



מערכת לניהול תדלוק אוטומטי לציי רכב

המגישה: רחל עוברני

ת.ז.: 214862807

חלופה: אפליקציות אסינכרוניות

שנה: קיץ תשפ"ג



תוכן עניינים

2	מבוא
5	בסיס נתונים
5	קשרי גומלין
5	תיאור טבלאות
9	צד שרת
9	שכבת הModel
14	שכבת הViewModel
20	שכבת הSQLBuilder
22	שכבת שרותי רשת
34	Host
36	הרחבות
36	העברת קבצים
37	הצפנה
38	צד לקוח
46	מדריך למשתמש
46	ממשק לקוח
53	ממשק מנהל
60	סימולטור תדלוק
60	רפלקציה



מערכת לניהול תדלוק אוטומטי לצי רכב בס"ד

מבוא

הרקע לפרויקט

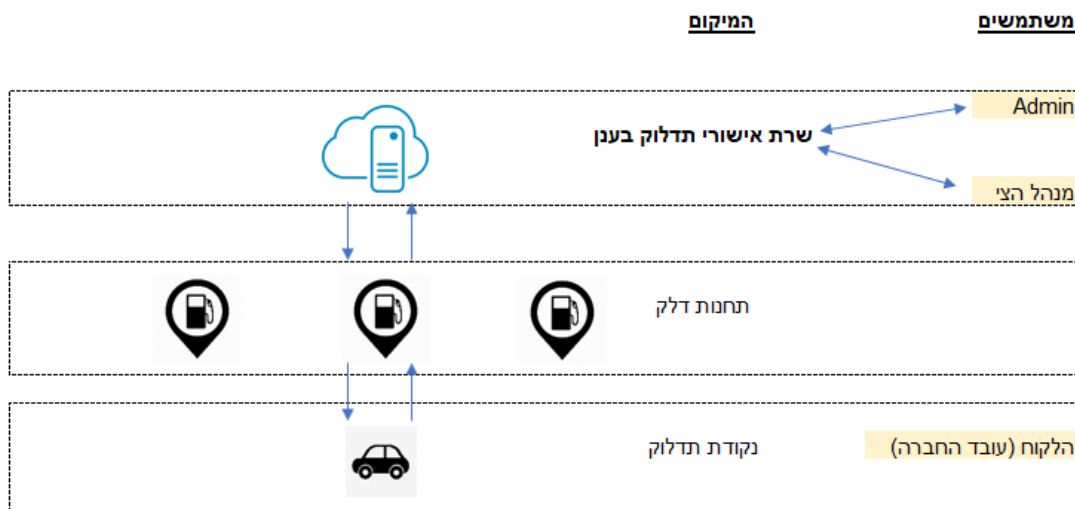
- שם הפרויקט
- מערך לניהול הוצאות הדלק בצי רכב.
- תיאור
- מאפשרת למנהלי צי רכב ניהול ושליטה בהוצאות הדלק של העובדים.
- קהל יעד
- חברות המחזיקות צי רכב.
- בחירת הנושא
- יש לי הכרות בתחום מכיוון שאבי עובד בנושא.

מטרות המערכת

- לאפשר למנהלי צי רכב לשלוט בהוצאות הדלק באמצעות הרשאות תדלוק והגבלות פר עובד / קבוצת עובדים.
- לאפשר למנהלי צי רכב לקבל מידע מרוכז באמצעות דוחות תקופתיים אודות צריכת הדלק של העובדים.
- מאפשר לעובדי החברה אמצעי נוח ויעיל לתדלוק עצמאי.

תאור המערכת

- המערכת נועדה לאפשר **לחברת הדלק** להציע **לחברות המחזיקות צי רכב** שירות **תדלוק אוטומטי**.
חברת הדלק, תדע לזהות את הלקוח שנכנס לתחנת הדלק באמצעות חיישן שיותקן ברכבו ותאפשר לו לתדלק באופן עצמאי ללא צורך בהמתנה ושימוש באמצעי תשלום.
בעת התדלוק, חברת הדלק תבדוק בזמן אמת את הרשאות התדלוק של הלקוח מול שרת הנמצא בענן ובהתאם לכך תאשר / תסרב לתדלוק.
חברת הדלק תאפשר למנהל הצי ממשק ניהול (מערכת זו) שתאפשר לו לקבוע את הרשאות והגבלות התדלוק של העובדים ואף דוחות פעילות שוטפת.
להלן דיאגרמה של מערכת :





בס"ד מערכת לניהול תדלוק אוטומטי לצ"י רכב

גבולות המערכת של הלקוח

- המערכת עוסקת:
 - החלת הרשאות והגבלות תדלוק על עובדי החברה.
 - קבלת מידע מרכז אודות הוצאות התדלוק של עובדי החברה.
 - ניהול התשלומים בין הצי וחברת הדלק.
- המערכת **איננה** עוסקת:
 - מערכת סליקה - לא ניתן לשלם באמצעות מערכת זאת.
 - ממשק ללקוח הסופי - עובד החברה לא יכול להיכנס למערכת כדי לצפות בפעילויות התדלוק שלו. הממשק של העובד הוא המסוף של משאבת הדלק. (לדוגמא, אם הוא חרג מהסכום שמותר לו, בעת ניסיון לתדלוק הוא יקבל הודעה מתאימה במסוף המשאבה).
 - (אך למען הנוחות ייצרתי סימולטור תדלוק כדי להראות דוגמא איך המערכת עובדת).

פירוט משתמשים

- **Admin – מנהל בחברת הדלת**
 - הקמת צ"י רכב חדשים (צי רכב פר חברה) + יצירת חשבון למנהלי הצי.
 - לקבוע האם הצי יהיה פעיל או לא.
 - לקבוע את הגבלות האשראי של הצי.
 - לקבוע את הנחות שלהם עבור צריכת דלק / מוצרים בתחנות.
 - להפיק דוחות פעילות תקופתיים.
- **מנהל צי רכב**
 - להוסיף רכבים חדשים לצי.
 - לקבוע חוקים והגבלות כמו : באיזה תחנות ניתן לתדלק, אילו מוצרים ניתן לרכוש והגבלות סכום בטווח יומי או חודשי.
 - להפיק דוחות פעילות שוטפים.
- **העובד בחברה**
 - לתדלק באופן עצמאי (בהתאם להגבלות התדלוק שנקבעו על מנהל הצי).

סביבת הפיתוח



בס"ד מערכת לניהול תדלוק אוטומטי לציי רכב

Visual Studio 2019

מסד נתונים Access 2019

שפות תכנות

צד שרת: C#, WCF

צד לקוח: UWP [Xaml + C#]

אתגרים מרכזיים:

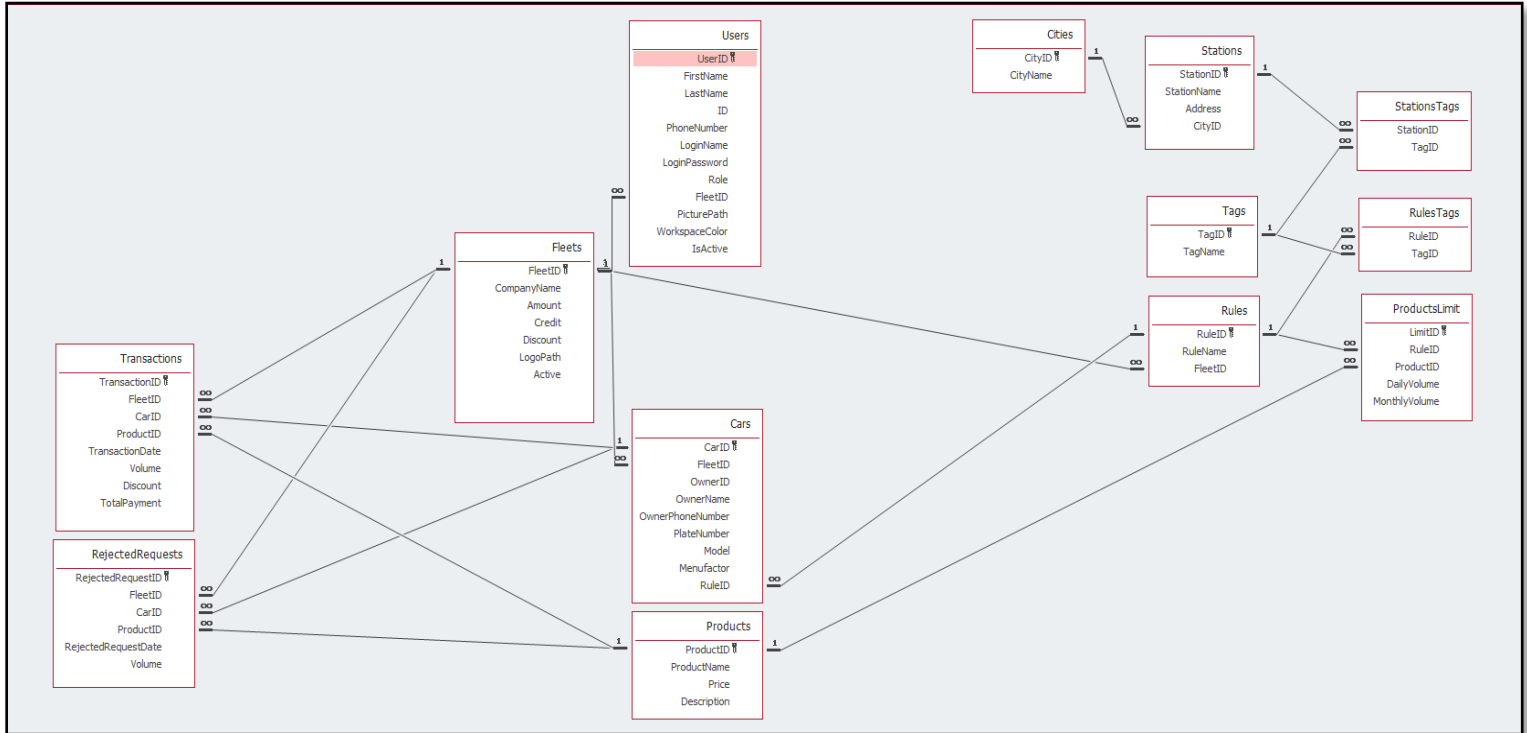
- אפיון המערכת. לשם כך למדתי את "עולם התוכן" הרלוונטי למשימה, אפיון התפקידים השונים (משתמשי המערכת (ועיצוב UI)
- תכנון המערכת באופן שיאפשר להתמודד עם ביצועי מערכת גבוהים שתאפשר מענה בזמן אמת לאלפי בקשות לאישור תדלוק בו זמנית
- פיתוח מערכת שעובדת באינטגרציה מול מערכות אחרות (בקר הדלק בתחנה שמבקש מהשרת אישורי תדלוק) כאשר אין לי את צד הלקוח בפועל (הייתי צריכה לכתוב סימולטור שידמה את הלקוח)



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

בסיס נתונים

קשרי גומלין



תיאור טבלאות

טבלת: Users משתמשים			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
UserID	קוד משתמש	מספר אוטומטי	
FirstName	שם פרטי	טקסט קצר	
LastName	שם משפחה	טקסט קצר	
ID	ת"ז	טקסט קצר	ת"ז תקינה עם 9 ספרות (ספרות בלבד)
PhoneNumber	מספר פלאפון	טקסט קצר	מספר פלאפון עם 10 ספרות (ספרות בלבד)
LoginName	שם כניסה	טקסט קצר	יחיד
LoginPassword	סיסמת כניסה	טקסט קצר	
Role	תפקיד	מספר	1) מנהל חברת דלק, 2) מנהל חברת עסקים
FleetID	קוד צי	מספר	קשר גומלין קיים
PicturePath	נתיב תמונת פרופיל	טקסט ארוך	
WorkspaceColor	צבע סביבת עבודה	טקסט קצר	
IsActive	פעיל?	כן/לא	



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

טבלת: Fleets ציים				
שם אנגלי	שם עברי	סוג	בדיקות תקינות	
FleetID	קוד צי	מספר אוטומטי		🔑
CompanyName	שם חברה	טקסט קצר		
Amount	סכום לתשלום	מטבע	מספר בלבד מעל 0	
Credit	זכות	מטבע	מספר חיובי בלבד	
Discount	הנחה	מספר	מספר בלבד מעל 0	
LogoPath	נתיב הלוגו	טקסט ארוך		
Active	פעיל?	כן/לא		

טבלת: Cars רכבים				
שם אנגלי	שם עברי	סוג	בדיקות תקינות	
CarID	קוד רכב	מספר אוטומטי		🔑
FleetID	קוד צי	מספר	קשר גומלין קיים	
OwnerName	שם בעל הרכב	טקסט קצר		
OwnerPhoneNumber	פלאפון בעל הרכב	טקסט קצר	פלאפון תקין עם 10 ספרות	
PlateNumber	מספר רישוי	טקסט קצר	יחיד	
Model	דגם	טקסט קצר		
Menufactor	יצרן	טקסט קצר		
RuleID	קוד חוק	מספר	קשר גומלין קיים	

טבלת: Rules חוקים				
שם אנגלי	שם עברי	סוג	בדיקות תקינות	
RuleID	קוד חוק	מספר אוטומטי		🔑
RuleName	שם חוק	טקסט קצר		
FleetID	קוד צי	מספר	קשר גומלין קיים	

טבלת: Stations תחנות				
שם אנגלי	שם עברי	סוג	בדיקות תקינות	
StationID	קוד תחנה	מספר אוטומטי		🔑
StationName	שם תחנה	טקסט קצר	עברית בלבד	
Address	כתובת	טקסט קצר		
CityID	קוד עיר	מספר	קשר גומלין קיים	



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

טבלת: Cities ערים			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
CityID	קוד עיר	מספר אוטומטי	
CityName	שם עיר	טקסט קצר	

טבלת: Tags תגים			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
TagID	קוד תג	מספר אוטומטי	
TagName	שם תג	טקסט קצר	

טבלת: Products מוצרים			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
ProductID	קוד מוצר	מספר אוטומטי	
ProductName	שם מוצר	טקסט קצר	
Price	מחיר	מטבע	מספר בלבד מעל 0
Description	תאור	טקסט ארוך	

טבלת: ProductsLimit הגבלת מוצרים			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
LimitID	קוד הגבלה	מספר אוטומטי	
RuleID	קוד חוק	מספר	קשר גומלין קיים
ProductID	קוד מוצר	מספר	קשר גומלין קיים
DailyVolume	כמות ליום	מספר	
MonthlyVolume	כמות לחודש	מספר	

טבלת: RejectedRequests בקשות שנדחו			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
RejectedRequestID	קוד בקשה דחיה	מספר אוטומטי	
FleetID	קוד צי	מספר	קשר גומלין קיים
CarID	קוד רכב	מספר	קשר גומלין קיים
ProductID	קוד מוצר	מספר	קשר גומלין קיים
RejectedRequestDate	תאריך בקשה דחיה	תאריך/שעה	
Volume	כמות	מספר	מספר בלבד מעל 0



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

טבלת: Transactions עסקאות			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
TransactionID	קוד עסקה	קוד אוטומטי	
FleetID	קוד צי	מספר	קשר גומלין קיים
CarID	קוד רכב	מספר	קשר גומלין קיים
ProductID	קוד מוצר	מספר	קשר גומלין קיים
TransactionDate	תאריך עסקה	תאריך/שעה	
Volume	כמות	מספר	מספר בלבד מעל 0
Discount	הנחה	מספר	מספר בלבד מעל 0
TotalPayment	סה"כ לתשלום	מטבע	מספר בלבד מעל 0

טבלת: RulesTags תגים של חוקים			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
RuleID	קוד חוק	מספר	קשר גומלין קיים
TagID	קוד תג	מספר	קשר גומלין קיים

טבלת: StationsTags תגים של תחנות			
שם אנגלי	שם עברי	סוג	בדיקות תקינות
StationID	קוד תחנה	מספר	קשר גומלין קיים
TagID	קוד תג	מספר	קשר גומלין קיים

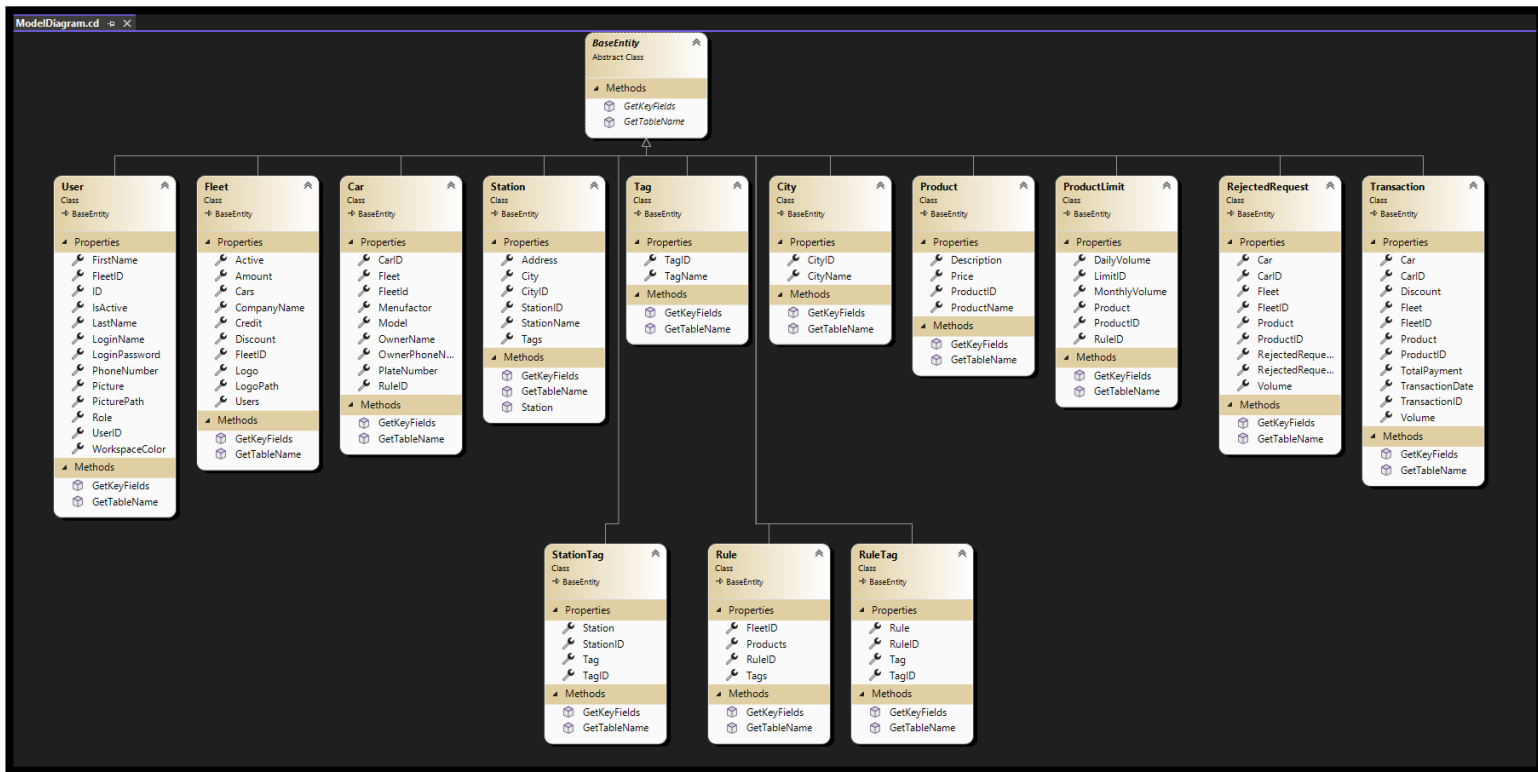


מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

צד שרת

שכבת Model

שכבת המודל היא שכבה המייצגת את מסד הנתונים (Access). כל טבלה מיוצגת באמצעות Class. הערכים שנשלפו ממסד הנתונים מאוחסנים באוספים של Classes אלו.



כמה מחלקות מתוך שכבת ה - Model

```
[DataContract]
public abstract class BaseEntity
{
    public abstract string GetTableName();
    public abstract string[] GetKeyFields();
}
```

• מחלקת BaseEntity
מחלקה המייצגת פריט בודד (שורה) מהאקסס, היא מחלקת הבסיס לכל מחלקות ה model.



• מחלקת User

מחלקה המייצגת משתמש במערכת, בין מנהל ובין לקוח.

ובו משתנים – Variable המחזיקים את פרטי המשתמש.

```
[DataContract]
public class User : BaseEntity
{
    [NoDB]
    [DataMember]
    public int UserID { get; set; }

    [DataMember]
    public string FirstName { get; set; }

    [DataMember]
    public string LastName { get; set; }

    [DataMember]
    public string ID { get; set; }

    [DataMember]
    public string PhoneNumber { get; set; }

    [DataMember]
    public string LoginName { get; set; }

    [DataMember]
    public string LoginPassword { get; set; }

    [DataMember]
    public int Role { get; set; }

    [DataMember]
    public int? FleetID { get; set; }

    public string PicturePath { get; set; }

    [DataMember]
    public Color WorkspaceColor { get; set; }

    [DataMember]
    public bool IsActive { get; set; }

    [NoDB]
    [DataMember]
    public byte[] Picture { get; set; }

    public override string GetTableName ()
    {
        return "Users";
    }

    public override string [] GetKeyFields()
    {
        return new string[] { "UserID" };
    }
}
```



• מחלקת Fleet

מחלקה המייצגת צי של רכבים.
ובו משתנים - Variable המחזיקים
את פרטי הצי.

```
[DataContract]
public class Fleet : BaseEntity
{
    [NoDB]
    [DataMember]
    public int FleetID { get; set; }

    [DataMember]
    public string CompanyName { get; set; }

    [DataMember]
    public double Amount { get; set; }

    [DataMember]
    public double Credit { get; set; }

    [DataMember]
    public double Discount { get; set; }

    [DataMember]
    public string LogoPath { get; set; }

    [DataMember]
    public bool Active { get; set; }

    [NoDB]
    [DataMember]
    public List<User> Users { get; set; }

    [NoDB]
    [DataMember]
    public byte[] Logo { get; set; }

    [NoDB]
    public List<Car> Cars { get; set; }

    public override string GetTableName()
    {
        return "Fleets";
    }

    public override string[] GetKeyFields()
    {
        return new string[] { "FleetID" };
    }
}
```



```
[DataContract]
public class Station : BaseEntity
{
    [NoDB]
    [DataMember]
    public int StationID { get; set; }

    [DataMember]
    public string StationName { get; set; }

    [DataMember]
    public string Address { get; set; }

    [DataMember]
    public int CityID { get; set; }

    [NoDB]
    [DataMember]
    public City City { get; set; }

    [NoDB]
    [DataMember]
    public List<Tag> Tags { get; set; }

    public override string GetTableName()
    {
        return "Stations";
    }

    public override string[] GetKeyFields()
    {
        return new string[] { "StationID" };
    }
}
```

• מחלקת Station

מחלקה המייצגת תחנה אחת
בחברת תדלוק.
ובו משתנים - Variable המחזיקים
את פרטי התחנה.



```
[DataContract]
public class Product : BaseEntity
{
    [NoDB]
    [DataMember]
    public int ProductID { get; set; }

    [DataMember]
    public string ProductName { get; set; }

    [DataMember]
    public double Price { get; set; }

    [DataMember]
    public string Description { get; set; }

    public override string GetTableName()
    {
        return "Products";
    }

    public override string[] GetKeyFields()
    {
        return new string[] { "ProductID" };
    }
}
```

• מחלקת **Product**

מחלקה המייצגת מוצר הקיים
בתחנת דלק.
ובו משתנים - Variable המחזיקים
את פרטי המוצר.

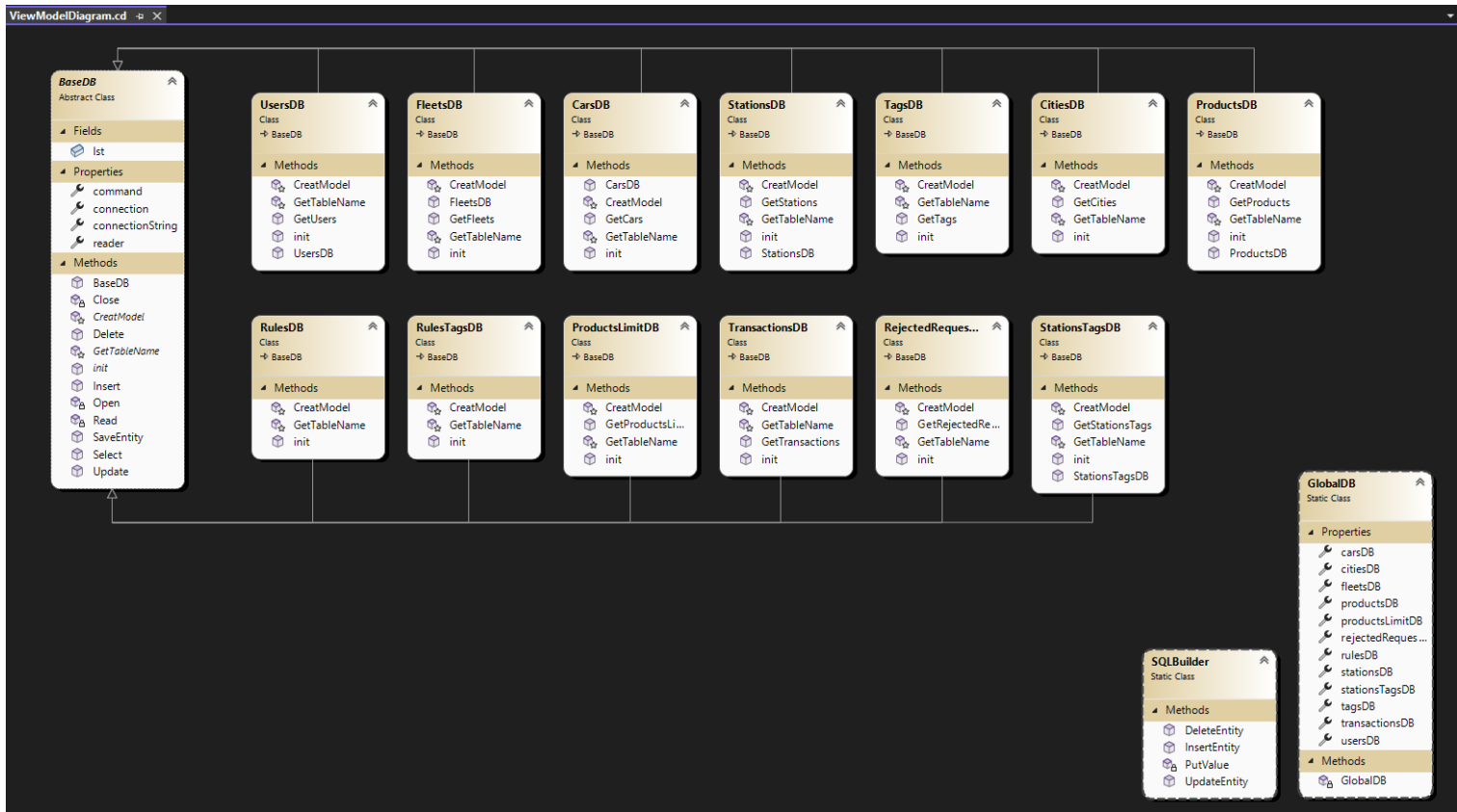


מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

שכבת ה ViewModel

שכבת זו, מתווכת בין שכבת המודל לבין המסד נתונים.

היא אחראית לטעון את הנתונים מהמסד נתונים בעזרת ה `SqlBilder` ולאחסן אותם בתוך השכבת ה `Model`.
וכן לשמור נתונים משכבת ה `Model` בעזרת ה `SqlBilder` במסד הנתונים.





• מחלקת BaseDB

מחלקה בסיסית ליצירת תקשורת
בין המסד הנתונים לפרויקט -
שליפה ושמירה של נתונים.

```
namespace ViewModel
{
    public abstract class BaseDB
    {
        public string connectionString { get; set; }
        public OleDbConnection connection { get; set; }
        public OleDbCommand command { get; set; }
        public OleDbDataReader reader { get; set; }

        public List<BaseEntity> lst;

        public BaseDB()
        {
            var dbFile = new FileInfo(@"..\..\..\DB\FulingManager.mdb");
            connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source='" +
                                dbFile.FullName + @'.mdb"';
            connection = new OleDbConnection(connectionString);
            command = new OleDbCommand();
            Select();
        }

        public void Select()
        {
            Open();
            Read();
            Close();
        }

        private void Open()
        {
            command.Connection = connection;

            // Open the connection to the access
            if (connection.State != ConnectionState.Open)
                connection.Open();
        }

        private void Read()
        {
            // Write a SQL command
            command.CommandText = "select * from " + GetTableName();

            // Premorm the command - the information return in an object
            reader = command.ExecuteReader();

            lst = new List<BaseEntity>();

            while (reader.Read())
            {
                lst.Add(CreatModel());
            }

            if (reader != null)
                reader.Close();
        }

        private void Close()
        {
            if (connection.State == ConnectionState.Open)
                connection.Close();
        }
    }
}
```




```
protected abstract BaseEntity CreatModel();
protected abstract string GetTableName();

public abstract void init();

public bool SaveEntity()
{
    Open();
    int result = command.ExecuteNonQuery();
    if (result > 0)
    {
        Read();
        init();
        Close();
        return true;
    }
    else
    {
        Close();
        return false;
    }
}

public bool Insert(BaseEntity entity)
{
    command.CommandText = SQLBuilder.InsertEntity(entity);
    return SaveEntity();
}

public bool Update(BaseEntity entity)
{
    command.CommandText = SQLBuilder.UpdateEntity(entity);
    return SaveEntity();
}

public bool Delete(BaseEntity entity)
{
    command.CommandText = SQLBuilder.DeleteEntity(entity);
    return SaveEntity();
}
}
```



• מחלקת GlobalDB

מחלקה סטטית שבה נשמרים
העצמים לאחר בנייתם בעת העלאת
הפרויקט.
וכך חוסכים על גישה דחופה לשרת.

```
namespace ViewModel
{
    public static class GlobalDB
    {
        public static UsersDB usersDB { get; set; }
        public static FleetsDB fleetsDB { get; set; }
        public static CarsDB carsDB { get; set; }
        public static ProductsDB productsDB { get; set; }
        public static CitiesDB citiesDB { get; set; }
        public static TagsDB tagsDB { get; set; }
        public static StationsTagsDB stationsTagsDB { get; set; }
        public static StationsDB stationsDB { get; set; }
        public static ProductsLimitDB productsLimitDB { get; set; }
        public static RulesTagsDB rulesTagsDB { get; set; }
        public static RejectedRequestsDB rejectedRequestsDB { get; set; }
        public static TransactionsDB transactionsDB { get; set; }

        public static RulesDB rulesDB { get; set; }

        static GlobalDB()
        {
            usersDB = new UsersDB();
            fleetsDB = new FleetsDB();
            carsDB = new CarsDB();
            tagsDB = new TagsDB();
            productsDB = new ProductsDB();
            productsLimitDB = new ProductsLimitDB();
            rulesTagsDB = new RulesTagsDB();
            citiesDB = new CitiesDB();
            stationsTagsDB = new StationsTagsDB();
            stationsDB = new StationsDB();
            rejectedRequestsDB = new RejectedRequestsDB();
            transactionsDB = new TransactionsDB();
            rulesDB = new RulesDB();

            fleetsDB.init();
            carsDB.init();
            usersDB.init();
            productsDB.init();
            citiesDB.init();
            tagsDB.init();
            stationsTagsDB.init();
            stationsDB.init();
            rejectedRequestsDB.init();
            transactionsDB.init();
            productsLimitDB.init();
            rulesTagsDB.init();
            rulesDB.init();
        }
    }
}
```



```
namespace ViewModel
{
    public class UsersDB : BaseDB

    public UsersDB() { }

    protected override BaseEntity CreatModel()
    {
        // Create a new User object
        User user = new User();

        // Fill the information from the access to the object
        user.UserID = Convert.ToInt32(reader["UserID"]);
        user.FirstName = reader["FirstName"].ToString();
        user.LastName = reader["LastName"].ToString();
        user.ID = reader["ID"].ToString();
        user.PhoneNumber = reader["PhoneNumber"].ToString();
        user.LoginName = reader["LoginName"].ToString();
        user.LoginPassword = reader["LoginPassword"].ToString();
        user.Role = Convert.ToInt32(reader["Role"]);

        var fleetID = reader["FleetID"].ToString();
        if (fleetID is "")
            user.FleetID = null;
        else
            user.FleetID = Convert.ToInt32(fleetID);

        user.PicturePath = reader["PicturePath"].ToString();

        // Convert hex to Color
        string colorFromDB = reader["WorkspaceColor"].ToString();
        user.WorkspaceColor = System.Drawing.ColorTranslator.FromHtml(colorFromDB);

        user.IsActive = Convert.ToBoolean(reader["IsActive"]);

        return user;
    }

    public override void init()
    {
        this.lst.ForEach((Action<BaseEntity>)(entity =>
        {
            User user = entity as User;
            if (user.PicturePath != "")
                user.Picture = File.ReadAllBytes(@"../../Images/" + user.PicturePath);

        }));
    }

    protected override string GetTableName()
    {
        return "Users";
    }

    public List<User> GetUsers()
    {
        return lst.Cast<User>().ToList();
    }
}
}
```



```
namespace ViewModel
{
    public class StationsDB : BaseDB
    {
        public StationsDB() { }

        protected override BaseEntity CreatModel()
        {
            // Create a new Fleet object
            Station station = new Station();

            // Fill the information from the access to the object
            station.StationID = Convert.ToInt32(reader["StationID"]);
            station.StationName = reader["StationName"].ToString();
            station.Address = reader["Address"].ToString();
            station.CityID = Convert.ToInt32(reader["CityID"]);

            return station;
        }
        public override void init()
        {
            this.lst.ForEach((Action<BaseEntity>)(entity =>
            {
                Station station = entity as Station;
                City city = GlobalDB.citiesDB.GetCities().FirstOrDefault(
                    x=> x.CityID == station.CityID);

                station.City = city;

                List<Tag> tags = GlobalDB.stationsTagsDB.GetStationsTags().Where(x =>
                    x.StationID == station.StationID).Select(x=>x.Tag).ToList();
                station.Tags = tags;
            }));
        }

        protected override string GetTableName()
        {
            return "Stations";
        }

        public List<Station> GetStations()
        {
            return lst.Cast<Station>().ToList();
        }
    }
}
```



שכבת SQLBuilder

מחלקה גנרית המשמשת לבניית שאילתות SQL על פי שם וסוג המאפיינים במחלקה.

קוד המחלקה:

```
public static class SQLBuilder
{
    public static string InsertEntity(BaseEntity entity)
    {
        Type type = entity.GetType();
        string command = "INSERT INTO " + entity.GetTableName() + " (";
        string values = " VALUES (";

        foreach (var property in type.GetProperties())
        {
            // שדה-שדה שאינו שייך ל DB
            // נשתמש בזה גם עבור שדה מזהה שאין צורך להקצות לו ערך
            if (property.GetCustomAttributes(typeof(NoDBAttribute), false).Length > 0)
                continue;

            object value = property.GetValue(entity);
            command += property.Name + " ,";
            values += PutValue(value) + " ,";
        }
        command = command.Substring(0, command.Length - 2) + ")";
        values = values.Substring(0, values.Length - 2) + ")";

        return command + values;
    }

    public static string DeleteEntity(BaseEntity entity)
    {
        Type type = entity.GetType();
        string command = "DELETE * FROM " + entity.GetTableName() + " WHERE ";

        string where = "";
        foreach (var keyField in entity.GetKeyFields())
        {
            if (where != "")
                where += " AND ";

            object value = entity.GetType().GetProperty(keyField).GetValue(entity);
            where += keyField + " = " + PutValue(value);
        }

        command += where;

        return command;
    }
}
```



```
public static string UpdateEntity(BaseEntity entity)
{
    Type type = entity.GetType();
    string command = "UPDATE " + entity.GetTableName() + " SET ";

    foreach (var property in type.GetProperties())
    {
        // שדה שאינו שייך ל DB
        // נשתמש בזה גם עבור שדה מזהה שאין צורך להקצות לו ערך
        if (property.GetCustomAttributes(typeof(NoDBAttribute), false).Length > 0)
            continue;

        object value = property.GetValue(entity);
        command += property.Name + " = " + PutValue(value) + " ,";
    }

    string where = "";
    foreach (var keyField in entity.GetKeyFields())
    {
        if (where != "")
            where += " AND ";

        object value = entity.GetType().GetProperty(keyField).GetValue(entity);
        where += keyField + " = " + PutValue(value);
    }

    command = command.Substring(0, command.Length - 2) + " WHERE " + where;

    return command;
}

private static string PutValue(object value)
{
    if (value == null)
        return "null";

    if (value is string || value is DateTime)
        return "'" + value + "'";

    if (value is Color)
    {
        Color c = (Color)value;

        return "'" + ColorTranslator.ToHtml(c) + "'";
    }

    if (value is BaseEntity)
    {
        BaseEntity e = value as BaseEntity;
        string k = e.GetKeyFields()[0];
        value = value.GetType().GetProperty(k).GetValue(value);
    }

    return value.ToString();
}
```



בס"ד מערכת לניהול תדלוק אוטומטי לציי רכב

שכבת שרותי רשת

שכבת שרותי הרשת בפרויקט נבנתה בטכנולוגיית WCF. בטכנולוגיה זו השירות כולל Interface + מימוש במחלקה.

שכבת שירותי הרשת משתמשת באובייקטים של שכבת הModel ובפונקציות שבשכבת ViewModel

בפרויקט זה ממשתי שני ממשקים עבור 2 סוגי לקוחות:

1. מנהל המערכת

הממשק כולל פונקציות לניהול הציים והגבלות התדלוק.
הגישה לשירות מתבצעת באמצעות תוכנת הניהול של מנהל הצי

2. הלקוח שבא לתדלק בתחנה

הממשק כולל פונקציות לבקשת אישור תדלוק ודיווח בסיום עסקה.
הגישה לשירות מתבצעת ע"י הבקר של משאבת התדלוק כאשר הלקוח מוציא את אקדח התדלוק ע"מ להתחיל בתהליך התדלוק.

קוד הInterface למנהל המערכת

```
using Model;
using System.Collections.Generic;
using System.ServiceModel;

namespace FuelingServices
{
    [ServiceContract]
    public interface IFuelingManagementService
    {
        // Login Page

        [OperationContract]
        List<User> GetUsers();

        [OperationContract]
        User GetUserByLoginName(string loginName);

        [OperationContract]
        void AddUser(User user);

        [OperationContract]
        void UpdateUser(User user);

        [OperationContract]
        void DeleteUser(User user);

        // Stations Page
        [OperationContract]
        List<Station> GetStations();

        [OperationContract]
        int GetStationsAmount();

        [OperationContract]
        List<Tag> GetTags();
    }
}
```



```
[OperationContract]
List<City> GetCities();

[OperationContract]
void AddStation(Station station);

[OperationContract]
void AddStationsTags(int stationId, List<Tag> tags);

[OperationContract]
void UpdateStation(Station station);

[OperationContract]
void UpdateStationsTags(int stationID, List<Tag> tags);

[OperationContract]
void DeleteStation(Station station);

// Fleets Page
[OperationContract]
List<Fleet> GetFleets();

[OperationContract]
int GetFleetsAmount();

[OperationContract]
void AddFleet(Fleet fleet);

[OperationContract]
void AddUsersToFleet(List<User> lstUsers);

[OperationContract]
void UpdateFleet(Fleet fleet);

[OperationContract]
void DeleteFleet(Fleet fleet);

// Products Page
[OperationContract]
int GetProductsAmount();

[OperationContract]
List<Product> GetProducts();

[OperationContract]
void AddProduct(Product product);

[OperationContract]
void UpdateProduct(Product product);

[OperationContract]
void DeleteProduct(Product product);

// Rules Page
[OperationContract]
List<Rule> GetFleetRules(int fleetId);

[OperationContract]
Rule GetRuleByID(int? ruleID);
```




```
[OperationContract]
int GetFleetRulesAmount(int fleetId);

[OperationContract]
void AddRule(Rule rule);

[OperationContract]
void UpdateRule(Rule rule);

[OperationContract]
void DeleteRule(Rule rule);

// WorkerCar Page
[OperationContract]
List<Car> GetFleetCars(int fleetId);

[OperationContract]
int GetFleetCarsAmount(int fleetId);

[OperationContract]
void AddWorkerCar(Car car);

[OperationContract]
void UpdateWorkerCar(Car car);

[OperationContract]
void DeleteWorkerCar(Car car);

// Transactions Page
[OperationContract]
List<Transaction> GetTransactions();

// RejectedRequest Page
[OperationContract]
List<RejectedRequest> GetRejectedRequests();

    }
}
```



```
public class FuelingManagmentService: IFuelingManagmentService
{
    // Login Page
    public List<User> GetUsers()
    {
        return GlobalDB.usersDB.GetUsers();
    }
    public User GetUserByLoginName(string loginName)
    {
        return GlobalDB.usersDB.GetUsers().FirstOrDefault(x => x.LoginName == loginName);
    }
    public void AddUser(User user)
    {
        GlobalDB.usersDB.Insert(user);
    }
    public void UpdateUser(User user)
    {
        if (user.Picture != null)
            File.WriteAllBytes(@"../../Images/" + user.PicturePath, user.Picture);

        GlobalDB.usersDB.Update(user);
    }
    public void DeleteUser(User user)
    {
        GlobalDB.usersDB.Delete(user);
    }

    // Stations Page
    public List<Station> GetStations()
    {
        return GlobalDB.stationsDB.GetStations();
    }
    public int GetStationsAmount()
    {
        return GlobalDB.stationsDB.GetStations().Count();
    }
    public List<Tag> GetTags()
    {
        return GlobalDB.tagsDB.GetTags();
    }
    public List<City> GetCities()
    {
        return GlobalDB.citiesDB.GetCities();
    }
    public void AddStation(Station station)
    {
        GlobalDB.stationsDB.Insert(station);
    }
    public void AddStationsTags(int stationID, List<Tag> tags)
    {
        StationTag st = new StationTag();
        st.StationID = stationID;

        foreach (var tag in tags as List<Tag>)
        {
            st.TagID = tag.TagID;
            GlobalDB.stationsTagsDB.Insert(st);
        }

        GlobalDB.stationsDB.GetStations().FirstOrDefault(x => x.StationID == stationID).Tags = tags;
    }
}
```



```
public void UpdateStation(Station station)
{
    GlobalDB.stationsDB.Update(station);
}

public void UpdateStationsTags(int stationID, List<Tag> tags)
{
    List<Tag> existingTags = GlobalDB.stationsDB.GetStations()
        .FirstOrDefault(x => x.StationID == stationID).Tags;

    StationTag st = new StationTag();
    st.StationID = stationID;

    // Add Tag to the StationsTagsList
    foreach (var tag in tags)
    {
        if (!existingTags.Exists(x => x.TagID == tag.TagID))
        {
            st.TagID = tag.TagID;
            GlobalDB.stationsTagsDB.Insert(st);
        }
    }

    // Delete Tag to the StationsTagsList
    foreach (var tag in existingTags)
    {
        if (!tags.Exists(x => x.TagID == tag.TagID))
        {
            st.TagID = tag.TagID;
            GlobalDB.stationsTagsDB.Delete(st);
        }
    }
}

public void DeleteStation(Station station)
{
    GlobalDB.stationsDB.Delete(station);
}

// Fleets Page
public List<Fleet> GetFleets()
{
    return GlobalDB.fleetsDB.GetFleets();
}

public int GetFleetsAmount()
{
    return GlobalDB.fleetsDB.GetFleets().Count();
}

public void AddFleet(Fleet fleet)
{
    // save the logo in the local folder
    if (fleet.Logo != null)
        File.WriteAllBytes(@"../../Images/" + fleet.LogoPath, fleet.Logo);

    GlobalDB.fleetsDB.Insert(fleet);
}

public void AddUsersToFleet(List<User> lstUsers)
{
    foreach (var user in lstUsers)
    {
        GlobalDB.usersDB.Insert(user);
    }

    Fleet f = GlobalDB.fleetsDB.GetFleets().FirstOrDefault(x => x.FleetID ==
        lstUsers[0].FleetID);

    if (f.Users == null)
        f.Users = new List<User>();
    f.Users = lstUsers;
}
```



```
public void UpdateFleet(Fleet fleet)
{
    if (fleet.Logo != null)
        File.WriteAllBytes(@"../../Images/" + fleet.LogoPath, fleet.Logo);

    GlobalDB.fleetsDB.Update(fleet);
}
public void DeleteFleet(Fleet fleet)
{
    GlobalDB.fleetsDB.Delete(fleet);
}

// Products Page
public int GetProductsAmount()
{
    return GlobalDB.productsDB.GetProducts().Count();
}
public List<Product> GetProducts()
{
    return GlobalDB.productsDB.GetProducts();
}
public void AddProduct(Product product)
{
    GlobalDB.productsDB.Insert(product);
}
public void UpdateProduct(Product product)
{
    GlobalDB.productsDB.Update(product);
}
public void DeleteProduct(Product product)
{
    GlobalDB.productsDB.Delete(product);
}

// Rules Page
public List<Rule> GetFleetRules(int fleetId)
{
    return GlobalDB.rulesDB.GetRules().Where(x => x.FleetID == fleetId).ToList();
}
public int GetFleetRulesAmount(int fleetId)
{
    return GlobalDB.rulesDB.GetRules().Where(x => x.FleetID == fleetId).Count();
}
public Rule GetRuleByID(int? ruleID)
{
    if (ruleID == null)
        return new Rule();

    return GlobalDB.rulesDB.GetRules().FirstOrDefault(x => x.RuleID == ruleID);
}
public void AddRule(Rule rule)
{
    GlobalDB.rulesDB.Insert(rule);

    int currentRuleID = GlobalDB.rulesDB.GetRules().Max(x => x.RuleID);
```



```
// הוספת תגים לחוק
foreach (var tag in rule.Tags)
{
    RuleTag rt = new RuleTag();
    rt.RuleID = currentRuleID;
    rt.TagID = tag.TagID;
    Insert(rt);
}

GlobalDB.rulesTagsDB.                // הוספת מוצרים מוגבלים לחוק
foreach (var productLimit in rule.LimitedProducts)
{
    ProductLimit pl = productLimit;
    pl.RuleID = currentRuleID;
    //pl.ProductID = productLimit.Product.ProductID;

    GlobalDB.productsLimitDB.Insert(pl);
}

// רשימת התגים והמוצרים המוגבלים
// נכנסים ב init
// ובגלל שההוספה של החוק הראשונה ורק לאחר מכן מתבצעת ההוספה של התגים והמוצרים
// לכן יש לקרוא לפונקציה init
GlobalDB.rulesDB.init();
}

public void UpdateRule(Rule rule)
{
    // מחיקת התגים הקיימים בחוק זה
    List<RuleTag> oldRuleTags = GlobalDB.rulesTagsDB.GetRulesTags().
                                Where(x => x.RuleID == rule.RuleID).ToList();
    foreach (var ruleTag in oldRuleTags)
    {
        GlobalDB.rulesTagsDB.Delete(ruleTag);
    }

    // מחיקת מוצרים מוגבלים הקיימים בחוק זה
    List<ProductLimit> oldProductLimit =
        GlobalDB.productsLimitDB.GetProductsLimits().
        Where(x => x.RuleID == rule.RuleID).ToList();
    foreach (var productLimit in oldProductLimit)
    {
        GlobalDB.productsLimitDB.Delete(productLimit);
    }

    // הוספת תגים לחוק
    foreach (var tag in rule.Tags)
    {
        RuleTag rt = new RuleTag();
        rt.RuleID = rule.RuleID;
        rt.TagID = tag.TagID;

        GlobalDB.rulesTagsDB.Insert(rt);
    }

    // הוספת מוצרים מוגבלים לחוק
    foreach (var productLimit in rule.LimitedProducts)
    {
        ProductLimit pl = productLimit;
        pl.RuleID = rule.RuleID;
        //pl.ProductID = productLimit.Product.ProductID;

        GlobalDB.productsLimitDB.Insert(pl);
    }
}
```



```
// רשימת התגים והמוצרים המוגבלים
// init ב נכנסים
// ובגלל שההוספה של החוק הראשונה ורק לאחר מכן מתבצעת ההוספה של התגים והמוצרים
// Init לכן יש לקרוא לפונקציית
GlobalDB.rulesDB.init();
}

public void DeleteRule(Rule rule)
{
    // מחיקת התגים הקיימים בחוק זה
    if (rule.Tags != null)
    {
        List<RuleTag> oldRuleTags = GlobalDB.rulesTagsDB.GetRulesTags().
            Where(x => x.RuleID == rule.RuleID).ToList();
        foreach (var ruleTag in oldRuleTags)
        {
            GlobalDB.rulesTagsDB.Delete(ruleTag);
        }
    }

    // מחיקת מוצרים מוגבלים הקיימים בחוק זה
    if (rule.LimitedProducts != null)
    {
        List<ProductLimit> oldProductLimit = GlobalDB.productsLimitDB.
            GetProductsLimits().Where(x => x.RuleID == rule.RuleID).ToList();

        foreach (var productLimit in oldProductLimit)
        {
            GlobalDB.productsLimitDB.Delete(productLimit);
        }
    }

    // מחיקת החוק מכל הרכבים הקשורים לחוק זה
    List<Car> relatedCars = GetFleetCars(rule.FleetID);
    foreach (Car car in relatedCars)
    {
        car.RuleID = null;
        GlobalDB.carsDB.Update(car);
    }

    GlobalDB.rulesDB.Delete(rule);
}

// WorkerCars Page
public int GetFleetCarsAmount(int fleetId)
{
    return GlobalDB.carsDB.GetCars().Where(x => x.FleetID == fleetId).Count();
}

public List<Car> GetFleetCars(int fleetId)
{
    return GlobalDB.carsDB.GetCars().Where(x => x.FleetID == fleetId).ToList();
}

public void AddWorkerCar(Car car)
{
    GlobalDB.carsDB.Insert(car);
}

public void UpdateWorkerCar(Car car)
{
    GlobalDB.carsDB.Update(car);
}

public void DeleteWorkerCar(Car car)
{
    GlobalDB.carsDB.Delete(car);
}
```



```
// Transactions Page
public List<Transaction> GetTransactions()
{
    return GlobalDB.transactionsDB.GetTransactions();
}

// RejectedRequest Page
public List<RejectedRequest> GetRejectedRequests()
{
    return GlobalDB.rejectedRequestsDB.GetRejectedRequests();
}
}
```



מערכת לניהול תדלוק אוטומטי לצ"י רכב בס"ד

קוד ה Interface ללקוח שבא לתדלק

```
using Model;
using System.ServiceModel;

namespace FuelingServices
{
    [ServiceContract]
    public interface IFuelingAuthorizationService
    {
        // FuellingSimulator
        [OperationContract]
        FuellingRespond FuellingRequest(FuellingRequest request);
    }
}
```

קוד מחלקת המימוש לשירות

```
using Model;
using System;
using System.Collections.Generic;
using System.Linq;
using ViewModel;

namespace FuelingServices
{
    public class FuelingAuthorizationService : IFuelingAuthorizationService
    {
        // FuellingSimulator
        public FuellingRespond FuellingRequest(FuellingRequest request)
        {
            FuellingRespond respond = new FuellingRespond();
            Car currentCar = GlobalDB.carsDB.GetCars().FirstOrDefault(x =>
                x.PlateNumber == request.PlateNumber);

            double allowAmount = GetAllowAmount(currentCar, request);

            if (allowAmount > 0)
            {
                respond.IsAllowed = true;
                respond.Amount = allowAmount;
            }
            else
            {
                respond.IsAllowed = false;
                respond.Amount = 0;
                respond.Explanation = "מוצר זה לא בחוק.";

                // דווח על בקשה שנדחתה
                RejectedRequest rejected = new RejectedRequest();
                rejected.CarID = currentCar.CarID;
                rejected.FleetID = GlobalDB.fleetsDB.GetFleets().
                    FirstOrDefault(x => x.FleetID == currentCar.FleetID).FleetID;
                rejected.RejectedRequestDate = DateTime.Now;
                rejected.ProductID = request.ChosenProduct.ProductID;
            }

            return respond;
        }
    }
}
```




```
private double GetAllowAmount(Car currentCar, FuellingRequest request)
{
    // אם התחנה לא קיימת בחוק
    foreach (var tag in currentCar.Rule.Tags)
    {
        if (!request.GasStation.Tags.Exists(x => x.TagID == tag.TagID))
            return 0;
    }

    // אם המוצר לא קיים בחוק
    ProductLimit p = currentCar.Rule.LimitedProducts.FirstOrDefault(x =>
        x.ProductID == request.ChosenProduct.ProductID);

    if (p == null)
    {
        return 0;
    }

    List<Transaction> lst = GetTransactionsByCarAndProductID(currentCar.CarID,
        request.ChosenProduct.ProductID);

    double dailyVolumeUsed = 0;
    double monthlyVolumeUsed = 0;

    // בודק שימוש
    foreach (var tran in lst)
    {
        // בחודש הנוכחי
        if (tran.TransactionDate.Year == DateTime.Now.Year &&
            tran.TransactionDate.Month == DateTime.Now.Month)
        {
            // ביום הנוכחי
            if (tran.TransactionDate.Day == DateTime.Now.Day)
            {
                dailyVolumeUsed += tran.Volume;
            }

            monthlyVolumeUsed += tran.Volume;
        }
    }

    // חריגה מהכמות המותרת
    if (monthlyVolumeUsed >= p.MonthlyVolume ||
        dailyVolumeUsed >= p.DailyVolume)
    {
        return 0;
    }

    var result = (p.DailyVolume - dailyVolumeUsed <= p.MonthlyVolume -
        monthlyVolumeUsed) ?
        p.DailyVolume - dailyVolumeUsed : p.MonthlyVolume - monthlyVolumeUsed;

    return result;
}
```



```
private List<Transaction> GetTransactionsByCarAndProductID
                                (int carID, int productID)
{
    List<Transaction> lst = new List<Transaction>();

    foreach (var tran in GlobalDB.transactionsDB.GetTransactions())
    {
        if (tran.CarID == carID && tran.ProductID == productID)
        {
            lst.Add(tran);
        }
    }
    return lst;
}
}
```



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

Host

כדי ששרותי הרשת שבניתי יוחזקו באויר וניתן יהיה לגשת אליהם מהלקוח, בניתי אפליקצית Host בצד שרת. שתפקידה להפעיל את שירות הרשת ולהשאירו פתוח לגישה. האפליקציה נכתבה ב-Console Application.

קוד התוכנית

```
namespace Host
{
    internal class Program
    {
        static void Main(string[] args)
        {
            //CheckDB dbd = new CheckDB();
            Console.WriteLine("Welcome to the server side...");
            ServiceHost FuelingManagerService = new
                ServiceHost(typeof(FuelingManagmentService));
            ServiceHost AuthorizationService = new
                ServiceHost(typeof(FuelingAuthorizationService));
            FuelingManagerService.Open();
            AuthorizationService.Open();
            Console.ReadLine();
        }
    }
}
```

התוכנית נדרשת להשתמש במידע אודות שירותי הרשת

מידע זה נמצא בקובץ App.config בפרויקט Host

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>

  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="Binding1"
          maxReceivedMessageSize = "100000000"
          security mode="None" />
      </binding>
    </basicHttpBinding>
  </bindings>

  <services>
    <service name="FuelingServices.FuelingManagmentService">
      <endpoint address="FuelingManagmentService"
        binding="basicHttpBinding"
        bindingConfiguration="Binding1"
        contract="FuelingServices.IFuelingManagmentService">
        <identity>
          <dns value="localhost" />
        </identity>
      </endpoint>
      <endpoint address="mex" binding="mexHttpBinding"
        contract="IMetadataExchange" />
    </service>
  </services>
</system.serviceModel>
</configuration>
```



```
<host>
    <baseAddresses>
        <add
baseAddress="http://localhost:8733/FuelingManagerService/" />
        </baseAddresses>
    </host>
</service>

<service name="FuelingServices.FuelingAuthorizationService">
    <endpoint address="FuelingAuthorizationService"
        binding="basicHttpBinding"
        bindingConfiguration="Binding1"

contract="FuelingServices.IFuelingAuthorizationService">
        <identity>
            <dns value="localhost" />
        </identity>
    </endpoint>

        <endpoint address="mex" binding="mexHttpBinding"
contract="IMetadataExchange" />
    </host>
        <baseAddresses>
            <add
baseAddress="http://localhost:8733/AuthorizationService/" />
            </baseAddresses>
        </host>
    </service>
</services>
<behaviors>
    <serviceBehaviors>
        <behavior>
            <serviceMetadata httpGetEnabled="True"
httpsGetEnabled="True"/>
            <serviceDebug includeExceptionDetailInFaults="True" />
        </behavior>
    </serviceBehaviors>
</behaviors>
</system.serviceModel>

</configuration>
```



הרחבות העברת קבצים

במהלך מימוש הפרויקט אפשרתי העלאת תמונות ושמירתן, כגון - בחירת תמונת פרופיל וכן הוספת לוגו בפתיחת צי רכבים חדש.

אראה כמה שלבים למימוש פעולה זאת :

1. לאפשר למשתמש לפתוח את ה Explorer ולבחור תמונה
2. להמיר את התמונה ל `Byte[]` ולהעבירם לשרת
3. השרת שמקבל את ה `Byte[]` ממיר אותם לתמונה, שומר אותה בתקיה אצלו ולמסד הנתונים מכניס את שמה. וכן להפך.
(בקריאת התמונה מהמסד נתונים, השרת קורא את שם התמונה, מחפש אצלו את התמונה, ממיר ל `Byte[]` שולח למשתמש ובצד השתמש התמונה מאוחסנת התוך ה `ImageSource`).

בצד הלקוח

```
// פתיחת סייר הקבצים
OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "All Images | *.jpg;*.jpeg;*.tif;*.tiff;*.bmp;*.png;" +
    "|JPEG Files (*.jpeg)|*.jpeg" +
    "|PNG Files (*.png)|*.png" +
    "|JPG Files (*.jpg)|*.jpg" +
    "|GIF Files (*.gif)|*.gif";

ofd.Multiselect = false;
var result = ofd.ShowDialog();

if (result == true)
{
    ImageSource imageSource = new BitmapImage(new Uri(ofd.FileName));
    imgProfil.Source = imageSource;

    // שמירת שם התמונה
    Global.CurrentUser.PicturePath = ofd.SafeFileName;
    // שמירת התמונה ב Byte[]
    Global.CurrentUser.Picture = File.ReadAllBytes(ofd.FileName);
}
```

בצד השרת

```
// שמירת התמונה בתקיה המקומית
if (user.Picture != null)
    File.WriteAllBytes(@"../../Images/" + user.PicturePath, user.Picture);
```

```
// שליפת שם התמונה מהמסד נתונים ויצירתה מתוך התקיה המקומית
if (user.PicturePath != "")
    user.Picture = File.ReadAllBytes(@"../../Images/" +
        user.PicturePath);
```



הצפנה

בפרויקט זה ממשתי הצפנה לסיסמות הכניסה של הלקוחות, זה התבטא בכך שבמסד הנתונים שמורות סיסמאות מוצפנות וכך הם נשלחות ללקוח, אצל הלקוח נמצא מפתח ההצפנה והוא ע"י פיענוח יודע אם : סיסמת הכניסה שהזין המשתמש נכונה, וכן בעת החלפת הסיסמה הלקוח יכול לראות את הסיסמה המקורית ללא הצפנה, ורגע לפני ששולחים את אובייקט המשתמש עם הסיסמה החדשה לעדכון, מצפינים אותה וכך שולחים לשרת לעדכון.

במחלקה הגלובלית

```
// מפתח ההצפנה
public static readonly int EncodingKey = 3;

// הצפנת סיסמה
public static string EncodingPassword(string password)
{
    string encodingPassword = "";
    for (int i = 0; i < password.Length; i++)
    {
        encodingPassword += (Char)(password[i] + EncodingKey);
    }

    return encodingPassword;
}

// פיענוח סיסמה
public static string DecodingPassword(string password)
{
    string decodingPassword = "";
    for (int i = 0; i < password.Length; i++)
    {
        decodingPassword += (Char)(password[i] - EncodingKey);
    }

    return decodingPassword;
}
```

בעמוד הכניסה

```
// האם הסיסמה שהוקשה נכונה
if (currentUser.LoginPassword != Global.EncodingPassword(pswLogin.Password))
{
    txtMessage.Text = "סיסמה זו אינה נכונה";
    pswLogin.Password = "";
    pswLogin.Focus();
    return;
}
```

בעמוד שינוי הגדרות פרופיל

```
// פיענוח סיסמת המשתמש והכנסתה לפקד 'סיסמה'
tmpPassword = Global.DecodingPassword(Global.CurrentUser.LoginPassword);
pswPassword.Password = tmpPassword;
```

```
// הצפנת הסיסמה החדשה והכנסתה לסיסמת המשתמש
Global.CurrentUser.LoginPassword = Global.EncodingPassword(tmpPassword);
// שליחת אובייקט המשתמש לעדכון
Global.adminClient.UpdateUser(Global.CurrentUser);
```

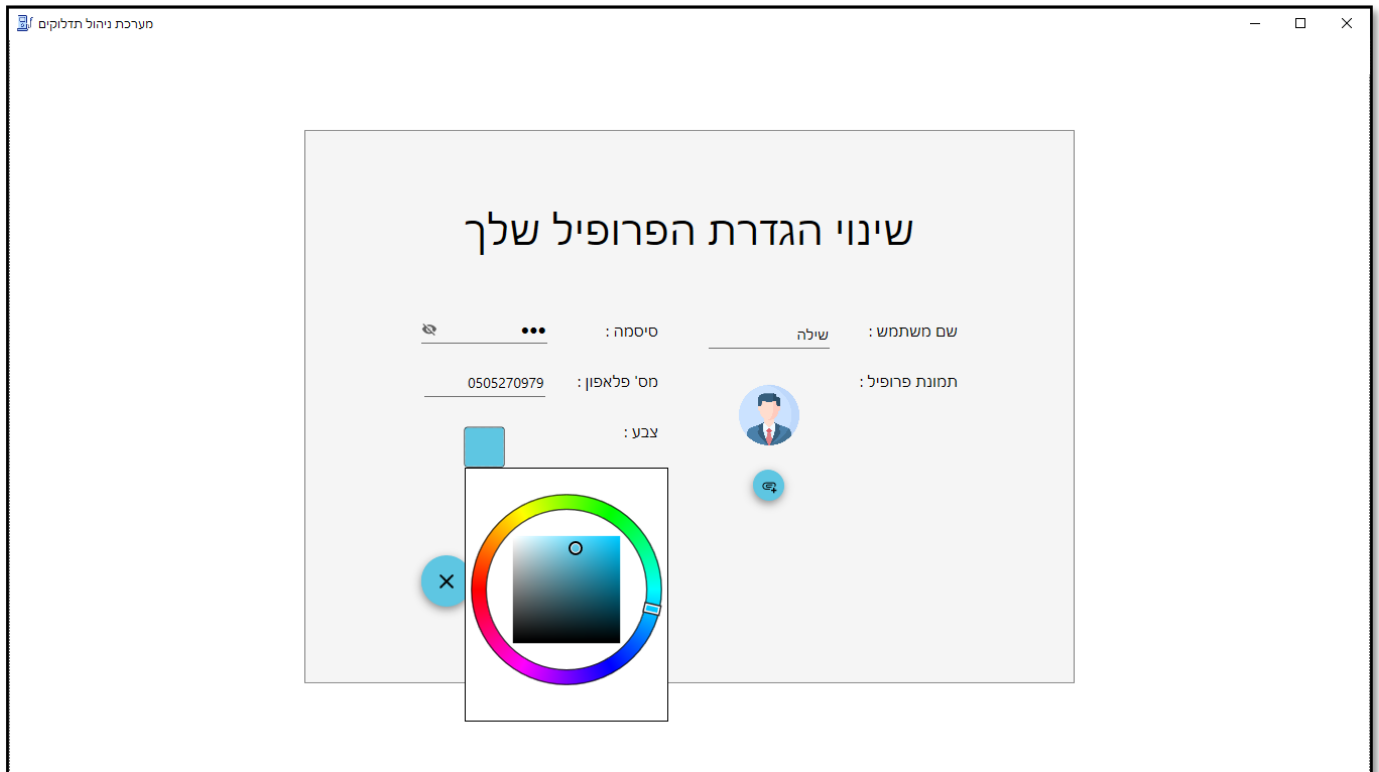


מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

צד לקוח

האפליקציה ללקוח נכתבה בWPF

להלן חלק מהדפים (תרשים מלא במדריך למשתמש)



```
<Page x:Class="Manager.Setting"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:Manager"
xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
xmlns:colorpicker="clr-namespace:ColorPicker;assembly=ColorPicker"
mc:Ignorable="d"
d:DesignHeight="500" d:DesignWidth="800"
Title="Setting"
Background="White">

<Grid FlowDirection="RightToLeft">
<Border BorderBrush="Gray"
BorderThickness="1"
VerticalAlignment="Center"
HorizontalAlignment="Center"
Background="WhiteSmoke">

<Grid Margin="100,50">
<Grid.RowDefinitions>
<!-- כותרת -->
<RowDefinition Height="Auto"/>
<RowDefinition Height="10"/>
<!-- פרטי המשתמש -->
<RowDefinition Height="Auto"/>
<RowDefinition Height="10"/>
<!-- לחצנים -->
<RowDefinition Height="Auto"/>
<RowDefinition Height="10"/>
</Grid.RowDefinitions>
```



```
<TextBlock Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"
    Text="שינוי הגדרת הפרופיל שלך" FontSize="40" Margin="20, 20, 20, 40"
    HorizontalAlignment="Center"/>

<Grid Grid.Row="2">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="20"/>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="Auto"/>
    </Grid.ColumnDefinitions>

    <StackPanel Grid.Column="0">
        <TextBlock Text="שם משתמש:" Margin="15" FontSize="15"/>
        <TextBlock Text="תמונת פרופיל:" Margin="15" FontSize="15"/>
    </StackPanel>

    <StackPanel Grid.Column="1">
        <TextBox x:Name="txtLoginName" Width="120" Margin="15,15,15,5"
            FontSize="14" Text="{Binding LoginName}"
            KeyDown="txtLoginName_KeyDown"/>
        <TextBlock x:Name="txtLoginNameError" Text="שם זה כבר שמור במערכת"
            Foreground="Red" FontSize="12" Visibility="Hidden" />

        <Image x:Name="imgProfil" Margin="15" MaxWidth="120" MaxHeight="60"
            Source="{Binding Picture}"/>
        <Button x:Name="btnUploadImg" Click="btnUploadImg_Click"
            Style="{StaticResource AddPictureButtonStyle}"/>
    </StackPanel>

    <StackPanel Grid.Column="3" >
        <TextBlock Text="סיסמה:" Margin="15" FontSize="15" />
        <TextBlock Text="מס' פלאפון:" Margin="15" FontSize="15"/>
        <TextBlock Text="צבע:" Margin="15" FontSize="15" />
    </StackPanel>

    <StackPanel Grid.Column="4" HorizontalAlignment="Left">
        <PasswordBox x:Name="pswPassword" Width="125" Margin="15,10"
            FontSize="14" MaxLength="10"
            HorizontalAlignment="Left"
            KeyUp="pswPassword_KeyUp"
            Style="{StaticResource MaterialDesignRevealPasswordBox}"/>

        <TextBox x:Name="txtPhoneNumber" Width="120" Margin="15"
            FontSize="14" Style="{StaticResource Validation}"/>
        <Binding Path="PhoneNumber">
            <Binding.ValidationRules>
                <local:IsCellPhone/>
            </Binding.ValidationRules>
        </Binding>
    </TextBox>

        <colorpicker:PortableColorPicker x:Name="colorPicker"
            SelectedColor="{Binding WorkspaceColor}"
            ColorState="{Binding ElementName=main,
                Path=ColorState, Mode=TwoWay}" Width="40" Height="40"
            Margin="0,15" FlowDirection="LeftToRight"
            ColorChanged="ColorPicker_ColorChanged"/>
    </StackPanel>
```




מערכת לניהול תדלוק אוטומטי לצ"י רכב בס"ד

```
</Grid>

<!-- לחצנים -->
<StackPanel Grid.Row="4" Orientation="Horizontal"
    HorizontalAlignment="Right" Margin="0,20,0,0" >
    <TextBlock x:Name="txtError" Foreground="Red" FontSize="13"
        VerticalAlignment="Center" Visibility="Collapsed"/>
    <Button Margin="15" FontSize="18" Click="btnSave_Click"
        Style="{StaticResource SaveButtonStyle}"/>
    <Button Margin="15" FontSize="18" Click="btnCancel_Click"
        Style="{StaticResource CancelButtonStyle}"/>
</StackPanel>
</Grid>
</Border>
</Grid>
</Page>
```

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציים מוצרים תחנות

חיפוש: שם תחנה ערים תגים

קוד תחנה	שם תחנה	כתובת	עיר	תגים
1998	הדסים	כניסה צפונית לאבן יהודה מצומת קדימה הדסים	אילת	תחנה שומרת שבת
2997	אופקים	שדרות הרצל	אופקים	
3996	פז אור יהודה	יהונתן נתניהו 6	אור יהודה	תחנה שומרת שבת כספומט
4995	סביון	כביש מסובים יהוד - צומת סביון	אילת	נגישות לנכה חנות Yellow כספומט בקו הירוק
6993	צבר	דרך ירושלים 8	אלפי מנשה	תחנה שומרת שבת כספומט
7992	אדום אילת	ששת הימים מול מפעלי תובלה	אילת	בקו הירוק
8991	לשם אילת	דרך הערבה 4	אילת	
9990	טל עופרה	צומת טל עופרה בכניסה לאלפי מנשה	אלפי מנשה	
10989	בעינה	כביש צפת עכו - דייר אל	אסד	
11988	אפק החדשה	מרחב מועצה אזורית סמוך לכביש 79	אפק	
12987	ארגמן	מושב ארגמן בקעת הירדן	ארגמן	
13986	אריאל	הכניסה לרחוב מוריה מול האונברסיטה	אריאל	
14985	עד הלום	א.ת. עד-הלום כביש אשדוד אשקלון	אשדוד	
15984	אשדוד עיר	שד' בני ברית א.ת. הקלה	אשדוד	תחנה שומרת שבת נגישות לנכה חנות Yellow
16983	ניר גלים	יציאה צפונית של אשדוד ליד ניר גלים	אשדוד	
17982	הכושם אשדוד	רח' הכושם 3	אשדוד	

257 תחנות

```
<Page x:Class="Manager.StationsList"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:Manager"
    xmlns:controls="clr-namespace:Manager.Controls"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="1200"
    Title="StationsList"
    Background="White">

<Grid FlowDirection="RightToLeft">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="auto"/>
            <RowDefinition/>
        </Grid.RowDefinitions>
```



```
<StackPanel Grid.Row="0" Orientation="Horizontal" HorizontalAlignment="Left"
    VerticalAlignment="Center">
    <TextBlock Text="חיפוש : " Margin="10,20,10,0" FontSize="15" VerticalAlignment="Center"/>

    <TextBox x:Name="txtSearchByStationName" Width="150" Margin="8,20,8,7"
        FontSize="15" Height="30"
        TextChanged="txtX_TextChanged"
        materialDesign:HintAssist.Hint="שם תחנה"/>

    <ComboBox Grid.Column="2" Grid.Row="4" Name="cmbCities" Margin="8,10"
        FontSize="15" DisplayMemberPath="CityName" Width="150"
        materialDesign:HintAssist.Hint="ערים"
        Style="{StaticResource MaterialDesignFloatingHintComboBox}"
        SelectionChanged="cmbX_SelectionChange"/>

    <ComboBox Grid.Column="2" Grid.Row="4" Name="cmbTags" Margin="8,10" FontSize="15"
        DisplayMemberPath="TagName" Width="150"
        materialDesign:HintAssist.Hint="תגים"
        Style="{StaticResource MaterialDesignFloatingHintComboBox}"
        SelectionChanged="cmbX_SelectionChange"/>

    <Button Height="25" Margin="10,0"
        x:Name="btnClearFilter"
        Visibility="Collapsed"
        BorderThickness="0"
        Style="{StaticResource MaterialDesignOutlinedButton}"
        Click="btnClearFilter_Click">
        <Image Width="20" Margin="-10" Source="/Images/clear-filter.png"/>
    </Button>

</StackPanel>

<ListView x:Name="dgStations" Grid.Row="1" FontSize="16"
    ScrollViewer.VerticalScrollBarVisibility="Auto">

    <ListView.ItemContainerStyle>
        <Style TargetType="ListViewItem">
            <Setter Property="Margin" Value="20,0,0,0" />
        </Style>
    </ListView.ItemContainerStyle>

    <ListView.View>
        <GridView>
            <GridViewColumn Header="קוד תחנה" DisplayMemberBinding="{Binding
StationID}"/>
            <GridViewColumn Header="שם תחנה" DisplayMemberBinding="{Binding
StationName}"/>
            <GridViewColumn Header="כתובת" DisplayMemberBinding="{Binding Address}"/>
            <GridViewColumn Header="עיר" DisplayMemberBinding="{Binding
City.CityName}"/>
            <GridViewColumn Header="תגים" Width="Auto">
                <GridViewColumn.CellTemplate>
                    <DataTemplate>
                        <ItemsControl ItemsSource="{Binding Tags}">

                            <ItemsControl.ItemsPanel>
                                <ItemsPanelTemplate>
                                    <WrapPanel Orientation="Horizontal"/>
                                </ItemsPanelTemplate>
                            </ItemsControl.ItemsPanel>
                        </ItemsControl>
                    </DataTemplate>
                </GridViewColumn.CellTemplate>
            </GridViewColumn>
        </GridView>
    </ListView.View>
</ListView>
```



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

```
<ItemsControl.ItemTemplate>
    <DataTemplate>
        <Grid Margin="4,0">
            <Border Background="{DynamicResource PrimaryHueMidBrush}"
                    CornerRadius="16"
                    Opacity="0.2"
                    BorderThickness="2"/>

            <TextBlock Text="{Binding TagName}"
                    Foreground="{DynamicResource PrimaryHueDarkBrush}"
                    Padding="10,5"/>

        </Grid>
    </DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>
</DataTemplate>
</GridViewColumn.CellTemplate>
</GridViewColumn>
</GridView>
</ListView.View>
</ListView>
</Grid>

<Frame x:Name="frm" NavigationUIVisibility="Hidden"/>

</Grid>
</Page>
```

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה תחנות **חוקים** רכבי העובדים

עדכון חוק קיים

שם חוק: ראש צוות

הגבלת מוצרים

מוצר	הגבלה יומית	הגבלה חודשית	
Y Burger	2	20	🗑
בנין 98	5	3000	🗑

הוסף

הגבלת תגים

תחנה שומרת שבת ✓ שטיפת רכב ✓ בקו הירוק

גישות לנכה Yellow חנות ✓ כספומט

מספר התחנות האפשריות: 0

מספר הרכבים המחוברים לחוק זה: 0

7 חוקים



```
<Page x:Class="Manager.AddRule"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:Manager"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="AddRule"
    Background="White"
    x:Name="myPage">

    <ScrollViewer VerticalScrollBarVisibility="Auto" FlowDirection="RightToLeft">
        <Grid FlowDirection="RightToLeft">
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>

            <TextBlock x:Name="txtHeader" Grid.Row="0" Grid.Column="0"
                Grid.ColumnSpan="2" Text="הוספת חוק חדש" FontSize="40" Margin="10"
                HorizontalAlignment="Center" VerticalAlignment="Center"/>

            <StackPanel Grid.Row="1" Orientation="Horizontal">
                <TextBlock FontSize="20" Margin="20, 0,15,50" Text="שם חוק: " />
                <TextBox x:Name="txtRuleName" FontSize="17" Margin="0, 0,20,50"
                    Width="150" Text="{Binding RuleName}" />
            </StackPanel>

            <!-- מוצרים -->
            <StackPanel Grid.Row="2" Grid.Column="0">
                <Border Padding="5,2" BorderThickness="1" Margin="20, 0"
                    Opacity="0.7" Width="Auto"
                    HorizontalAlignment="Left" BorderBrush="Black">
                    <TextBlock Text="הגבלת מוצרים" FontSize="18"
                        HorizontalAlignment="Center" VerticalAlignment="Center" />
                </Border>
                <DataGrid x:Name="dgLimitedProducts" FontSize="15"
                    Background="White" Margin="40,10,0,0"
                    CanUserAddRows="False" AutoGenerateColumns="False"
                    VerticalAlignment="Bottom">
                    <DataGrid.Columns>
                        <DataGridComboBoxColumn x:Name="productsColumn" Header="מוצר"
                            DisplayMemberPath="ProductName"
                            SelectedValuePath="ProductID"
                            SelectedValueBinding="{Binding ProductID}"
                            Width="200"/>

                        <DataGridTextColumn Width="150" Header="הגבלה יומית"
                            Binding="{Binding DailyVolume}" CanUserSort="True"/>
                        <DataGridTextColumn Width="150" Header="הגבלה חודשית"
                            Binding="{Binding MonthlyVolume}" />
                    </DataGrid.Columns>
                    <DataGrid.TemplateColumn>
                        <DataGridTemplateColumn.CellTemplate>
                            <DataTemplate>
                                <Button Click="btnDeleteLimitedProduct_Click"
                                    Style="{StaticResource MaterialDesignFlatButton}"
                                    <materialDesign:PackIcon Kind="Delete"/>
                                </Button>
                            </DataTemplate>
                        </DataGrid.TemplateColumn>
                    </DataGrid.TemplateColumn>
                </DataGrid>
            </StackPanel>
        </Grid>
    </ScrollViewer>
</Page>
```



```
</DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>

</DataGrid.Columns>
</DataGrid>
<Button Content="הוסף" FontSize="20" Width="100"
        Click="btnAdd_Click" Style="{StaticResource
        MaterialDesignFlatButtonDarkButton}"/>

</StackPanel>

<!-- תגים -->
<StackPanel Grid.Row="3" Grid.Column="0">
    <Border Padding="5,2" BorderThickness="1" Margin="20 25,20,0"
            Opacity="0.7" Width="Auto"
            HorizontalAlignment="Left" BorderBrush="Black">
        <TextBlock Text="הגבלת תגים" FontSize="18"
                VerticalAlignment="Center"/>
    </Border>

    <ListBox x:Name="lstbTags" FontSize="15" Width="400"
            Margin="40,20, 20,20"
            HorizontalAlignment="Left"
            VerticalAlignment="Center"
            SelectionMode="Multiple"
            SelectedValue="{Binding TagID}"
            SelectedValuePath="TagID"
            DisplayMemberPath="TagName"
            Style="{StaticResource MaterialDesignFilterChipPrimaryListBox}"/>
</StackPanel>

<Grid Grid.Row="4" Margin="10,20,10,0">

    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="1.5*"/>
        <ColumnDefinition Width="2*"/>
    </Grid.ColumnDefinitions>

    <StackPanel Grid.Column="0" HorizontalAlignment="Left"
            VerticalAlignment="Center">
        <TextBlock Text="מספר התחנות האפשריות" FontSize="15"
                Margin="10"/>
        <TextBlock Text="מספר הרכבים המחוברים לחוק זה" FontSize="15"
                Margin="10"/>
    </StackPanel>

    <StackPanel Grid.Column="1" HorizontalAlignment="Left"
            VerticalAlignment="Center">

        <TextBlock FontSize="15" Margin="10" Text="{Binding
            RelatedStationsCount, ElementName=myPage}"/>
        <TextBlock FontSize="15" Margin="10" Text="{Binding
            RelatedCarsCount, ElementName=myPage}"/>
    </StackPanel>

</Grid>
```



```
<StackPanel Grid.Column="2" HorizontalAlignment="Center"
            VerticalAlignment="Center">

    <Button Margin="5" Style="{StaticResource
        MaterialDesignFlatButton}">
        <materialDesign:PackIcon Kind="ArrowLeft" Height="20"
            Width="40" HorizontalAlignment="Center"/>
    </Button>

    <Button Margin="5" Style="{StaticResource
        MaterialDesignFlatButton}">
        <materialDesign:PackIcon Kind="ArrowLeft" Height="20"
            Width="40" HorizontalAlignment="Center"/>
    </Button>

</StackPanel>

<StackPanel Grid.Column="3" Grid.ColumnSpan="2"
            Orientation="Horizontal" HorizontalAlignment="Right"
            Margin="20,40,20,0">

    <TextBlock x:Name="txtErrorr" Foreground="red" FontSize="15"
        Text="יש למלא את כל השדות" VerticalAlignment="Bottom"
        Visibility="Hidden" Margin="10"/>

    <Button x:Name="btnSave"
        Click="btnSave_Click"
        Style="{StaticResource SaveButtonStyle}"/>
    <Button x:Name="btnCancel"
        Click="btnCancel_Click"
        Style="{StaticResource CancelButtonStyle}"/>
</StackPanel>
</Grid>

</Grid>
</ScrollView>
</Page>
```



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מדריך למשתמש

ממשק לקוח

לקוח יקר

אפליקציית "מערכת לניהול תדלוק אוטומטי לציי רכב" תוכננה במיוחד כדי שתוכל לאפשר לעובדיך שירות תדלוק אוטומטי ולנהל ולשלוט בהוצאות הדלק של העובדים באמצעות הרשאות תדלוק והגבלות פר עובד / קבוצת עובדים. וכן לקבל מידע מרוכז באמצעות דוחות תקופתיים אודות צריכת הדלק של העובדים בצורה נוחה ויעילה.

מערכת ניהול תדלוקים

ערב טוב

שם פרופיל

סיסמה

יש להכניס שם משתמש
וסיסמה כדי להיכנס.



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

לחצן לשינוי הפרופיל (בהמשך)

אפשרויות חיפוש

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה תחנות חוקים רכבי העובדים

חיפוש: שם עובד מספר לוחית

קוד רכב	תז העובד	שם העובד	פלאפון העובד	מספר לוחית	דגם	יצרן	חוק משויך
1	214862807	ראובן	0559874568	12345678	Mercedes	Litex	ראש צוות
5	332186774	גילה	0556750529	32587941	Volovo	Bulgaralpine	ראש צוות

לחצני הוספה עריכה מחיקה (מסכים בהמשך)

2 רכבי עובדים

אפשרות להתנתקות וחזרה למסך הכניסה

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה תחנות חוקים רכבי העובדים

חיפוש: שם עובד מספר לוחית

קוד רכב	תז העובד	שם העובד	פלאפון העובד	מספר לוחית	דגם	יצרן	חוק משויך
1	214862807	ראובן	0559874568	12345678	Mercedes	Litex	ראש צוות
5	332186774	חוק פרטי שימוש	0556750529	32587941	Volovo	Bulgaralpine	ראש צוות

ע"י מקש ימני ניתן לבחור מהתפריט המוצע.
חוק – יאפשר לראות את החוק המשויך של אותו עובד במסך החוק (בהמשך).
פרטי שימוש – יאפשר לראות את השימושים שלו, עסקות הצלחה ועסקות דחיה בפרוט מלא (בהמשך).

2 רכבי עובדים



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

ע"י לחיצה בלחצן העדכון מגיעים לדף זה

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה תחנות חוקים **רכבי העובדים**

עדכון פרטי עובד

שם עובד:	ראובן	מספר לוחית:	12345678
פלאפון:	0559874568	דגם:	Mercedes
תעודת זהות:	214862807	יצרן:	Litex
חוק:	ראש צוות		

2 רכבי עובדים

שמירה או ביטול
הפעולה

ע"י לחיצה בלחצן ההוספה מגיעים לדף זה

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה תחנות חוקים **רכבי העובדים**

הוספת רכב עובד :

שם עובד:		מספר לוחית:	
פלאפון:		דגם:	
תעודת זהות:		יצרן:	
חוק:			

2 רכבי עובדים

יש למלא את כל השדות, וכן מספר הפלאפון חייב להיות 10 ספרות וכמו כן הת"ז צריכה להיות תקינה. כמו כן לא ניתן להכניס רכב פעמיים (ע"י מספר לוחית דומה).



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

רחל עוברני

התנתק

עסקות דחיה

עסקות הצלחה

תחנות

חוקים

רכבי העובדים

לחצני הוספה עריכה ומחיקה (מסכים בהמשך)

ניתן לעשות חיפוש ע"י שם החוק

שם חוק

קוד חוק

עובדים חדשים

2

עובדים מעל 2 שנים

3

עובדים מעל 5

4

עובד ניסיון

5

ניסיון

6

מנהלים

7

ראש צוות

8

7 חוקים

מערכת ניהול תדלוקים

רחל עוברני

התנתק

עסקות דחיה

עסקות הצלחה

תחנות

חוקים

רכבי העובדים

עדכון חוק קיים

שם חוק

ראש צוות

הגבלת מוצרים

מוצר

הגבלה יומית

הגבלה חודשית

מחיקת המוצר

Y Burger

2

20

בנזין 98

5

3000

הוספת המוצר

הוסף

הגבלת תגים

תחנה שומרת שבת

שטיפת רכב

בקו הירוק

נגישות לנכה

תנות Yellow

כספומט

הדלקה וכיבוי תגים להגבלה

דף התחנות עם הסינון לפי ההגבלה הנתונה

0

מספר התחנות האפשריות

דף הרכבי עובדים עם הסינון לפי ההגבלה הנתונה

0

מספר הרכבים המחוברים לחוק זה

שמירה או ביטול הפעולה

7 חוקים



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה תחנות **חוקים** רכבי העובדים

הוספת חוק חדש

שם חוק: _____

הגבלת מוצרים

מוצר הגבלה יומית הגבלה חודשית

הוסף

הגבלת תגים

תחנה שומרת שבת שטיפת רכב בקו הירוק נגישות לנכה חנות Yellow כספומט

מספר התחנות האפשריות: 0
מספר הרכבים המחוברים לחוק זה: 0

7 חוקים

אפשרות חיפוש ברשימת התחנות הכללית ע"י שם תחנה / עיר / תג

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה עסקות הצלחה **תחנות** חוקים רכבי העובדים

חיפוש: שם תחנה ערים תגים

קוד תחנה	שם תחנה	כתובת	עיר	תגים
1998	הדסים	כניסה צפונית לאבן יהודה מצומת קדימה הדסים	אילת	תחנה שומרת שבת
2997	אופקים	שדרות הרצל	אופקים	
3996	פז אור יהודה	יהונתן נתניהו 6	אור יהודה	תחנה שומרת שבת כספומט
4995	סביון	כביש מסובים יהוד - צומת סביון	אילת	נגישות לנכה חנות Yellow כספומט בקו הירוק
6993	צבר	דרך ירושלים 8	אלפי מנשה	תחנה שומרת שבת כספומט
7992	אדום אילת	ששת הימים מול מפעלי תובלה	אילת	בקו הירוק
8991	לשם אילת	דרך הערבה 4	אילת	
9990	טל עופרה	צומת טל עופרה בכניסה לאלפי מנשה	אלפי מנשה	
10989	בעינה	כביש צפת עכו - דייר אל	אסד	
11988	אפק החדשה	מרחב מועצה אזורית סמוך לכביש 79	אפק	
12987	ארגמן	מושב ארגמן בקעת הירדן	ארגמן	
13986	אריאל	הכניסה לרחוב מוריה מול האונברסיטה	אריאל	
14985	עד הלום	את. עד-הלום כביש אשדוד אשקלון	אשדוד	
15984	אשדוד עיר	שד' בני ברית את. הקלה	אשדוד	תחנה שומרת שבת נגישות לנכה חנות Yellow
16983	ניר גלים	יציאה צפונית של אשדוד ליד ניר גלים	אשדוד	
17982	הבושם אשדוד	רח' הבושם 3	אשדוד	

257 תחנות



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה | **עסקות הצלחה** | תחנות | חוקים | רכבי העובדים

חיפוש: קוד רישוי רכב

קוד עסקה	קוד רישוי רכב	מוצר	תאריך	כמות	הנחה	סה"כ לתשלום
1	32587941	Y Burger	AM 12:00:00 1/1/2023	1	0	35
2	12345678	בנדין 98	AM 12:00:00 8/5/2022	200	0	300

טבלת דוחות לגבי עסקות שצולחו ע"י העובדים בצי, וכן אפשרות חיפוש עסקות לפי קוד רישוי הרכב

2 עסקות

מערכת ניהול תדלוקים

רחל עוברני
התנתק

עסקות דחיה | עסקות הצלחה | תחנות | חוקים | רכבי העובדים

חיפוש: קוד רישוי רכב

קוד דחיה	קוד רישוי רכב	מוצר	תאריך	כמות
----------	---------------	------	-------	------

טבלת דוחות לגבי עסקות שנדחו לעובדי הצי, וכן אפשרות חיפוש עסקות לפי קוד רישוי הרכב

0 דחיות



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

שינוי הגדרת הפרופיל שלך

שם משתמש: _____ רחלי

שם המשתמש חייב להיות ייחודי (ז"א לא נמצא בשימוש אצל משתמש אחר).

סיסמה: _____

מספר הפלאפון חייב להכיל מספרים בלבד וכן 10 ספרות

מס' פלאפון: 7884585569

תמונת פרופיל: _____

ניתן להעלות תמונת פרופיל שתוצג בכניסה

ניתן לבחור צבע לסביבת העבודה, ואם לא בחרים הערך הברירת מחדל הוא

צבע: _____

השדות הינן חובה

×

📁




מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד


ממשק מנהל
מנהל יקר

אפליקציית הניהול מיועדת לסייע לך לאפשר חבילת תדלוק למנהלי ציי רכבים - ע"י הקמת ציי רכבים חדשים + יצירת חשבון למנהל הצי, לקבוע הנחות עבור צריכת דלק / מוצרים בתחנות, וכן להפיק דוחות פעילות תקופתיים.

מערך ניהול תדלוקים

ערב טוב

שילה 

סיסמה 

יש להכניס שם משתמש וסיסמה כדי להיכנס.
תמונת הפרופיל תשתנה במידה ונמצא משתמש כזה
והוא הגדיר את תמונת הפרופיל שלו.
אחרת תוצג תמונת ברירת מחדל של המערכת.

מערך ניהול תדלוקים

שילה עוברני
התנתק

לחצן לשינוי הפרופיל (מסך מתאים בהמשך).

אפשרויות חיפוש

אפשרויות להתנתקות וחזרה למסך הכניסה

לחצני הוספה עריכה מחיקה (מסכים בהמשך)

קוד תחנה	שם תחנה	ערים	תגים	מוצרים
1998	הדסים	כניסה צפונית לאבן יהודה מצומת קדימה	ליד פנימיית הדסים	אילת
2997	אופקים	שדרות הרצל		אופקים
3996	פז אור יהודה	יהונתן נתניהו 6		אור יהודה
4995	סביון	כביש מסובים יהוד - צומת סביון		אילת
6993	צבר	דרך ירושלים 8		אלפי מנשה
7992	אדום אילת	ששת הימים מול מפעלי תובלה		אילת
8991	לשם אילת	דרך הערבה 4		אילת
9990	טל עופרה	צומת טל עופרה בכניסה לאלפי מנשה		אלפי מנשה
10989	בעינה	כביש צפת עכו - דייר אל		אסד
11988	אפק החדשה	מרחב מועצה אזורית סמוך לכביש 79		אפק
12987	ארגמן	מושב ארגמן בקעת הירדן		ארגמן
13986	אריאל	הכניסה לרחוב מוריה מול האונברסיטה		אריאל
14985	עד הלום	את. עד-הלום כביש אשדוד אשקלון		אשדוד
15984	אשדוד עיר	שד' בני ברית את. הקלה		אשדוד
16983	ניר גלים	יציאה צפונית של אשדוד ליד ניר גלים		אשדוד
17982	הבושם אשדוד	רח' הבושם 3		אשדוד

257 תחנות



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

כאשר לחצו על לחצן ההוספה

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציייםמוצריםתחנות

הוספת תחנת תדלוק חדשה :

שם תחנה :

כתובת :

עיר :

תגים :

שם התחנה חייב להיות בעברית

שם תחנה :
כתובת :
עיר :
תגים :
תחנה שומרת שבת
שטיפת רכב
בקו הירוק
נגישות לנכה
חנות Yellow
כספומט

השדות חייבות להיות מלאות

שמירה / ביטול הפעולה

257 תחנות

כאשר לחצו על לחצן ההעריכה

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציייםמוצריםתחנות

עדכון תחנה קיימת

שם תחנה :

כתובת :

עיר :

תגים :

הדסים
כניסה צפונית לאבן יהודה מצומו
אילת

שם תחנה :
כתובת :
עיר :
תגים :
תחנה שומרת שבת ✓
שטיפת רכב
בקו הירוק
נגישות לנכה
חנות Yellow
כספומט

שמירה / ביטול הפעולה

257 תחנות



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציים מוצרים תחנות

חיפוש: שם צי

כל ☐ פעיל ☐ לא פעיל ☐

קוד צי	שם צי	שם איש קשר	טלפון איש קשר	סכום לתשלום	הנחה	לוגו	
1	אוסם	ת	0583235695	50	10		<input checked="" type="checkbox"/>
2	עלית	רחל	7884585569	4808	0		<input checked="" type="checkbox"/>
33	פפרוני	נועם	0556847899	1458	0		<input checked="" type="checkbox"/>
55	בנק ירושלים	דד	1414141414	1212	0		<input type="checkbox"/>
59	מיוצר בישראל	שרון	8981812956	124	0		<input checked="" type="checkbox"/>
61	DrLogo	יוני	8523697412	1124	0		<input checked="" type="checkbox"/>
63	הפניקס	מוריה	0534106215	2458	0		<input type="checkbox"/>

לחצני הוספה עריכה מחיקה (מסכים בהמשך)

7 ציים

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציים מוצרים תחנות

הוספת צי חדש :

שם החברה:

אנשי קשר:

לוגו:

המסך המתאים בעמוד הבא.

השדות :
סכום לתשלום, נקודות, הנחה
חייבים להיות מספרים וחייביים

הסכום לתשלום:

נקודות:

הנחה:

פעיל? ☐

אשרות להעלאת תמונת לוגו של החברה

לחצן השמירה יוכל להתבצע רק השדות הבאים מלאים:
שם החברה, אנשי קשר לפחות אחד, הסכום לתשלום

7 ציים



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

אפשרות להוספה, ועריכת אנשי הקשר של החברה.

שם פרטי	שם משפחה	תז	פלאפון	פעיל?
מוריה	עוברני	034179887	0534106215	<input checked="" type="checkbox"/>
אלישבע	עוברני	333127710	0556751875	<input checked="" type="checkbox"/>

שם מלא : שם פרטי : פלאפון : פעיל? ☐

שם משפחה : תעודת זהות : בעת הוספה או עדכון איש קשר יש למלא את כל השדות.

תעודת זהות חייבת להכיל 9 ספרות ולהיות תקינה עם ספרת

הפלאפון חייב להכיל 10 ספרות, בלבד.

✕ + 💾

ע"י לחיצה על העדכון במסך הציים ניתן לערוך את הצי הנבחר במסך זה.

מערכת ניהול תדלוקים

שילה עוברני התנתק

תחנות מוצרים ציים

עדכון צי קיים

שם החברה :	אוסם	הסכום לתשלום :	500
אנשי קשר :	2	נקודות :	0
לוגו :		הנחה :	10
		פעיל ?	<input checked="" type="checkbox"/>

✕ 💾

7 ציים



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

ע"י לחיצה על לחצן ימני על צי מסוים ניתן לראות את רשימת רכבי העובדים של אותה חברה.

מערכת ניהול תדלוקים

שילה עוברני
התנתק

צייים מוצרים תחנות

חיפוש: שם צי ☐ לא פעיל ☐ פעיל ☒ הכל

קוד צי	שם צי	שם איש קשר	טלפון איש קשר	סכום לתשלום	הנחה	לוגו
1	אוסם	ת	0583235695	500	10	
2	עלית	רחל	7884585569	4808	רשימת רכבים	
33	פפרוני	נועם	0556847899	1458	ט	
55	בנק ירושלים	דד	1414141414	1212	0	
59	מיוצר בישראל	שרון	8981812956	124	0	
61	DrLogo	יוני	8523697412	1124	0	
63	הפניקס	מוריה	0534106215	2458	0	

7 צייים

מסך המציג את רשימת הרכבי העובדים של צי מסוים.

מערכת ניהול תדלוקים

שילה עוברני
התנתק

צייים מוצרים תחנות

לרשימת הציים

הרכבים בחברת אוסם

קוד רכב	מספר לוחית	דגם	יצרן	שם בעל הרכב	פלאפון בעל הרכב
3	12378456	Scania	Brooks	שמעון	0123654785

ע"י לחיצה על כפתור זה ניתן לחזור לרשימת הציים.

7 צייים



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציים מוצרים תחנות

חיפוש: שם מוצר מחיר

קוד מוצר	שם מוצר	מחיר	תאור
1	רחיצה	20	
11	טעינה חשמלית	100	
13	בנזין 98	150	
14	פלאפל בריבוע	15	
15	גפ"מ (גז)	20	
16	Y Burger	35	

לחצני הוספה עריכה מחיקה (מסכים בהמשך)

אפשרויות חיפוש

6 מוצרים

מערכת ניהול תדלוקים

שילה עוברני
התנתק

ציים מוצרים תחנות

כאשר לחצו על לחצן הוספת מוצר

הוספת מוצרים חדשה:

שם מוצר: _____

מחיר: _____

תאור: _____

מחיר - יכול להיות אך ורק מספר וחיובי.

לשמירה יש למלא את השדות: שם מוצר ומחיר.

6 מוצרים



מערכת לניהול תדלוק אוטומטי לציי רכב בס"ד

מערכת ניהול תדלוקים

כאשר לחצו על לחצן עדכון מוצר.

שילה עוברני
התנתק

צייים מוצרים תחנות

עדכון מוצר קיים

שם מוצר: גפ"מ (גז)

מחיר: 20

תאור:

לשמירה השדות:
שם מוצר ומחיר
חייבות להיות מלאות.

6 מוצרים

מערכת ניהול תדלוקים

שינוי הגדרת הפרופיל שלך

שם המשתמש חייב להיות ייחודי (ז"א לא נמצא בשימוש אצל משתמש אחר).

שם משתמש: שילה

תמונת פרופיל:

סיסמה: ...

מס' פלאפון: 0505270979

צבע:

ניתן לבחור צבע לסביבת העבודה, ואם לא בוחרים הערך הברירת מחדל הוא

ניתן להעלות תמונת פרופיל שתוצג בכניסה

השדות הינן חובה

מספר הפלאפון חייב להכיל מספרים בלבד וכן 10 ספרות



בס"ד מערכת לניהול תדלוק אוטומטי לציי רכב

סימולטור תדלוק

מערכת ניהול תדלוקים

מספר רישוי : 9 / 0

תחנה : תחנות

מוצר : מוצרים

שליחת בקשה

שליחה לבדיקת הבקשה.

תוצאה :

תגובת השרת אם הוא יכול לצרוך את המוצר, ואם כן מה הכמות המקסימלית.

קוד תחנה :

קוד רכב :

קוד מוצר :

כמות :

דווח

אם הוא אכן יכול לצרוך את המוצר, מתמלא לו אוטומטית השדות, חוץ משדה – ה"כמות" שבו הוא יכול להכניס עד המכסה המותרת ואז עליו ללחוץ על כפתור הדווח.

רפלקציה

נהנתי מאד מבנית הפרויקט ומהלמידה הנרחבת במהלך כל הבניה,

לא מעט מהידע שצברתי הוא באמצעות סיוע לחברות שנתקלו בשגיאות או אתגרים שונים (חלקם מורכבים).
ואולי הכי חשוב, למדתי כיצד לחפש ולאתר פתרונות לבעיות.