בס"ד



בית הספר: מאיר

שם העבודה: פליסה

מגישה: מיכל מזוז

ת. ז.: 214378762

המנחה: חנה אייזנשטין

חלופה: מערכות מנהליות

שנה: קיץ תשפ"א

תוכן עניינים

[גורמים מעורבים: 3](#_Toc106022945)

[תיחום ארגוני: 4](#_Toc106022946)

[תיחום תהליכי: 4](#_Toc106022947)

[יעדים: 4](#_Toc106022948)

[המחלקה Dal.cs 6](#_Toc106022949)

[המחלקה GeneralDB 8](#_Toc106022950)

[מחלקות הרבים:*nameTable*DB: 8](#_Toc106022951)

[מחלקות היחיד nameTableRow 23](#_Toc106022952)

[טופס מסך הבית 38](#_Toc106022953)

[טופס ספקים 40](#_Toc106022954)

[טופס הוספת ספק חדש 42](#_Toc106022955)

[טופס לקוחות 45](#_Toc106022956)

[טופס הוספת לקוח 47](#_Toc106022957)

[טופס ניהול מערכת 50](#_Toc106022958)

[טופס הזמנות 62](#_Toc106022959)

[טופס מוצרים 66](#_Toc106022960)

[טופס מידות ואורכים 74](#_Toc106022961)

[מסך ניהול מערכת 82](#_Toc106022962)

[מסך הבית 83](#_Toc106022963)

[מסך ספקים 84](#_Toc106022964)

[מסך משני הוספת ספק 84](#_Toc106022965)

[מסך משני עדכון ספק 85](#_Toc106022966)

[מסך משני מחיקת ספק 85](#_Toc106022967)

[מסך משני הצגת ספק 86](#_Toc106022968)

[מסך משני רענן תצוגה 86](#_Toc106022969)

[מסך לקוחות 87](#_Toc106022970)

[מסך משני הוספת לקוח 87](#_Toc106022971)

[מסך משני עדכון לקוח 88](#_Toc106022972)

[מסך משני הצגת לקוח 88](#_Toc106022973)

[מסך משני רענן תצוגה 89](#_Toc106022974)

[מסך משני מחיקת לקוח 89](#_Toc106022975)

[מסך מוצרים 90](#_Toc106022976)

[הוספת מוצר 90](#_Toc106022977)

[מחיקת מוצר 91](#_Toc106022978)

[ריענון תצוגה 92](#_Toc106022979)

[הצגת מוצר 92](#_Toc106022980)

[עריכת מוצר 93](#_Toc106022981)

[מסך הזמנות מספקים 94](#_Toc106022982)

[מסך קנית לקוח 95](#_Toc106022983)

[מסך ניהול הזמנות 96](#_Toc106022984)

מבוא

מהי חנות? PLISSEE

פליסה היא חנות המספקת תלבושת אחידה לבנות סמינר בירושלים.

לחנות פליסה מגיעות לקוחות מכל רחבי הארץ!

צוות פליסה נותן מענה ושרות מסור לכל לקוחה ומסייע לך לרכוש את הפריט הרצוי והמתאים לך במיוחד.

רכישת פריט תלבושת בPLISSEE מבטיחה לך הופעה מוקפדת, נוחות לאורך היום ואיכות לאורך זמן.

פליסה נחשבת למותג אהוב ונחשק בגלל ההתאמה המדויקת לא פחות מ-10 אורכים לחצאיות פליסה!, וכן מבחר הפריטים וטווח המחירים הרחב.

צוות פליסה:

צוות פליסה כולל מעצבות אופנה מקצועיות העמלות להביא פריטים עדכניים, איכותיים ובמחירים הוגנים.

שעות פעילות החנות:

בכל יום בשעות אחר הצהריים ובשעות הערב המוקדמות.

ניתנת אפשרות של החלפה עד שבועיים מיום הרכישה, והחזרה תוך 48 שעות משעת הרכישה.

תיאור התוכנה ומטרתה:

תוכנת PLISSEE תוכננה ועוצבה כדי לספק הן ללקוחות הן לספקים והן לעובדים בה שירות ברמה הגבוה ביותר.

התוכנה מאפשרת התמצאות קלה, מעבר מהיר בין השירותים השונים וגישה נוחה ובטוחה למגוון פעילויות בנושא הקניה וההזמנה.

מטרות משניות:

* שמירה מסודרת של נתוני ההזמנות והקניות.
* יחס אדיב ללקוחות.
* מחירים נוחים.
* צפייה בפרטי הספקים, הלקוחות, המוצרים, ניתן להוסיף/לעדכן/למחוק את הפרטים השונים.
* אפשרות בלעדית לצפות/לעדכן/למחוק את ההזמנות מספקים וקניות של לקוחות.
* ביצוע הזמנות מספקים בצורה נוחה ובטוחה.

מה הוביל אותי לבחירת הנושא?

קרובת משפחתי בעלת עסק זה, ובפעמים רבות בזמני לחץ ועומס בחנותה היא נעזרה בי, נהנתי לעקוב אחר התנהלות החנות, וכך עלה במוחי הרעיון ליצור תוכנה שתסייע ותקל הן למוכרת ותפשט את תהליך ההזמנה, והן לרוכשים המגיעים מכל רחבי הארץ, כדי שכולם יוכלו ליהנות מחוויית קניה פרקטית ואיכותית.

תרשים מבנה ארגוני של החנות:

מהות המשימה:

במערכת המידע הידנית של החנות יש בעיות אחדות:

1. אין מאגר מסודר של לקוחות וספקים ולכן כאשר רוצים להוסיף לקוח/ספק חדש יכולה להיווצר כפילות.
2. קשה לעקוב אחר המוצרים המוזמנים ולכן יכולות להיווצר כפילויות בהזמנות הסחורה מהספקים.
3. קושי במעקב אחר המלאי בחנות.
4. קשה לערוך קניה מסודרת של לקוח ולפעמים נוצרות טעויות בחשבון.

תיחום המערכת:

גורמים מעורבים:

|  |  |  |
| --- | --- | --- |
| גורם | תחום אחריות | רמת מעורבות |
| מנהל החנות | קביעת מחירים.  ניהול קטלוג.  הוספת ספקים חדשים  מעקב אחר קניות הלקוחות והזמנות מספקים | משתמש עיקרי |
| מנהל חשבונות | מקבל דו"חות תשלומי קניות ופרטי הזמנות מספקים. | משתמש משני |
| מוכרים | עריכת חשבון ללקוח וניהול ההזמנה.  הוספת לקוח.  עדכון פרטי לקוח.  העברת מידע לאחראי ההזמנות על המוצרים החסרים. | משתמש עיקרי. |
| אחראי הזמנות ספקים | הזמנת מוצרים מספקים.  קבלת ההזמנה.  עדכון המלאי. | משתמש עיקרי |

תיחום ארגוני:

המערכת נועדה לפעילותם של מנהל החנות, קופאי ואחראי ההזמנות מספקים.

תיחום תהליכי:

1. סיוע בניהול מלאי המוצרים.
2. ניהול קטלוג המוצרים.
3. ניהול מאגר הלקוחות.
4. עזרה במעקב אחר הקניות והתשלומים.
5. סיוע בניהול הספקים והזמנות מספקים.
6. המערכת לא מטפלת בחישוב משכורות עובדים.

יעדים:

|  |  |
| --- | --- |
| שם היעד | תיאור |
| ניהול מאגרים | ניהול מאגרי המידע הבאים: לקוחות, ספקים, מוצרים, הזמנות מספקים, קניות. |
| טיפול בפרטי לקוח | * קליטת נתונים פורמליים של לקוח והוספתם למאגר הלקוחות. * עדכון פרטי לקוח לפי דרישת הלקוח או המנהל. * מציאת פרטי לקוח בחיפוש חכם. * הצגת פרטי לקוח מסוים. |
| טיפול בפרטי ספק | * קליטת נתונים פורמליים של ספק והוספתם למאגר הספקים. * עדכון פרטי ספק לפי דרישת הספק או המנהל. * מציאת פרטי ספק בחיפוש חכם. * הצגת פרטי ספק מסוים. |
| טיפול בפרטי מוצר | * קליטת נתונים פורמליים של מוצר והוספתם למאגר המוצרים. * עדכון פרטי מוצר לפי דרישת המנהל. * מציאת פרטי מוצר על פי קטגוריה: שם מוצר/תיאור/סוג מוצר.... * הצגת פרטי מוצר מסוים. * הסרה של מוצר מקטלוג המוצרים. |
| טיפול בפרטי הזמנה מספק | שליחת הזמנה לספק ע"י:   * קליטת נתונים פורמליים של הזמנה והוספתם למאגר ההזמנות. * עדכון פרטי ספק לפי דרישת המוכר או אחראי ההזמנות מספקים. * הצגת פרטי הזמנה מסוימת. * אישור קבלת הזמנה מספק. * ביטול הזמנה מספק. |
| טיפול בפרטי קניה | * קליטת נתונים פורמליים של קניה והוספתם למאגר הקניות. * עדכון פרטי קניה לפי דרישת המוכר. * מציאת פרטי קניה על פי תאריך/ לקוח/ סטטוס. * הצגת פרטי קניה מסוים. |

אילוצי המערכת:

1. **לו"ז-** הגשת הפרויקט עד סוף יוני 2022.
2. **כ"א-** פיתוח ומימוש התבצע ע"י תלמידת תיכון במסגרת פרויקט גמר.
3. **טכנולוגיה-** שימוש במחשב PC.
4. סביבת פיתוח- C#- visual studio 10 , מסד נתונים Access 2016 בטכנולוגית Window Forms.

מבנה וארכיטקטורה:

תוכנת 'פליסה' היא תוכנה לניהול חנות פליסה המתייחסת ל-2 תהליכים מרכזיים:

1. מכירת יצירות ללקוח, וניהול מאגר היצירות.
2. הזמנת מוצרים מהספק.

התוכנה נכתבה בשפת C# בסביבת Visual Studio 2019.

מחלקות ועצמים, קלט/פלט, ממשקי משתמש ידידותיים, תמיכה בבסיס נתונים, הורשה , טפסים.

התוכנית נבנתה בצורת חשיבה של DNA – מודל שלושת השכבות.

התוכנית חולקה לשלוש שכבות כאשר כל שכבה מטפלת בתחום שונה ומכילה כמה מחלקות ו/או טפסים שונים. בתוך כל מחלקה הוגדרו משתנים ואובייקטים מתאימים המיוחדים למחלקה. כל מחלקה מטפלת בתחום מסוים ומתקשרת לטופס/המחלקה המתאימים לה. צורת כתיבה זו נועדה להקל על המתכנת הן בכתיבת הקוד והן בתחזוקה השוטפת של הקוד. כמובן שבין המחלקות והטפסים שבשכבות השונות קיים קשר, וקימת התייחסות מתוך מחלקה אחת לעצמים המוגדרים במחלקות אחרות אך הכל נעשה תוך שמירה על בטחון והבטחת הנתונים והפונקציות.

פרוט השכבות:

1. DAL Data Access Layer – שכבה המטפלת בקישור התוכנה עם בסיס הנתונים.
2. BLL Business Logic Layer – שכבה המטפלת בכל הפונקציות הקשורות לתוכנה עצמה.
3. GUI Graphic User Inerface – שכבה המטפלת בממשק החיצוני של התוכנה.

מדריך למפתח

שכבת הנתונים

שכבה זו מטפלת בקישור התוכנה עם בסיס הנתונים.

ובה מחלקה לתקשורת עם ה DataBase . Dal.cs

המחלקה Dal.cs

היחידה ש"מדברת" עם ה- DataBase כל שאר המחלקות פונות ל-DataBase דרך מחלקה זו. במלקה זו יש מספר פונקציות ששולפות, מעדכנות, מוחקות ומוסיפות ל- DataBase.

public class Dal

{

private static Dal instance;

private OleDbConnection con;

private DataSet ds;

private Dal(string connectionString)

{

con = new OleDbConnection(connectionString);

ds = new DataSet();

}

public static Dal GetInstance()

{

if (instance == null)

{

string path = System.IO.Directory.GetCurrentDirectory();

int x = path.IndexOf("\\bin");

path = path.Substring(0, x) + "\\Dal\\ Databaseplisse1.accdb";

instance = new Dal(@"Provider= Microsoft.ACE.OLEDB.12.0;Data Source='" + path + "';Persist Security Info=True");

}

return instance;

}

public void AddTable(string tableName)

{

if(!ds.Tables.Contains(tableName))

{

OleDbDataAdapter adapter = new OleDbDataAdapter(" Select \* from " + tableName, con);

adapter.Fill(ds, tableName);

}

}

public DataTable GetTable(string tableName)

{

if (!ds.Tables.Contains(tableName))

AddTable(tableName);

return ds.Tables[tableName];

}

public bool RemoveTable(string tableName)

{

bool suceed = true;

try

{

ds.Tables.Remove(tableName);

}

catch

{

suceed = false;

}

return suceed;

}

public void Update(string tableName)

{

OleDbDataAdapter adapter = new OleDbDataAdapter("select\*from " + tableName, con);

OleDbCommandBuilder builder = new OleDbCommandBuilder(adapter);

adapter.InsertCommand = builder.GetInsertCommand();

adapter.UpdateCommand = builder.GetUpdateCommand();

adapter.DeleteCommand = builder.GetDeleteCommand();

adapter.Update(ds, tableName);

}

public void Update()

{

foreach(DataTable table in ds.Tables)

{

Update(table.TableName);

}

}

}

שכבת העסק - BLL

שכבה המטפלת בכל הפונקציות והלוגיקה הקשורות לתוכנה עצמה.

המחלקות המוכלות בשכבה:

* מחלקת בסיס לטיפול בטבלה אחת מתוך ה DataBase – GeneralDB
* מחלקה עבור כל טבלה מה DataBaseהיורשת מהמחלקה GeneralDB
* מחלקות *nameTable*Ro*w* –תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני.

המחלקה GeneralDB

היא היחידה המתקשרת עם מחלקת Dal. מנהלת קשר עם טבלה בודדת מתוך ה DataBase. לא ניתך ליצור ממנה עצמים כיוון שהיא כללית ולא ספציפית לטבלה מסוימת ב DataBase. המחלקות היורשות ממנה הם מתייחסות כל אחת לטבלה אחרת מהמאגר.

תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני.

public abstract class GeneralDB

{

protected DataTable table;

public GeneralDB(string tableName)

{

table = Dal.Dal.GetInstance().GetTable(tableName);

}

public int Size()

{

return table.Rows.Count;

}

public void Update()

{

Dal.Dal.GetInstance().Update(table.TableName);

}

public void Add(DataRow dr)

{

table.Rows.Add(dr);

this.Save();

}

public virtual void Save()

{

Dal.Dal.GetInstance().Update(table.TableName);

}

}

מחלקות הרבים:*nameTable*DB:

כל אחת מן המחלקות הבאות יורשת מהמחלקה GeneralDB ומנהלת את הטיפול בטבלה בודדת מתוך ה DataBase

**מחלקת CustomerDB**

class CustomerDB:GenerlDB

{

public CustomerDB() : base("tbl\_customers")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

list.Add(new Customer(dr));

}

}

public List<Customer> GetList()

{

list.Clear();

DataTableToList();

return list;

}

//Add new customer

public void AddNew(Customer c)

{

c.Dr = table.NewRow();

c.FillDataRow();

this.Add(c.Dr);

}

//Search for customer by customer\_tz

public Customer Find(string tz)

{

return GetList().Find(x => x.Tz == tz);

}

//Update row in the table by the noun

public void UpDateRow(Customer c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(string tz)

{

Customer c = this.Find(tz);

this.GetList().Find(x => x.Tz == tz);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת CityDB**

class CityDB:GenerlDB

{

public CityDB() : base("tbl\_city")

{

{

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new City(dr));

}

}

public List<City> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new city

public void AddNew(City c)

{

c.Dr = table.NewRow();

c.FillDataRow();

this.Add(c.Dr);

}

//Search for customer by customer\_tz

public City Find(int codeC)

{

return GetList().Find(x => x.CityCode == codeC);

}

//Update row in the table by the noun

public void UpDateRow(City c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int cityCode)

{

City c = this.Find(cityCode);

this.GetList().Find(x => x.CityCode == cityCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת CompanyDB**

**class CompanyDB:GenerlDB**

{

public CompanyDB() : base("tbl\_company")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new Company(dr));

}

}

public List<Company> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new company

public void AddNew(Company c)

{

c.Dr = table.NewRow()

c.FillDataRow();

this.Add(c.Dr);

}

//Search for company by compantCode

public Company Find(int companyCode)

{

return GetList().Find(x => x.CompanyCode == companyCode);

}

//Update row in the table by the noun

public void UpDateRow(Company c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int companyCode)

{

Company c = this.Find(companyCode);

this.GetList().Find(x => x.CompanyCode == companyCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת DetailPaymentsDB**

class DetailPaymentsDB:GenerlDB

{

public DetailPaymentsDB() : base("tbl\_detail\_payments")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

list.Add(new DetailPayments(dr));

}

}

public List<DetailPayments> GetList()

{

list.Clear();

DataTableToList();

return list;

}

//Add new DetailPayments

public void AddNew(DetailPayments d)

{

d.Dr = table.NewRow();

d.FillDataRow();

this.Add(d.Dr);

}

//Search for customer by customer\_tz

public DetailPayments Find(int paymentCode,int paymentNumber)

{

return GetList().Find(x => x.PaymentCode == paymentCode&&x.PaymentNumber==paymentNumber);

}

//Update row in the table by the noun

public void UpDateRow(DetailPayments c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int paymentCode, int paymentNumber)

{

DetailPayments c = this.Find(paymentCode, paymentNumber);

this.GetList().Find(x => x.PaymentCode == paymentCode && x.PaymentNumber == paymentNumber);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת InstitutionDB**

class InstitutionDB:GenerlDB

{

List<Institution> list = new List<Institution>();

public InstitutionDB() : base("tbl\_instituation")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

list.Add(new Institution(dr));

}

}

public List<Institution> GetList()

{

list.Clear();

DataTableToList();

return list;

}

//Add new institution

public void AddNew(Institution i)

{

i.Dr = table.NewRow();

i.FillDataRow();

this.Add(i.Dr);

}

//Search for Institution by institutionCode

public Institution Find(int institutionCode)

{

return GetList().Find(x => x.InstitutionCode == institutionCode);

}

//Update row in the table by the noun

public void UpDateRow(Institution c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int institutionCode)

{

Institution c = this.Find(institutionCode);

this.GetList().Find(x => x.InstitutionCode == institutionCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת InstitutionProductDB**

class InstitutionProductDB:GenerlDB

{

public InstitutionProductDB() : base("tbl\_institution's\_product")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new InstitutionProduct(dr));

}

}

public List<InstitutionProduct> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new InstitutionProduct

public void AddNew(InstitutionProduct i)

{

i.Dr = table.NewRow();

i.FillDataRow();

this.Add(i.Dr);

}

//Search for InstitutionProduct by productCode && institutionCode

public InstitutionProduct Find(int productCode,int institutionCode)

{

return GetList().Find(x => x.ProductCode == productCode&&x.InstitutionCode==institutionCode);

}

//Update row in the table by the noun

public void UpDateRow(InstitutionProduct c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int productCode, int institutionCode)

{

InstitutionProduct c = this.Find(productCode,institutionCode);

this.GetList().Find(x => x.ProductCode == productCode && x.InstitutionCode == institutionCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת LengthDB**

class LengthDB:GenerlDB

{

public LengthDB() : base("tbl\_length")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new Length(dr));

}

}

public List<Length> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new length

public void AddNew(Length l)

{

l.Dr = table.NewRow();

l.FillDataRow();

this.Add(l.Dr);

}

//Search for Length by lengthCode

public Length Find(int lengthCode)

{

return GetList().Find(x => x.LengthCode == lengthCode);

}

//Update row in the table by the noun

public void UpDateRow(Length c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int lengthCode)

{

Length c = this.Find(lengthCode);

this.GetList().Find(x => x.LengthCode == lengthCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת OrderDetailDB**

class OrderDetailDB:GenerlDB

{

public OrderDetailDB() : base("tbl\_order\_detail")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new OrderDetail(dr));

}

}

public List<OrderDetail> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new OrderDetail

public void AddNew(OrderDetail o)

{

o.Dr = table.NewRow();

o.FillDataRow();

this.Add(o.Dr);

}

//Search for OrderDetail by orderNumber && productCode

public OrderDetail Find(int orderNumber,int productCode,int sizeCode,int lengthCode)

{

return GetList().Find(x => x.OrderNumber == orderNumber && x.ProductCode == productCode && x.SizeCode == sizeCode&&x.LengthCode==lengthCode);

}

//Update row in the table by the noun

public void UpDateRow(OrderDetail c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int orderNumber, int productCode, int sizeCode, int lengthCode)

{

OrderDetail c = this.Find(orderNumber, productCode,sizeCode,lengthCode);

this.GetList().Find(x => x.OrderNumber == orderNumber && x.ProductCode == productCode && x.SizeCode == sizeCode && x.LengthCode == lengthCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת PaymentsDB**

class PaymentsDB:GenerlDB

{

public PaymentsDB() : base("tbl\_payments")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new Payments(dr));

}

}

public List<Payments> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new payments

public void AddNew(Payments m)

{

m.Dr = table.NewRow();

m.FillDataRow();

this.Add(m.Dr);

}

//Search for customer by customer\_tz

public Payments Find(int paymentCode)

{

return GetList().Find(x => x.PaymentCode == paymentCode);

}

//Update row in the table by the noun

public void UpDateRow(Payments c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int paymentCode)

{

Payments c = this.Find(paymentCode);

this.GetList().Find(x => x.PaymentCode == paymentCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת ProductModelsDB**

class ProductModelsDB:GenerlDB

{

public ProductModelsDB() : base("tbl\_products\_models")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new ProductModels(dr));

}

}

public List<ProductModels> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new ProductModels

public void AddNew(ProductModels m)

{

m.Dr = table.NewRow();

m.FillDataRow();

this.Add(m.Dr);

}

//Search for ProductModels by pCode

public ProductModels Find(int pCode)

{

return GetList().Find(x => x.ProductCode == pCode);

}

//Update row in the table by the noun

public void UpDateRow(ProductModels c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int typeCode)

{

ProductModels c = this.Find(typeCode);

this.GetList().Find(x => x.TypeCode == typeCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

**מחלקת ProductModelsDB**

class ProductSizeDB:GenerlDB

{

public ProductSizeDB() : base("tbl\_products\_sizes")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new ProductType(dr));

}

}

public List<ProductType> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new ProductType

public void AddNew(ProductType p)

{

p.Dr = table.NewRow();

p.FillDataRow();

this.Add(p.Dr);

}

//Search for ProductType by typeCode

public ProductType Find(int typeCode)

{

return GetList().Find(x => x.TypeCode == typeCode);

}

//Update row in the table by the noun

public void UpDateRow(ProductType c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int typeCode)

{

ProductType c = this.Find(typeCode);

this.GetList().Find(x => x.TypeCode == typeCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת ProductTypeDB**

class ProductTypeDB:GenerlDB

{

public ProductTypeDB() : base("tbl\_product\_type")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new ProductType(dr));

}

}

public List<ProductType> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new ProductType

public void AddNew(ProductType p)

{

p.Dr = table.NewRow();

p.FillDataRow();

this.Add(p.Dr);

}

//Search for ProductType by typeCode

public ProductType Find(int typeCode)

{

return GetList().Find(x => x.TypeCode == typeCode);

}

//Update row in the table by the noun

public void UpDateRow(ProductType c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int typeCode)

{

ProductType c = this.Find(typeCode);

this.GetList().Find(x => x.TypeCode == typeCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת SaleDB**

class SaleDB:GenerlDB

{

public SaleDB() : base("tbl\_sale")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new Sale(dr));

}

}

public List<Sale> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new sale

public void AddNew(Sale s)

{

s.Dr = table.NewRow();

s.FillDataRow();

this.Add(s.Dr);

}

//Search for sale by salesCode

public Sale Find(int salesCode)

{

return GetList().Find(x => x.SalesCode == salesCode);

}

//Update row in the table by the noun

public void UpDateRow(Sale c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int salesCode)

{

Sale c = this.Find(salesCode);

this.GetList().Find(x => x.SalesCode == salesCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת SaleDetailDB**

class SaleDetailDB:GenerlDB

{

public SaleDetailDB() : base("tbl\_sale\_detail")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new SaleDetail(dr));

}

}

public List<SaleDetail> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new SaleDetail

public void AddNew(SaleDetail s)

{

s.Dr = table.NewRow();

s.FillDataRow();

this.Add(s.Dr);

}

//Search for SaleDetail by saleCode & productCode & sizeCode & lengthCode

public SaleDetail Find(int saleCode,int productCode, int sizeCode, int lengthCode)

{

return GetList().Find(x =>x.SaleCode==saleCode&& x.ProductCode == productCode && x.SizeCode == sizeCode && x.LengthCode == lengthCode);

}

//Update row in the table by the noun

public void UpDateRow(SaleDetail c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int saleCode, int productCode, int sizeCode, int lengthCode)

{

SaleDetail c = this.Find(saleCode,productCode,sizeCode,lengthCode);

this.GetList().Find(x => x.SaleCode == saleCode && x.ProductCode == productCode && x.SizeCode == sizeCode && x.LengthCode == lengthCode);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת SizeDB**

class SizeDB:GenerlDB

{

public SizeDB() : base("tbl\_sizes")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new Size(dr));

}

}

public List<Size> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new size

public void AddNew(Size s)

{

s.Dr = table.NewRow();

s.FillDataRow();

this.Add(s.Dr);

}

//Search for size by sizeCode

public Size Find(int sizeCode)

{

return GetList().Find(x => x.SizeCode == sizeCode);

}

//Update row in the table by the noun

public void UpDateRow(Size c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(int sizeCod)

{

Size c = this.Find(sizeCod);

this.GetList().Find(x => x.SizeCode == sizeCod);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

**מחלקת SupplierDB**

class SupplierDB:GenerlDB

{

public SupplierDB() : base("tbl\_suppliers")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new Supplier(dr));

}

}

public List<Supplier> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new Supplier

public void AddNew(Supplier s)

{

s.Dr = table.NewRow();

s.FillDataRow();

this.Add(s.Dr);

}

//Search for Supplier by suppliercode

public Supplier Find(int suppliercode)

{

return GetList().Find(x => x.SupplierCode == suppliercode);

}

//Delete row in the table by the key

public void DeleteRow(int code)

{

Supplier supplier =Find(code);

if(supplier!=null)

{

supplier.Dr.Delete();

Update();

}

}

//Update row in the table by the noun

public void UpDateRow(Supplier c)

{

c.FillDataRow();

this.Update();

}

}

**מחלקת SuppliersOrderDB**

class SuppliersOrderDB:GenerlDB

{

public SuppliersOrderDB() : base("tbl\_suppliers\_order")

{

}

public void DataTableToList()

{

foreach (DataRow dr in table.Rows)

{

lst.Add(new SuppliersOrder(dr));

}

}

public List<SuppliersOrder> GetList()

{

lst.Clear();

DataTableToList();

return lst;

}

//Add new SuppliersOrder

public void AddNew(SuppliersOrder s)

{

s.Dr = table.NewRow();

s.FillDataRow();

this.Add(s.Dr);

}

//Search for SuppliersOrder by orderPrice

public SuppliersOrder Find(double orderPrice)

{

return GetList().Find(x => x.OrderPrice == orderPrice);

}

//Update row in the table by the noun

public void UpDateRow(SuppliersOrder c)

{

c.FillDataRow();

this.Update();

}

//Delete row in the table by the key

public void DeleteRow(double orderPrice)

{

SuppliersOrder c = this.Find(orderPrice);

this.GetList().Find(x => x.OrderPrice == orderPrice);

if (c != null)

{

c.Dr.Delete();

this.Update();

}

}

}

מחלקות היחיד nameTableRow

כל מחלקה מהן, מטפלת ברשומה אחת מתוך טבלה מסוימת.

תכונותיה: משתנים המקבילים לשדות של אותה טבלה (גם בסוג הנתונים), אוביקט מסוג DataRow.

**מחלקת יחיד – עיר**

public class City

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int cityCode;

private string nameCity;

public DataRow Dr { get => dr; set => dr = value; }

public int CityCode { get => cityCode; set => cityCode = value; }

public string NameCity

{

get => nameCity; set

{

if (!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

nameCity = value;

}

}

public City()

{

}

public City(DataRow dr)//מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

cityCode = Convert.ToInt32(dr["city\_code"]);

NameCity = dr["city\_name"].ToString();

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

dr["city\_code"] = cityCode;

dr["city\_name"] = nameCity;

}

public override string ToString()

{

return nameCity;

}

}

**מחלקת יחיד – חברה**

public class Company

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private string companyName;

private int companyCode;

private int supplierCode;

public string CompanyName

{

get => companyName; set

{

if (!Validation.IsEnglish1(value))

throw new Exception("יש להקיש אותיות באנגלית");

companyName = value;

}

}

public int CompanyCode { get => companyCode; set => companyCode = value; }

public int SupplierCode { get => supplierCode; set => supplierCode = value; }

public DataRow Dr { get => dr; set => dr = value; }

public Company()

{

}

public Company(DataRow dr) מילוי תכונות העצם מתוך השורה/ /

{

this.Dr = dr;

companyCode = Convert.ToInt32(dr["company\_code"]);

supplierCode = Convert.ToInt32(dr["supplier\_code"]);

companyName = dr["company\_name"].ToString();

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

dr["company\_code"]=companyCode;

dr["company\_name"] = companyName;

dr["supplier\_code"] = supplierCode;

}

public override string ToString()

{

return companyName;

}

public Supplier SupplierOfCompany()

{

SupplierDB tblSupplier = new SupplierDB();

return tblSupplier.Find(supplierCode);

}

}

**מחלקת יחיד – לקוח**

public class Customer

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private string tz;

private string firstName;

private string lastName;

private string phone;

private string note;

private int citycode;

private string status;

public string Tz

{

get => tz; set

{

if (!Validation.CheckId(value))

throw new Exception("יש להקיש מספרים");

tz = value;

}

}

public string FirstName

{

get => firstName; set

{

if (!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

firstName = value;

}

}

public string LastName

{

get => lastName; set

{

if (!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

lastName = value;

}

}

public int Citycode { get => citycode; set => citycode = value; }

public DataRow Dr { get => dr; set => dr = value; }

public string Phone

{

get => phone; set

{

if (!Validation.IsPelepon(value))

throw new Exception("יש להקיש מספרים");

phone = value;

}

}

public string Note { get => note; set => note = value; }

public string Status { get => status; set => status = value; }

public Customer()

{

}

public Customer(DataRow dr)//מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

tz = dr["customer\_tz"].ToString();

firstName = dr["first\_name"].ToString();

lastName = dr["last\_name"].ToString();

phone = dr["phone"].ToString();

note= dr["note1"].ToString();

citycode =Convert.ToInt32 (dr["city\_code"]);

status= Convert.ToString(dr["status"]);

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

dr["customer\_tz"] = tz;

dr["first\_name"] = firstName;

dr["last\_name"] = lastName;

dr["phone"] = phone;

dr["note1"] = note;

dr["city\_code"] = citycode;

dr["status"] = status;

}

public override string ToString()

{

return firstName + " " + lastName;

}

public City CityOfC()

{

CityDB tblCity = new CityDB();

return tblCity.Find(citycode);

}

}

**מחלקת יחיד – פרטי תשלום**

public class DetailPayments

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int paymentCode;//קוד תשלום

private int paymentNumber;//מספר תשלומים

private int creditNumber;//מס אשראי

private int creditCw;//3 ספרות בגב הכרטיס

private int chequeNumber;//מספר המחאה

private int branchNumber;//מספר סניף

private string paymentWay;//אופן התשלום

private string customerTz;//מס זהות של הלקוח

private string bankNumber;//מספר בנק

private string userAccount;//מספר חשבון

private DateTime creditForce;//תוקף הכרטיס

public DataRow Dr { get => dr; set => dr = value; }

public int PaymentCode { get => paymentCode; set => paymentCode = value; }

public int PaymentNumber { get => paymentNumber; set => paymentNumber = value; }

public int CreditNumber { get => creditNumber; set => creditNumber = value; }

public int CreditCw { get => creditCw; set => creditCw = value; }

public int ChequeNumber { get => chequeNumber; set => chequeNumber = value; }

public int BranchNumber { get => branchNumber; set => branchNumber = value; }

public string PaymentWay { get => paymentWay; set => paymentWay = value; }

public string CustomerTz { get => customerTz; set => customerTz = value; }

public string BankNumber { get => bankNumber; set => bankNumber = value; }

public string UserAccount { get => userAccount; set => userAccount = value; }

public DateTime CreditForce { get => creditForce; set => creditForce = value; }

public DetailPayments()

{

}

public DetailPayments(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr; //אוביקט מסוג שורה השומר עותק של הרשומה

PaymentCode = Convert.ToInt32(dr["payment\_code"]);

PaymentNumber = Convert.ToInt32(dr["payment\_number"]);

CreditNumber = Convert.ToInt32(dr["credit\_number"]);

CreditCw = Convert.ToInt32(dr["credit\_cw"]);

ChequeNumber = Convert.ToInt32(dr["cheque\_number"]);

BranchNumber = Convert.ToInt32(dr[" branch\_number"]);

PaymentWay = dr["payment\_way"].ToString();

CustomerTz = dr["customer\_tz"].ToString();

BankNumber = dr["bank\_number"].ToString();

UserAccount = dr["user\_account"].ToString();

creditForce = Convert.ToDateTime(dr["credit\_force"]);

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

dr["payment\_code"] = PaymentCode;

dr["payment\_number"] = PaymentNumber;

dr["credit\_number"] = CreditNumber;

dr["credit\_cw"] = CreditCw;

dr["cheque\_number"] = ChequeNumber;

dr["branch\_number"] = BranchNumber;

dr["payment\_way"] = PaymentWay;

dr["customer\_tz"] = CustomerTz;

dr["bank\_number"] = BankNumber;

dr["user\_account"] = UserAccount;

dr["credit\_force"] = creditForce;

}

public Payments PaymentOfDrtail()

{

PaymentsDB tblPayments = new PaymentsDB();

return tblPayments.Find(paymentCode);

}

}

**מחלקת יחיד – מוסד**

public class Institution

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int institutionCode;

private string institutionName;

public int InstitutionCode { get => institutionCode; set => institutionCode = value; }

public string InstitutionName

{

get => institutionName; set

{

if (!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

institutionName = value;

}

}

public DataRow Dr { get => dr; set => dr = value; }

public Institution()

{

}

public Institution(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

institutionCode = Convert.ToInt32(dr["institution\_code"]);

institutionName = dr["institution\_name"].ToString();

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

dr["institution\_code"] = institutionCode;

dr["institution\_name"] = institutionName;

}

public override string ToString()

{

return institutionName;

}

}

**מחלקת יחיד – פרטים למוסד**

public class InstitutionProduct

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int productCode;

private int institutionCode;

public int ProductCode { get => productCode; set => productCode = value; }

public int InstitutionCode { get => institutionCode; set => institutionCode = value; }

public DataRow Dr { get => dr; set => dr = value; }

public InstitutionProduct()

{

}

public InstitutionProduct(DataRow dr)//מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

productCode = Convert.ToInt32(dr["product\_code"]);

institutionCode = Convert.ToInt32(dr["institution\_code"]);

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["product\_code"] = productCode;

Dr["institution\_code"] = institutionCode;

}

public ProductModels ProductModelsOfInstituation()

{

ProductModelsDB tblProductModels = new ProductModelsDB();

return tblProductModels.Find(productCode);

}

public Institution InstituationOfInstProduct()

{

InstitutionDB tblInstitution = new InstitutionDB();

return tblInstitution.Find(institutionCode);

}

}

**מחלקת יחיד – אורך**

public class Length

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int lengthCode;

private int lengthNum;

public Length()

{

}

public Length(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

lengthCode = Convert.ToInt32(dr["length\_code"]);

lengthNum = Convert.ToInt32(dr["length\_num"]);

}

public int LengthCode { get => lengthCode; set => lengthCode = value; }

public int LengthNum { get => lengthNum; set => lengthNum = value; }

public DataRow Dr { get => dr; set => dr = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["length\_code"] = lengthCode;

Dr["length\_num"] = lengthNum;

}

public override string ToString()

{

return lengthNum.ToString();

}

}

**מחלקת יחיד – פרטי הזמנה**

class OrderDetail

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int orderNumber;

private int productCode;

private int sizeCode;

private int lengthCode;

private int ordersAmount;

private int arrivingAmount;

private int finalPrice;

private string status;

public OrderDetail()

{

}

public OrderDetail(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

orderNumber = Convert.ToInt32(dr["order\_number"]);

productCode = Convert.ToInt32(dr["product\_code"]);

sizeCode = Convert.ToInt32(dr["size\_code"]);

lengthCode = Convert.ToInt32(dr["length\_code"]);

ordersAmount = Convert.ToInt32(dr["orders\_amount"]);

arrivingAmount = Convert.ToInt32(dr["arriving\_amount"]);

finalPrice = Convert.ToInt32(dr["final\_price"]);

status= dr["status"].ToString();

}

public DataRow Dr { get => dr; set => dr = value; }

public int OrderNumber { get => orderNumber; set => orderNumber = value; }

public int ProductCode { get => productCode; set => productCode = value; }

public int SizeCode { get => sizeCode; set => sizeCode = value; }

public int LengthCode { get => lengthCode; set => lengthCode = value; }

public int OrdersAmount { get => ordersAmount; set => ordersAmount = value; }

public int ArrivingAmount { get => arrivingAmount; set => arrivingAmount = value; }

public int FinalPrice { get => finalPrice; set => finalPrice = value; }

public string Status { get => status; set => status = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["order\_number"] = orderNumber;

Dr["product\_code"] = productCode;

Dr["size\_code"] = sizeCode;

Dr["length\_code"] = lengthCode;

Dr["orders\_amount"] = ordersAmount;

Dr["arriving\_amount"] = arrivingAmount;

Dr["final\_price"] = finalPrice;

Dr["status"] = status;

}

public SuppliersOrder SuppliersOrderOfDetail()

{

SuppliersOrderDB tblSupplierOrder = new SuppliersOrderDB();

return tblSupplierOrder.Find(orderNumber);

}

public ProductSize ProductSizeOfOrderDetail()

{

ProductSizeDB tblProductSize = new ProductSizeDB();

return tblProductSize.Find(productCode, sizeCode, lengthCode);

}

}

**מחלקת יחיד – תשלומים**

public class Payments

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int salesCode;

private int paymentCode;

private int paymentNumber;

public Payments()

{

}

public Payments(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

SalesCode = Convert.ToInt32(dr["sales\_code"]);

PaymentCode = Convert.ToInt32(dr["payment\_code"]);

PaymentNumber = Convert.ToInt32(dr["payment\_number"]);

}

public DataRow Dr { get => dr; set => dr = value; }

public int SalesCode { get => salesCode; set => salesCode = value; }

public int PaymentCode { get => paymentCode; set => paymentCode = value; }

public int PaymentNumber { get => paymentNumber; set => paymentNumber = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["sales\_code"] = SalesCode;

Dr["payment\_code"] = PaymentCode;

Dr["payment\_number"] = PaymentNumber;

}

}

**מחלקת יחיד – מוצרים**

public class ProductModels

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int productCode;

private int typeCode;

private int purchasePrice;

private int sallingPrice;

private int companyCode;

private string status;

private string productDescription;

public ProductModels()

{

}

public ProductModels(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

productCode = Convert.ToInt32(dr["product\_code"]);

CompanyCode = Convert.ToInt32(dr["company\_code"]);

typeCode = Convert.ToInt32(dr["type\_code"]);

purchasePrice = Convert.ToInt32(dr["purchase\_price"]);

sallingPrice = Convert.ToInt32(dr["salling\_price"]);

status = dr["status"].ToString();

ProductDescription = dr["type\_product"].ToString();

}

public int ProductCode { get => productCode; set => productCode = value; }

public int TypeCode { get => typeCode; set => typeCode = value; }

public int PurchasePrice { get => purchasePrice; set => purchasePrice = value; }

public int SallingPrice { get => sallingPrice; set => sallingPrice = value; }

public string Status { get => status; set => status = value; }

public DataRow Dr { get => dr; set => dr = value; }

public int CompanyCode { get => companyCode; set => companyCode = value; }

public string ProductDescription

{

get => productDescription; set

{

if(!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

productDescription = value;

}

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["product\_code"] =productCode;

Dr["type\_code"] = typeCode;

Dr["purchase\_price"] = purchasePrice;

Dr["salling\_price"] = sallingPrice;

Dr["company\_code"] = companyCode;

Dr["status"] = status;

Dr["type\_product"] = ProductDescription;

}

public Company CompanyOfP()

{

CompanyDB tblCompany = new CompanyDB();

return tblCompany.Find(companyCode);

}

public ProductType ProductOfModels()

{

ProductTypeDB tblProduct = new ProductTypeDB();

return tblProduct.Find(typeCode);

}

}

**מחלקת יחיד – מידות למוצר**

public class ProductSize

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int productCode;

private int sizeCode;

private int lengthCode;

private int stocksAmount;

private int status;

public ProductSize()

{

}

public ProductSize(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

ProductCode = Convert.ToInt32(dr["product\_code"]);

SizeCode = Convert.ToInt32(dr["size\_code"]);

LengthCode = Convert.ToInt32(dr["length\_code"]);

StocksAmount = Convert.ToInt32(dr["stocks\_amount"]);

Status = Convert.ToInt32(dr["status"]);

}

public DataRow Dr { get => dr; set => dr = value; }

public int ProductCode { get => productCode; set => productCode = value; }

public int SizeCode { get => sizeCode; set => sizeCode = value; }

public int LengthCode { get => lengthCode; set => lengthCode = value; }

public int StocksAmount { get => stocksAmount; set => stocksAmount = value; }

public int Status { get => status; set => status = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["product\_code"] = ProductCode;

Dr["size\_code"] = SizeCode;

Dr["length\_code"] = LengthCode;

Dr["stocks\_amount"] = StocksAmount;

Dr["status"] = Status;

}

public ProductModels ProductSizeOfModels()

{

ProductModelsDB tblProductModels = new ProductModelsDB();

return tblProductModels.Find(productCode);

}

public Size SizeOfProductSize()

{

SizeDB tblSize = new SizeDB();

return tblSize.Find(sizeCode);

}

public Length LengthOfProductSize()

{

LengthDB tblLength = new LengthDB();

return tblLength.Find(lengthCode);

}

}

**מחלקת יחיד – סוג מוצר**

public class ProductType

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int typeCode;

private string typeDescription;

public ProductType()

{

}

public ProductType(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

typeCode = Convert.ToInt32(dr["type\_code"]);

TypeDescription = dr["type\_description"].ToString();

}

public DataRow Dr { get => dr; set => dr = value; }

public int TypeCode { get => typeCode; set => typeCode = value; }

public string TypeDescription

{

get => typeDescription; set

{

if(!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

typeDescription = value;

}

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["type\_code"] = typeCode;

Dr["type\_description"] = TypeDescription;

}

public override string ToString()

{

return typeDescription;

}

}

**מחלקת יחיד – מכירה**

Public class Sale

{

DataRow dr;//אוביקט מסוג שורה השומר עותק של הרשומה

private string customerTz;

private string shippingMethod;

private string status;

private int salesCode;

private int finalPurchase;

private int shippingCost;

private DateTime date;

public Sale()

{

}

public Sale(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

CustomerTz = dr["customer\_tz"].ToString();

ShippingMethod = dr["shipping\_method"].ToString();

Status = dr["status"].ToString();

SalesCode = Convert.ToInt32(dr["sales\_code"]);

FinalPurchase = Convert.ToInt32(dr["final\_purchase"]);

ShippingCost = Convert.ToInt32(dr["shipping\_cost"]);

date = Convert.ToDateTime(dr["date"]);

}

public DataRow Dr { get => dr; set => dr = value; }

public string CustomerTz { get => customerTz; set => customerTz = value; }

public string ShippingMethod { get => shippingMethod; set => shippingMethod = value; }

public string Status { get => status; set => status = value; }

public int SalesCode { get => salesCode; set => salesCode = value; }

public int FinalPurchase { get => finalPurchase; set => finalPurchase = value; }

public int ShippingCost { get => shippingCost; set => shippingCost = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["customer\_tz"] = CustomerTz;

Dr["shipping\_method"] = ShippingMethod;

Dr["status"] = Status;

Dr["sales\_code"] = SalesCode;

Dr["shipping\_cost"] = ShippingCost;

Dr["final\_purchase"] = FinalPurchase;

Dr["date"] = date;

}

public override string ToString()

{

return salesCode.ToString();

}

}

**מחלקת יחיד – פרטי מכירה**

public class SaleDetail

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int saleCode;

private int productCode;

private int sizeCode;

private int lengthCode;

private int amount;

private int finalPrice;

private bool status;

public SaleDetail()

{

}

public SaleDetail(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

SaleCode = Convert.ToInt32(dr["sale\_ code"]);

ProductCode = Convert.ToInt32(dr["product\_code"]);

SizeCode = Convert.ToInt32(dr["size\_code"]);

LengthCode = Convert.ToInt32(dr["length\_code"]);

Amount = Convert.ToInt32(dr["amount"]);

FinalPrice = Convert.ToInt32(dr["final\_price"]);

// Status = Convert.ToBoolean(dr["status"]);

if (dr["Status"].Equals(true))

this.Status = true;

else

this.Status = false;

}

public DataRow Dr { get => dr; set => dr = value; }

public int SaleCode { get => saleCode; set => saleCode = value; }

public int ProductCode { get => productCode; set => productCode = value; }

public int SizeCode { get => sizeCode; set => sizeCode = value; }

public int LengthCode { get => lengthCode; set => lengthCode = value; }

public int Amount { get => amount; set => amount = value; }

public int FinalPrice { get => finalPrice; set => finalPrice = value; }

public bool Status { get => status; set => status = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["sale\_ code"] = SaleCode;

Dr["product\_code"] = ProductCode;

Dr["size\_code"] = SizeCode;

Dr["length\_code"] = LengthCode;

Dr["amount"] = Amount;

Dr["final\_price"] = FinalPrice;

Dr["status"] = Status;

}

public ProductSize SaleDetailOFSize()

{

ProductSizeDB tblProductSize = new ProductSizeDB();

return tblProductSize.Find(productCode, sizeCode, lengthCode);

}

}

**מחלקת יחיד – מידות**

public class Size

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private int sizeCode;

private string description;

public Size()

{

}

public Size(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

sizeCode = Convert.ToInt32(dr["size\_code"]);

description = dr["description"].ToString();

}

public DataRow Dr { get => dr; set => dr = value; }

public int SizeCode { get => sizeCode; set => sizeCode = value; }

public string Description

{

get => description; set

{

if (!Validation.IsNum(value))

throw new Exception("יש להקיש מספרים");

description = value;

}

}

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["size\_code"] = sizeCode;

Dr["description"] = description;

}

public override string ToString()

{

return description;

}

}

**מחלקת יחיד – ספקים**

public class Supplier

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private string supplierName;

private string phone;

private string note1;

private string mail;

private int citycode;

private int supplierCode;

private string status;

public Supplier()

{

}

public Supplier(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

Status = dr["status"].ToString();

SupplierCode = Convert.ToInt32(dr["supplier\_code"]);

Citycode = Convert.ToInt32(dr["city\_code"]);

SupplierName = dr["supplier\_name"].ToString();

Phone = dr["phone"].ToString();

Note1 = dr["note1"].ToString();

Mail = dr["mail"].ToString();

}

public DataRow Dr { get => dr; set => dr = value; }

public string SupplierName { get => supplierName; set

{

if (!Validation.IsHebrew(value))

throw new Exception("יש להקיש אותיות בעברית");

supplierName = value;

}

}

public string Phone

{

get => phone; set

{

if (!Validation.IsPelepon(value))

throw new Exception("יש להקיש מספרים");

phone = value;

}

}

public string Note1 { get => note1; set => note1 = value; }

public int Citycode { get => citycode; set => citycode = value; }

public string Status { get => status; set => status = value; }

public string Mail

{

get => mail; set

{

if (!Validation.IsMail(value))

throw new Exception("כתובת המייל אינה תקינה");

mail = value;

}

}

public int SupplierCode { get => supplierCode; set => supplierCode = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["supplier\_code"] = SupplierCode;

Dr["supplier\_name"] = SupplierName;

Dr["phone"] = Phone;

Dr["note1"] = Note1;

Dr["city\_code"] = Citycode;

Dr["status"] = Status;

Dr["mail"] = Mail;

}

public override string ToString()

{

return SupplierName;

}

public City CityOfSupplier()

{

CityDB tblCity = new CityDB();

return tblCity.Find(citycode);

}

}

**מחלקת יחיד – הזמנה מספק**

public class SuppliersOrder

{

DataRow dr; //אוביקט מסוג שורה השומר עותק של הרשומה

private string orderNumber;

private int supplierCode;

private double orderPrice;

private bool status;

private DateTime date;

public SuppliersOrder()

{

}

public SuppliersOrder(DataRow dr) //מילוי תכונות העצם מתוך השורה

{

this.Dr = dr;

orderNumber = dr["order\_number"].ToString();

supplierCode = Convert.ToInt32(dr["supplier\_code"]);

orderPrice =Convert.ToDouble(dr[" order\_price"]);

status = Convert.ToBoolean(dr["status"]);

date = Convert.ToDateTime(dr["order\_date"]);

}

public DataRow Dr { get => dr; set => dr = value; }

public string OrderNumber { get => orderNumber; set => orderNumber = value; }

public int SupplierCode { get => supplierCode; set => supplierCode = value; }

public double OrderPrice { get => orderPrice; set => orderPrice = value; }

public bool Status { get => status; set => status = value; }

public DateTime Date { get => date; set => date = value; }

public void FillDataRow()//מילוי השורה מתוך תכונות העצם

{

Dr["order\_number"] = orderNumber;

Dr["supplier\_code"] = supplierCode;

Dr["order\_price"] = orderPrice;

Dr["date"] = date;

Dr["status"] = status;

}

public Supplier SupplierOfOrders()

{

SupplierDB tblsupplier = new SupplierDB();

return tblsupplier.Find(supplierCode);

}

}

שכבת היישום GUI

שכבה המטפלת בממשק החיצוני של התוכנה.

טופס מסך הבית

public partial class FormHome : Form

{

public FormHome()

{

InitializeComponent();

}

private void butsupplier\_Click(object sender, EventArgs e)

{

pictureBox2.Visible = false;

panel4.Visible = true;

panbutton.Height = butsupplier.Height;

panbutton.Top = butsupplier.Top;

panel4.Controls.Clear();

FormSupplier s = new FormSupplier();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

SupplierDB tblSuppliers = new SupplierDB();

s.dgv\_supplier.DataSource = tblSuppliers.GetList().Where(x=>x.Status=="פעיל").Select(a => new

{

קוד\_ספק = a.SupplierCode,

הערות = a.Note1,

שם = a.SupplierName,

טלפון = a.Phone,

עיר = a.CityOfSupplier().NameCity

}

).OrderBy(x=>x.שם).ToList();

}

private void butcust\_Click(object sender, EventArgs e)

{

panbutton.Height = butcust.Height;

panbutton.Top = butcust.Top;

pictureBox2.Visible = false;

panel4.Visible = true;

panel4.Controls.Clear();

UserCustomer s = new UserCustomer();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

CustomerDB tblCustomers = new CustomerDB();

s.dgv\_customers.DataSource= tblCustomers.GetList().Where(x => x.Status == "פעיל").Select

(x => new

{

תעודת\_זהות=x.Tz,

שם\_פרטי=x.FirstName,

שם\_משפחה=x.LastName,

טלפון=x.Phone,עיר=x.CityOfC().NameCity,

הערות=x.Note

}

)

.OrderBy(x=>x.שם\_משפחה).ToList();

}

private void butprod\_Click(object sender, EventArgs e)

{

panbutton.Height = butprod.Height;

panbutton.Top = butprod.Top;

pictureBox2.Visible = false;

panel4.Visible = true;

panel4.Controls.Clear();

UserProducts s = new UserProducts();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

ProductModelsDB tblProducts = new ProductModelsDB();

s.dgv\_products.DataSource = tblProducts.GetList().Where(x => x.Status == "פעיל").Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק=x.CompanyOfP().SupplierOfCompany().SupplierName

}

).OrderBy(x => x.סוג\_מוצר).ToList();

}

private void butorder\_Click(object sender, EventArgs e)

{

panbutton.Height = butorder.Height;

panbutton.Top = butorder.Top;

panel4.Controls.Clear();

UserOrders s = new UserOrders();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

ProductModelsDB tblProducts = new ProductModelsDB();

}

private void butcompa\_Click(object sender, EventArgs e)

{

panbutton.Height = but\_salling.Height;

panbutton.Top = but\_salling.Top;

panel4.Controls.Clear();

UserSale s = new UserSale();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

private void btn\_management\_Click(object sender, EventArgs e)

{

pictureBox2.Visible = false;

panel4.Visible = true;

panel4.Controls.Clear();

UserManagement s = new UserManagement();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

private void btn\_orders\_arriving\_Click(object sender, EventArgs e)

{

panbutton.Height = btn\_orders\_arriving.Height;

panbutton.Top = btn\_orders\_arriving.Top;

}

}

טופס קוד ראשי

public partial class FormCode : Form

{

public FormCode()

{

InitializeComponent();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

if (e.KeyChar == 13)

{

if (textBox1.Text == "1234")

{

FormHome op = new FormHome();

this.Hide();

op.ShowDialog();

}

else

{

MessageBox.Show("הקוד שהוקש שגוי");

textBox1.Text = "";

}

}

}

}

טופס הבית

public partial class FormHome : Form

{

public FormHome()

{

InitializeComponent();

}

private void butsupplier\_Click(object sender, EventArgs e)

{

pictureBox2.Visible = false;

panel4.Visible = true;

panbutton.Height = butsupplier.Height;

panbutton.Top = butsupplier.Top;

panel4.Controls.Clear();

FormSupplier s = new FormSupplier();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

SupplierDB tblSuppliers = new SupplierDB();

s.dgv\_supplier.DataSource = tblSuppliers.GetList().Where(x=>x.Status=="פעיל").Select(a => new

{

קוד\_ספק = a.SupplierCode,

הערות = a.Note1,

שם = a.SupplierName,

טלפון = a.Phone,

עיר = a.CityOfSupplier().NameCity

}

).OrderBy(x=>x.שם).ToList();

}

private void butcust\_Click(object sender, EventArgs e)

{

panbutton.Height = butcust.Height;

panbutton.Top = butcust.Top;

pictureBox2.Visible = false;

panel4.Visible = true;

panel4.Controls.Clear();

UserCustomer s = new UserCustomer();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

CustomerDB tblCustomers = new CustomerDB();

s.dgv\_customers.DataSource= tblCustomers.GetList().Where(x => x.Status == "פעיל").Select

(x => new

{

תעודת\_זהות=x.Tz,

שם\_פרטי=x.FirstName,

שם\_משפחה=x.LastName,

טלפון=x.Phone,עיר=x.CityOfC().NameCity,

הערות=x.Note

}

)

.OrderBy(x=>x.שם\_משפחה).ToList();

}

private void butprod\_Click(object sender, EventArgs e)

{

panbutton.Height = butprod.Height;

panbutton.Top = butprod.Top;

pictureBox2.Visible = false;

panel4.Visible = true;

panel4.Controls.Clear();

UserProducts s = new UserProducts();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

ProductModelsDB tblProducts = new ProductModelsDB();

s.dgv\_products.DataSource = tblProducts.GetList().Where(x => x.Status == "פעיל").Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק=x.CompanyOfP().SupplierOfCompany().SupplierName

}

).OrderBy(x => x.סוג\_מוצר).ToList();

}

private void butorder\_Click(object sender, EventArgs e)

{

panbutton.Height = butorder.Height;

panbutton.Top = butorder.Top;

panel4.Controls.Clear();

UserOrders s = new UserOrders();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

ProductModelsDB tblProducts = new ProductModelsDB();

}

private void butcompa\_Click(object sender, EventArgs e)

{

panbutton.Height = but\_salling.Height;

panbutton.Top = but\_salling.Top;

panel4.Controls.Clear();

UserSale s = new UserSale();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

private void btn\_management\_Click(object sender, EventArgs e)

{

pictureBox2.Visible = false;

panel4.Visible = true;

panel4.Controls.Clear();

UserManagement s = new UserManagement();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

private void btn\_orders\_arriving\_Click(object sender, EventArgs e)

{

panbutton.Height = btn\_orders\_arriving.Height;

panbutton.Top = btn\_orders\_arriving.Top;

panel4.Controls.Clear();

UserOrdersArriving s = new UserOrdersArriving();

s.Location = new Point(148, 98);

panel4.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

}

טופס ספקים

public partial class FormSupplier : UserControl

{

SupplierDB tblSuppliers;

Supplier su;

public FormSupplier()

{

InitializeComponent();

tblSuppliers = new SupplierDB();

su = new Supplier();

cmb\_search.DataSource = tblSuppliers.GetList().Where(x => x.Status == "פעיל").ToList();

cmb\_search.SelectedIndex = -1;

}

private void FormSupplier\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Control c = this.Parent;

FormOneSupplier f1 = new FormOneSupplier();

f1.Location = new Point(148, 98);

this.Parent.Controls.Clear();

c.Controls.Add(f1);

f1.Dock = DockStyle.Fill;

}

private void btn\_raanen\_Click(object sender, EventArgs e)

{

dgv\_supplier.DataSource = tblSuppliers.GetList().Select(s => new

{

קוד\_ספק = s.SupplierCode,

הערות = s.Note1,

שם = s.SupplierName,

טלפון = s.Phone,

עיר = s.CityOfSupplier().NameCity,

סטטוס = s.Status

}

).OrderBy(s=>s.שם).ToList();

}

private void button4\_Click(object sender, EventArgs e)

{

if (dgv\_supplier.SelectedRows.Count > 0)

{

DialogResult s = MessageBox.Show("?האם ברצונך למחוק ספק זה", "אישור מחיקה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (s == DialogResult.Yes)

{

int code = Convert.ToInt32(dgv\_supplier.SelectedRows[0].Cells[0].Value.ToString());

su = tblSuppliers.Find(code);

su.Status="לא פעיל";

tblSuppliers.UpDateRow(su);

dgv\_supplier.DataSource = tblSuppliers.GetList().Select(v => new

{

קוד\_ספק = v.SupplierCode,

הערות = v.Note1,

שם = v.SupplierName,

טלפון = v.Phone,

עיר = v.CityOfSupplier().NameCity

}).OrderBy(v=>v.שם).ToList();

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (dgv\_supplier.SelectedRows.Count > 0)

{

int code = Convert.ToInt32(dgv\_supplier.SelectedRows[0].Cells[0].Value.ToString());

su = tblSuppliers.Find(code);//חיפוש עצם הלקוח עפ תז שנבחרה בגריד

FormOneSupplier f = new FormOneSupplier(su);//שולחת את העצם שמצא לטופס

f.ReadOnly();

f.Location = new Point(148, 98);

Control c = this.Parent;

this.Parent.Controls.Clear();

c.Controls.Add(f);

f.Dock = DockStyle.Fill;

dgv\_supplier.DataSource = tblSuppliers.GetList().Select(s => new

{

קוד\_ספק = s.SupplierCode,

הערות = s.Note1,

שם = s.SupplierName,

טלפון = s.Phone,

עיר = s.CityOfSupplier().NameCity,

סטטוס = s.Status

}).ToList();

}

}

private void change\_sup\_Click(object sender, EventArgs e)

{

if (dgv\_supplier.SelectedRows.Count > 0)

{

int code = Convert.ToInt32(dgv\_supplier.SelectedRows[0].Cells[0].Value.ToString());

su = tblSuppliers.Find(code);//חיפוש עצם הלקוח עפ תז שנבחרה בגריד

FormOneSupplier f = new FormOneSupplier(su);//שולחת את העצם שמצא לטופס

f.Location = new Point(148, 98);

Control c = this.Parent;

this.Parent.Controls.Clear();

c.Controls.Add(f);

f.Dock = DockStyle.Fill;

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

dgv\_supplier.DataSource = tblSuppliers.GetList().Where(x => x.ToString() == cmb\_search.Text).Select(s => new

{

קוד\_ספק = s.SupplierCode,

הערות = s.Note1,

שם = s.SupplierName,

טלפון = s.Phone,

עיר = s.CityOfSupplier().NameCity,

סטטוס = s.Status

}).ToList();

}

}

טופס הוספת ספק חדש

public partial class FormOneSupplier : UserControl

{

CityDB tblCity;

SupplierDB tblSuppliers;

Supplier s;

bool flagUpdate = false;

public FormOneSupplier()

{

InitializeComponent();

tblSuppliers = new SupplierDB();

tblCity = new CityDB();

cmb\_city.DataSource = tblCity.GetList();

cmb\_city.SelectedIndex = -1;

}

public FormOneSupplier(Supplier sup):this()

{

this.s = sup;

Fill(sup);

flagUpdate = true;

}

public void Fill(Supplier s)

{

Supplier sup = s;

txt\_mail.Text = s.Mail;

txt\_name.Text = s.SupplierName;

txt\_phone.Text = s.Phone;

txt\_note1.Text = s.Note1;

cmb\_city.SelectedItem = s.CityOfSupplier();

cmb\_city.Text = s.CityOfSupplier().ToString();

}

public void ReadOnly()

{

btn\_save.Visible = false;

txt\_mail.ReadOnly = true;

txt\_name.ReadOnly = true;

txt\_phone.ReadOnly = true;

txt\_note1.ReadOnly = true;

cmb\_city.Enabled = false;

btn\_save.Visible = false;

}

private void label4\_Click(object sender, EventArgs e)

{

}

public bool CreateField(Supplier s)

{

bool flagOk = true;

errorProvider1.Clear();

try

{

if (txt\_name.Text == "")

throw new Exception("הקש שם");

if (txt\_name.Text.Length < 2)

throw new Exception("הקש שם ");

s.SupplierName = txt\_name.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_name, ex.Message);

flagOk = false;

}

try

{

if (txt\_phone.Text == "")

throw new Exception("הקש מספר טלפון");

s.Phone = txt\_phone.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_phone, ex.Message);

flagOk = false;

}

try

{

if (cmb\_city.SelectedIndex < 0)

throw new Exception("בחר עיר");

s.Citycode = ((City)cmb\_city.SelectedItem).CityCode;

}

catch (Exception ex)

{

errorProvider1.SetError(cmb\_city, ex.Message);

flagOk = false;

}

try

{

if (txt\_mail.Text == "")

throw new Exception("הקש כתובת מייל");

s.Mail=txt\_mail.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_mail, ex.Message);

flagOk = false;

}

if(!flagUpdate)//אם אתה במצב הוספה צור קוד חדש

s.SupplierCode= tblSuppliers.GetNextKey();

s.Note1 = txt\_note1.Text;

s.Status = "פעיל";

return flagOk;

}

private void button1\_Click(object sender, EventArgs e)

{

if (flagUpdate)

{

if (CreateField(s))

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן ספק זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

btn\_save.Enabled = false;

tblSuppliers.UpDateRow(s);

OpenUc();

}

}

}

else

{

Supplier s = new Supplier();

if (CreateField(s))

{

DialogResult d = MessageBox.Show("?האם בצונך להוסיף ספק זה", "אישור הוספה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

btn\_save.Enabled = false;

tblSuppliers.AddNew(s);

OpenUc();

}

}

}

}

private void OpenUc()

{

FormSupplier f1 = new FormSupplier();

f1.Location = new Point(148, 98);

this.Controls.Clear();

Controls.Add(f1);

f1.Dock = DockStyle.Fill;

f1.dgv\_supplier.DataSource = tblSuppliers.GetList().

Select(v => new { קוד\_ספק = v.SupplierCode,

הערות = v.Note1, שם = v.SupplierName,

טלפון = v.Phone,

עיר = v.CityOfSupplier().NameCity,

סטטוס = v.Status })

.OrderBy(v=>v.שם).ToList();

}

private void txt\_phone\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_name\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void txt\_note1\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

{

טופס לקוחות

public partial class UserCustomer : UserControl

{

CustomerDB tblCustomers;

Customer cus;

public UserCustomer()

{

InitializeComponent();

tblCustomers = new CustomerDB();

cus = new Customer();

}

private void btn\_add\_cus\_Click(object sender, EventArgs e)

{

Control c = this.Parent;

UserSingleCustomer f1 = new UserSingleCustomer();

f1.Location = new Point(148, 98);

this.Parent.Controls.Clear();

c.Controls.Add(f1);

f1.Dock = DockStyle.Fill;

}

private void btn\_change\_cus\_Click(object sender, EventArgs e)

{

if (dgv\_customers.SelectedRows.Count > 0)

{

string tz = dgv\_customers.SelectedRows[0].Cells[0].Value.ToString();

cus = tblCustomers.Find(tz);//חיפוש עצם הלקוח עפ תז שנבחרה בגריד

UserSingleCustomer f = new UserSingleCustomer(cus);//שולחת את העצם שמצא לטופס

f.Location = new Point(148, 98);

Control c = this.Parent;

this.Parent.Controls.Clear();

c.Controls.Add(f);

f.Dock = DockStyle.Fill;

}

}

private void btn\_show\_cus\_Click(object sender, EventArgs e)

{

if (dgv\_customers.SelectedRows.Count > 0)

{

string tz = dgv\_customers.SelectedRows[0].Cells[0].Value.ToString();

cus = tblCustomers.Find(tz);//חיפוש עצם הלקוח עפ תז שנבחרה בגריד

UserSingleCustomer f = new UserSingleCustomer(cus);//שולחת את העצם שמצא לטופס

f.ReadOnly();

f.Location = new Point(148, 98);

Control c = this.Parent;

this.Parent.Controls.Clear();

c.Controls.Add(f);

f.Dock = DockStyle.Fill;

dgv\_customers.DataSource = tblCustomers.GetList().Select(x => new

{

תעודת\_זהות = x.Tz,

שם\_פרטי = x.FirstName,

שם\_משפחה = x.LastName,

טלפון = x.Phone,

עיר = x.CityOfC().NameCity,

}).OrderBy(x=>x.שם\_משפחה).ToList();

}

}

private void btn\_reanun\_cus\_Click(object sender, EventArgs e)

{

dgv\_customers.DataSource = tblCustomers.GetList().Select(x => new

{

תעודת\_זהות = x.Tz,

שם\_פרטי = x.FirstName,

שם\_משפחה = x.LastName,

טלפון = x.Phone,

עיר = x.CityOfC().NameCity,

}).OrderBy(x => x.שם\_משפחה).ToList();

}

private void btn\_home2\_Click(object sender, EventArgs e)

{

if (dgv\_customers.SelectedRows.Count > 0)

{

DialogResult s = MessageBox.Show("?האם ברצונך למחוק לקוח זה", "אישור מחיקה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (s == DialogResult.Yes)

{

string tz = dgv\_customers.SelectedRows[0].Cells[0].Value.ToString();

cus = tblCustomers.Find(tz);

cus.Status = "לא פעיל";

tblCustomers.UpDateRow(cus);

dgv\_customers.DataSource = tblCustomers.GetList().Select(x => new

{

תעודת\_זהות = x.Tz,

שם\_פרטי = x.FirstName,

שם\_משפחה = x.LastName,

טלפון = x.Phone,

עיר = x.CityOfC().NameCity

}).OrderBy(x => x.שם\_משפחה).ToList();

}

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if(txt\_search.Text!=" ")

dgv\_customers.DataSource= tblCustomers.GetList().Where(x => x.LastName.ToString().StartsWith(txt\_search.Text)).Select(x => new

{

תעודת\_זהות = x.Tz,

שם\_פרטי = x.FirstName,

שם\_משפחה = x.LastName,

טלפון = x.Phone,

עיר = x.CityOfC().NameCity

}).OrderBy(x => x.שם\_משפחה).ToList();

}

}

טופס הוספת לקוח

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

public partial class UserSingleCustomer : UserControl

{

CityDB tblCity;

CustomerDB tblCustomers;

Customer c;

bool flagUpdate = false;

int num = 0;

public UserSingleCustomer()

{

InitializeComponent();

tblCustomers = new CustomerDB();

tblCity = new CityDB();

cmb\_city.DataSource = tblCity.GetList();

cmb\_city.SelectedIndex = -1;

}

public UserSingleCustomer(string customerId,int num):this()

{

txt\_tz.Text = customerId;

txt\_tz.ReadOnly = true;

this.num = num;

}

public UserSingleCustomer(Customer cu) : this()

{

this.c = cu;

Fill(cu);

flagUpdate = true;

}

public void Fill(Customer cu)

{

Customer c = cu;

txt\_tz.Text = c.Tz;

txt\_fname.Text = c.FirstName;

txt\_lname.Text = c.LastName;

txt\_phone.Text = c.Phone;

txt\_note2.Text = c.Note;

cmb\_city.SelectedItem = c.CityOfC();

cmb\_city.Text = c.CityOfC().ToString();

}

public void ReadOnly()

{

btn\_save.Visible = false;

txt\_tz.ReadOnly = true;

txt\_fname.ReadOnly = true;

txt\_lname.ReadOnly = true;

txt\_phone.ReadOnly = true;

txt\_note2.ReadOnly = true;

cmb\_city.Enabled = false;

}

public bool CreateField(Customer s)

{

bool flagOk = true;

errorProvider1.Clear();

try

{

if (txt\_tz.Text == "")

throw new Exception("הקש תעודת זהות");

s.Tz = txt\_tz.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_tz, ex.Message);

flagOk = false;

}

try

{

if (txt\_fname.Text == "")

throw new Exception("הקש שם פרטי");

if(txt\_fname.Text.Length<2)

throw new Exception("הקש שם פרטי");

s.FirstName = txt\_fname.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_fname, ex.Message);

flagOk = false;

}

try

{

if (txt\_phone.Text == "")

throw new Exception("הקש מספר טלפון");

s.Phone = txt\_phone.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_phone, ex.Message);

flagOk = false;

}

try

{

if (cmb\_city.SelectedIndex < 0)

throw new Exception("בחר עיר");

s.Citycode = ((City)cmb\_city.SelectedItem).CityCode;

}

catch (Exception ex)

{

errorProvider1.SetError(cmb\_city, ex.Message);

flagOk = false;

}

try

{

if (txt\_mail.Text == "")

throw new Exception("הקש כתובת מייל");

s.Mail = txt\_mail.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_mail, ex.Message);

flagOk = false;

}

try

{

if (txt\_lname.Text == "")

throw new Exception("הקש שם משפחה");

if (txt\_fname.Text.Length < 2)

throw new Exception("הקש שם משפחה");

s.LastName = txt\_lname.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_lname, ex.Message);

flagOk = false;

}

s.Note = txt\_note2.Text;

s.Status="פעיל";

return flagOk;

}

private void btn\_save\_Click(object sender, EventArgs e)

{

if (flagUpdate)

{

if (CreateField(c))

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן לקוח זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

btn\_save.Enabled = false;

tblCustomers.UpDateRow(c);

OpenUc();

txt\_tz.Text = "";

}

}

}

else

{

Customer s = new Customer();

if (this.tblCustomers.Find(txt\_tz.Text) != null)

MessageBox.Show("ת.ז זו קיימת במערכת");

else

{

if (CreateField(s))

{

DialogResult d = MessageBox.Show("?האם בצונך להוסיף לקוח זה", "אישור הוספה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

btn\_save.Enabled = false;

tblCustomers.AddNew(s);

}

if (num == 1)

{

Control c = this.Parent;

CloseSingleCustomer();

}

else

OpenUc();

}

}

}

}

private void CloseSingleCustomer()

{

foreach (Control item in this.Parent.Controls)

{

if (item is UserSale)

{

item.Visible = true;

}

}

this.Parent.Controls.Remove(this);

}

private void OpenUc()

{

UserCustomer f1 = new UserCustomer();

f1.Location = new Point(148, 98);

this.Controls.Clear();

Controls.Add(f1);

f1.Dock = DockStyle.Fill;

f1.dgv\_customers.DataSource = tblCustomers.GetList().Select(x => new { תעודת\_זהות = x.Tz, שם\_פרטי = x.FirstName, שם\_משפחה = x.LastName, טלפון = x.Phone, עיר = x.CityOfC().NameCity}).ToList();

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void cmb\_city\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void txt\_note2\_TextChanged(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void txt\_phone\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_tz\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_fname\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void txt\_lname\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void txt\_note2\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

}

}

טופס ניהול מערכת

public partial class UserManagement : UserControl

{

ProductTypeDB tblProductType;

CityDB tblCity;

LengthDB tblLength;

CompanyDB tblCompany;

InstitutionDB tblInstitution;

SizeDB tblSize;

SupplierDB tblS;

Supplier s;

City c;

Institution i;

Bll.Size ss;

Company cc;

ProductType p;

Length l;

bool flagUpdate = false;

public UserManagement()

{

InitializeComponent();

tblProductType = new ProductTypeDB();

tblCity = new CityDB();

tblLength = new LengthDB();

tblCompany = new CompanyDB();

tblInstitution = new InstitutionDB();

tblSize = new SizeDB();

tblS = new SupplierDB();

s = new Supplier();

c = new City();

i = new Institution();

ss = new Bll.Size();

cc = new Company();

p = new ProductType();

l = new Length();

panel1.Visible = false;

panel4.Visible = false;

panel5.Visible = false;

panel6.Visible = false;

panel7.Visible = false;

panel8.Visible = false;

cmb\_city.DataSource = tblCity.GetList().OrderBy(x=>x.NameCity).ToList();

cmb\_city.SelectedIndex = -1;

cmb\_type\_p.DataSource = tblProductType.GetList();

cmb\_type\_p.SelectedIndex = -1;

cmb\_length.DataSource = tblLength.GetList();

cmb\_length.SelectedIndex = -1;

cmb\_company.DataSource = tblCompany.GetList();

cmb\_company.SelectedIndex = -1;

cmb\_institution.DataSource = tblInstitution.GetList();

cmb\_institution.SelectedIndex = -1;

cmb\_sizes.DataSource = tblSize.GetList();

cmb\_sizes.SelectedIndex = -1;

cmb\_supplier.DataSource = tblS.GetList();

cmb\_supplier.SelectedIndex = -1;

}

private void btn\_home7\_Click(object sender, EventArgs e)

{

panel7.Visible = true;

}

private void btn\_add\_Click(object sender, EventArgs e)

{

panel1.Visible = true;

}

private void btn\_home1\_Click(object sender, EventArgs e)

{

panel4.Visible = true;

}

private void btn\_home2\_Click(object sender, EventArgs e)

{

panel5.Visible = true;

flagUpdate = false;

}

private void btn\_home6\_Click(object sender, EventArgs e)

{

panel6.Visible = true;

}

private void btn\_home8\_Click(object sender, EventArgs e)

{

panel8.Visible = true;

}

public bool CreateType(ProductType t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (txt\_new\_p.Text == "")

throw new Exception("יש להקיש סוג מוצר");

if (!Validation.IsHebrew(txt\_new\_p.Text))

throw new Exception("יש להקיש אותיות בעברית");

t.TypeDescription = txt\_new\_p.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_new\_p, ex.Message);

flag = false;

}

if(!flagUpdate)

t.TypeCode = tblProductType.GetNextKey();

return flag;

}

private void btn\_save\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (ProductType item in tblProductType.GetList())

{

if (item.ToString().Equals(txt\_new\_p.Text))

{

MessageBox.Show("סוג זה קיים במערכת");

flag1 = false;

}

}

if (flagUpdate&&flag1)

{

if (CreateType(p))

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן סוג זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

tblProductType.UpDateRow(p);

cmb\_type\_p.DataSource = null;

cmb\_type\_p.DataSource = tblProductType.GetList();

cmb\_type\_p.SelectedIndex = -1;

txt\_new\_p.Text = "";

panel1.Visible = false;

}

}

}

else

{

if (flag1)

{

ProductType p = new ProductType();

if (CreateType(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף סוג זה ", "הוספת מוצר", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblProductType.AddNew(p);

cmb\_type\_p.DataSource = null;

cmb\_type\_p.DataSource = tblProductType.GetList();

cmb\_type\_p.Text = p.ToString();

cmb\_type\_p.DataSource = tblProductType.GetList();

txt\_new\_p.Text = "";

}

}

}

else

{

txt\_new\_p.Text = "";

panel1.Visible = false;

}

}

}

public bool CreateCompany(Company t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (txt\_company.Text == "")

throw new Exception("יש להקיש שם חברה");

if (!Validation.IsHebrew(txt\_company.Text))

throw new Exception("יש להקיש אותיות בעברית");

t.CompanyName = txt\_company.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_company, ex.Message);

flag = false;

}

try

{

if(cmb\_supplier.SelectedIndex<0)

throw new Exception("הקש שם ספק");

t.SupplierCode = ((Supplier)cmb\_supplier.SelectedItem).SupplierCode; ;

}

catch (Exception ex)

{

errorProvider1.SetError(cmb\_supplier, ex.Message);

flag = false;

}

if(!flagUpdate)

t.CompanyCode = tblCompany.GetNextKey();

return flag;

}

private void button2\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (Company item in tblCompany.GetList())

{

if (item.ToString().Equals(txt\_company.Text))

{

MessageBox.Show("חברה זו קיימת במערכת");

flag1 = false;

}

}

if (flagUpdate&&flag1)

{

if (CreateCompany(cc))

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן פרטים אלו", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

tblCompany.UpDateRow(cc);

cmb\_company.DataSource = null;

cmb\_company.DataSource = tblCompany.GetList();

cmb\_company.SelectedIndex = -1;

txt\_company.Text = "";

panel4.Visible = false;

}

}

}

else

{

if (flag1)

{

Company p = new Company();

if (CreateCompany(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף שם זה ", "הוספת חברה", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblCompany.AddNew(p);

cmb\_company.DataSource = null;

cmb\_company.DataSource = tblCompany.GetList();

cmb\_company.Text = p.ToString();

cmb\_company.DataSource = tblCompany.GetList();

txt\_company.Text = "";

cmb\_supplier.SelectedIndex = -1;

}

}

}

else

{

txt\_company.Text = "";

panel4.Visible = false;

}

}

}

public bool CreateLength(Length l)

{

bool flagOk = true;

errorProvider1.Clear();

try

{

if (txt\_length.Text == "")

throw new Exception("הקש אורך ");

if (!Bll.Validation.IsNum(txt\_length.Text))

throw new Exception("יש להקיש ספרות");

l.LengthNum =Convert.ToInt32( txt\_length.Text);

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_length, ex.Message);

flagOk = false;

}

if(!flagUpdate)

l.LengthCode = tblLength.GetNextKey();

return flagOk;

}

private void button4\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (Length item in tblLength.GetList())

{

if (item.ToString().Equals(txt\_length.Text))

{

MessageBox.Show("אורך זה קיים במערכת");

flag1 = false;

}

}

if (flagUpdate&&flag1)

{

if (CreateLength(l))

{

foreach (Length item in tblLength.GetList())

{

if (item.ToString().Equals(txt\_length.Text))

{

MessageBox.Show("אורך זה קיים במערכת");

flag1 = false;

}

}

if (flag1)//אם האורך לא קיים במערכת

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן אורך זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

l.LengthNum =Convert.ToInt32( txt\_length.Text);

tblLength.UpDateRow(l);

cmb\_length.DataSource = null;

cmb\_length.DataSource = tblLength.GetList();

cmb\_length.SelectedIndex = -1;

txt\_length.Text = "";

panel5.Visible = false;

}

}

}

}

else

{

if (flag1)

{

Length p = new Length();

if (CreateLength(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף אורך זה ", "הוספת אורך", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblLength.AddNew(p);

cmb\_length.DataSource = null;

cmb\_length.DataSource = tblLength.GetList();

cmb\_length.Text = p.ToString();

cmb\_length.DataSource = tblLength.GetList();

txt\_length.Text = "";

}

}

}

else

{

txt\_length.Text = "";

panel1.Visible = false;

}

}

}

public bool CreateCity(City t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (txt\_city.Text == "")

throw new Exception("יש להקיש שם עיר");

if (!Validation.IsHebrew(txt\_city.Text))

throw new Exception("יש להקיש אותיות בעברית");

t.NameCity = txt\_city.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_city, ex.Message);

flag = false;

}

if(!flagUpdate)

t.CityCode = tblCity.GetNextKey();

return flag;

}

private void button10\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (City item in tblCity.GetList())

{

if (item.ToString().Equals(txt\_city.Text))

{

MessageBox.Show("עיר זו קיימת במערכת");

flag1 = false;

}

}

//עידכון

if (flagUpdate)

{

if (CreateCity(c)&& flag1)

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן עיר זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

tblCity.UpDateRow(c);

cmb\_city.DataSource = null;

cmb\_city.DataSource = tblCity.GetList();

cmb\_city.SelectedIndex = -1;

txt\_city.Text = "";

panel8.Visible = false;

}

}

}

else

{

//הוספה

if (flag1)

{

City p = new City();

if (CreateCity(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף עיר זאת ", "הוספת עיר", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblCity.AddNew(p);

cmb\_city.DataSource = null;

cmb\_city.DataSource = tblCity.GetList();

cmb\_city.Text = p.ToString();

cmb\_city.DataSource = tblCity.GetList();

txt\_city.Text = "";

}

}

}

else

{

txt\_city.Text = "";

panel8.Visible = false;

}

}

}

public bool CreateInstituation(Institution t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (txt\_instituation.Text == "")

throw new Exception("יש להקיש שם עיר");

if (!Validation.IsHebrew(txt\_instituation.Text))

throw new Exception("יש להקיש אותיות בעברית");

t.InstitutionName = txt\_instituation.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_instituation, ex.Message);

flag = false;

}

if(!flagUpdate)

t.InstitutionCode = tblInstitution.GetNextKey();

return flag;

}

private void button8\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (Institution item in tblInstitution.GetList())

{

if (item.ToString().Equals(txt\_instituation.Text))

{

MessageBox.Show("מוסד זה קיים במערכת");

flag1 = false;

}

}

if (flagUpdate&&flag1)

{

if (CreateInstituation(i))

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן מוסד זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

tblInstitution.UpDateRow(i);

cmb\_institution.DataSource = null;

cmb\_institution.DataSource = tblInstitution.GetList();

cmb\_institution.SelectedIndex = -1;

txt\_instituation.Text = "";

panel7.Visible = false;

}

}

}

else

{

if (flag1)

{

Institution p = new Institution();

if (CreateInstituation(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף מוסד זה ", "הוספת מוסד", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblInstitution.AddNew(p);

this.Visible = false;

cmb\_institution.DataSource = null;

cmb\_institution.DataSource = tblInstitution.GetList();

cmb\_institution.Text = p.ToString();

cmb\_institution.DataSource = tblInstitution.GetList();

txt\_instituation.Text = "";

}

}

}

else

{

txt\_instituation.Text = "";

panel7.Visible = false;

}

}

}

public bool CreateSize(Bll.Size t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (txt\_size.Text == "")

throw new Exception("שדה חובה");

if (!Validation.IsNum(txt\_size.Text))

throw new Exception("הקש מספרים");

t.Description = txt\_size.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_size, ex.Message);

flag = false;

}

if(!flagUpdate)

t.SizeCode = tblSize.GetNextKey();

return flag;

}

private void button6\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (Bll.Size item in tblSize.GetList())

{

if (item.ToString().Equals(txt\_size.Text))

{

MessageBox.Show("מידה זו קיימת במערכת");

flag1 = false;

}

}

if (flagUpdate&&flag1)

{

if (CreateSize(ss))

{

DialogResult d = MessageBox.Show("?האם בצונך לעדכן מידה זו", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

tblSize.UpDateRow(ss);

cmb\_sizes.DataSource = null;

cmb\_sizes.DataSource = tblSize.GetList();

cmb\_sizes.SelectedIndex = -1;

txt\_size.Text = "";

panel6.Visible = false;

}

}

}

else

{

if (flag1)

{

Bll.Size p = new Bll.Size();

if (CreateSize(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף מידה זו ", "הוספת מידה", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblSize.AddNew(p);

cmb\_sizes.DataSource = null;

cmb\_sizes.DataSource = tblSize.GetList();

cmb\_sizes.Text = p.ToString();

cmb\_sizes.DataSource = tblSize.GetList();

txt\_size.Text = "";

}

}

}

else

{

txt\_size.Text = "";

panel6.Visible = false;

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

txt\_company.Text = "";

cmb\_supplier.SelectedIndex = -1;

panel4.Visible = false;

}

private void btn\_cancel\_Click(object sender, EventArgs e)

{

txt\_new\_p.Text = "";

cmb\_type\_p.SelectedIndex = -1;

panel1.Visible = false;

}

private void button3\_Click(object sender, EventArgs e)

{

txt\_length.Text = "";

cmb\_length.SelectedIndex = -1;

panel5.Visible = false;

}

private void button5\_Click(object sender, EventArgs e)

{

txt\_size.Text = "";

cmb\_sizes.SelectedIndex = -1;

panel6.Visible = false;

}

private void button7\_Click(object sender, EventArgs e)

{

txt\_instituation.Text = "";

cmb\_institution.SelectedIndex = -1;

panel7.Visible = false;

}

private void button9\_Click(object sender, EventArgs e)

{

txt\_city.Text = "";

cmb\_city.SelectedIndex = -1;

panel8.Visible = false;

}

private void btn\_home11\_Click(object sender, EventArgs e)

{

if (cmb\_city.SelectedIndex>-1)

{

int code = ((City)cmb\_city.SelectedItem).CityCode;

c = tblCity.Find(code);

panel8.Visible = true;

txt\_city.Text = c.NameCity;

flagUpdate = true;

}

else

MessageBox.Show("בחר עיר שברצונך לעדכן");

}

private void btn\_home12\_Click(object sender, EventArgs e)

{

if (cmb\_institution.SelectedIndex > -1)

{

int code = ((Institution)cmb\_institution.SelectedItem).InstitutionCode;

i = tblInstitution.Find(code);

panel7.Visible = true;

txt\_instituation.Text = i.InstitutionName;

flagUpdate = true;

}

else

MessageBox.Show("בחר מוסד שברצונך לעדכן");

}

private void btn\_home14\_Click(object sender, EventArgs e)

{

if (cmb\_sizes.SelectedIndex > -1)

{

int code = ((Bll.Size)cmb\_sizes.SelectedItem).SizeCode;

ss = tblSize.Find(code);

panel6.Visible = true;

txt\_size.Text = ss.Description;

flagUpdate = true;

}

else

MessageBox.Show("בחר מידה שברצונך לעדכן");

}

private void btn\_home10\_Click(object sender, EventArgs e)

{

if (cmb\_company.SelectedIndex > -1)

{

cc = ((Company)cmb\_company.SelectedItem);

panel4.Visible = true;

txt\_company.Text = cc.CompanyName;

cmb\_supplier.SelectedItem = cc.SupplierOfCompany();

cmb\_supplier.Text = cc.SupplierOfCompany().SupplierName;

flagUpdate = true;

}

else

MessageBox.Show("בחר חברה שברצונך לעדכן");

}

private void btn\_home9\_Click(object sender, EventArgs e)

{

if (cmb\_type\_p.SelectedIndex > -1)

{

int code = ((ProductType)cmb\_type\_p.SelectedItem).TypeCode;

p = tblProductType.Find(code);

panel1.Visible = true;

txt\_new\_p.Text = p.TypeDescription;

flagUpdate = true;

}

else

MessageBox.Show("בחר סוג שברצונך לעדכן");

}

private void btn\_home13\_Click(object sender, EventArgs e)

{

if (cmb\_length.SelectedIndex > -1)

{

int code = ((Length)cmb\_length.SelectedItem).LengthCode;

l = tblLength.Find(code);

panel5.Visible = true;

txt\_length.Text =l.LengthNum.ToString();

flagUpdate = true;

}

else

MessageBox.Show("בחר אורך שברצונך לעדכן");

}

private void txt\_length\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_size\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_new\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void txt\_city\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void txt\_instituation\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

}

טופס הזמנות

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

OrderDetail od;

SuppliersOrder sO;

SuppliersOrderDB tblSuplierOrder;

SupplierDB tblSuppliers;

CompanyDB tblCompany;

ProductModelsDB tblProductModels;

ProductSizeDB tblProductSize;

OrderDetailDB tblOrderDetail;

Supplier s;

Company c;

ProductModels p;

ProductSize ps;

bool flag = false;

public UserOrders()

{

InitializeComponent();

tblSuppliers = new SupplierDB();

tblCompany = new CompanyDB();

tblSuplierOrder = new SuppliersOrderDB();

tblProductModels = new ProductModelsDB();

tblProductSize = new ProductSizeDB();

tblOrderDetail = new OrderDetailDB();

tblSuplierOrder = new SuppliersOrderDB();

s = new Supplier();

c = new Company();

p = new ProductModels();

ps = new ProductSize();

cmb\_search.DataSource = tblSuppliers.GetList().Where(x => x.Status == "פעיל").ToList();

cmb\_search.SelectedIndex = -1;

flag = true;

btn\_continue.Visible = false;

btn\_save.Visible = false;

btn\_remove.Visible = false;

}

private void cmb\_search\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmb\_search.SelectedIndex > -1)

{

s = ((Supplier)cmb\_search.SelectedItem);

listbox\_company.DataSource=tblCompany.

GetList().Where(x => x.SupplierCode == s.SupplierCode).

Select(x => x.CompanyName.ToString()).ToList();

listbox\_company.SelectedIndex = -1;

}

else

listbox\_company.DataSource=null;

}

private void company\_SelectedIndexChanged(object sender, EventArgs e)

{

if (listbox\_company.SelectedItems.Count > 0)

{

c = tblCompany.GetList().FirstOrDefault(x => x.CompanyName == listbox\_company.SelectedItem);

dgv\_orders.DataSource = tblProductModels.GetList().

Where(x => x.CompanyCode == c.CompanyCode).

Select(x => new

{

קוד\_מוצר = x.ProductCode,

תאור\_מוצר = x.ProductDescription,

סןג\_מוצר = x.ProductOfModels().TypeDescription

}).ToList();

}

}

private void btn\_save\_p\_Click(object sender, EventArgs e)

{

double price = 0.0;

bool flag = true;

bool flag1 = true;//בדיקה אם מוסיך כמות למוצר המופיע בליסט ויו

cmb\_search.Enabled = false;

foreach (UserSiseLength us in ua.Controls)

{

if (us.numericUpDown1.Value > 0)

{

int x = Convert.ToInt32(us.numericUpDown1.Value);

foreach (ListViewItem itemList in listView1.Items)

{

if (Convert.ToInt32(itemList.SubItems[0].Text)==us.ps.ProductCode &&

(itemList.SubItems[5].Text) == us.ps.SizeOfProductSize().Description &&

Convert.ToInt32(itemList.SubItems[6].Text) == us.ps.LengthOfProductSize().LengthNum)

{

MessageBox.Show("האם ברצונך לעדכן את הכמות");

x += Convert.ToInt32(itemList.SubItems[3].Text);

listView1.Items.Remove(itemList);

}

}

price = 0.0;

for (int i = 1; i <=x; i++)

{

price += us.ps.ProductSizeOfModels().PurchasePrice;//חישוב מחיר על פי המוצר המשויך ליוזר הנוכחי

}

p = us.ps.ProductSizeOfModels();

ListViewItem listView2 = new ListViewItem(new string[]

{

p.ProductCode.ToString(),

p.ProductOfModels().TypeDescription,

p.PurchasePrice.ToString(),

x.ToString(),

p.ProductDescription,

us.textBox1.Text,

us.textBox2.Text,

price.ToString()

});

listView1.Items.Add(listView2);

}

}

panel1.Controls.Remove(ua);

foreach (Control item in panel1.Controls)

{

item.Visible = true;

}

if (flag)

{

int code = Convert.ToInt32(dgv\_orders.SelectedRows[0].Cells[0].Value);

p = tblProductModels.Find(code);

ListViewItem listView2 = new ListViewItem(new string[]

{

p.ProductCode.ToString(),

p.ProductOfModels().TypeDescription,

p.PurchasePrice.ToString(),

p.ProductDescription,

price.ToString()

});

btn\_save.Visible = true;

btn\_remove.Visible = true;

}

}

private void btn\_home1\_Click(object sender, EventArgs e)

{

sO = new SuppliersOrder();

sO.OrderNumber =tblSuplierOrder.GetNextKey();

sO.SupplierCode =((Supplier)cmb\_search.SelectedItem).SupplierCode;

sO.OrderPrice =0;

sO.Status =true;

sO.Date =DateTime.Now;

tblSuplierOrder.AddNew(sO);

//הוספת הזמנה

foreach (ListViewItem item in listView1.Items)

{

ps = tblProductSize.GetList().FirstOrDefault

(x =>

(x.ProductCode == Convert.ToInt32(item.SubItems[0].Text))&&

x.SizeOfProductSize().Description==item.SubItems[5].Text&&

x.LengthOfProductSize().LengthNum.ToString()==item.SubItems[6].Text);

od = new OrderDetail();

od.OrderNumber =sO.OrderNumber;

od.OrdersAmount =Convert.ToInt32(item.SubItems[3].Text);

od.ProductCode = ps.ProductCode;

od.SizeCode =ps.SizeCode;

od.LengthCode =ps.LengthCode;

od.ArrivingAmount =0;

od.FinalPrice =Convert.ToInt32(item.SubItems[5].Text);

od.Status = "פעיל";

tblOrderDetail.AddNew(od);

}

DialogResult d = MessageBox.Show("?האם בצונך להוסיף הזמנה זו", "אישור הוספה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

MessageBox.Show("ההזמנה נשמרה בהצלחה");

}

}

UserAllSizeLength ua;

private void btn\_home2\_Click(object sender, EventArgs e)

{

bool flag=true;

if(cmb\_search.SelectedIndex==-1)

{

MessageBox.Show("בחר ספק");

flag = false;

}

if (flag)

{

foreach (Control item in panel1.Controls)

{

item.Visible = false;

}

ua = new UserAllSizeLength();

UserSiseLength s = new UserSiseLength();

int code = Convert.ToInt32(dgv\_orders.SelectedRows[0].Cells[0].Value);

p = tblProductModels.Find(code);

foreach (ProductSize item in tblProductSize.GetList().Where(x => x.ProductCode == p.ProductCode))

{

s = new UserSiseLength();

s.ps = item;

s.textBox1.Text = item.SizeOfProductSize().ToString();

s.textBox2.Text = item.LengthOfProductSize().ToString();

ua.Controls.Add(s);

s.Dock = DockStyle.Top;

}

panel1.Visible = true;

panel1.Controls.Add(ua);

ua.Dock = DockStyle.Fill;

btn\_continue.Visible = true;

}

}

private void btn\_remove\_Click\_1(object sender, EventArgs e)

{

foreach (ListViewItem item in listView1.Items)

{

if (listView1.SelectedItems.Count > 0)

{

DialogResult s = MessageBox.Show("?האם ברצונך למחוק מוצר זה", "אישור מחיקה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (s == DialogResult.Yes)

{

listView1.Items.Remove(listView1.SelectedItems[0]);

}

}

}

}

}

טופס ניהול הזמנות

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

public partial class UserOrdersArriving : UserControl

{

OrderDetail od;

SuppliersOrder sO;

SuppliersOrderDB tblSuplierOrder;

SupplierDB tblSuppliers;

CompanyDB tblCompany;

ProductModelsDB tblProductModels;

ProductSizeDB tblProductSize;

OrderDetailDB tblOrderDetail;

SizeDB tbls;

LengthDB tblLength;

Bll.Size ss;

Supplier s;

Company c;

ProductModels p;

ProductSize ps;

public UserOrdersArriving()

{

InitializeComponent();

tblSuppliers = new SupplierDB();

tblCompany = new CompanyDB();

tblSuplierOrder = new SuppliersOrderDB();

tblProductModels = new ProductModelsDB();

tblProductSize = new ProductSizeDB();

tblOrderDetail = new OrderDetailDB();

s = new Supplier();

c = new Company();

p = new ProductModels();

ps = new ProductSize();

tbls = new SizeDB();

tblLength = new LengthDB();

cmb\_search.DataSource = tblSuppliers.GetList().Where(x => x.Status == "פעיל").ToList();

cmb\_search.SelectedIndex = -1;

}

private void cmb\_search\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmb\_search.SelectedIndex > -1)

{

s = ((Supplier)cmb\_search.SelectedItem);

dgv\_orders.DataSource = tblSuplierOrder.

GetList().Where(x => x.Status == true && x.SupplierCode == s.SupplierCode).

Select(x => new {מספר\_הזמנה=x.OrderNumber = x.OrderNumber, תאריך\_הזמנה = x.Date}).ToList();

}

else

dgv\_orders.DataSource = null;

}

int orderNumber = 0;

private void dgv\_orders\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgv\_orders.SelectedRows.Count > 0)

{

orderNumber = Convert.ToInt32(dgv\_orders.SelectedRows[0].Cells[0].Value);

sO = tblSuplierOrder.Find(orderNumber);

dgv\_orders\_details.DataSource = tblOrderDetail.GetList().Where(x => x.OrderNumber == sO.OrderNumber && x.Status=="פעיל")

.Select(x => new

{

מספר\_הזמנה = x.OrderNumber,

קוד\_מוצר=x.ProductCode,

כמות = x.OrdersAmount,

מידה = x.ProductSizeOfOrderDetail().SizeOfProductSize().Description,

אורך = x.ProductSizeOfOrderDetail().LengthOfProductSize().LengthNum,

סטטוס=x.Status

}).ToList();

}

}

private void dgv\_orders\_details\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if(dgv\_orders\_details.SelectedRows.Count>0)

{

numericUpDown1.Value= Convert.ToInt32(dgv\_orders\_details.SelectedRows[0].Cells[2].Value);

}

}

private void btn\_change\_amount\_Click(object sender, EventArgs e)

{

{

Bll.Size s= tbls.GetList().FirstOrDefault(x => x.Description == Convert.ToString(dgv\_orders\_details.SelectedRows[0].Cells[3].Value));

Length l = tblLength.GetList().FirstOrDefault(x => x.LengthNum == Convert.ToInt32(dgv\_orders\_details.SelectedRows[0].Cells[4].Value));

od = tblOrderDetail.Find

(Convert.ToInt32(dgv\_orders\_details.SelectedRows[0].Cells[0].Value),

Convert.ToInt32(dgv\_orders\_details.SelectedRows[0].Cells[1].Value), s.SizeCode

, l.LengthCode

);

ps = tblProductSize.GetList().FirstOrDefault

(x => x.ProductCode == Convert.ToInt32(dgv\_orders\_details.SelectedRows[0].Cells[1].Value)

&& x.SizeCode == s.SizeCode

&& x.LengthCode == l.LengthCode);

}

od.ArrivingAmount=Convert.ToInt32(numericUpDown1.Value);

od.Status = "לא פעיל";

ps.StocksAmount+= Convert.ToInt32(numericUpDown1.Value);

tblOrderDetail.UpDateRow(od);

tblProductSize.UpDateRow(ps);

dgv\_orders.DataSource = null;

dgv\_orders\_details.DataSource = tblOrderDetail.GetList().Where(x => x.OrderNumber == sO.OrderNumber && x.Status=="פעיל")

.Select(x => new

{

מספר\_הזמנה = x.OrderNumber,

קוד\_מוצר = x.ProductCode,

כמות = x.OrdersAmount,

מידה = x.ProductSizeOfOrderDetail().SizeOfProductSize().Description,

אורך = x.ProductSizeOfOrderDetail().LengthOfProductSize().LengthNum,

סטטוס = x.Status,

x.ArrivingAmount

}).ToList();

if(dgv\_orders\_details.Rows.Count==0)

{

sO = tblSuplierOrder.Find(orderNumber);

sO.Status = false;

tblSuplierOrder.UpDateRow(sO);

if (cmb\_search.SelectedIndex > -1)

{

dgv\_orders.DataSource = null;

s = ((Supplier)cmb\_search.SelectedItem);

dgv\_orders.DataSource = tblSuplierOrder.

GetList().Where(x => x.Status == true && x.SupplierCode == s.SupplierCode).

Select(x => new { מספר\_הזמנה = x.OrderNumber = x.OrderNumber, תאריך\_הזמנה = x.Date }).ToList();

}

else

dgv\_orders.DataSource = null;

}

}

}

}

טופס מוצרים

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

public partial class UserProducts : UserControl

{

bool flagUpdate = false;

ProductModelsDB tblProductModels;

ProductModels product;

ProductTypeDB tblProductType;

InstitutionDB tblInstitution;

InstitutionProduct ip;

Company c;

CompanyDB tblc;

Supplier s;

SupplierDB tbls;

InstitutionProductDB tblInstitutionProduct = new InstitutionProductDB();

public UserProducts()

{

InitializeComponent();

panel1.Visible = false;

tblProductModels = new ProductModelsDB();

product = new ProductModels();

tblProductType = new ProductTypeDB();

tblInstitution = new InstitutionDB();

ip = new InstitutionProduct();

c = new Company();

tblc = new CompanyDB();

s = new Supplier();

tbls = new SupplierDB();

cmb\_type\_p.DataSource = tblProductType.GetList();

cmb\_type\_p.SelectedIndex = -1;

cmb\_company.DataSource = tblc.GetList();

cmb\_company.SelectedIndex = -1;

cmb\_instituation.DataSource = tblInstitution.GetList();

cmb\_instituation.SelectedIndex = -1;

panel4.Visible = false;

notPossible();

}

public UserProducts(ProductModels pro) : this()

{

this.product = pro;

Fill(pro);

flagUpdate = true;

}

public void Fill(ProductModels p)

{

ProductModels pro = p;

InstitutionProduct ip = tblInstitutionProduct.GetList().FirstOrDefault(x => x.ProductCode == pro.ProductCode);

txt\_descr\_p.Text = p.ProductDescription;

txt\_descr\_p.Text = p.ProductOfModels().TypeDescription.ToString();

txt\_pur\_p.Text = p.PurchasePrice.ToString();

txt\_salling\_p.Text = p.SallingPrice.ToString();

cmb\_type\_p.SelectedItem = p.ProductOfModels().ToString();

cmb\_type\_p.Text = p.ProductOfModels().ToString();

cmb\_company.SelectedItem = p.CompanyOfP().ToString();

cmb\_company.Text = p.CompanyOfP().ToString();

cmb\_instituation.SelectedItem = p.Status.ToString();

txt\_code\_p.Text = p.ProductCode.ToString();

cmb\_instituation.SelectedItem = ip.InstituationOfInstProduct().InstitutionName;

}

public void ReadOnly()

{

btn\_continue\_p.Visible = false;

txt\_descr\_p.ReadOnly = true;

txt\_pur\_p.ReadOnly = true;

txt\_salling\_p.ReadOnly = true;

cmb\_company.Enabled = false;

cmb\_instituation.Enabled = false;

cmb\_type\_p.Enabled = false;

txt\_code\_p.ReadOnly = true;

}

public bool CreateField(ProductModels p)

{

bool flagOk = true;

errorProvider1.Clear();

try

{

if (cmb\_company.SelectedIndex < 0)

throw new Exception("הקש שם חברה");

p.CompanyCode = ((Company)cmb\_company.SelectedItem).CompanyCode;

}

catch (Exception ex)

{

errorProvider1.SetError(cmb\_company, ex.Message);

flagOk = false;

}

try

{

if (txt\_descr\_p.Text == "")

throw new Exception("הקש תאור מוצר");

p.ProductDescription = txt\_descr\_p.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_descr\_p, ex.Message);

flagOk = false;

}

try

{

if (txt\_salling\_p.Text == "")

throw new Exception("הקש מחיר מכירה");

if (!Validation.IsNum(txt\_salling\_p.Text))

throw new Exception("יש להקיש ספרות");

p.SallingPrice = Convert.ToInt32(txt\_salling\_p.Text); ;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_salling\_p, ex.Message);

flagOk = false;

}

try

{

if (cmb\_instituation.SelectedIndex < 0)

throw new Exception("בחר שם מוסד");

// p.Status = cmb\_instituation.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(cmb\_instituation, ex.Message);

flagOk = false;

}

try

{

if (txt\_pur\_p.Text == "")

throw new Exception("הקש מחיר רכישה");

if (!Validation.IsNum(txt\_pur\_p.Text))

throw new Exception("יש להקיש ספרות");

p.PurchasePrice = Convert.ToInt32(txt\_pur\_p.Text);

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_pur\_p, ex.Message);

flagOk = false;

}

try

{

if (cmb\_type\_p.SelectedIndex < 0)

throw new Exception("בחר סוג");

p.TypeCode = ((ProductType)cmb\_type\_p.SelectedItem).TypeCode;

}

catch (Exception ex)

{

errorProvider1.SetError(cmb\_type\_p, ex.Message);

flagOk = false;

}

if (!flagUpdate)//אם אתה במצב הוספה צור קוד חדש

p.ProductCode = tblProductModels.GetNextKey();

p.Status = "פעיל";

return flagOk;

}

public void Possible()

{

panel1.Visible = false;

panel2.Visible = false;

panel3.Visible = true;

txt\_code\_p.ReadOnly = true;

}

public void notPossible()

{

panel1.Visible = false;

panel2.Visible = true;

panel3.Visible = false;

flagUpdate = false;

//flagAdd = false;

txt\_code\_p.ReadOnly = false;

}

private void btn\_new\_Click(object sender, EventArgs e)

{

Possible();

//flagAdd = true;

txt\_code\_p.Text = tblProductModels.GetNextKey().ToString();

txt\_descr\_p.Text = "";

txt\_pur\_p.Text = "";

txt\_salling\_p.Text = "";

cmb\_instituation.Text = "";

cmb\_type\_p.Text = "";

}

private void btn\_add\_Click(object sender, EventArgs e)

{

panel1.Visible = true;

}

private void txt\_code\_p\_TextChanged(object sender, EventArgs e)

{

dgv\_products.DataSource = tblProductModels.GetList().

Where(x => x.ProductCode.ToString().

StartsWith(txt\_code\_p.Text)).Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

private void txt\_descr\_p\_TextChanged(object sender, EventArgs e)

{

dgv\_products.DataSource = tblProductModels.GetList().

Where(x => x.ProductDescription.ToString().

StartsWith(txt\_descr\_p.Text)).Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

private void txt\_pur\_p\_TextChanged(object sender, EventArgs e)

{

dgv\_products.DataSource = tblProductModels.GetList().

Where(x => x.PurchasePrice.ToString().

StartsWith(txt\_pur\_p.Text)).Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_הזמנה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

private void txt\_salling\_p\_TextChanged(object sender, EventArgs e)

{

dgv\_products.DataSource = tblProductModels.GetList().

Where(x => x.SallingPrice.ToString().

StartsWith(txt\_salling\_p.Text)).Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_הזמנה= x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

private void cmb\_type\_p\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmb\_type\_p.SelectedIndex > -1)

{

dgv\_products.DataSource = tblProductModels.GetList().

Where(x => x.ProductOfModels().TypeDescription.ToString().

StartsWith(cmb\_type\_p.SelectedItem.ToString())).Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_הזמנה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

}

).ToList();

}

}

private void txt\_company\_TextChanged(object sender, EventArgs e)

{

if (cmb\_company.SelectedIndex > -1)

{

dgv\_products.DataSource = tblProductModels.GetList().

Where(x => x.CompanyOfP().CompanyName.ToString().

StartsWith(cmb\_company.SelectedItem.ToString())).Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_הזמנה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

}

public bool CreateType(ProductType t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (txt\_new\_p.Text == "")

throw new Exception("יש להקיש סוג מוצר");

t.TypeDescription = txt\_new\_p.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_new\_p, ex.Message);

flag = false;

}

t.TypeCode = tblProductType.GetNextKey();

return flag;

}

private void btn\_save\_p\_Click(object sender, EventArgs e)

{

if (flagUpdate)

{

if (CreateField(product))

{

DialogResult d = MessageBox.Show("?האם ברצונך לערוך מוצר זה", "אישור עריכה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

// btn\_continue\_p.Enabled = false;

tblProductModels.UpDateRow(product);

OpenU(2);

}

}

}

else

{

product = new ProductModels();

ip = new InstitutionProduct();

if (CreateField(product))

{

DialogResult d = MessageBox.Show("?האם ברצונך להוסיף מוצר זה", "אישור הוספה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

btn\_save.Enabled = false;

tblProductModels.AddNew(product);

ip.ProductCode = product.ProductCode;

ip.InstitutionCode = ((Institution)cmb\_instituation.SelectedItem).InstitutionCode;

tblInstitutionProduct.AddNew(ip);

OpenU(1);

}

}

}

}

private void OpenU(int num)

{

UserProduct\_s\_l f1 = new UserProduct\_s\_l(product);

if (num == 2)

{

f1.btn\_save.Visible = false;

f1.btn\_return.Visible = true;

}

else

{

f1.btn\_return.Visible = false;

}

f1.Location = new Point(148, 98);

// this.Visible = false;//.Clear();

Controls.Add(f1);

f1.BringToFront();

f1.Dock = DockStyle.Fill;

}

private void btn\_save\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (ProductType item in tblProductType.GetList())

{

if (item.ToString().Equals(txt\_new\_p.Text))

{

MessageBox.Show("סוג זה קיים במערכת");

flag1 = false;

}

}

if (flag1)

{

ProductType p = new ProductType();

if (CreateType(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף סוג זה ", "הוספת מוצר", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblProductType.AddNew(p);

this.Visible = false;

cmb\_type\_p.DataSource = null;

cmb\_type\_p.DataSource = tblProductType.GetList();

cmb\_type\_p.Text = p.ToString();

cmb\_type\_p.DataSource = tblProductType.GetList();

this.Visible = true;

this.Controls.Clear();

UserProducts s = new UserProducts();

s.Location = new Point(148, 98);

this.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

}

}

else

{

txt\_new\_p.Text = "";

panel1.Visible = false;

}

}

private void btn\_cancel\_Click(object sender, EventArgs e)

{

txt\_new\_p.Text = "";

panel1.Visible = false;

}

private void btn\_home5\_Click(object sender, EventArgs e)

{

if (dgv\_products.SelectedRows.Count > 0)

{

flagUpdate = true;

int code = Convert.ToInt32(dgv\_products.SelectedRows[0].Cells[0].Value.ToString());

product = tblProductModels.Find(code);//חיפוש עצם הלקוח עפ תז שנבחרה בגריד

Possible(); // UserProducts p=new UserProducts(product);

Fill(product);

}

}

private void cmb\_status\_p\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmb\_instituation.SelectedIndex > -1)

{

//int codeInstituation = ((Institution)cmb\_instituation.SelectedItem).InstitutionCode;

dgv\_products.DataSource = tblInstitutionProduct.GetList().Where(x=> x.InstituationOfInstProduct().InstitutionName.StartsWith(cmb\_instituation.Text)).Select(x => new { קוד\_מוצר = x.ProductModelsOfInstituation().ProductCode,

מחיר\_הזמנה = x.ProductModelsOfInstituation().PurchasePrice,

מחיר\_מכירה = x.ProductModelsOfInstituation().SallingPrice,

שם\_החברה = x.ProductModelsOfInstituation().CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductModelsOfInstituation().ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductModelsOfInstituation().ProductDescription,

שם\_ספק = x.ProductModelsOfInstituation().CompanyOfP().SupplierOfCompany().SupplierName

}).ToList();

}

}

private void btn\_cancel\_p\_Click(object sender, EventArgs e)

{

notPossible();

errorProvider1.Clear();

txt\_descr\_p.Text = "";

txt\_new\_p.Text = "";

txt\_pur\_p.Text = "";

txt\_salling\_p.Text ="";

txt\_code\_p.Text = "";

cmb\_company.SelectedIndex = -1;

cmb\_instituation.SelectedIndex = -1;

cmb\_type\_p.SelectedIndex = -1;

cmb\_company.Enabled = true;

cmb\_instituation.Enabled = true;

cmb\_type\_p.Enabled = true;

txt\_descr\_p.ReadOnly = false;

txt\_pur\_p.ReadOnly = false;

txt\_salling\_p.ReadOnly = false;

txt\_code\_p.ReadOnly = false;

}

private void btn\_home4\_Click(object sender, EventArgs e)

{

if (dgv\_products.SelectedRows.Count > 0)

{

int code = Convert.ToInt32(dgv\_products.SelectedRows[0].Cells[0].Value.ToString());

product = tblProductModels.Find(code);//חיפוש עצם הלקוח עפ תז שנבחרה בגריד

Possible(); // UserProducts p=new UserProducts(product);

Fill(product);

ReadOnly();

}

}

private void txt\_instituation\_TextChanged(object sender, EventArgs e)

{

dgv\_products.DataSource = tblProductModels.GetList().

Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

});

}

private void btn\_home3\_Click(object sender, EventArgs e)

{

dgv\_products.DataSource = tblProductModels.GetList().

Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

private void btn\_\_Click(object sender, EventArgs e)

{

if (dgv\_products.SelectedRows.Count > 0)

{

DialogResult s = MessageBox.Show("?האם ברצונך למחוק מוצר זה", "אישור מחיקה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (s == DialogResult.Yes)

{

int code = Convert.ToInt32(dgv\_products.SelectedRows[0].Cells[0].Value.ToString());

product = tblProductModels.Find(code);

product.Status = "לא פעיל";

tblProductModels.UpDateRow(product);

dgv\_products.DataSource = tblProductModels.GetList().Where(x=>x.Status=="פעיל").

Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).OrderBy(v => v.קוד\_מוצר).ToList();

}

}

}

private void txt\_pur\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_code\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_salling\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_new\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void txt\_descr\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

private void cmb\_company\_KeyPress(object sender, KeyPressEventArgs e)

{

}

private void btn\_home1\_Click(object sender, EventArgs e)

{

panel4.Visible = true;

}

private void button1\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

panel4.Visible = false;

}

private void btn\_home2\_Click(object sender, EventArgs e)

{

panel4.Visible = true;

}

private void button2\_Click(object sender, EventArgs e)

{

bool flag1 = true;

foreach (Institution item in tblInstitution.GetList())

{

if (item.ToString().Equals(textBox1.Text))

{

MessageBox.Show("מוסד זה קיים במערכת");

flag1 = false;

}

}

if (flag1)

{

Institution p = new Institution();

if (CreateInstituation(p))

{

DialogResult d = MessageBox.Show("? האם ברצונך להוסיף מוסד זה ", "הוספת מוסד", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

tblInstitution.AddNew(p);

this.Visible = false;

cmb\_instituation.DataSource = null;

cmb\_instituation.DataSource = tblInstitution.GetList();

cmb\_instituation.Text = p.ToString();

cmb\_instituation.DataSource = tblInstitution.GetList();

this.Visible = true;

this.Controls.Clear();

UserProducts s = new UserProducts();

s.Location = new Point(148, 98);

this.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

}

}

else

{

txt\_new\_p.Text = "";

panel1.Visible = false;

}

}

public bool CreateInstituation(Institution t)

{

bool flag = true;

errorProvider1.Clear();

try

{

if (textBox1.Text == "")

throw new Exception("יש להקיש שם מוסד");

t.InstitutionName = textBox1.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(textBox1, ex.Message);

flag = false;

}

t.InstitutionCode = tblInstitution.GetNextKey();

return flag;

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsHebrewChar(e.KeyChar);

}

}

טופס מידות ואורכים

public partial class UserProduct\_s\_l : UserControl

{

SizeDB tblsize;

LengthDB tblLength;

ProductModels p;

ProductModelsDB tblPm;

ProductSizeDB tblPro\_Size;

ProductSize ps;

bool flagUpdate = false;

public UserProduct\_s\_l(ProductModels pm)

{

InitializeComponent();

btn\_return.Visible = false;

tblLength = new LengthDB();

tblsize = new SizeDB();

tblPro\_Size = new ProductSizeDB();

ps = new ProductSize();

tblPm = new ProductModelsDB();

this.p = pm;

FillSizeLength();

flagUpdate = true;

}

private void FillSizeLength()

{

int m = 0;

int l = 0;

foreach (Bll.Size item in tblsize.GetList())

{

//

checkedListBox\_size.Items.Add(item);

if (tblPro\_Size.GetList().Count(x => (x.SizeCode == item.SizeCode) && (p.ProductCode == x.ProductCode) && x.Status==0)!=0 )

checkedListBox\_size.SetItemChecked(m,true);

m++;

}

foreach (Length item in tblLength.GetList().Where(x=>x.LengthNum!=0))

{

checkedListBox\_length.Items.Add(item);

if (tblPro\_Size.GetList().Count(x => (x.LengthCode == item.LengthCode) && (p.ProductCode == x.ProductCode && x.Status == 0)) != 0)

checkedListBox\_length.SetItemChecked(l, true);

l++;

}

}

public bool CreateField(ProductSize ps,Length l,Bll.Size s)

{

bool flagOk = true;

ps.ProductCode = p.ProductCode;

ps.LengthCode =l.LengthCode;

ps.SizeCode =s.SizeCode;

return flagOk;

}

private void btn\_save\_Click(object sender, EventArgs e)

{

DialogResult d = MessageBox.Show("?האם בצונך להוסיף מידות ואורכים אלו", "אישור הוספה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

btn\_return.Visible = true;

foreach (Bll.Size itemSize in checkedListBox\_size.CheckedItems)

{

if(checkBox1.Checked)

{

OpenUc();

}

else//הוספת מידות למוצרים ללא אורך

{

ps = new ProductSize();

ps.ProductCode = p.ProductCode;

ps.LengthCode =1;

ps.SizeCode = itemSize.SizeCode;

if (tblPro\_Size.Find(ps.ProductCode, ps.SizeOfProductSize().SizeCode, ps.LengthCode) == null)

{

tblPro\_Size.AddNew(ps);

}

else

{

ps = tblPro\_Size.Find(ps.ProductCode, ps.SizeOfProductSize().SizeCode, ps.LengthCode);

ps.Status = 0;

tblPro\_Size.UpDateRow(ps);

}

OpenUc();

}

}

}

}

private void OpenUc()

{

UserProducts f1 = new UserProducts();

f1.Location = new Point(148, 98);

this.Controls.Clear();

Controls.Add(f1);

f1.Dock = DockStyle.Fill;

f1.dgv\_products.DataSource = tblPm.GetList().

Select(x => new

{

קוד\_מוצר = x.ProductCode,

מחיר\_קניה = x.PurchasePrice,

מחיר\_מכירה = x.SallingPrice,

שם\_החברה = x.CompanyOfP().CompanyName,

סוג\_מוצר = x.ProductOfModels().TypeDescription,

תאור\_מוצר = x.ProductDescription,

שם\_ספק = x.CompanyOfP().SupplierOfCompany().SupplierName

}

).ToList();

}

private void lab\_\_Click(object sender, EventArgs e)

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked)

checkedListBox\_length.Visible = true;

else

checkedListBox\_length.Visible = false;

}

private void checkedListBox\_length\_ItemCheck(object sender, ItemCheckEventArgs e)

{

int status = 0;

if (flagUpdate)

{

if (checkedListBox\_length.GetItemChecked(checkedListBox\_length.SelectedIndex))

{

status = 1;

}

int countL = tblPro\_Size.GetList().Count(x => x.ProductCode == p.ProductCode && ((Length)(checkedListBox\_length.SelectedItem)).LengthCode == x.LengthCode);

if (countL > 0)

{

foreach (ProductSize item in tblPro\_Size.GetList().Where(x => x.ProductCode == p.ProductCode && ((Length)(checkedListBox\_length.SelectedItem)).LengthCode == x.LengthCode))

{

item.Status = status;

tblPro\_Size.UpDateRow(item);

}

}

//אם לא קקים עדיין בודק האם להוסיף

else if ((checkedListBox\_length.GetItemChecked(checkedListBox\_length.SelectedIndex)) == false)

{

//עובר על כל המידות ומוסיף אורך לכל מידה

foreach (Bll.Size itemSize in checkedListBox\_size.CheckedItems)

{

{

ps = new ProductSize();

if (CreateField(ps, ((Length)(checkedListBox\_length.SelectedItem)), itemSize))

{

tblPro\_Size.AddNew(ps);

}

}

//בדיקה שלא קיים בטבלה מוצר במידה ובאורך הנוכחי

//הוספת לטבלת productSize

}

}

}

//אם לא קיים בכלל צריך לעשות הוספה

}

private void checkedListBox\_size\_ItemCheck(object sender, ItemCheckEventArgs e)

{

int status = 0;

if (flagUpdate)

{

if (checkedListBox\_size.GetItemChecked(checkedListBox\_size.SelectedIndex))

{

status = 1;

}

//בודק אם המידה כבר קיימת במוצר אם לא מבצע הוספה אם קיים מבצע רק עדכון

int countL = tblPro\_Size.GetList().Count(x => x.ProductCode == p.ProductCode && ((Bll.Size)(checkedListBox\_size.SelectedItem)).SizeCode == x.SizeCode);

if (countL > 0)

{

foreach (ProductSize item in tblPro\_Size.GetList().Where(x => x.ProductCode == p.ProductCode && ((Bll.Size)(checkedListBox\_size.SelectedItem)).SizeCode == x.SizeCode))

{

item.Status = status;

tblPro\_Size.UpDateRow(item);

}

}

//אם לא קיים מבצע הוספה

else if ((checkedListBox\_size.GetItemChecked(checkedListBox\_size.SelectedIndex)) == false)

{

foreach (Bll.Length itemLength in checkedListBox\_length.CheckedItems)

{

{//מוסיף חדש

ps = new ProductSize();

if (CreateField(ps, itemLength, ((Bll.Size)(checkedListBox\_size.SelectedItem))))

{

tblPro\_Size.AddNew(ps);

}

}

//בדיקה שלא קיים בטבלה מוצר במידה ובאורך הנוכחי

//הוספת לטבלת productSize

}

}

//אם לא קיים בכלל צריך לעשות הוספה

}

}

private void CloseSingleCustomer()

{

foreach (Control item in this.Parent.Controls)

{

if (item is UserSale)

{

item.Visible = true;

}

}

this.Parent.Controls.Remove(this);

}

private void btnPrev\_Click(object sender, EventArgs e)

{

if(checkedListBox\_size.CheckedItems.Count==0)

{

DialogResult dia = MessageBox.Show("אין מידות קיימות למוצר, המוצר יהפוך ללא פעיל, האם ברצונך להמשיך?", "", MessageBoxButtons.YesNo);

if(dia==DialogResult.Yes)

{

p.Status = "לא פעיל";

tblPm.UpDateRow(p);

this.Parent.Controls.Remove(this);

}

}

else

this.Parent.Controls.Remove(this);

}

}

טופס החזרת מוצרים

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

public partial class UserReturnProduct : UserControl

{

SizeDB tbls;

LengthDB tblLength;

Supplier s;

Company c;

ProductModels p;

ProductSize ps;

Sale sale;

SaleDetail saleD;

SaleDB tblSale;

SaleDetailDB tblSaleDetail;

ProductSizeDB tblProductSize;

public UserReturnProduct()

{

InitializeComponent();

s = new Supplier();

c = new Company();

p = new ProductModels();

ps = new ProductSize();

tbls = new SizeDB();

tblLength = new LengthDB();

tblSale = new SaleDB();

tblSaleDetail = new SaleDetailDB();

tblProductSize = new ProductSizeDB();

cmb\_search.DataSource = tblSale.GetList().Where(x => x.Status == "פעיל").ToList();

cmb\_search.SelectedIndex = -1;

}

private void cmb\_search\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmb\_search.SelectedIndex > -1)

{

sale = ((Sale)cmb\_search.SelectedItem);

dgv\_salling.DataSource = tblSale.

GetList().Where(x => x.Status=="פעיל" && x.SalesCode == sale.SalesCode).

Select(x => new

{

קוד\_קניה=x.SalesCode

,תאריך\_קניה=x.Date

,מחיר\_הקניה=x.FinalPurchase

}).ToList();

}

else

dgv\_salling.DataSource = null;

}

private void dgv\_salling\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgv\_salling.SelectedRows.Count > 0)

{

int sale\_code = Convert.ToInt32(dgv\_salling.SelectedRows[0].Cells[0].Value);

sale = tblSale.Find(sale\_code);

dgv\_salling\_details.DataSource = tblSaleDetail.GetList().Where(x => x.SaleCode == sale.SalesCode && x.Status)

.Select(x => new

{

קוד\_מוצר=x.ProductCode,

שם\_מוצר=x.SaleDetailOFSize().ProductSizeOfModels().ProductDescription,

מידה=x.SaleDetailOFSize().SizeOfProductSize().Description,

אורך=x.SaleDetailOFSize().LengthOfProductSize().LengthNum,

כמות=x.Amount,

מחיר\_ליחידה=x.SaleDetailOFSize().ProductSizeOfModels().SallingPrice,

מחיר\_סופי=x.FinalPrice

}).ToList();

}

}

private void dgv\_salling\_details\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (dgv\_salling\_details.SelectedRows.Count > 0)

{

numericUpDown1.Value = Convert.ToInt32(dgv\_salling\_details.SelectedRows[0].Cells[4].Value);

}

}

private void btn\_change\_amount\_Click(object sender, EventArgs e)

{

Bll.Size s = tbls.GetList().FirstOrDefault(x => x.Description == Convert.ToString(dgv\_salling\_details.SelectedRows[0].Cells[2].Value));

Length l = tblLength.GetList().FirstOrDefault(x => x.LengthNum == Convert.ToInt32(dgv\_salling\_details.SelectedRows[0].Cells[3].Value));

saleD = tblSaleDetail.Find

(Convert.ToInt32(dgv\_salling.SelectedRows[0].Cells[0].Value),

Convert.ToInt32(dgv\_salling\_details.SelectedRows[0].Cells[0].Value),

s.SizeCode,

l.LengthCode

);

ps = tblProductSize.GetList().FirstOrDefault

(x => x.ProductCode == Convert.ToInt32(dgv\_salling\_details.SelectedRows[0].Cells[0].Value)

&& x.SizeCode == s.SizeCode && x.LengthCode == l.LengthCode);

saleD.Amount = Convert.ToInt32(dgv\_salling\_details.SelectedRows[0].Cells[4].Value)-Convert.ToInt32(numericUpDown1.Value);

ps.StocksAmount += Convert.ToInt32(numericUpDown1.Value);

tblSaleDetail.UpDateRow(saleD);

tblProductSize.UpDateRow(ps);

dgv\_salling.DataSource = null;

dgv\_salling\_details.DataSource = tblSaleDetail.GetList().Where(x => x.SaleCode == saleD.SaleCode && x.Status)

.Select(x => new

{

קוד\_מוצר = x.ProductCode,

שם\_מוצר = x.SaleDetailOFSize().ProductSizeOfModels().ProductDescription,

מידה = x.SaleDetailOFSize().SizeOfProductSize().Description,

אורך = x.SaleDetailOFSize().LengthOfProductSize().LengthNum,

כמות = x.Amount,

מחיר\_ליחידה = x.SaleDetailOFSize().ProductSizeOfModels().SallingPrice,

מחיר\_סופי = x.FinalPrice

}).ToList();

if (dgv\_salling\_details.Rows.Count == 0)

{

int sale\_code = Convert.ToInt32(dgv\_salling.SelectedRows[0].Cells[0].Value);

sale = tblSale.Find(sale\_code);

sale.Status = "לא פעיל";

tblSale.UpDateRow(sale);

if (cmb\_search.SelectedIndex > -1)

{

dgv\_salling.DataSource = null;

sale = ((Sale)cmb\_search.SelectedItem);

dgv\_salling.DataSource = tblSale.

GetList().Where(x => x.Status=="פעיל" && x.SalesCode == sale.SalesCode).

Select(x => new {

קוד\_קניה = x.SalesCode,

תאריך\_קניה = x.Date,

מחיר\_הקניה = x.FinalPurchase

}).ToList();

}

else

dgv\_salling.DataSource = null;

}

double price = saleD.SaleDetailOFSize().ProductSizeOfModels().SallingPrice \* Convert.ToDouble(numericUpDown1.Value);

MessageBox.Show("הזיכוי בסך" + " " + price);

}

}

}

טופס מכירות

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

public partial class UserSale : UserControl

{

OrderDetail od;

SuppliersOrder sO;

SuppliersOrderDB tblSuplierOrder;

SupplierDB tblSuppliers;

CompanyDB tblCompany;

ProductModelsDB tblProductModels;

ProductSizeDB tblProductSize;

OrderDetailDB tblOrderDetail;

DetailPaymentsDB tblDetailPayments;

SaleDetailDB tblSaleDetail;

SaleDB tblSale;

SizeDB tblSize=new SizeDB();

LengthDB tblLength=new LengthDB();

Supplier s;

Company c;

Customer cus;

ProductModels p;

ProductSize ps;

DetailPayments dp;

Sale sale;

SaleDetail saleD;

CustomerDB tblCustomer;

bool flag\_d\_t\_p=false;

public UserSale()

{

InitializeComponent();

tblSuppliers = new SupplierDB();

tblCompany = new CompanyDB();

tblSuplierOrder = new SuppliersOrderDB();

tblProductModels = new ProductModelsDB();

tblProductSize = new ProductSizeDB();

tblOrderDetail = new OrderDetailDB();

tblSize = new SizeDB();

tblLength = new LengthDB();

tblCustomer = new CustomerDB();

tblSale = new SaleDB();

tblDetailPayments = new DetailPaymentsDB();

tblSaleDetail = new SaleDetailDB();

s = new Supplier();

c = new Company();

p = new ProductModels();

ps = new ProductSize();

cus = new Customer();

cmb\_search.DataSource = tblProductModels.GetList().Where(x => x.Status == "פעיל").Select(x=>x.ProductCode).ToList();

cmb\_search.SelectedIndex = -1;

panel\_payment.Visible = false;

panel\_visa.Visible = false;

panel\_check.Visible = false;

txt\_id.Visible = false;

textBox1.Visible = false;

}

public UserSale(string tz) : this()

{

txt\_id.Text = tz;

txt\_id.ReadOnly = true;

}

private void cmb\_search\_SelectedIndexChanged(object sender, EventArgs e)

{

cmb\_size.DataSource = tblProductSize.GetList().Where

(x => x.ProductCode ==Convert.ToInt32(cmb\_search.SelectedItem))

.Select(x=>x.SizeOfProductSize().Description).Distinct().ToList();

}

private void cmb\_size\_SelectedIndexChanged(object sender, EventArgs e)

{

cmb\_length.DataSource = tblProductSize.GetList().Where

(x =>x.SizeOfProductSize().Description==Convert.ToString(cmb\_size.SelectedItem)

&& x.ProductCode == Convert.ToInt32(cmb\_search.SelectedItem))

.Select(x => x.LengthOfProductSize().LengthNum).ToList();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (cmb\_length.SelectedIndex > -1 && numericUpDown1.Value > 0)

{

int codeSize = tblSize.GetList().FirstOrDefault(x=>x.Description==cmb\_size.SelectedItem.ToString()).SizeCode;

int codeLength = tblLength.GetList().FirstOrDefault(x => x.LengthNum == Convert.ToInt32(cmb\_length.SelectedItem.ToString())).LengthCode;

ps = tblProductSize.Find(Convert.ToInt32(cmb\_search.SelectedItem), codeSize,codeLength);

int amount = Convert.ToInt32(numericUpDown1.Value);

bool flag = true;

//בדיקה אם יש את הכמות במלאי

if (amount > ps.StocksAmount)

MessageBox.Show("הכמות במלאי:" + ps.StocksAmount + " בחר כמות מתאימה");

//הוספה לליטס

else

{

//בדיקה אם המוצר כבר קיים

foreach (ListViewItem item in listview\_salling.Items)

{

if (Convert.ToInt32(item.SubItems[0].Text) == ps.ProductCode&&

item.SubItems[2].Text == ps.SizeOfProductSize().Description &&

Convert.ToInt32(item.SubItems[3].Text) == ps.LengthOfProductSize().LengthNum)

{

amount += Convert.ToInt32(item.SubItems[4].Text);

//בדיקה אם יש את הכמות במלאי

if (amount > ps.StocksAmount)

{

MessageBox.Show("הכמות במלאי:" + ps.StocksAmount + " בחר כמות מתאימה");

flag = false;

}

else

listview\_salling.Items.Remove(item);

}

}

if (flag)

{

p = tblProductModels.Find(ps.ProductCode);

ListViewItem listview\_sale = new ListViewItem(new string[]

{

ps.ProductCode.ToString(),

p.ProductDescription,ps.SizeOfProductSize().Description,

ps.LengthOfProductSize().LengthNum.ToString(),

amount.ToString(),

p.SallingPrice.ToString(),

(amount\*p.SallingPrice).ToString()

});

listview\_salling.Items.Add(listview\_sale);

double temp = 0;

foreach (ListViewItem item in listview\_salling.Items)

{

temp+= Convert.ToDouble(item.SubItems[6].Text);

}

txt\_finalPrice.Text= "";

txt\_finalPrice.Text=temp.ToString();

}

}

}

}

private void btn\_remove\_Click(object sender, EventArgs e)

{

foreach (ListViewItem item in listview\_salling.Items)

{

if (listview\_salling.SelectedItems.Count > 0)

{

DialogResult s = MessageBox.Show("?האם ברצונך למחוק מוצר זה", "אישור מחיקה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (s == DialogResult.Yes)

{

listview\_salling.Items.Remove(listview\_salling.SelectedItems[0]);

}

}

}

}

private void btn\_payments\_Click(object sender, EventArgs e)

{

bool flag = true;

if(listview\_salling.Items.Count==0)

{

MessageBox.Show("בחר מוצרים");

flag = false;

}

if(flag)

panel\_payment.Visible = true;

}

private void radioB\_cheq\_CheckedChanged(object sender, EventArgs e)

{

if (radioB\_cheq.Checked)

{

panel\_check.Location = new Point(53, 240);

panel\_check.Visible = true;

panel\_visa.Visible = false;

}

}

private void radioB\_creditcard\_CheckedChanged(object sender, EventArgs e)

{

if (radioB\_creditcard.Checked)

{

panel\_visa.Visible = true;

panel\_check.Visible = false;

}

}

private void radiob\_cash\_CheckedChanged(object sender, EventArgs e)

{

panel\_visa.Visible = false;

panel\_check.Visible = false;

}

public bool CreateFields(DetailPayments dp)

{

bool flagOk=true;

errorProvider1.Clear();

if (radioB\_creditcard.Checked)

{

try

{

if (txt\_cw.Text == "")

throw new Exception("שדה חובה");

if (!Validation.IsNum(txt\_cw.Text))

throw new Exception("יש להקיש מספרים");

if ((Validation.HowLongDigit(txt\_cw.Text) != 3))

throw new Exception("הקש ספרות בגב הכרטיס");

dp.CreditCw = txt\_cw.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_cw, ex.Message);

flagOk = false;

}

try

{

//5326-1053-0326-0981

if (maskedTxtb\_creditcard.Text == " - - -")

throw new Exception("שדה חובה");

if (!Validation.CheckVisa(maskedTxtb\_creditcard.Text))

throw new Exception("יש להקיש מספרים");

dp.CreditNumber = maskedTxtb\_creditcard.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(maskedTxtb\_creditcard, ex.Message);

flagOk = false;

}

try

{

if (dateTimePicker1.CustomFormat == "בחר")

throw new Exception("שדה חובה");

if (dateTimePicker1.Value.Date < DateTime.Now.Date)

throw new Exception("כרטיס לא תקף");

dp.CreditForce = dateTimePicker1.Value;

}

catch (Exception ex)

{

errorProvider1.SetError(dateTimePicker1, ex.Message);

flagOk = false;

}

dp.PaymentCode = tblDetailPayments.GetNextKey();

dp.PaymentNumber = Convert.ToInt32(nupdown\_paymentNumber.Value);

dp.ChequeNumber = txt\_cheque\_number.Text;

dp.BankNumber = txt\_bank\_number.Text;

dp.BranchNumber = txt\_branch\_number.Text;

dp.UserAccount = txt\_user\_account.Text;

dp.SalesCode = tblSale.GetNextKey();

}

else

{

if (radioB\_cheq.Checked)

{

//מספר מחאה

try

{

if (txt\_cheque\_number.Text == "")

throw new Exception("שדה חובה");

if (!Validation.IsNum(txt\_cheque\_number.Text))

throw new Exception("יש להקיש מספרים");

if ((Validation.HowLongDigit(txt\_cheque\_number.Text) != 7))

throw new Exception("יש להקיש מספר המחאה תקין");

dp.ChequeNumber = txt\_cheque\_number.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_cheque\_number, ex.Message);

flagOk = false;

}

//מספר בנק

try

{

if (txt\_bank\_number.Text == "")

throw new Exception("שדה חובה");

dp.BankNumber = txt\_bank\_number.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_bank\_number, ex.Message);

flagOk = false;

}

//סניף בנק

try

{

if (txt\_branch\_number.Text == "")

throw new Exception("שדה חובה");

if (!Validation.IsNum(txt\_branch\_number.Text))

throw new Exception("יש להקיש מספרים");

if ((Validation.HowLongDigit(txt\_branch\_number.Text) < 3) || (Validation.HowLongDigit(txt\_branch\_number.Text) > 3))

throw new Exception("לא תקין מספר סניף");

dp.BranchNumber = txt\_branch\_number.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_branch\_number, ex.Message);

flagOk = false;

}

//מספר חשבון

try

{

if (txt\_user\_account.Text == "")

throw new Exception("שדה חובה");

if (!Validation.IsNum(txt\_user\_account.Text))

throw new Exception("יש להקיש מספרים");

if ((Validation.HowLongDigit(txt\_user\_account.Text) < 6) || (Validation.HowLongDigit(txt\_user\_account.Text) > 9))

throw new Exception("הקש מספר חשבון תקין");

dp.UserAccount = txt\_user\_account.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_user\_account, ex.Message);

flagOk = false;

}

dp.PaymentCode = tblDetailPayments.GetNextKey();

dp.PaymentNumber = Convert.ToInt32(numericUpDown1.Value);

dp.CreditNumber = maskedTxtb\_creditcard.Text;

dp.CreditForce = DateTime.Today; ;

dp.CreditCw = txt\_cw.Text;

dp.SalesCode = tblSale.GetNextKey();

}

else

{

dp.PaymentCode = tblDetailPayments.GetNextKey();

dp.PaymentNumber = 0;

dp.CreditNumber = "0000000000000000";

dp.CreditForce = DateTime.Today;

dp.CreditCw = "000";

dp.SalesCode = tblSale.GetNextKey();

}

}

if (rb\_members.Checked)

{

try

{

if (txt\_id.Text == "")

throw new Exception("שדה חובה");

if (txt\_id.Text == "")

throw new Exception("הקש תעודת זהות");

dp.CustomerTz = txt\_id.Text;

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_id, ex.Message);

flagOk = false;

}

}

if (radiob\_cash.Checked)

dp.PaymentWay = "מזומן";

else

if (radioB\_cheq.Checked)

dp.PaymentWay = "צ'ק";

dp.PaymentWay = "אשראי";

if (rb\_general.Checked)

dp.CustomerTz = tblCustomer.Find( textBox1.Text).Tz;

return flagOk;

}

private void btn\_sale\_Click(object sender, EventArgs e)

{

dp = new DetailPayments();

CreateFields(dp);

tblDetailPayments.AddNew(dp);

sale = new Sale();

if (rb\_general.Checked)

sale.CustomerTz = textBox1.Text;

else

sale.CustomerTz = txt\_id.Text;

sale.SalesCode = tblSale.GetNextKey();

sale.Date = DateTime.Now;

sale.FinalPurchase = Convert.ToInt32(txt\_finalPrice.Text);

sale.Status = "פעיל";

tblSale.AddNew(sale);

foreach (ListViewItem items in listview\_salling.Items)

{

saleD = new SaleDetail();

saleD.SaleCode = sale.SalesCode;

//saleD.ProductCode =Convert.ToInt32(items.SubItems[0].Text);

ProductSize ps = new ProductSize();

ps = tblProductSize.Find(Convert.ToInt32(items.SubItems[0].Text), tblSize.GetList().FirstOrDefault(x => x.Description == (items.SubItems[2].Text)).SizeCode, tblLength.GetList().FirstOrDefault(x => x.LengthNum == Convert.ToInt32(items.SubItems[3].Text)).LengthCode);

saleD.ProductCode = ps.ProductCode;

saleD.SizeCode = ps.SizeCode;

saleD.LengthCode = ps.LengthCode;

saleD.Amount = Convert.ToInt32(items.SubItems[4].Text);

saleD.FinalPrice = Convert.ToInt32(items.SubItems[6].Text);

saleD.Status = true;

tblSaleDetail.AddNew(saleD);

}

MessageBox.Show("הקניה הושלמה בהצלחה");

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

flag\_d\_t\_p = true;

dateTimePicker1.CustomFormat = "MM / yy";

}

private void rb\_members\_CheckedChanged(object sender, EventArgs e)

{

txt\_id.Visible = true;

}

private void txt\_id\_TextChanged(object sender, EventArgs e)

{

if(txt\_id.Text.Length==9)

{

cus = tblCustomer.GetList().FirstOrDefault(x => x.Tz == txt\_id.Text);

if (cus == null)

{

if ((Validation.CheckId(txt\_id.Text)) && (tblCustomer.Find(txt\_id.Text)==null))

{

UserSingleCustomer up = new UserSingleCustomer(txt\_id.Text, 1);

Control c = this.Parent;

up.Location = new Point(0, 0);

txt\_id.ReadOnly = true;

this.Visible = false;

c.Controls.Add(up);

up.Dock = DockStyle.Fill;

errorProvider1.Clear();

}

else

{

try

{

MessageBox.Show("ttttt");

if (Validation.CheckId(txt\_id.Text))

throw new Exception("הקש תעודת זהות תקינה");

}

catch (Exception ex)

{

errorProvider1.SetError(txt\_id, ex.Message);

}

}

}

else

{

txt\_id.Visible = true;

txt\_id.ReadOnly = true;

}

}

}

private void UserSale\_Load(object sender, EventArgs e)

{

}

private void btn\_return\_Click(object sender, EventArgs e)

{

this.Controls.Clear();

UserReturnProduct s = new UserReturnProduct();

s.Location = new Point(148, 98);

this.Controls.Add(s);

s.Dock = DockStyle.Fill;

}

private void rb\_general\_CheckedChanged(object sender, EventArgs e)

{

textBox1.Text = "111111111";

}

private void txt\_cw\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void dateTimePicker1\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void maskedTxtb\_creditcard\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

private void txt\_id\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = Validation.IsNumberChar(e.KeyChar);

}

}

}

טופס מידות ואורכים

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using PlisseeProgect.Bll;

namespace PlisseeProgect.Gui

{

public partial class UserSiseLength : UserControl

{

public ProductSize ps;

public UserSiseLength()

{

InitializeComponent();

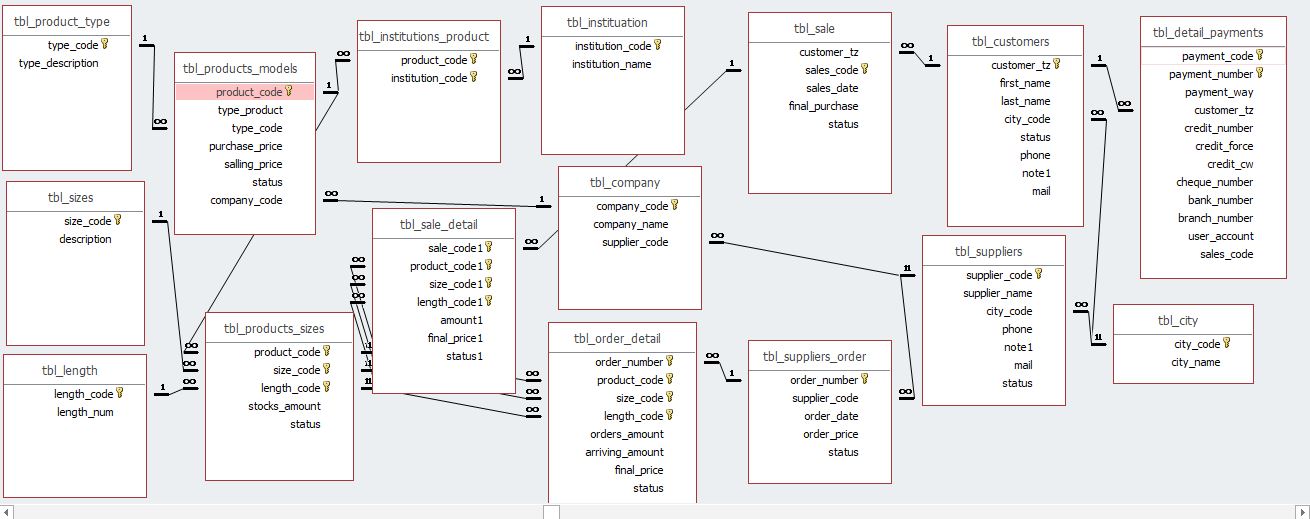
}

}

}

בסיס הנתונים

קשרי גומלין

****

טבלאות

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת לקוחות | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | customer\_tz | תעודת זהות | טקסט קצר | תקינות ת.ז. |
|  | last\_name | שם משפחה | טקסט קצר |  |
|  | first\_name | שם פרטי | טקסט קצר |  |
|  | city\_code | עיר | מספר |  |
|  | status | סטטוס | טקסט קצר |  |
|  | note1 | הערות | טקסט ארוך |  |
|  | phone | טלפון | טקסט קצר | תקינות מספר טלפון |
|  | mail | מייל | טקסט קצר | תקינות מייל |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת ערים | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | city\_code | קוד עיר | מספר |  |
|  | city\_name | שם עיר | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת חברות | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | company\_code | קוד חברה | מספר |  |
|  | company\_name | שם חברה | טקסט קצר |  |
|  | supplier\_code | קוד ספק | מספר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת מוסדות | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | institution\_code | קוד מוסד | מספר |  |
|  | institution\_name | שם מוסד | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת מוצרים למוסדות | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | product\_code | קוד מוצר | מספר |  |
| מפתח ראשי | institution\_code | קוד מוסד | מספר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת אורכים | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | length\_code | קוד אורך | מספר |  |
|  | length\_num | אורך | מספר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת פרטי הזמנה | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | order\_number | מספר הזמנה | מספר |  |
| מפתח ראשי | product\_code | קוד מוצר | מספר |  |
| מפתח ראשי | size\_code | קוד מידה | מספר |  |
| מפתח ראשי | length\_code | קוד אורך | מספר |  |
|  | orders\_amount | כמות מוזמנת | מספר |  |
|  | arriving\_amount | כמות מתקבלת | מספר |  |
|  | final\_price | סך הכל למוצר(כמות \*מחיר) | מספר |  |
|  | status | סטטוס | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת פרטי תשלום | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | payment\_code | קוד תשלום | מספר |  |
| מפתח ראשי | payment\_number | מספר תשלומים | מספר |  |
|  | payment\_way | אופן התשלום(אשראי,מזמן,צי'ק) | טקסט קצר |  |
|  | customer\_tz | מספר זהות של הלקוח | טקסט קצר | תקינות ת.ז. |
|  | credit\_number | מספר אשראי | מספר |  |
|  | credit\_force | תוקף הכרטיס | תאריך/שעה |  |
|  | credit\_cw | 3 ספרות בגב הכרטיס | מספר |  |
|  | cheque\_number | מספר המחאה | מספר |  |
|  | bank\_number | מספר בנק | טקסט קצר |  |
|  | branch\_number | מספר סניף | מספר |  |
|  | user\_account | מספר חשבון | טקסט קצר |  |
|  | sales\_code | קוד מכירה | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת מוצרים | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | product\_code | קוד מוצר | מספר |  |
|  | type\_product | תאור(שם) | טקסט קצר |  |
|  | type\_code | סוג(חולצה,חצאית) | מספר |  |
|  | purchase\_price | מחיר קניה | מספר |  |
|  | salling\_price | מחיר מכירה | מספר |  |
|  | status | סטטוס | טקסט קצר |  |
|  | company\_code | קוד חברה | מספר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת סוגי מוצרים | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | type\_code | קוד סוג | מספר |  |
|  | type\_description | תאור סוג(חצאית,חולצה) | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת מידות למוצר | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | product\_code | קוד מוצר | מספר |  |
| מפתח ראשי | size\_code | קוד מידה | מספר |  |
| מפתח ראשי | length\_code | קוד אורך | מספר |  |
|  | stocks\_amount | כמות במלאי | מספר |  |
|  | status | סטטוס(באיזה מידה קים) | מספר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת מידות | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | size\_code | קוד מידה | מספר |  |
|  | description | תאור | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת מכירות | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
|  | customer\_tz | מספר זהות של הלקוח | טקסט קצר | תקינות ת.ז. |
| מפתח ראשי | sales\_code | קוד מכירה | מספר |  |
|  | sales\_date | תאריך מכירה | תאריך/שעה |  |
|  | final\_purchase | סך קניה | מספר |  |
|  | status | סטטוס | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת הזמנות מספקים | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | order\_number | מספר הזמנה | מספר |  |
|  | supplier\_code | קוד ספק | מספר |  |
|  | order\_date | תאריך הזמנה | תאריך/שעה |  |
|  | order\_price | סך ההזמנה | מספר |  |
|  | status | סטטוס | טקסט קצר |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת פרטי מכירה | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | sale\_ code | קוד מכירה | מספר |  |
| מפתח ראשי | product\_code | קוד מוצר | מספר |  |
| מפתח ראשי | size\_code | קוד מידה | מספר |  |
| מפתח ראשי | length\_code | קוד אורך | מספר |  |
|  | amount | כמות | מספר |  |
|  | final\_price | amount\*price | מספר |  |
|  | status | זיכוי | כן/לא |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| טבלת ספקים | | | | |
|  | שם אנגלי | שם עברי | סוג | בדיקת תקינות |
| מפתח ראשי | supplier\_ code | קוד ספק | מספר |  |
|  | supplier\_name | שם ספק | טקסט קצר |  |
|  | city\_code | קוד עיר | מספר |  |
|  | phone | טלפון | טקסט קצר |  |
|  | note1 | הערות | טקסט ארוך |  |
|  | mail | מייל | טקסט קצר |  |
|  | status | סטטוס | טקסט קצר |  |

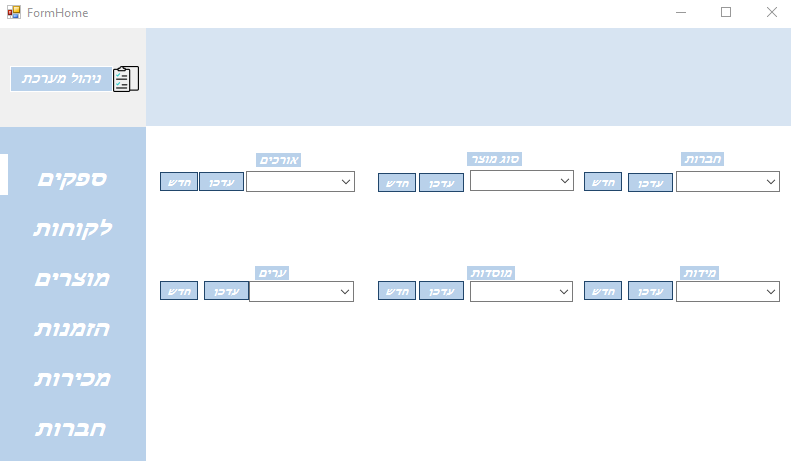
מדריך למשתמש

מסך קוד



במסך זה מקישים את הקוד על מנת להכנס לתוכנה

מסך ניהול מערכת



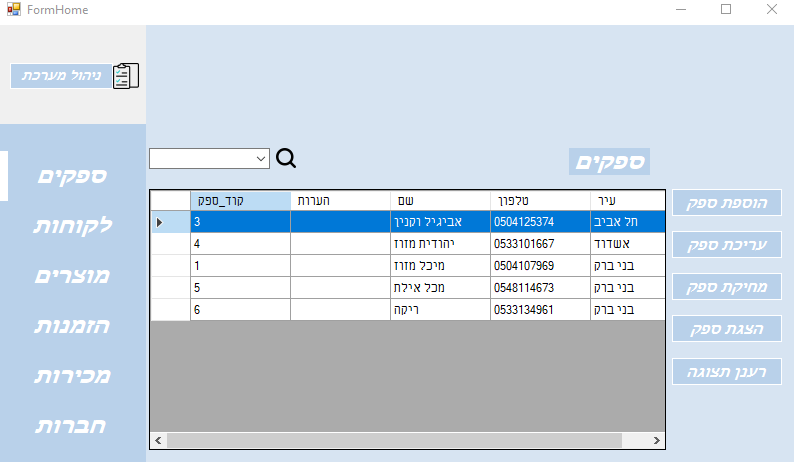
ניהול המערכת ע"י הוספת חברות/סוגי מוצרים/אורכים/מידות/מוסדות/ערים ועידכונם.

מסך הבית



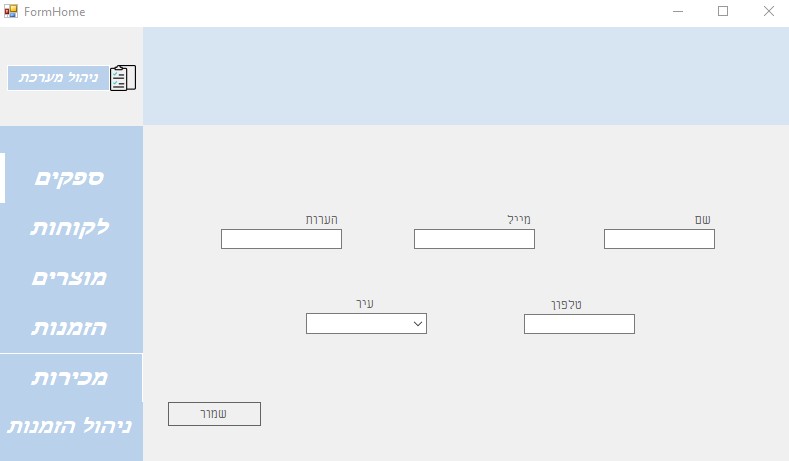
ממסך זה ניתן לגשת לספקים/לקוחות/מוצרים/הזמנות ומכירות.

מסך ספקים



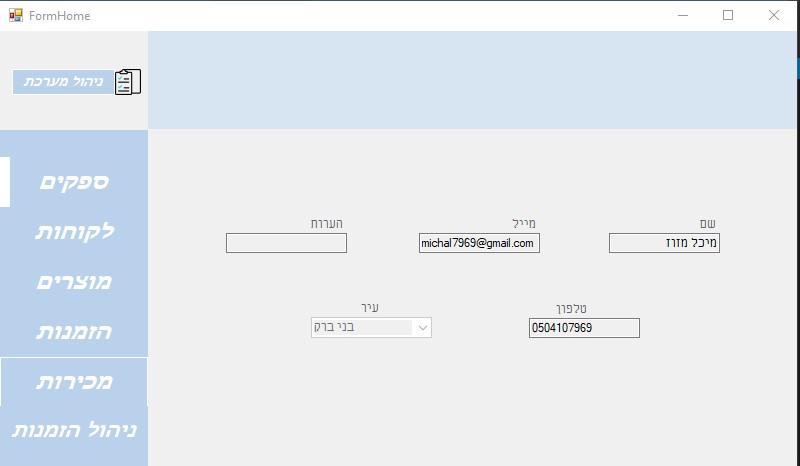
במסך זה ניתן לצפות בספקים ובפרטיהם וכן להוסיף/לעדכן/למחוק/להציג ספק וברענן ניתן לצפות בספקים הפעילים ובאילו שלא.

מסך משני הוספת ספק



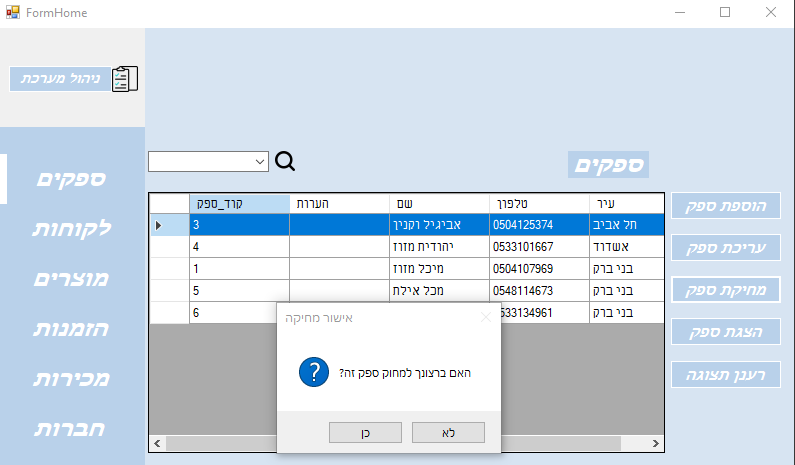
במסך זה מוסיפים ספק חדש.

מסך משני עדכון ספק



במסך זה מעדכנים את פרטי הספק.

מסך משני מחיקת ספק



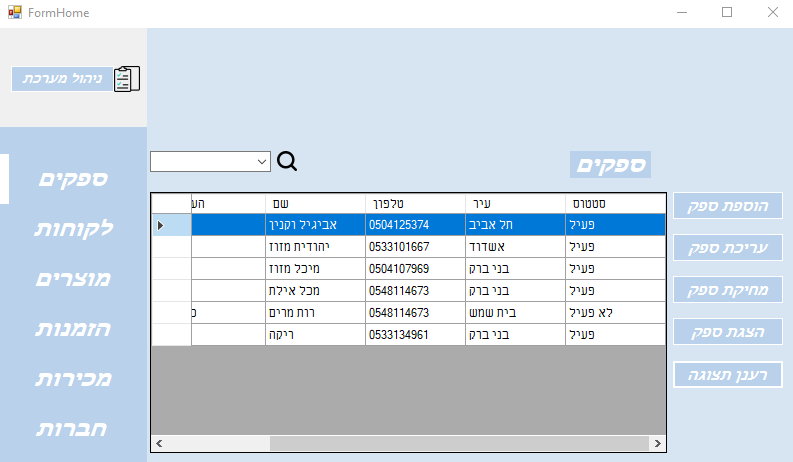
מסך זה מאפשר מחיקת ספק והפיכתו למצב לא פעיל.

מסך משני הצגת ספק



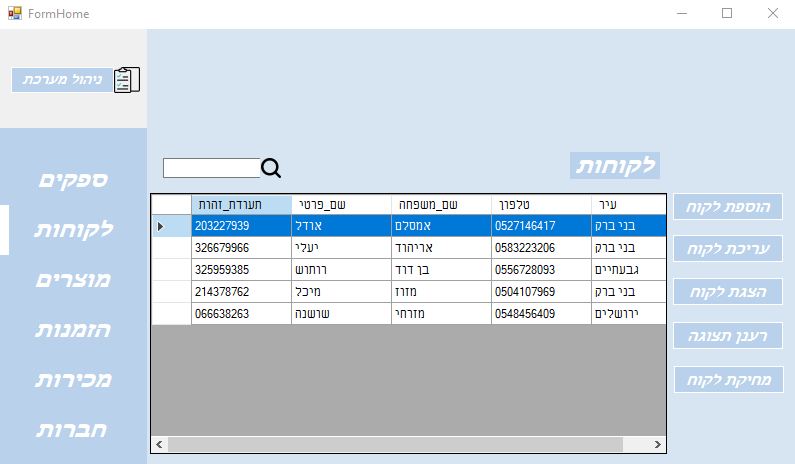
במסך זה מופיעים פרטי הספק.

מסך משני רענן תצוגה



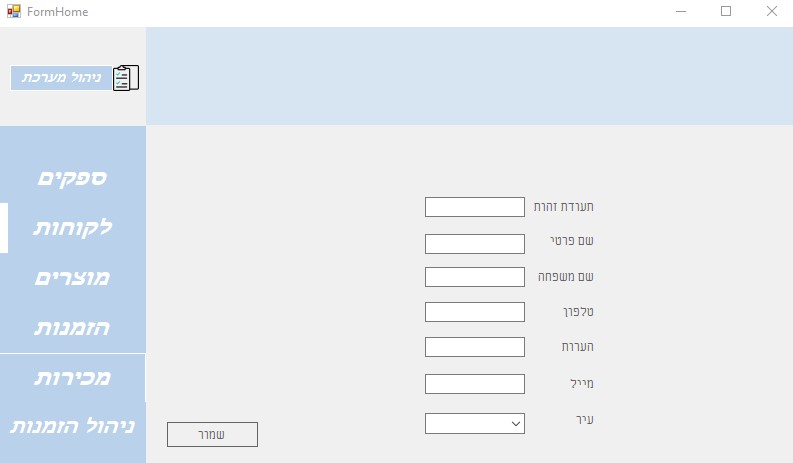
במסך זה ניתן לצפות בספקים הפעילים וכן באילו שלא.

מסך לקוחות



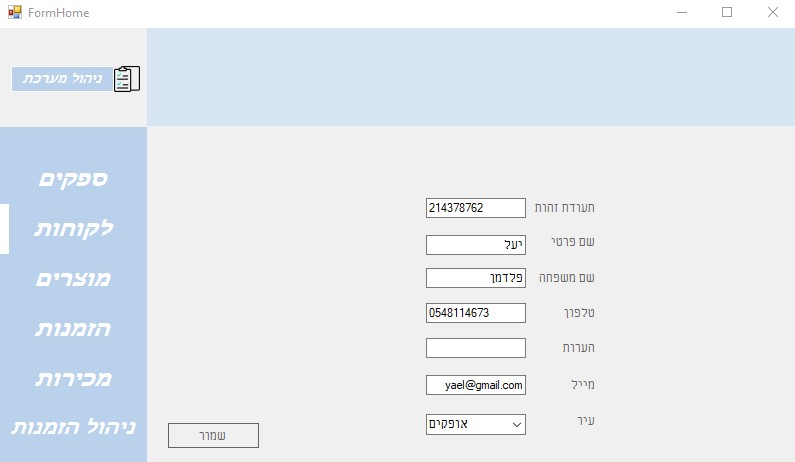
במסך זה ניתן לצפות בלקוחות ובפרטיהם וכן להוסיף/לעדכן/למחוק/להציג לקוח וברענן ניתן לצפות בלקוחות הפעילים ובאילו שלא.

מסך משני הוספת לקוח



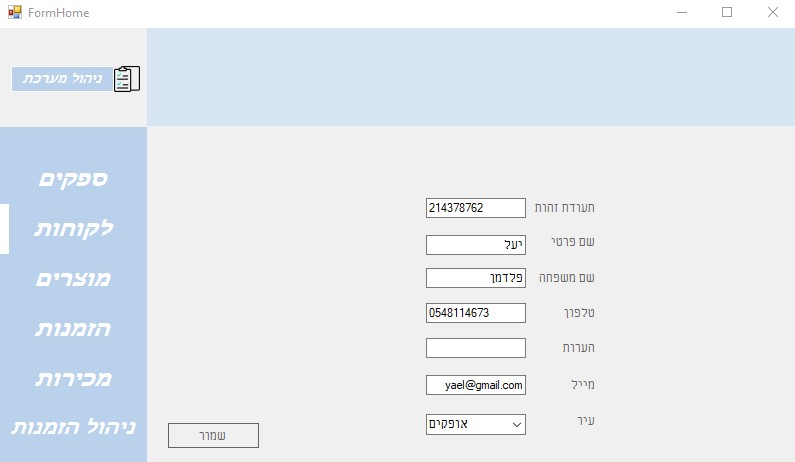
מסך זה מוסיף לקוח חדש.

מסך משני עדכון לקוח



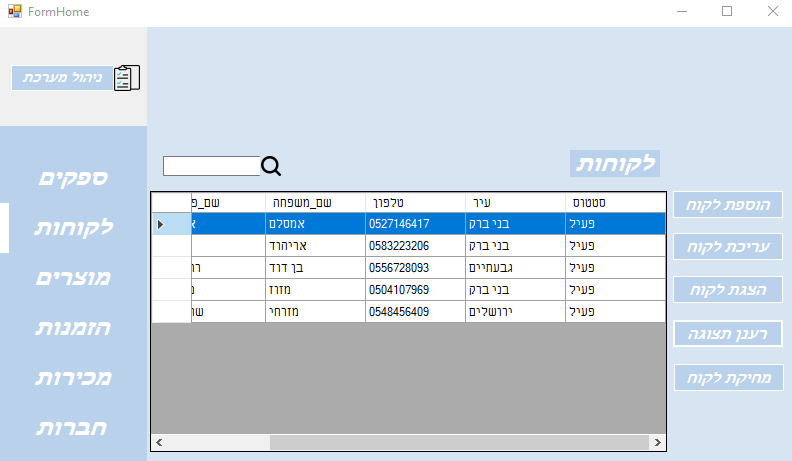
מסך זה מעדכן פרטי לקוח.

מסך משני הצגת לקוח



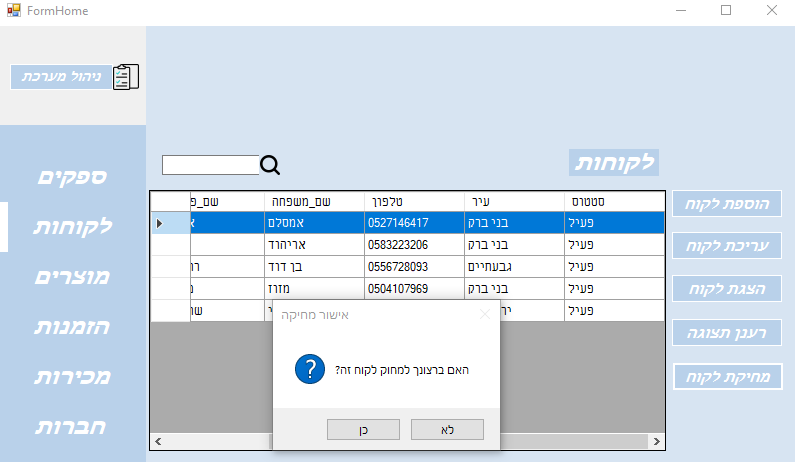
במסך זה מופיעים פרטי הלקוח.

מסך משני רענן תצוגה



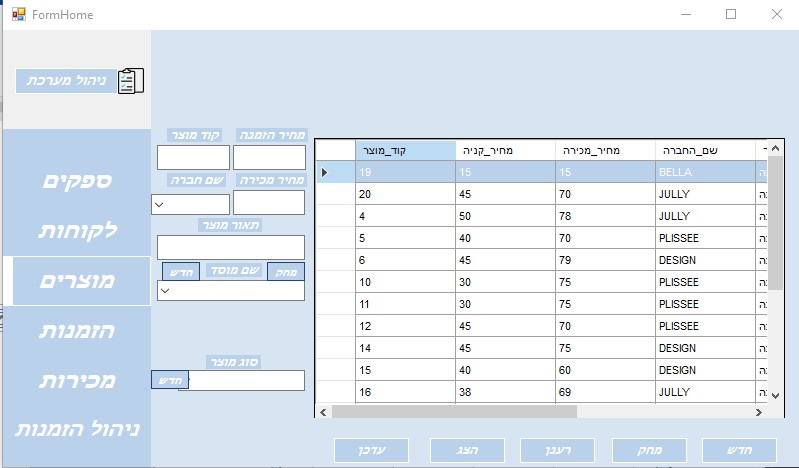
במסך זה מופיעים הלקוחות הפעילים ואת אילו שלא.

מסך משני מחיקת לקוח



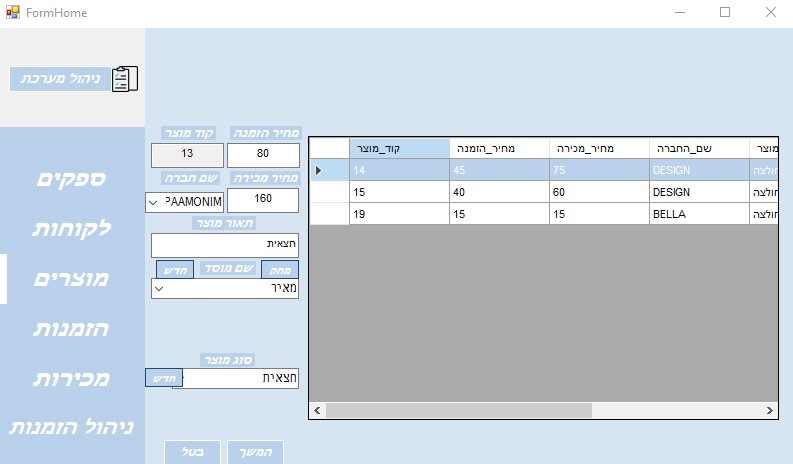
מסך זה מוחק לקוח מרשימת הלקוחות.

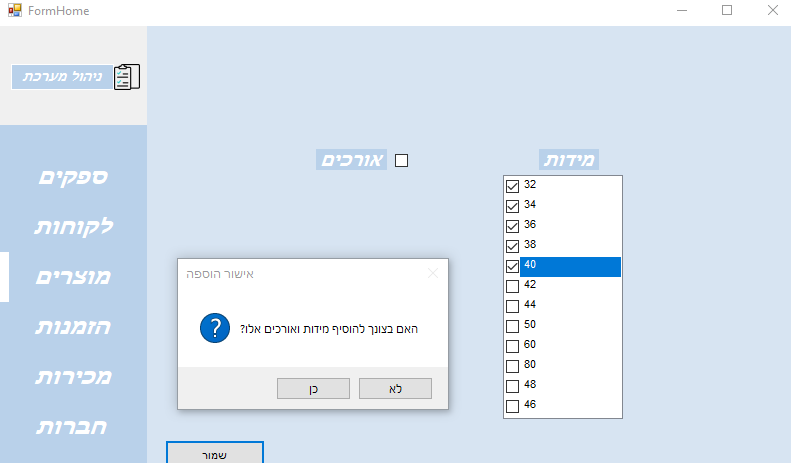
מסך מוצרים



במסך זה ניתן לערוך חיפוש על המוצר שרוצה וכן לבצע הוספה/מחק/הצג/עדכן וכן רענן שבו ניתן לצפות במוצרים הפעילים ובאילו שלא.

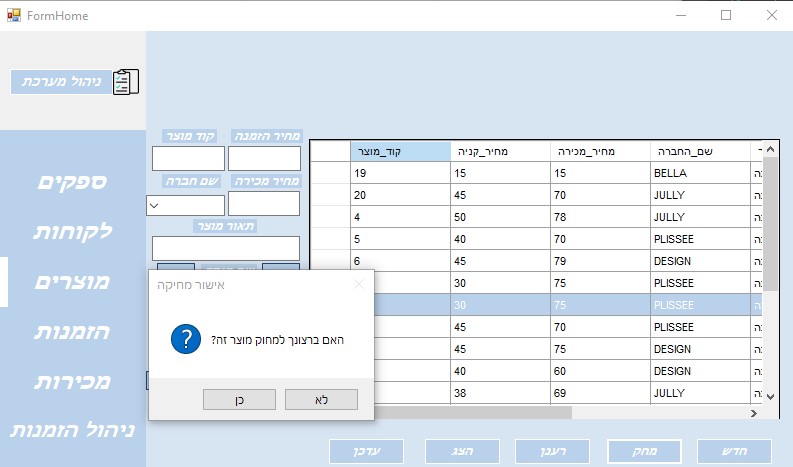
הוספת מוצר





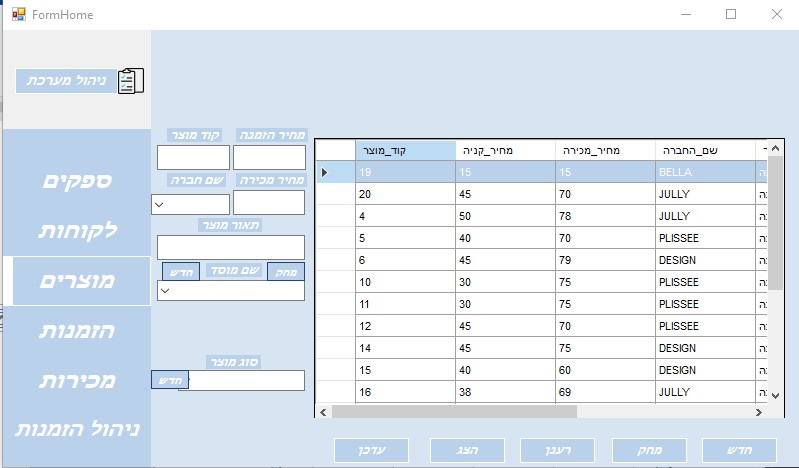
מסך זה מוסיף מוצר חדש וע"י לחיצה על מקש המשך ניפתח לי מסך נוסף של בחירת אורכים ומידות לכל מוצר.

מחיקת מוצר



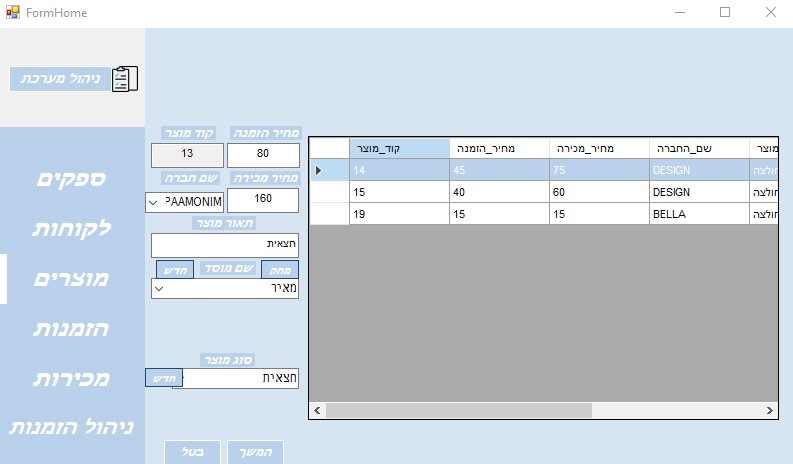
ניתן לבחור מוצר מטבלת מוצרים ולמחוק אותו.

ריענון תצוגה



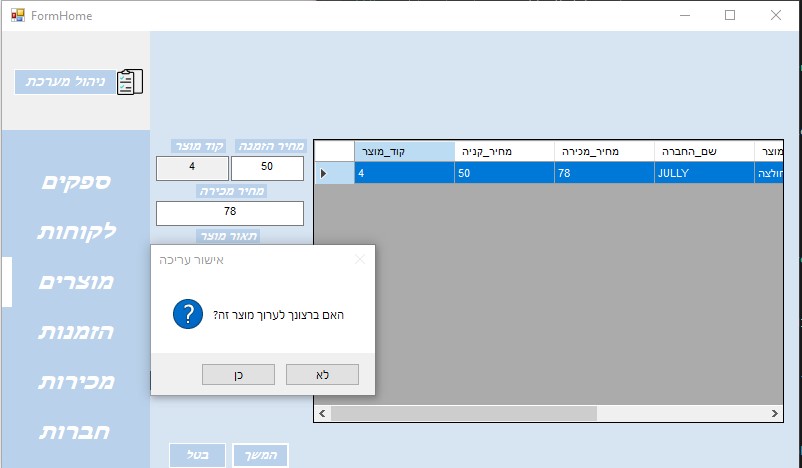
בלחיצה על רענן ניתן לצפות במוצרים פעילים וגם באילו שלא.

הצגת מוצר



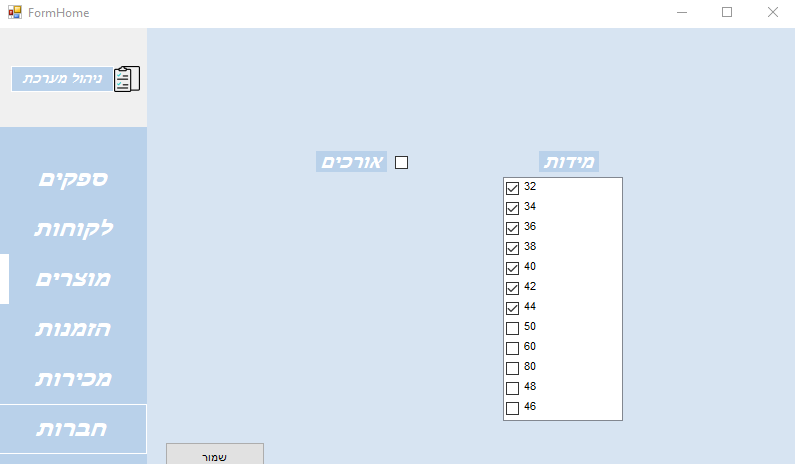
הצגת פרטי המוצר שנבחר בטבלה.

עריכת מוצר

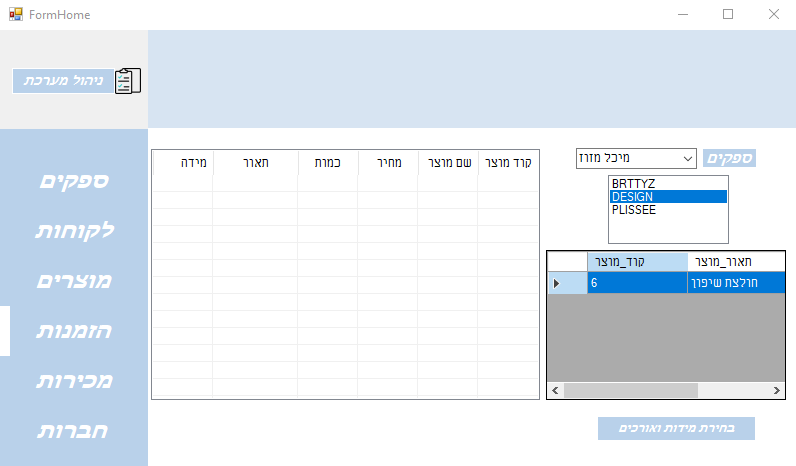


עריכת המוצר שנבחר בטבלת מוצרים וע"י לחיצה על המשך...

נפתח מסך נוסף לעדכון המידות והאורכים.

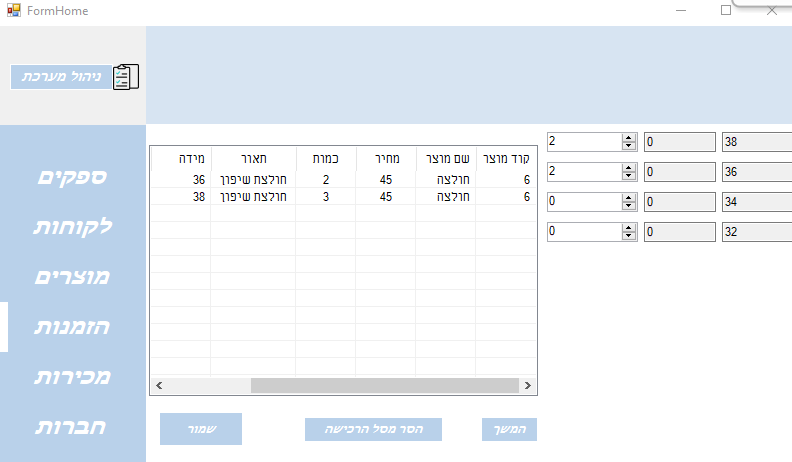


מסך הזמנות מספקים

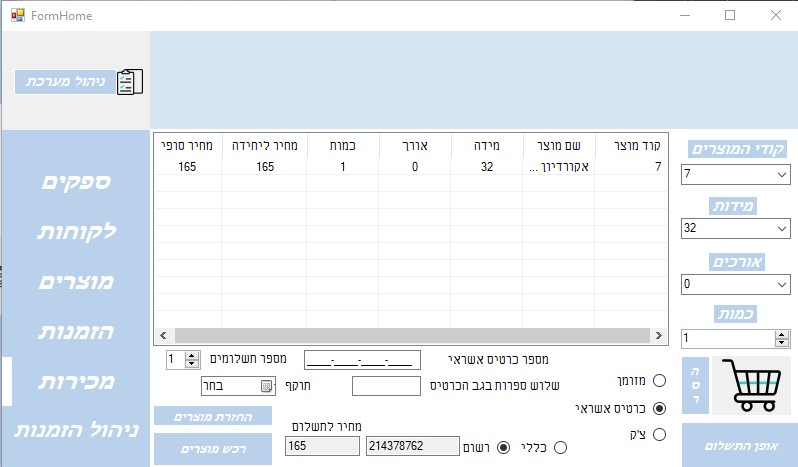


במסך זה בוחרים שם ספק שמעוניינים להזמין אצלו סחורה, ובכל בחירה של ספק מופיעים שמות החברות שהספק מתעסק עימם ואת המוצרים שיש מכל חברה.

ע"י לחיצה על לחצן בחירת מידות ואורכים נפתח על מסך זה חלק נוסף של בחירת הכמות מכל מידה ואורך של אותו מוצר.

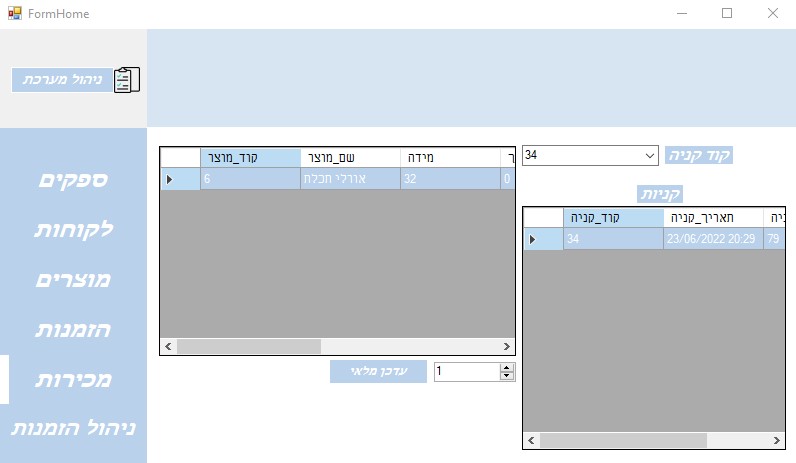


מסך קנית לקוח



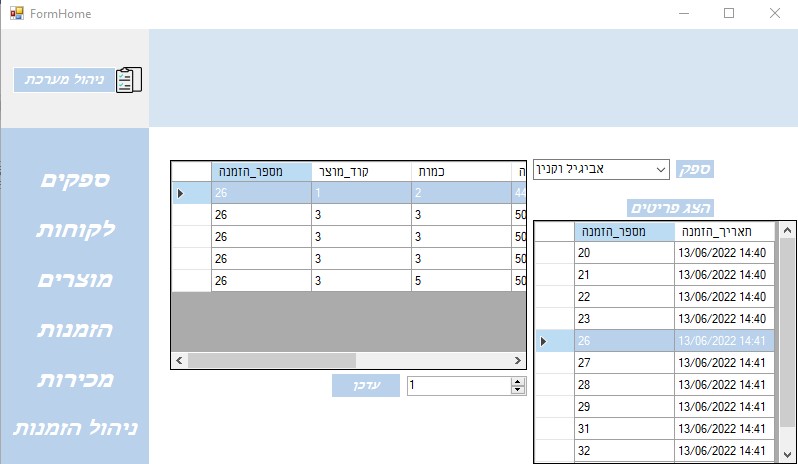
במסך זה מתבצע תהליך הרכישה של לקוח.

מסך החזרת מוצרים



במסך זה מתבצע תהליך של החזרת מוצרים

מסך ניהול הזמנות



במסך זה מתבצע עדכון הכמות שהגיע מכל מוצר בהזמנה שהזמינו מהספק.

עץ תהליכים:

רפלקציה:

העבודה על הפרויקט הייתה עבורי משימה מהנה ומאתגרת כאחד.

במהלך העבודה התעניינתי רבות בתחום זה,

נפעמתי לגלות עד כמה באמת יש חשיבות לתלבושת ולטיב איכותה.

הרי כולנו יודעות שרוב שנות הילדות והבגרות עוברות עלינו עם תלבושת...

אז למה לא להשקיע במוצרים איכותיים שישביעו את רצונן

של התלמידות ויאפשרו להם נוחות מקסימלית.

ברוך ד' אני מרוצה מהתוצאה ושמחה שבחרתי לעשות את הפרויקט בנושא זה ממנו למדתי

הרבה מה הן התהליכים והעמל הרב המושקע ביצור תלבושות והבנתי עד כמה ניהול עסק זה אתגר לא פחות מהכנת פרויקט, אולי זה לא בפן המאמץ המחשבתי אלא בפן הפיזי

במיוחד כאשר העסק פרטי שהמנהל הוא גם מזמין מספקים וגם מוכר ללקוחות.

וערכתי לגיסתי גדלה על כך שמנהלת בית ביד רמה וגם עסק ושום דבר לא בא על חשבון השני.