



סמינר בית יעקב ע"ש הרב י.א. וולף זצ"ל
בהנהלת **הרב ח.א. כהנא** שליט"א
רח' אמרי ברוך 30 בני ברק טלפון 03-6714500 פקס 03-6714548

שם העבודה: קייק בייק

המגישה: חני פרוביין

ת.ז: 214861635

שם בית הספר: כהנא

המנחה: המורה גינאויר והמורה מליח

חלופה: מערכות מנהליות – wpf

שנה: 2023

תוכן

3	מבוא
3	תאור הארגון
3	מטרות הארגון:
5	נתוח מערכת
7	הגורמים המעורבים:
8	יעדי המערכת:
9	קשרי גומלין מצב קיים
9	בעיות וכיוון פתרון :
12	תרשים DFD0
14	מטרות המערכת העתידית
15	עץ-תהליכים
22	4. שכבת ה־Model
25	שכבת ה־ViewModel
39	הנחות יסוד לנרמול
40	טבלאות
47	טופס מנהל ראשי
	טופס לקוחות
	שגיאה! הסימניה אינה מוגדרת.
48	טופס מנהל תחנה
46	פרטי לקוח
49	טופס הוספת מוצר
	טופס פרטי מכירה
	שגיאה! הסימניה אינה מוגדרת.

מבוא

תאור הארגון

אתר המכירה נוסד בשנת 2015.

מערכת " קייק בייק " מספקת מגוון של סוגי מוצרים שונים ומתכונים ברמות שונות. הצורך עלה היות והרצון לאפות גבוה והטרחת המרובה ביציאה לקניית המוצרים הנדרשים גורם לעיתים אף ללכת למספר חניות כדי להשיג את המצרכים ומרתייע מלאפות.

לכן המערכת מרכזת את הקניות במקום אחד ודואג לחווית קניה ללקוחות. החברה מספקת מוצרי אפיה רבים במקום אחד והבאתם לבית הלקוח, העובדים אורזים את המוצרים בהתאם להזמנות הלקוחות בצורה מאורגנת ואסתטית.

מטרות הארגון:

מטרת על:

רווח

מטרות נילוות:

הנגשת מוצרים והקלה לצרכנים

גבולות המערכת: לשים לב לתת הסבר לכל משתמש את החלק שלו

המנהל אחראי על הכנסת המתכונים והמוצרים למערכת (אחראי על משכורות העובדים-הפנימיים ועל ייצוב מצבה הכלכלי של החברה). ובנוסף התנהלות מול ספקים כשמקבל התראה על מלאי נמוך ממוצר מסוים ולעדכנו במערכת כשמגיע.

וכן להעביר את דוחות הקניה לעובדים לאריזת קניות הלקוחות וכן לדאוג להעביר את קנית הלקוח ע"י שליח לביתו.

הלקוח מכניס את פרטיו למערכת ובאמצעות הקוד שהכניס יכול להכנס למערכת ולבחור מתכון רצוי ולהזמין את המוצרים הנדרשים, כמו כן יכול להזמין מוצרים נוספים וכן להגיב על מתכונים ע"י כתיבת תגובה או לחיצה על "לייק".

חברת השליחויות מקבלת את ההודעה שישנה הזמנה הצריכה לצאת ליעדה ושולחת

שליח להעביר את החבילה לבית הלקוח

אופן התשלום נעשה מראש בכרטיסי אשראי

המערכת מתנהלת עם לקוחות, פרטי הזמנה, מוצרים, מתכונים, הזמנות לקוח,

קהל היעד

לקוחות

הסיבות לבחירת הנושא -

בחרתי בנושא זה מכיוון שההובי שלי הוא אפיה ורציתי לבנות משהו שאני אוהבת להתעסק אתו.

אתגרים מרכזיים

לפני הכתיבה עשיתי עבודה ובדקתי כיצד מתנהלים אתרים של רכישות מוצרים ורכישות בהתאם למתכון נבחר.

חידושים

בתוכנה תהיה אפשרות לראות את **התמונות של המתכונים**, בנוסף הלקוחות יכולים **להביע את דעתם על המתכון ולכתב הערות**, וכן ללחוץ על ה"לייק" כשהכינו ונהנו ובכך לאפשר למשתמשים מידע האם המתכון מוצלח, כמו כן האתר **משווק ממוצרים מסוימים מספר סוגים** של אריזות המכילות כמויות שונות ומאפשר למשתמש לבחור את המוצר בכמות הרצויה לו.

נתוח מערכת

מסמך ייזום

תרשים המבנה האירגוני: (לא כולל אנשים המקבלים שרות)



תיחום תהליכים:

המערכת מטפלת ב:

טיפול בלקוח

טיפול במוצר

טיפול במתכון

נהול תגובות

נהול קניות

המערכת לא מטפלת ב:

הזמנה למלאי חנות האתר

העברת קניה ללקוח

אין בקרה על תגובות חוזרות של לקוח

סליקת אשראי

הגורמים המעורבים:

שם הגורם	רמת מעורבות	תיאור
מנהל	יוזם, משתמש, צרכן	טיפול במוצר חדש טיפול במתכון חדש טיפול בלקוח נהול תגובות נהול קניות
לקוח	משתמש, צרכן	טיפול בלקוח נהול תגובות נהול קניות טיפול בתשלום

יעדי המערכת:

שם יעד	פרוט	רמה
טיפול בלקוח	הוספת לקוח חדש איתור לקוח קיים עדכון לקוח קיים מחיקת לקוח קיים	תפעולי
טיפול במתכון	הוספת מתכון חדש איתור מתכון קיים מחיקת מתכון קיים עדכון מתכון קיים	תפעולי
טיפול בקניות	יצירת קניה חדשה עדכון מלאי מוצרים סדור התשלום	תפעולי
ניהול תגובות	איתור מתכון הוספת תגובה חדשה הוספת לייק	תפעולי
טיפול במוצר	הוספת מוצר חדש איתור מוצר קיים מחיקת מוצר קיים עדכון מוצר קיים	תפעולי

אילוצים ומגבלות:

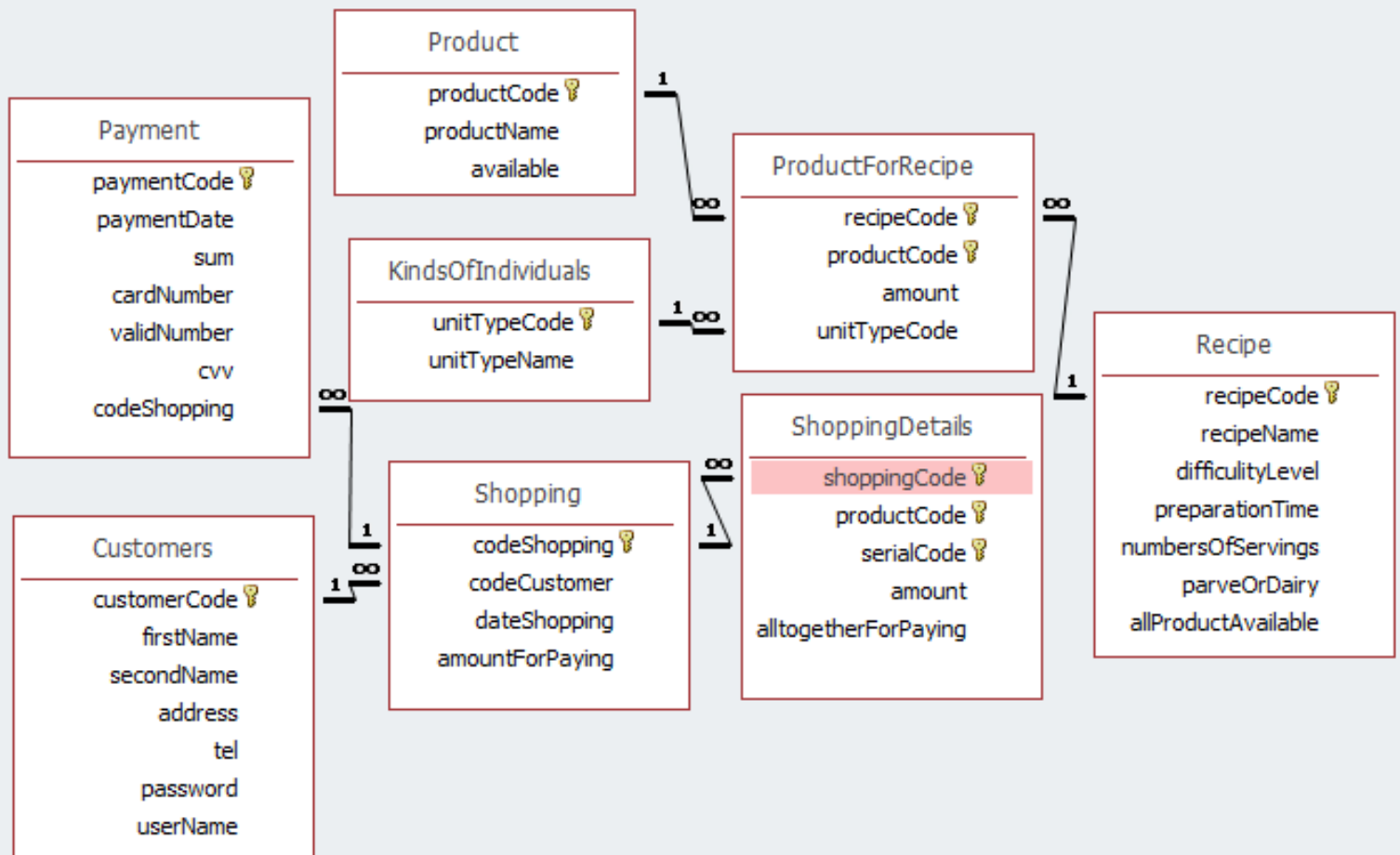
אילוצי כוח אדם: המערכת מפותחת ע"י תלמידת תיכון חני פרוביין.

העזרה ניתנת ע"י מורת התיכון למחשבים.

אילוצי אופי המערכת: המערכת צריכה להיות ידידותית, נוחה להפעלה, חסינה בפני שגיאות משתמש, בעלת מראה אסתטי, נקי וגמישה לשינויים עתידיים.

אילוצי זמן: המערכת תפעל לאורך כל השנה, תהיה מוכנה עד לתאריך 15/05/2023.

קשרי גומלין מצב קיים



בעיות וכיוון פתרון:

שם בעיה: מתכון ללא תגובות משתמשים
תאור בעיה: המערכת אינה מאפשרת רישום תגובות לקוחות על מתכון שביצעו.

<p>תאור סיבת בעיה: לקוח המעוניין להכין מתכון מסוים אינו יודע מה טיבו ואם אכן הוא מוצלח.</p> <p>תאור הנזק: הלקוח עלול להכין מתכון לא מוצלח.</p> <p>כיוון פתרון: יש לתת לכל מתכון קוד ולהגדיר תכונה לתגובה .</p>

<p>שם בעיה: אין אפשרות להציג את המתכון</p> <p>תאור בעיה: המערכת לא אפשרה הכנסת תמונת מתכון על מנת להכניס את מראה התוצאה במערכת השיקולים בקבלת החלטה.</p> <p>תאור סיבת בעיה: במאגר אין אפשרות להכניס המונה של המתכון.</p> <p>תאור הנזק: לקוח שאינו חשוף לתוצאה המוגמרת, אינו יכול להגיע לתוצאה מייטבית.</p> <p>כיוון פתרון: יש להגדיר תכונות לתמונה ולקשר את התמונה לקוד מתכון מסוים.</p>

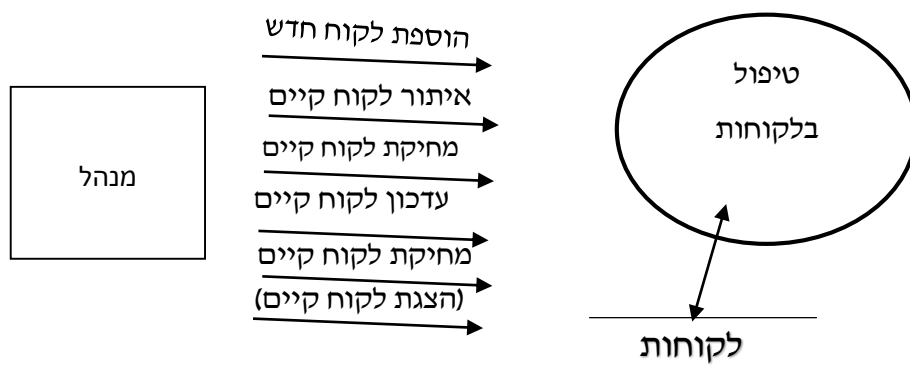
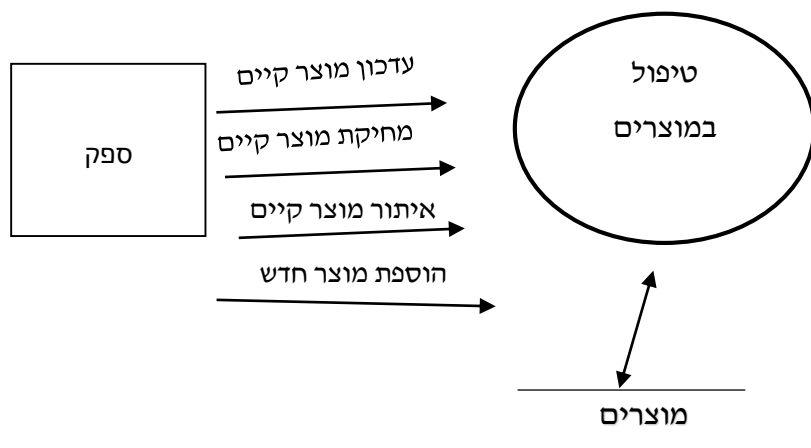
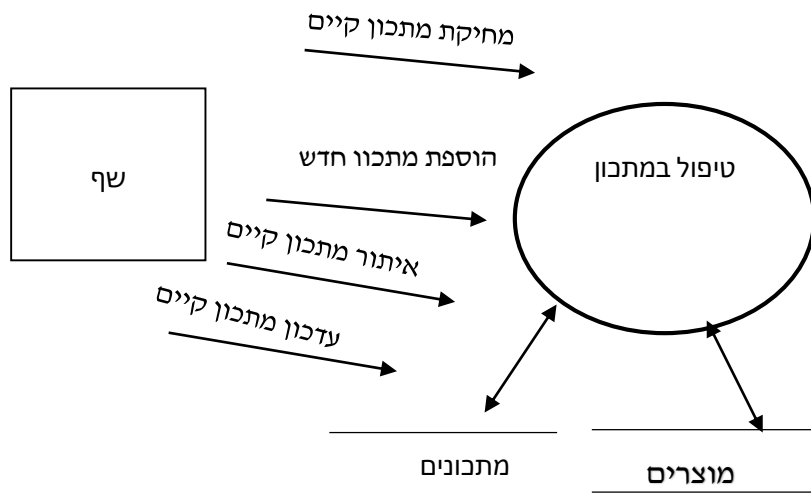
<p>שם בעיה: אי התאמת המוצר לדרישת מתכון.</p> <p>תאור בעיה: במתכון יש מוצר מסוים שהמשתמש צריך ממנו רק כמות מועטת.</p> <p>תאור סיבת בעיה: המערכת מציגה מוצר בכמות כל אחד כמוצר נפרד</p> <p>תאור הנזק: לקוח נשאר עם כמות עודפת</p> <p>כיוון פתרון: יש לפרט לכל מוצר קיים את הגדלים הקיימים לרכישה.</p>
--

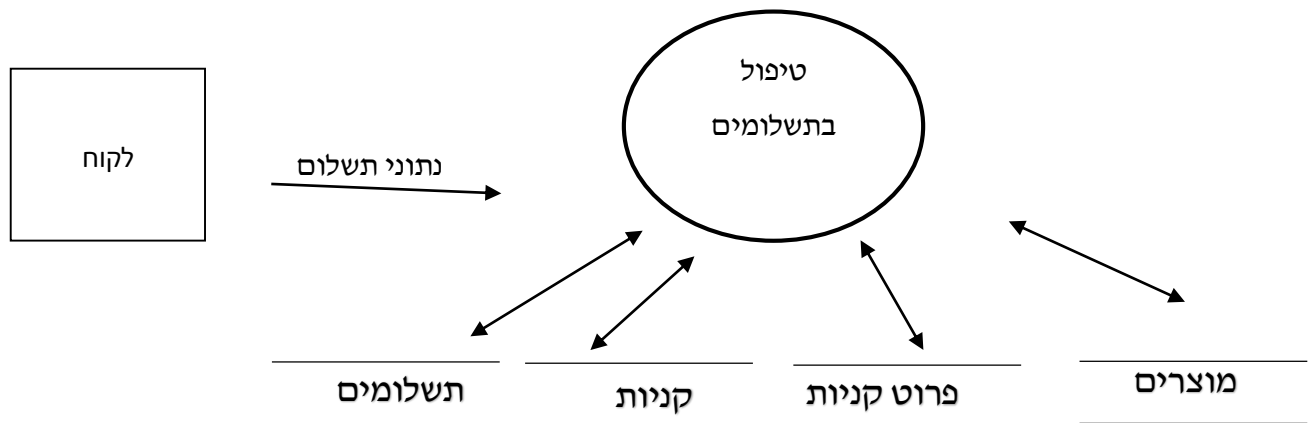
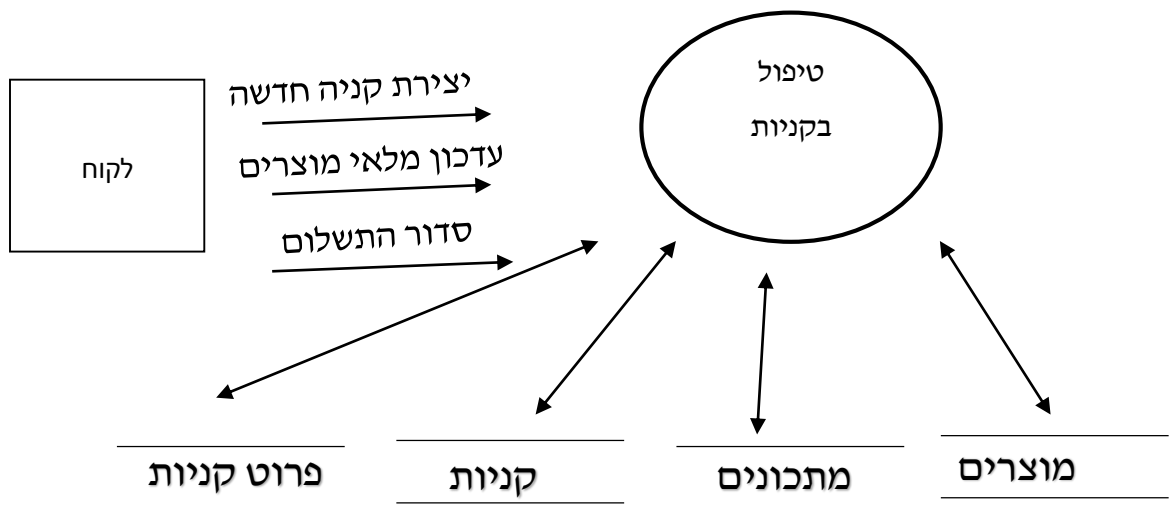
<p>שם בעיה: אין אפשרות לקבל את טיב המתכון .</p> <p>תאור בעיה: במתכון אין אפשרות לראות את מספר הלקוחות שאהבו את המתכון.</p>
--

תאור סיבת בעיה: המערכת מציגה את המתכון ללא מספר האנשים שאהבו את המתכון.

תאור הנזק: הלקוח עלול להכין מתכון לא מוצלח.

כיוון פתרון: יש לתת לכל מתכון תכונה ל"לייק" שבו הלקוח לוחץ ומעלה את מספר הלייקים.





מטרות המערכת העתידית

שם יעד המערכת: מתכון מוצג ויזואלית באתר

תיאור: המערכת תציג לכל מתכון את תמונתו.

מצב קיים: יש אפשרות להכין את המתכון רק לפי הוראות המתכון בלי לראות מצב מוגמר.

מצב רצוי: המערכת תציג את תמונת המתכון .

מדד: חווית שימוש.¹

שם יעד המערכת: מוצר קיים במספר גדלים.

תיאור: למוצר יש מספר גדלים והלקוח בוחר את הרצוי לו.

מצב קיים: לכל מוצר יש גודל אחד

מצב רצוי: המערכת תציג לכל מוצר את הגדלים הקיימים ממנו.

מדד: צריכה מותאמת

שם יעד המערכת: במתכון יוצגו תגובות מתאימות למתכון.

תיאור: הלקוח יוכל להגיב על מתכון מסוים ולאפשר יידע נרחב יותר ללקוחות הבאים.

מצב קיים: אין אפשרות להגיב על מתכון.

מצב רצוי: בכל מתכון יהיה אופציה ללקוח להגיב.

מדד: חווית שמוש

שם יעד המערכת: קבלת מידע על מתכון מוצלח כן/לא

תיאור: במידה והלקוח נהנה מהמתכון הוא ילחץ על ה"לייק"

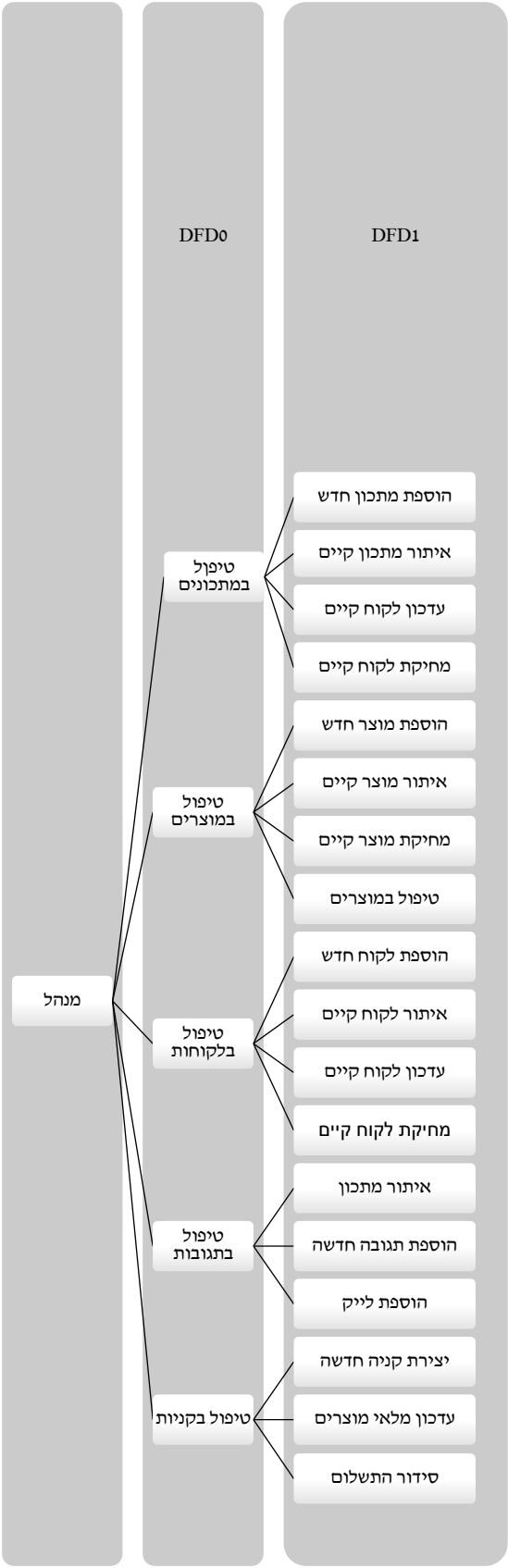
מצב קיים: קשה לדעת את טיבו של המתכון.

מצב רצוי: לכל מתכון יוצג כמה אנשים נהנו ממנו.

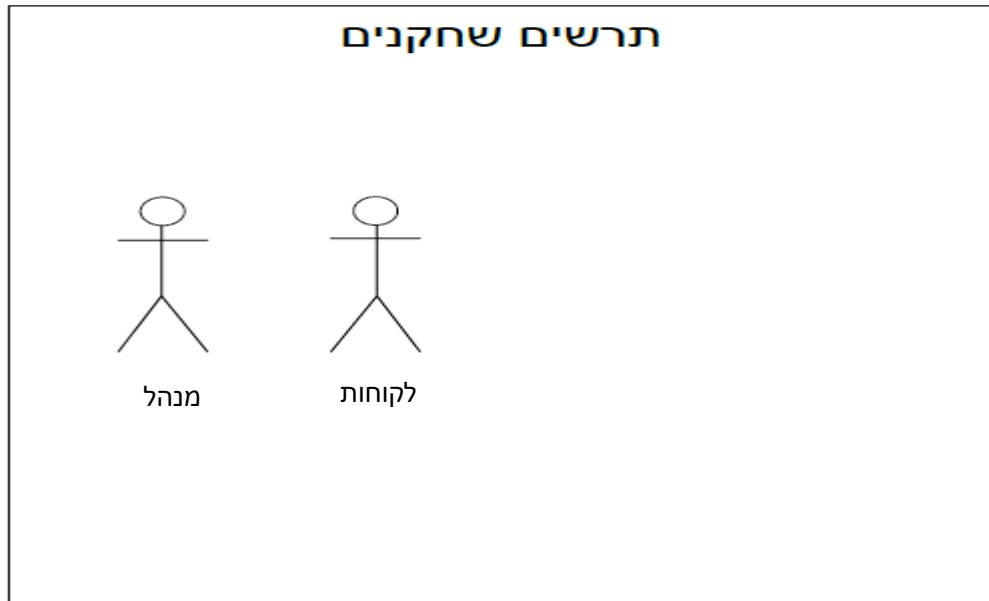
מדד: חווית שמוש

¹ מדד הוא גורם לפיו נבדק שיפור בפיתוח תוכנה קיימים מדדים פופולריות השימוש, המרות, איכות הדוחות, חווית שימוש, עמידה בלוח זמנים, משך זמן ביצוע, תקצוב ועוד

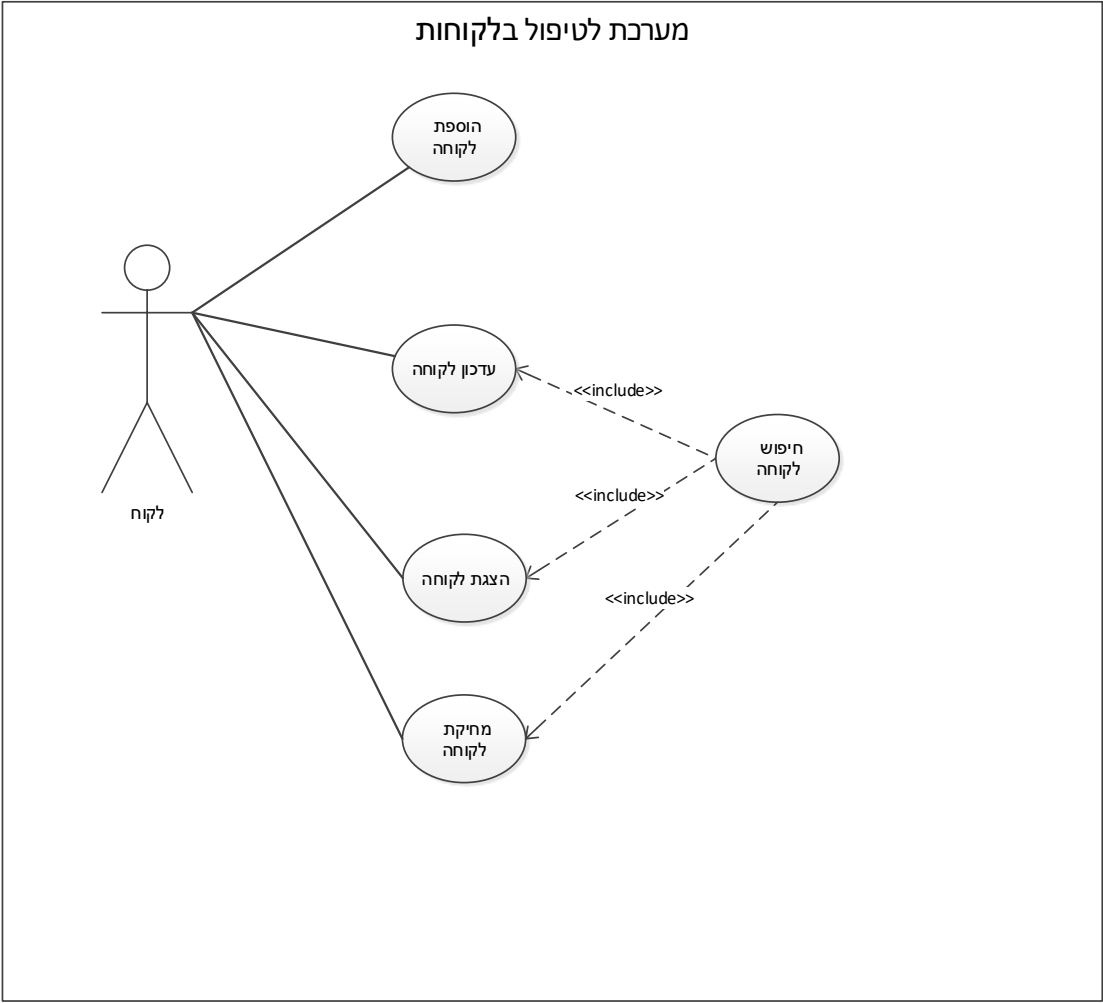
עץ-תהליכים



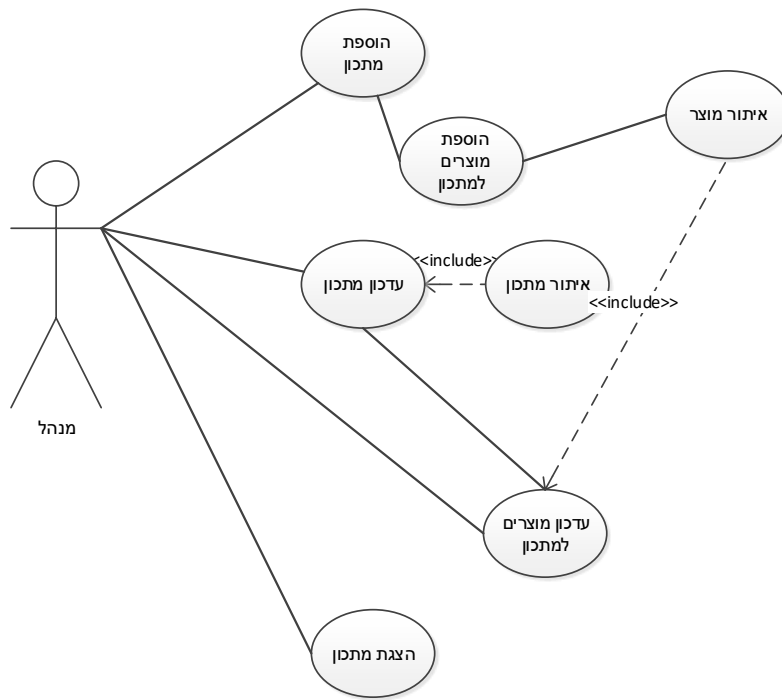
תרשימי UML



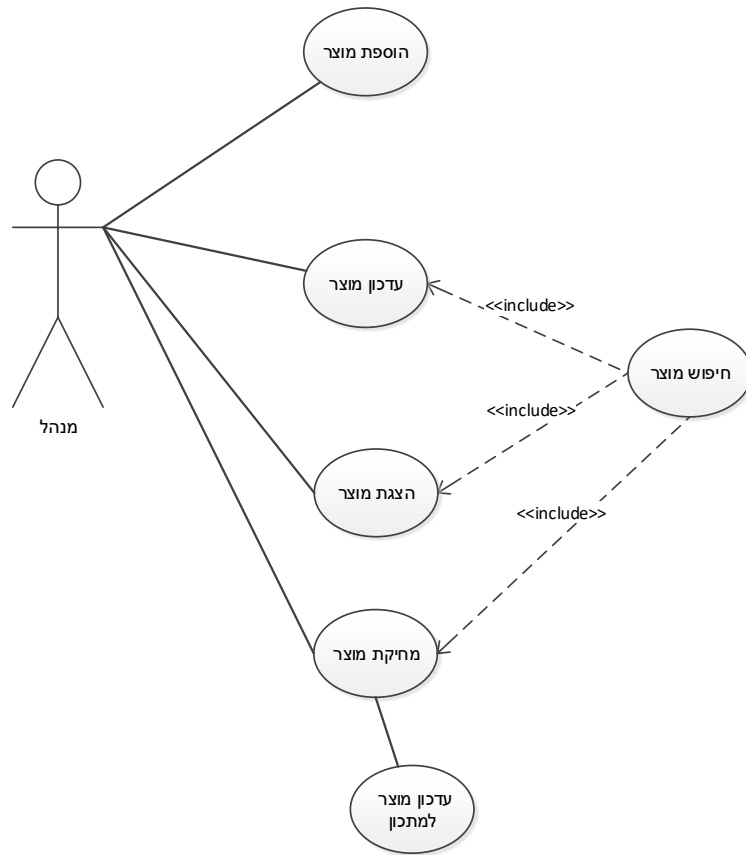
Use Case



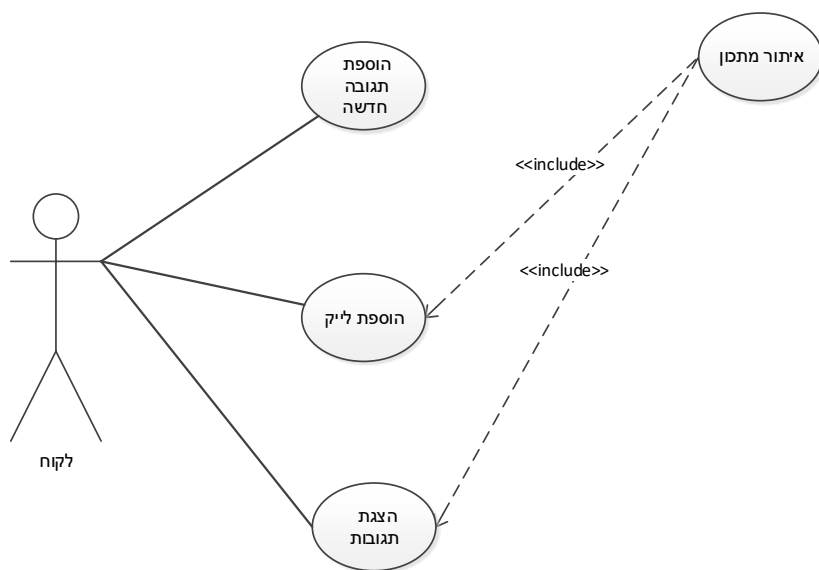
מערכת לטיפול במתכונים

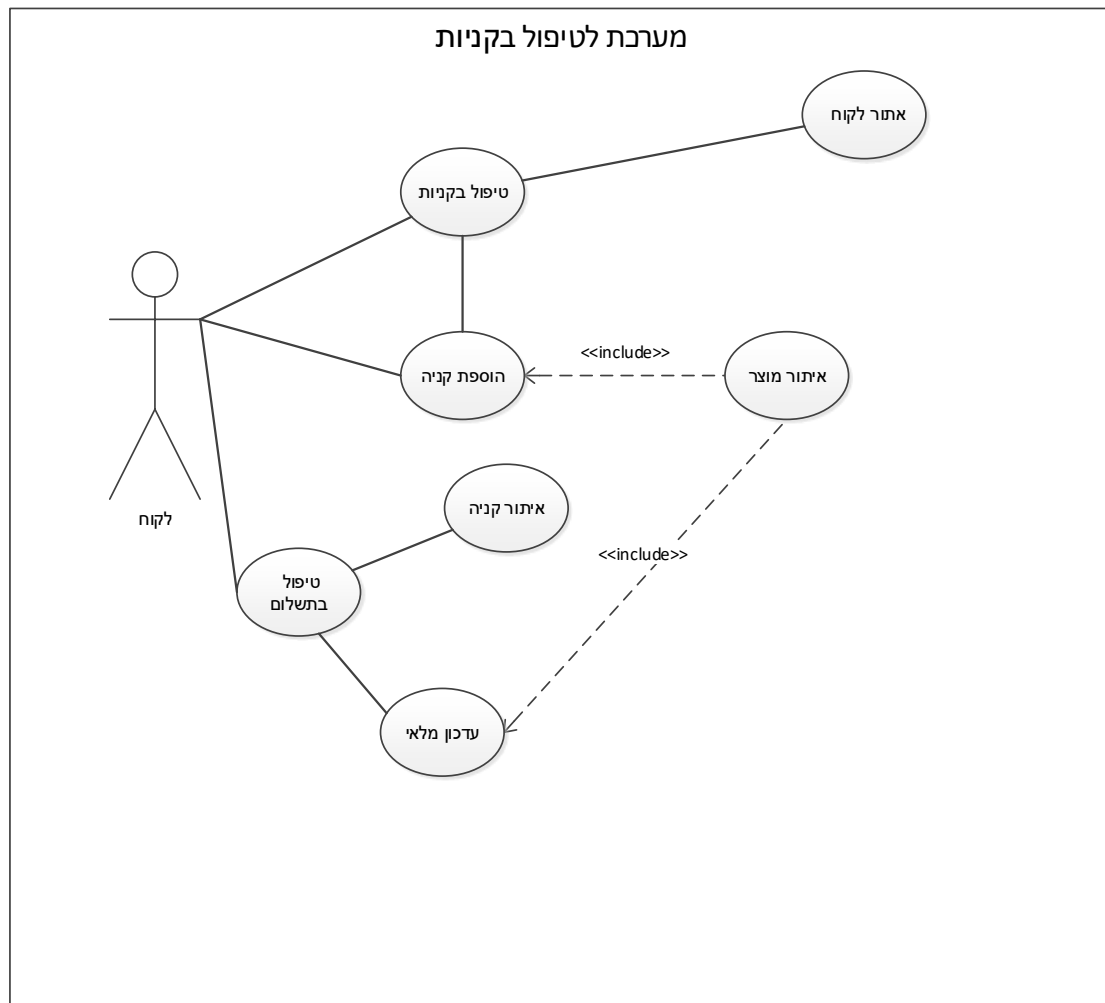


מערכת לטיפול במוצרים



מערכת לטיפול בתגובות





מבנה וארכיטקטורה

תאור המערכת

תוכנת "קייק בייק" היא תוכנה לניהול מכירת מוצרי אפיה המבצעת את

התהליכים:

1. טיפול במוצרים
2. טיפול במתכונים
3. טיפול בתגובות

4. טיפול בלקוחות

5. טיפול בקניות

סביבת הפיתוח ושפות תכנות:

סביבת הפיתוח

Visual Studio 2022

מסד נתונים Access 2016

שפות תכנות

צד לקוח: wpf [Xaml + C#]

השכבות:

שכבת הנתונים בנויה בצורה של ORM מכילה שני פרויקטים Model, ViewModel

שכבת התצוגה

התוכנית נבנתה בצורת חשיבה של – DNA מודל שלושת השכבות.

התוכנית חולקה לשכבות 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 ואבטחת הנתונים והפונקציות. פרוט השכבות:

1. Model – שכבה המטפלת בכל המחלקות המייצגות את המידע בפרויקט ופונקציות לוגיות שונות.
2. ViewModel – מטפלת בקישור התוכנה למסד הנתונים
3. GUI – Grapihic User Inteface – מטפלת בממשק משתמש מבוצע ע"י אפליקציה בWPF

1. שכבת הModel

שכבת המודל משקפת את מבנה הטבלאות במסד הנתונים.

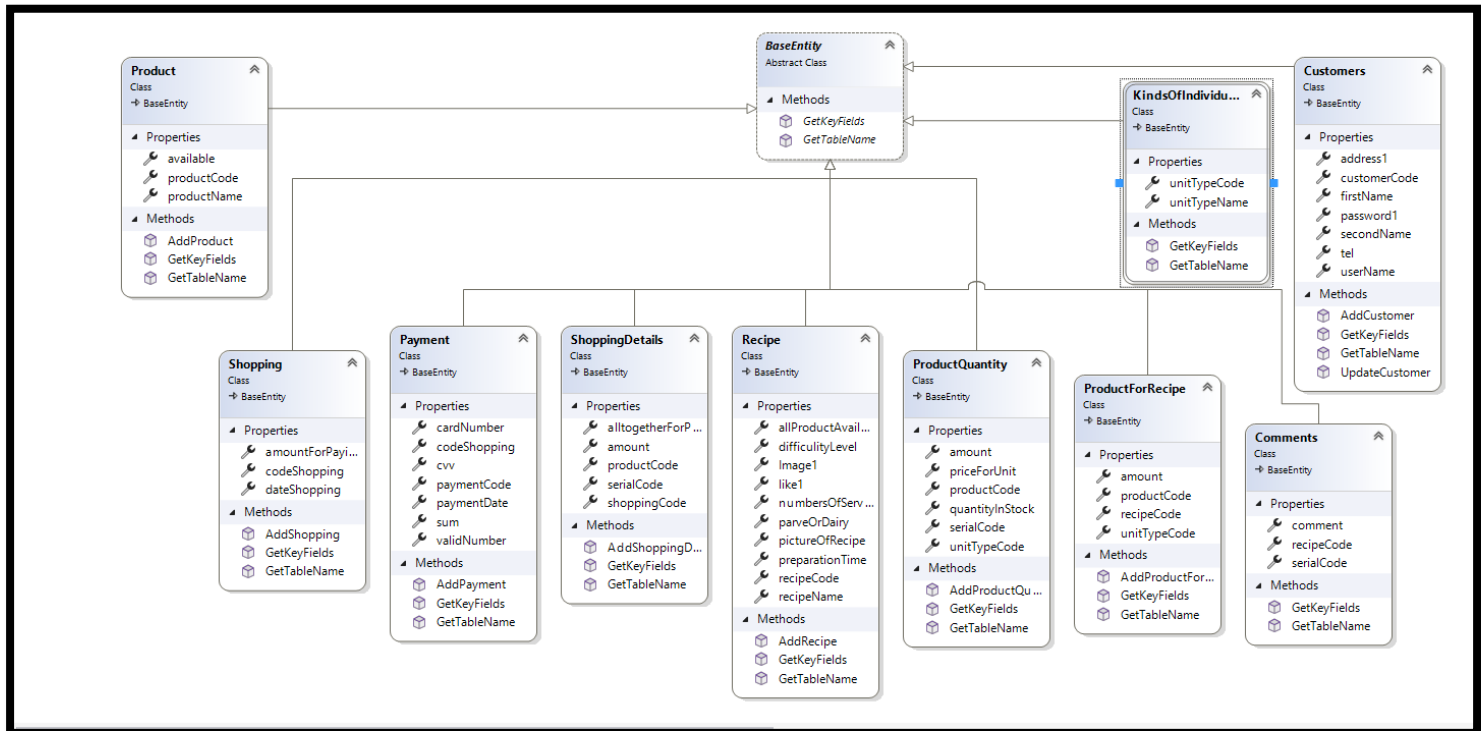
באמצעות המחלקות ברכיב זה ממירים את הייצוג הרלציוני (טבלאי) של מסד

הנתונים בייצוג OOP המתאים לקוד.

לכל טבלה שרלוונטית בפרויקט ישנה מחלקה ב-Model.

במחלקות הללו ישנם קשרים בין הטבלאות באמצעות יחסי הכלה וכן שדות חישוביים.

תרשים המחלקות UML



מחלקת BaseEntity מחלקת אב אבסטרקטית המטפלת ברשומה בודדה מתוך הטבלה.

המחלקה מאחדת את הפונקציונליות של כל סוגי רשומות היחיד. וממנה יורשים סוגים שונים של מחלקות יחיד.

קוד מחלקת BaseEntity:

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

```

public class Customers : BaseEntity//לקוח
{
    public int customerCode { get; set; } //קוד לקוח
    public string firstName { get; set; } //שם פרטי
    public string secondName { get; set; } //שם משפחה
    public string address1 { get; set; } //כתובת
    public string tel { get; set; } //טלפון
    public string password1 { get; set; } //סיסמה
    public string userName { get; set; } //שם משתמש

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

```

```

public class Product:BaseEntity//מוצר
{
    public int productCode { get; set; } //קוד מוצר
    public string productName { get; set; } //שם מוצר
    public bool available { get; set; } //זמינות

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

```

```

public class Recipe:BaseEntity//מתכון
{
    public int recipeCode { get; set; } //קוד מתכון
    public string recipeName { get; set; } //שם מתכון
    public string difficultyLevel { get; set; } //רמת קושי
    public int preparationTime { get; set; } //זמן הכנה
    public int numbersOfServings { get; set; } //מספר מנות
    public string parveOrDairy { get; set; } //פרווה/חלבי
    public bool allProductAvailable { get; set; } //האם כל המוצרים קיימים
    public int like1 { get; set; } //
    public string pictureOfRecipe { get; set; } //
    public BitmapImage Image1
    {
        get
        {
            return MyImage.GetImage(pictureOfRecipe);
        }
    }
}

```

```

public class Comments : BaseEntity//תגובות
{
    public Recipe recipeCode { get; set; } //קוד מתכון
    public int serialCode { get; set; } //קוד סידורי
    public string comment { get; set; } //תגובה
}

```



```

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

    public class KindsOfIndividuals : BaseEntity//סוגי יחידות
    {
        public int unitTypeCode { get; set; } //קוד סוג יחידה
        public string unitTypeName { get; set; } //שם סוג יחידה

        public override string GetTableName()//
        {
            return "KindsOfIndividuals";
        }

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

```

שכבת ה-ViewModel

מחלקת המיפוי היא המחלקה המקשרת בין מחלקת המודל לבין הטבלה המקבילה במסד הנתונים.

באמצעותה שולפים מידע עבור מחלקות ה-Model, מעדכנים מידע, מוסיפים ומוחקים רשומות.

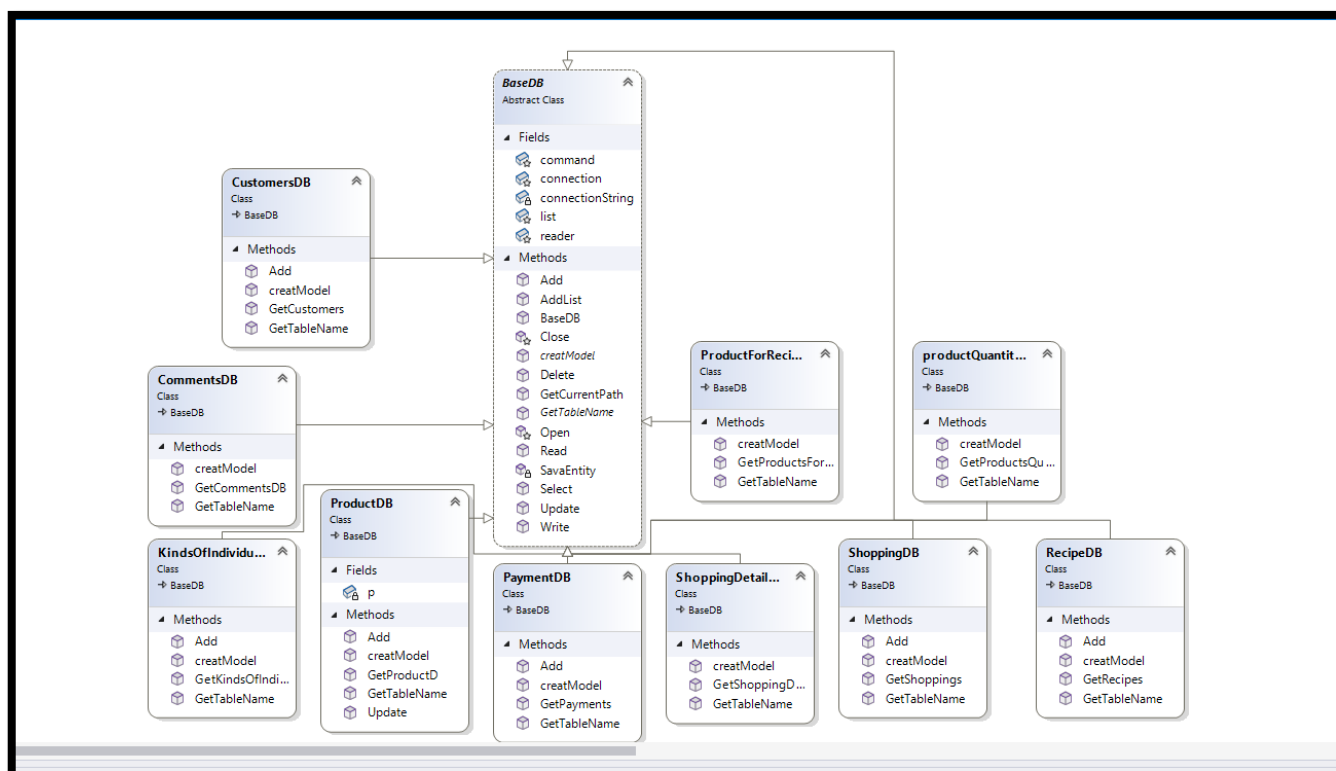
מחלקת המיפוי מאפשרת לבצע CRUD (Create, Read, Update, Delete) מלא למידע הקיים בטבלה.

המחלקות המוכלות בשכבה :

מחלקת בסיס לטיפול בטבלה אחת מתוך ה-BaseDB – DataBase
 מחלקה עבור כל טבלה מה-BaseDB היורשת מהמחלקה BaseDB
 מחלקות *nameTableRow* –תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני.

תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני

תרשים המחלקות UML



המחלקה BaseDB:

המחלקה BaseDB היא היחידה המתקשרת עם שכבת ה DAL מנהלת קשר עם טבלה בודדת מתוך ה DataBase. לא ניתן 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.

קוד המחלקה BaseDB:

```

public abstract class BaseDB
{
    private string connectionString; // מחרוזת הקישור מכיל פרטים להתחברות לקובץ
    protected OleDbConnection connection; // מייצר את החיבור פותח וסוגר
    protected OleDbCommand command; // מבצע את הפקודה - שליח
    protected OleDbDataReader reader; // קורא את התשובה
    protected List<BaseEntity> list = new List<BaseEntity>(); // שומר את נתוני הטבלה כאובייקטים ברשימה

    public BaseDB()
    {
        connectionString = @"provider=Microsoft.ACE.OLEDB.12.0;Data
source=" + GetCurrentPath() + "Data\\bakingProduct.accdb";
        connection = new OleDbConnection(connectionString);
        command = new OleDbCommand();
        command.Connection = connection;
        Select();
    }

    protected void Open() // פותחת קישור

```

```

{
    command.Connection = connection;
    if (connection.State != ConnectionState.Open)
        connection.Open();
}

protected void Close()//סוגרת קישור
{
    if (connection.State == ConnectionState.Open)
        connection.Close();
}

public void Read()//קוראת נתונים מטבלה מסוימת ומכניסה לרשימה
{
    list.Clear();
    command.CommandText = "select * from " + GetTableName();
    reader = command.ExecuteReader();
    while(reader .Read ())
    {
        list.Add(creatModel());
    }
    if (reader != null)
        reader.Close();
}

public void Select()//מבצע את כל השליפה
{
    Open();
    Read();
    Close();
}

public abstract string GetTableName();
public abstract BaseEntity creatModel();
public static string GetCurrentPath()//הנתיב את מחזירה - נוכחי נתיב החזור
של הנוכחי
{
    string path = System.IO.Directory.GetCurrentDirectory();
    string[] arr = path.Split('\\');
    path = "";
    for (int i = 0; i < arr.Length - 2; i++)
    {
        path += arr[i] + "\\";
    }
    return path;
}

private bool SavaEntity()//שוב ושולפת האקסס את מעדכנת
{
    Open();
    bool x = Write();
    if (x)//התבצע השינוי אם
    {
        Read();//הטבלה את מחדש קורא
        Close();
        return true;
    }
    else
    {
        Close();
        return false;
    }
}

```

```

public bool Write()
{
    int x = command.ExecuteNonQuery(); //הפקודה הפעלת
    return x > 0;
}
public virtual bool Add(BaseEntity entity)
{
    command.CommandText = SQLBuilder.InsertSQL(entity);
    return SavaEntity();
}
public virtual bool AddList(List<BaseEntity> entities) //הוספה מבצעת
לאקסס פריטים של
{
    int x = 0;
    Open();
    foreach (var item in entities)
    {
        command.CommandText = SQLBuilder.InsertSQL(item);
        x += command.ExecuteNonQuery(); //הפקודה הפעלת
    }

    if (x > 0) //התבצע השינוי אם
    {
        Read(); //הטבלה את מחדש קורא
        Close();
        return true;
    }
    else
    {
        Close();
        return false;
    }
}

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

public bool Delete(BaseEntity entity)
{
    command.CommandText = SQLBuilder.DeleteSQL(entity);
    return SavaEntity();
}
}

```

קוד המחלקה MyDB - מחלקה היוצרת עצמים סטטיים עבור שאר מחלקות ViewModel

```
public class MyDB
{
    public static RecipeDB recipeDB = new RecipeDB();
    public static CommentsDB commentsDB = new CommentsDB();
    public static ProductDB productDB = new ProductDB();
    public static KindsOfIndividualsDB kindsOfIndividualsDB = new KindsOfIndividualsDB();
    public static ProductForRecipeDB ProductForRecipeDB = new ProductForRecipeDB();
    public static productQuantityDB productQuantityDB = new productQuantityDB();
    public static CustomersDB customersDB = new CustomersDB();
    public static ShoppingDB shoppingDB = new ShoppingDB();
    public static ShoppingDetailsDB shoppingDetailsDB = new ShoppingDetailsDB();
    public static PaymentDB paymentDB = new PaymentDB();
}
```

```
public class CustomersDB:BaseDB//לקוחות
{
    public override BaseEntity creatModel()
    {
        Customers item = new Customers();
        item.customerCode = Convert.ToInt32(reader["customerCode"]);
        item.firstName = reader["firstName"].ToString();

        item.secondName = reader["secondName"].ToString();
        item.address1 = reader["address1"].ToString();
        item.tel = reader["tel"].ToString();

        item.password1 = reader["password1"].ToString();
        item.userName = reader["userName"].ToString();

        return item;
    }
}
```

```
public class ProductDB : BaseDB//מוצר
{
    Product p= new Product();

    public override BaseEntity creatModel()
    {
        Product item = new Product();
        item.productCode = Convert.ToInt32(reader["productCode"]);
        item.productName = reader["productName"].ToString();

        item.available = Convert.ToBoolean(reader["available"]);

        return item;
    }
}
```

```

public class RecipeDB:BaseDB//מתכון
{
    public override BaseEntity creatModel()
    {
        Recipe item = new Recipe();
        item.recipeCode = Convert.ToInt32(reader["recipeCode"]);
        item.recipeName = reader["recipeName"].ToString();

        item.difficultyLevel = reader["difficultyLevel"].ToString();
        item.preparationTime =
Convert.ToInt32(reader["preparationTime"]);
        item.numbersOfServings =
Convert.ToInt32(reader["numbersOfServings"]);

        item.parveOrDairy = reader["parveOrDairy"].ToString();
        item.like1 = Convert.ToInt32(reader["like1"]);
        item.pictureOfRecipe =
reader["pictureOfRecipe"].ToString();
        return item;
    }
}

```

```

public class CommentsDB : BaseDB//תגובות
{
    public override BaseEntity creatModel()
    {
        Comments item = new Comments();
        int recipeCode = Convert.ToInt32(reader["recipeCode"]);
        item.recipeCode = MyDB.recipeDB.GetRecipes().FirstOrDefault(x
=> x.recipeCode == recipeCode);
        item.serialCode = Convert.ToInt32(reader["serialCode"]);

        item.comment = reader["comment"].ToString();

        return item;
    }
}

```

```

public class KindsOfIndividualsDB:BaseDB//סוגי יחידות
{
    public override BaseEntity creatModel()
    {
        KindsOfIndividuals item = new KindsOfIndividuals();
        item.unitTypeCode = Convert.ToInt32(reader["unitTypeCode"]);
        item.unitTypeName = reader["unitTypeName"].ToString();
        return item;
    }
    public override string GetTableName()
    {
        return "KindsOfIndividuals";
    }
}

```

קוד המחלקה

מחלקת SQLBuilder

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

קוד המחלקה:

```
public class SQLBuilder
{
    public static string InsertSQL(BaseEntity entity)
    {
        Type type = entity.GetType();
        string command = "Insert Into " + entity.GetTableName() + "
(";
        string values = " Values (";
        foreach (var item in type.GetProperties())
        {
            if (item.CanWrite)
            {
                object value = item.GetValue(entity);
                command += item.Name + " , ";
                values += PutValue(value) + " , ";
            }
        }
        command = command.Substring(0, command.Length - 2) + ")";
        values = values.Substring(0, values.Length - 2) + ")";
        return command + values;
    }

    public static string UpdateSQL(BaseEntity entity)
    {
        Type type = entity.GetType();
        string command = "Update " + entity.GetTableName() + " set ";
        foreach (var item in type.GetProperties())
        {
            if (item.CanWrite)
            {
                object value = item.GetValue(entity);
                command += item.Name + " = " + PutValue(value) + " , ";
            }
        }
        string where = "";
        foreach (var item in entity.GetKeyFields())
        {
            if (where != "")
                where += " And ";
            object value =
entity.GetType().GetProperty(item).GetValue(entity);
            where += item + " = " + PutValue(value);
        }
        command = command.Substring(0, command.Length - 2) + " Where "
+ where;
        return command;
    }
}
```

שכבת היישום GUI

להלן מספר טפסים משכבה זו:

טופס MyStyle - עיצוב

טופס שמכיל את כל הגדרות העיצוב וכל טפסי התצוגה מקושרים אליו ומממשים את הגדרות העיצוב שבו.

```
<ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <Style TargetType="Control" x:Key="baseStyle">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="HorizontalAlignment" Value="Left"/>
        <!--<Setter Property="Background" Value="LightPink"/>-->
        <Setter Property="Width" Value="200"/>
    </Style>
    <Style TargetType="TextBox" BasedOn="{StaticResource baseStyle}">
        <Style.Triggers>
            <Trigger Property="Validation.HasError" Value="true">
                <Setter Property="Validation.ErrorTemplate">
                    <Setter.Value>
                        <ControlTemplate>
                            <StackPanel>
                                <AdornedElementPlaceholder/>
                                <TextBlock HorizontalAlignment="Right"
Text="{Binding [0].ErrorContent}" Foreground="LightPink"/>
                            </StackPanel>
                        </ControlTemplate>
                    </Setter.Value>
                </Setter>
            </Trigger>
            <Setter Property="BorderBrush" Value="LightPink"/>
        </Style.Triggers>
    </Style>
    <Style TargetType="ComboBox" BasedOn="{StaticResource baseStyle}">
    </Style>

    <Style TargetType="DatePicker" BasedOn="{StaticResource baseStyle}">
    </Style>
    <Style TargetType="PasswordBox" BasedOn="{StaticResource baseStyle}">
    </Style>

    <Style TargetType="TextBlock">
        <Setter Property="FontSize" Value="18"/>
    </Style>
    <Style TargetType="Grid">
        <Setter Property="FlowDirection" Value="RightToLeft"/>
        <Setter Property="Margin" Value="5"/>
    </Style>
    <Style TargetType="StackPanel">
        <Setter Property="FlowDirection" Value="RightToLeft"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>
    <Style TargetType="UniformGrid">
        <Setter Property="FlowDirection" Value="RightToLeft"/>
    </Style>
```



```

</Style>

<Style TargetType="Button">
    <Setter Property="Background" Value="LightPink"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="Margin" Value="5"/>
    <Setter Property="MaxWidth" Value="200"/>
    <Setter Property="MinWidth" Value="50"/>
    <Setter Property="Height" Value="42"/>
    <Setter Property="Padding" Value="5"/>
</Style>

<Style TargetType="Button" x:Key="LikeButtonStyle">
    <Setter Property="Background" Value="White"/>
    <Setter Property="BorderBrush" Value="Gray"/>
    <Setter Property="BorderThickness" Value="1"/>
    <Setter Property="Padding" Value="5,2"/>
    <Setter Property="Cursor" Value="Hand"/>
</Style>

<Style TargetType="TextBlock" x:Key="LikeCountStyle">
    <Setter Property="Margin" Value="5,0,0,0"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="VerticalAlignment" Value="Center"/>
</Style>

</ResourceDictionary>

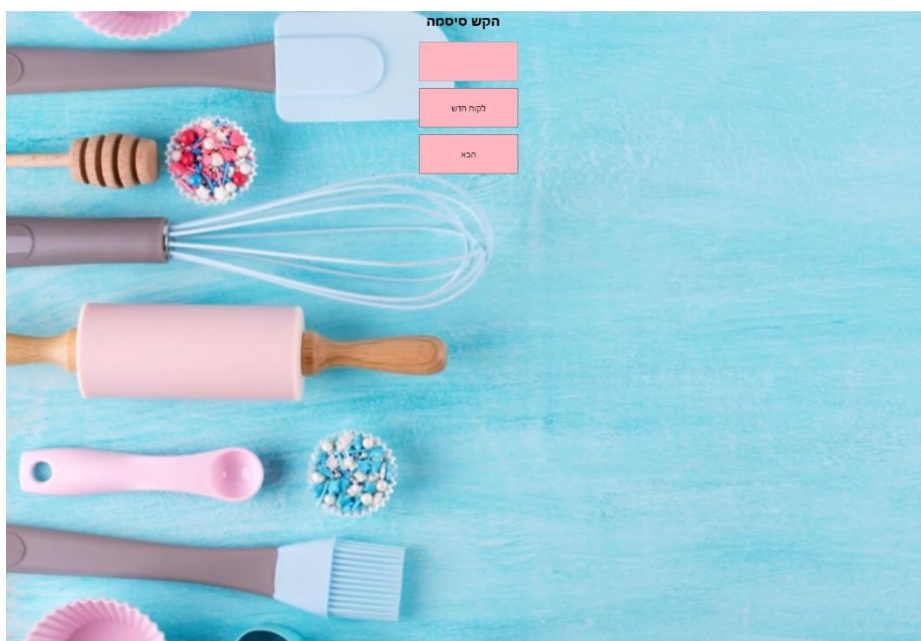
```

דף כניסת משתמשים – התחברות

המשתמש מקיש את סיסמתו ובהתאם לסיסמתו נפתח המסך הבא

ללקוח דף הצגת מתכונים ואפשרות למעבר לקניה

ולמנהל עדכון מוצרים ומתכונים וכו'



קוד xaml:

```
<Page x:Class="BakingProductProject.Login"
      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:BakingProductProject"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="Login">

    <Grid>
        <StackPanel >
            <StackPanel.Background>
                <ImageBrush Stretch="Uniform"
ImageSource="/MyPictures/picture.png"/>
            </StackPanel.Background>
            <TextBlock Text="סיסמה הקש" Height="36" Width="128"
FontWeight="Black"/>
            <TextBox Height="36" Width="127" x:Name="txtId"/>
            <TextBlock Text="שגויים פרטים" Visibility="Collapsed"
x:Name="txtError"/>

            <Button x:Name="btnAddCustomer" Content="חדש לקוח"
Click="btnAddCustomer_Click" Height="36" Width="127"/>
            <Button Content="הבא" Click="Button_Click" Height="36"
Width="128"/>

        </StackPanel>

    </Grid>
</Page>
```

קוד c#:

```
public partial class Login : Page
{
    public Login()
    {
        InitializeComponent();
    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
        txtError.Visibility = Visibility.Collapsed;
        Global.customersUser =
MyDB.customersDB.GetCustomers().FirstOrDefault(x => x.password1 ==
txtId.Text);
        if (Global.customersUser != null)
        {
            txtId.Text = "";
            NavigationService nav =
NavigationService.GetNavigationService(this);
```

```

        nav.Navigate(new CustomrtEnter());
    }
    else
    {
        // Global.customersUser = MyDB.customers(txtId.Text);
        if (Global.customersUser != null)
        {
            txtId.Text = "";

            NavigationService nav =
NavigationService.GetNavigationService(this); //לעבוד בדיקה לעשות האם לבדוק
            nav.Navigate(new WorkerEnter());

        }
        else
        {
            if (txtId.Text == "מנהל אני")
            {
                Global.IsManager = true;
                txtId.Text = "";

                NavigationService nav =
NavigationService.GetNavigationService(this);
                nav.Navigate(new MainWindow());
            }
            else
            {
                txtError.Visibility = Visibility.Visible;
            }
        }
    }

}

}

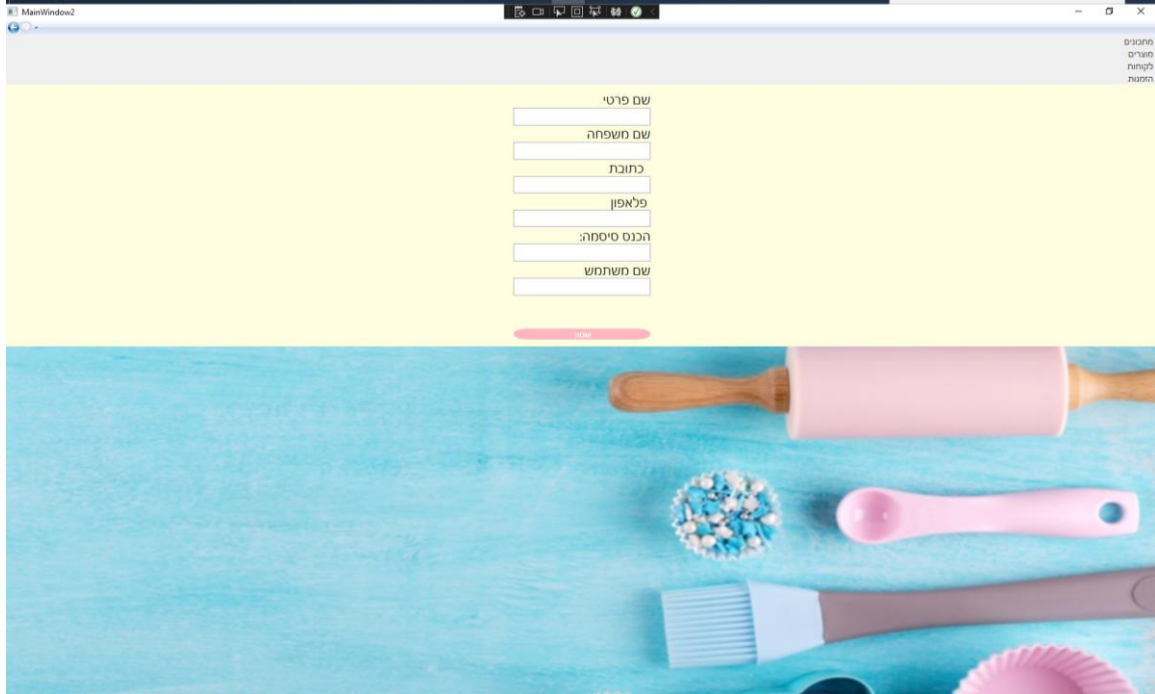
private void btnAddCustomer_Click(object sender, RoutedEventArgs
e)
{
    NavigationService nav =
NavigationService.GetNavigationService(this);
    nav.Navigate(new AddUpdateCustomers());
}

}

```

דף הוספת לקוח חדש

הלקוח מכניס את פרטיו ולוחץ על שמור לשמירת פרטיו במערכת.



קוד xaml:

```
<Page x:Class="BakingProductProject.UI.AddUpdateCustomers"
      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:BakingProductProject.UI"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="AddUpdateCustomers">
    <Page.Resources>
        <ResourceDictionary Source="MyStyle.xaml"/>
    </Page.Resources>
    <Grid>
        <StackPanel HorizontalAlignment="Center">

            <TextBlock Text="שם פרטי"/>
            <TextBox>
                <Binding Path="firstName">
                    <Binding.ValidationRules>
                        <local:HebrewValide/>
                    </Binding.ValidationRules>
                </Binding>
            </TextBox>
            <TextBlock Text="שם משפחה"/>
            <TextBox>
                <Binding Path="secondName">
                    <Binding.ValidationRules>
                        <local:HebrewValide/>
                    </Binding.ValidationRules>
                </Binding>
            </TextBox>
            <TextBlock Text="כתובת"/>
            <TextBox>
                <Binding Path="address1">
                    <Binding.ValidationRules>
```

```

        <local:AdressValide/>
    </Binding.ValidationRules>
</Binding>
</TextBox>
<TextBlock Text=" פלאפון " />
<TextBox Height="25">
    <Binding Path="tel">
        <Binding.ValidationRules>
            <local:NumValide/>
        </Binding.ValidationRules>
    </Binding>
</TextBox>

<TextBlock Text="סיסמה הכנס:"/>

<TextBox>
    <Binding Path="password1">
        <Binding.ValidationRules>
            <local:PasswordValide/>
        </Binding.ValidationRules>
    </Binding>
</TextBox>
<TextBlock Text="משתמש שם"/>
<TextBox>
    <Binding Path="userName">
        <Binding.ValidationRules>
            <local:HebrewValide/>
        </Binding.ValidationRules>
    </Binding>
</TextBox>

<TextBlock x:Name="txtErrorr" Foreground="Red"/>
<TextBlock x:Name="txtOkC" Foreground="Green"/>

<Button Content="שמור" Click="Button_Click" />

</StackPanel>
</Grid>
</Page>

```

קוד C# :

```

public partial class AddUpdateCustomers : Page
{
    Customers c;
    int status = 1;
    public AddUpdateCustomers()
    {
        InitializeComponent();
        c = new Customers();
    }
}

```

```

        this.DataContext = c;
    }
    public AddUpdateCustomers(Customers c)
    {
        InitializeComponent();
        this.c = c;
        this.DataContext = c;
    }

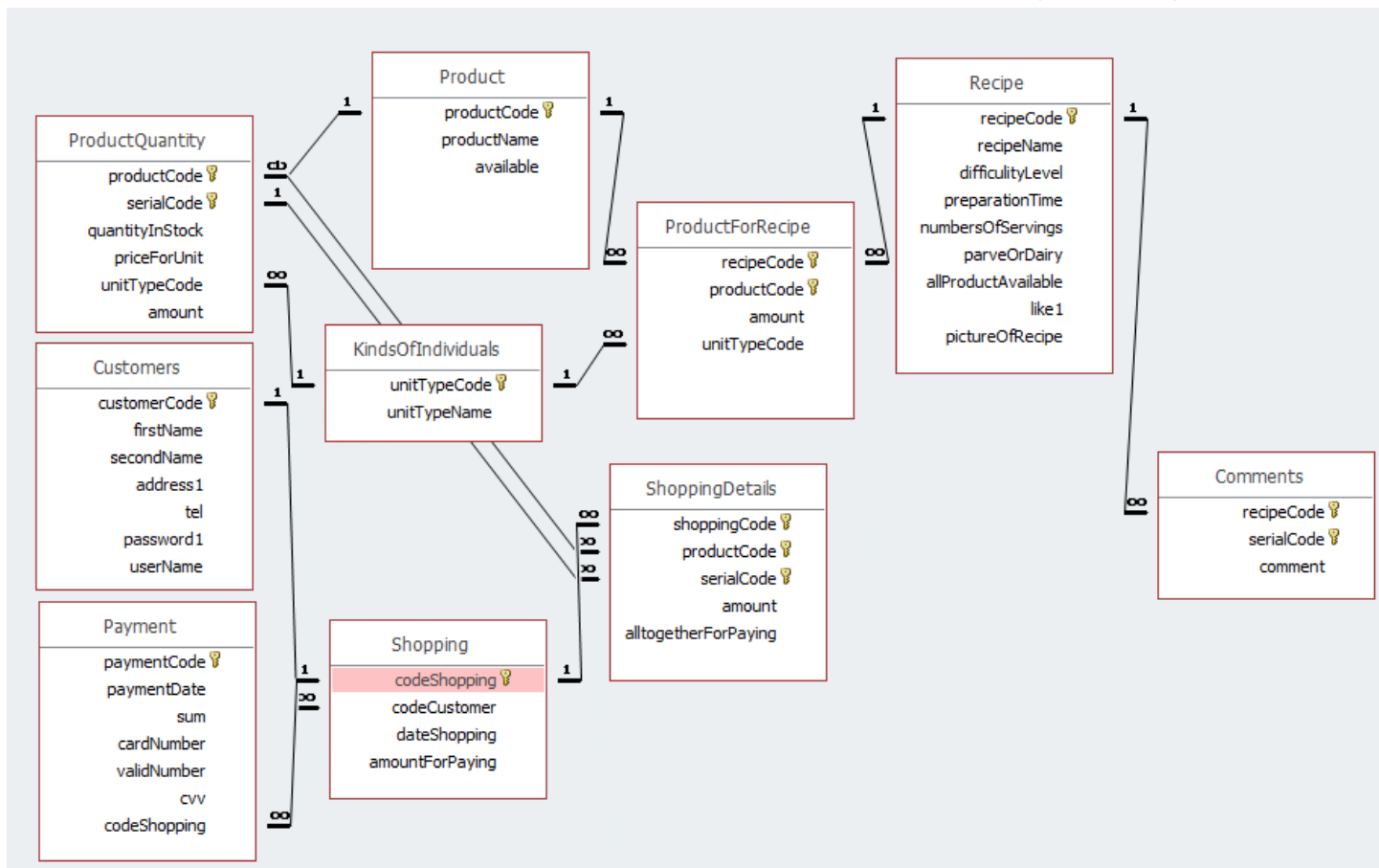
    private void Button_Click(object sender, RoutedEventArgs e)
    {
        if (c.firstName == null || c.secondName == null || c.address1
== null || c.tel == null || c.password1 == null || c.userName == null)
        {
            txtErrorr.Text = "חובה שדות הם הנתונים כל. נתונים לך חסרים";
        }
        else
        {
            if (status == 1)
            {
                MyDB.customersDB.Add(c);
                txtOkC.Text = c.secondName + c.firstName + "נשמר";
            }
            if (status == 2)
            {
                MyDB.customersDB.Update(c);
                txtOkC.Text = c.secondName + c.firstName + "עודכן";
            }
        }

        if (Global.IsManager == false && Global.customersUser
!= null)
        {
            //הדף את מרוקנות אלו שורות שתי
            c = new Customers();
            if (Global.IsManager)
            {
                c = new Customers();
                this.DataContext = c;
            }
        }
    }
}

```

5. בסיס הנתונים

קשרי גומלין



הנחות יסוד לנרמול

1. לכל מוצר יש כמה גדלים.
2. כל מוצר קיים בהרבה מתכונים.
3. בכל מתכון יש הרבה מוצרים.
4. לכל מתכון יש הרבה תגובות.
5. להרבה תגובות יש מתכון אחד.
6. להרבה מוצרים יש מתכון.
7. הרבה מתכונים מכילים את אותו מוצר.
8. סוג יחידה יכול להופיע במוצרים בגודל רבים.
9. מוצר בגודל יכול להופיע הרבה פעמים בפירוט קניות של קניות שונות.
10. מוצר בגודל שונה יכול להופיע הרבה פעמים בפירוט קניות לקניה אחת.
11. קניה אחת מכילה הרבה פרטי קניה.
12. כל פירוט קניות שייך לקניה אחת.
13. ללקוח אחד יש הרבה קניות.
14. יש הרבה קניות השייכים ללקוח אחד.
15. לקניה אחת יש הרבה תשלומים.
16. הרבה תשלומים יכולים להיות בקניה אחת.
17. הרבה גדלים קיימים באותו מוצר.
18. הרבה מוצרים בגודל יכולים להופיע בסוג יחידה אחת.
19. סוג יחידה קיים במוצרים רבים במתכון.
20. הרבה מוצרים במתכון קיים סוג יחידה אחד.

טבלאות

תגובות-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	recipeCode	קוד מתכון	מספר	מספר
מפתח ראשי	serialCode	שם סידורי	מספר	מספר
	Comment	תגובה	טקסט	עברית בלבד

לקוחות-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	customerCode	קוד לקוח	מספר	מספר
	firstName	שם פרטי	טקסט	עברית בלבד
	secondName	שם משפחה	טקסט	עברית בלבד
	address	כתובת	טקסט	כתובת
	tel	טלפון	טקסט	טלפון או פלאפון
	password	סיסמה	טקסט	סיסמה
	userName	שם משתמש	טקסט	עברית בלבד

סוגי יחידים-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	unitTypeCode	קוד סוג יחידה	מספר	מספר

unitTypeName	שם סוג יחידה	טקסט	עברית בלבד
--------------	--------------	------	------------

תשלום-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח מספר	paymentCode	קוד תשלום	מספר	מספר
	paymentDate	תאריך תשלום	טקסט	
	sum	סכום		מספר
	cardNumber	מספר כרטיס		מספר
	validNumber	תוקף		מספר
	cvv	3 ספרות		מספר
	codeShopping	קוד קניה		מספר

מוצר-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח מספר	productCode	קוד מוצר	מספר	מספר
	productName	שם מוצר	טקסט	עברית בלבד
	available	האם המוצר קיים	כן/לא	

מוצרים למתכון-

	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	recipeCode	קוד מתכון	מספר	מספר
מפתח ראשי	productCode	קוד מוצר	מספר	מספר
	amount	כמות	מספר	מספר
	unitTypeCode	קוד סוג יחידה	מספר	מספר

כמות מוצר-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	productCode	קוד מוצר	מספר	מספר
מפתח ראשי	serialCode	קוד סידורי	מספר	מספר
	quantityInStock	כמות במלאי	מספר	מספר
	priceForUnit	מחיר ליחידה	מספר	מספר
	unitTypeCode	קוד סוג יחידה	מספר	מספר
	amount	כמות	מספר	מספר

מתכון-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	recipeCode	קוד מתכון	מספר	מספר
	recipeName	שם מתכון	טקסט	עברית בלבד

	טקסט	רמת קושי	difficultyLevel
מספר	מספר	זמן הכנה	preparationTime
מספר	מספר	מספר מנות	numbersOfServings
	טקסט	פרווה או חלבי	parveOrDairy
	כן/לא	האם כל המוצרים בחנות	allProductAvailable
מספר	מספר	"לייק"	like
	טקסט	תמונת מתכון	pictureOfRecipe

קניה-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	codeShopping	קוד קניה	מספר	מספר
	codeCustomer		מספר	מספר
	dateShopping	תאריך קניה	תאריך/שעה	
	amountForPaying	סכום לתשלום	מספר	מספר

פרטי קניה-				
	שם אנגלי	שם עברי	סוג	בדיקות תקינות
מפתח	codeShopping	קוד קניה	מספר	מספר
	Code		מספר	מספר
	serialCode	קוד סידורי	מספר	מספר

amount	כמות	מספר	מספר
alltogetherForPaying	סך הכל לתשלום	מספר	מספר

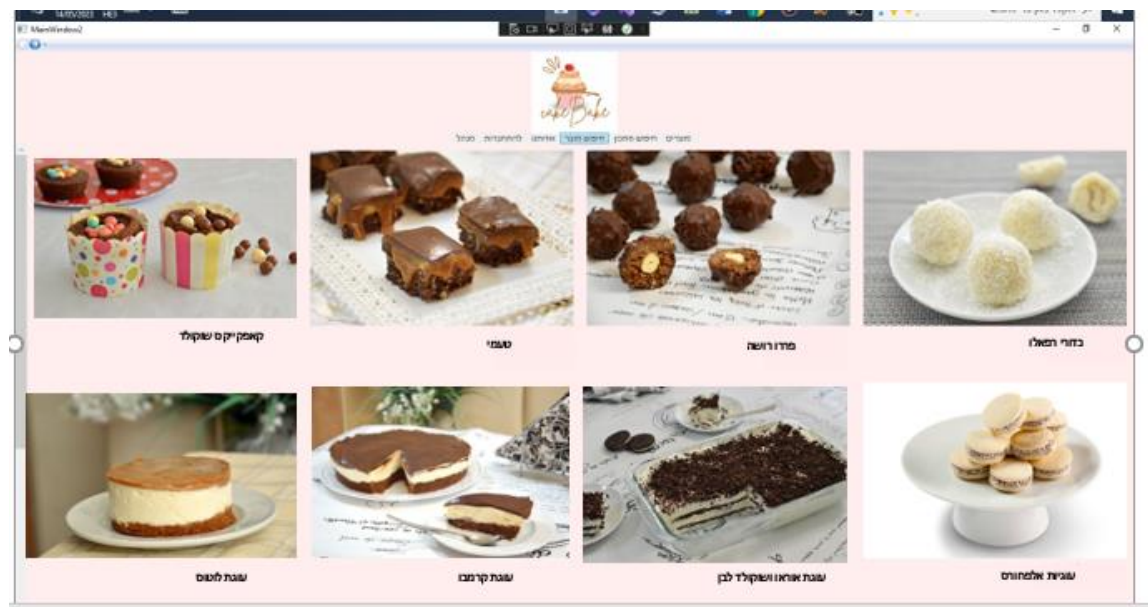
6. מדריך למשתמש

מערכת " קייק בייק " מספקת מגוון של סוגי מוצרים שונים ומתכונים ברמות שונות. הצורך עלה היות והרצון לאפות גבוה והטרחת המרובה ביציאה לקניית המוצרים הנדרשים גורם לעיתים אף ללכת למספר חניות כדי להשיג את המצרכים ומרתיע מלאפות.

לכן המערכת מרכזת את הקניות במקום אחד ודואג לחווית קניה ללקוחות. החברה מספקת מוצרי אפיה רבים במקום אחד והבאתם לבית הלקוח, העובדים אורזים את המוצרים בהתאם להזמנות הלקוחות בצורה מאורגנת ואסתטית .

דף הבית

בדף זה מוצג קטלוג העוגות ודרכו נכנסים למערכת.



מסך מוצרים

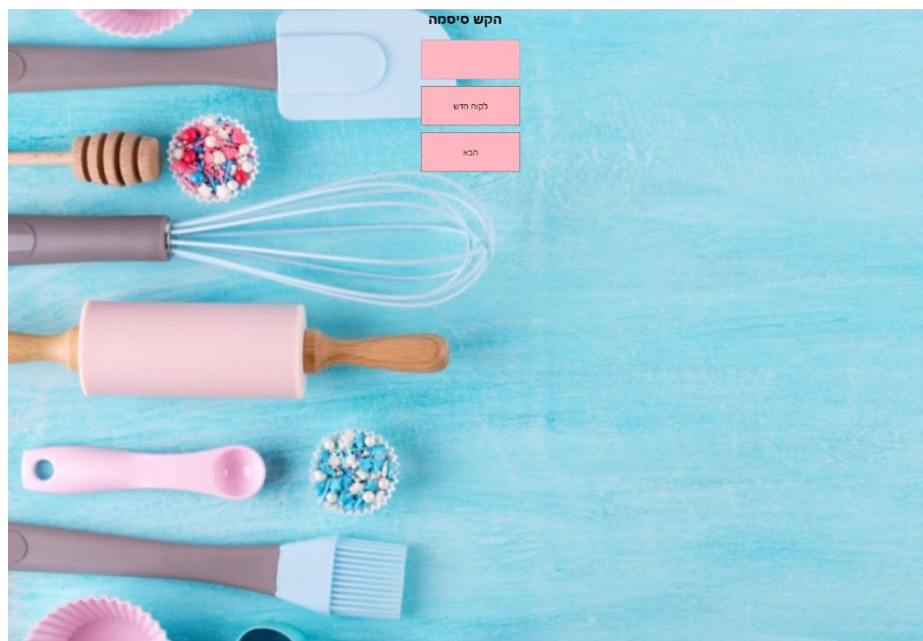
שם מוצר	חזק המוצר קיים
שוקולד לבן	שול
חמאה	שול
חלב	שול
קוקוס	שול
אגוז לוז	שול
קורנפלקס	שול
סמרח נאט	שול
סמרח נטלה	שול
שוקולד מריר	שול
שמן	שול
אגוז מר	שול
שוקולד חלב	שול
חמאת בוטנים	שול
דבש	שול
סוכר	שול
שמנת מתוקה	שול
קצפת עמית	שול
ריבת חלב	שול

דף אודותנו



הוקם בשנת 2015 "CakeBake" האתר
אנו מספקים מגוון רחב של מוצרי אפיה
וכן מתכונים ברמות שונות
וזה כדי להקל עליכם ולאפשר לכם
חוויית קניה אמיתית !!!
... והקליקו להזמנה בחרו במתכון
אנו נדאג לארוז לכם את המוצרים
בצורה אסטטית ומאורגנת ולשלוח לכיתכם בהקדם
אז בהצלחה ובתאבון!!!

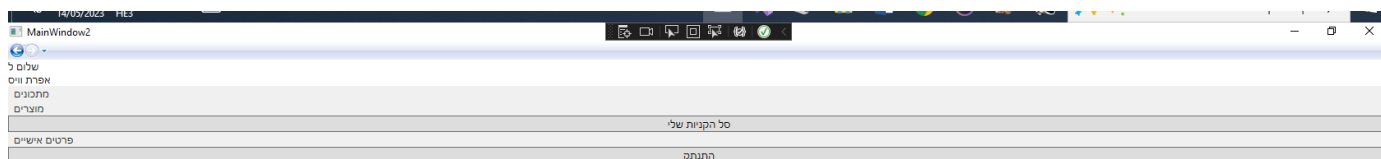
דף התחברות



לאחר הקשת הסיסמה מוצג מסך ללקוח

פרטי לקוח

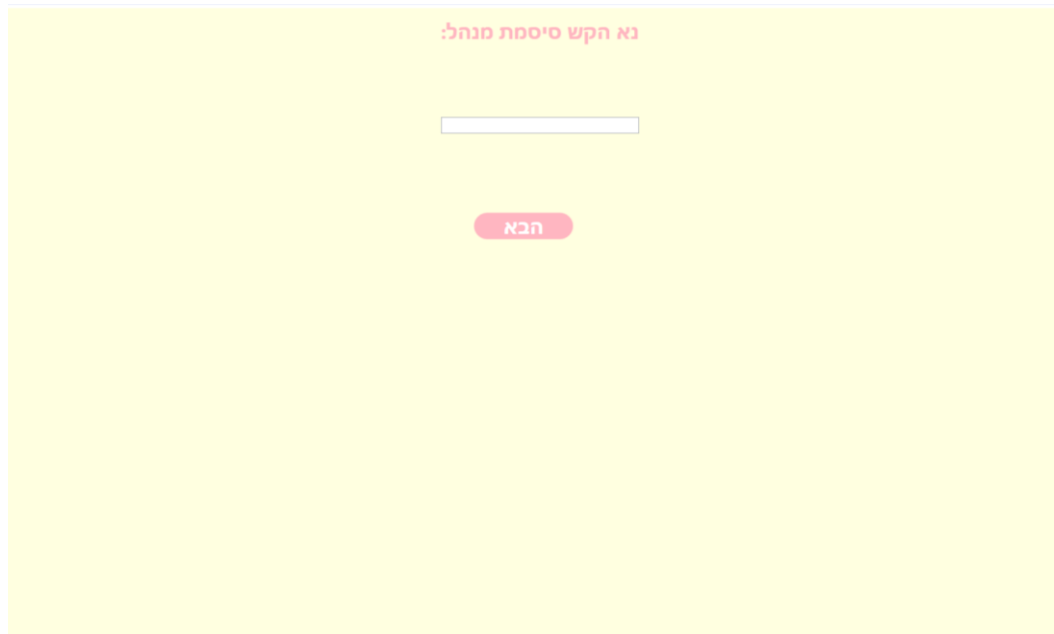
- צפיה בסל הקניות
- צפיה בפרטים
- ביצוע הזמנה חדשה
- צפיה במתכונים
- צפיה במוצרים



במידה והלקוח לא רשום במערכת הוא יכול להכניס את פרטיו ולהתחבר

טופס מנהל ראשי

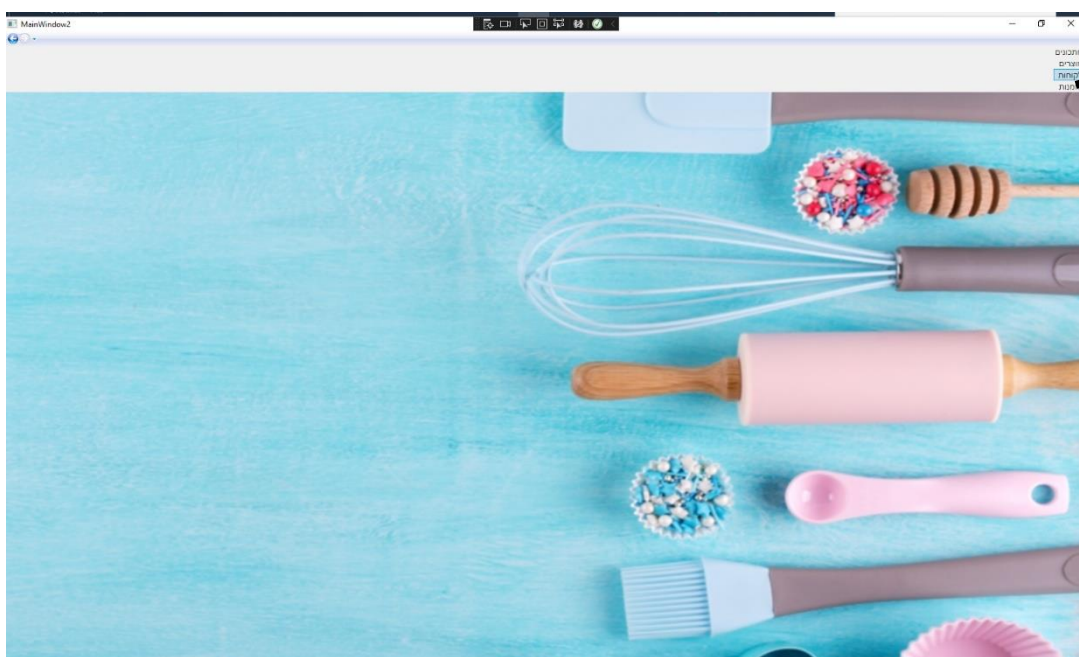
- יש גישה לכל המאגרים כמו מוצרים, מתכונים, לקוחות, קניות.



טופס מנהל

- צפיה בלקוחות
- צפיה במוצרים
- צפיה במתכונים
- הוספת מוצרים
- יצירת לקוח חדש

דף הבית של המנהל



מתכונים- הצג הכל
הוסף

מוצרים- הצג הכל
הוסף

לקוחות- הצג הכל
הוסף

הזמנות- הצג הכל
הוסף

דף הוספת מתכון
המנהל מכניס פרטי מתכון ותמונה ובוחר מרשימת המוצרים את המוצרים הנצרכים להכנת המתכון

The screenshot shows a web application window titled 'MainWindow2'. The interface is in Hebrew. It features a form for adding a recipe. The form is divided into several sections:

- שם מתכון** (Recipe Name): A text input field.
- רמת קושי** (Difficulty Level): A dropdown menu.
- זמן הכנה** (Preparation Time): A text input field.
- מספר מנות** (Number of Servings): A text input field.
- פרווה/חלבי** (Vegan/Dairy): A dropdown menu.
- לתיק** (For Diet): A text input field.
- All Products Available**: A checkbox.
- מוצרים למתכון זה** (Products for this recipe): A list of products with checkboxes and input fields for quantity.
- סוג יחידה** (Unit Type): A dropdown menu.
- כמות** (Quantity): A text input field.

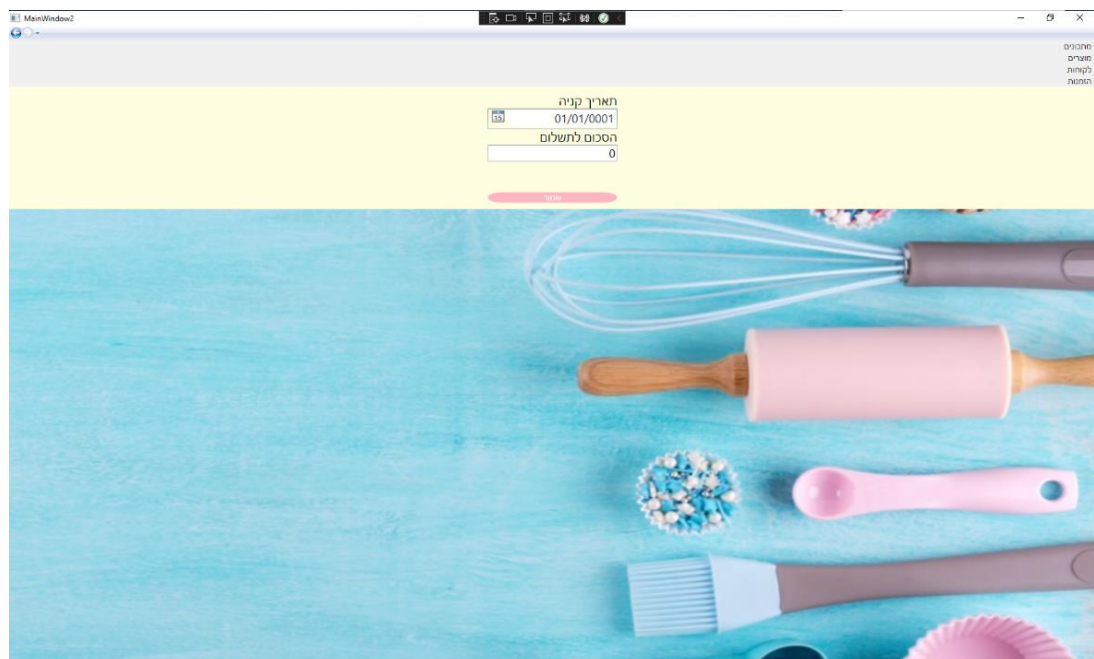
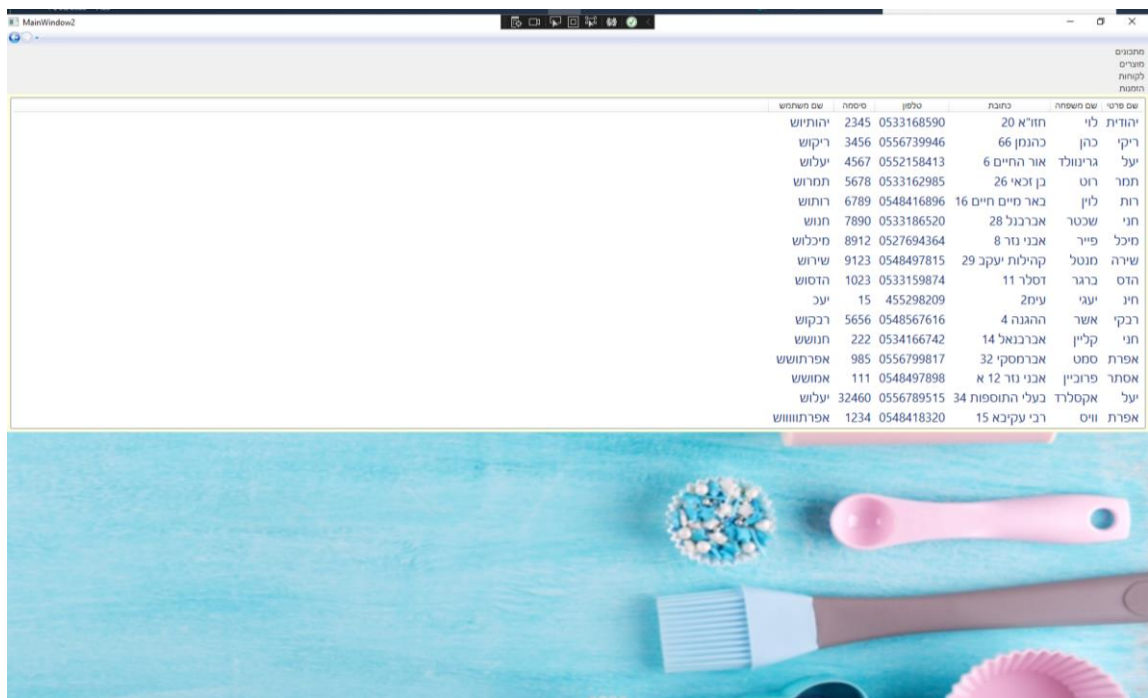
The background image shows various kitchen tools like a rolling pin, whisk, and brushes.

טופס הוספת מוצר
המנהל מכניס מוצר חדש ואת הגדלים הקיימים ממנו ומוסיף לרשימת המוצרים.

The screenshot shows a web application window titled 'MainWindow2'. The interface is in Hebrew. It features a form for adding a new product. The form is divided into several sections:

- שם מוצר** (Product Name): A text input field.
- האם המוצר קיים?** (Is the product existing?): A checkbox.
- הוספת מוצר נוסף** (Add another product): A button.

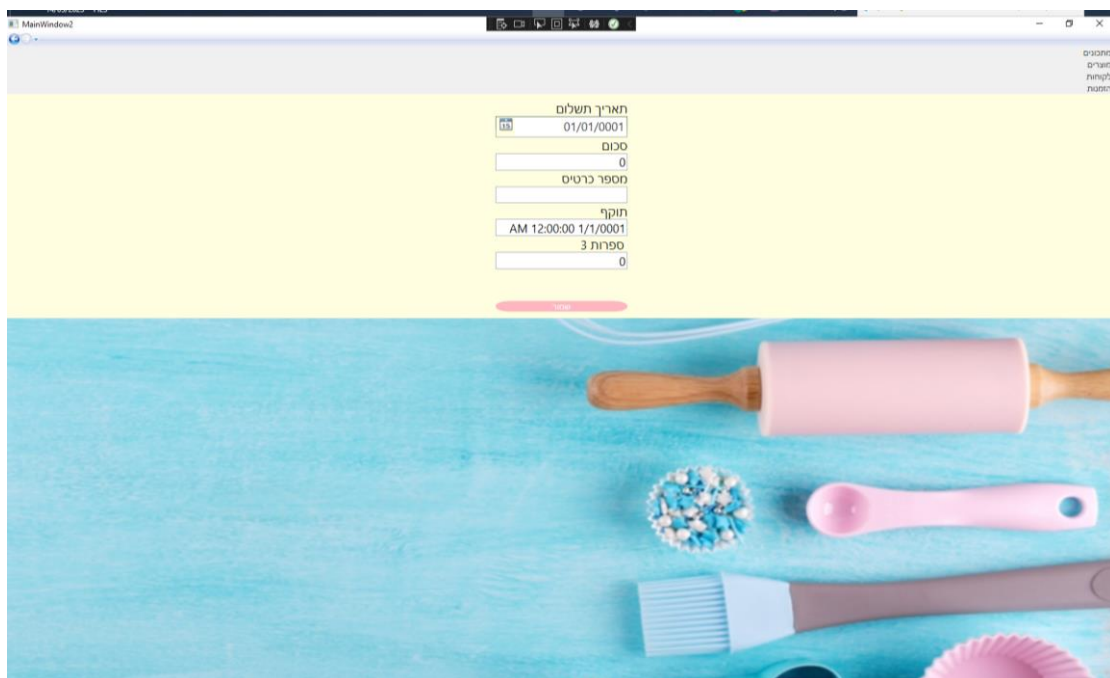
The background image shows various kitchen tools like a whisk, rolling pin, and brushes.



דף תשלום

טופס בצע קניה

- ע"י שמור הקניה מתווספת לאותו לקוח ההזמנה הסופית



7. רפלקציה

במשך עבודתי על פרויקט זה נהניתי לגלות לעומק את עולם התכנות. עבודה זו הייתה שונה מהחומרים שנלמדו בשנים קודמות. ע"י יצירת הפרויקט הרגשתי שאני נחשפת ליצירה שמתקבלת עם כתיבת הקוד. כמובן שהעבודה ערכה זמן רב ולכן זה העסיק אותי זמן ממושך מה שגרם לי לחשוב הרבה על התהליך ולשדרגו. אני מאד נהניתי לעבוד על הפרויקט משום שיש לי קרבה לעסק עליו נכתבה התוכנה, וכמובן בגלל התוצאה.

מכאן ואילך אייחס משמעות רבה יותר לעבודה וכן לארגון מסודר של הדברים באופן שיקל על המתכנת.

במהלך הכתיבה נתקלתי בקשיים כגון אירועים שהפסיקו את הלמידה (סגרים וכו') אך לא רק.

הכתיבה היתה מורכבת ופעמים רבות נאלצתי לשנות את העבודה הרבה שנעשתה למשהו אחר.

אם הייתי כותבת מחדש את התוכנה כיום הייתי מארגנת בדף רשימות ברורות של דברים שצריכים להיעשות וכן חושבת מראש על העיצוב ומייחסת לו משמעות רצינית.

אם העבודה הייתה נעשית בזוגות יכול להיות שהעומס הרב שהקשה במהלך הפרויקט היה מתמעט.

אך אם זאת הידיעה שהפרויקט הוא שלי נחמדה לא פחות.

סך הכול נהניתי מהעבודה על אף המורכבות.