidth="0.0"
text="Category" />
        <TableColumn prefWidth="75.0" text="Price">
            <columns>
                <TableColumn fx:id="supplierPriceCol" prefWidth="75.0"
text="Supllier price (₹)" />
                <TableColumn fx:id="priceCol" prefWidth="75.0" text="Client
price (₹)" />
            </columns>
        </TableColumn>
        <TableColumn fx:id="weightCol" prefWidth="75.0" text="Weight (KG)" />
        <TableColumn fx:id="quantityCol" prefWidth="75.0" text="Quantity" />
    </columns>
</TableView>
</children>
<opaqueInsets>
    <Insets />
</opaqueInsets>
</VBox>

```

```

</center>
<left>
    <AnchorPane fx:id="side" prefHeight="667.0" prefWidth="194.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Button layoutY="128.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" stylesheets="@application.css"
text="Inventory">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="163.0" mnemonicParsing="false" onAction="#CategoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Category">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="198.0" mnemonicParsing="false" onAction="#CustomerPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Customer">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="233.0" mnemonicParsing="false" onAction="#orderPageBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Order">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="303.0" mnemonicParsing="false" onAction="#statisticsBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Statistics">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="268.0" mnemonicParsing="false" onAction="#ShelvesBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Shelves">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Logout">
                <font>

```

```

        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button fx:id="expensesBtn" layoutY="334.0" mnemonicParsing="false"
onAction="#expensesBtn" prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Expenses">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
</children>
<BorderPane.margin>
    <Insets />
</BorderPane.margin>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" BorderPane.alignment="CENTER_LEFT">
        <graphic>
            <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</top>
</BorderPane>

```



## קובץ addProduct.fxml

זה קובץ עיצוב שמכיל את הדף של הוספת מוצר חדש.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="450.0" prefWidth="423.0" style="-fx-border-width:
2px 2px 2px 2px; -fx-border-color: black;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.AddProductController">
    <children>
        <Label id="exit" alignment="CENTER" cache="true" layoutX="397.0" layoutY="7.0"
onMouseReleased="#exitBtn" prefHeight="19.0" prefWidth="19.0" style="-fx-background-
radius: 50%; -fx-background-color: #0598ff;" styleClass="exit"
stylesheets="@application.css" text="x" textFill="WHITE">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Label>
        <Label layoutX="135.0" layoutY="17.0" text="Add product">
            <font>
                <Font size="24.0" />
            </font>
        </Label>
        <Button layoutX="362.0" layoutY="417.0" mnemonicParsing="false"
onAction="#addProductToInventory" prefHeight="25.0" prefWidth="53.0"
styleClass="addBtn" stylesheets="@application.css" text="Add">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
        <TextField fx:id="productNameField" layoutX="146.0" layoutY="65.0"
onKeyTyped="#productNameTyped" promptText="Product name" styleClass="fields"
stylesheets="@application.css" />
        <TextField fx:id="productCodeField" layoutX="146.0" layoutY="115.0"
onKeyTyped="#productCodeTyped" promptText="Product code" styleClass="fields"
stylesheets="@application.css" />
        <TextField fx:id="quantityField" layoutX="146.0" layoutY="215.0"
onKeyTyped="#quantityTyped" promptText="Quantity" styleClass="fields"
stylesheets="@application.css" />
        <TextField fx:id="priceField" layoutX="146.0" layoutY="316.0"
promptText="Client price" styleClass="fields" stylesheets="@application.css" />
        <ComboBox fx:id="choiceBoxCategory" layoutX="147.0" layoutY="164.0"
prefWidth="150.0" promptText="Category" styleClass="fields"
stylesheets="@application.css" />
        <TextField fx:id="supplierPrice" layoutX="146.0" layoutY="366.0"
promptText="Supplier price" styleClass="fields" stylesheets="@application.css" />
    </children>
</AnchorPane>
```

```

        <FontAwesomeIcon fill="#7794a7" glyphName="PLUS_CIRCLE" layoutX="277.0"
layoutY="49.0" size="3em" />
        <Label fx:id="productCodeLabel" layoutX="146.0" layoutY="145.0"
prefHeight="18.0" prefWidth="270.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="quantityLabel" layoutX="146.0" layoutY="246.0" prefHeight="18.0"
prefWidth="270.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="clientPriceLabel" layoutX="147.0" layoutY="346.0"
prefHeight="18.0" prefWidth="270.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="supllierPriceLabel" layoutX="146.0" layoutY="398.0"
prefHeight="18.0" prefWidth="270.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <TextField fx:id="weightField" layoutX="146.0" layoutY="265.0"
prefHeight="30.0" prefWidth="153.0" promptText="Weight" styleClass="fields"
stylesheets="@application.css" />
        <Label fx:id="weightLabel" layoutX="146.0" layoutY="296.0" prefHeight="18.0"
prefWidth="270.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
    </children>
</AnchorPane>

```

## קובץ EditProductPage.fxml

זה קובץ עיצוב שמכיל את הדף של עריכת נתונים של המוצר.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="450.0" prefWidth="423.0" style="-fx-border-width:
2px 2px 2px 2px; -fx-border-color: black;" stylesheets="@application.css"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.EditProductController">
    <children>
        <Label id="exit" alignment="CENTER" cache="true" layoutX="396.0" layoutY="8.0"
onMouseReleased="#exitBtn" prefHeight="19.0" prefWidth="19.0" style="-fx-background-
radius: 50%; -fx-background-color: #0598ff;" styleClass="exit"
stylesheets="@application.css" text="x" textFill="WHITE">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Label>
        <Label layoutX="128.0" layoutY="19.0" text="Edit product">
            <font>
                <Font size="24.0" />
            </font>
        </Label>
        <Button focusTraversable="false" layoutX="357.0" layoutY="414.0"
mnemonicParsing="false" onAction="#editProductFromTable" prefHeight="25.0"
prefWidth="58.0" styleClass="addBtn" stylesheets="@application.css" text="Edit">
            <font>
                <Font name="System Bold" size="12.0" />
            </font></Button>
        <TextField fx:id="productNameField" focusTraversable="false" layoutX="175.0"
layoutY="62.0" onKeyTyped="#productNameTyped" promptText="Product name"
styleClass="fields" stylesheets="@application.css" />
        <TextField fx:id="quantityField" focusTraversable="false" layoutX="175.0"
layoutY="212.0" onKeyTyped="#quantityTyped" promptText="Quantity" styleClass="fields"
/>
        <TextField fx:id="priceField" focusTraversable="false" layoutX="175.0"
layoutY="313.0" promptText="Client price" styleClass="fields" />
        <ComboBox fx:id="choiceBoxCategory" focusTraversable="false" layoutX="175.0"
layoutY="162.0" prefHeight="30.0" prefWidth="153.0" promptText="None"
styleClass="fields" />
        <TextField fx:id="productCodeField" disable="true" editable="false"
focusTraversable="false" layoutX="175.0" layoutY="112.0" styleClass="fields"
stylesheets="@application.css" />
        <Label layoutX="85.0" layoutY="68.0" text="Product name : ">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
    </children>
</AnchorPane>
```

```

<Label layoutX="85.0" layoutY="116.0" text="Product code :">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label layoutX="85.0" layoutY="164.0" text="Category :">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label layoutX="86.0" layoutY="217.0" text="Quantity :">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label layoutX="86.0" layoutY="317.0" text="Client price :">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label layoutX="86.0" layoutY="369.0" text="Supplier price :">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<TextField fx:id="supllierPriceField" focusTraversable="false" layoutX="174.0"
layoutY="363.0" prefHeight="30.0" prefWidth="153.0" promptText="Supplier price"
styleClass="fields" />
<FontAwesomeIcon glyphName="EDIT" layoutX="273.0" layoutY="48.0" size="2.3em" />
<Label fx:id="supplierPriceLabel" layoutX="171.0" layoutY="396.0"
prefHeight="7.0" prefWidth="242.0" textFill="RED">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label fx:id="clientPriceLabel" layoutX="176.0" layoutY="343.0"
prefHeight="18.0" prefWidth="242.0" textFill="RED">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label fx:id="quantityLabel" layoutX="176.0" layoutY="242.0" prefHeight="18.0"
prefWidth="242.0" textFill="RED">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<TextField fx:id="weightField" focusTraversable="false" layoutX="174.0"
layoutY="262.0" prefHeight="30.0" prefWidth="154.0" promptText="Weight"
styleClass="fields" stylesheets="@application.css" />
<Label fx:id="weightLabel" layoutX="176.0" layoutY="294.0" prefHeight="18.0"
prefWidth="242.0" textFill="RED">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
<Label layoutX="85.0" layoutY="268.0" text="Weight :">
  <font>
    <Font name="System Bold" size="12.0" />
  </font>
</Label>
</children>
</AnchorPane>

```

## קובץ CategoryPage.fxml

זה קובץ עיצוב שמכיל את הדף של ניהול הקטגוריות.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.CategoryController">
    <center>
        <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
            <children>
                <VBox alignment="CENTER_LEFT" prefHeight="700.0" prefWidth="770.0"
VBox.vgrow="ALWAYS">
                    <children>
                        <Label text="Categories List" textFill="#ff0505"
VBox.vgrow="ALWAYS">
                            <VBox.margin>
                                <Insets left="90.0" top="100.0" />
                            </VBox.margin>
                            <font>
                                <Font name="System Bold" size="24.0" />
                            </font>
                        </Label>
                        <HBox alignment="TOP_RIGHT" prefHeight="26.0" prefWidth="590.0"
VBox.vgrow="ALWAYS">
                            <VBox.margin>
                                <Insets left="70.0" top="20.0" />
                            </VBox.margin>
                            <children>
                                <HBox prefHeight="26.0" prefWidth="445.0"
HBox.hgrow="ALWAYS">
                                    <children>
                                        <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                            <HBox.margin>
                                                <Insets left="20.0" />
                                            </HBox.margin>
                                        </TextField>
                                    </children>
                                </HBox>
                                <Button focusTraversable="false" mnemonicParsing="false"
onAction="#removeCategoryBtn" styleClass="removeBtn" stylesheets="@application.css"
text="Remove">
                                    <HBox.margin>
```

```

        <Insets right="10.0" />
    </HBox.margin>
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button focusTraversable="false" mnemonicParsing="false"
onAction="#editCategoryBtn" prefHeight="26.0" prefWidth="40.0" styleClass="editBtn"
stylesheets="@application.css" text="Edit" HBox.hgrow="ALWAYS">
    <HBox.margin>
        <Insets right="10.0" />
    </HBox.margin>
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button focusTraversable="false" mnemonicParsing="false"
onAction="#addProductToInventory" prefHeight="44.0" prefWidth="42.0"
styleClass="addBtn" stylesheets="@application.css" text="Add" HBox.hgrow="ALWAYS">
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
    <HBox.margin>
        <Insets right="90.0" />
    </HBox.margin>
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
</Button>
</children>
</HBox>
    <TableView fx:id="categoryTable" cache="true" editable="true"
prefHeight="509.0" prefWidth="674.0" VBox.vgrow="ALWAYS">
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
    <VBox.margin>
        <Insets bottom="100.0" left="90.0" right="90.0" top="5.0" />
    </VBox.margin>
    <columns>
        <TableColumn fx:id="categoryIdCol" prefWidth="75.0"
sortable="false" text="Category id" />
        <TableColumn fx:id="categoryName" prefWidth="75.0"
sortable="false" text="Category name" />
    </columns>
</TableView>
</children>
<opaqueInsets>
    <Insets />
</opaqueInsets>
<VBox.margin>
    <Insets />
</VBox.margin>
</VBox>
</children>
</VBox>
</center>

```

```

<left>
  <AnchorPane fx:id="side" prefHeight="226.0" prefWidth="216.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
    <children>
      <Button layoutY="128.0" mnemonicParsing="false" onAction="#inventoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Inventory">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="163.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" stylesheets="@application.css"
text="Category">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="198.0" mnemonicParsing="false" onAction="#CustomerPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Customer">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="233.0" mnemonicParsing="false" onAction="#orderPageBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Order">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="303.0" mnemonicParsing="false" onAction="#statisticsBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Statistics">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="268.0" mnemonicParsing="false" onAction="#shelvesBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Shelves">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
    </children>
  </AnchorPane>

```

```

        </Button>
        <Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Logout">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
        <Button fx:id="expensesBtn" layoutY="334.0" mnemonicParsing="false"
onAction="#expensesBtn" prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Expenses">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
    </children>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" BorderPane.alignment="CENTER_LEFT">
        <graphic>
            <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</top>
</BorderPane>

```



## קובץ addCategoryPage.fxml

זה קובץ עיצוב שמכיל את הדף של הוספת קטגוריה חדשה.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.DialogPane?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="185.0" prefWidth="423.0" style="-fx-border-width: 2px 2px 2px
2px; -fx-border-color: black;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.AddCategoryPageController">
    <children>
        <Label id="exit" alignment="CENTER" cache="true" layoutX="397.0" layoutY="6.0"
onMouseReleased="#exitBtn" prefHeight="19.0" prefWidth="19.0" style="-fx-background-
radius: 50%; -fx-background-color: #0598ff;" styleClass="exit"
stylesheets="@application.css" text="x" textFill="WHITE">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Label>
        <Label layoutX="121.0" layoutY="42.0" text="Add Category">
            <font>
                <Font size="24.0" />
            </font>
        </Label>
        <Button layoutX="354.0" layoutY="146.0" mnemonicParsing="false"
onAction="#addNewCategoryButton" prefHeight="25.0" prefWidth="53.0"
styleClass="addBtn" stylesheets="@application.css" text="Add">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
        <TextField fx:id="categoryNameField" layoutX="147.0" layoutY="93.0"
onKeyTyped="#categoryNameTyped" promptText="Category name" />
        <DialogPane layoutX="-57.0" layoutY="-349.0" maxHeight="-Infinity" maxWidth="-
Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="300.0"
prefWidth="480.0">
            <expandableContent>
                <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="200.0"
prefWidth="320.0" />
            </expandableContent>
        </DialogPane>
        <FontAwesomeIcon fill="#7794a7" glyphName="PLUS_CIRCLE" layoutX="280.0"
layoutY="75.0" size="3em" />
    </children>
</Pane>
```

## קובץ EditCategoryPage.fxml

זה קובץ עיצוב שמכיל את הדף של עריכת קטגוריה.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="185.0" prefWidth="423.0" style="-fx-border-width: 2px 2px 2px
2px; -fx-border-color: black;" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.EditCategoryPageController">
    <children>
        <Label id="exit" alignment="CENTER" cache="true" layoutX="397.0" layoutY="6.0"
onMouseReleased="#exitBtn" prefHeight="19.0" prefWidth="19.0" style="-fx-background-
radius: 50%; -fx-background-color: #0598ff;" styleClass="exit"
stylesheets="@application.css" text="x" textFill="WHITE">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Label>
        <Label layoutX="106.0" layoutY="35.0" text="Edit Category">
            <font>
                <Font size="24.0" />
            </font>
        </Label>
        <Button layoutX="344.0" layoutY="146.0" mnemonicParsing="false"
onAction="#updateCategoryBtn" prefHeight="25.0" prefWidth="63.0" styleClass="addBtn"
stylesheets="@application.css" text="Edit">
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
            <font>
                <Font name="System Bold" size="12.0" />
            </font></Button>
        <TextField fx:id="categoryNameField" focusTraversable="false" layoutX="167.0"
layoutY="113.0" onKeyTyped="#categoryNameTyped" promptText="Category name" />
        <TextField fx:id="categoryIdField" disable="true" editable="false"
focusTraversable="false" layoutX="167.0" layoutY="80.0" />
        <Label layoutX="70.0" layoutY="84.0" text="Category id :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label layoutX="70.0" layoutY="115.0" text="Category name :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <FontAwesomeIcon glyphName="EDIT" layoutX="269.0" layoutY="65.0" size="2.3em" />
    </children>
</Pane>
```

## קובץ CustomerPage.fxml

זה קובץ עיצוב שמכיל את הדף של הוספת לקוח חדש למערכת.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.CustomerController">
    <center>
        <VBox alignment="CENTER" cache="true" cacheHint="QUALITY" fillWidth="false"
prefHeight="552.0" prefWidth="784.0" BorderPane.alignment="CENTER">
            <children>
                <VBox alignment="TOP_CENTER" prefHeight="700.0" prefWidth="384.0"
style="-fx-background-color: white; -fx-background-radius: 10px;"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="40.0" left="200.0" right="200.0" top="90.0" />
                    </VBox.margin>
                    <effect>
                        <DropShadow />
                    </effect>
                    <children>
                        <Label alignment="TOP_CENTER" contentDisplay="CENTER"
text="Customer details" VBox.vgrow="ALWAYS">
                            <opaqueInsets>
                                <Insets />
                            </opaqueInsets>
                            <font>
                                <Font name="System Bold" size="18.0" />
                            </font>
                            <VBox.margin>
                                <Insets top="40.0" />
                            </VBox.margin>
                        </Label>
                        <TextField fx:id="fullnamefield" onKeyTyped="#fullNameTyped"
prefWidth="200.0" promptText="Full name" style="-fx-border-width: 2px 2px 2px 2px; -
fx-border-color: black; -fx-background-color: transparent; -fx-border-radius: 10px;"
VBox.vgrow="NEVER">
                            <VBox.margin>
                                <Insets left="75.0" right="75.0" top="35.0" />
                            </VBox.margin>
                        </TextField>
                        <Label fx:id="fullNameLabel" prefHeight="18.0" prefWidth="391.0"
textFill="RED">
                            <VBox.margin>
                                <Insets left="80.0" top="10.0" />
                            </VBox.margin>
                        </Label>
                    </children>
                </VBox>
            </children>
        </VBox>
    </center>
</BorderPane>
```

```

        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Label>
    <TextField fx:id="idfield" onKeyTyped="#idTyped" prefWidth="200.0"
promptText="I.D" style="-fx-border-width: 2px 2px 2px 2px; -fx-border-color: black; -
fx-background-color: transparent; -fx-border-radius: 10px;" VBox.vgrow="NEVER">
        <VBox.margin>
            <Insets left="75.0" right="75.0" top="10.0" />
        </VBox.margin>
    </TextField>
    <Label fx:id="idLabel" prefHeight="18.0" prefWidth="391.0"
textFill="RED">
        <VBox.margin>
            <Insets left="80.0" top="10.0" />
        </VBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Label>
    <TextField fx:id="telephonefield" onKeyTyped="#telephoneTyped"
prefWidth="200.0" promptText="Telephone" style="-fx-border-width: 2px 2px 2px 2px; -
fx-border-color: black; -fx-background-color: transparent; -fx-border-radius: 10px;"
VBox.vgrow="NEVER">
        <VBox.margin>
            <Insets left="75.0" right="75.0" top="10.0" />
        </VBox.margin>
    </TextField>
    <Label fx:id="telephoneLabel" prefHeight="18.0" prefWidth="391.0"
textFill="RED">
        <VBox.margin>
            <Insets left="80.0" top="10.0" />
        </VBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Label>
    <TextField fx:id="emailfield" prefWidth="200.0" promptText="Email"
style="-fx-border-width: 2px 2px 2px 2px; -fx-border-color: black; -fx-background-
color: transparent; -fx-border-radius: 10px;" VBox.vgrow="NEVER">
        <VBox.margin>
            <Insets left="75.0" right="75.0" top="10.0" />
        </VBox.margin>
    </TextField>
    <Label fx:id="emailLabel" prefHeight="18.0" prefWidth="391.0"
textFill="RED">
        <VBox.margin>
            <Insets left="80.0" top="10.0" />
        </VBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Label>
    <TextField fx:id="addressfield" onKeyTyped="#addressTyped"
prefWidth="200.0" promptText="Address" style="-fx-border-width: 2px 2px 2px 2px; -fx-
border-color: black; -fx-background-color: transparent; -fx-border-radius: 10px;"
VBox.vgrow="NEVER">
        <VBox.margin>
            <Insets left="75.0" right="75.0" top="10.0" />
        </VBox.margin>
    </TextField>
    <Label fx:id="addressLabel" prefHeight="18.0" prefWidth="391.0"
textFill="RED">

```

```

        <VBox.margin>
            <Insets left="80.0" top="10.0" />
        </VBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Label>
    <HBox alignment="CENTER" prefHeight="32.0" prefWidth="409.0">
        <children>
            <Button focusTraversable="false" mnemonicParsing="false"
onAction="#addCustomerBtn" styleClass="addBtn" stylesheets="@application.css"
text="Add">

                <padding>
                    <Insets bottom="5.0" left="20.0" right="20.0" top="5.0" />
                </padding>
                <HBox.margin>
                    <Insets right="15.0" />
                </HBox.margin>
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button mnemonicParsing="false" onAction="#clearBtn"
styleClass="removeBtn" stylesheets="@application.css" text="Clear">
                <padding>
                    <Insets bottom="5.0" left="20.0" right="20.0" top="5.0" />
                </padding>
                <HBox.margin>
                    <Insets right="15.0" />
                </HBox.margin>
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
        </children>
    </HBox>
    <VBox.margin>
        <Insets top="10.0" />
    </VBox.margin>
</HBox>
<HBox alignment="CENTER" prefHeight="85.0" prefWidth="542.0">
    <VBox.margin>
        <Insets top="10.0" />
    </VBox.margin>
    <children>
        <Label onMouseReleased="#CustomerListPage"
styleClass="CustomerListLabel" stylesheets="@application.css" text="Customer List"
underline="true">

            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Label>
    </children>
</HBox>
</children>
</VBox>
</children>
</VBox>
</center>

```

```

<left>
  <AnchorPane fx:id="side" prefHeight="667.0" prefWidth="208.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
    <children>
      <Button layoutY="128.0" mnemonicParsing="false" onAction="#inventoryPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Inventory">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="163.0" mnemonicParsing="false" onAction="#CategoryPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Category">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="198.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" stylesheets="@application.css"
text="Customer">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="233.0" mnemonicParsing="false" onAction="#orderPageBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Order">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="303.0" mnemonicParsing="false" onAction="#statisticsBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Statistics">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
      <Button layoutY="268.0" mnemonicParsing="false" onAction="#shelvesBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Shelves">
        <font>
          <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
          <Cursor fx:constant="HAND" />
        </cursor>
      </Button>
    </children>
  </AnchorPane>

```

```

        </Button>
        <Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Logout">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
        <Button fx:id="expensesBtn" layoutY="334.0" mnemonicParsing="false"
onAction="#expensesBtn" prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Expenses">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
    </children>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" stylesheets="@application.css"
BorderPane.alignment="CENTER_LEFT">
        <graphic>
            <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</top>
</BorderPane>

```

## קובץ CustomerList.fxml

זה קובץ עיצוב שמכיל את דף של ניהול הלקוחות.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.CustomerListController">
    <center>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Customer List" textFill="RED" VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="14.0" prefWidth="1000.0">
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField" stylesheets="@application.css">
                                    <HBox.margin>
                                        <Insets left="40.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button mnemonicParsing="false" onAction="#removeCustomerFromList"
styleClass="removeBtn" stylesheets="@application.css" text="Remove">
                                    <HBox.margin>
                                        <Insets right="10.0" />
                                    </HBox.margin>
                                    <font>
                                        <Font name="System Bold" size="12.0" />
                                    </font>
                                    <cursor>
                                        <Cursor fx:constant="HAND" />
                                    </cursor>
                                </Button>
                                <Button mnemonicParsing="false" onAction="#editCustomerBtn"
styleClass="editBtn" stylesheets="@application.css" text="Edit" textFill="#171719">
```



```

        <HBox.margin>
            <Insets right="40.0" />
        </HBox.margin>
        <opaqueInsets>
            <Insets />
        </opaqueInsets>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Button>
</children>
<VBox.margin>
    <Insets bottom="5.0" />
</VBox.margin>
</HBox>
<TableView fx:id="customerTable" prefHeight="200.0" prefWidth="200.0"
stylesheets="@application.css" VBox.vgrow="ALWAYS">
    <columns>
        <TableColumn fx:id="fnamecol" prefWidth="75.0" text="Full name" />
        <TableColumn fx:id="idcol" prefWidth="75.0" text="I.D" />
        <TableColumn fx:id="emailcol" prefWidth="75.0" text="Email" />
        <TableColumn fx:id="telcol" prefWidth="75.0" text="Telephone" />
        <TableColumn fx:id="addresscol" prefWidth="75.0" text="Address" />
    </columns>
    <VBox.margin>
        <Insets bottom="20.0" left="40.0" right="40.0" />
    </VBox.margin>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
</TableView>
<HBox>
    <children>
        <Button mnemonicParsing="false" onAction="#backToCustomerPage"
styleClass="addBtn" text="Back">
            <HBox.margin>
                <Insets left="40.0" />
            </HBox.margin>
            <graphic>
                <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
            </graphic>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Button>
    </children>
    <VBox.margin>
        <Insets bottom="20.0" />
    </VBox.margin>
</HBox>
</children>
</VBox>
</center>
</BorderPane>

```

## קובץ EditCustomerPage.fxml

זה קובץ עיצוב שמכיל את דף של עריכת פרטים אישיים של הלקוח.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>

<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="380.0" prefWidth="491.0" style="-fx-border-width: 2px 2px 2px
2px; -fx-border-color: black;" stylesheets="@application.css"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.EditCustomerPageController">
    <children>
        <Label id="exit" alignment="CENTER" cache="true" layoutX="464.0" layoutY="8.0"
onMouseReleased="#exitBtn" prefHeight="19.0" prefWidth="19.0" style="-fx-background-
radius: 50%; -fx-background-color: #0598ff;" styleClass="exit"
stylesheets="@application.css" text="x" textFill="WHITE">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Label>
        <Label layoutX="133.0" layoutY="19.0" text="Edit Customer Details">
            <font>
                <Font size="24.0" />
            </font>
        </Label>
        <Button layoutX="421.0" layoutY="342.0" mnemonicParsing="false"
onAction="#updateCustomerBtn" prefHeight="25.0" prefWidth="59.0" styleClass="addBtn"
stylesheets="@application.css" text="Edit">
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
            <font>
                <Font name="System Bold" size="12.0" />
            </font></Button>
        <TextField fx:id="fullNameField" focusTraversable="false" layoutX="205.0"
layoutY="73.0" prefHeight="25.0" prefWidth="190.0" promptText="Full name"
styleClass="fields" />
        <TextField fx:id="idfield" disable="true" editable="false"
focusTraversable="false" layoutX="205.0" layoutY="128.0" prefHeight="25.0"
prefWidth="190.0" styleClass="fields" />
        <Label layoutX="139.0" layoutY="79.0" text="Full name :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label layoutX="139.0" layoutY="135.0" text="I.D :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
```

```

        <TextField fx:id="phoneField" focusTraversable="false" layoutX="207.0"
layoutY="233.0" prefHeight="29.0" prefWidth="190.0" promptText="Phone"
styleClass="fields" />
        <TextField fx:id="emailField" focusTraversable="false" layoutX="207.0"
layoutY="180.0" prefHeight="29.0" prefWidth="190.0" promptText="Email"
styleClass="fields" />
        <Label focusTraversable="false" layoutX="139.0" layoutY="187.0" text="Email :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <TextField fx:id="addressField" focusTraversable="false" layoutX="207.0"
layoutY="288.0" prefHeight="29.0" prefWidth="190.0" promptText="Address"
styleClass="fields" />
        <Label layoutX="139.0" layoutY="240.0" text="Phone :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label layoutX="139.0" layoutY="294.0" text="Address :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <FontAwesomeIcon glyphName="EDIT" layoutX="373.0" layoutY="49.0" size="2.3em"
/>
        <Label fx:id="fullNameLabel" layoutX="205.0" layoutY="106.0" prefHeight="18.0"
prefWidth="280.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="idLabel" layoutX="205.0" layoutY="159.0" prefHeight="18.0"
prefWidth="280.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="emailLabel" layoutX="205.0" layoutY="212.0" prefHeight="18.0"
prefWidth="280.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="phoneLabel" layoutX="207.0" layoutY="266.0" prefHeight="18.0"
prefWidth="280.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <Label fx:id="addressLabel" layoutX="207.0" layoutY="321.0" prefHeight="18.0"
prefWidth="280.0" textFill="RED">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
    </children>
</Pane>

```

## קובץ OrderPage.fxml

זה קובץ עיצוב שמכיל את דף של הוספת מוצרים לסל הקניות.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ButtonBar?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.orderController">
    <center>
        <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
            <children>
                <HBox prefHeight="48.0" prefWidth="784.0">
                    <children>
                        <Label text="Customers List">
                            <font>
                                <Font name="System Bold" size="18.0" />
                            </font>
                            <HBox.margin>
                                <Insets left="70.0" />
                            </HBox.margin>
                        </Label>
                        <HBox alignment="CENTER_RIGHT" prefHeight="100.0" prefWidth="200.0"
HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchCustomerField"
focusTraversable="false" onKeyTyped="#searchCustomerFieldTyped" prefHeight="26.0"
prefWidth="233.0" promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets right="70.0" top="10.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                    </children>
                    <VBox.margin>
                        <Insets top="30.0" />
                    </VBox.margin>
                </HBox>
                <TableView fx:id="customerTable" prefHeight="200.0" prefWidth="200.0">
                    <columns>
                        <TableColumn fx:id="fnamecol" prefWidth="75.0" text="Full name" />
                        <TableColumn fx:id="idcol" prefWidth="75.0" text="I.D" />
                        <TableColumn fx:id="emailcol" prefWidth="75.0" text="Email" />
                        <TableColumn fx:id="telcol" prefWidth="75.0" text="Telephone" />
                    </columns>
                </TableView>
            </children>
        </VBox>
    </center>
</BorderPane>
```

```

        <TableColumn fx:id="addresscol" prefWidth="75.0" text="Address" />
    </columns>
    <VBox.margin>
        <Insets left="70.0" right="70.0" />
    </VBox.margin>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
</TableView>
<ButtonBar prefHeight="11.0" prefWidth="784.0">
    <buttons>
        <Button alignment="CENTER" focusTraversable="false" minWidth="0.0"
mnemonicParsing="false" onAction="#chooseCustomerFromTableBtn" prefHeight="26.0"
styleClass="addBtn" text="Add">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Button>
    </buttons>
    <VBox.margin>
        <Insets bottom="10.0" right="70.0" top="10.0" />
    </VBox.margin>
</ButtonBar>
<HBox alignment="CENTER">
    <children>
        <Label text="Customer name :" HBox.hgrow="ALWAYS">
            <HBox.margin>
                <Insets top="3.0" />
            </HBox.margin>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <TextField fx:id="customernamefield" disable="true">
            <HBox.margin>
                <Insets left="5.0" />
            </HBox.margin>
        </TextField>
        <Label text="I.D : " HBox.hgrow="ALWAYS">
            <HBox.margin>
                <Insets left="20.0" top="3.0" />
            </HBox.margin>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
        </Label>
        <TextField fx:id="customeridfield" disable="true">
            <HBox.margin>
                <Insets left="5.0" />
            </HBox.margin>
        </TextField>
    </children>
</HBox>
<HBox prefHeight="48.0" prefWidth="784.0">
    <children>
        <Label text="Products List">
            <font>
                <Font name="System Bold" size="18.0" />
            </font>
            <HBox.margin>
                <Insets left="70.0" />
            </HBox.margin>
        </Label>
    </children>
</HBox>

```

```

        <HBox alignment="CENTER_RIGHT" prefHeight="100.0" prefWidth="200.0"
HBox.hgrow="ALWAYS">
            <children>
                <TextField fx:id="searchProductField"
focusTraversable="false" onKeyTyped="#searchProductFieldTyped" prefHeight="26.0"
prefWidth="233.0" promptText="Search" styleClass="searchField">
                    <HBox.margin>
                        <Insets right="70.0" top="10.0" />
                    </HBox.margin>
                </TextField>
            </children>
        </HBox>
    </children>
    <VBox.margin>
        <Insets top="5.0" />
    </VBox.margin>
</HBox>
<TableView fx:id="productsTable" prefHeight="208.0" prefWidth="643.0"
VBox.vgrow="ALWAYS">
    <columns>
        <TableColumn fx:id="productIdCol" prefWidth="75.0" text="Product id" />
        <TableColumn fx:id="shelfIdCol" prefWidth="75.0" text="Shelf id" />
        <TableColumn fx:id="productNameCol" prefWidth="75.0" text="Product name" />
        <TableColumn fx:id="categoryCol" prefWidth="75.0" text="Category" />
        <TableColumn fx:id="priceCol" prefWidth="75.0" text="Price" />
        <TableColumn fx:id="quantityCol" prefWidth="75.0" text="Quantity" />
    </columns>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
    <VBox.margin>
        <Insets bottom="10.0" left="70.0" right="70.0" />
    </VBox.margin>
</TableView>
    <HBox alignment="TOP_RIGHT" prefHeight="26.0" prefWidth="784.0">
        <children>
            <Label alignment="CENTER" onMouseReleased="#orderListPage"
prefHeight="26.0" prefWidth="94.0" styleClass="addBtn" text="Orders">
                <font>
                    <Font name="System Bold" size="13.0" />
                </font>
                <HBox.margin>
                    <Insets right="10.0" />
                </HBox.margin>
            </Label>
            <Label fx:id="shoppingCartLabel" alignment="CENTER"
onMouseReleased="#shoppingCartBtn" prefHeight="26.0" prefWidth="94.0"
styleClass="addBtn" text="Shopping cart">
                <font>
                    <Font name="System Bold" size="13.0" />
                </font>
                <HBox.margin>
                    <Insets right="10.0" />
                </HBox.margin>
            </Label>
            <Label fx:id="addLabels" alignment="CENTER"
onMouseReleased="#addProductToShoppingCart" prefHeight="26.0" prefWidth="86.0"
styleClass="addBtn" text="Add">
                <HBox.margin>
                    <Insets right="70.0" />
                </HBox.margin>
                <font>
                    <Font name="System Bold" size="14.0" />
                </font>
            </Label>
        </children>
    </HBox>

```

```

        </font>
        <graphic>
            <Label>
                <graphic>
                    <FontAwesomeIcon fill="WHITE" glyphName="SHOPPING_CART" />
                </graphic>
                <padding>
                    <Insets right="10.0" />
                </padding>
            </Label>
        </graphic>
    </Label>
</children>
<VBox.margin>
    <Insets bottom="20.0" />
</VBox.margin>
</HBox>
</children>
</VBox>
</center>
<left>
    <AnchorPane fx:id="side" prefHeight="678.0" prefWidth="216.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Button layoutY="128.0" mnemonicParsing="false"
onAction="#inventoryPageBtn" prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Inventory">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="163.0" mnemonicParsing="false" onAction="#CategoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Category">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="198.0" mnemonicParsing="false" onAction="#CustomerPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Customer">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="233.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" text="Order">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="303.0" mnemonicParsing="false" onAction="#statisticsBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Statistics">

```

```

        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
    <Button layoutY="268.0" mnemonicParsing="false" onAction="#shelvesBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Shelves">
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
    <Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Logout">
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
    <Button fx:id="expensesBtn" layoutY="338.0" mnemonicParsing="false"
onAction="#expensesBtn" prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Expenses">
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
</children>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" stylesheets="@application.css"
BorderPane.alignment="CENTER_LEFT">
        <graphic>
            <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</top>
</BorderPane>

```



## קובץ Shopping\_cart.fxml

זה קובץ עיצוב שמכיל את דף של סל הקניות.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.ShoppingCartController">
    <bottom>
        <VBox alignment="CENTER_LEFT" prefHeight="700.0" prefWidth="770.0"
stylesheets="@application.css" BorderPane.alignment="CENTER">
            <children>
                <Label text="Shopping cart" textFill="#f21a1a">
                    <font>
                        <Font name="System Bold" size="20.0" />
                    </font>
                    <VBox.margin>
                        <Insets bottom="10.0" left="70.0" top="50.0" />
                    </VBox.margin>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="20.0" prefWidth="1000.0">
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets left="70.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button focusTraversable="false" mnemonicParsing="false"
onAction="#removeProductFromShopingCartList" styleClass="removeBtn"
stylesheets="@application.css" text="Remove">
                                    <HBox.margin>
                                        <Insets right="10.0" />
                                    </HBox.margin>
                                    <cursor>
                                        <Cursor fx:constant="HAND" />
                                    </cursor>
                                    <font>
                                        <Font name="System Bold" size="12.0" />
                                    </font>
                                </Button>
                            </children>
                        </HBox>
                    </children>
                </HBox>
            </children>
        </VBox>
    </bottom>
</BorderPane>
```

```

        </Button>
        <Button mnemonicParsing="false"
onAction="#editQuantityForProductBtn" styleClass="editBtn"
stylesheets="@application.css" text="Edit" textFill="#171719">
            <HBox.margin>
                <Insets right="70.0" />
            </HBox.margin>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
    </children>
    <VBox.margin>
        <Insets bottom="10.0" />
    </VBox.margin>
</HBox>
<TableView fx:id="shoppingCartTable" cache="true" editable="true"
prefHeight="509.0" prefWidth="674.0" VBox.vgrow="ALWAYS">
    <VBox.margin>
        <Insets bottom="5.0" left="70.0" right="70.0" top="5.0" />
    </VBox.margin>
    <columns>
        <TableColumn fx:id="productIdCol" prefWidth="75.0" sortable="false"
text="Product id" />
        <TableColumn fx:id="shelfNumberCol" prefWidth="75.0" text="Shelf
number" />
        <TableColumn fx:id="productNameCol" prefWidth="75.0"
sortable="false" text="Product name" />
        <TableColumn fx:id="quantityCol" prefWidth="75.0" sortable="false"
text="Quantity" />
        <TableColumn fx:id="priceCol" prefWidth="75.0" sortable="false"
text="Price" />
    </columns>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
</TableView>
<HBox nodeOrientation="RIGHT_TO_LEFT" prefWidth="722.0"
VBox.vgrow="ALWAYS">
    <children>
        <Label fx:id="totalPriceLabel" text="0">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <HBox.margin>
                <Insets />
            </HBox.margin>
        </Label>
        <Label text="Total price :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <HBox.margin>
                <Insets left="2.0" top="0.5" />
            </HBox.margin>
        </Label>
    </children>
    <VBox.margin>
        <Insets right="70.0" />
    </VBox.margin>

```

```

</HBox>
<HBox nodeOrientation="RIGHT_TO_LEFT">
  <children>
    <Button mnemonicParsing="false" onAction="#addOrderToOrderList"
prefHeight="26.0" prefWidth="123.0" styleClass="addBtn" text="Place order">
      <HBox.margin>
        <Insets left="70.0" />
      </HBox.margin>
      <cursor>
        <Cursor fx:constant="HAND" />
      </cursor>
      <font>
        <Font name="System Bold" size="12.0" />
      </font>
    </Button>
    <HBox alignment="CENTER_LEFT" nodeOrientation="LEFT_TO_RIGHT"
prefHeight="0.0" prefWidth="891.0" HBox.hgrow="ALWAYS">
      <children>
        <Button mnemonicParsing="false"
onAction="#BackToOrderPageBtn" prefHeight="26.0" prefWidth="61.0" styleClass="addBtn"
text="Back">
          <graphic>
            <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
          </graphic>
          <font>
            <Font name="System Bold" size="12.0" />
          </font>
          <HBox.margin>
            <Insets left="70.0" />
          </HBox.margin>
        </Button>
      </children>
    </HBox>
  </children>
  <VBox.margin>
    <Insets top="5.0" />
  </VBox.margin>
</HBox>
<HBox>
  <VBox.margin>
    <Insets bottom="30.0" left="70.0" top="10.0" />
  </VBox.margin>
</HBox>
</children>
<opaqueInsets>
  <Insets />
</opaqueInsets>
<BorderPane.margin>
  <Insets />
</BorderPane.margin>
</VBox>
</bottom>
</BorderPane>

```

## קובץ OrderListPage.fxml

זה קובץ עיצוב שמכיל את דף של ניהול הזמנות.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.OrderListController">
    <top>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Orders List" textFill="RED" VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="18.0" prefWidth="1000.0">
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets left="40.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button mnemonicParsing="false" onAction="#deleteOrderBtn"
styleClass="removeBtn" stylesheets="@application.css" text="Remove">
                                    <HBox.margin>
                                        <Insets right="10.0" />
                                    </HBox.margin>
                                    <cursor>
                                        <Cursor fx:constant="HAND" />
                                    </cursor>
                                    <font>
                                        <Font name="System Bold" size="12.0" />
                                    </font>
                                </Button>
                                <Button mnemonicParsing="false" onAction="#viewOrderBtn"
prefHeight="23.0" prefWidth="78.0" styleClass="addBtn" stylesheets="@application.css"
text="View order" HBox.hgrow="ALWAYS">
                                    </Button>
                                </children>
                            </HBox>
                        </children>
                    </HBox>
                </Label>
            </children>
        </VBox>
    </top>
</BorderPane>
```

```

        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
        <HBox.margin>
            <Insets right="40.0" />
        </HBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Button>
</children>
<VBox.margin>
    <Insets bottom="10.0" top="20.0" />
</VBox.margin>
</HBox>
<TableView fx:id="orderTable" focusTraversable="false" prefHeight="200.0"
prefWidth="200.0" stylesheets="@application.css" VBox.vgrow="ALWAYS">
    <columns>
        <TableColumn fx:id="orderIdCol" prefWidth="75.0" text="Order id" />
        <TableColumn fx:id="customerIdCol" prefWidth="75.0" text="Customer
id" />
        <TableColumn fx:id="totalPriceCol" prefWidth="75.0" text="Total
price" />
        <TableColumn fx:id="orderDateCol" prefWidth="75.0" text="Order
date" />
    </columns>
    <VBox.margin>
        <Insets bottom="20.0" left="40.0" right="40.0" />
    </VBox.margin>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
</TableView>
<HBox>
    <children>
        <Button mnemonicParsing="false" onAction="#backToOrderPage"
styleClass="addBtn" text="Back">
            <HBox.margin>
                <Insets left="40.0" />
            </HBox.margin>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <graphic>
                <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
            </graphic>
        </Button>
    </children>
    <VBox.margin>
        <Insets bottom="20.0" />
    </VBox.margin>
</HBox>
</children>
</VBox>
</top>
</BorderPane>

```

## קובץ viewOrder.fxml

זה קובץ עיצוב שמכיל את דף של הצגת המוצרים בתוך ההזמנה.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.viewOrderController">
    <top>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="View order" textFill="RED" VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="10.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="0.0" prefWidth="1000.0">
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets left="40.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button focusTraversable="false" mnemonicParsing="false"
onAction="#deleteProduct" styleClass="removeBtn" stylesheets="@application.css"
text="Remove">
                            <HBox.margin>
                                <Insets right="10.0" />
                            </HBox.margin>
                            <cursor>
                                <Cursor fx:constant="HAND" />
                            </cursor>
                            <font>
                                <Font name="System Bold" size="12.0" />
                            </font>
                        </Button>
                    </children>
                </HBox>
            </children>
        </VBox>
    </top>
</BorderPane>
```

```

        <Button mnemonicParsing="false" onAction="#editBtn"
styleClass="editBtn" stylesheets="@application.css" text="Edit" textFill="#171719">
        <HBox.margin>
            <Insets right="40.0" />
        </HBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
</children>
<VBox.margin>
    <Insets bottom="10.0" />
</VBox.margin>
</HBox>
<TableView fx:id="viewOrderTable" focusTraversable="false"
prefHeight="200.0" prefWidth="200.0" stylesheets="@application.css"
VBox.vgrow="ALWAYS">
    <columns>
        <TableColumn fx:id="orderIdCol" prefWidth="75.0" sortable="false"
text="Order id" />
        <TableColumn fx:id="shelfNumberCol" prefWidth="75.0"
sortable="false" text="Shelf number" />
        <TableColumn fx:id="productIdCol" prefWidth="75.0" sortable="false"
text="Product id" />
        <TableColumn fx:id="productNameCol" prefWidth="75.0"
sortable="false" text="Product name" />
        <TableColumn fx:id="quantityCol" prefWidth="75.0" sortable="false"
text="Quantity" />
        <TableColumn fx:id="priceCol" prefWidth="75.0" sortable="false"
text="Price" />
    </columns>
    <VBox.margin>
        <Insets bottom="5.0" left="40.0" right="40.0" />
    </VBox.margin>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
</TableView>
<HBox nodeOrientation="RIGHT_TO_LEFT" prefWidth="722.0">
    <children>
        <Label fx:id="totalPriceLabel" text="0">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <HBox.margin>
                <Insets top="1.5" />
            </HBox.margin>
        </Label>
        <Label text="Total price :">
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <HBox.margin>
                <Insets left="2.0" top="0.5" />
            </HBox.margin>
        </Label>
    </children>
<VBox.margin>
    <Insets right="40.0" />
</VBox.margin>

```

```

        </HBox>
        <HBox>
            <children>
                <Button mnemonicParsing="false" onAction="#backToOrderPage"
styleClass="addBtn" text="Back">
                    <HBox.margin>
                        <Insets left="40.0" />
                    </HBox.margin>
                    <graphic>
                        <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
                    </graphic>
                    <font>
                        <Font name="System Bold" size="12.0" />
                    </font>
                </Button>
            </children>
            <VBox.margin>
                <Insets bottom="20.0" />
            </VBox.margin>
        </HBox>
    </children>
</VBox>
</top>
</BorderPane>

```



## קובץ ShelvesPage.fxml

זה קובץ עיצוב שמכיל את דף של ניהול המדפים.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.ShelvesController">
    <center>
        <VBox alignment="CENTER_LEFT" prefHeight="700.0" prefWidth="770.0"
BorderPane.alignment="CENTER_LEFT">
            <children>
                <Label text="Shelves list" textFill="#ff0505">
                    <VBox.margin>
                        <Insets left="70.0" top="130.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="24.0" />
                    </font>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="148.0" prefWidth="714.0"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets left="70.0" top="20.0" />
                    </VBox.margin>
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets right="50.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button focusTraversable="false" mnemonicParsing="false"
onAction="#removeShelfBtn" styleClass="removeBtn" stylesheets="@application.css"
text="Remove">
                            <HBox.margin>
                                <Insets right="10.0" />
                            </HBox.margin>
                            <cursor>
                                <Cursor fx:constant="HAND" />
                            </cursor>
                        </Button>
                    </children>
                </HBox>
            </children>
        </VBox>
    </center>
</BorderPane>
```

```

        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Button>
    <Button mnemonicParsing="false" onAction="#editShelfBtn"
styleClass="editBtn" stylesheets="@application.css" text="Edit" textFill="#171719">
        <HBox.margin>
            <Insets right="10.0" />
        </HBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
    <Button mnemonicParsing="false" onAction="#addShelfBtn"
prefHeight="26.0" prefWidth="57.0" styleClass="addBtn" stylesheets="@application.css"
text="Add" HBox.hgrow="ALWAYS">
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
        <HBox.margin>
            <Insets right="10.0" />
        </HBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Button>
    <Button mnemonicParsing="false" onAction="#viewShelfBtn"
prefHeight="26.0" prefWidth="76.0" styleClass="addBtn" stylesheets="@application.css"
text="View shelf">
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <HBox.margin>
            <Insets right="70.0" />
        </HBox.margin>
    </Button>
</children>
</HBox>
<TableView fx:id="shelvesTable" cache="true" editable="true"
prefHeight="509.0" prefWidth="506.0" VBox.vgrow="ALWAYS">
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
    <VBox.margin>
        <Insets bottom="5.0" left="70.0" right="70.0" top="5.0" />
    </VBox.margin>
    <columns>
        <TableColumn fx:id="shelfNumberCol" prefWidth="75.0" text="Shelf
number" />
        <TableColumn fx:id="maxWeightCol" prefWidth="75.0" text="Maximum
weight (KG)" />
    </columns>
</TableView>
<HBox alignment="TOP_RIGHT" prefHeight="177.0" prefWidth="784.0">
    <children>
        <Label alignment="CENTER" onMouseReleased="#addProductBtn"
prefHeight="26.0" prefWidth="105.0" styleClass="addBtn" text="Add product">

```

```

        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <graphic>
            <FontAwesomeIcon fill="WHITE" glyphName="PLUS" size="1.1em" />
        </graphic>
        <HBox.margin>
            <Insets right="70.0" />
        </HBox.margin>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</children>
</HBox>
</children>
<opaqueInsets>
    <Insets />
</opaqueInsets>
</VBox>
</center>
<left>
    <AnchorPane fx:id="side" prefHeight="668.0" prefWidth="219.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Button layoutY="128.0" mnemonicParsing="false" onAction="#inventoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Inventory">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="163.0" mnemonicParsing="false" onAction="#CategoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Category">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="198.0" mnemonicParsing="false" onAction="#CustomerPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Customer">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="233.0" mnemonicParsing="false" onAction="#orderPageBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Order">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
        </children>
    </AnchorPane>
</left>
</BorderPane>
</Scene>
</Stage>
</FXML>

```

```

        <Button layoutY="303.0" mnemonicParsing="false" onAction="#statisticsBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Statistics">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button layoutY="268.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" text="Shelves">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Logout">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
    <Button fx:id="expensesBtn" layoutX="2.0" layoutY="334.0"
mnemonicParsing="false" onAction="#expensesBtn" prefHeight="35.0" prefWidth="216.0"
styleClass="menuBtn" stylesheets="@application.css" text="Expenses">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
</children>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" stylesheets="@application.css"
BorderPane.alignment="CENTER_LEFT">
    <graphic>
        <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
    </graphic>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Label>
</top>
</BorderPane>

```

## קובץ ShelvesPage.fxml

זה קובץ עיצוב שמכיל את דף של הצגת המוצרים בתוך המדף.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.ViewShelfController">
    <top>
        <VBox alignment="CENTER_LEFT" prefHeight="700.0" prefWidth="770.0"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Products List" textFill="#ff0505">
                    <VBox.margin>
                        <Insets bottom="20.0" left="20.0" top="70.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="24.0" />
                    </font>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="26.0" prefWidth="590.0">
                    <children>
                        <HBox prefHeight="26.0" prefWidth="445.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets left="20.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button focusTraversable="false" mnemonicParsing="false"
onAction="#removeProductBtn" styleClass="removeBtn" stylesheets="@application.css"
text="Remove">
                            <HBox.margin>
                                <Insets right="10.0" />
                            </HBox.margin>
                            <font>
                                <Font name="System Bold" size="12.0" />
                            </font>
                            <cursor>
                                <Cursor fx:constant="HAND" />
                            </cursor>
                        </Button>
                    </children>
                </HBox>
            </children>
        </VBox>
    </top>
</BorderPane>
```

```

        <Button focusTraversable="false" mnemonicParsing="false"
onAction="#editProductBtn" prefHeight="26.0" prefWidth="40.0" styleClass="editBtn"
stylesheets="@application.css" text="Edit" HBox.hgrow="ALWAYS">
            <HBox.margin>
                <Insets right="20.0" />
            </HBox.margin>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
        </Button>
    </children>
</HBox>
    <TableView fx:id="productsTable" cache="true" editable="true"
prefHeight="509.0" prefWidth="731.0" VBox.vgrow="ALWAYS">
        <columnResizePolicy>
            <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
        </columnResizePolicy>
        <VBox.margin>
            <Insets bottom="10.0" left="20.0" right="20.0" top="5.0" />
        </VBox.margin>
        <columns>
            <TableColumn fx:id="shelfNumberCol" prefWidth="51.199949622154236"
text="Shelf number" />
            <TableColumn fx:id="productIdCol" minWidth="1.60003662109375"
prefWidth="234.3999786376953" text="Product id" />
            <TableColumn fx:id="productNameCol" minWidth="0.0" prefWidth="0.0"
text="Product name" />
            <TableColumn fx:id="weightCol" prefWidth="75.0" text="Weight (KG)"
/>
            <TableColumn fx:id="quantityCol" prefWidth="75.0" text="Quantity"
/>
        </columns>
    </TableView>
    <HBox nodeOrientation="RIGHT_TO_LEFT" prefWidth="722.0">
        <children>
            <Label fx:id="currentWeightLabel" text="0">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <HBox.margin>
                    <Insets top="1.5" />
                </HBox.margin>
            </Label>
            <Label text="Current weight :">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <HBox.margin>
                    <Insets left="2.0" top="0.5" />
                </HBox.margin>
            </Label>
        </children>
        <VBox.margin>
            <Insets right="20.0" />
        </VBox.margin>
    </HBox>
    <HBox prefHeight="7.0" prefWidth="1000.0">
        <VBox.margin>
            <Insets bottom="50.0" />
        </VBox.margin>
    </HBox>

```

```

        <children>
            <Button focusTraversable="false" mnemonicParsing="false"
onAction="#backBtn" prefHeight="26.0" prefWidth="59.0" styleClass="addBtn"
text="Back">
                <HBox.margin>
                    <Insets left="20.0" />
                </HBox.margin>
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <graphic>
                    <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
                </graphic>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
        </children>
    </HBox>
</children>
<opaqueInsets>
    <Insets />
</opaqueInsets>
</VBox>
</top>
</BorderPane>

```

זה קובץ עיצוב שמכיל את דף של סטטיסטיקות של העסק.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.statisticsController">
    <center>
        <VBox prefHeight="200.0" prefWidth="100.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <HBox prefHeight="146.0" prefWidth="784.0">
                    <children>
                        <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="200.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
                            <graphic>
                                <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                                    <children>
                                        <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Revenue">
                                            <font>
                                                <Font name="System Bold" size="18.0" />
                                            </font>
                                        </Label>
                                        <Label id="statisticsIntoLabels" fx:id="revenueLabel"
alignment="CENTER" prefHeight="62.0" prefWidth="200.0" text="0">
                                            <font>
                                                <Font name="System Bold" size="18.0" />
                                            </font>
                                        </Label>
                                    </children>
                                </VBox>
                            </graphic>
                        <HBox.margin>
                            <Insets left="40.0" top="30.0" />
                        </HBox.margin>
                    </Label>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="200.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
                            <graphic>
                                <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                                    <children>
                                        <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Expenses">
```



```

        <font>
            <Font name="System Bold" size="18.0" />
        </font>
    </Label>
    <Label id="statisticsIntoLabels" fx:id="expensesLabel"
alignment="CENTER" prefHeight="59.0" prefWidth="198.0" text="0">
        <font>
            <Font name="System Bold" size="18.0" />
        </font>
    </Label>
</children>
</VBox>
</graphic>
<HBox.margin>
    <Insets left="20.0" top="30.0" />
</HBox.margin>
</Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="200.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
    <graphic>
        <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
            <children>
                <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Profit">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels" fx:id="profitLabel"
alignment="CENTER" prefHeight="55.0" prefWidth="200.0" text="0">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
            </children>
        </VBox>
    </graphic>
    <cursor>
        <Cursor fx:constant="DEFAULT" />
    </cursor>
    <HBox.margin>
        <Insets left="20.0" top="30.0" />
    </HBox.margin>
</Label>
</children>
</HBox>
<HBox prefHeight="146.0" prefWidth="784.0">
    <children>
        <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" onMouseReleased="#bestSellingBtn" prefHeight="85.0"
prefWidth="170.0" stylesheets="@application.css" textAlignment="CENTER"
textOverrun="CLIP">
            <graphic>
                <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                    <children>
                        <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Best-selling">
                            <font>
                                <Font name="System Bold" size="18.0" />
                            </font>

```

```

        </Label>
        <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="62.0" prefWidth="170.0" text="Product">
            <font>
                <Font name="System Bold" size="18.0" />
            </font>
        </Label>
    </children>
</VBox>
</graphic>
<HBox.margin>
    <Insets left="40.0" top="30.0" />
</HBox.margin>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
</Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" onMouseReleased="#bestSellingTop3Btn" prefHeight="85.0"
prefWidth="170.0" stylesheets="@application.css" textAlignment="CENTER"
textOverrun="CLIP">
        <graphic>
            <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                <children>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Best-selling">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="62.0" prefWidth="170.0" text="Top 3">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                </children>
            </VBox>
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
        <HBox.margin>
            <Insets left="10.0" top="30.0" />
        </HBox.margin>
    </Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" onMouseReleased="#ordersNumberBtn" prefHeight="85.0"
prefWidth="170.0" stylesheets="@application.css" textAlignment="CENTER"
textOverrun="CLIP">
        <graphic>
            <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                <children>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Orders number ">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="62.0" prefWidth="170.0" text="of customers">

```

```

        <font>
            <Font name="System Bold" size="18.0" />
        </font>
    </Label>
</children>
</VBox>
</graphic>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
<HBox.margin>
    <Insets left="10.0" top="30.0" />
</HBox.margin>
</Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" onMouseReleased="#orderInMonthBtn" prefHeight="85.0"
prefWidth="170.0" stylesheets="@application.css" textAlignment="CENTER"
textOverrun="CLIP">
        <graphic>
            <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                <children>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Orders in ">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                    <Label id="statisticsIntoLabels" fx:id="monthLabel"
alignment="CENTER" prefHeight="62.0" prefWidth="170.0">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                </children>
            </VBox>
        </graphic>
    </cursor>
    <Cursor fx:constant="HAND" />
</cursor>
<HBox.margin>
    <Insets left="10.0" top="30.0" />
</HBox.margin>
</Label>
</children>
</HBox>
<HBox prefHeight="146.0" prefWidth="784.0">
    <children>
        <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
            <graphic>
                <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                    <children>
                        <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Customers number">
                            <font>
                                <Font name="System Bold" size="18.0" />
                            </font>
                        </Label>

```

```

        <Label id="statisticsIntoLabels"
fx:id="customerNumberLabel" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="">
        <font>
            <Font name="System Bold" size="18.0" />
        </font>
    </Label>
</children>
</VBox>
</graphic>
<HBox.margin>
    <Insets left="40.0" top="30.0" />
</HBox.margin>
<cursor>
    <Cursor fx:constant="DEFAULT" />
</cursor>
</Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
    <graphic>
        <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
            <children>
                <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Orders number">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels"
fx:id="ordersNumberLabel" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
            </children>
        </VBox>
    </graphic>
    <HBox.margin>
        <Insets left="10.0" top="30.0" />
    </HBox.margin>
    <cursor>
        <Cursor fx:constant="DEFAULT" />
    </cursor>
</Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
    <graphic>
        <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
            <children>
                <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Categories number">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels"
fx:id="categoryNumberLabel" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="">

```

```

        <font>
            <Font name="System Bold" size="18.0" />
        </font>
    </Label>
</children>
</VBox>
</graphic>
<cursor>
    <Cursor fx:constant="DEFAULT" />
</cursor>
<HBox.margin>
    <Insets left="10.0" top="30.0" />
</HBox.margin>
</Label>
</children>
</HBox>
<HBox prefHeight="142.0" prefWidth="784.0">
    <children>
        <VBox prefHeight="200.0" prefWidth="180.0">
            <children>
                <Label text="Inventory" textFill="RED">
                    <font>
                        <Font name="System Bold" size="14.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
                    <graphic>
                        <VBox alignment="CENTER" prefHeight="200.0"
prefWidth="100.0" styleClass="statisticsLabels" stylesheets="@application.css">
                            <children>
                                <Label id="statisticsIntoLabels"
alignment="CENTER" prefHeight="60.0" prefWidth="170.0" text="Products number">
                                    <font>
                                        <Font name="System Bold" size="18.0" />
                                    </font>
                                </Label>
                                <Label id="statisticsIntoLabels"
fx:id="productsNumberLabel" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="">
                                    <font>
                                        <Font name="System Bold" size="18.0" />
                                    </font>
                                </Label>
                            </children>
                        </VBox>
                    </graphic>
                <cursor>
                    <Cursor fx:constant="DEFAULT" />
                </cursor>
                <VBox.margin>
                    <Insets left="10.0" top="10.0" />
                </VBox.margin>
            </Label>
        </children>
    <HBox.margin>
        <Insets left="40.0" />
    </HBox.margin>
</VBox>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">

```

```

        <graphic>
            <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                <children>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Total products">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                    <Label id="statisticsIntoLabels"
fx:id="productQuantityLabel" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="0">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                </children>
            </VBox>
        </graphic>
        <cursor>
            <Cursor fx:constant="DEFAULT" />
        </cursor>
        <HBox.margin>
            <Insets left="10.0" top="30.0" />
        </HBox.margin>
    </Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" onMouseReleased="#productsTheCategoryBtn" prefHeight="85.0"
prefWidth="170.0" stylesheets="@application.css" textAlignment="CENTER"
textOverrun="CLIP">
        <graphic>
            <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
                <children>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Products In the">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                    <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="62.0" prefWidth="170.0" text="Categories">
                        <font>
                            <Font name="System Bold" size="18.0" />
                        </font>
                    </Label>
                </children>
            </VBox>
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
        <HBox.margin>
            <Insets left="10.0" top="30.0" />
        </HBox.margin>
    </Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" onMouseReleased="#PrordutsInCategoriesBtn" prefHeight="85.0"
prefWidth="170.0" stylesheets="@application.css" textAlignment="CENTER"
textOverrun="CLIP">
        <graphic>

```

```

        <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
        <children>
            <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Percentage of products">
                <font>
                    <Font name="System Bold" size="15.0" />
                </font>
            </Label>
            <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="62.0" prefWidth="170.0" text="In the categories">
                <font>
                    <Font name="System Bold" size="15.0" />
                </font>
            </Label>
        </children>
    </VBox>
</graphic>
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
<HBox.margin>
    <Insets left="10.0" top="30.0" />
</HBox.margin>
</Label>
</children>
</HBox>
<HBox prefHeight="146.0" prefWidth="784.0">
    <children>
        <VBox prefHeight="200.0" prefWidth="180.0">
            <children>
                <Label text="Shelf" textFill="RED">
                    <font>
                        <Font name="System Bold" size="14.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
                    <graphic>
                        <VBox alignment="CENTER" prefHeight="200.0"
prefWidth="100.0" styleClass="statisticsLabels" stylesheets="@application.css">
                            <children>
                                <Label id="statisticsIntoLabels"
alignment="CENTER" prefHeight="60.0" prefWidth="170.0" text="Shelves number">
                                    <font>
                                        <Font name="System Bold" size="18.0" />
                                    </font>
                                </Label>
                                <Label id="statisticsIntoLabels"
fx:id="shelvesNumberLabel" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="0">
                                    <font>
                                        <Font name="System Bold" size="18.0" />
                                    </font>
                                </Label>
                            </children>
                        </VBox>
                    </graphic>
                </cursor>
                <Cursor fx:constant="DEFAULT" />
            </cursor>
        </VBox.margin>
    </children>

```

```

        <Insets left="10.0" top="10.0" />
    </VBox.margin>
    </Label>
</children>
<HBox.margin>
    <Insets left="40.0" />
</HBox.margin>
</VBox>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
    <graphic>
        <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
            <children>
                <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Products number">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels"
fx:id="prodQuantityInShelf" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="0">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
            </children>
        </VBox>
    </graphic>
    <cursor>
        <Cursor fx:constant="DEFAULT" />
    </cursor>
    <HBox.margin>
        <Insets left="10.0" top="30.0" />
    </HBox.margin>
</Label>
    <Label id="statisticsIntoLabels" alignment="CENTER"
contentDisplay="CENTER" prefHeight="85.0" prefWidth="170.0"
stylesheets="@application.css" textAlignment="CENTER" textOverrun="CLIP">
    <graphic>
        <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0"
styleClass="statisticsLabels" stylesheets="@application.css">
            <children>
                <Label id="statisticsIntoLabels" alignment="CENTER"
prefHeight="60.0" prefWidth="170.0" text="Total products">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <Label id="statisticsIntoLabels"
fx:id="totalProductInShelf" alignment="CENTER" prefHeight="62.0" prefWidth="170.0"
text="0">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
            </children>
        </VBox>
    </graphic>
    <cursor>
        <Cursor fx:constant="DEFAULT" />

```



```

        </cursor>
        <HBox.margin>
            <Insets left="10.0" top="30.0" />
        </HBox.margin>
    </Label>
</children>
</HBox>
</children>
</VBox>
</center>
<left>
    <AnchorPane fx:id="side" prefHeight="667.0" prefWidth="174.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Button layoutY="128.0" mnemonicParsing="false" onAction="#inventoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Inventory">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="163.0" mnemonicParsing="false" onAction="#CategoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Category">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="198.0" mnemonicParsing="false" onAction="#CustomerPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Customer">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="233.0" mnemonicParsing="false" onAction="#orderPageBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Order">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="303.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" text="Statistics">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="268.0" mnemonicParsing="false" onAction="#shelvesBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Shelves">
                <font>

```

```

        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Logout">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
<Button fx:id="expensesBtn" layoutY="338.0" mnemonicParsing="false"
onAction="#expensesBtn" prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Expenses">
    <font>
        <Font name="System Bold" size="12.0" />
    </font>
    <cursor>
        <Cursor fx:constant="HAND" />
    </cursor>
</Button>
</children>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" stylesheets="@application.css"
BorderPane.alignment="CENTER_LEFT">
        <graphic>
            <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</top>
</BorderPane>

```

## קובץ productsInCategories.fxml

זה קובץ עיצוב שמכיל את דף של הצגת אחוזים של המוצרים לכל קטגוריה במלאי.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.chart.PieChart?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.productsInCategoriesController">
    <bottom>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="percentage of products in the categories" textFill="RED"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <PieChart fx:id="pieChart" prefHeight="400.0" prefWidth="622.0">
                    <VBox.margin>
                        <Insets left="200.0" right="200.0" top="40.0" />
                    </VBox.margin>
                </PieChart>
                <HBox>
                    <children>
                        <Button mnemonicParsing="false" onAction="#backToStatisticsPage"
styleClass="addBtn" stylesheets="@application.css" text="Back" textFill="WHITE">
                            <HBox.margin>
                                <Insets left="50.0" />
                            </HBox.margin>
                            <cursor>
                                <Cursor fx:constant="HAND" />
                            </cursor>
                            <font>
                                <Font name="System Bold" size="12.0" />
                            </font>
                            <graphic>
                                <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT" size="1.2em" />
                            </graphic>
                        </Button>
                    </children>
                    <VBox.margin>
                        <Insets top="100.0" />
                    </VBox.margin>
                </HBox>
            </children>
        </VBox>
    </bottom>
</BorderPane>
```

```
        </children>
    </VBox>
</bottom>
</BorderPane>
```

## קובץ Best-selling-page.fxml

זה קובץ עיצוב שמכיל את דף של הצגת כמה נמכר כל מוצר.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.bestSellingProductController">
    <top>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label fx:id="titleLabel" text="Best-selling product" textFill="RED"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox prefHeight="0.0" prefWidth="1000.0">
                    <children>
                        <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                            <HBox.margin>
                                <Insets right="50.0" />
                            </HBox.margin>
                        </TextField>
                    </children>
                    <VBox.margin>
                        <Insets bottom="5.0" left="40.0" />
                    </VBox.margin>
                </HBox>
                <TableView fx:id="productTable" focusTraversable="false"
prefHeight="200.0" prefWidth="200.0" stylesheets="@application.css"
VBox.vgrow="ALWAYS">
                    <columns>
                        <TableColumn fx:id="productIdCol" prefWidth="75.0" text="Product id" />
                        <TableColumn fx:id="productNameCol" prefWidth="75.0" text="Product name" />
                        <TableColumn fx:id="numberSoldCol" prefWidth="75.0" text="Number sold" />
                    </columns>
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" right="40.0" />
                    </VBox.margin>
                    <columnResizePolicy>
                        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
                    </columnResizePolicy>
                </TableView>
            </children>
        </VBox>
    </top>
</BorderPane>
```

```

        </columnResizePolicy>
    </TableView>
    <HBox>
        <children>
            <Button mnemonicParsing="false" onAction="#backToOrderPage"
styleClass="addBtn" text="Back">
                <HBox.margin>
                    <Insets left="40.0" />
                </HBox.margin>
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <graphic>
                    <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
                </graphic>
            </Button>
        </children>
        <VBox.margin>
            <Insets bottom="20.0" />
        </VBox.margin>
    </HBox>
</children>
</VBox>
</top>
</BorderPane>

```

## קובץ Best-selling-page-top-3.fxml

זה קובץ עיצוב שמכיל את דף של הצגת שלושה מוצרים הכי נמכרים.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.chart.BarChart?>
<?import javafx.scene.chart.CategoryAxis?>
<?import javafx.scene.chart.NumberAxis?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.bestSellingTop3Controller">
    <bottom>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Best-selling Products" textFill="RED" VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <BarChart fx:id="barChart" animated="false"
stylesheets="@application.css" title="Best-selling">
                    <xAxis>
                        <CategoryAxis label="Product" prefHeight="41.0" prefWidth="685.0"
side="BOTTOM" fx:id="x" />
                    </xAxis>
                    <yAxis>
                        <NumberAxis fx:id="y" label="Number sold" prefHeight="336.0"
prefWidth="46.0" side="LEFT" />
                    </yAxis>
                    <VBox.margin>
                        <Insets bottom="70.0" left="120.0" right="120.0" top="100.0" />
                    </VBox.margin>
                </BarChart>
                <HBox>
                    <children>
                        <Button mnemonicParsing="false" onAction="#backToStatisticsPage"
styleClass="addBtn" text="Back">
                            <HBox.margin>
                                <Insets left="40.0" />
                            </HBox.margin>
                            <font>
                                <Font name="System Bold" size="12.0" />
                            </font>
                            <graphic>
                                <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
                            </graphic>
                        </children>
                    </HBox>
                </children>
            </VBox>
        </bottom>
    </BorderPane>
```

```
        </Button>
    </children>
    <VBox.margin>
        <Insets bottom="20.0" />
    </VBox.margin>
</HBox>
</children>
</VBox>
</bottom>
</BorderPane>
```



## קובץ Best-selling-page-top-3.fxml

זה קובץ עיצוב שמכיל את דף של הצגת שלושה מוצרים הכי נמכרים.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.OrdersNumberController">
    <top>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Orders number of customers" textFill="RED"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox prefHeight="0.0" prefWidth="1000.0">
                    <children>
                        <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                            <HBox.margin>
                                <Insets right="50.0" />
                            </HBox.margin>
                        </TextField>
                    </children>
                    <VBox.margin>
                        <Insets bottom="5.0" left="40.0" />
                    </VBox.margin>
                </HBox>
                <TableView fx:id="productTable" focusTraversable="false"
prefHeight="200.0" prefWidth="200.0" stylesheets="@application.css"
VBox.vgrow="ALWAYS">
                    <columns>
                        <TableColumn fx:id="customerIdCol" prefWidth="75.0" text="Customer
id" />
                        <TableColumn fx:id="customerNameCol" prefWidth="75.0"
text="Customer name" />
                        <TableColumn fx:id="phoneCol" prefWidth="75.0" text="Phone" />
                        <TableColumn fx:id="ordersNumberCol" prefWidth="75.0" text="Orders
number" />
                    </columns>
```

```

        <VBox.margin>
            <Insets bottom="20.0" left="40.0" right="40.0" />
        </VBox.margin>
        <columnResizePolicy>
            <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
        </columnResizePolicy>
    </TableView>
    <HBox>
        <children>
            <Button mnemonicParsing="false" onAction="#backToOrderPage"
styleClass="addBtn" stylesheets="@application.css" text="Back" textFill="WHITE">
                <HBox.margin>
                    <Insets left="40.0" />
                </HBox.margin>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <graphic>
                    <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
                </graphic>
            </Button>
        </children>
    </HBox>
    <VBox.margin>
        <Insets bottom="20.0" />
    </VBox.margin>
</HBox>
</children>
</VBox>
</top>
</BorderPane>

```

## קובץ OrderInMonth.fxml

זה קובץ עיצוב שמכיל את דף של הצגת כל ההזמנות בחודש הנוכחי.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.OrderInMonthController">
    <top>
        <VBox prefHeight="700.0" prefWidth="1000.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Orders List" textFill="RED" VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="30.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox prefHeight="0.0" prefWidth="1000.0">
                    <children>
                        <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                            <HBox.margin>
                                <Insets right="50.0" />
                            </HBox.margin>
                        </TextField>
                    </children>
                    <VBox.margin>
                        <Insets bottom="5.0" left="40.0" />
                    </VBox.margin>
                </HBox>
                <TableView fx:id="orderTable" focusTraversable="false" prefHeight="200.0"
prefWidth="200.0" stylesheets="@application.css" VBox.vgrow="ALWAYS">
                    <columns>
                        <TableColumn fx:id="orderIdCol" prefWidth="75.0" text="Order id" />
                        <TableColumn fx:id="customerIdCol" prefWidth="75.0" text="Customer
id" />
                        <TableColumn fx:id="totalPriceCol" prefWidth="75.0" text="Total
price" />
                        <TableColumn fx:id="orderDateCol" prefWidth="75.0" text="Order
date" />
                    </columns>
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" right="40.0" />
                    </VBox.margin>
                </TableView>
            </children>
        </VBox>
    </top>
</BorderPane>
```

```

        </VBox.margin>
        <columnResizePolicy>
            <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
        </columnResizePolicy>
    </TableView>
    <HBox>
        <children>
            <Button mnemonicParsing="false" onAction="#backToOrderPage"
styleClass="addBtn" stylesheets="@application.css" text="Back">
                <HBox.margin>
                    <Insets left="40.0" />
                </HBox.margin>
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
                <graphic>
                    <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
                </graphic>
            </Button>
        </children>
        <VBox.margin>
            <Insets bottom="20.0" />
        </VBox.margin>
    </HBox>
</children>
</VBox>
</top>
</BorderPane>

```

## קובץ ProductsInTheCategory.fxml

זה קובץ עיצוב שמכיל את דף של הצגת מספר המוצרים בכל קטגוריה.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="application.ProductInTheCategoryController">
    <bottom>
        <VBox prefHeight="700.0" prefWidth="1000.0" styleClass="searchField"
stylesheets="@application.css" BorderPane.alignment="CENTER">
            <children>
                <Label text="Products In the category" textFill="RED"
VBox.vgrow="ALWAYS">
                    <VBox.margin>
                        <Insets bottom="20.0" left="40.0" top="50.0" />
                    </VBox.margin>
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                </Label>
                <HBox prefHeight="0.0" prefWidth="1000.0">
                    <children>
                        <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                            <HBox.margin>
                                <Insets right="50.0" />
                            </HBox.margin>
                        </TextField>
                    </children>
                    <VBox.margin>
                        <Insets bottom="5.0" left="40.0" />
                    </VBox.margin>
                </HBox>
                <TableView fx:id="categoryTable" focusTraversable="false"
prefHeight="200.0" prefWidth="200.0" stylesheets="@application.css"
VBox.vgrow="ALWAYS">
                    <columns>
                        <TableColumn fx:id="categoryIdCol" prefWidth="75.0" text="Category
id" />
                        <TableColumn fx:id="categoryNameCol" prefWidth="75.0"
text="Category name" />
                        <TableColumn fx:id="productsNumberCol" prefWidth="75.0"
text="Products number" />
                    </columns>
                    <VBox.margin>
```

```

        <Insets bottom="20.0" left="40.0" right="40.0" />
    </VBox.margin>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
</TableView>
<HBox>
    <children>
        <Button mnemonicParsing="false" onAction="#backToOrderPage"
styleClass="addBtn" text="Back">
            <HBox.margin>
                <Insets left="40.0" />
            </HBox.margin>
            <cursor>
                <Cursor fx:constant="HAND" />
            </cursor>
            <font>
                <Font name="System Bold" size="12.0" />
            </font>
            <graphic>
                <FontAwesomeIcon fill="WHITE" glyphName="ARROW_LEFT"
size="1.2em" />
            </graphic>
        </Button>
    </children>
    <VBox.margin>
        <Insets bottom="20.0" />
    </VBox.margin>
</HBox>
</children>
</VBox>
</bottom>
</BorderPane>

```

## קובץ Expenses.fxml

זה קובץ עיצוב שמכיל את דף של הצגת כל ההוצאות העובדים וגם הוצאות של מים וחשמל.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import de.jensd.fx.glyphs.fontawesome.FontAwesomeIcon?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.Cursor?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="700.0" prefWidth="1000.0" styleClass="back"
stylesheets="@application.css" xmlns="http://javafx.com/javafx/15.0.1"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="application.ExpensesController">
    <center>
        <VBox prefHeight="200.0" prefWidth="100.0" stylesheets="@application.css"
BorderPane.alignment="CENTER">
            <children>
                <Label text="Expenses" textFill="RED">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                    <VBox.margin>
                        <Insets bottom="10.0" left="20.0" top="30.0" />
                    </VBox.margin>
                </Label>
                <Label text="Customers salary">
                    <font>
                        <Font name="System Bold" size="18.0" />
                    </font>
                    <VBox.margin>
                        <Insets bottom="20.0" left="30.0" />
                    </VBox.margin>
                </Label>
                <HBox alignment="TOP_RIGHT" prefHeight="0.0" prefWidth="784.0">
                    <children>
                        <HBox prefHeight="100.0" prefWidth="200.0" HBox.hgrow="ALWAYS">
                            <children>
                                <TextField fx:id="searchField" focusTraversable="false"
onKeyTyped="#searchFieldTyped" prefHeight="26.0" prefWidth="233.0"
promptText="Search" styleClass="searchField">
                                    <HBox.margin>
                                        <Insets left="20.0" />
                                    </HBox.margin>
                                </TextField>
                            </children>
                        </HBox>
                        <Button mnemonicParsing="false" onAction="#updateSalaryBtn"
prefHeight="26.0" prefWidth="108.0" styleClass="addBtn"
stylesheets="@application.css" text="Update Salary" HBox.hgrow="ALWAYS">
                            <cursor>
                                <Cursor fx:constant="HAND" />
                            </cursor>
                        </Button>
                    </children>
                </HBox>
            </children>
        </VBox>
    </center>
</BorderPane>
```

```

        </cursor>
        <HBox.margin>
            <Insets right="20.0" />
        </HBox.margin>
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
    </Button>
</children>
</HBox>
<TableView fx:id="employeesTable" focusTraversable="false"
prefHeight="257.0" prefWidth="744.0" stylesheets="@application.css">
    <columns>
        <TableColumn fx:id="fullNameCol" prefWidth="75.0" sortable="false"
text="Full name" />
        <TableColumn fx:id="userNameCol" prefWidth="75.0" sortable="false"
text="User name" />
        <TableColumn fx:id="salaryDayCol" prefWidth="75.0" sortable="false"
text="Salary day" />
        <TableColumn fx:id="generalSalaryCol" prefWidth="75.0"
sortable="false" text="General Salary" />
        <TableColumn fx:id="salaryDateCol" prefWidth="75.0"
sortable="false" text="Update salary date" />
    </columns>
    <columnResizePolicy>
        <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
    </columnResizePolicy>
    <VBox.margin>
        <Insets bottom="20.0" left="20.0" right="20.0" top="10.0" />
    </VBox.margin>
</TableView>
<HBox alignment="CENTER" prefHeight="10.0" prefWidth="784.0">
    <children>
        <Label fx:id="currentMonthLabel" alignment="CENTER" text="Current
month">
            <font>
                <Font name="System Bold" size="18.0" />
            </font>
            <HBox.margin>
                <Insets />
            </HBox.margin>
        </Label>
    </children>
</HBox>
<HBox alignment="CENTER" prefHeight="10.0" prefWidth="784.0">
    <children>
        <Label text="Electrical expenses">
            <font>
                <Font name="System Bold" size="18.0" />
            </font>
            <HBox.margin>
                <Insets right="185.0" />
            </HBox.margin>
        </Label>
        <Label alignment="CENTER" prefHeight="27.0" prefWidth="150.0"
text="Water expenses">
            <font>
                <Font name="System Bold" size="18.0" />
            </font>
            <HBox.margin>
                <Insets />
            </HBox.margin>
        </Label>
    </children>
</HBox>

```



```

        </children>
        <VBox.margin>
            <Insets top="20.0" />
        </VBox.margin>
    </HBox>
    <HBox alignment="CENTER" prefHeight="20.0" prefWidth="784.0">
        <children>
            <TextField fx:id="electricalExpField" focusTraversable="false"
prefHeight="26.0" prefWidth="198.0" promptText="Electrical expenss" style="-fx-
border-color: black; -fx-border-width: 2px 2px 2px 2px;">
                <HBox.margin>
                    <Insets right="140.0" />
                </HBox.margin>
            </TextField>
            <TextField fx:id="waterExpField" prefWidth="198.0"
promptText="Water expenss" style="-fx-border-color: black; -fx-border-width: 2px 2px
2px 2px;" />
        </children>
        <VBox.margin>
            <Insets top="10.0" />
        </VBox.margin>
    </HBox>
    <HBox alignment="CENTER" prefHeight="3.0" prefWidth="784.0">
        <VBox.margin>
            <Insets top="10.0" />
        </VBox.margin>
        <children>
            <Button mnemonicParsing="false" onAction="#electricalUpdateBtn"
styleClass="addBtn" text="Update">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <HBox.margin>
                    <Insets />
                </HBox.margin>
            </Button>
            <Button mnemonicParsing="false" onAction="#showElectricalExpBtn"
prefHeight="26.0" prefWidth="58.0" styleClass="addBtn" text="Show">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <HBox.margin>
                    <Insets left="20.0" right="205.0" />
                </HBox.margin>
            </Button>
            <Button mnemonicParsing="false" onAction="#waterUpdateField"
styleClass="addBtn" text="Update">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <HBox.margin>
                    <Insets right="20.0" />
                </HBox.margin>
            </Button>
            <Button mnemonicParsing="false" onAction="#showWaterExpBtn"
prefWidth="58.0" styleClass="addBtn" text="Show">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
            </Button>
        </children>
    </HBox>
</children>

```

```

    </VBox>
</center>
<left>
    <AnchorPane fx:id="side" prefHeight="667.0" prefWidth="174.0" style="-fx-
background-color: #0598ff;" BorderPane.alignment="CENTER">
        <children>
            <Button layoutY="128.0" mnemonicParsing="false" onAction="#inventoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn"
stylesheets="@application.css" text="Inventory">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="163.0" mnemonicParsing="false" onAction="#CategoryBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Category">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="198.0" mnemonicParsing="false" onAction="#CustomerPage"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Customer">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="233.0" mnemonicParsing="false" onAction="#orderPageBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Order">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="303.0" mnemonicParsing="false" onAction="#statisticsBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Statistics">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="268.0" mnemonicParsing="false" onAction="#shelvesBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Shelves">
                <font>
                    <Font name="System Bold" size="12.0" />
                </font>
                <cursor>
                    <Cursor fx:constant="HAND" />
                </cursor>
            </Button>
            <Button layoutY="632.0" mnemonicParsing="false" onAction="#LogoutBtn"
prefHeight="35.0" prefWidth="216.0" styleClass="menuBtn" text="Logout">

```

```

        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
    <Button layoutY="334.0" mnemonicParsing="false" prefHeight="35.0"
prefWidth="216.0" styleClass="menuBtn" stylesheets="@application.css"
text="Expenses">
        <font>
            <Font name="System Bold" size="12.0" />
        </font>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Button>
</children>
</AnchorPane>
</left>
<top>
    <Label alignment="CENTER" onMouseReleased="#menuClicked" prefHeight="35.0"
prefWidth="34.0" styleClass="menuLabel" stylesheets="@application.css"
BorderPane.alignment="CENTER_LEFT">
        <graphic>
            <FontAwesomeIcon glyphName="BARS" selectionStart="4" size="2.3em"
tabSize="16" />
        </graphic>
        <cursor>
            <Cursor fx:constant="HAND" />
        </cursor>
    </Label>
</top>
</BorderPane>

```

# **קוד קבצי Java של התוכנה**

**(נכתב בסביבת פיתוח Eclipse)**

## קובץ SignupController.java

זאת מחלקה ההרשמה למערכת.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.BorderPane;

public class SignupController implements Initializable{
    @FXML
    private BorderPane Loginpage;

    @FXML
    private PasswordField passwordField;

    @FXML
    private TextField emailField,idField,lastNameField,firstNameField,userField;

    @FXML
    private CheckBox checkbox1;

    @FXML
    private Label errtxt,firstNameLabel,lastNameLabel,emailLabel,idLabel,userNameLabel,passwordLabel,SignUpbtn,exitbtn;

    ObservableList<User> userList;

    private Label []allLabels = new Label[7];

    /* allLabels[0] => for errtxt label , allLabels[1] => for first name label ,
    allLabels[2] => for last name label , allLabels[3] => for email label , allLabels[4] => for id label
    allLabels[5] => for user name label , allLabels[6] => for password label*/

    @FXML
    void loginbtn(MouseEvent event) throws IOException {
        SwitchingPages.switchPage(event,"Login.fxml", "Management products - Login");
    }
    // this function add new user
    @FXML
    void SignUpbutton(ActionEvent event) throws SQLException {
        Field field = new Field();

        // checks if all field is not empty and checks if the check box is selected
        if(checkbox1.isSelected() && field.checkEmptyFields(userField, lastNameField, idField, firstNameField, emailField, passwordField))
        {

            Boolean idFlag = field.checkIdField(idField, idLabel,"blue");
            Boolean userNameFlag = field.checkUserNameField(userField, userNameLabel);
            Boolean emailFlag = field.checkEmailField(emailField, emailLabel,"blue");
            Boolean firstNameFlag = field.checkNameField(firstNameField, firstNameLabel);
            Boolean lastNameFlag = field.checkNameField(lastNameField, lastNameLabel);
            Boolean passwordFlag = field.checkPasswordField(passwordField, passwordLabel);

            // check fields is valid
            if(idFlag && userNameFlag && emailFlag && firstNameFlag && lastNameFlag && passwordFlag) {

                User user = new
                User(firstNameField.getText(),lastNameField.getText(),emailField.getText(),Integer.parseInt(idField.getText()),userField.getText(),passwordField.getText());

                //check if user name is exists
                if(user.userNameIsTaken(userList,user) == true) {
                    field.changeBorderColor(userField, "blue"); // change border color to red
                    DatabaseManager.addUser(user);//add user
                    clearLabels(); //clear the labels
                    Message.showMessageDialog(AlertType.INFORMATION,"you've successfully registered for the
                    application", "successful registration");
                }
            }
        }
    }
}
```

```

        }else {
            // if user name is exists
            field.changeBorderColor(userField, "red"); // change border color to red
            userNameLabel.setText("*User name is taken");
        }
    }
} else {
    clearLabels(); // clear labels
    // if you have empty fields and not selected to check box
    if(!(checkbox1.isSelected())) && !field.checkEmptyFields(userField, passwordField, lastNameField, idField, firstNameField,
emailField))
        errtxt.setText("*Empty feilds and select the checkbox");
    // if check box is selected
    else if!(checkbox1.isSelected())
        errtxt.setText("*Select the check box");
    else // if you have emoty fields
        errtxt.setText("*You have empty feilds");
    }
}

//this function clear the content in the fields
private void clearLabels() {
    for(int i = 0;i<allLabels.length;i++) {
        allLabels[i].setText("");
    }
}

// first name field -> on key typed
@FXML
void firstNameTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(firstNameField.getText(), 15))
        Conditions.removeLastIndexInString(firstNameField);
}

// last name field -> on key typed
@FXML
void lastNameTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(lastNameField.getText(), 15))
        Conditions.removeLastIndexInString(lastNameField);
}

// id field -> on key typed
@FXML
void idTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(idField.getText(), 9))
        Conditions.removeLastIndexInString(idField);
}

// user name field -> on key typed
@FXML
void userNameTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(userField.getText(), 13))
        Conditions.removeLastIndexInString(userField);
}

// password field -> on key typed
@FXML
void passwordTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(passwordField.getText(), 15))
        Conditions.removeLastIndexInString(passwordField);
}

// set user list
private void setUsers() throws ClassNotFoundException, SQLException {
    userList = DatabaseManager.getUsers(); // get all users
}

//errtxt ,firstNameLabel,lastNameLabel,emailLabel,idLabel,userNameLabel,passwordLabel
private void setAllLabelsInArray() {
    allLabels[0] = errtxt; // error text
    allLabels[1] = firstNameLabel; // first name label
    allLabels[2] = lastNameLabel; // last name label
    allLabels[3] = emailLabel; // email label
    allLabels[4] = idLabel; // id label
    allLabels[5] = userNameLabel; // user name label
    allLabels[6] = passwordLabel; // password label
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    try {
        setUsers();
        setAllLabelsInArray();
    }
}

```

```
    } catch (ClassNotFoundException | SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}  
}
```

## קובץ Controller.java

זאת מחלקת הכניסה למערכת.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.*;

public class Controller implements Initializable {

    @FXML
    private BorderPane Loginpage;

    @FXML
    private Label exitbtn, SignUpbtn, errtxt, userNameLabel, passwordLabel;

    @FXML
    private TextField userNameField;

    private Label []allLabels = new Label[3];

    @FXML
    private PasswordField passwordField;

    ObservableList<User> userList;
    Manager manager;

    //switch window
    // switch to sign up window
    @FXML
    void SignUpbtn(MouseEvent event) throws IOException{
        SwitchingPages.switchPage(event, "SignUp.fxml", "Management products - Sign up");
    }

    //this function is activated if you press the button and checks if the details are correct
    @FXML
    void Loginbutton(MouseEvent event) throws IOException, ClassNotFoundException, SQLException {
        Field field = new Field();
        if(field.checkEmptyFields(userNameField, passwordField)) { // check if the all fields is not empty
            if(manager.getUserName().equals(userNameField.getText())) { // check if this the
managerif(manager.checkPassword(manager, passwordField.getText()) == true) { // checks if the password of the manager is correct
                manager.setStatus(true);
                DatabaseManager.editManagerStatus(manager);
                SwitchingPages.openWindowInCenter("inventory", event, "Management
products - Inventory");
            } else { // if the password of the manager is not correct
                clearLabels();
                passwordLabel.setText("*Check your password again!");
                field.changeBorderColor(passwordField, "red");
                field.changeBorderColor(userNameField, "blue");
            }
        } else { // user
            User user = new User();
            if((user = user.checkUserName(userList, userNameField.getText())) != null) { //check if the user is existsif(user.checkPassword(user,
passwordField.getText())) { // if the user name exists check the password is correct
                SwitchingPages.openWindowInCenter("inventory", event, "Management products - Inventory");
            } else { // if the password is not correct
                clearLabels();
                passwordLabel.setText("*Check your password again!");
                field.changeBorderColor(passwordField, "red");
                field.changeBorderColor(userNameField, "blue");
            }
        }
    } else { // if the user name is not correct
        clearLabels();
        userNameLabel.setText("*User name is not exist");
        field.changeBorderColor(passwordField, "blue");
    }
}
```



```

        field.changeBorderColor(userNameField, "red");
    }
}
} else {
    // if you have empty fields
    clearLabels(); // clear all labels
    // if the user name and the password fields is empty
    if(userNameField.getText().isEmpty() && (passwordField.getText().isEmpty())) {
        errtxt.setText("*Your fields is empty");
        field.changeBorderColor(passwordField, "red");
        field.changeBorderColor(userNameField, "red");
    }
    // if the user name field is empty
    else if(userNameField.getText().isEmpty()) {
        errtxt.setText("*User name field is empty");
        field.changeBorderColor(passwordField, "blue");
        field.changeBorderColor(userNameField, "red");
    }
    // if the password field is empty
    else if(passwordField.getText().isEmpty()) {
        errtxt.setText("*password field is empty");
        field.changeBorderColor(passwordField, "red");
        field.changeBorderColor(userNameField, "blue");
    }
}
}
}

// this function clear all labels
private void clearLabels() {
    for(int i = 0; i < allLabels.length; i++) {
        allLabels[i].setText("");
    }
}

//set all users in the database in ObservableList
private void setUserList() throws ClassNotFoundException, SQLException {
    userList = DatabaseManager.getUsers();
}

//set the manager in the database
private void setManager() throws SQLException, ClassNotFoundException {
    manager = DatabaseManager.getManager();
}

// set the labels in array
private void setAllLabel() {
    allLabels[0] = userNameLabel; // user name label
    allLabels[1] = passwordLabel; // password label
    allLabels[2] = errtxt;
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    try {
        setUserList();
        setManager();
        setAllLabel();
    } catch (SQLException | ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

## קובץ InventoryController.java

המחלקה הזאת מציגה את החלון של המוצרים שנמצאים במלאי.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

public class InventoryController implements Initializable{

    @FXML
    private TableView<Product> productsTable;

    @FXML
    private TableColumn<Product, Integer> productIdCol;

    @FXML
    private TableColumn<Product, String> productNameCol,categoryCol;

    @FXML
    private TableColumn<Product, Double> priceCol,supplierPriceCol,weightCol;

    @FXML
    private TableColumn<Product, Integer> quantityCol;

    @FXML
    private TextField searchField;

    @FXML
    private AnchorPane side;

    ObservableList<Product> productList;

    @FXML
    private Button expensesBtn;

    //switch window
    @FXML
    void expensesBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"Expenses.fxml","Management products - Expenses");
    }
    @FXML
    void orderPageBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        SwitchingPages.switchPage(event,"OrderPage.fxml","Management products - Order");
    }
    @FXML
    void CategoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CategoryPage.fxml","Management products - Category");
    }
    @FXML
    void statisticsBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml","Management products - Statistics");
    }
    @FXML
    void CustomerPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CustomerPage.fxml","Management products - Customer");
    }
    @FXML
```

```

void ShelvesBtn(ActionEvent event) throws IOException {
    SwitchingPages.switchPage(event,"ShelvesPage.fxml","Management products - Shelves");
}
@FXML
void logoutBtn(ActionEvent event) throws IOException {
    SwitchingPages.openWindowInCenter("Login",event,"Management products - Login");
}
@FXML
void addProductToInventory(ActionEvent event) throws IOException, SQLException, ClassNotFoundException {
    FXMLLoader loader = new FXMLLoader();
    loader.setLocation(getClass().getResource("addProduct.fxml"));
    Stage window = new Stage(StageStyle.UNDECORATED);
    window.setScene(new Scene(loader.load()));
    AddProductController controller = loader.getController();
    controller.setController(this);
    window.initModality(Modality.APPLICATION_MODAL);
    window.show();
}
//Edit product button
@FXML
void editProductBtn(ActionEvent event) throws IOException, SQLException, ClassNotFoundException, InterruptedException {
    // if user selected product from the table
    if(!(productsTable.getSelectionModel().isEmpty())) {
        Product selectedProduct = productsTable.getSelectionModel().getSelectedItem(); // get selected product
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("EditProductPage.fxml"));
        Stage window = new Stage(StageStyle.UNDECORATED);
        window.setScene(new Scene(loader.load()));
        EditProductController controller = loader.getController();
        controller.setController(this);
        controller.setSelectedProductToFields(selectedProduct);
        window.initModality(Modality.APPLICATION_MODAL);
        window.show();
    } else
        Message.showMessageDialogWithOKButton(AlertType.NONE,"You have to choose product from table", "Update product");
}

// this function update product and refreshing the products table
public void editProductFromTable(Product product) throws SQLException, ClassNotFoundException {
    DatabaseManager.updateProduct(product); // update product
    refreshTable(DatabaseManager.getProducts()); //refresh products table
}
//side
private void sideManaged() {
    side.setTranslateX(-215);
    side.setVisible(false);
    side.setManaged(false);
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    try {
        refreshTable(DatabaseManager.getProducts());
        sideManaged();
        Conditions.checkManagerStatus(expensesBtn);
        DatabaseManager.removeShoppingCart();
    } catch (ClassNotFoundException | SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

//set all columns of the table
public void refreshTable(ObservableList<Product> list) throws ClassNotFoundException {
    productIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("productCode"));
    productNameCol.setCellValueFactory(new PropertyValueFactory<Product,String>("productName"));
    categoryCol.setCellValueFactory(new PropertyValueFactory<Product,String>("Category"));
    priceCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("price"));
    quantityCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("quantity"));
    supplierPriceCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("supplierPrice"));
    weightCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("weight"));
    supplierPriceCol.setSortable(false);
    priceCol.setSortable(false);
    productsTable.setItems(list);
    new RowTable().changeRowsColor(productsTable);
}
// menu clicked
@FXML

```

```

void menuClicked(MouseEvent event) {
    new AnimationClass().sideAnimation(side);
}
// this function add product to inventory
public void addProductToTable(Product product) throws SQLException, ClassNotFoundException {
    productList = DatabaseManager.getProducts();// get all products in inventory
    // checks if the product code and product name is not exists in the table
    if(product.productNameIsTaken(product.productList)==true && !product.checkProductIsExists(productList)) {
        DatabaseManager.addProduct(product); // add product to the table
        refreshTable(DatabaseManager.getProducts()); // refresh a products table
    }else {
        // if a product name and product code is exists
        if(product.productNameIsTaken(product.productList) == false && product.checkProductIsExists(productList))
            Message.showMessageDialogWithOKButton(AlertType.WARNING,"Product name and product code is taken","");
        // if product name is exists else if(product.productNameIsTaken(product.productList)==false)
        Message.showMessageDialogWithOKButton(AlertType.WARNING,"Product name is taken","");
        // if product code is exists
        else if(!product.checkProductIsExists(productList))
            Message.showMessageDialogWithOKButton(AlertType.WARNING,"Product code is taken","");
    }
}

//this function remove the product from the table
@FXML
void removeProduct(ActionEvent event) throws SQLException, ClassNotFoundException {
    if(!(productsTable.getSelectionModel().isEmpty())) { // if user selected product from the table
        Product selectedProduct = productsTable.getSelectionModel().getSelectedItem();
        DatabaseManager.deleteProduct(selectedProduct, "product"); // delete the product
        refreshTable(DatabaseManager.getProducts());// refresh the products table
    }else
        Message.showMessageDialogWithOKButton(AlertType.NONE,"You have to choose product from table", "Remove product");
}

// search product from the table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty())
        refreshTable(DatabaseManager.getProducts());
    else
        refreshTable(Conditions.getItemssSearch(DatabaseManager.getProducts(),searchField.getText()));
}
}

```

## קובץ AddProductController.java

המחלקה הזאת מציגה את החלון הוספת מוצרים למלאי.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.Event;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;

public class AddProductController implements Initializable{

    @FXML
    private ComboBox<String> choiceBoxCategory;

    @FXML
    private TextField productNameField,priceField,productCodeField,quantityField,supplierPrice,weightField;

    @FXML
    private Label quantityLabel,clientPriceLabel,supplierPriceLabel,productCodeLabel,weightLabel;

    InventoryController controller;

    // add all categories in combo box
    private void setCategories() throws ClassNotFoundException, SQLException {
        ObservableList<Category> categoryList = DatabaseManager.getCategories(); // get all categories
        for(int i=0;i<categoryList.size();i++)
            choiceBoxCategory.getItems().add(categoryList.get(i).getCategoryName()); // add category to combo box
    }

    @FXML
    void addProductToInventory(ActionEvent event) throws SQLException, ClassNotFoundException, IOException {
        Field field = new Field();
        if(field.checkEmptyFields(quantityField, productNameField, productCodeField, priceField,supplierPrice) &&
            !(choiceBoxCategory.getSelectionModel().isEmpty())) {

            Boolean codeFlag = field.checkNumberField(productCodeField, productCodeLabel); // check product code field is valid
            Boolean quantityFlag = field.checkNumberField(quantityField, quantityLabel); // check quantity field is valid
            Boolean clientPriceFlag = field.checkFloatNumberField(priceField, clientPriceLabel); // check client price field is valid
            Boolean supplierPriceFlag = field.checkFloatNumberField(supplierPrice, supplierPriceLabel); // check supplier price field is valid
            Boolean weightFlag = field.checkFloatNumberField(weightField, weightLabel); // check weight field is valid

            if(codeFlag && quantityFlag && clientPriceFlag && supplierPriceFlag && weightFlag) { // if everything is correct the condition exists
                Product product = new Product(Integer.parseInt(productCodeField.getText()),productNameField.getText(),
                    choiceBoxCategory.getSelectionModel().getSelectedItem(), Conditions.get2DigitsAfterPoint(supplierPrice.getText()),
                    Conditions.get2DigitsAfterPoint(priceField.getText()),Integer.parseInt(quantityField.getText()), Double.parseDouble(weightField.getText()));

                if(product.checkSupplierPrice(product) == false) { // check if the supplier price bigger than client price
                    if(Message.showMessageDialogWithYesNoButtons(AlertType.WARNING,"Are you sure? , supplier price bigger than client price", null) ==
                        buttonType.YES) {
                        controller.addProductToTable(product); // then sending the object to the function
                        exitStage(event); // and exit the window
                    }
                }
            } else {
                controller.addProductToTable(product); // then sending the object to the function
                exitStage(event); // and exit the window
            }
        } else
            Message.showMessageDialogWithOKButton(AlertType.ERROR, "The fields is empty", "Empty field");
    }

    // on key typed event for product code
    @FXML
    void productCodeTyped(KeyEvent event) {
        if(!Conditions.checkStringLength(productCodeField.getText(), 6)) // if string length of the product code field bigger than 6
    }
```

```

        Conditions.removeLastIndexInString(productCodeField); // remove last index in the field
    }
    // on key typed event for product name
    @FXML
    void productNameTyped(KeyEvent event) {if(!Conditions.checkStringLength(productNameField.getText(), 15)) // if string length of the
product name field bigger than 15
        Conditions.removeLastIndexInString(productNameField); // remove last index in the field
    }
    // on key typed event for quantity field
    @FXML
    void quantityTyped(KeyEvent event) {
        if(!Conditions.checkStringLength(quantityField.getText(), 6)) // if string length of the quantity field bigger than 6
            Conditions.removeLastIndexInString(quantityField); // remove last index in the field
    }
    // exit button
    @FXML
    void exitBtn(MouseEvent event) {
        exitStage(event);
    }
    // exit the window
    private void exitStage(Event event) {
        Node source = (Node) event.getSource();
        Stage stage = (Stage) source.getScene().getWindow();
        stage.close();
    }
    // set inventory controller
    public void setController(InventoryController controller) {
        this.controller = controller;
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        // TODO Auto-generated method stub
        try {
            setCategorys();
        } catch (ClassNotFoundException | SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

## קובץ EditProductController.java

המחלקה הזאת מציגה את החלון של עדכון מוצרים.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;

import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;

public class EditProductController implements Initializable{

    @FXML
    private TextField productNameField,priceField,productCodeField,quantityField,supplierPriceField,weightField;

    @FXML
    private ComboBox<String> choiceBoxCategory;

    @FXML
    private Label supplierPriceLabel,clientPriceLabel,quantityLabel,weightLabel;

    InventoryController controller;

    // exit button
    @FXML
    void exitBtn(MouseEvent event) {
        SwitchingPages.exitStage(event); //close the window
    }

    // edit product button
    // this function edit product details from products table
    @FXML
    void editProductFromTable(ActionEvent event) throws IOException, SQLException, ClassNotFoundException {
        Field field = new Field();
        // if all fields is not empty
        if(field.checkEmptyFields(quantityField, productNameField, priceField,supplierPriceField) &&
        !(choiceBoxCategory.getSelectionModel().isEmpty())) {

            Boolean quantityFlag = field.checkNumberField(quantityField, quantityLabel);
            Boolean clientPriceFlag = field.checkFloatNumberField(priceField, clientPriceLabel);
            Boolean supplierPriceFlag = field.checkFloatNumberField(supplierPriceField, supplierPriceLabel);
            Boolean weightFlag = field.checkFloatNumberField(weightField, weightLabel);

            // check if all fields is valid
            if(quantityFlag && clientPriceFlag && supplierPriceFlag && weightFlag) {
                // if the all fields is valid
                Product product = new Product(Integer.parseInt(productCodeField.getText()),productNameField.getText(),
                choiceBoxCategory.getSelectionModel().getSelectedItem(),Conditions.get2DigitisAfterPoint(supplierPriceField.getText()),
                Conditions.get2DigitisAfterPoint(priceField.getText()),Integer.parseInt(quantityField.getText()),Double.parseDouble(weightField.getText()));
                // check if supplier price bigger than client price
                if(product.checkSupplierPrice(product) == false) {
                    // if the supplier price bigger than client price receive a message with yes and no button
                    if(Message.showMessagewithYesNoButtons(AlertType.WARNING,"Are you sure? , supplier price bigger than client
                    price", null) == buttonType.YES) {
                        // if the user clicked yes edit the product details
                        controller.editProductFromTable(product);
                        SwitchingPages.exitStage(event); // close the window
                    }
                } else { // if client price bigger than supplier price update the product details and close the window
                    controller.editProductFromTable(product);
                    SwitchingPages.exitStage(event); // exit
                }
            }
        } else // if you have empty fields
```

```
Message.showMessageDialog(AlertType.ERROR, "The fields is empty", "Empty fields");
```

```
    }

    // set all details of the product in the fields
    public void setSelectedProductToFields(Product product) {
productCodeField.setText(String.valueOf(product.getProductCode())); // product code
productNameField.setText(product.getProductName()); // product name
priceField.setText(String.valueOf(product.getPrice())); // client price
quantityField.setText(String.valueOf(product.getQuantity())); // quantity
weightField.setText(String.valueOf(product.getWeight())); // weight
choiceBoxCategory.getSelectionModel().select(product.getCategory()); // category
supplierPriceField.setText(String.valueOf(product.getSupplierPrice())); // supplier price
    }

    // set all categories name in combo box
    public void setCategoriesForCamboBox() throws ClassNotFoundException, SQLException {
ObservableList<Category> categorys = DatabaseManager.getCategorys(); // get all categories
for(int i = 0; i < categorys.size(); i++)
choiceBoxCategory.getItems().add(categorys.get(i).getCategoryName()); // add category to combo box
    }

    // product name field -> on key typed
    @FXML
    void productNameTyped(KeyEvent event) {
if(!Conditions.checkStringLength(productNameField.getText(), 15)) // if the length of the string in the field bigger than 15
Conditions.removeLastIndexInString(productNameField); // remove last index in the field
    }
    // quantity name field -> on key typed
    @FXML
    void quantityTyped(KeyEvent event) {
if(!Conditions.checkStringLength(quantityField.getText(), 6)) // if the length of the string in the field bigger than 6
Conditions.removeLastIndexInString(quantityField); // remove last index in the field
    }

    // set inventory controller
    public void setController(InventoryController controller) {
this.controller = controller;
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
// TODO Auto-generated method stub
try {
setCategoriesForCamboBox();
} catch (ClassNotFoundException | SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
    }
}
```



## קובץ CategoryController.java

המחלקה הזאת מציגה את החלון של הקטגוריות.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

public class CategoryController implements Initializable{

    @FXML
    private TableView<Category> categoryTable;
    @FXML
    private TableColumn<Category, String> categoryName;

    @FXML
    private TableColumn<Category, Integer> categoryIdCol;

    @FXML
    private TextField categoryNametxt;

    ObservableList<Category> categoryList;

    @FXML
    private TextField searchField;

    @FXML
    private AnchorPane side;

    @FXML
    private Button expensesBtn;

    /* switch page */
    @FXML
    void expensesBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"Expenses.fxml","Management Product - Expenses");
    }
    @FXML
    void CustomerPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CustomerPage.fxml","Management Product - Customer");
    }
    @FXML
    void inventoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"inventory.fxml","Management Product - Inventory");
    }
    @FXML
    void orderPageBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"OrderPage.fxml","Management Product - Order");
    }
    @FXML
    void statisticsBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml","Management Product - Statistics");
    }
    @FXML
    void shelvesBtn(ActionEvent event) throws IOException{
        SwitchingPages.switchPage(event, "shelvesPage.fxml","Management Product - Shelves");
    }
}
```

```

    }

    @FXML
    void addProductToInventory(ActionEvent event) throws IOException {
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("addCategoryPage.fxml"));
        Stage Window = new Stage(StageStyle.UNDECORATED);
        Window.setScene(new Scene(loader.load()));
        AddCategoryPageController controller = loader.getController();
        controller.setController(this);
        Window.initModality(Modality.APPLICATION_MODAL);
        Window.show();
    }

    // edit category button
    @FXML
    void editCategoryBtn(ActionEvent event) throws IOException {
        if(!(categoryTable.getSelectionModel().isEmpty())) { // i user selected category from the table
            // open edit window
            Category selectedCategory = categoryTable.getSelectionModel().getSelectedItem();
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(getClass().getResource("EditCategoryPage.fxml"));
            Stage Window = new Stage(StageStyle.UNDECORATED);
            Window.setScene(new Scene(loader.load()));
            EditCategoryPageController controller = loader.getController();
            controller.setController(this);
            controller.setCategoryNameInField(selectedCategory);
            Window.initModality(Modality.APPLICATION_MODAL);
            Window.show();
        } else
            Message.showMessageDialogWithOKButton(AlertType.NONE, "You have to choose category from table", "Update category");
    }

    @FXML
    void logoutBtn(ActionEvent event) throws IOException {
        SwitchingPages.openWindowInCenter("Login", event, "Management products - Login");
    }
//side
private void sideManaged() {
    side.setTranslateX(-215);
    side.setVisible(false);
    side.setManaged(false);
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    // TODO Auto-generated method stub
    try {
        refreshTable(DatabaseManager.getCategorys());
        sideManaged();
        Conditions.checkManagerStatus(expensesBtn);
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}

//refresh categories table
public void refreshTable(ObservableList<Category> list) throws ClassNotFoundException {
    new RowTable().changeRowsColor(categoryTable);
    categoryTable.setItems(list);
    categoryName.setCellValueFactory(new PropertyValueFactory<Category,String>("categoryName"));
    categoryIdCol.setCellValueFactory(new PropertyValueFactory<Category,Integer>("categoryId"));
}

//menu clicked
@FXML
void menuClicked(MouseEvent event) {
    new AnimationClass().sideAnimation(side);
}

//add category button
public void addCategory(Category category) throws SQLException, ClassNotFoundException {
    categoryList = DatabaseManager.getCategorys();
    if(category.categoryNameIsTaken(category, categoryList) == true) { //if the category code or category name is taken
        DatabaseManager.addCategory(category); //add category to the table in the data base if the category is not taken
        refreshTable(DatabaseManager.getCategorys());
    }
    else// if the category is taken
        Message.showMessageDialogWithOKButton(AlertType.WARNING, "This category is exists.", "");
}

public void editCategory(Category category, Category oldCategory) throws SQLException, ClassNotFoundException {
    categoryList = DatabaseManager.getCategorys();
    if(category.categoryNameIsTaken(category, categoryList) == true) { // checks is the category is taken
        DatabaseManager.editCategory(category); // if not taken edit the data of the category
    }
}

```

```

        DatabaseManager.updateProductCategory(category,oldCategory);//and update all the products belong to the
category
        refreshTable(DatabaseManager.getCategorys());
    }
    else // if category is exists
        Message.showMessageBoxWithOKButton(AlertType.WARNING,"This category is exists.", "");
    }

    //remove category button
@FXML
void removeCategoryBtn(ActionEvent event) throws SQLException, ClassNotFoundException {
    Category selectedCategory = categoryTable.getSelectionModel().getSelectedItem();
    ObservableList<Product> productsList = DatabaseManager.getProducts();
    selectedCategory.setProductList(productsList);
    if(!selectedCategory.checksCategory(selectedCategory))//check if the category not contains products
        DatabaseManager.deleteCategory(selectedCategory);// if the category not contains products delete the category
    else // if the category contains product
        Message.showMessageBoxWithOKButton(AlertType.ERROR, "You can not delete the category because there are products in
the category", "");

    refreshTable(DatabaseManager.getCategorys());
}

//search specific category in the table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty())
        refreshTable(DatabaseManager.getCategorys());
    else
        refreshTable(Conditions.getItemssSearch(DatabaseManager.getCategorys(),searchField.getText()));
}
}

```

## קובץ AddCategoryPageController.java

המחלקה הזאת מציגה את החלון של הוספת קטגוריות.

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;

public class AddCategoryPageController{

    @FXML
    private TextField categoryNameField;

    CategoryController controller;

    // exit button
    @FXML
    void exitBtn(MouseEvent event) {
        SwitchingPages.exitStage(event);
    }

    //add category button
    @FXML
    void addNewCategoryButton(ActionEvent event) throws SQLException, IOException, ClassNotFoundException {

        if (Message.showMessageWithYesNoButtons(AlertType.NONE,"are you sure?", "New category") == buttonType.YES) { // message yes or
no
            if(new Field().checkEmptyFields(categoryNameField)) {
                Category category = new Category(categoryNameField.getText());
                SwitchingPages.exitStage(event);
                controller.addCategory(category);//add new category
            }else
                Message.showMessageWithOKButton(AlertType.ERROR, "The category name field is empty", "Empty fields");

        }
    }

    //this function check if the length of the category name content bigger than 15
    //max length of the category name is 15
    @FXML
    void categoryNameTyped(KeyEvent event) {
        if(!Conditions.checkStringLength(categoryNameField.getText(), 15))
            Conditions.removeLastIndexInString(categoryNameField);// if true remove last index
    }

    // set category controller
    public void setController(CategoryController controller) {
        this.controller = controller;
    }
}
```

## קובץ EditCategoryPageController.java

המחלקה הזאת מציגה את החלון של עדכון קטגוריות.

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import javafx.event.ActionEvent;
import javafx.event.Event;
import javafx.fxml.FXML;
import javafx.scene.Node;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;

public class EditCategoryPageController{

    @FXML
    private TextField categoryNameField,categoryIdfield;

    CategoryController controller;

    Category category;
    // update category button
    @FXML
    void updateCategoryBtn(ActionEvent event) throws IOException, SQLException, ClassNotFoundException {
        if(new Field().checkEmptyFields(categoryNameField)) { //if field is empty
            if(!category.getCategoryName().equals(categoryNameField.getText())) { //if the category name is changed
                Category newCategory = new Category(Integer.parseInt(categoryIdfield.getText()),categoryNameField.getText());
                exitStage(event);
                controller.editCategory(newCategory,category); //edit category
            } else
                Message.showMessageDialogWithOKButton(AlertType.WARNING, "the category name not changed", null);
        } else
            Message.showMessageDialogWithOKButton(AlertType.ERROR,"The category name field is empty","Empty field");
    }
    //exit button
    @FXML
    void exitBtn(MouseEvent event) {
        exitStage(event);
    }
    //exit the window
    public void exitStage(Event event){
        Node source = (Node) event.getSource();
        Stage stage = (Stage) source.getScene().getWindow();
        stage.close();
    }
    //this function checks the length of the category name field
    @FXML
    void categoryNameTyped(KeyEvent event) {
        if(!Conditions.checkStringLength(categoryNameField.getText(), 15))
            Conditions.removeLastIndexInString(categoryNameField); // if true remove last index
    }

    public void setCategoryNameInField(Category category) {
        this.category = category;
        categoryIdfield.setText(String.valueOf(category.getCategoryId())); // set category id in field
        categoryNameField.setText(category.getCategoryName()); //set category name in field
    }
    //set controller
    public void setController(CategoryController controller) {
        this.controller = controller;
    }
}
```

## קובץ CustomerController.java

המחלקה הזאת מציגה את החלון של הלקוחות .

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;

public class CustomerController implements Initializable{

    @FXML
    private TextField fullnamefield,idfield,telephonefield,emailfield,addressfield;

    @FXML
    private Label fullNameLabel,idLabel,telephoneLabel,emailLabel,addressLabel;

    @FXML
    private AnchorPane side;

    @FXML
    private Button expensesBtn;

    /* switch window */
    @FXML
    void expensesBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"Expenses.fxml","Management products - Expenses");
    }
    @FXML
    void CategoryPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CategoryPage.fxml","Management products - Category");
    }
    @FXML
    void inventoryPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"inventory.fxml","Management products - Inventory");
    }
    @FXML
    void orderPageBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"OrderPage.fxml","Management products - Order");
    }
    @FXML
    void CustomerListPage(MouseEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CustomerList.fxml","Management products - Customers list");
    }
    @FXML
    void statisticsBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"statistics.fxml","Management products - Statistics");
    }
    @FXML
    void shelvesBtn(ActionEvent event) throws IOException{
        SwitchingPages.switchPage(event,"shelvesPage.fxml","Management products - shelves");
    }
    @FXML
    void logoutBtn(ActionEvent event) throws IOException {
        SwitchingPages.openWindowInCenter("Login", event,"Management products - Login");
    }
    private void sideManaged() {
        side.setTranslateX(-215);
        side.setVisible(false);
        side.setManaged(false);
    }
    // INIT
    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        try {
            Conditions.checkManagerStatus(expensesBtn);
        } catch (ClassNotFoundException | SQLException e) {
```

```

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    sideManaged();
}
// this function add customer to the customers list in database
@FXML
void addCustomerBtn(ActionEvent event) throws SQLException, ClassNotFoundException {
    Field field = new Field();
    // if all fields is not empty
    if(field.checkEmptyFields(telphonefield, idfield, fullnamefield, emailfield, addressfield)) {

        field.changeBorderColor(telphonefield, idfield, fullnamefield, emailfield, addressfield);

        Boolean fullNameFlag = field.checkFullNameField(fullnamefield, fullNameLabel);
        Boolean telephoneFlag = field.checkPhoneNumberField(telphonefield, telephoneLabel);
        Boolean idFlag = field.checkIdField(idfield, idLabel, "black");
        Boolean emailFlag = field.checkEmailField(emailfield, emailLabel, "black");
        Boolean addressFlag = field.checkAddressField(addressfield, addressLabel);
        // checks all field is valid
        if(fullNameFlag && telephoneFlag && idFlag && emailFlag && addressFlag) {

            if (Message.showMessageDialogWithYesNoButtons(AlertType.NONE, "are you sure?", "New customer") == buttonType.YES) {

                Customer customer = new Customer(fullnamefield.getText(), Integer.parseInt(idfield.getText()),
                    ,emailfield.getText(),telphonefield.getText(),addressfield.getText());

                // check if id of the customer is exists
                if(customer.customerIdsTaken(customer, DatabaseManager.getCustomer())==true)
                    DatabaseManager.addCustomer(customer); // if id not exists add customer to customers list
                else // if the customer id is exists receive a message
                    Message.showMessageDialogWithOKButton(AlertType.WARNING, "this customer is exists.", "");
            }
        }
    }
    // change color to the empty fields
    field.changeBorderColor(telphonefield, idfield, fullnamefield, emailfield, addressfield);
    Message.showMessageDialogWithOKButton(AlertType.ERROR, "You have empty fields", "Empty fields");
}
}
// menu clicked
@FXML
void menuClicked(MouseEvent event) {
    new AnimationClass().sideAnimation(side);
}
// id field -> on key typed
@FXML
void idTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(idfield.getText(),9)) // if the string length in the field bigger than 9
        Conditions.removeLastIndexInString(idfield); // remove last index in the field
}
// address field -> on key typed
@FXML
void addressTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(addressfield.getText(),20)) // if the string length in the field bigger than 20
        Conditions.removeLastIndexInString(addressfield); // remove last index in the field
}
// full name field -> on key typed
@FXML
void fullNameTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(fullnamefield.getText(),20)) // if the string length in the field bigger than 20
        Conditions.removeLastIndexInString(fullnamefield); // remove last index in the field
}
// telephone field -> on key typed
@FXML
void telephoneTyped(KeyEvent event) {
    if(!Conditions.checkStringLength(telphonefield.getText(),12)) // if the string length in the field bigger than 12
        Conditions.removeLastIndexInString(telphonefield); // remove last index in the field
}
//this function for clear text fields when the user click on the clear button
@FXML
void clearBtn(ActionEvent event) {
    fullnamefield.setText("");
    idfield.setText("");
    telphonefield.setText("");
    emailfield.setText("");
    addressfield.setText("");
}
}
}

```

## קובץ CustomerListController.java

המחלקה הזאת מציגה את אשימת הלקוחות.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

public class CustomerListController implements Initializable {

    @FXML
    private TableView<Customer> customerTable;

    @FXML
    private TableColumn<Customer, String> fnamecol,addresscol,emailcol,telcol;

    @FXML
    private TableColumn<Customer, Integer> idcol;

    @FXML
    private TextField searchField;

    /* switch pages */
    @FXML
    void editCustomerBtn(ActionEvent event) throws IOException {
        // if the user selected customer from table
        if(!(customerTable.getSelectionModel().isEmpty())) {
            // open a new window (edit details of the customer)
            Customer selectedCustomer = customerTable.getSelectionModel().getSelectedItem();
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(getClass().getResource("EditCustomerPage.fxml"));
            Stage Window = new Stage(StageStyle.UNDECORATED);
            Window.setScene(new Scene(loader.load()));
            EditCustomerPageController controller = loader.getController();
            controller.setController(this);
            controller.setCustomerDetailsInField(selectedCustomer);
            Window.initModality(Modality.APPLICATION_MODAL);
            Window.show();
        } else
            Message.showMessageDialog(AlertType.NONE,"You have to choose Customer from table", "Update
Customer");
    }

    @FXML
    void backToCustomerPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "CustomerPage.fxml","Management products - Customer");
    }

    // remove customer button
    // this function remove customer from the customers table
    @FXML
    void removeCustomerFromList(ActionEvent event) throws SQLException, ClassNotFoundException {
        Customer selectedCustomer = customerTable.getSelectionModel().getSelectedItem();
        DatabaseManager.removeCustomer(selectedCustomer); // remove the customer from the database
        refreshTable(DatabaseManager.getCustomer()); // refresh customers table
    }
    // edit details of the customer
    public void editPersonalDetailsForCustomer(Customer customer) throws SQLException, ClassNotFoundException {
        DatabaseManager.updateCustomer(customer); // update customer
    }
}
```



```

        refreshTable(DatabaseManager.getCustomer()); // refresh customers table
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        // TODO Auto-generated method stub
        try {
            refreshTable(DatabaseManager.getCustomer());
        } catch (ClassNotFoundException | SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    // refresh customers table
    public void refreshTable(ObservableList<Customer> list) throws ClassNotFoundException {
        fnamecol.setCellValueFactory(new PropertyValueFactory<Customer,String>("fullName"));
        idcol.setCellValueFactory(new PropertyValueFactory<Customer,Integer>("id"));
        addresscol.setCellValueFactory(new PropertyValueFactory<Customer,String>("address"));
        emailcol.setCellValueFactory(new PropertyValueFactory<Customer,String>("email"));
        telcol.setCellValueFactory(new PropertyValueFactory<Customer,String>("telephone"));
        customerTable.setItems(list);
    }

    // search field
    // search customer from the table
    @FXML
    void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
        if(searchField.getText().isEmpty())
            refreshTable(DatabaseManager.getCustomer());
        else
            refreshTable(Conditions.getItemssSearch(DatabaseManager.getCustomer(),searchField.getText()));
    }
}

```

## קובץ orderController.java

המחלקה הזאת מציגה את חלון ההזמנות.

```
package application;
import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
public class orderController implements Initializable {

    @FXML
    private TableView<Customer> customerTable;

    @FXML
    private TableColumn<Customer, String> fnamecol,emailcol,telcol,addresscol;

    @FXML
    private TableColumn<Customer, Integer> idcol;

    @FXML
    private TextField customernamefield,customeridfield,searchProductField,searchCustomerField;

    @FXML
    private TableView<Product> productsTable;

    @FXML
    private TableColumn<Product, Integer> productIdCol,quantityCol,shelfIdCol;

    @FXML
    private TableColumn<Product, String> categoryCol,productNameCol;

    @FXML
    private TableColumn<Product, Double> priceCol;

    @FXML
    private Button chooseCustomerFromTableBtn,shoppingCartBtn,btn,expensesBtn;

    @FXML
    private Label shoppingCartLabel,addLabels;
    Customer customerSelected;
    ObservableList<Product> productsList;
    @FXML
    private AnchorPane side;

    /* switch page */
    @FXML
    void expensesBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "Expenses.fxml","Management products - Expenses");
    }

    @FXML
    void orderListPage(MouseEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(checksShoppingCart() == true)
            SwitchingPages.switchPage(event,"orderListPage.fxml","Management products - Orders list");
    }
    @FXML
    void statisticsBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(checksShoppingCart() == true)
            SwitchingPages.switchPage(event, "statistics.fxml","Management products - Statistics");
    }
    @FXML
    void CategoryBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
```

```

        if(checksShoppingCart() == true)
            SwitchingPages.switchPage(event,"CategoryPage.fxml","Management products - Category");
    }
    @FXML
    void CustomerPage(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(checksShoppingCart() == true)
            SwitchingPages.switchPage(event,"CustomerPage.fxml","Management products - Customer");
    }
    @FXML
    void inventoryPageBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(checksShoppingCart() == true)
            SwitchingPages.switchPage(event,"inventory.fxml","Management products - Inventory");
    }
    @FXML
    void shelvesBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(checksShoppingCart() == true)
            SwitchingPages.switchPage(event, "shelvesPage.fxml","Management products - Shelves");
    }
    @FXML
    void logoutBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(checksShoppingCart() == true)
            SwitchingPages.openWindowInCenter("Login",event,"Management products - Login");
    }
}

@FXML
void shoppingCartBtn(MouseEvent event) throws IOException {
    FXMLLoader loader = new FXMLLoader();
    loader.setLocation(getClass().getResource("Shopping_cart.fxml"));
    Stage shoppingCartWindow = (Stage)((Node)event.getSource()).getScene().getWindow();
    shoppingCartWindow.setScene(new Scene(loader.load()));
    ShoppingCartController controller = loader.getController();
    shoppingCartWindow.setTitle("Management products - Shopping cart");
    customerSelected = new Customer(customernamefield.getText(),Integer.parseInt(customeridfield.getText()));
    controller.setCustomerSelected(customerSelected);
    shoppingCartWindow.show();
}

/* set columns to the tables */
//set columns of the product table
private void refreshProductsTable(ObservableList<Product> productList) throws ClassNotFoundException {
    productIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("productCode"));
    productNameCol.setCellValueFactory(new PropertyValueFactory<Product,String>("productName"));
    categoryCol.setCellValueFactory(new PropertyValueFactory<Product,String>("Category"));
    priceCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("price"));
    quantityCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("quantity"));
    shelfIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("shelfNumber"));
    productsTable.setItems(productList);//set all the products in the database
    new RowTable().changeRowsColor(productsTable);
}

//set columns of the customers table
private void refreshCustomerTable(ObservableList<Customer> customerList) throws ClassNotFoundException {
    fNameCol.setCellValueFactory(new PropertyValueFactory<Customer,String>("fullName"));
    idCol.setCellValueFactory(new PropertyValueFactory<Customer,Integer>("id"));
    addressCol.setCellValueFactory(new PropertyValueFactory<Customer,String>("address"));
    emailCol.setCellValueFactory(new PropertyValueFactory<Customer,String>("email"));
    telCol.setCellValueFactory(new PropertyValueFactory<Customer,String>("telephone"));
    customerTable.setItems(customerList);//set all the customers in the database
}

// this function checks if the user clicked yes clear a shopping cart
private Boolean checksShoppingCart() throws ClassNotFoundException, SQLException {
    if(!DatabaseManager.getProductFromShoppingCart().isEmpty()) { // checks if the shopping cart is not empty
        if(Message.showMessageBoxWithYesNoButtons(AlertType.WARNING, "Are you sure? , if you go through a page you will delete the products in the shopping cart", "") == ButtonType.YES) {
            // if the user chosen yes in the message return all product to shelves
            LogoutClass.returnProductsToShelves();
            DatabaseManager.removeShoppingCart(); // remove all product in shopping cart
            return true;
        }
        // if user chosen no in message return false
        return false;
    }
    // if shopping cart empty return true
    return true;
}

//this function disable add and the chopping cart label in accordance the flag

```

```

private void disableLabels(Boolean flag) {
    addLabels.setDisable(flag);
    shoppingCartLabel.setDisable(flag);
}
// disable shopping cart and add labels
private void chekcsCustomerFieldsIsEmpty() throws ClassNotFoundException, SQLException {
    // if customer fields is empty disable the labels
    if(customeridfield.getText().isEmpty()) {
        disableLabels(true);
    } else // if customer fields is not empty activate the labels
        disableLabels(false);
}
//side
private void sideManaged() {
    side.setTranslateX(-215);
    side.setVisible(false);
    side.setManaged(false);
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    // TODO Auto-generated method stub
    try {
        refreshCustomerTable(DatabaseManager.getCustomer()); // refresh customer table
        refreshProductsTable(DatabaseManager.getProductsInShelves());
        chekcsCustomerFieldsIsEmpty();
        sideManaged();
        Conditions.checkManagerStatus(expensesBtn);
    } catch (ClassNotFoundException | SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
// menu clicked
@FXML
void menuClicked(MouseEvent event) {
    new AnimationClass().sideAnimation(side);
}

//setCustomerSelectedInFields function set the customer name and id in fields
public void setCustomerSelectedInFields(Customer customer) throws ClassNotFoundException, SQLException {
    customernamefield.setText(customer.getFullName()); // set name of customer in the customer name field
    customeridfield.setText(String.valueOf(customer.getId())); // set id of customer in the customer id field
    chekcsCustomerFieldsIsEmpty();
}
// add customer button
//chooseCustomerFromTableBtn function selects the customer from table
@FXML
void chooseCustomerFromTableBtn(ActionEvent event) throws SQLException, NumberFormatException, ClassNotFoundException {
    if(!(customerTable.getSelectionModel().isEmpty())) { // checks if the user selected customer from the table
        customerSelected = customerTable.getSelectionModel().getSelectedItem(); // get the selected customer object
        if(!(customernamefield.getText().isEmpty()) && !(customeridfield.getText().isEmpty())) { // checks if the field is not
empty
            if(DatabaseManager.getProductFromShoppingCart().isEmpty()) { // if the shopping cart is empty
                setCustomerSelectedInFields(customerSelected);
            } else if(customerSelected.getId() != Integer.parseInt(customeridfield.getText())) { // if the selected customer is
not equal to current customer
                //Receive a message
                if(Message.showMessageDialogWithYesNoButtons(AlertType.WARNING, "If you change the customer you will lose
all the products that are in the shopping cart", "") == buttonType.YES) {
                    // if user clicked yes in the message
                    setCustomerSelectedInFields(customerSelected); // change customer
                    LogoutClass.returnProductsToShelves(); // and return all product in shopping cart to shopping cart
                    DatabaseManager.removeShoppingCart(); // remove all product in shopping cart
                    refreshProductsTable(DatabaseManager.getProductsInShelves()); // refresh the table
                }
            }
        } else // if the fields is empty
            setCustomerSelectedInFields(customerSelected);
        // activate shopping cart and add labels
        disableLabels(false);
    } else
        Message.showMessageDialogWithOKButton(AlertType.WARNING, "Please choose customer from table", "Customer selection");
}

// this function insert product to shopping cart
private void insertProductToShoppingCart(Product productSelected, Product shoppingCartProduct, Shelf shelf, int quantity) throws
ClassNotFoundException, SQLException {
    Order order = new Order();
    order.setProductsList(DatabaseManager.getProductFromShoppingCart());
}

```

```

        // if inserted product is exists in shopping cart
        if(order.isProductExists(productSelected)) {
            // update the quantity of the product
            int productQuantity =
DatabaseManager.getProductQuantity("shopping_cart",productSelected.getProductCode());
            DatabaseManager.updateProduct("Shopping_cart", shoppingCartProduct, productQuantity + quantity);
        } else { // if inserted product is not exists in shopping cart
            // add the product to shopping cart
            shoppingCartProduct.setSupplierPrice(productSelected.getSupplierPrice());
            shoppingCartProduct.setWeight(productSelected.getWeight());
            shoppingCartProduct.setCategory(productSelected.getCategory());
            DatabaseManager.insertProductToShoppingCart(shoppingCartProduct);
        }
    }
    // add product to shopping cart button
    //this function add product to shopping cart
    @FXML
    void addProductToShoppingCart(MouseEvent event) throws SQLException, ClassNotFoundException, InterruptedException {
        // if the user selected product from the table
        if(!(productsTable.getSelectionModel().isEmpty())) {
            Product productSelected = productsTable.getSelectionModel().getSelectedItem();
            productSelected.setShelvesList(DatabaseManager.getShelves());
            Shelf shelf = productSelected.getShelf(productSelected.getShelfNumber()); // get the shelf of the selected product

            // check if the quantity of selected product bigger than zero
            if(productSelected.getQuantity() > 0) {
                int quantity = Message.showMessageDialog("Product quantity", "Please enter product quantity :"); // get the number of
message with text field
                Product shoppingcart = new Product(productSelected.getProductCode(),productSelected.getProductName(),quantity
                    ,productSelected.getPrice());
                // if the string returned from the message bigger than zero
                if(quantity > 0) {
                    // if the product is zero receive a message
                    if(!productSelected.isProductQuantityZero(productSelected.getQuantity())){
                        Message.showMessageDialogWithOKButton(AlertType.WARNING,"You dont have more of this product","");
                    } else if(productSelected.checkProduct(quantity, productSelected.getQuantity())) { // if product quantity bigger than
selected quantity
                        //update product quantity in shelf according to selected quantity
                        DatabaseManager.updateProductInShelf("shelf_product",productSelected, shelf, productSelected.getQuantity()
- quantity);
                        insertProductToShoppingCart(productSelected,shoppingcart,shelf,quantity); // insert product to shopping cart
                    } else
                        // if selected product bigger than product quantity receive a message
                        if(Message.showMessageDialogWithYesNoButtons(AlertType.INFORMATION, "You have more " +
productSelected.getQuantity() + " in storage Do you want him?", "") == buttonType.YES) {
                            // if user clicked yes get all product quantity in shelf and add him to shopping cart
                            DatabaseManager.updateProductInShelf("shelf_product",productSelected, shelf, 0);
                            shoppingcart.setQuantity(productSelected.getQuantity());
                            insertProductToShoppingCart(productSelected,shoppingcart,shelf,productSelected.getQuantity());
                        }
                }
                refreshProductsTable(DatabaseManager.getProductsInShelves());
            } else // if the quantity of the selected product equal to zero receive a message
                Message.showMessageDialogWithOKButton(AlertType.WARNING,"You dont have more of this product","");
        } else // if the user not selected product from the table
            Message.showMessageDialogWithOKButton(AlertType.WARNING,"Please choose product `from table","Product selection");
    }

    //search product from the shelves
    @FXML
    void searchProductFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
        if(searchProductField.getText().isEmpty())
            refreshProductsTable(DatabaseManager.getProductsInShelves());
        else
            refreshProductsTable(Conditions.getItemssSearch(DatabaseManager.getProductsInShelves(),searchProductField.getText()));
    }

    //search customer from the customers table
    @FXML
    void searchCustomerFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
        if(searchCustomerField.getText().isEmpty())
            refreshCustomerTable(DatabaseManager.getCustomer());
        else
            refreshCustomerTable(Conditions.getItemssSearch(DatabaseManager.getCustomer(),searchCustomerField.getText()));
    }
}

```

## קובץ ShoppingCartController.java

המחלקה הזאת מציגה מה יש בתוך ההזמנה.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.Calendar;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;

public class ShoppingCartController implements Initializable{

    @FXML
    private TableView<Product> shoppingCartTable;

    @FXML
    private TableColumn<Product, String> productNameCol;

    @FXML
    private TableColumn<Product, Integer> quantityCol,productIdCol;

    @FXML
    private TableColumn<Product, Double> priceCol;

    @FXML
    private Label totalPriceLabel;

    Customer customerSelected;

    Calendar orderDate = null;
    Order orderAdded = null;

    @FXML
    private TextField searchField;

    /* switch pages*/
    @FXML
    void BackToOrderPageBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        // switch window to order window
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("OrderPage.fxml"));
        Stage shoppingCartWindow = (Stage)((Node)event.getSource()).getScene().getWindow();
        shoppingCartWindow.setScene(new Scene(loader.load()));
        orderController controller = loader.getController();
        controller.setCustomerSelectedInFields(customerSelected);
        shoppingCartWindow.show();
    }

    // remove product button
    // this function remove product from shopping cart
    @FXML
    void removeProductFromShoppingCartList(ActionEvent event) throws ClassNotFoundException, SQLException {
        // check if the user selected product from the table
        if(!(shoppingCartTable.getSelectionModel().isEmpty())) {
            Product selectedProduct = shoppingCartTable.getSelectionModel().getSelectedItem();
            selectedProduct.setShelvesList(DatabaseManager.getShelves());

            //receive a message if you want to remove the product
            if(Message.showMessageBoxWithYesNoButtons(AlertType.NONE,"Are you sure?","Delete product") == buttonType.YES) {
                removeProduct(selectedProduct);
                setTotalPrice(); // update total price after deleted the product
            }
        }
    }
}
```

```

        refreshTable(DatabaseManager.getProductFromShoppingCart()); // refresh table
    }
    }else // if user not selected product from the table
        Message.showMessageDialog(AlertType.NONE, "Please choose product from table", "Delete
product");
    }

    // remove product from the products table
    private void removeProduct(Product product) throws ClassNotFoundException, SQLException {
        // get quantity of the product in shelf

        // update products in shelf

        Shelf sh = new Shelf();
        sh.setProductsList(DatabaseManager.getProductsInShelves());
        Product []products = sh.getProductInShelves(product);
        String str = "";

        for(int i = 0; i < products.length;i++) {
            products[i].setShelvesList(DatabaseManager.getShelves());
            Shelf shelf = products[i].getShelf(products[i].getShelfNumber());
            int productQuantity = DatabaseManager.getProductQuantityInShelf("shelf_product",product,shelf);
            shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));

            if(shelf.checkWeight(product, product.getQuantity())) {
                DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity +
product.getQuantity());

                if(str != "")
                    str += " and";
                str += " " + product.getQuantity() + " of this product returns to shelf number : " +
shelf.getShelfNumber();

                product.setQuantity(0);
            } else {
                int quantity = shelf.getProductQunatity(products[i]);
                product.setQuantity(product.getQuantity() - quantity);
                DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity +
quantity);

                if(str != "")
                    str += " and";
                str += " " + quantity + " of this product returns to shelf number : " + shelf.getShelfNumber();
            }
            if(product.getQuantity() <= 0)
                break;
        }

        Message.showMessageDialog(AlertType.INFORMATION, str, null);

        DatabaseManager.deleteProduct(product,"shopping_cart"); // delete selected product in shelf
    }

    // this function set a total price of the order
    public void setTotalPrice() throws ClassNotFoundException, SQLException {
        Order order = new Order();
        // set total price of products in shopping cart

        totalPriceLabel.setText(String.valueOf(Conditions.get2DigitsAfterPoint(String.valueOf(order.getTotalPrice(DatabaseManager.getPro
ductFromShoppingCart())) + "₹"));
    }

    // edit product button
    // this function edit quantity of the product in shopping cart
    @FXML
    void editQuantityForProductBtn(ActionEvent event) throws SQLException, ClassNotFoundException, InterruptedException {
        // if user selected product is not empty
        if(!(shoppingCartTable.getSelectionModel().isEmpty())) {

            Product productSelected = shoppingCartTable.getSelectionModel().getSelectedItem();
            productSelected.setShelvesList(DatabaseManager.getShelves());
            String str = "";
            productSelected = productSelected.getCurrentProduct(DatabaseManager.getProductFromShoppingCart());
            // receive a message with text field
            int quantitySelected = Message.showMessageDialog("Product quantity","Please enter product quantity :");
            // get product quantity in the shelf
            int quantityInShelves = 0;

            // if the string in the message is valid
            if(quantitySelected > 0) {

```

```

        if(productSelected.getQuantity() != quantitySelected) { // if product quantity not equal to selected product
            if(productSelected.getQuantity() > quantitySelected) {
                int number = productSelected.getQuantity() - quantitySelected;
                for(Product product : DatabaseManager.getProductsInShelves()) {
                    if(product.getProductCode() == productSelected.getProductCode()) {
                        productSelected.setShelvesList(DatabaseManager.getShelves());
                        Shelf shelf = productSelected.getShelf(product.getShelfNumber());

                        shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));

                        int productQuantity =
DatabaseManager.getProductQuantityInShelf("shelf_product",productSelected,shelf);
                        if(shelf.checkWeight(productSelected, number)) {
                            DatabaseManager.updateProduct("shopping_cart",
productSelected, quantitySelected);

                            DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf, productQuantity + number); // update quantity in the
shelf

                            if(str != "")
                                str += " and";
                            str += " " + number + " of this product returns to shelf

number : " + shelf.getShelfNumber();

                            productSelected.setQuantity(0);
                            break;
                        } else {
                            int quantity =

shelf.getProductQunatity(productSelected);

                            productSelected.setQuantity(product.getQuantity() -
quantity);

                            number -= quantity;

                            DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity + quantity);
                            if(str != "")
                                str += " and";
                            str += " " + quantity + " of this product returns to shelf

number : " + shelf.getShelfNumber();

                            }
                        }
                    }
                }

                Message.showMessageDialog(AlertType.INFORMATION, str, null);

            } else {
                int number = quantitySelected - productSelected.getQuantity();

                for(Product product : DatabaseManager.getProductsInShelves()) {
                    if(product.getProductCode() == productSelected.getProductCode()) {
                        productSelected.setShelvesList(DatabaseManager.getShelves());
                        Shelf shelf = productSelected.getShelf(product.getShelfNumber());
                        shelf.setProductsList(DatabaseManager.getProductsInShelves());

                        if((quantityInShelves =

shelf.getProductQuantity(productSelected)) >= number) {
                            int productQuantity =
DatabaseManager.getProductQuantityInShelf("shelf_product",productSelected,shelf);

                            if(productQuantity >= number) {

                                DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity - number);
                                DatabaseManager.updateProduct("shopping_cart",
productSelected, quantitySelected);

                                if(str != "")
                                    str += " and";
                                str += " " + number + " of this product get in shelf

number : " + shelf.getShelfNumber();

                                break;
                            } else {
                                number -= productQuantity;
                                if(str != "")
                                    str += " and";
                                str += " " + productQuantity + " of this product returns

to shelf number : " + shelf.getShelfNumber();

                                DatabaseManager.updateProductInShelf("shelf_product", product, shelf,0);
                                DatabaseManager.updateProduct("shopping_cart",
productSelected, quantitySelected);

                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```



```

        }
    }
}
if(str != "")
    Message.showMessageDialog(AlertType.INFORMATION, str, null);

if(quantityInShelves < number && quantityInShelves > 0) {
    if(Message.showMessageDialog(AlertType.INFORMATION,
        "You have more " + quantityInShelves + " in the shelves", null) == buttonType.YES) {
        Shelf sh = new Shelf();
        sh.setProductsList(DatabaseManager.getProductsInShelves());
        Product []products = sh.getProductInShelves(productSelected);
        DatabaseManager.updateProduct("shopping_cart",
            productSelected, productSelected.getQuantity() + quantityInShelves);

        for(int i = 0; i < products.length; i++) {
            products[i].setShelvesList(DatabaseManager.getShelves());
            Shelf shelf =
            products[i].getShelf(products[i].getShelfNumber());
            int productQuantity =
            DatabaseManager.getProductQuantityInShelf("shelf_product", productSelected, shelf);

            if(quantityInShelves > products[i].getQuantity()) {
                if(str != "")
                    str += " and";
                str += " " + productQuantity + " of this
product get in shelf number : " + shelf.getShelfNumber();

                DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf, 0);
            } else {
                if(str != "")
                    str += " and";
                str += " " + products[i].getQuantity() + " of
this product get in shelf number : " + shelf.getShelfNumber();

                DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf, productQuantity - products[i].getQuantity());
                quantityInShelves -= products[i].getQuantity();
                if(quantityInShelves <= 0)
                    break;
            }
        }

        Message.showMessageDialog(AlertType.INFORMATION, str, null);
    }
} else if(quantityInShelves == 0) {
    Message.showMessageDialog(AlertType.INFORMATION, "You
dont have more of this product in shelves", null);
}

}
} else // if quantity not changed
    Message.showMessageDialog(AlertType.INFORMATION, "quantity not changed", "");
} else if(quantitySelected == 0) { // if quantity selected equal to zero
    if(Message.showMessageDialog(AlertType.INFORMATION, "You cant enter zero you can
remove the product if do you want?", "") == buttonType.YES)
        removeProduct(productSelected); // remove product from the table
    }

    refreshTable(DatabaseManager.getProductFromShoppingCart()); //refresh products table
    setTotalPrice(); // update total price
} else // if the user not selected product from the table
    Message.showMessageDialog(AlertType.NONE, "Please choose product from table", "");
}

// set customer of the order
public void setCustomerSelected(Customer customer) {
    customerSelected = customer;
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) { //initialize method
    // TODO Auto-generated method stub
    try {
        refreshTable(DatabaseManager.getProductFromShoppingCart());
    }
}

```

```

        setTotalPrice();
    } catch (SQLException | ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
//refresh products table
public void refreshTable(ObservableList<Product> list) throws SQLException, ClassNotFoundException {
    quantityCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("quantity"));
    productNameCol.setCellValueFactory(new PropertyValueFactory<Product,String>("productName"));
    productIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("productCode"));
    priceCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("price"));
    shoppingCartTable.setItems(list);
}
//this function for add order to orders table in database
public void AddOrder(Order orderAdded) throws SQLException, ClassNotFoundException {
    if(!shoppingCartTable.getItems().isEmpty()) { //check if the shopping cart is not empty
        if(Message.showMessageBoxWithYesNoButtons(AlertType.INFORMATION, "Are you sure do you want to place
order ?", null) == buttonType.YES) {
            // if the user clicked yes in the message
            DatabaseManager.addProductToOrder(DatabaseManager.getOrderCode(orderAdded)); // add
products to order according to order code
            DatabaseManager.removeShoppingCart(); // remove products in shopping cart
            setTotalPrice(); // set total price (zero)
            refreshTable(DatabaseManager.getProductFromShoppingCart()); // refresh table
            Message.showMessageBoxWithOKButton(AlertType.INFORMATION,"The order was successfully
placed", null);
        }
    } else
        Message.showMessageBoxWithOKButton(AlertType.WARNING, "Shopping cart is empty" , "");
}
//place order button
//this function add order to orders table
@FXML
void addOrderToOrderList(ActionEvent event) throws SQLException, ClassNotFoundException {
    String orderPrice = totalPriceLabel.getText().substring(0, (totalPriceLabel.getText().length()-1)); // get total price
    orderAdded = new Order(customerSelected.getId(),Double.parseDouble(orderPrice),new DateClass().getDateToday());
    AddOrder(orderAdded); // add order
}

// search product from products table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty())
        refreshTable(DatabaseManager.getProductFromShoppingCart());
    else
        refreshTable(Conditions.getItemssSearch(DatabaseManager.getProductFromShoppingCart(),searchField.getText()));
}
}

```

## קובץ viewOrderController.java

המחלקה הזאת מציגה החלון של עריכת הזמנה.

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;

public class viewOrderController{

    @FXML
    private TableView<Product> viewOrderTable;

    @FXML
    private TableColumn<Product, Integer> orderIdCol,productIdCol,quantityCol;

    @FXML
    private TableColumn<Product, String> productNameCol;

    @FXML
    private TableColumn<Product, Double> priceCol;

    @FXML
    private Label totalPriceLabel;

    @FXML
    private TextField searchField;

    Order order;
    Customer customer;
    ObservableList<Product> viewOrderList;

    @FXML
    void backToOrderPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "OrderListPage.fxml","Management products - Orders list");
    }
    //this function delete product in the order
    @FXML
    void deleteProduct(ActionEvent event) throws ClassNotFoundException, SQLException {
        Product productSelected = viewOrderTable.getSelectionModel().getSelectedItem();
        productSelected.setShelvesList(DatabaseManager.getShelves());
        Order order = new Order(productSelected.getOrderId());
        order.setProductsList(DatabaseManager.getProductOfOrder(order));

        // get current product
        if((productSelected = order.getCurrentProduct(productSelected)) != null) {
            // check if shelf is vacant
            removeProduct(productSelected,productSelected.getQuantity());
            DatabaseManager.deleteProductInOrder(productSelected, order);
            // delete product in order
        }else
            Message.showMessageDialogWithOKButton(AlertType.ERROR, "You have error", "Error");

        refreshOrderTable(DatabaseManager.getProductOfOrder(order)); // refresh table
        setTotalPrice(); // set total price after remove product
    }

    private void checkProductIsExists(Product product , int quantity) throws ClassNotFoundException, SQLException {
        if(product.checkProductIsExists(DatabaseManager.getProducts())) {
            int productQuantity = DatabaseManager.getProductQuantity("product", product.getProductCode());
            product.setQuantity(quantity + productQuantity);
            DatabaseManager.updateProduct(product);
        }else { // if product is not existing in inventory
            Category category = new Category(product.getCategory());
            // if the category is not exists
            if(category.categoryNamesTaken(category, DatabaseManager.getCategorys()))
                DatabaseManager.addCategory(category); // add category
            DatabaseManager.addProduct(product); // add product to inventory
        }
    }
}
```

```

    }

    // remove product from the products table
    private void removeProduct(Product product,int quan) throws ClassNotFoundException, SQLException {
        // get quantity of the product in shelf
        // update products in shelf

        product = product.getCurrentProduct(DatabaseManager.getProductOfOrder(order));
        Shelf sh = new Shelf();
        sh.setProductsList(DatabaseManager.getProductsInShelves());
        Product []products;
        String str = "";
        int res = quan;
        if((products = sh.getProductInShelves(product)) != null) {
            product.setShelvesList(DatabaseManager.getShelves());
            if(product.getVacantShelf(products) >= quan) {
                for(int i = 0; i < products.length;i++) {
                    products[i].setShelvesList(DatabaseManager.getShelves());
                    Shelf shelf = products[i].getShelf(products[i].getShelfNumber());
                    int productQuantity = DatabaseManager.getProductQuantityInShelf("shelf_product",product,shelf);
                    shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));
                    if(shelf.checkWeight(product, res)) {
                        DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity + res);
                        if(str != "")
                            str += " and";
                        str += " " + res + " of " + product.getProductName() + " returns to shelf number : " +
shelf.getShelfNumber();

                        product.setQuantity(0);
                        break;
                    } else {
                        int quantity = shelf.getProductQunatity(products[i]);

                        if(quantity > 0) {
                            if(str != "")
                                str += " and";
                            if(quantity >= quan) {
                                System.out.println(quan + " res");
                                product.setQuantity(product.getQuantity() - quan);
                                DatabaseManager.updateProductInShelf("shelf_product", product, shelf,
productQuantity + res);

                                str += " " + res + " of " + product.getProductName() + " returns to shelf
number : " + shelf.getShelfNumber();

                                break;
                            } else {
                                product.setQuantity(product.getQuantity() - quantity);
                                DatabaseManager.updateProductInShelf("shelf_product", product, shelf,
productQuantity + quantity);

                                str += " " + quantity + " of " + product.getProductName() + " returns to shelf
number : " + shelf.getShelfNumber();

                                res -= quantity;
                            }
                        }
                    }
                }
            }
            if(product.getQuantity() <= 0 || res <= 0)
                break;
        }
        Message.showMessageDialogWithOKButton(AlertType.INFORMATION, str, null);
    } else {
        checkProductIsExists(product,quan);
        Message.showMessageDialogWithOKButton(AlertType.INFORMATION, product.getProductName() + "
is back in stock", null);
    }
}

} else if(products == null) {
    checkProductIsExists(product,quan);
    Message.showMessageDialogWithOKButton(AlertType.INFORMATION, product.getProductName() + " is back in
stock", null);
}

}

// set order and refresh products table on the order and set total price of the order
public void init(Order order) throws ClassNotFoundException, SQLException {
    this.order = order;
    refreshOrderTable(DatabaseManager.getProductOfOrder(order)); // refresh products table on the order
    setTotalPrice(); // total price
}

//set total price of the order

```

```

private void setTotalPrice() throws ClassNotFoundException, SQLException {
    Double totalPrice = order.getTotalPrice(DatabaseManager.getProductOfOrder(order)); //get total price
    totalPriceLabel.setText(String.valueOf(Conditions.get2DigitisAfterPoint(String.valueOf(totalPrice))) + "₹"); // set total
price in the label
    DatabaseManager.updateTotalPriceOfOrder(order, Conditions.get2DigitisAfterPoint(String.valueOf(totalPrice))); // update
total price of the order
}
//set columns of the table
public void refreshOrderTable(ObservableList<Product> list) throws ClassNotFoundException {
    viewOrderTable.setItems(list);
    orderIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("orderId"));
    productIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("productId"));
    productNameCol.setCellValueFactory(new PropertyValueFactory<Product,String>("productName"));
    quantityCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("quantity"));
    priceCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("price"));
}

@FXML
void editBtn(ActionEvent event) throws ClassNotFoundException, SQLException, InterruptedException {

    if(!(viewOrderTable.getSelectionModel().isEmpty())) { // if the user selected to the table
        Product productSelected = viewOrderTable.getSelectionModel().getSelectedItem();
        int quantitySelected = Message.showMessageBoxWithTextField("Product quantity", "Please enter product quantity
:");

        productSelected.setShelvesList(DatabaseManager.getShelves());
        Order order = new Order(productSelected.getId());
        order.setProductsList(DatabaseManager.getProductOfOrder(order));
        String str = "";
        int quantityInShelves = 0;

        if(quantitySelected > 0) { // the quantity selected in message bigger than zero
            if(productSelected.getQuantity() != quantitySelected) { // if the product selected not equal to
quantity selected in message
                if(productSelected.getQuantity() > quantitySelected) { // if the product selected bigger
than quantity selected in message

                    int diff = productSelected.getQuantity() - quantitySelected;
                    removeProduct(productSelected, diff);
                    DatabaseManager.updateProductInOrder(productSelected,
order, quantitySelected);

                } else { // if the product selected less than quantity selected in message
                    int number = quantitySelected - productSelected.getQuantity();
                    if (productSelected.isProductExistsInShelves() == true) {
                        for (Product product : DatabaseManager.getProductsInShelves()) {
                            if (product.getProductCode() ==
productSelected.getProductCode()) {

                                productSelected.setShelvesList(DatabaseManager.getShelves());

                                Shelf shelf =
productSelected.getShelf(product.getShelfNumber());

                                shelf.setProductsList(DatabaseManager.getProductsInShelves());

                                if ((quantityInShelves =
shelf.getProductQuantity(productSelected)) >= number) {

                                    int productQuantity =
DatabaseManager.getProductQuantityInShelf("shelf_product", productSelected, shelf);

                                    if (productQuantity >= number) {

                                        DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity - number);

                                        DatabaseManager.updateProductInOrder(productSelected, order, quantitySelected);

                                        if (str != "")
                                            str += " and";
                                        str += " " + number + " of this product get in

                                        break;

                                    } else {
                                        if (productQuantity > 0) {
                                            number -= productQuantity;
                                            if (str != "")
                                                str += " and";

                                            str += " " + productQuantity + "

of this product returns to shelf number : " + shelf.getShelfNumber());

                                        DatabaseManager.updateProductInShelf("shelf_product", product, shelf, 0);

```

```

DatabaseManager.updateProductInOrder(productSelected, order, quantitySelected);

DatabaseManager.updateProduct("shopping_cart", productSelected, quantitySelected);
    }
    }
    }
    }
    if(str != "")

Message.showMessageWithOKButton(AlertType.INFORMATION, str, null);

if(quantityInShelves < number && quantityInShelves > 0) {

    if(Message.showMessageWithYesNoButtons(AlertType.INFORMATION, "You have more " + quantityInShelves + " in the shelves",
null) == buttonType.YES) {

        Shelf sh = new Shelf();

        sh.setProductsList(DatabaseManager.getProductsInShelves());

        Product []products =

sh.getProductInShelves(productSelected);

        DatabaseManager.updateProductInOrder(productSelected, order, productSelected.getQuantity() + quantityInShelves);
        for(int i = 0; i < products.length; i++) {

            products[i].setShelvesList(DatabaseManager.getShelves());

            Shelf shelf =

products[i].getShelf(products[i].getShelfNumber());

            int productQuantity =

DatabaseManager.getProductQuantityInShelf("shelf_product", productSelected, shelf);

            if(quantityInShelves >

                if(productQuantity > 0) {
                    if(str != "")
                        str += "

and";

                    str += " " +

productQuantity + " of this product get in shelf number : " + shelf.getShelfNumber());

                DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf, 0);

                }
            } else {
                if(str != "")
                    str += " and";
                str += " " +

products[i].getQuantity() + " of this product get in shelf number : " + shelf.getShelfNumber());

                DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf, productQuantity - products[i].getQuantity());
                }
            }
            quantityInShelves -=

            if(quantityInShelves <= 0)
                break;

        }

        Message.showMessageWithOKButton(AlertType.INFORMATION, str, null);
        }
    } else if(quantityInShelves == 0) {

        Message.showMessageWithOKButton(AlertType.INFORMATION, "You dont have more of this product in shelves", null);
        }
    } else
        Message.showMessageWithOKButton(AlertType.NONE,

productSelected.getProductName() + " is not existing in shelves", "");
    }
    } else // if quantity not changed
        Message.showMessageWithOKButton(AlertType.NONE, "Quantity of product not
changed", "");
    }
} else
    Message.showMessageWithOKButton(AlertType.NONE, "Please choose product from table", "");

refreshOrderTable(DatabaseManager.getProductOfOrder(order));
setTotalPrice();

```

```

    }

@FXML
void savePdfBtn(ActionEvent event) throws ClassNotFoundException, SQLException {
    new PdfFile().saveOrderPdf(viewOrderTable, order);
}

//search order if from the order table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty())
        refreshOrderTable(DatabaseManager.getProductOfOrder(order));
    else

    refreshOrderTable(Conditions.getItemssSearch(DatabaseManager.getProductOfOrder(order),searchField.getText()));
}
}

```

## קובץ OrderListController.java

המחלקה הזאת מציגה את רשימת ההזמנות.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.Collections;
import java.util.ResourceBundle;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;

public class OrderListController implements Initializable{

    @FXML
    private TableView<Order> orderTable;

    @FXML
    private TableColumn<Order,Integer> orderIdCol,customerIdCol;
    @FXML
    private TableColumn<Order, Double> totalPriceCol;

    @FXML
    private TableColumn<Order, String> orderDateCol;

    @FXML
    private TextField searchField;

    ObservableList<Order> ordersList;

    //switch window
    @FXML
    void backToOrderPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "OrderPage.fxml","Management products - Order");
    }
    @FXML
    void viewOrderBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
        if(!(orderTable.getSelectionModel().isEmpty())) {
            Order order = orderTable.getSelectionModel().getSelectedItem();
            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(getClass().getResource("viewOrder.fxml"));
            Stage viewOrderPage = (Stage)((Node)event.getSource()).getScene().getWindow();
            viewOrderPage.setScene(new Scene(loader.load()));
            viewOrderController controller = loader.getController();
            controller.init(order);
            viewOrderPage.setTitle("Management products - View order");
            viewOrderPage.show();
        } else
            Message.showMessageDialog(AlertType.WARNING, "choose order from table", "");
    }
    //return products to storage after remove the order
    private void returnProductToStorage(Order order) throws ClassNotFoundException, SQLException {
        ObservableList<Product> productList = DatabaseManager.getProductOfOrder(order);
        ObservableList<Product> productsInStorage = DatabaseManager.getProducts();

        if(productsList.size() > 0) {
            for(Product product : productList) {
                //check product if exists in inventory
                if(product.checkProductsExists(productsInStorage)) { // if product is exists update the quantity of the product
                    int productQuantity = DatabaseManager.getProductQuantity("product", product.getProductCode());
                    DatabaseManager.updateProduct("product",product,productQuantity + product.getQuantity());
                } else { //if the product not exists insert the product to storage
                    Category category = new Category(product.getCategory());
                    // if category is not exists add category if(category.categoryNameIsTaken(category,
                    DatabaseManager.getCategories()))
                    DatabaseManager.addCategory(category); // add category
                    DatabaseManager.addProduct(product); // add product to storage
                }
            }
        }
    }
}
```



```

        }
    }
    Message.showMessageWithOKButton(AlertType.INFORMATION, "the products return to the storage", null);
}
// delete order button
// this function delete order from orders table
@FXML
void deleteOrderBtn(ActionEvent event) throws SQLException, ClassNotFoundException {
    if(Message.showMessageWithYesNoButtons(AlertType.INFORMATION, "are you sure do you want to remove the order ?", null) ==
buttonType.YES) {
        Order selectedOrder = orderTable.getSelectionModel().getSelectedItem();
        returnProductToStorage(selectedOrder);//return all products to storage
        DatabaseManager.removeOrderQuery(selectedOrder,"orders");//remove order
        DatabaseManager.removeOrderQuery(selectedOrder,"order_product");//remove all products belong to order
        refreshOrderTable(DatabaseManager.getOrders());
    }
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    // TODO Auto-generated method stub
    try {
        refreshOrderTable(DatabaseManager.getOrders());
    } catch (ClassNotFoundException | SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
//set columns of the order table
public void refreshOrderTable(ObservableList<Order> list) throws ClassNotFoundException {
    Collections.sort(list); // sort orders table
    orderTable.setItems(list);
    orderIdCol.setCellValueFactory(new PropertyValueFactory<Order,Integer>("orderId"));
    customerIdCol.setCellValueFactory(new PropertyValueFactory<Order,Integer>("customerId"));
    totalPriceCol.setCellValueFactory(new PropertyValueFactory<Order,Double>("totalPrice"));
    orderDateCol.setCellValueFactory(new PropertyValueFactory<Order,String>("orderDate"));
}

//search order id from orders table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty()) // if the field is empty display all order
        refreshOrderTable(DatabaseManager.getOrders());
    else // else display depending on the content of the field
        refreshOrderTable(Conditions.getItemssSearch(DatabaseManager.getOrders(),searchField.getText()));
}
}

```

## קובץ ShelvesController.java

המחלקה הזאת מציגה את חלון המדפים בעסק.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;

import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Button;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.image.Image;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

public class ShelvesController implements Initializable{

    @FXML
    private TextField searchField;

    @FXML
    private TableView<Shelf> shelvesTable;

    @FXML
    private TableColumn<Shelf, Integer> shelfNumberCol;

    @FXML
    private TableColumn<Shelf, Double> maxWeightCol;

    @FXML
    private AnchorPane side;
    @FXML
    private Button expensesBtn;

    //switch page
    @FXML
    void expensesBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"Expenses.fxml","Expenses products - Expenses");
    }
    @FXML
    void CategoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CategoryPage.fxml","Management products - Category");
    }

    @FXML
    void CustomerPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CustomerPage.fxml","Management products - Customer");
    }
    @FXML
    void inventoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"inventory.fxml","Management products - Inventory");
    }

    @FXML
    void orderPageBtn(ActionEvent event) throws IOException {

        SwitchingPages.switchPage(event,"OrderPage.fxml","Management products - Order");
    }
    @FXML
    void statisticsBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"statistics.fxml","Management products - Statistics");
    }
}
```

```

@FXML
void logoutBtn(ActionEvent event) throws IOException {
    SwitchingPages.openWindowInCenter("Login",event,"Management products - Login");
}
@FXML
void addShelfBtn(ActionEvent event) throws IOException {
    FXMLLoader loader = new FXMLLoader();
    loader.setLocation(getClass().getResource("addShelf.fxml"));
    Stage window = new Stage(StageStyle.UNDECORATED);
    window.setScene(new Scene(loader.load()));
    AddShelfController controller = loader.getController();
    controller.setController(this);
    window.initModality(Modality.APPLICATION_MODAL);
    window.show();
}

@FXML
void editShelfBtn(ActionEvent event) throws IOException {
    if(!(shelvesTable.getSelectionModel().isEmpty())) {
        Shelf selectedShelf = shelvesTable.getSelectionModel().getSelectedItem();
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("EditShelf.fxml"));
        Stage window = new Stage(StageStyle.UNDECORATED);
        window.setScene(new Scene(loader.load()));
        EditShelfController controller = loader.getController();
        controller.setController(this);
        controller.setShelfData(selectedShelf);
        window.initModality(Modality.APPLICATION_MODAL);
        window.show();
    } else
        Message.showMessageDialogWithOKButton(AlertType.ERROR, "Please choose shelf from the table.", "");
}

@FXML
void viewShelfBtn(ActionEvent event) throws IOException, ClassNotFoundException, SQLException {
    if(!(shelvesTable.getSelectionModel().isEmpty())) {
        Shelf selectedShelf = shelvesTable.getSelectionModel().getSelectedItem();
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("ViewShelf.fxml"));
        Stage window = (Stage)((Node)event.getSource()).getScene().getWindow();
        window.setScene(new Scene(loader.load()));
        ViewShelfController controller = loader.getController();
        controller.setShelf(selectedShelf);
        window.show();
    } else
        Message.showMessageDialogWithOKButton(AlertType.ERROR, "Please choose shelf from the table.", "");
}

// open add product page
@FXML
void addProductBtn(MouseEvent event) throws IOException {
    if(!shelvesTable.getSelectionModel().isEmpty()) {
        Shelf selectedShelf = shelvesTable.getSelectionModel().getSelectedItem();
        switchToAddProductToShelfWindow(selectedShelf);
    } else
        Message.showMessageDialogWithOKButton(AlertType.WARNING, "Choose shelf from the table", null);
}

//open the a chosen shelf
private void switchToAddProductToShelfWindow(Shelf shelf) throws IOException {
    FXMLLoader loader = new FXMLLoader();
    loader.setLocation(getClass().getResource("AddProductToShelf.fxml"));
    Stage window = new Stage(StageStyle.UNDECORATED);
    window.setScene(new Scene(loader.load()));
    AddProductToShelf controller = loader.getController();
    controller.setShelf(shelf);
    controller.setController(this);
    window.initStyle(StageStyle.DECORATED);
    window.getIcons().add(new Image(getClass().getResourceAsStream("/inventoryIcon.png")));//icon
    window.setResizable(false);
    window.setTitle("Management products - Add Product to shelf");
    window.initModality(Modality.APPLICATION_MODAL);
    window.show();
}

//set all columns of the table
public void refreshTable(ObservableList<Shelf> list) throws ClassNotFoundException {
    shelfNumberCol.setCellValueFactory(new PropertyValueFactory<Shelf,Integer>("shelfNumber"));
    maxWeightCol.setCellValueFactory(new PropertyValueFactory<Shelf,Double>("maxWeight"));
    shelvesTable.setItems(list);
    new RowTable().changeRowsColor(shelvesTable);
}

```

```

    }
    //add shelf to table view
    public void addShelf(Shelf shelf) throws ClassNotFoundException, SQLException {
        DatabaseManager.addShelf(shelf);
        refreshTable(DatabaseManager.getShelves());
    }
    //update shelf in the table
    public void editShelf(Shelf shelf) throws ClassNotFoundException, SQLException {
        DatabaseManager.updateShelf(shelf);
        refreshTable(DatabaseManager.getShelves());
    }

    // menu clicked
    @FXML
    void menuClicked(MouseEvent event) {
        new AnimationClass().sideAnimation(side);
    }

    // remove shelf button
    // this function remove shelf from shelves table
    @FXML
    void removeShelfBtn(ActionEvent event) throws ClassNotFoundException, SQLException {
        // if the user selected shelf from shelves table
        if(!(shelvesTable.getSelectionModel().isEmpty())) {
            Shelf selectedShelf = shelvesTable.getSelectionModel().getSelectedItem();

            if(DatabaseManager.getProductsInShelf(selectedShelf).size() == 0) { // if the shelf is empty (not have products in the shelf)
                if(Message.showMessageDialogWithYesNoButtons(AlertType.INFORMATION, "Are you sure you want to delete the shelf?", "") ==
buttonType.YES) {
                    DatabaseManager.removeShelf(selectedShelf); // remove the shelf
                    refreshTable(DatabaseManager.getShelves()); // refresh shelves table
                }
            } else // if the shelf is not empty (have products in the shelf)
                Message.showMessageDialogWithOKButton(AlertType.ERROR, "You cant remove this shelf because you have products in the
shelf.", "");

        } else // if the user not selected shelf from shelves table
            Message.showMessageDialogWithOKButton(AlertType.ERROR, "Please choose shelf from the table.", "");
    }

    //search number of shelf in shelves table
    @FXML
    void searchFieldTyped(KeyEvent event) throws ClassNotFoundException {
        if(searchField.getText().isEmpty())
            refreshTable(DatabaseManager.getShelves());
        else
            refreshTable(Conditions.getItemssSearch(DatabaseManager.getShelves(),searchField.getText()));
    }
    // side
    private void sideManaged() {
        side.setTranslateX(-215);
        side.setVisible(false);
        side.setManaged(false);
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        try {
            refreshTable(DatabaseManager.getShelves());
            sideManaged();
            Conditions.checkManagerStatus(expensesBtn);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}

```

## קובץ AddShelfController.java

המחלקה הזאת מציגה את חלון הוספת המדפים.

```
package application;

import java.sql.SQLException;

import javafx.event.ActionEvent;
import javafx.event.Event;
import javafx.fxml.FXML;
import javafx.scene.Node;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;

public class AddShelfController {

    @FXML
    private TextField shelfNumberField,maxWeightField;

    @FXML
    private Label shelfNumberLabel,maxWeightLabel;

    ShelvesController controller;

    @FXML
    void addShelf(ActionEvent event) throws ClassNotFoundException, SQLException {
        Field field = new Field();
        if(field.checkEmptyFields(shelfNumberField, maxWeightField)) {

            Boolean codeField = field.checkNumberField(shelfNumberField, shelfNumberLabel);
            Boolean maxWeight = field.checkFloatNumberField(maxWeightField, maxWeightLabel);

            if(codeField && maxWeight) {

                Shelf shelf = new Shelf(Integer.parseInt(shelfNumberField.getText()),
Conditions.get2DigitsAfterPoint(maxWeightField.getText()));
                if(shelf.checkShelfExists(DatabaseManager.getShelves())) {
                    exitStage(event);
                    controller.addShelf(shelf);
                }
                else
                    Message.showMessageDialog(AlertType.WARNING, "The shelf you selected exists", "");
            }
            else
                Message.showMessageDialog(AlertType.ERROR, "You have empty fields", "Empty fields");
        }
    }

    @FXML
    void shelfNumberTyped(KeyEvent event) {
        if(!Conditions.checkStringLength(shelfNumberField.getText(), 6))
            Conditions.removeLastIndexInString(shelfNumberField);
    }

    @FXML
    void maxWeightTyped(KeyEvent event) {
    }

    @FXML
    void exitBtn(MouseEvent event) {
        exitStage(event);
    }

    public void exitStage(Event event) {
        Node source = (Node) event.getSource();
        Stage stage = (Stage) source.getScene().getWindow();
        stage.close();
    }

    public void setController(ShelvesController controller) {
        this.controller = controller;
    }
}
```

## קובץ EditShelfController.java

המחלקה הזאת מציגה את חלון עדכון המדפים.

```
package application;

import java.sql.SQLException;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Label;
import javafx.scene.input.MouseEvent;

public class EditShelfController {

    ShelvesController controller;

    @FXML
    private TextField shelfNumberField,maxWeightField;

    @FXML
    private Label maxWeightLabel;

    Shelf shelf;

    // exit button
    @FXML
    void exitBtn(MouseEvent event) {
        SwitchingPages.exitStage(event);
    }

    //edit shelf button
    // this function edit shelf data
    @FXML
    void editShelf(ActionEvent event) throws NumberFormatException, ClassNotFoundException, SQLException {
        Field field = new Field();
        // check if max weight field is empty
        if(field.checkEmptyFields(maxWeightField)) {
            // check max weight field is valid
            if(field.checkFloatNumberField(maxWeightField, maxWeightLabel)) {
                shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));
                Double maxWeight = Conditions.get2DigitsAfterPoint(maxWeightField.getText());
                // if the max weight is bigger equal than products weight in the shelf
                if(shelf.getShelfWeight() <= maxWeight) {
                    Shelf shelf = new Shelf(Integer.parseInt(shelfNumberField.getText()),maxWeight);
                    controller.editShelf(shelf); // edit shelf from the shelves table
                    SwitchingPages.exitStage(event); // exit the window
                } else // if the products weight bigger than max weight receive a message
                    Message.showMessageDialog(AlertType.WARNING,"the products weight in shelf is " + shelf.getShelfWeight() +
                    " you have enter number bigger than " + shelf.getShelfWeight() + "or equal", null);
            }

            // else // if max weight field is empty
            Message.showMessageDialog(AlertType.WARNING, "The maximum field is empty", "");
        }

        // set the clicked shelf data in the fields
        public void setShelfData(Shelf shelf) {
            this.shelf = shelf;
            shelfNumberField.setText(String.valueOf(shelf.getShelfNumber()));
            maxWeightField.setText(String.valueOf(shelf.getMaxWeight()));
        }

        // set shelves controller
        public void setController(ShelvesController controller) {
            this.controller = controller;
        }

    }
}
```

## קובץ ViewShelfController.java

המחלקה הזאת מציגה את חלון עריכת המדף.

```
package application;

import java.io.IOException;
import java.sql.SQLException;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Label;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;

public class ViewShelfController {

    @FXML
    private TextField searchField;

    @FXML
    private TableView<Product> productsTable;

    @FXML
    private TableColumn<Product, Integer> productIdCol, quantityCol, shelfNumberCol;

    @FXML
    private TableColumn<Product, String> productNameCol;

    @FXML
    private TableColumn<Product, Double> weightCol;

    Shelf shelf;

    @FXML
    private Label currentWeightLabel;

    // switch window
    // switch to shelves window
    @FXML
    void backBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "ShelvesPage.fxml", "Management products - Shelves");
    }

    // refresh shelf table
    private void refreshTable(ObservableList<Product> list) throws ClassNotFoundException, SQLException {
        shelfNumberCol.setCellValueFactory(new PropertyValueFactory<Product, Integer>("shelfNumber"));
        productIdCol.setCellValueFactory(new PropertyValueFactory<Product, Integer>("productCode"));
        productNameCol.setCellValueFactory(new PropertyValueFactory<Product, String>("productName"));
        quantityCol.setCellValueFactory(new PropertyValueFactory<Product, Integer>("quantity"));
        weightCol.setCellValueFactory(new PropertyValueFactory<Product, Double>("weight"));
        productsTable.setItems(list);
        setCurrentWeightLabel();
    }

    // set shelf and refresh the shelves table
    public void setShelf(Shelf shelf) throws ClassNotFoundException, SQLException {
        this.shelf = shelf;
        refreshTable(DatabaseManager.getProductsInShelf(shelf));
    }

    // this function calculate the weight of the product in the shelf
    private void setCurrentWeightLabel() throws ClassNotFoundException, SQLException {
        shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));
        currentWeightLabel.setText(String.valueOf(Conditions.get2DigitsAfterPoint(String.valueOf(shelf.getShelfWeight())) + " Kg"));
    }

    //edit product button

    @FXML
    void editProductBtn(ActionEvent event) throws ClassNotFoundException, SQLException, InterruptedException {
        // if user selected product from the table
        if(!productsTable.getSelectionModel().isEmpty()) {
            Product productSelected = productsTable.getSelectionModel().getSelectedItem();
        }
    }
}
```

```

// check if the product is exists in inventory
if(productSelected.checkProductIsExists(DatabaseManager.getProducts())) {
    int quantitySelected = Message.showMessageWithTextField("Product quantity", "Please enter product quantity :");
    // if product exists if shelf get quantity of the product in inventory
    int productQuantity = DatabaseManager.getProductQuantity("product", productSelected.getProductCode());

    if(quantitySelected > 0) { // if the string in the message is valid and bigger than zero

        int difference;

        if(productSelected.getQuantity() != quantitySelected) { // if product quantity not equal to quantity selected
            if(productSelected.getQuantity() > quantitySelected) { // if product quantity bigger than quantity
                // calculate the difference between product selected (quantity) quantity to quantity
                difference = productSelected.getQuantity() - quantitySelected;
                // if product is exists in inventory
                if(productSelected.checkProductIsExists(DatabaseManager.getProducts())) {
                    // update product quantity in inventory
                    DatabaseManager.updateProduct("product", productSelected, productQuantity
+ difference);
                    // update product quantity in shelf
                    DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf,
quantitySelected);
                } else { // if product is not exists in inventory
                    // update product quantity in shelf
                    DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf, quantitySelected);
                    productSelected.setQuantity(difference);
                    // add product to inventory with the difference product selected quantity to
                    DatabaseManager.addProduct(productSelected);
                }
            } else if (quantitySelected > productSelected.getQuantity()) { // if quantity selected bigger than product selected
                if(productQuantity > 0) { // check if product quantity in inventory bigger than zero
                    shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));
                    // calculate the difference between quantity selected quantity to product selected
                    difference = quantitySelected - productSelected.getQuantity();
                    // check if weight of the new selected product is enough put it on the shelf
                    if(!shelf.checkWeight(productSelected, difference) || productQuantity < (quantitySelected -
productSelected.getQuantity())) {
                        checkVacanteWeight(productSelected, productQuantity); // check quantity of products
                        can insert to shelf
                    } else { // if the weight of the products is enough put it on the shelf
                        // update product in inventory
                        DatabaseManager.updateProduct("product", productSelected, productQuantity -
difference);
                        // update product in shelf
                        DatabaseManager.updateProductInShelf("shelf_product", productSelected, shelf,
quantitySelected);
                    }
                } else // if product quantity in inventory equal to zero
                    Message.showMessageWithOKButton(AlertType.WARNING, "You dont have more of this
product", "");
            }

            refreshTable(DatabaseManager.getProductsInShelf(shelf)); // refresh products table
        } else // if quantity not changed
            Message.showMessageWithOKButton(AlertType.INFORMATION, "quantity not changed", "");

        } else if(quantitySelected == 0) { // if quantity selected is zero
            if(Message.showMessageWithYesNoButtons(AlertType.INFORMATION, "do you want to delete the product
from the shelf ?", null) == buttonType.YES) {
                checkProductIsExists(productSelected); // remove the product
                refreshTable(DatabaseManager.getProductsInShelf(shelf)); // refresh products table
            }
        }

    } else // if product not exists in inventory
        Message.showMessageWithOKButton(AlertType.INFORMATION, "this product is not exists", "");
    } else // if user not selected product from the table
        Message.showMessageWithOKButton(AlertType.NONE, "Please choose product from table", "");
}

```



```

// this function checks if the shelf have vacant weight
private void checkVacanteWeight(Product product,int productQuantity) throws ClassNotFoundException, SQLException {
    int number = shelf.getProductQuantity(product); // get the quantity of the product can insert to shelf
    if(number > 0) { // if can insert product to shelf (shelf is full)
        if(productQuantity < number) // if the quantity of the product can insert to shelf bigger than product quantity in
storage
            number = productQuantity;
            // receive a message how many products can insert to shelf
            if(Message.showMessageBoxWithYesNoButtons(AlertType.INFORMATION, "The quantity of products that can
be put in is " + number + " are you sure? ", null) == buttonType.YES) {
                // if user clicked yes
                DatabaseManager.updateProduct("product",product,productQuantity - number); // update product in
the inventory
                DatabaseManager.updateProductInShelf("shelf_product",product,shelf, product.getQuantity() + number); //
update the product in shelf
            }

        } else // if can't insert product to shelf (shelf is full)
            // receive a message difference between max weight to shelf weight
            Message.showMessageBoxWithOKButton(AlertType.WARNING,"The free weight on the shelf is " +
Conditions.get2DigitsAfterPoint(String.valueOf(shelf.getWeightDifference()))
+ " KG and the weight of the product is " + product.getWeight() + " KG", null);
    }
}

// this function checks if product exists in storage or not exists
private void checkProductIsExists(Product product) throws ClassNotFoundException, SQLException{

    product = product.getCurrentProduct(DatabaseManager.getProductsInShelf(shelf)); // get product in the shelf
    // if product is exists in the storage
    if(product.checkProductIsExists(DatabaseManager.getProducts())) {
        // get quantity of the product in the storage
        int productQuantity = DatabaseManager.getProductQuantity("product",product.getProductCode());
        DatabaseManager.updateProduct("product",product, productQuantity + product.getQuantity()); // update product in the
storage
        DatabaseManager.deleteProductInShelf("shelf_product",product, shelf); // delete the product from the shelf
    } else { // if product not exists in storage
        Category category = new Category(product.getCategory());
        if(category.categoryNameIsTaken(category, DatabaseManager.getCategories()))
            DatabaseManager.addCategory(category); // add category
        DatabaseManager.addProduct(product); // add product to storage
        DatabaseManager.deleteProductInShelf("shelf_product",product, shelf); // delete the product from the storage
    }
}

}

// this function for remove product
@FXML
void removeProductBtn(ActionEvent event) throws ClassNotFoundException, SQLException {
    // checks if the user selected product in the products table
    if(!(productsTable.getSelectionModel().isEmpty())) {
        Product selectedProduct = productsTable.getSelectionModel().getSelectedItem();
        checkProductIsExists(selectedProduct); // remove the product
        refreshTable(DatabaseManager.getProductsInShelf(shelf)); // refresh products table
    } else
        Message.showMessageBoxWithOKButton(AlertType.NONE, "Please choose product from table", "");
}

// search product in the shelf
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty())
        refreshTable(DatabaseManager.getProductsInShelf(shelf));
    else
        refreshTable(Conditions.getItemssSearch(DatabaseManager.getProductsInShelf(shelf),searchField.getText()));
}
}

```

## קובץ AddProductToShelf.java

המחלקה הזאת מציגה את חלון הוספת מוצר למדף.

```
package application;

import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;

import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;

public class AddProductToShelf implements Initializable {

    @FXML
    private TextField searchField;

    @FXML
    private TableView<Product> productsTable;

    @FXML
    private TableColumn<Product, String> categoryCol, productNameCol;

    @FXML
    private TableColumn<Product, Double> weightCol;

    @FXML
    private TableColumn<Product, Integer> quantityCol, productIdCol;

    ShelvesController controller;
    Product product;
    Shelf shelf;
    ObservableList<Product> products;

    //this function for add product to shelf
    @FXML
    void addProduct(ActionEvent event) throws ClassNotFoundException, SQLException, InterruptedException {
        if(!productsTable.getSelectionModel().isEmpty()) {

            int quantity = Message.showMessageDialogWithTextField("Add products", "Please enter quantity of product :");

            if(quantity > 0) {
                Product selectedProduct = productsTable.getSelectionModel().getSelectedItem();
                int productQuantity = DatabaseManager.getProductQuantity("product", selectedProduct.getProductCode());
                shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));

                // checkWeight() -> check if the selected quantity of product can to put it in shelf
                // checkProduct() -> check if the quantity in the storage bigger than selected product
                if(shelf.checkWeight(selectedProduct, quantity) && selectedProduct.checkProduct(quantity, productQuantity))
                {
                    if(selectedProduct.isProductQuantityZero(selectedProduct.getQuantity())) { // if the product selected
                        is zero
                    }
                    else
                        checkProductsExists(selectedProduct, productQuantity, quantity);
                    Message.showMessageDialogWithOKButton(AlertType.WARNING, "You dont have more of
this product", null);
                }
                else
                    checkFreeWeight(selectedProduct, productQuantity);
            }

            controller.refreshTable(DatabaseManager.getShelves());
            refreshTable(DatabaseManager.getProducts());
        }
        else
            Message.showMessageDialogWithOKButton(AlertType.WARNING, "choose product from the table", null);
    }

    // this function for check vacant weight
    private void checkFreeWeight(Product product, int productQuantity) throws ClassNotFoundException, SQLException {
        int number = shelf.getProductQuantity(product); // returns how many products can be put on the shelf
        if(number > 0) {

```

```

        if(product.getQuantity() < number) // if number bigger than product quantity in storage
            number = product.getQuantity();// number equal to product quantity in storage
        //Message yes or no
        if(Message.showMessageWithYesNoButtons(AlertType.INFORMATION, "The quantity of products that can
be put in is " + number +" are you sure? ", null) == buttonType.YES) {
            checkProductIsExists(product,productQuantity,number); // insert product to shelf
        }
    } else
        Message.showMessageWithOKButton(AlertType.WARNING,"The vacant weight on the shelf is " +
Conditions.get2DigitisAfterPoint(String.valueOf(shelf.getWeightDifference()))
+ " KG and the weight of the product is " + product.getWeight() + " KG", null);
}
// this function checks products is exists in shelf
private void checkProductIsExists(Product product,int productQuantity, int quantity) throws ClassNotFoundException, SQLException {

    product = product.getCurrentProduct(DatabaseManager.getProducts());

    if(shelf.checkProductIsExists(product) == -1) { // if product is not exists in shelf
        DatabaseManager.updateProduct("product", product, productQuantity - quantity); // update product in storage
        product.setShelvesList(DatabaseManager.getShelves());
        DatabaseManager.addProductToShelf(product,shelf,quantity);// add product to shelf
    } else { // if product is exists in shelf
        int productQunatityInshelf = shelf.checkProductIsExists(product); // get quantity of the product in shelf
        //update product in shelf -> product quantity in shelf plus selected quantity
        DatabaseManager.updateProductInShelf("shelf_product",product,shelf, productQunatityInshelf + quantity);
        // update product in storage -> product quantity in storage - quantity selected
        DatabaseManager.updateProduct("product", product, productQuantity - quantity);
    }
}
// search product name or product code from the table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty()) // if search field is empty set all products in the table
        refreshTable(DatabaseManager.getProducts());
    else // if the search field is not empty, you will see all the products that contain the string that the user entered
        refreshTable(Conditions.getItemssSearch(DatabaseManager.getProducts(),searchField.getText()));
}

@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    try {
        refreshTable(DatabaseManager.getProducts());
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}

//set all columns of the table
private void refreshTable(ObservableList<Product> list) throws ClassNotFoundException {
    new RowTable().changeRowsColor(productsTable); // change color to the table rows
    categoryCol.setCellValueFactory(new PropertyValueFactory<Product,String>("category"));
    productNameCol.setCellValueFactory(new PropertyValueFactory<Product,String>("productName"));
    weightCol.setCellValueFactory(new PropertyValueFactory<Product,Double>("weight"));
    quantityCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("quantity"));
    productIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("productCode"));
    productsTable.setItems(list);
}
//set shelves controller
public void setController(ShelvesController controller) {
    this.controller = controller;
}
// set shelf the user selected
public void setShelf(Shelf shelf) {
    this.shelf = shelf;
}
}

```

## קובץ statisticsController.java

המחלקה שולטת על חלון הסטטיסטיקה של העסק.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.paint.Color;

public class statisticsController implements Initializable{

    @FXML
    private Label
revenueLabel,productsNumberLabel,categoryNumberLabel,customerNumberLabel,productQuantityLabel,ordersNumberLabel,shelvesNumberLa
bel,
expensesLabel,profitLabel,label,monthLabel,prodQuantityInShelf,totalProductInShelf;

    @FXML
    private AnchorPane side;

    @FXML
    private Button expensesBtn;

    Double expenses = 0.0;
    Double revenue = 0.0;

    //switch window

    @FXML
    void PrordutsInCategoriesBtn(MouseEvent event) throws IOException {
        SwitchingPages.switchPage(event, "productsInCategories.fxml","Management products - products in categories");
    }
    @FXML
    void expensesBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "Expenses.fxml","Management products - Expenses");
    }
    @FXML
    void CategoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "CategoryPage.fxml","Management products - Category");
    }
    @FXML
    void CustomerPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "CustomerPage.fxml","Management products - Customer");
    }
    @FXML
    void orderPageBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "OrderPage.fxml","Management products - Orders");
    }
    @FXML
    void inventoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "inventory.fxml","Management products - Inventory");
    }
    @FXML
    void bestSellingBtn(MouseEvent event) throws IOException {
        SwitchingPages.switchPage(event, "Best-selling-page.fxml","Management products - Best-selling");
    }
    @FXML
    void bestSellingTop3Btn(MouseEvent event) throws IOException {
        SwitchingPages.switchPage(event, "best-selling-top-3.fxml","Management products - Best-selling-top-3");
    }
    @FXML
```

```

void productsTheCategoryBtn(MouseEvent event) throws IOException {
    SwitchingPages.switchPage(event, "ProductsInTheCategory.fxml", "Management products - Products in the category");
}
@FXML
void shelvesBtn(ActionEvent event) throws IOException {
    SwitchingPages.switchPage(event, "shelvesPage.fxml", "Management products - Shelves");
}

@FXML
void ordersNumberBtn(MouseEvent event) throws IOException {
    SwitchingPages.switchPage(event, "OrdersNumberOfCustomer.fxml", "Management products - Orders number");
}

@FXML
void orderInMonthBtn(MouseEvent event) throws IOException {
    DateClass date = new DateClass();
    SwitchingPages.switchPage(event, "OrderInMonth.fxml", "Management products - Orders In " +
date.getMonth(date.getCurrentMonth()) + " month");
}
// set revenues of the business in label
private void setRevenueLabel() throws ClassNotFoundException, SQLException {
    ObservableList<Order> ordersList = DatabaseManager.getOrders();
    Product p = new Product();
    p.setOrdersList(ordersList);
    revenue = p.calculateRevenue(); // calculate revenue
    revenueLabel.setText(String.valueOf(revenue) + "₺");
}

// calculate expenses from the products
private Double calculateExpenses(ObservableList<Product> productsList) {
    Double expenses = 0.0;

    for(Product product: productsList)
        expenses += product.getQuantity() * product.getSupplierPrice();

    return expenses;
}

// set expenses of the business in label
private void setExpensesLabel() throws ClassNotFoundException, SQLException {
    // expenses from the Storage
    expenses += calculateExpenses(DatabaseManager.getProducts());

    // expenses from the shelves
    for(Shelf shelf : DatabaseManager.getShelves())
        expenses += calculateExpenses(DatabaseManager.getProductsInShelf(shelf));

    // expenses from the orders
    for(Order order : DatabaseManager.getOrders())
        expenses += calculateExpenses(DatabaseManager.getProductOfOrder(order));

    // expenses from water and electrical expenses
    for(Expenses exp : DatabaseManager.getExpenses())
        expenses += exp.getTotal();

    // expenses from employee salaries
    for(User employee : DatabaseManager.getEmployees())
        expenses += ((Employee)employee).getGeneralSalary();

    expensesLabel.setText(String.valueOf(Conditions.get2DigitsAfterPoint(String.valueOf(expenses))) + "₺");
}

// set profit of the business in label
// profit equals (revenue - expenses) -> profit of the business
private void setProfitLabel() {
    profitLabel.setText(String.valueOf(Conditions.get2DigitsAfterPoint(String.valueOf(revenue - expenses))) + "₺");
}
private void setLabels() throws ClassNotFoundException, SQLException {
    productsNumberLabel.setText(String.valueOf(DatabaseManager.getProducts().size())); // products number
    categoryNumberLabel.setText(String.valueOf(DatabaseManager.getCategories().size())); // categories number
    customerNumberLabel.setText(String.valueOf(DatabaseManager.getCustomer().size())); // customers number
    ordersNumberLabel.setText(String.valueOf(DatabaseManager.getOrders().size())); // orders number
    shelvesNumberLabel.setText(String.valueOf(DatabaseManager.getShelves().size())); // shelves number
    prodQuantityInShelf.setText(String.valueOf(getProductNumber())); // products number in shelves
    DateClass date = new DateClass();
    monthLabel.setText(date.getMonth(date.getCurrentMonth()) + " month"); // set the current month
}

// get products number in the shelves

```

```

        public int getProductNumber() throws ClassNotFoundException, SQLException {
            ObservableList<Product> list = FXCollections.observableArrayList();

            for(Product product : DatabaseManager.getProductsInShelves()) {
                if(!product.checkProductsExists(list)) // if product not exists add product to list
                    list.add(product); // add product to list
            }
            // return list size
            return list.size();
        }

        // set total of all products in label
        private void setProductsQuantityLabel() throws ClassNotFoundException, SQLException {
            Product product = new Product();
            productQuantityLabel.setText(String.valueOf(product.getQuantityOfAllProducts(DatabaseManager.getProducts()))); // total products
in inventory
            totalProductInShelf.setText(String.valueOf(product.getQuantityOfAllProducts(DatabaseManager.getProductsInShelves()))); // total
products in shelves
        }
        // menu clicked
        @FXML
        void menuClicked(MouseEvent event) {
            new AnimationClass().sideAnimation(side);
        }
        // animation (flashing labels)
        private void setAnimation() throws ClassNotFoundException, SQLException {
            if(revenue - expenses < 0) // if the profit less than zero
                new AnimationClass().flashingLabel(profitLabel, 2.0, Color.LIGHTGREY, Color.RED);
            if(DatabaseManager.getCustomer().size() == 0) // if customers number equal to zero
                new AnimationClass().flashingLabel(customerNumberLabel, 2.0, Color.LIGHTGREY, Color.RED);
            if(DatabaseManager.getOrders().size() == 0) // if orders number equal to zero
                new AnimationClass().flashingLabel(ordersNumberLabel, 2.0, Color.LIGHTGREY, Color.RED);
        }

        // logout button
        @FXML
        void logoutBtn(ActionEvent event) throws IOException {
            SwitchingPages.openWindowInCenter("Login",event,"Management products - Login");
        }
        //side
        private void sideManaged() {
            side.setTranslateX(-215);
            side.setVisible(false);
            side.setManaged(false);
        }
        @Override
        public void initialize(URL arg0, ResourceBundle arg1) {
            try {
                setExpensesLabel(); // set expenses in the label
                setProfitLabel(); // set profit in the label
                setLabels(); // set all label (shelves number , customer number , orders number , categories number
...
                setProductsQuantityLabel(); // set total of all products
                setAnimation(); // set animation for the labels
                sideManaged();
                setRevenueLabel(); // set revenues in the label
                Conditions.checkManagerStatus(expensesBtn);
            } catch (ClassNotFoundException | SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}

```

## קובץ bestSellingProductController.java

המחלקה שולטת על חלון המוצרים הכי נמכרים בחודש הנוכחי

```
package application;
```

```
import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.text.ParseException;
import java.util.Calendar;
import java.util.Collections;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
```

```
public class bestSellingProductController implements Initializable{
```

```
    @FXML
    private TableView<Product> productTable;
```

```
    @FXML
    private TableColumn<Product, Integer> productIdCol;
```

```
    @FXML
    private TableColumn<Product, String> productNameCol;
```

```
    @FXML
    private TableColumn<Product, Integer> numberSoldCol;
```

```
    @FXML
    private TextField searchField;
```

```
    @FXML
    private Label titleLabel;
```

```
    @FXML
    void backToOrderPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml", "Management products - Statistics");
    }
```

```
@Override
public void initialize(URL arg0, ResourceBundle arg1) {
    // TODO Auto-generated method stub
    try {
        refreshTable(getSortedProducts());
        setTitleLabel();
    } catch (ClassNotFoundException | SQLException | ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
private void setTitleLabel() {
    DateClass date = new DateClass();
    titleLabel.setText(titleLabel.getText() + "In " + date.getMonth(date.getCurrentMonth()) + " month");
}
```

```
    //this function get the best selling product in the month current month
private ObservableList<Product> getSortedProducts() throws ClassNotFoundException, SQLException, ParseException {
    ObservableList<Product> productsList = DatabaseManager.getProductsOfOrders();
    ObservableList<Product> newList = FXCollections.observableArrayList();
```

```
    DateClass date = new DateClass();
```

```
    // get all orders in data base
    for(Order order : DatabaseManager.getOrders()) {
        String orderDate = order.getOrderDate();
        // if placed order in the current month
        if(date.getMonth(orderDate) == date.getCurrentMonth() && date.getYear(orderDate) ==
            date.getCurrentYear()) {
```

```

for (Product product : DatabaseManager.getProductOfOrder(order)) { // get all product in the order
order.setProductsList(newList);
// check if the product exists in the new list (list of all product with number sold of the
product)
if(!order.isProductExists(product)) {
order.setProductsList(productsList);
// calculate number sold of products
product.setNumberSold(order.calculateNumberSoldOfProduct(product));
newList.add(product); // add product to list
}
}
}

Collections.sort(newList); //sort the product as per number sold
return newList;
}
//refresh table
private void refreshTable(ObservableList<Product> list) throws ClassNotFoundException, SQLException {
productIdCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("productCode"));
productNameCol.setCellValueFactory(new PropertyValueFactory<Product,String>("productName"));
numberSoldCol.setCellValueFactory(new PropertyValueFactory<Product,Integer>("numberSold"));
productTable.setItems(list);
}
// search product from the table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException, ParseException {
if(searchField.getText().isEmpty())
refreshTable(getSortedProducts());
else
refreshTable(Conditions.getItemssSearch(getSortedProducts(),searchField.getText()));
}
}

```



## קובץ bestSellingTop3Controller.java

המחלקה שולטת על חלון שלושה המוצרים הכי נמכרים בחודש הנוכחי.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.text.ParseException;
import java.util.Collections;
import java.util.ResourceBundle;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.chart.BarChart;
import javafx.scene.chart.CategoryAxis;
import javafx.scene.chart.NumberAxis;
import javafx.scene.chart.XYChart;

public class bestSellingTop3Controller implements Initializable{

    @FXML
    private BarChart<String,Number> barChart;

    @FXML
    private CategoryAxis x;

    @FXML
    private NumberAxis y;

    @FXML
    void backToStatisticsPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml","Management products - Statistics");
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        // TODO Auto-generated method stub
        try {
            setProductInBarChart();
        } catch (ClassNotFoundException | SQLException | ParseException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    // this function return the best selling products in the current month
    private ObservableList<Product> getSortedProducts() throws ClassNotFoundException, SQLException, ParseException {
        ObservableList<Product> productList = DatabaseManager.getProductsOfOrders();
        ObservableList<Product> newList = FXCollections.observableArrayList();

        DateClass date = new DateClass();

        // get all orders in data base
        for(Order order : DatabaseManager.getOrders()) {
            String orderDate = order.getOrderDate();
            // if placed order in the current month
            if(date.getMonth(orderDate) == date.getCurrentMonth() && date.getYear(orderDate) ==
date.getCurrentYear()) {
                for (Product product : DatabaseManager.getProductOfOrder(order)) { // get all product in the order
                    order.setProductsList(newList);
                    // check if the product exists in the new list (list of all product with number sold of the
product)
                    if(!order.isProductExists(product)) {
                        order.setProductsList(productList);
                        // calculate number sold of products
                        product.setNumberSold(order.calculateNumberSoldOfProduct(product));
                        newList.add(product); // add product to list
                    }
                }
            }
        }

        Collections.sort(newList);//sort the product as per number sold
        return newList;
    }
}
```

```

// this function show bar chart with top 3 selling products
private void setProductInBarChart() throws ClassNotFoundException, SQLException, ParseException {
    Product p = new Product();
    ObservableList<Product> productsList = getSortedProducts();
    ObservableList<Product> newList = FXCollections.observableArrayList();

    for(int i = 0; i < productsList.size() && i < 3; i++) //this loop add product to list
        newList.add(productsList.get(i));

    XYChart.Series<String, Number> s = new XYChart.Series<>();

    for(int i = 0; i < p.getTop3Products(productsList).size(); i++)
        s.getData().add(new XYChart.Data<>(newList.get(i).getProductName(), newList.get(i).getNumberSold())); //
bar in bar chart

    barChart.setLegendVisible(false); //remove legend item
    barChart.getData().add(s); //add the bars to barchart

}
}

```

## קובץ productsInCategoriesController.java

המחלקה שולטת על חלון הצגת כל אחוז המוצרים בתוך המלאי ומציג אותם בעוגה.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.chart.PieChart;
import javafx.scene.control.Tooltip;

public class productsInCategoriesController implements Initializable {

    @FXML
    private PieChart pieChart;

    @FXML
    void backToStatisticsPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml", "Management products - statistics");
    }

    // this function calculate perecrntage of th products in the category and set the data in pie chart
    private void setPieChart() throws ClassNotFoundException, SQLException {
        ObservableList<PieChart.Data> pieChartData = FXCollections.observableArrayList();

        for(Category category : DatabaseManager.getCategorys()) {
            category.setProductList(DatabaseManager.getProducts());
            pieChartData.add(new PieChart.Data(category.getCategoryName(),
(double)category.getQuantityInCategory() / category.getQuantityOfAllProducts() * 100));
        }

        pieChart.setTitle("categories"); // set title of the pie chart
        pieChart.setData(pieChartData); // set data of all categories
        // view the perecntage of the products in the category
        pieChart.getData().forEach(data ->{
            String percentage = String.format("%.2f%", data.getPieValue());
            Tooltip tool = new Tooltip(percentage);
            Tooltip.install(data.getNode(), tool);
        });
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {

        try {
            setPieChart();
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}
```

## קובץ OrderInMonthController.java

המחלקה שולטת על חלון הצגת כל ההזמנות בחודש הנוכחי.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.text.ParseException;
import java.util.Collections;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;

public class OrderInMonthController implements Initializable{

    @FXML
    private TextField searchField;

    @FXML
    private TableView<Order> orderTable;

    @FXML
    private TableColumn<Order,Integer> orderIdCol,customerIdCol;

    @FXML
    private TableColumn<Order, Double> totalPriceCol;

    @FXML
    private TableColumn<Order, String> orderDateCol;
    /* switch page */
    @FXML
    void backToOrderPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml","Management products - Statistics");
    }

    // search order number of the specific order
    @FXML
    void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException, ParseException {
        if(searchField.getText().isEmpty())
            refreshOrderTable(getOrdersInLastMonth());
        else
            refreshOrderTable(Conditions.getItemsSearch(getOrdersInLastMonth(),searchField.getText()));
    }

    // refresh order table
    private void refreshOrderTable(ObservableList<Order> ordersList) {
        Collections.sort(ordersList); // sort the list
        orderTable.setItems(ordersList);
        orderIdCol.setCellValueFactory(new PropertyValueFactory<Order,Integer>("orderId"));
        customerIdCol.setCellValueFactory(new PropertyValueFactory<Order,Integer>("customerId"));
        totalPriceCol.setCellValueFactory(new PropertyValueFactory<Order,Double>("totalPrice"));
        orderDateCol.setCellValueFactory(new PropertyValueFactory<Order,String>("orderDate"));
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        try {
            refreshOrderTable(getOrdersInLastMonth()); // refresh orders table
        } catch (ClassNotFoundException | SQLException | ParseException e) {
            e.printStackTrace();
        }
    }

    // this function return all order in the current order
    private ObservableList<Order> getOrdersInLastMonth() throws ClassNotFoundException, SQLException, ParseException
    {
        ObservableList<Order> list = FXCollections.observableArrayList();
        ObservableList<Order> ordersList = DatabaseManager.getOrders();

        DateClass date = new DateClass();
    }
}
```

```

        for(Order order : ordersList) {
            // if the month of the order has been placed equal to current month and the order year equal to the
current year
            if(date.getMonth(order.getOrderDate()) == date.getCurrentMonth() &&
date.getYear(order.getOrderDate()) == date.getCurrentYear()) {
                list.add(order); // if true add the order to list
            }
        }
        Collections.sort(list); // sort the list
        return list;
    }
}

```

## קובץ OrderNumberController.java

המחלקה שולטת על חלון המציג כמה הזמנות כל לקוח הזמין.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.Collections;
import java.util.ResourceBundle;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;

public class OrdersNumberController implements Initializable{

    @FXML
    private TableView<Customer> productTable;

    @FXML
    private TableColumn<Customer, Integer> customerIdCol,ordersNumberCol;

    @FXML
    private TableColumn<Customer, String> customerNameCol,phoneCol;

    @FXML
    private TextField searchField;

    // switch page
    // switch to statistics window
    @FXML
    void backToOrderPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"statistics.fxml","Management products - Statistics");
    }
    //refresh customers table
    private void refreshTable(ObservableList<Customer> list) throws ClassNotFoundException, SQLException {
        customerIdCol.setCellValueFactory(new PropertyValueFactory<Customer,Integer>("id"));
        ordersNumberCol.setCellValueFactory(new PropertyValueFactory<Customer,Integer>("numberOfOrders"));
        customerNameCol.setCellValueFactory(new PropertyValueFactory<Customer,String>("fullName"));
        phoneCol.setCellValueFactory(new PropertyValueFactory<Customer,String>("telephone"));
        productTable.setItems(list);
    }

    // this function return all the orders number of the customers
    public ObservableList<Customer> getCustomersSorted() throws ClassNotFoundException, SQLException{
        ObservableList<Customer> customersList = DatabaseManager.getCustomer(); // get all customer
        ObservableList<Order> ordersList = DatabaseManager.getOrders(); // get all order
        ObservableList<Customer> newList = FXCollections.observableArrayList();

        for(Customer c : customersList) {
            Customer customer = new Customer(c.getId(),c.getFullName(),c.getTelephone());
            customer.setOrdersList(ordersList); // set orders list
            customer.calculateOrdersNumber(customer); // calculate orders number of the customer according to id of the
customer
            newList.add(customer); // add customer to list
        }
        // sort customers list
        Collections.sort(newList);
        return newList;
    }

    // search customer from customers list
    @FXML
    void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
        if(searchField.getText().isEmpty())
            refreshTable(getCustomersSorted());
        else
            refreshTable(Conditions.getItemssSearch(getCustomersSorted(),searchField.getText()));
    }

    @Override
```

```
public void initialize(URL arg0, ResourceBundle arg1) {  
    // TODO Auto-generated method stub  
    try {  
        refreshTable(getCustomersSorted());  
    } catch (ClassNotFoundException | SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```

## קובץ ProductInTheCategoryController.java

המחלקה שולטת על חלון המציג כמה הזמנות כל לקוח הזמין.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.util.Collections;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;

public class ProductInTheCategoryController implements Initializable {

    @FXML
    private TableView<Category> categoryTable;

    @FXML
    private TableColumn<Category, Integer> categoryIdCol, productsNumberCol;

    @FXML
    private TableColumn<Category, String> categoryNameCol;

    @FXML
    private TextField searchField;

    // switch page
    // switch to statistics window
    @FXML
    void backToOrderPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event, "statistics.fxml", "Management products - Statistics");
    }

    // this function return products number in the categories
    private ObservableList<Category> getCategories() throws ClassNotFoundException, SQLException {
        ObservableList<Category> categorysList = DatabaseManager.getCategorys(); // get all categories
        ObservableList<Category> newCategorysList = FXCollections.observableArrayList();

        for(Category category : categorysList) {
            Category newCategory = new Category(category.getCategoryId(), category.getCategoryName());
            category.setProductList(DatabaseManager.getProducts()); // set all products in the category
            newCategory.setProductNumber(newCategory.getNumberOfProduct(category)); // calculate number of
products in the category
            newCategorysList.add(newCategory); // add category to new list
        }

        Collections.sort(newCategorysList); // sort the list of categories
        return newCategorysList;
    }

    // refresh products table
    private void refreshTable(ObservableList<Category> list) throws ClassNotFoundException, SQLException {
        categoryIdCol.setCellValueFactory(new PropertyValueFactory<Category, Integer>("categoryId"));
        categoryNameCol.setCellValueFactory(new PropertyValueFactory<Category, String>("categoryName"));
        productsNumberCol.setCellValueFactory(new PropertyValueFactory<Category, Integer>("productNumber"));
        categoryTable.setItems(list);
    }

    // search category from categories table
    @FXML
    void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
        if(searchField.getText().isEmpty())
            refreshTable(getCategories());
        else
            refreshTable(Conditions.getItemssSearch(getCategories(), searchField.getText()));
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        try {
```



```
        refreshTable(getCategories()); // refresh categories table
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}
```

## קובץ ExpensesController.java

המחלקה שולטת על חלון המציג כמה הזמנות כל לקוח הזמין.

```
package application;

import java.io.IOException;
import java.net.URL;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.ResourceBundle;

import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.Label;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;

public class ExpensesController implements Initializable{

    @FXML
    private TableView<User> employeesTable;

    @FXML
    private TableColumn<User, String> fullNameCol,salaryDateCol,userNameCol;

    @FXML
    private TableColumn<User, Double> salaryDayCol,generalSalaryCol;

    @FXML
    private AnchorPane side;

    @FXML
    private TextField searchField;

    @FXML
    private Label currentMonthLabel;

    @FXML
    private TextField electricalExpField,waterExpField;
    DateClass date = new DateClass();
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("dd/MM/yyyy");

    /*      switch pages */

    // switch to category window
    @FXML
    void CategoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CategoryPage.fxml","Management products - Category");
    }
    // switch to customer window
    @FXML
    void CustomerPage(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"CustomerPage.fxml","Management Product - Customer");
    }
    // switch to inventory window
    @FXML
    void inventoryBtn(ActionEvent event) throws IOException {
        SwitchingPages.switchPage(event,"inventory.fxml","Management Product - Inventory");
    }
    // switch to login window
    @FXML
    void logoutBtn(ActionEvent event) throws IOException {
        SwitchingPages.openWindowInCenter("Login",event,"Management products - Login");
    }
    // menu clicked
    @FXML
    void menuClicked(MouseEvent event) {
        new AnimationClass().sideAnimation(side);
    }
}
```

```

// switch to order window
@FXML
void orderPageBtn(ActionEvent event) throws IOException {
    SwitchingPages.switchPage(event, "OrderPage.fxml", "Management Product - Order");
}
// switch to shelves window
@FXML
void shelvesBtn(ActionEvent event) throws IOException {
    SwitchingPages.switchPage(event, "shelvesPage.fxml", "Management Product - Shelves");
}
// switch to statistics window
@FXML
void statisticsBtn(ActionEvent event) throws IOException {
    SwitchingPages.switchPage(event, "statistics.fxml", "Management Product - Statistics");
}
//this function search employee from the table
@FXML
void searchFieldTyped(KeyEvent event) throws ClassNotFoundException, SQLException {
    if(searchField.getText().isEmpty())
        refreshTable(DatabaseManager.getEmployees());
    else
        refreshTable(Conditions.getItemssSearch(DatabaseManager.getEmployees(),searchField.getText()));
}
// update salary button
@FXML
void updateSalaryBtn(ActionEvent event) throws InterruptedException, ClassNotFoundException, SQLException, ParseException {
    if(!employeesTable.getSelectionModel().isEmpty()) {
        Employee selectedEmployee = (Employee) employeesTable.getSelectionModel().getSelectedItem();
        Double salaryDay = (double) Message.showMessageDialog("Employee salary", "Enter salary day of the employee
: ");

        Employee employee = new Employee();
        // if the selected salary bigger than zero
        if(salaryDay > -1) {
            //get the employee data
            if((employee = (Employee) selectedEmployee.checkUserName(DatabaseManager.getEmployees(),
selectedEmployee.getUserName()))!= null) {
                if(employee.getUpdateSalaryDate().equals("-")) { // if update date equal to '-' so the employee did
not receive a salary for the day
                    employee.setUpdateSalaryDate(new DateClass().getDateToday()); // update the date to
the current date

                    employee.setSalaryDay(salaryDay);
                    employee.setGeneralSalary(employee.getGeneralSalary() + salaryDay); // add the general
salary of the employee and the selected salary

                    DatabaseManager.editEmployee(employee);//edit employee
                } else {
                    Date updateSalaryDate = simpleDateFormat.parse(employee.getUpdateSalaryDate());
                    Date todayDate = simpleDateFormat.parse(new DateClass().getDateToday());
                    // if the employee received a salary per day
                    // checks the last date of the employee received salary per day
                    if(updateSalaryDate.equals(todayDate)) {
                        if(Message.showMessageDialog(AlertType.INFORMATION, "Do you want
to update the salary day?", null) == buttonType.YES) {
                            if(employee.getSalaryDay() > salaryDay) // if the salary day bigger than
selected salary per day
                                employee.setGeneralSalary(employee.getGeneralSalary() -
(employee.getSalaryDay() - salaryDay));

                            else //if the selected salary day bigger than salary day
                                employee.setGeneralSalary(employee.getGeneralSalary() +
(salaryDay - employee.getSalaryDay()));

                            employee.setSalaryDay(salaryDay);
                            DatabaseManager.editEmployee(employee);//edit employee
                        }
                    }
                }
            }
            refreshTable(DatabaseManager.getEmployees());
        }
    } else
        Message.showMessageDialog(AlertType.WARNING, "Choose employee form the table", null);
}
//refresh employees table
private void refreshTable(ObservableList<User> list) {
    fullNameCol.setCellValueFactory(new PropertyValueFactory<User,String>("fullName"));
    userNameCol.setCellValueFactory(new PropertyValueFactory<User,String>("userName"));
    salaryDayCol.setCellValueFactory(new PropertyValueFactory<User,Double>("salaryDay"));
    generalSalaryCol.setCellValueFactory(new PropertyValueFactory<User,Double>("generalSalary"));
    salaryDateCol.setCellValueFactory(new PropertyValueFactory<User,String>("updateSalaryDate"));
}

```

```

        employeesTable.setItems(list);
    }
    // update employee date
    private void updateEmployee() throws ClassNotFoundException, SQLException, ParseException {
        for(User employee: DatabaseManager.getEmployees()) {
            if(!((Employee) employee).getUpdateSalaryDate().equals("-")) {
                Date updateSalaryDate = simpleDateFormat.parse(((Employee) employee).getUpdateSalaryDate()); // get the
last date of the employee get salary
                Date todayDate = simpleDateFormat.parse(new DateClass().getDateToday()); // current date

                if(!todayDate.equals(updateSalaryDate)) { // if the update date salary of the employee not equal to current date
                    ((Employee) employee).setUpdateSalaryDate("-");
                    ((Employee) employee).setSalaryDay(0.0);
                    DatabaseManager.editEmployee((Employee) employee); // edit the employee
                }
            }
        }
    }
    //side
    private void sideManaged() {
        side.setTranslateX(-215);
        side.setVisible(false);
        side.setManaged(false);
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        // TODO Auto-generated method stub
        try {
            updateEmployee();
            sideManaged();
            refreshTable(DatabaseManager.getEmployees());
            currentMonthLabel.setText(date.getMonth(date.getCurrentMonth()) + " month");
        } catch (ClassNotFoundException | SQLException | ParseException e) {
            e.printStackTrace();
        }
    }

    // this fuction check if the manager paid expenses type in current month
    private void isExpensesPaid(Expenses expenses,ObservableList<Expenses> expList) throws ClassNotFoundException, SQLException,
    ParseException {
        // if the manager paid expenses type
        if(expenses.isExpensesPaid(expList) != null) {
            DatabaseManager.editExpensess(expenses); // update expenses type
            Message.showMessageDialogWithOKButton(AlertType.INFORMATION,"update " +
expenses.getDescription().toLowerCase() + " expenses in " + date.getMonth(date.getCurrentMonth()).toLowerCase() + " month", null);
        } else { // if the manager not paid expenses type
            DatabaseManager.addExpensess(expenses); // add expenses type
            Message.showMessageDialogWithOKButton(AlertType.INFORMATION,"add " +
expenses.getDescription().toLowerCase() + " expenses in " + date.getMonth(date.getCurrentMonth()).toLowerCase() + " month", null);
        }
    }

    // electrical update button
    // this function update electrical expenses in current month
    @FXML
    void electricalUpdateBtn(ActionEvent event) throws ClassNotFoundException, SQLException, ParseException {
        Field field = new Field();
        // if electrical field is empty
        if(field.checkEmptyFields(electricalExpField)) {
            if(Conditions.checkFloatNumber(electricalExpField)) {
                field.changeBorderColor(electricalExpField, "black");
                DateClass date = new DateClass();
                Expenses electrical = new Electrical(Double.parseDouble(electricalExpField.getText()), date.getDateToday());
                // update electrical expenses for the current month
                ObservableList<Expenses> expList = DatabaseManager.getExpensesType("Electrical");
                electricalExpField.setText("");
                isExpensesPaid(electrical,expList);
            } else { // if content of field is not valid
                field.changeBorderColor(electricalExpField, "red");
                Message.showMessageDialogWithOKButton(AlertType.ERROR, "Enter a float number", null);
            }
        } else { // if field is empty
            field.changeBorderColor(electricalExpField, "red");
            Message.showMessageDialogWithOKButton(AlertType.WARNING, "electrical expenses field is empty", null);
        }
    }

    // update water expenses field in button
    // this function update water expenses in current month
    @FXML
    void waterUpdateField(ActionEvent event) throws ClassNotFoundException, SQLException, ParseException {

```

```

Field field = new Field();
if(field.checkEmptyFields(waterExpField)) { // check if water expenses field is not empty
    if(Conditions.checksFloatNumber(waterExpField)) { // check if the content in the field is valid
        field.changeBorderColor(waterExpField, "black");
        Expenses water = new Water(Double.parseDouble(waterExpField.getText()), date.getDateToday());
        // update water expenses for the current month
        ObservableList<Expenses> expList = DatabaseManager.getExpensesType("Water");
        waterExpField.setText("");
        isExpensesPaid(water,expList); // this function checks if the the manager update the expenses type in current
month

    } else { // if content of field is not valid
        field.changeBorderColor(waterExpField, "red");
        Message.showMessageWithOKButton(AlertType.WARNING, "Enter a float number", null);
    }
} else { // if field is empty
    field.changeBorderColor(waterExpField, "red");
    Message.showMessageWithOKButton(AlertType.WARNING, "Water expenss field is empty", null);
}
}

// show electrical expenses in message button
// this function show message contains total electrical expenses in the current month
@FXML
void showElectricalExpBtn(ActionEvent event) throws ClassNotFoundException, SQLException, ParseException {
    Expenses expenses = new Expenses();
    ObservableList<Expenses> expList = DatabaseManager.getExpensesType("Electrical");
    // if the manager update the electrical expenses in current month
    if((expenses = expenses.isExpensesPaid(expList)) != null) {
        // show message contains electrical expenses of the bussniss in the current month
        Message.showMessageWithOKButton(AlertType.INFORMATION, "Your total electrical expenses is " +
expenses.getTotal() + " in this month", null);
    } else // if the manager not paid the electrical expenses in current month
        Message.showMessageWithOKButton(AlertType.WARNING, "you not paid water expenses in current month" , null);
    }

    // show water expenses message button
    // this function show message contains total water expenses in the current month
    @FXML
    void showWaterExpBtn(ActionEvent event) throws ClassNotFoundException, SQLException, ParseException {
        Expenses expenses = new Expenses();
        ObservableList<Expenses> expList = DatabaseManager.getExpensesType("Water"); // get all water expenses
        // if the manager update in current month
        if((expenses = expenses.isExpensesPaid(expList)) != null) {
            // show message contains water expenses of the bussniss in the current month
            Message.showMessageWithOKButton(AlertType.INFORMATION, "Your total water expenses is " + expenses.getTotal()
+ " in this month", null);
        } else // if the manager not paid the water expenses in current month
            Message.showMessageWithOKButton(AlertType.WARNING, "you not paid water expenses in current month" , null);
        }
    }
}

```

## קובץ DatabaseManager.java

מחלקת ניהול בסיס הנתונים.

```
package application;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.ParseException;
import com.mysql.jdbc.PreparedStatement;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

public class DatabaseManager {

    static Connection conn;
    static PreparedStatement preparedStmt;
    static ResultSet rs;
    static Statement st;

    //connection to data base
    public static Connection getConn() throws ClassNotFoundException {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Management_inventory?autoReconnect=true&useSSL=false", "root", "");
            return conn;
        } catch (SQLException e) {
            System.out.println(e);
            return null;
        }
    }

    //***** Insert -> Delete -> get data (Select) -> Update *****//

    //***** Insert query *****//

    //add new category to the system
    public static void addCategory(Category category) throws SQLException, ClassNotFoundException {
        conn();
        preparedStmt = (PreparedStatement) conn.prepareStatement("insert into category values(?,?)");
        preparedStmt.setString(1, null); //Increment number -> (category id)
        preparedStmt.setString(2, category.getCategoryName()); //category name
        preparedStmt.execute();
    }

    // this function add new product
    public static void addProduct(Product product) throws SQLException, ClassNotFoundException {
        conn();
        preparedStmt = (PreparedStatement) conn.prepareStatement("insert into product values(?,?,?,?,?,?,?)"); //insert new product
        preparedStmt.setInt(1, product.getProductCode()); //product code
        preparedStmt.setString(2, product.getProductName()); //produce name
        preparedStmt.setString(3, product.getCategory()); //category name
        preparedStmt.setDouble(4, product.getPrice()); //price
        preparedStmt.setInt(5, product.getQuantity()); //quantity
        preparedStmt.setDouble(6, product.getSupplierPrice()); //supplier price
        preparedStmt.setDouble(7, product.getWeight()); //weight
        preparedStmt.execute();
    }

    //this function add new product to shelf
    public static void addProductToShelf(Product product, Shelf shelf, int quantity) throws SQLException, ClassNotFoundException {
        conn();
        preparedStmt = (PreparedStatement) conn.prepareStatement("insert into shelf_product values(?,?,?,?,?,?,?)");
        preparedStmt.setInt(1, product.getShelf(shelf.getShelfNumber()).getShelfNumber()); //shelf number
        preparedStmt.setInt(2, product.getProductCode()); //product code
        preparedStmt.setString(3, product.getProductName()); //product name
        preparedStmt.setInt(4, quantity); //quantity
        preparedStmt.setDouble(5, product.getWeight()); //weight
        preparedStmt.setString(6, product.getCategory()); //category name
        preparedStmt.setDouble(7, product.getPrice()); // client price
        preparedStmt.setDouble(8, product.getSupplierPrice()); //Supplier price
        preparedStmt.execute();
    }

    // add shelf to shelf table in database
```

```

public static void addShelf(Shelf shelf) throws SQLException, ClassNotFoundException {
    conn();
    preparedStmt = (PreparedStatement) conn.prepareStatement("insert into shelf values(?,?)");
    preparedStmt.setInt(1, shelf.getShelfNumber()); //shelf number
    preparedStmt.setDouble(2, shelf.getMaxWeight()); // max weight
    preparedStmt.execute();
}
// add new user to the system
public static void addUser(User user) throws SQLException, ClassNotFoundException {
    conn();
    preparedStmt=(PreparedStatement) conn.prepareStatement("insert into user values(?,?,?,?,?,?,?)"); // insert new user
    preparedStmt.setString(1, user.getFirstName()); //first name
    preparedStmt.setString(2, user.getLastName()); //last name
    preparedStmt.setString(3, user.getEmail()); //email
    preparedStmt.setInt(4, user.getId()); //id
    preparedStmt.setString(5, user.getUserName()); //user name
    preparedStmt.setString(6, user.getPassword()); // password
    preparedStmt.setDouble(7, 0.0); // salary
    preparedStmt.setDouble(8, 0.0); // General salary
    preparedStmt.setString(9, "-"); // update salary date
    preparedStmt.execute();
}
// this function add new customer
public static void addCustomer(Customer customer) throws SQLException, ClassNotFoundException {
    conn();
    preparedStmt = (PreparedStatement) conn.prepareStatement("insert into customer values(?,?,?,?,?)"); //insert
    preparedStmt.setString(1, customer.getFullName()); //full name
    preparedStmt.setInt(2, customer.getId()); //id
    preparedStmt.setString(3, customer.getTelephone()); //Telephone
    preparedStmt.setString(4, customer.getEmail()); //email
    preparedStmt.setString(5, customer.getAddress()); //address
    preparedStmt.execute();
}
//insert product to shopping cart table in database
public static void insertProductToShoppingCart(Product product) throws SQLException, ClassNotFoundException {
    conn();
    preparedStmt = (PreparedStatement) conn.prepareStatement("insert into shopping_cart values(?,?,?,?,?,?,?)");
    preparedStmt.setString(1, product.getProductName()); //product name
    preparedStmt.setDouble(2, product.getPrice()); //price
    preparedStmt.setInt(3, product.getQuantity()); //quantity
    preparedStmt.setInt(4, product.getProductCode()); //product code
    preparedStmt.setDouble(5, product.getSupplierPrice()); //supplier price
    preparedStmt.setDouble(6, product.getWeight()); // weight of the product
    preparedStmt.setString(7, product.getCategory()); // weight of the product
    preparedStmt.execute();
}
// this function add expenses type (water or electrical)
public static void addExpensess(Expenses exp) throws ClassNotFoundException, SQLException {
    conn();
    DateClass date = new DateClass();
    String query = "insert into expenss values(?,?,?,?,?)";
    preparedStmt = (PreparedStatement) conn.prepareStatement(query);
    preparedStmt.setString(1, exp.getDescription()); // description
    preparedStmt.setDouble(2, exp.getTotal()); // total of expenses type
    preparedStmt.setInt(3, date.getCurrentMonth()); // current month
    preparedStmt.setInt(4, date.getCurrentYear()); // current year
    preparedStmt.setString(5, exp.getExpDate());
    preparedStmt.execute();
}
/***** Remove query *****/

public static void removeOrderQuery(Order order,String tableName) throws SQLException, ClassNotFoundException {
    conn();
    preparedStmt = (PreparedStatement) conn.prepareStatement("DELETE FROM " + tableName + " WHERE orderId="+
order.getId());
    preparedStmt.execute();
}
//delete specific category
public static void deleteCategory(Category category) throws SQLException, ClassNotFoundException {
    conn();
    preparedStmt = (PreparedStatement) conn.prepareStatement("DELETE FROM category WHERE categoryId=?");
    preparedStmt.setInt(1, category.getCategoryId());
    preparedStmt.execute();
}
//remove specific shelf
public static void removeShelf(Shelf shelf) throws SQLException, ClassNotFoundException {
    conn();

```

```

        preparedStmt = (PreparedStatement) conn.prepareStatement("DELETE FROM shelf WHERE shelfNumber=?");
        preparedStmt.setInt(1, shelf.getShelfNumber());
        preparedStmt.execute();
    }
    //remove specific customer
    public static void removeCustomer(Customer customer) throws SQLException, ClassNotFoundException {
        conn();
        preparedStmt = (PreparedStatement) conn.prepareStatement("DELETE FROM customer WHERE id=?");
        preparedStmt.setInt(1, customer.getId()); //id of customer
        preparedStmt.execute();
    }
    // remove product from specific order
    public static void deleteProductInOrder(Product product, Order order) throws ClassNotFoundException, SQLException {
        conn();
        String query = "DELETE FROM order_product WHERE productId=? AND orderId=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setInt(1, product.getProductCode());
        product.setOrdersList(getOrders());
        preparedStmt.setInt(2, product.getCurrentOrder(order.getId()).getId());
        preparedStmt.execute();
    }
    public static void deleteProductInShelf(String tableName, Product product, Shelf shelf) throws SQLException,
    ClassNotFoundException {
        conn();
        String query = "DELETE FROM " + tableName + " WHERE productId=? AND shelfNumber=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setInt(1, product.getProductCode());
        product.setShelvesList(getShelves());
        preparedStmt.setInt(2, product.getShelf(shelf.getShelfNumber()).getShelfNumber());
        preparedStmt.execute();
    }
    //this function to delete specific product
    public static void deleteProduct(Product product, String tableName) throws SQLException, ClassNotFoundException {
        conn();
        //delete from table name where product id equals to specific product id
        preparedStmt = (PreparedStatement) conn.prepareStatement("DELETE FROM " + tableName + " WHERE productId=?");
        preparedStmt.setInt(1, product.getProductCode());
        preparedStmt.execute();
    }
    //clear the shopping cart table
    public static void removeShoppingCart() throws SQLException, ClassNotFoundException {
        conn();
        preparedStmt = (PreparedStatement) conn.prepareStatement("DELETE FROM shopping_cart");
        preparedStmt.execute();
    }
    /***** Get all data of items *****/

    //this function return all products in the shopping cart
    public static ObservableList<Product> getProductFromShoppingCart() throws ClassNotFoundException, SQLException {
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM shopping_cart"); // select from the shopping cart table
        ObservableList<Product> productList = FXCollections.observableArrayList();

        while(rs.next()) {
            Product productAddToShoppingCart = new
Product(rs.getInt("productId"), rs.getString("product_name"), rs.getInt("quantity"), rs.getDouble("price"), rs.getDouble("weight"), rs.getDouble("supplierPrice"), rs.getString("categoryName"));
            productList.add(productAddToShoppingCart); //add all product in shopping cart to the list
        }

        return productList;
    }

    //this function return all products in the shelf
    public static ObservableList<Product> getProductsInShelf(Shelf shelf) throws ClassNotFoundException, SQLException {
        conn();
        ObservableList<Product> productList = FXCollections.observableArrayList();

        st = conn.createStatement();
        //select the shelf_product table where id of the shelf
        rs = st.executeQuery("SELECT * FROM shelf_product WHERE shelfNumber = " + shelf.getShelfNumber() + "");

        while(rs.next()) {
            Product product = new
Product(rs.getInt("shelfNumber"), rs.getInt("productId"), rs.getString("productName"), rs.getInt("quantity"), rs.getDouble("weight"), rs.getString("categoryName"), rs.getDouble("price"), rs.getDouble("supplierPrice"));
            productList.add(product); //add all products to list
        }

        return productList;
    }

```



```

    }

    //this function get all orders in database
    public static ObservableList<Order> getOrders() throws ClassNotFoundException, SQLException {
        ObservableList<Order> ordersList = FXCollections.observableArrayList();
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM orders");//select the table of orders
        while(rs.next()) {
            Order order = new
Order(rs.getInt("orderId"),rs.getInt("customerId"),rs.getDouble("totalPrice"),rs.getString("orderDate"));
            ordersList.add(order);//add all orders to list
        }

        return ordersList;
    }
    //this function get all shelves in database
    public static ObservableList<Shelf> getShelves() throws ClassNotFoundException {
        ObservableList<Shelf> shelvesList = FXCollections.observableArrayList();
        try {
            conn();
            st = conn.createStatement();
            rs = st.executeQuery("SELECT * FROM shelf");
            while(rs.next()) {
                Shelf shelf = new Shelf(rs.getInt("shelfNumber"),rs.getDouble("maxWeight"));
                shelvesList.add(shelf);//add all shelves to list
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return shelvesList;
    }

    //this function return all categories
    public static ObservableList<Category> getCategorys() throws ClassNotFoundException, SQLException {
        conn();
        ObservableList<Category> categoryList = FXCollections.observableArrayList();

        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM category");//select the table of the category
        while(rs.next()) {
            Category category = new Category(rs.getInt("categoryId"),rs.getString("categoryName"));
            categoryList.add(category);//add all categories to list
        }

        return categoryList;
    }

    //this function return all users exists in the system
    public static ObservableList<User> getUsers() throws SQLException, ClassNotFoundException {
        ObservableList<User> userList = FXCollections.observableArrayList();
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * from user");//select table from user table in data base;

        // add all users to list
        while(rs.next()){
            User user = new
User(rs.getString("fname"),rs.getString("lname"),rs.getString("email"),rs.getInt("id"),rs.getString("username"),rs.getString("password"));
            userList.add(user);
        }
        //return all users
        return userList;
    }

    //get all employees in data base
    public static ObservableList<User> getEmployees() throws SQLException, ClassNotFoundException {
        ObservableList<User> employeesList = FXCollections.observableArrayList();
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * from user");//select table from employee table in data base;

        // add all employees to list
        while(rs.next()){
            User employee = new Employee(rs.getString("userName"),rs.getString("fname") + " " +
rs.getString("lname"),rs.getDouble("salaryDay"),rs.getDouble("generalSalary"),rs.getString("updateSalaryDate"));
            employeesList.add(employee);//add all employee to list
        }

        //return all employees
        return employeesList;
    }

```

```

    }

    // this function return the data of the manager
    public static Manager getManager() throws SQLException, ClassNotFoundException {
        Manager manager;
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * from manager");//select table from manager table in data base;
        rs.last();
        manager = new
Manager(rs.getString("fname"),rs.getString("lname"),rs.getString("email"),rs.getInt("id"),rs.getString("username"),rs.getString("password"),rs.ge
tBoolean("status"));
        return manager;//return the manager
    }
    //this function return all the products in storage
    public static ObservableList<Product> getProducts() throws ClassNotFoundException, SQLException {
        ObservableList<Product> productList = FXCollections.observableArrayList();
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM product");//select all products

        while(rs.next()) {
            Product product = new Product(rs.getInt("productId"), rs.getString("productName"),
rs.getString("category"), rs.getDouble("supplierPrice"),rs.getDouble("price"),rs.getInt("quantity"),rs.getDouble("weight"));
            productList.add(product);//add all products to the list
        }

        return productList;//return all products
    }
    public static int getProductQunatityInOrder(Product product,Order order) throws SQLException, ClassNotFoundException
{
        conn();
        st = conn.createStatement();
        // select from the specific table where product id of the product and order id of product
        rs = st.executeQuery("SELECT * FROM order_product WHERE productId = " + product.getProductCode() + "
AND orderId = " + order.getOrderid() + "");
        rs.last();
        return rs.getInt("quantity"); //return quantity
    }
    // this function get specific product quantity
    public static int getProductQuantityInShelf(String tableName,Product product,Shelf shelf) throws SQLException,
ClassNotFoundException {
        conn();
        st = conn.createStatement();

        // select from the specific table where product id of the product
        rs = st.executeQuery("SELECT * FROM " + tableName + " WHERE productId = " + product.getProductCode()
+ " AND shelfNumber = " + shelf.getShelfNumber() + "");
        rs.last();
        return rs.getInt("quantity"); //return quantity
    }

    //this function get all customers in database
    public static ObservableList<Customer> getCustomer() throws ClassNotFoundException, SQLException {
        ObservableList<Customer> customerList = FXCollections.observableArrayList();

        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM customer");
        while(rs.next()) {
            Customer customer = new
Customer(rs.getString("name"),rs.getInt("id"),rs.getString("email"),rs.getString("telephone")
, rs.getString("address"));
            customerList.add(customer);//add customer to list
        }

        return customerList;//return all customers in database
    }

    private static void conn() throws ClassNotFoundException {
        if(conn == null)
            conn = getConn();
    }

    //get all products of orders
    public static ObservableList<Product> getProductsOfOrders() throws ClassNotFoundException, SQLException {
        ObservableList<Product> productsList = FXCollections.observableArrayList();
        conn();

```

```

        st = conn.createStatement();
        //select from order_product table

        rs = st.executeQuery("SELECT * FROM order_product");

        while(rs.next()) {
            Product product = new Product(rs.getInt("productId"),
rs.getString("product_name"),rs.getInt("quantity"), rs.getDouble("price"),rs.getDouble("supplierPrice"));
            productsList.add(product);//add all products to the list
        }

        return productsList;
    }
    //this function return all products of specific order
    public static ObservableList<Product> getProductOfOrder(Order order) throws ClassNotFoundException, SQLException
    {
        ObservableList<Product> productsList = FXCollections.observableArrayList();
        conn();
        st = conn.createStatement();
        // select from order_product table where order id equal to order id in the product
        rs = st.executeQuery("SELECT * FROM order_product WHERE orderId =" + order.getOrderId()
+""");

        while(rs.next()) {
            Product product = new Product(order.getOrderId(),rs.getInt("productId"),
rs.getString("product_name"), rs.getDouble("price"),
rs.getInt("quantity"),rs.getDouble("weight"),rs.getDouble("supplierPrice"),rs.getString("categoryName"));
            productsList.add(product);//add all the products of specific order to list
        }

        return productsList;
    }

    //this function return all products in the shelves
    public static ObservableList<Product> getProductsInShelves() throws ClassNotFoundException, SQLException {
        ObservableList<Shelf> shelvesList = DatabaseManager.getShelves(); // get all the shelves
        ObservableList<Product> productsInShelves = FXCollections.observableArrayList();

        for(Shelf shelf : shelvesList) {
            ObservableList<Product> productsInShelf = DatabaseManager.getProductsInShelf(shelf);
            for(Product product : productsInShelf)
                productsInShelves.add(product); // add all the products in the shelf to list
        }
        // return all the product in the shelves
        return productsInShelves;
    }

    // this function get specific product quantity
    public static int getProductQuantity(String tableName,int id) throws SQLException, ClassNotFoundException {
        conn();
        st = conn.createStatement();
        //select from the specific table where product id of the product
        rs = st.executeQuery("SELECT * FROM "+ tableName +" WHERE productId =" + id +""");
        rs.last();
        return rs.getInt("quantity");//return quantity
    }

    //this function return the last order code inserted to database
    //insert the order to the orders table in database
    public static int getOrderCode(Order order) throws SQLException, ClassNotFoundException {
        conn();
        String query = "insert into orders values(?,?,?,?)";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query,
Statement.RETURN_GENERATED_KEYS);
        preparedStmt.setString(1,null);//increment number
        preparedStmt.setDouble(2,order.getTotalPrice());//total price of order
        preparedStmt.setInt(3,order.getCustomerId());//customer id
        preparedStmt.setString(4,order.getOrderDate());//date of the order
        preparedStmt.execute();
        //get last inserted row in database
        ResultSet rs = preparedStmt.getGeneratedKeys();
        rs.next();
        return rs.getInt(1);
    }

    public static ObservableList<Expenses> getExpenss() throws ClassNotFoundException, SQLException {
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM expenss");

        ObservableList<Expenses> expList = FXCollections.observableArrayList();

```

```

        Expenses expenses = new Expenses();

        while(rs.next()) {
            if(rs.getString("description").equals("Water")) {
                expenses = new Water(rs.getDouble("total"),rs.getString("expDate"));
            } else
                expenses = new Electrical(rs.getDouble("total"),rs.getString("expDate"));

            expList.add(expenses);
        }
        return expList;
    }

    public static ObservableList<Expenses> getExpensesType(String description) throws SQLException,
    ClassNotFoundException{
        ObservableList<Expenses> expList = FXCollections.observableArrayList();

        for(Expenses exp : getExpenses()) {
            if(exp.getDescription().equals(description))
                expList.add(exp);
        }

        return expList;
    }

    //this function for add all product in shopping cart to specific order
    public static void addProductToOrder(int orderNumber) throws SQLException, ClassNotFoundException {
        conn();
        st = conn.createStatement();
        rs = st.executeQuery("SELECT * FROM shopping_cart");

        //ObservableList<Product> productsList = FXCollections.observableArrayList();
        Order order = new Order(orderNumber);

        while(rs.next()) {
            Product product = new
            Product(rs.getInt("productId"),rs.getString("product_name"),rs.getInt("quantity"),rs.getDouble("price"),rs.getDouble("weight"),rs.getDouble("sup
            llierPrice"),rs.getString("categoryName"));
            insertProductToOrder(product,order);
        }

        //insert the product to order_product table
        private static void insertProductToOrder(Product product,Order order) throws SQLException, ClassNotFoundException {
            conn();
            String query = "insert into order_product values(?,?,?,?,?,?,?)";
            preparedStmt = (PreparedStatement) conn.prepareStatement(query);
            preparedStmt.setInt(1, order.getOrderid()); //order code
            preparedStmt.setInt(2, product.getProductCode()); //product code
            preparedStmt.setString(3, product.getProductname()); //product name
            preparedStmt.setInt(4, product.getQuantity()); //quantity
            preparedStmt.setDouble(5, product.getPrice()); //price
            preparedStmt.setDouble(6, product.getSupplierPrice()); //Supplier price
            preparedStmt.setDouble(7, product.getWeight()); //weight of the product
            preparedStmt.setString(8, product.getCategory()); //weight of the product
            preparedStmt.execute();
        }

        /***** Update query *****/

        //this function update product quantity in shelf
        public static void updateProductInShelf(String tabelName,Product product,Shelf shelf,int quantity) throws ClassNotFoundException,
        SQLException {
            conn();
            String query = "update " + tabelName + " set quantity=? where productId=? AND shelfNumber=?";
            preparedStmt = (PreparedStatement) conn.prepareStatement(query);
            preparedStmt.setInt(1, quantity);
            preparedStmt.setInt(2, product.getProductCode());
            product.setShelvesList(getShelves());
            preparedStmt.setInt(3, product.getShelf(shelf.getShelfNumber()).getShelfNumber());
            preparedStmt.execute();
        }

        //update quantity of product
        public static void updateProduct(String tableName,Product product,int quantity) throws SQLException, ClassNotFoundException {
            conn();
            String query = "update " + tableName + " set quantity = ? where productId =?";
            preparedStmt = (PreparedStatement) conn.prepareStatement(query);
            preparedStmt.setInt(1,quantity); //update the quantity
            preparedStmt.setInt(2,product.getProductCode()); // product code
            preparedStmt.execute();
        }
    }

```

```

    }

    //update quantity of product in order
    public static void updateProductInOrder(Product product, Order order, int quantity) throws SQLException, ClassNotFoundException {
        conn();
        String query = "update order_product set quantity = ? where productId =? AND orderId=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setInt(1, quantity); //update the quantity
        preparedStmt.setInt(2, product.getProductCode()); // product code
        product.setOrdersList(getOrders());
        preparedStmt.setInt(3, product.getCurrentOrder(order.getOrderId()).getOrderId()); // product code
        preparedStmt.execute();
    }

    public static void updateProductCategory(Category newCategory, Category oldCategory) throws SQLException,
    ClassNotFoundException {
        conn();
        String query = "update product set category=? where category=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setString(1, newCategory.getCategoryName());
        preparedStmt.setString(2, oldCategory.getCategoryName());
        preparedStmt.execute();
    }

    // update specific customer
    public static void updateCustomer(Customer customer) throws SQLException, ClassNotFoundException {
        conn();
        //set new name and telephone and email and address of customer where id of the customer
        String query = "UPDATE customer SET name=?,telephone=?,email=?,address=? WHERE id=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setString(1, customer.getFullName()); //full name
        preparedStmt.setString(2, customer.getTelephone()); //telephone
        preparedStmt.setString(3, customer.getEmail()); //email
        preparedStmt.setString(4, customer.getAddress()); //address
        preparedStmt.setInt(5, customer.getId()); //id of the customer
        preparedStmt.execute();
    }

    public static void updateProduct(Product product) throws SQLException, ClassNotFoundException {
        conn();
        String query = "update product set productName=?,category=?,price=?,quantity=?,supplierPrice=?,weight=? where
productId=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setString(1, product.getProductName());
        preparedStmt.setString(2, product.getCategory());
        preparedStmt.setDouble(3, product.getPrice());
        preparedStmt.setInt(4, product.getQuantity());
        preparedStmt.setDouble(5, product.getSupplierPrice());
        preparedStmt.setDouble(6, product.getWeight());
        preparedStmt.setInt(7, product.getProductCode());
        preparedStmt.execute();
    }

    //update specific shelf
    public static void updateShelf(Shelf shelf) throws SQLException, ClassNotFoundException {
        conn();
        String query = "update shelf set maxWeight=? where shelfNumber=?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setDouble(1, shelf.getMaxWeight()); //max weight
        preparedStmt.setInt(2, shelf.getShelfNumber()); //shelf number
        preparedStmt.execute();
    }

    //this function update the total price of the order
    public static void updateTotalPriceOfOrder(Order order, Double totalPrice) throws SQLException, ClassNotFoundException {
        conn();
        String query = "update orders set totalPrice = ? WHERE orderId = ?";
        preparedStmt = (PreparedStatement) conn.prepareStatement(query);
        preparedStmt.setDouble(1, totalPrice);
        preparedStmt.setInt(2, order.getOrderId());
        preparedStmt.execute();
    }

    //this function update category
    public static void editCategory(Category category) throws SQLException, ClassNotFoundException {
        conn();
        //update category name where category id equals to specific category id
        preparedStmt = (PreparedStatement) conn.prepareStatement("update category set categoryName=? where categoryId=?");
        preparedStmt.setString(1, category.getCategoryName());
        preparedStmt.setInt(2, category.getCategoryId());
        preparedStmt.execute();
    }

    public static void editEmployee(Employee employee) throws SQLException, ClassNotFoundException {
        conn();
        //update category name where category id equals to specific category id
    }

```

```

        preparedStmt = (PreparedStatement) conn.prepareStatement("update user set
salaryDay=?,generalSalary=?,updateSalaryDate=? where userName=?");
        preparedStmt.setDouble(1, employee.getSalaryDay());
        preparedStmt.setDouble(2, employee.getGeneralSalary());
        preparedStmt.setString(3, employee.getUpdateSalaryDate());
        preparedStmt.setString(4, employee.getUserName());
        preparedStmt.execute();
    }
    // this fuction edit expenses type (water or electrical)
    public static void editExpenses(Expenses exp) throws ClassNotFoundException, SQLException, ParseException {
        conn();
        DateClass date = new DateClass();
        preparedStmt = (PreparedStatement) conn.prepareStatement("update expenss set total=? where description=? AND
month=? AND year=?");
        preparedStmt.setDouble(1, exp.getTotal()); // new total of expenses type
        preparedStmt.setString(2, exp.getDescription()); // description (expenses type)
        preparedStmt.setInt(3, date.getMonth(exp.getExpDate())); // update the date to current date
        preparedStmt.setInt(4, date.getYear(exp.getExpDate()));
        preparedStmt.execute();
    }
    //this function edit the status of manager (true / false)
    public static void editManagerStatus(Manager manager) throws SQLException, ClassNotFoundException {
        conn();
        preparedStmt = (PreparedStatement) conn.prepareStatement("update manager set status = ?");
        preparedStmt.setBoolean(1, manager.getStatus());
        preparedStmt.execute();
    }
}

```

## קובץ Message.java

המחלקה הזאת אחראית על הצגת ההודעות.

```
package application;
import java.util.Optional;
import javafx.event.EventHandler;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.control.TextField;
import javafx.scene.control.TextInputDialog;
import javafx.scene.input.KeyEvent;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.ButtonBar.ButtonData;
import javafx.scene.image.Image;
import javafx.stage.Stage;

enum buttonType {
    YES,
    NO,
}
//class for messages
public class Message {
    // show message with OK button
    // the function gettingfunction_name(type of alert,content text of message,header text of message)
    public static void showMessageWithOKButton(AlertType type,String contentText,String headerText) {
        ButtonType okBtn = new ButtonType("Ok", ButtonData.OK_DONE);//button of message
        Alert alert = new Alert(type,contentText,okBtn);// alert getting (type of message,content text,button)
        alert.setTitle("Message");//title of message
        alert.setHeaderText(headerText);//header text of message
        Stage stage = (Stage)alert.getDialogPane().getScene().getWindow();//get the window of message
        stage.getIcons().add(new Image("inventoryIcon.png"));//icon
        stage.show();//show stage (show message)
    }

    public static buttonType showMessageWithYesNoButtons(AlertType type,String contentText,String headerText) {
        ButtonType yesBtn = new ButtonType("Yes", ButtonData.YES);//YES button
        ButtonType noBtn = new ButtonType("No", ButtonData.NO);//NO button
        Alert alert = new Alert(type,contentText,yesBtn,noBtn);// alert getting (type of message,content text,okButton,noButton)
        alert.setTitle("Message");//title of message
        alert.setHeaderText(headerText);//header text of message
        Stage stage = (Stage)alert.getDialogPane().getScene().getWindow();//get the window of message
        stage.getIcons().add(new Image("inventoryIcon.png"));//icon
        Optional<ButtonType> result = alert.showAndWait();//show the message and for action

        if(result.get() == yesBtn) {
            return buttonType.YES;//if the user clicked yes return yes
        }
        return buttonType.NO;//if the user clicked to button does not button yes return no
    }
}
//this function show message with text field
public static int showMessageWithTextField(String headerText, String contentText) throws InterruptedException {
    TextInputDialog dialog = new TextInputDialog();//show dialog with text field
    dialog.setTitle("Message");//title of message
    dialog.setHeaderText(headerText);//header text of message
    dialog.setContentText(contentText);//content text of message
    TextField messageField = dialog.getEditor();
    messageField.setStyle("-fx-background-color: lightblue;-fx-border-color: black;");
    Stage stage = (Stage)dialog.getDialogPane().getScene().getWindow();//get the window of message
    stage.getIcons().add(new Image("inventoryIcon.png"));//icon
    messageField.setOnKeyTyped(new EventHandler<KeyEvent>(){
        @Override
        public void handle(KeyEvent ke) {
            if(!Conditions.checkStringLength(messageField.getText(),6)// if length of field bigger than 6 remove the last index
                Conditions.removeLastIndexInString(messageField); // remove last index in field
        }
    });

    Optional<String> result = dialog.showAndWait();//show the message and for action

    // if message contains digits and a field its not empty
    if(Conditions.isContainsDigits(messageField.getText()) && !result.isEmpty() && messageField.getText().length() != 0)
        return Integer.parseInt(result.get());
    else if(!Conditions.isContainsDigits(messageField.getText())) // if the field in message not contains digits
        showMessageWithOKButton(AlertType.ERROR,"Please enter digits 0-9 only!", ""); // receive a message and
    return -1;
}

return -1;
}
```

## קובץ Field.java

המחלקה הזאת אחראית על ההרשאות בכל שדה.

```
package application;

import javafx.scene.control.Label;
import javafx.scene.control.TextField;

//emptyFields is interface
public class Field implements emptyFields{
    //checks empty fields
    @Override
    public Boolean checkEmptyFields(TextField field1,TextField field2) {
        // if all fields is not empty return true else return false
        if(!field1.getText().isEmpty() && !field2.getText().isEmpty())
            return true;

        return false;
    }

    // check empty field
    @Override
    public Boolean checkEmptyFields(TextField field1) {
        // if field is not empty return true else return false
        if(!field1.getText().isEmpty()) {
            return true;
        }

        return false;
    }

    //checks empty field
    @Override
    public Boolean checkEmptyFields(TextField field1, TextField field2, TextField field3, TextField
field4,TextField field5) {
        // if all fields is not empty return true else return false
        if(!field1.getText().isEmpty() && !field2.getText().isEmpty() &&
!field3.getText().isEmpty() && !field4.getText().isEmpty() &&
!field5.getText().isEmpty())
            return true;

        return false;
    }

    //checks empty fields
    @Override
    public Boolean checkEmptyFields(TextField field1,TextField field2,TextField field3,TextField
field4) {
        // if all fields is not empty return true else return false
        if(!field1.getText().isEmpty() && !field2.getText().isEmpty() &&
!field3.getText().isEmpty() && !field4.getText().isEmpty())
            return true;

        return false;
    }

    //checks empty fields
    @Override
    public Boolean checkEmptyFields(TextField field1, TextField field2, TextField field3, TextField
field4,
        TextField field5, TextField field6) {
        // if all fields is not empty return true else return false
        if(!field1.getText().isEmpty() && !field2.getText().isEmpty() &&
!field3.getText().isEmpty() && !field4.getText().isEmpty() &&
!field5.getText().isEmpty() && !field6.getText().isEmpty())
            return true;

        return false;
    }

    //this function for id field
    public Boolean checkIdField(TextField field,Label label,String color) {
        if(!Conditions.isContainsDigits(field.getText())) { // if id field content is not valid
            label.setText("Please enter digits 0-9 only!");
            changeBorderColor(field,"red"); // change border color to red
            return false;
        }else if(!Conditions.isIdPersonValid(field.getText())) { // if id field is not valid
            label.setText("I.D length must be 9!");
            changeBorderColor(field,"red"); // change border color to red
            return false;
        }else {
            changeBorderColor(field,color); // change border color
        }
    }
}
```



```

        label.setText("");
    }
    return true;
}
//this function for user name field
public Boolean checkUserNameField(TextField field,Label label) {
    if(!Conditions.isContainsUsrInField(field.getText())) { // if user field content is not valid
        changeBorderColor(field,"red");// change border color to red
        label.setText("*First three characters must be 'usr'");
        return false;
    }else {
        changeBorderColor(field,"blue");// change border color to blue
        label.setText("");
    }
    return true;
}
//this function for address field
public Boolean checkAddressField(TextField field,Label label) {
    if(!Conditions.isAddressValid(field.getText())) { // if address field content is not valid
        label.setText("*Address is invalid!");
        changeBorderColor(field,"red");// change border color to red
        return false;
    }else {
        changeBorderColor(field,"black");// change border color to black
    }
    return true;
}
// email field
public Boolean checkEmailField(TextField field,Label label,String color) {
    if(!Conditions.isEmailValid(field.getText())) { // if email field content is not valid
        label.setText("*Email is invalid!");
        changeBorderColor(field,"red");// change border color to red
        return false;
    }else {
        changeBorderColor(field,color); // change border color
        label.setText("");
    }
    return true;
}
//phone number field
public Boolean checkPhoneNumberField(TextField field,Label label) {
    if(!Conditions.isContainsDigitsWithLine(field.getText())) { // if phone number field content is not
valid
        changeBorderColor(field,"red");// change border color to red
        return false;
    }
    else if(Conditions.isPhoneNumberValid(field.getText()) ||
Conditions.isTelphoneNumberValid(field.getText())) {
        // if field content is phone number or telephone number
        changeBorderColor(field,"black");// change border color to black
        label.setText("");
        return true;
    }
    else {
        changeBorderColor(field,"red");// change border color to red
        label.setText("*Phone number is invalid!");
    }
    return false;
}
// first name or last name field
public Boolean checkNameField(TextField field,Label label) {
    if(!Conditions.isContainsCharacter(field.getText())) { //if first or last name field is not valid
        label.setText("*Please enter character a-z or A-Z");
        changeBorderColor(field,"red");// change border color to red
        return false;
    }else {
        label.setText("");
        changeBorderColor(field,"blue"); // change border color to blue
    }
    return true;
}
// number field
public Boolean checkNumberField(TextField field,Label label) {
    if(!Conditions.isContainsDigits(field.getText())) { // if the field content contains only
digits
        //if not contains digits
        label.setText("Please Enter digits 0-9 only!");
        changeBorderColor(field,"red");// change border color to red
        return false;
    }
}

```

```

        }else {
            changeBorderColor(field,"black");// change border color to black
            label.setText("");
        }
        return true;
    }
    //full name field
    public Boolean checkFullNameField(TextField field,Label label) {
        if(!Conditions.isFullNameValid(field)){// if full name field content not valid
            changeBorderColor(field,"red");// change border color to red
            label.setText("Please enter character a-z or A-Z or space");
            return false;
        }else {
            label.setText("");
            changeBorderColor(field,"black");// change border color to black
        }
        return true;
    }
    // password field
    public Boolean checkPasswordField(TextField field,Label label) {
        if(!Conditions.checkPasswordLength(field.getText(), 6)) { // if password field is not
valid
            changeBorderColor(field,"red");// change border color to red
            label.setText("*Enter password between 6-15 characters");
            return false;
        }else {
            changeBorderColor(field,"blue");// change border color to blue
            label.setText("");
        }
        return true;
    }
    // float number field
    public Boolean checkFloatNumberField(TextField field,Label label) {
        if(!Conditions.checksFloatNumber(field)){ // if float number field is not valid
            changeBorderColor(field,"red");// change border color to red
            label.setText("*Please enter number or float number");
            return false;
        }else {
            label.setText("");
            changeBorderColor(field,"black");// change border color to black
        }
        return true;
    }
    //change color field
    public void changeBorderColor(TextField field , String color) {
        field.setStyle(field.getStyle() + "-fx-border-color: " + color + ";" );
    }
    //change color field
    public void changeBorderColor(TextField field1, TextField field2, TextField field3, TextField
field4,
        TextField field5) {
        TextField[] fields = new TextField[5];
        fields[0] = field1;
        fields[1] = field2;
        fields[2] = field3;
        fields[3] = field4;
        fields[4] = field5;

        for(int i = 0;i < fields.length;i++) {
            if(fields[i].getText().isEmpty()) // if field is empty
                fields[i].setStyle(fields[i].getStyle() + "-fx-border-color:red;");//
change border color to red
            else // if field is not empty
                fields[i].setStyle(fields[i].getStyle() + "-fx-border-color:black;");//
change border color to black
        }
    }
}

```

## קובץ emptyFiels.java

זה ממשיך מכיל כל הפונקציות (האם השדות ריקות)

```
package application;

import javafx.scene.control.TextField;

//empty fields interface
public interface emptyFields {
    // check two fields is empty
    public Boolean checkEmptyFields(TextField field1,TextField field2);
    // check one field is empty
    public Boolean checkEmptyFields(TextField field1);
    // check five fields is empty
    public Boolean checkEmptyFields(TextField field1,TextField field2,TextField field3,TextField
field4,TextField field5);
    // check six fields is empty
    public Boolean checkEmptyFields(TextField field1,TextField field2,TextField field3,TextField
field4,TextField field5,TextField field6);
    // check four fields is empty
    public Boolean checkEmptyFields(TextField field1,TextField field2,TextField field3,TextField
field4);
}
```

## קובץ DateClass.java

המחלקה הזאת אחראית על ניהול התאריכים.

```
package application;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class DateClass {

    Calendar c;
    SimpleDateFormat simpleDateFormat;
    Date date;

    public DateClass() {
        simpleDateFormat = new SimpleDateFormat("dd/MM/yyyy");
        this.c = Calendar.getInstance();
    }

    // this function return the current date in string
    public String getDateToday() {
        return String.valueOf(c.get(Calendar.DAY_OF_MONTH) + "/" + (c.get(Calendar.MONTH) + 1) +
"/" + c.get(Calendar.YEAR));
    }
    // this function return the current year
    @SuppressWarnings("deprecation")
    public int getCurrentYear() {
        return Calendar.getInstance().getTime().getYear();
    }
    // this function return the current month
    @SuppressWarnings("deprecation")
    public int getCurrentMonth() {
        return Calendar.getInstance().getTime().getMonth();
    }

    @SuppressWarnings("deprecation")
    public int getMonth(String orderDate) throws ParseException {
        Date date = simpleDateFormat.parse(orderDate);
        return date.getMonth();
    }
    @SuppressWarnings("deprecation")
    public int getYear(String orderDate) throws ParseException {
        Date date = simpleDateFormat.parse(orderDate);
        return date.getYear();
    }
    // this function getting number and return the month in string
    public String getMonth(int month){
        String[] monthNames = {"January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"};
        return monthNames[month];
    }
}
```

## קובץ Person.java

המחלקה הזאת מגדירה תכונות אדם.

```
package application;

public class Person {

    protected String firstName;
    protected String lastName;
    protected String email;
    protected int id;

    public Person() {}

    public Person(String firstName,String lastName,String email,int id) {
        this.setFirstName(firstName);
        this.setLastName(lastName);
        this.setEmail(email);
        this.setId(id);
    }

    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

}
```

## קובץ User.java

המחלקה הזאת יורשת ממחלקת אדם כמה תכונות ובנוסף יש הגדרת תכונות למשתמש.

```
package application;
import javafx.collections.ObservableList;

public class User extends Person{

    protected String userName;
    protected String password;

    public User() {
        super();
    }
    public User(String userName) {
        this.setUserName(userName);
    }

    public User(String firstName, String lastName, String email, int id,String userName,String
password) {
        super(firstName, lastName, email, id);
        this.setUserName(userName);
        this.setPassword(password);

    }

    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    //function check if user name is Taken
    public boolean userNameIsTaken(ObservableList<User> userList,User user) {
        for(User u : userList) { //loop for user list
            if(u.getUserName().equals(user.getUserName())) { //if user name is taken return
false
                return false;
            }
        }
        // if user name not taken return true
        return true;
    }
    //function check if user name is exist
    public User checkUserName(ObservableList <User> userList,String userName) {
        for(User u : userList) { //loop for user list
            if(u.getUserName().equals(userName)) { //if user name is exist return user
                return u;
            }
        }
        // if user name not exist return null
        return null;
    }
    //this function check if password belongs to user
    public boolean checkPassword(User user,String Password) {
        if(user.getPassword().equals(Password)) { //if password of user equal to equal to the
password entered by the user
            return true;
        }
        //if password not equal return false
        return false;
    }
}
```

## קובץ Manager.java

המחלקה הזאת יורשת ממחלקת אדם כמה תכונות ובנוסף יש הגדרת תכונות למנהל.

```
package application;

public class Manager extends Person{

    private String userName;
    private String password;
    private Boolean status;

    public Manager(String firstName, String lastName, String email, int id,String userName,String
password,Boolean status) {
        super(firstName, lastName, email, id);
        this.setUserName(userName);
        this.setPassword(password);
        this.setStatus(status);
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public Boolean getStatus() {
        return status;
    }

    public void setStatus(Boolean status) {
        this.status = status;
    }

    // this function checks if the password of the manager is correct
    public boolean checkPassword(Manager m,String password) {
        if(m.getPassword().equals(password)) // if password of the manager is correct return true
            return true;
        // if password of the manager is wrong return false
        return false;
    }
}
```

## קובץ Customer.java

מחלקת לקוח.

```
package application;

import javafx.collections.ObservableList;

public class Customer implements Comparable<Customer>{

    private String fullName;
    private String address;
    private String email;
    private String telephone;
    private int id;
    private int numberOfOrders;
    ObservableList<Order> orders;

    public Customer(String fullName,int id,String email,String telephone,String address) {
        this.setFullName(fullName);
        this.setAddress(address);
        this.setEmail(email);
        this.setTelephone(telephone);
        this.setId(id);
    }
    public Customer(String fullName, int id) {
        this.setFullName(fullName);
        this.setId(id);
    }
    public Customer(int id,String fullName,String telephone) {
        this.setFullName(fullName);
        this.setTelephone(telephone);
        this.setId(id);
        this.numberOfOrders = 0;
    }
    public void setOrdersList(ObservableList<Order> orders) {
        this.orders = orders;
    }
    public String getFullName() {
        return fullName;
    }

    public void setFullName(String fullName) {
        this.fullName = fullName;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getTelephone() {
        return telephone;
    }

    public void setTelephone(String telephone) {
        this.telephone = telephone;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```



```

    }

    public int getNumberOfOrders() {
        return numberOfOrders;
    }

    public void setNumberOfOrders(int numberOfOrders) {
        this.numberOfOrders = numberOfOrders;
    }

    // this function calculate orders number of the customer
    public void calculateOrdersNumber(Customer customer){
        for(Order order : this.orders) {
            if(order.getCustomerId() == customer.getId())
                this.numberOfOrders++;
        }
    }

    //this function checks if the customer exists
    public Boolean customerIdIsTaken(Customer customer,ObservableList<Customer> customerList) {
        for(Customer c : customerList)
            if(c.getId() == customer.getId())//if the customer exists return false
                return false;
            //if the customer not exists return true
        return true;
    }

    @Override
    public int compareTo(Customer customer) {
        if(customer.numberOfOrders != this.numberOfOrders) // sort according to order number of
            customer
            return customer.numberOfOrders - this.numberOfOrders;
        else if (customer.fullName.charAt(0) != this.fullName.charAt(0)) // sort according to the
            first character
            return customer.getFullName().charAt(0) - this.getFullName().charAt(0);
        else
            return customer.id - this.id; // sort according to the id
    }

}

}

```

## קובץ Category.java

מחלקת קטגוריה.

```
package application;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

public class Category implements Comparable<Category>{

    private String categoryName;
    private int categoryId;
    private int productNumber;

    ObservableList<Product> productsList;

    public Category(int categoryId,String categoryName) {
        this.setCategoryName(categoryName);
        this.setCategoryId(categoryId);
        this.productNumber = 0;
        productsList = FXCollections.observableArrayList();
    }
    public Category(String categoryName) {
        this.setCategoryName(categoryName);
    }

    public String getCategoryName() {
        return categoryName;
    }

    public void setCategoryName(String categoryName) {
        this.categoryName = categoryName;
    }

    public int getCategoryId() {
        return categoryId;
    }

    public void setCategoryId(int categoryId) {
        this.categoryId = categoryId;
    }

    public void setProductList(ObservableList<Product> productsList) {
        this.productsList = productsList;
    }

    public int getProductNumber() {
        return productNumber;
    }

    public void setProductNumber(int productNumber) {
        this.productNumber = productNumber;
    }

    //this function checks if category name is taken
    public Boolean categoryNameIsTaken(Category category,ObservableList<Category> categoryList) {
        for(Category c : categoryList)
            if(c.getCategoryName().equals(category.getCategoryName())) // if category name exists
                return false;

        return true;
    }

    //this function checks if the category contains product
    public Boolean checksCategory(Category category) {
        for(Product product:productsList) {
            if(product.getCategory().equals(category.getCategoryName()))
                return true;
        }
        return false;
    }

    //this function return number of products in the category
    public int getNumberOfProduct(Category category) {
        int quantity = 0;
        for(Product product : this.productsList) {
            if(product.getCategory().equals(category.getCategoryName()))
                quantity++;
        }
        return quantity;
    }

    //this function return quantity of all products in the storage
```

```

public int getQuantityOfAllProducts() {
    int quantity = 0;
    for(Product product : this.productsList)
        quantity += product.getQuantity();
    return quantity;
}
// this function returns quantity of products in spesfic category
public int getQuantityInCategory() {
    int quantity = 0;
    for(Product product : this.productsList)
        if(product.getCategory().equals(getCategoryName()))
            quantity += product.getQuantity();

    return quantity;
}

@Override
public int compareTo(Category category) {
    if(category.productNumber!= this.productNumber) //sorts by the products number sold
        return category.productNumber - this.productNumber;
    else if(category.categoryName.charAt(0) != this.categoryName.charAt(0))// sort by the first index in the string
        return this.categoryName.charAt(0) - category.categoryName.charAt(0);
    return category.categoryId - this.categoryId ;//if products number is equal then sorts by the category id
}

}

```

## קובץ Product.java

מחלקת מוצר.

```
package application;
import java.sql.SQLException;
import java.util.Collections;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

public class Product implements Comparable<Product>{
    private int productCode;
    private String productName;
    private String category;
    private double price;
    private Double weight;
    private int shelfNumber;
    private int quantity;
    private int orderId;
    private int numberSold;
    private double supplierPrice;
    ObservableList<Order> ordersList;
    ObservableList<Shelf> shelvesList;
    Order order;

    public Product() {}
    public Product(int productCode,String productName,String category,Double supplierPrice,Double price,int quantity,Double weight) {
        this.setProductCode(productCode);
        this.setProductName(productName);
        this.setCategory(category);
        this.setPrice(price);
        this.setQuantity(quantity);
        this.setSupplierPrice(supplierPrice);
        this.setWeight(weight);
        this.setShelfNumber(0);
    }
    public Product(int productCode,String productName,String category,Double price,int quantity) {
        this.setProductCode(productCode);
        this.setProductName(productName);
        this.setCategory(category);
        this.setPrice(price);
        this.setQuantity(quantity);
    }
    public Product(int orderId,int productId,String productName,double price,int quantity,Double weight,Double supplierPrice,String
categoryName) {
        this.setOrderId(orderId);
        this.setProductName(productName);
        this.setPrice(price);
        this.setQuantity(quantity);
        this.setProductCode(productId);
        this.setShelfNumber(shelfNumber);
        this.setCategory(categoryName);
        this.setWeight(weight);
        this.setSupplierPrice(supplierPrice);
    }
    public Product(int productCode,String productName,int quantity,Double price) {
        this.setProductCode(productCode);
        this.setProductName(productName);
        this.setQuantity(quantity);
        this.setPrice(price);
        this.setNumberSold(0);
        this.ordersList = FXCollections.observableArrayList();
    }
    public Product(int productCode,String productName,int quantity,Double price,Double supplierPrice) {
        this.setProductCode(productCode);
        this.setProductName(productName);
        this.setQuantity(quantity);
        this.setPrice(price);
        this.setSupplierPrice(supplierPrice);
    }
    public Product(int productCode,String productName,int numberSold) {
        this.setProductCode(productCode);
        this.setProductName(productName);
        this.setNumberSold(numberSold);
    }
    public Product(int shelfNumber,int productId,String productName,int quantity,Double weight,String category,Double price,Double
supplierPrice) {
        this.setShelfNumber(shelfNumber);
        this.setProductCode(productId);
    }
}
```

```

        this.setProductName(productName);
        this.setQuantity(quantity);
        this.setWeight(weight);
        this.setCategory(category);
        this.setPrice(price);
        this.setSupplierPrice(supplierPrice);
    }
    public Product(int productId,String productName,int quantity,Double price,Double weight,Double supplierPrice,String
categoryName) {
        this.setShelfNumber(shelfNumber);
        this.setProductCode(productId);
        this.setProductName(productName);
        this.setQuantity(quantity);
        this.setPrice(price);
        this.setSupplierPrice(supplierPrice);
        this.setWeight(weight);
        this.setCategory(categoryName);
    }
    public void setOrder(Order order) {
        this.order = order;
    }
    //set order list
    public void setOrdersList(ObservableList<Order> ordersList) {
        this.ordersList = ordersList;
    }

    // set shelves list
    public void setShelvesList(ObservableList<Shelf> shelvesList) {
        this.shelvesList = shelvesList;
    }

    // get shelves list
    public ObservableList<Shelf> getShelvesList(){
        return this.shelvesList;
    }
    //set product name
    public String getProductName() {
        return productName;
    }
    //set product name
    public void setProductName(String productName) {
        this.productName = productName;
    }
    //get category name
    public String getCategory() {
        return category;
    }
    //set category name
    public void setCategory(String category) {
        this.category = category;
    }
    //get client price
    public Double getPrice() {
        return price;
    }
    // set client price
    public void setPrice(Double price) {
        this.price = price;
    }
    //get quantity of product
    public int getQuantity() {
        return quantity;
    }
    //set quantity of product
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
    //get product code
    public int getProductCode() {
        return productCode;
    }
    // set product code
    public void setProductCode(int productCode) {
        this.productCode = productCode;
    }
    //get order id
    public int getOrderId() {
        return orderId;
    }
    //set order id

```

```

public void setOrderId(int orderId) {
    this.orderId = orderId;
}
// get number sold of product
public int getNumberSold() {
    return numberSold;
}
// set number sold of product
public void setNumberSold(int numberSold) {
    this.numberSold = numberSold;
}
// get supplier price of product
public double getSupplierPrice() {
    return supplierPrice;
}
// set supplier price of product
public void setSupplierPrice(double supplierPrice) {
    this.supplierPrice = supplierPrice;
}
// get weight of product
public Double getWeight() {
    return weight;
}
//set weight of product
public void setWeight(Double weight) {
    this.weight = weight;
}
//get shelf number
public int getShelfNumber() {
    return shelfNumber;
}
//set shelf number
public void setShelfNumber(int shelfNumber) {
    this.shelfNumber = shelfNumber;
}
public Order getOrder() {
    return this.order;
}

// get the current order by the order code
public Order getCurrentOrder(int orderCode){
    for(Order order : this.ordersList)
        if(order.getOrderId() == orderCode)
            return order;

    return null;
}

// this function calculate revenue
public Double calculateRevenue() {
    Double revenue = 0.0;
    for(Order order : this.ordersList)
        revenue += order.getTotalPrice();

    return revenue;
}
//this function checks if the supplier price bigger than client price
public Boolean checkSupplierPrice(Product product){
    if(product.getSupplierPrice() > product.getPrice())
        return false;

    return true;
}

//this function checks if the product name exists in the system
public Boolean productNamesTaken(Product product,ObservableList<Product> productList) {
    for(Product p:productList) {
        if(p.getProductName().equals(product.getProductName()))//if product name exists return false
            return false;
    }
    //if the product name not exists in the system
    return true;
}

// this function checks the code of product with the code of product list in storage and return the current product
public Product getCurrentProduct(ObservableList<Product> productsList) {
    for(Product product : productsList)
        if(getProductCode() == product.getProductCode())
            return product;

    return null;
}

```

```

//codeProductIsTaken checks if product is exists in inventory
public Boolean checkProductIsExists(ObservableList<Product> productList) {
    for(Product p : productList)
        if(p.getProductCode() == getProductCode())
            return true;

    return false;
}

//quantityCheck function getting function_name(the quantity of products the user entered,the quantity of product in inventory)
public Boolean checkProduct(int quantity,int productQuantity) {
    if(quantity > productQuantity)
        return false;

    return true;
}

//isProductQuantityZero checks if the quantity of the product equal to zero
public Boolean isProductQuantityZero(int quantity) {
    if(quantity == 0)
        return false;

    return true;
}

//this function return top 3 products sold
public ObservableList<Product> getTop3Products(ObservableList<Product> list){
    ObservableList<Product> products = FXCollections.observableArrayList();
    Collections.sort(list);//sort
    for(int i = 0;i < list.size() && i < 3;i++) //this loop add product to list
        products.add(list.get(i));

    return products;
}

// this function return the shelf by the shelf number
public Shelf getShelf(int shelfNumber) {
    for(Shelf sh : this.shelvesList)
        if(sh.getShelfNumber() == shelfNumber)
            return sh;

    return null;
}

// this function get number of product can insert the product to the shelves
public int getVacantShelf(Product []products) throws ClassNotFoundException, SQLException {
    int quantity = 0;
    Double diff = 0.0;
    for(int i = 0; i < products.length ; i++) {
        products[i].setShelvesList(this.shelvesList);
        Shelf shelf = products[i].getShelf(products[i].getShelfNumber());
        shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));
        diff = shelf.getWeightDifference();
        quantity += diff / getWeight();
    }
    return quantity;
}

//this function return quantity of all products in the storage
public int getQuantityOfAllProducts(ObservableList<Product> productList) {
    int quantity = 0;
    for(Product product : productList)
        quantity += product.getQuantity();

    return quantity;
}

// this function checks if the product is exists in any shelf
public Boolean isProductExistsInShelves() throws ClassNotFoundException, SQLException {
    for(Shelf shelf : this.shelvesList) {
        shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));
        if(shelf.checkProductIsExists(this) > -1)
            return true;
    }
    return false;
}

@Override
public int compareTo(Product product) {
    if(product.numberSold!= this.numberSold) //sorts by the number of products sold
        return product.numberSold - this.numberSold;
    return product.productCode - this.productCode ;//if number sold is equal then sorts by the product code
}
}

```

## Shelf.java קובץ

מחלקת מדף.

```
package application;

import java.sql.SQLException;
import java.util.Arrays;

import javafx.collections.ObservableList;

public class Shelf {

    private int shelfNumber;
    private Double maxWeight;
    ObservableList<Product> productsList;

    public Shelf() {}

    public Shelf(int shelfNumber, Double maxWeight) {
        this.setMaxWeight(maxWeight);
        this.setShelfNumber(shelfNumber);
    }

    public void setProductsList(ObservableList<Product> productsList) {
        this.productsList = productsList;
    }

    public int getShelfNumber() {
        return shelfNumber;
    }

    public void setShelfNumber(int shelfNumber) {
        this.shelfNumber = shelfNumber;
    }

    public Double getMaxWeight() {
        return maxWeight;
    }

    public void setMaxWeight(Double maxWeight) {
        this.maxWeight = maxWeight;
    }

    // this function return all products in the shelves of the received product
    public Product[] getProductInShelves(Product product){

        Product []products = new Product[0];

        for(Product p : this.productsList) {
            if(p.getProductCode() == product.getProductCode()) { // if product existing
                products = Arrays.copyOf(products, products.length + 1);
                products[products.length-1] = p; // add product to array
            }
        }
        // if product not existing in shelves return null
        if(products.length == 0)
            return null;

        return products;
    }

    //this function check if the product is exists if exists return quantity of the product else return -1 (not exists)
    public int checkProductsExists(Product product) {
        for(Product p : this.productsList) {
            if(product.getProductCode() == p.getProductCode())//if product code equal product code in the shelf
                return p.getQuantity();// if true return quantity of the product in the shelf
        }
        return -1;
    }

    //this function return the weight of the shelf
    public Double getShelfWeight() {
        Double weight = 0.0;
        for(Product product : this.productsList) {
            if(product.getShelfNumber() == getShelfNumber())
                weight += product.getQuantity() * product.getWeight();
        }
        return weight;
    }
}
```



```

//this function return all quantity of products in the shelves
public int getProductQuantity(Product []products) {
    int qunatity = 0;

    for(int i = 0 ;i < products.length ; i++)
        qunatity += products[i].getQuantity();
    return qunatity;
}

//returns how many products can be put on the shelf
public int getProductQunatity(Product product) {
    return (int) ((getMaxWeight() - getShelfWeight()) / product.getWeight());
}

// get the weight difference between max weight to products weight in the shelf
public Double getWeightDifference() {
    return getMaxWeight() - getShelfWeight();
}

shelf // this function checks the weight of the quantity of the product and the weight the products in the shelf if less than max weight of the
public Boolean checkWeight(Product product,int quantity) {
    if(product.getWeight() * quantity + getShelfWeight() <= getMaxWeight())
        return true;
    return false;
}

// this function returns quantity of specifc product in the shelves
public int getProductQuantity(Product p) {
    int number = 0;
    for(Product product : this.productsList) {
        if(p.getProductCode() == product.getProductCode())
            number += product.getQuantity();
    }
    return number;
}

//this function checks if the shelf number is taken
public Boolean checkShelfIsExists(ObservableList<Shelf> shelfList) {
    for(Shelf s:shelfList)
        if(getShelfNumber() == s.getShelfNumber()) //if shelf number equals to new shelf return false
            return false;

    //if the shelf number not exists return true
    return true;
}
}

```

## קובץ Order.java

מחלקת מדף.

```
package application;

import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

public class Order implements Comparable<Order>{

    private String orderDate;
    private int customerId;
    private Double totalPrice;
    private int orderId;
    ObservableList<Product> productsList;

    public Order() {
        this.totalPrice = (double) 0;
        this.productsList = FXCollections.observableArrayList();
    }
    public Order(int customerId,Double totalPrice,String orderDate) {
        this.setCustomerId(customerId);
        this.setOrderDate(orderDate);
        this.setTotalPrice(totalPrice);
    }
    public Order(int orderId,int customerId,Double totalPrice,String orderDate) {
        this.setOrderId(orderId);
        this.setCustomerId(customerId);
        this.setOrderDate(orderDate);
        this.setTotalPrice(totalPrice);
    }
    public Order(int orderId) {
        this.setOrderId(orderId);
    }
    public void setProductsList(ObservableList<Product> productsList) {
        this.productsList = productsList;
    }
    public String getOrderDate() {
        return orderDate;
    }
    public void setOrderDate(String orderDate2) {
        this.orderDate = orderDate2;
    }
    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public Double getTotalPrice() {
        return totalPrice;
    }
    public void setTotalPrice(Double totalPrice) {
        this.totalPrice = totalPrice;
    }
    public int getOrderId() {
        return orderId;
    }
    public void setOrderId(int orderId) {
        this.orderId = orderId;
    }
    // this function return the specific customer by the id of the customer
    public Customer getCurrentCustomer(ObservableList<Customer> customers,int id) {
        for(Customer customer : customers)
            if(customer.getId() == id)
                return customer;
    }

    return null;
}
```

```

//this function return products sold
public int calculateNumberSoldOfProduct(Product product) throws ParseException, ClassNotFoundException, SQLException {
    int numberSold = 0;

    DateClass date = new DateClass();

    for(Order order : DatabaseManager.getOrders()) {
        String orderDate = order.getOrderDate();
        if(date.getMonth(orderDate) == date.getCurrentMonth() && date.getYear(orderDate) == date.getCurrentYear()) {
            for (Product p : DatabaseManager.getProductOfOrder(order)) {
                if(product.getProductCode() == p.getProductCode())
                    numberSold += p.getQuantity();
            }
        }
    }

    return numberSold;
}

//this function checks if the product is exists in order
public Boolean isProductExists(Product p) {
    for(Product product : this.productsList)
        if(p.getProductCode() == product.getProductCode()) // if exists return true
            return true;
    return false; // if not exists return false
}

// this function return the specific product by the id of the product
public Product getCurrentProduct(Product product) {
    for(Product p : this.productsList)
        if(p.getProductCode() == product.getProductCode())
            return p;
    return null;
}

// this function return total price of the order
public Double getTotalPrice(ObservableList<Product> productsList) {
    Double totalPrice = (double) 0;
    for(Product product:productsList)
        totalPrice += (product.getQuantity()) * product.getPrice();
    return totalPrice;
}

@Override
public int compareTo(Order order) {
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("dd/MM/yyyy");

    try {
        Date date1 = simpleDateFormat.parse(order.getOrderDate());
        Date date2 = simpleDateFormat.parse(this.getOrderDate());

        return date1.compareTo(date2); // sort by date
    } catch (ParseException e) {
        e.printStackTrace();
    }
    return 0;
}

```

## קובץ Water.java

מחלקת הוצאות מים.

```
package application;

public class Water extends Expenses{

    protected String description;

    public Water() {
        super();
    }

    public Water(Double total,String expDate) {
        super(total,expDate);
        this.setDescription("Water");
    }

    public Water(Double total) {
        super(total);
        this.setDescription("Water");
    }

}
```

## קובץ Electrical.java

מחלקת הוצאות מים.

```
package application;

public class Electrical extends Expenses{

    protected String description;

    public Electrical(Double total,String expDate) {
        super(total,expDate);
        this.setDescription("Electrical");
    }

    public Electrical(Double total) {
        super(total);
        this.setDescription("Electrical");
    }

}
```

```
package application;

import java.sql.SQLException;

import javafx.application.Platform;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;

public class Conditions{

    //this function checks user field string
    public static Boolean isContainsUsrInField(String str) {
        if(str.length() > 3) { //checks if the string length bigger than 3 if not return false
            //check if the string of user contains 'usr' first three index
            if(str.charAt(0) == 'u' && str.charAt(1) == 's' && str.charAt(2) == 'r')
                return true;
            else
                return false;
        }
        return false;
    }

    // this function checks the minimum length of the password field
    public static Boolean checkPasswordLength(String str,int length) {
        if(str.length() < length) // if the string length less than length return false
            return false;
        // if the string length bigger than length return true
        return true;
    }

    //this function check the length of the string
    public static Boolean checkStringLength(String str,int length) {
        if(str.length() > length) //if the length of string bigger than length return false
            return false;
        // if the length of string less than length return true
        return true;
    }

    // check another time
    public static Boolean isUserNameFieldContainsDigits(String str) {
        for(int i = 3;i<str.length();i++) {
            if(str.charAt(i) > '0' && str.charAt(i) < '9')
                return true;
        }
        return false;
    }

    // this function checks if the string contains digits
    public static Boolean isContainsDigits(String str) {
        for(int i = 0;i < str.length();i++)
            if(str.charAt(i) < '0' || str.charAt(i) > '9') // if string[index] not digit return false
                return false;
        //if the string contains only digits return true
        return true;
    }

    //this function checks is string contains digits or '-'
    public static Boolean isContainsDigitsWithLine(String str) {
        for(int i = 0;i < str.length();i++)
            if((str.charAt(i) < '0' || str.charAt(i) > '9') && str.charAt(i) != '-') // if the string[index] not digit and not '-' return
                return false;
        //if the string contains '-' and digits return true
        return true;
    }

    //this function checks if the string contains numbers or point
    public static Boolean checksFloatNumber(TextField field) {
        int numberOfPoints = 0;
        String str = field.getText();

        if(str.charAt(0) == '.' || str.charAt(field.getText().length()-1) == '.')
            return false;
    }
}
```

```

for(int i = 0; i < str.length(); i++) {
    if(!((str.charAt(i) >= '0' && str.charAt(i) <= '9') || str.charAt(i) == '.')) {
        return false;
    }
    else {
        if(str.charAt(i) == '.')
            if(++numberOfPoints > 1) //if number of points bigger than 1 return false
                return false;
    }
}
return true;
}

// this function return the float number with 2 digits after the points
public static Double get2DigitsAfterPoint(String str) {
    int pointIndex = 0;
    if(str.contains(".")) { // if the number contains point
        if(str.charAt(str.length()-1) == '.') //if the last index in the float number return the number with zero in last index
            return Double.parseDouble(str + "0");

        pointIndex = str.indexOf('.');
    }
    else //if the number not contains point
        return Double.parseDouble(str);

    String strAfterPoint = str.substring(pointIndex + 1, str.length());
    //calculate the number of digits after the point
    if(strAfterPoint.length() <= 2) // if number of digits less equal than 2 return the number
        return Double.parseDouble(str);

    String newStr = str.substring(0, pointIndex);

    //if the number of digits after the point bigger than 2 return the number with two digits
    for(int i = pointIndex; i <= pointIndex + 2; i++) {
        newStr += str.charAt(i);
    }

    return Double.parseDouble(newStr);
}

//this function checks is the email is valid
public static Boolean isValidEmail(String str) {
    if(!str.contains("@"))
        return false;

    String []domainName = {"@gmail.com", "@walla.com"};

    int atSignIndex = str.indexOf("@"); //get the index of the sign
    int numberOfSign = 0; // number of @ in the string

    //check if the sign index equal zero return false
    if(atSignIndex == 0)
        return false;

    for(int i=0; i<str.length(); i++)
        if(str.charAt(i) == '@') //if the string[index] equal zero ,increment numberOfSign
            numberOfSign++;

    if(numberOfSign > 1) // if number of sign bigger than 1 return false
        return false;

    for(int i = 0; i < domainName.length; i++)
        //Take the string from the cursor to the end of the string and equal to the domain name
        if(str.substring(atSignIndex, (str.length()-1)).equals(domainName[i])) //if the end of the email string valid return
            return true;

    // if the email string is not valid return false
    return false;
}

//this function checks if the string contains character a-z or A-Z only.
public static Boolean isContainsCharacter(String str) {
    for(int i = 0; i < str.length(); i++) {
        //if the string[index] equal character not a-z or A-Z return false
        if((str.charAt(i) < 'a' || str.charAt(i) > 'z') && (str.charAt(i) < 'A' || str.charAt(i) > 'Z'))
            return false;
    }
}

```

```

    }
    // if the string contains a-z or A-Z return true
    return true;
}
// this function check if full name is valid
public static Boolean isFullNameValid(TextField field) {
    String str = field.getText();
    char space = ' ';

    // id string[0] equal space return false
    if((str.charAt(0) == space))
        return false;

    for(int i = 0; i < str.length(); i++) {
        // can be entered from A-Z or a-z and space if not return false
        if((str.charAt(i) < 'a' || str.charAt(i) > 'z') && (str.charAt(i) < 'A' || str.charAt(i) > 'Z') && str.charAt(i) != ' ')
            return false;
    }

    return true;
}
//check area code of telephone or phone number
private static Boolean isAreaCodeValid(String string, String []str) {
    for(int i = 0; i < str.length; i++) {
        if(string.equals(str[i]))
            return true;
    }
    return false;
}
// this function checks does the telephone number is valid
public static Boolean isPhoneNumberValid(String str) {
    char line = '-';

    String []localFormat = {"04", "03", "07"};

    if(str.length() != 11 && str.length() != 9)
        return false;
    // if the telephone string equal to 9
    if(str.length() == 9) {
        // check area code is valid
        if(isAreaCodeValid(str.substring(0,2), localFormat))
            return isContainsDigits(str.substring(2, str.length()-1)); // check if index between 2 to 9 contains
        return false;
    }
    // if str length bigger than 9
    // check if the index 2 and 6 in the string equal to line
    if(str.charAt(2) != line || str.charAt(6) != line)
        return false; // if not equal return false

    // check if area code is valid
    if(isAreaCodeValid(str.substring(0,2), localFormat)) {
        // if index 3 - 5 and 7 to last index in the string is contains digits
        if(isContainsDigits(str.substring(3, 5)) && isContainsDigits(str.substring(7, (str.length()-1))))
            return true;
    }
    return false;
}

// check if phone number is valid

public static Boolean isPhoneNumberValid(String str) {
    char line = '-';
    String []localFormat = {"054", "053", "052", "050", "057"};

    //if str length not equal to 12 or 10 return false
    if(str.length() != 12 && str.length() != 10)
        return false;

    // if str length equal to 10
    if(str.length() == 10) {
        // checks if area code is valid
        if(isAreaCodeValid(str.substring(0,3), localFormat) == true)
            return isContainsDigits(str.substring(3, str.length()-1));
        else
            return false;
    }
    // if index 3 or index 7 not equal to line return false
    if(str.charAt(3) != line || str.charAt(7) != line)
        return false;
}

```

```

        // if area code is valid
        if(isAreaCodeValid(str.substring(0,3),localFormat) == true) {
            if(isContainsDigits(str.substring(4, 6)) == true && isContainsDigits(str.substring(8, (str.length()-1))) == true)
                return true;
        }
        return false;
    }

    // this function checks address is valid
    public static Boolean isAddressValid(String str) {
        for(int i = 0;i<str.length();i++) {
            if((str.charAt(i) < 'a' || str.charAt(i) > 'z') && (str.charAt(i) < 'A' || str.charAt(i) > 'Z') && str.charAt(i) != '-' &&
str.charAt(i) != '-')
                return false;
        }
        return true;
    }

    //this function check id person is valid
    public static Boolean isIdPersonValid(String str) {
        //if string length equals 9 return true
        if(str.length() == 9)
            return true;
        return false;
    }

    //this fuction check status of manager
    public static void checkManagerStatus(Button button) throws ClassNotFoundException, SQLException {
        Manager manager = DatabaseManager.getManager();
        if(manager.getStatus().equals(true)) { // if status of manager equal to true
            button.setVisible(true); // button visable true
            button.setManaged(true);
        } else { // if status of manager equal to false
            button.setVisible(false); // button visable false
            button.setManaged(false);
        }
    }

    //Genetics function
    private static <T> Boolean checkStringInSearch(T object,String str) {
        // if an object was received from a product type
        if(object instanceof Product) {
            // if the product name or product code contains the string of the search field
            if(((Product) object).getProductName().contains(str) == true || String.valueOf(((Product)
object).getProductCode()).contains(str) == true)
                return true;
        } // if an object was received from a category type
        else if(object instanceof Category) {
            // if the category name or category id contains the string of the search field
            if(((Category) object).getCategoryName().contains(str) == true || String.valueOf(((Category)
object).getCategoryId()).contains(str) == true)
                return true;
        } // if an object was received from a customer type
        else if(object instanceof Customer) {
            // if the full name or id contains the string of the search field
            if(((Customer) object).getFullName().contains(str) == true || String.valueOf(((Customer)
object).getId()).contains(str) == true)
                return true;
        } // if an object was received from a shelf type
        else if(object instanceof Shelf) {
            // if the shelf number contains the string of the search field
            if(String.valueOf(((Shelf) object).getShelfNumber()).contains(str) == true )
                return true;
        } // if an object was received from a order type
        else if(object instanceof Order) {
            // if the customer id or order id contains the string of the search field
            if(String.valueOf(((Order) object).getCustomerId()).contains(str) || String.valueOf(((Order)
object).getOrderId()).contains(str))
                return true;
        } else if(object instanceof Employee) {
            if(((Employee)object).getUserName().contains(str) || ((Employee)object).getFullName().contains(str))
                return true;
        }
        return false;
    }

    //this function return items in search
    public static <T> ObservableList<T> getItemsSearch(ObservableList<T> list,String str) throws ClassNotFoundException{
        ObservableList<T> newList = FXCollections.observableArrayList();
    }

```



```

        for(T item : list)
            if(checkStringInSearch(item, str) == true)
                newList.add(item);

        return newList;
    }

    //this function remove last index in string
    public static void removeLastIndexInString(TextField textField) {
        String str = textField.getText();
        textField.setText(str.substring(0,str.length()-1)); // remove the last index
        Platform.runLater( new Runnable() {
            @Override
            public void run() {
                textField.positionCaret(str.length()-1); // set position caret in the end of the string
            }
        });
    }
}

```

## קובץ RowTable.java

מחלקת שינוי צבעים של השורות בטבלה.

```
package application;

import java.sql.SQLException;
import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.event.EventHandler;
import javafx.scene.control.TableRow;
import javafx.scene.control.TableView;
import javafx.scene.input.MouseEvent;
import javafx.util.Duration;

public class RowTable {

    // this function for mouse event
    private <T> void MouseEvent(TableRow<T> row,String color) {
        row.setOnMouseEntered(new EventHandler<MouseEvent>() {

            @Override
            public void handle(MouseEvent t) {
                // if mouse enter row in the table change color
                row.setStyle(row.getStyle() + "-fx-background-color: rgba(28, 183, 236, 0.25);");
            }
        });
        row.setOnMouseExited(new EventHandler<MouseEvent>() {

            @Override
            public void handle(MouseEvent t) {
                // if mouse exit row in the table change color
                row.setStyle(row.getStyle() + "-fx-border-color: -fx-box-border;-fx-background-color: " + color +";");
            }
        });
    }

    //this function change row color in the table
    public <T> void changeRowsColor(TableView<T> table) {

        // goes through all the rows within the table
        table.setRowFactory( tv -> new TableRow<T>(){

            @Override
            protected void updateItem(final T item, final boolean empty){
                super.updateItem(item, empty);
                //check if the row is empty and item not equal to null
                if (!empty && item != null ){
                    // set sort table false
                    for(int i = 0;i < table.getColumns().size();i++) {
                        table.getColumns().get(i).setSortable(false);
                    }
                    //if item is product
                    if(item instanceof Product) {
                        //check the quantity of the product
                        //if equals to zero

                        if(((Product)item).getQuantity() == 0) {
                            setStyle(getStyle() + "-fx-border-color: -fx-box-border;-fx-background-color: #ff3333;");//change
                            color of the row to red (ASCII code of the color)
                            MouseEvent(this,"#ff3333");
                        } else if (((Product)item).getQuantity() < 10) { // if the quantity of the product less than 10 change color the row
                            to orange
                            setStyle(getStyle() + "-fx-border-color: -fx-box-border;-fx-background-color: #FFCC00;");
                            MouseEvent(this,"#FFCC00");
                        } else { // else get style of the row in css file
                            setStyle(getStyle() + "-fx-border-color: -fx-box-border;-fx-background-color: lightgrey;");
                            MouseEvent(this,"lightgrey");
                        }
                    } else if (item instanceof Category) { // if item is category
                        try {
                            //set all products in the inventory in category
                            ((Category)item).setProductList(DatabaseManager.getProducts());
                        } catch (ClassNotFoundException | SQLException e) {
                            e.printStackTrace();
                        }
                    }
                    // check if the category is empty
                }
            }
        });
    }
}
```

```

        if(!((Category)item).checksCategory((Category)item)) {
            setStyle(getStyle() + "-fx-border-color: -fx-box-border;-fx-background-color:
#ff3333;");//change color to red
        } else {
            MouseEvent(this,"#ff3333");
            setStyle(getStyle() + "-fx-border-color: -fx-box-border;-fx-background-color:
lightgrey;");// else get style of the row in css file
            MouseEvent(this,"lightgrey");
        }
    } else if(item instanceof Shelf) { //if item is shelf
        try {
            //set all products of the shelf in the shelf
            ((Shelf)item).setProductsList(DatabaseManager.getProductsInShelf((Shelf)item));
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        //Check if the shelf weight is greater than 90% relative to the maximum weight
        if(((Shelf)item).getShelfWeight() / ((Shelf)item).getMaxWeight() >= 0.9) {
            setStyle(getStyle() + "-fx-border-color: -fx-box-border; -fx-background-color:
#ff3333;");//if true change the row background to red
            MouseEvent(this,"#ff3333");
        } else {
            setStyle(getStyle() + "-fx-border-color: -fx-box-border; -fx-background-color:
lightgrey;");
            MouseEvent(this,"lightgrey");
        }
    }
} else //else set style empty
    setStyle("");
}
});
}

//this function flashing the specific row in the table
public void flashingSingleRow(Table<Product> table,Product product) {

    // goes through all the rows within the table
    table.setRowFactory(tv -> new TableRow<Product>(){
        @Override
        protected void updateItem(final Product item, final boolean empty){
            super.updateItem(item, empty);

            if ( !empty && item != null ){
                if(item.getProductCode() == product.getProductCode()) { // check if the item equal to product code

                    String style = getStyle();

                    Timeline flasher = new Timeline(
                        new KeyFrame(Duration.seconds(0.3), e -> {
                            setStyle(getStyle() + "-fx-background-color: #ff3333;");
                        }),
                        new KeyFrame(Duration.seconds(0.5), e -> {
                            setStyle(style);
                        })
                    );

                    // if true flash the row
                    flasher.setCycleCount(5); // cycle count of the time line is 5
                    flasher.setAutoReverse(true);
                    flasher.play();
                }
            } else
                setStyle("");
        }
    });
}

}

```

## קובץ SwitchingPages.java

מחלקת העברת חלון לחלון אחר.

```
package application;

import java.io.IOException;
import javafx.event.Event;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class SwitchingPages {
    // switch page
    public static void switchPage(Event event,String pageName,String title) throws IOException {
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(SwitchingPages.class.getResource(pageName)); // switch a page according to page name
        Stage window = (Stage)((Node)event.getSource()).getScene().getWindow(); // get the window
        //get a last screen x and y and set a x and y in new screen
        window.setX(window.getX());
        window.setY(window.getY());
        window.setScene(new Scene(loader.load()));
        window.setResizable(false); // The screen cannot be resized
        window.setTitle(title); // title
        window.show();
    }
    // switch page to login
    public static void openWindowInCenter(String pageName,Event event,String title) throws IOException {
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(SwitchingPages.class.getResource( pageName + ".fxml"));
        Stage window = (Stage)((Node)event.getSource()).getScene().getWindow(); // get the window
        window.setScene(new Scene(loader.load()));
        window.setResizable(false);// The screen cannot be resized
        window.centerOnScreen();// set a screen in center
        window.setTitle(title); // title
        window.show();
    }
}

// exit window
public static void exitStage(Event event){
    Node source = (Node) event.getSource();
    Stage stage = (Stage) source.getScene().getWindow();
    stage.close();
}
}
```

## קובץ LogoutClass.java

מחלקת התנתקות של המשתמש ברגע ויש מוצרים בתוך סל הקניות.

```
package application;

import java.sql.SQLException;
import javafx.application.Platform;
import javafx.collections.ObservableList;
import javafx.scene.control.Alert.AlertType;

public class LogoutClass {

    static String str = "";
    //logout button
    public static void logoutWindow() {
        try {
            Manager manager = DatabaseManager.getManager();
            if (manager.getStatus().equals(true)) {
                manager.setStatus(false);
                DatabaseManager.editManagerStatus(manager);
            }
            if(!DatabaseManager.getProductFromShoppingCart().isEmpty()) {
                // receive a message
                if(Message.showMessageWithYesNoButtons(AlertType.INFORMATION,"Are you sure? , if you
exit you will delete the products in the shopping cart", "") == buttonType.YES) {
                    // if the user clicked yes return all product to shelves
                    returnProductsToShelves();
                    DatabaseManager.removeShoppingCart(); // remove all product in shopping cart
                    Platform.exit(); // exit
                }
            } else
                Platform.exit(); // exit
        } catch (ClassNotFoundException | SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    //this function return the all products in shopping cart to the shelves
    public static void returnProductsToShelves() throws ClassNotFoundException, SQLException {
        ObservableList<Product> productList = DatabaseManager.getProductFromShoppingCart(); // get all product in shopping cart
        for(Product product:productList)
            removeProduct(product);

        Message.showMessageWithOKButton(AlertType.INFORMATION, str, null);
        DatabaseManager.removeShoppingCart();
    }

    // remove product from the products table
    private static void removeProduct(Product product) throws ClassNotFoundException, SQLException {
        // get quantity of the product in shelf
        // update products in shelf

        Shelf sh = new Shelf();
        sh.setProductsList(DatabaseManager.getProductsInShelves());
        Product []products = sh.getProductInShelves(product);

        for(int i = 0; i < products.length;i++) {
            products[i].setShelvesList(DatabaseManager.getShelves());
            Shelf shelf = products[i].getShelf(products[i].getShelfNumber());
            int productQuantity = DatabaseManager.getProductQuantityInShelf("shelf_product",product,shelf);
            shelf.setProductsList(DatabaseManager.getProductsInShelf(shelf));

            if(shelf.checkWeight(product, product.getQuantity())) {
                DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity +
product.getQuantity());
                if(str != "")
                    str += " and"; // build string contains products returns it to shelves
                str += " " + product.getQuantity() + " of " + product.getProductname() + " returns to shelf number :
" + shelf.getShelfNumber();
            } else {
                product.setQuantity(0);
                int quantity = shelf.getProductQunatity(products[i]);
                product.setQuantity(product.getQuantity() - quantity);
            }
        }
    }
}
```

```

quantity);

        DatabaseManager.updateProductInShelf("shelf_product", product, shelf, productQuantity +

shelf.getShelfNumber());

        if(str != "")
            str += " and"; //build string contains products returns it to shelves
        str += " " + quantity + " of " + product.getProductName() +" returns to shelf number : " +

    }
    // if qunatity of product equal or less than zero do break
    if(product.getQuantity() <= 0)
        break;
    }
    DatabaseManager.deleteProduct(product,"shopping_cart"); // delete selected product in shelf

}

}

```

```
package application;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import com.pdfjet.A4;
import com.pdfjet.Cell;
import com.pdfjet.Color;
import com.pdfjet.CoreFont;
import com.pdfjet.Font;
import com.pdfjet.PDF;
import com.pdfjet.Page;
import com.pdfjet.Table;
import javafx.scene.control.TableView;
import javafx.geometry.Pos;
import javafx.scene.control.Alert.AlertType;

public class PdfFile {

    // this function to download pdf file of order details
    public void saveOrderPdf(Table<Product> viewOrderTable, Order order) throws ClassNotFoundException, SQLException {
        // customer of the order
        Customer customer = order.getCurrentCustomer(DatabaseManager.getCustomer(), order.getCustomerId());

        if(customer != null) {
            File out = new File("C:\\Users\\User\\Downloads\\order_" + order.getId() + ".pdf");
            FileOutputStream fos;
            try {
                fos = new FileOutputStream(out);
                PDF pdf = new PDF(fos);
                Page page = new Page(pdf, A4.PORTRAIT);

                //font for the headers table
                Font f1 = new Font(pdf, CoreFont.HELVETICA_BOLD);

                //font for the pdf table data
                Font f2 = new Font(pdf, CoreFont.HELVETICA);

                Table table = new Table();

                List<List<Cell>> tableData = new ArrayList<List<Cell>>();

                //table row
                List<Cell> row = new ArrayList<Cell>();

                // title of the file
                Cell cell = new Cell(f1, "Personal information");
                cell.setBrushColor(Color.red);
                cell.setNoBorders();
                row = new ArrayList<Cell>();
                row.add(cell);
                tableData.add(row);

                // set name
                Cell cellName = new Cell(f1, "Name : " + customer.getFullName()); //full name of the customer
                cellName.setNoBorders(); // set no border for the cell
                row = new ArrayList<Cell>(); // new row
                row.add(cellName);

                // empty columns
                for(int i=0; i < 2; i++) {
                    Cell none = new Cell(f1, "");
                    none.setNoBorders();
                    row.add(none);
                }

                // set date of order
                Cell date = new Cell(f1, "Date : " + order.getOrderDate());
                date.setNoBorders(); // set no border for the cell
                row.add(date);
                tableData.add(row);

                // set customer id
                Cell cellId = new Cell(f1, "I.D : " + String.valueOf(customer.getId())); //customer id of the customer
                cellId.setNoBorders();
                row = new ArrayList<Cell>();
            }
        }
    }
}
```

```

row.add(cellId);
tableData.add(row);

row = new ArrayList<Cell>();

//headers of the table

// set full name
Cell cell1 = new Cell(f1, viewOrderTable.getColumns().get(2).getText());
row.add(cell1);

// set order id
cell1 = new Cell(f1, viewOrderTable.getColumns().get(0).getText());
row.add(cell1);

// set quantity of product
cell1 = new Cell(f1, viewOrderTable.getColumns().get(3).getText());
row.add(cell1);

// set price of product
cell1 = new Cell(f1, viewOrderTable.getColumns().get(4).getText());
row.add(cell1);

//add headers to table
tableData.add(row);

//now let's get table view data, create a row for each and add row to table row
List<Product> productsList = viewOrderTable.getItems();

for (Product product : productsList) {
    Cell productName = new Cell(f2, product.getProductName());
    Cell orderId = new Cell(f2, String.valueOf(product.getOrderId()));
    Cell quantity = new Cell(f2, String.valueOf(product.getQuantity()));
    Cell price = new Cell(f2, String.valueOf(product.getPrice()));
    productName.setTextAlignment(Pos.CENTER.ordinal());
    row = new ArrayList<Cell>();
    row.add(productName);
    row.add(orderId);
    row.add(quantity);
    row.add(price);

    //add row to table
    tableData.add(row);
}
// empty row
row = new ArrayList<Cell>();
Cell none = new Cell(f1, "");
none.setNoBorders();
row.add(none);
tableData.add(row);

// total price row
row = new ArrayList<Cell>();
Double totalPrice = order.getTotalPrice(DatabaseManager.getProductOfOrder(order)); //get total price
Cell total = new Cell(f1, "Total price : " +
String.valueOf(Conditions.get2DigitisAfterPoint(String.valueOf(totalPrice))) + " ₪");
total.setBrushColor(Color.orangered);
total.setNoBorders();
row.add(total);
tableData.add(row);

table.setData(tableData);
table.setTextAlignInColumn(50, 0);
table.setPosition(70f, 60f);
table.setColumnWidth(0, 140f);
table.setColumnWidth(1, 100f);
table.setColumnWidth(2, 140f);
table.setColumnWidth(3, 140f);

//create a new page and add more table data if we get to the bottom of the current page
while (true) {
    table.drawOn(page);
    if (!table.hasMoreData()) {
        table.resetRenderedPagesCount();
        break;
    }
    page = new Page(pdf, A4.PORTRAIT);
}

pdf.flush();

```



```

        fos.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
    Message.showMessageDialog(AlertType.INFORMATION, "Saved to " + out.getAbsolutePath(), "PDF File");
} else
    Message.showMessageDialog(AlertType.INFORMATION, "this customer not existing you have to
add his details to customers list", "");
}
}

```

## ביבליוגרפיה

קישור לאתר :

[/https://stackoverflow.com](https://stackoverflow.com)

נעזרנו בספר בשם:

Java על כוס קפה

קישור הדרכה עבודה על **SceneBuilder** ב **JavaFX** :

<https://www.youtube.com/watch?v=fFq9-MLvIfE>