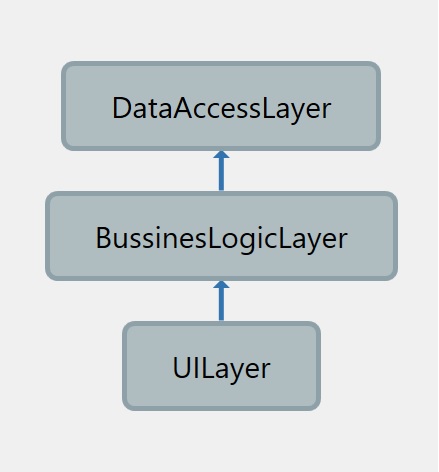
אם יש בעיה בפרוייקט ב GitHub, ניתן להוריד מ <https://www.dropbox.com/sh/15frpwsrt65wesw/AAD_upsmPzVoXvsRKwR3I4z0a?dl=0>

מגיש: עידן מלכה

**Price Comparison Project - Final**

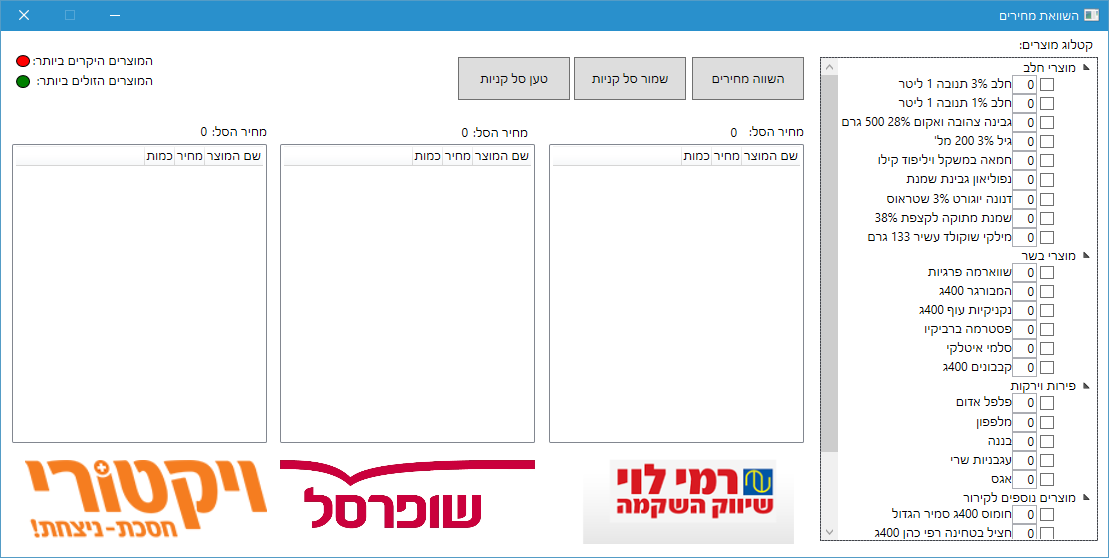
המערכת מורכבת מ 3 שכבות (פרוייקטים) הבונות את האפליקציה עפ"י גישה היררכית. השכבות הן שכבת ממשק המשתמש (UILayer), שכבת לוגיקת העבודה (BusinessLogicLayer) ושכבת הגישה למשאבים (DataAccessLayer).



דאגרמת התלויות (המיוצאת מ Visual Studio) :

1. UILayer : שכבת ממשק המשתמש

השכבה מכילה את חלון המשתמש והאפשרויות השונות לשימוש באפליקציה:



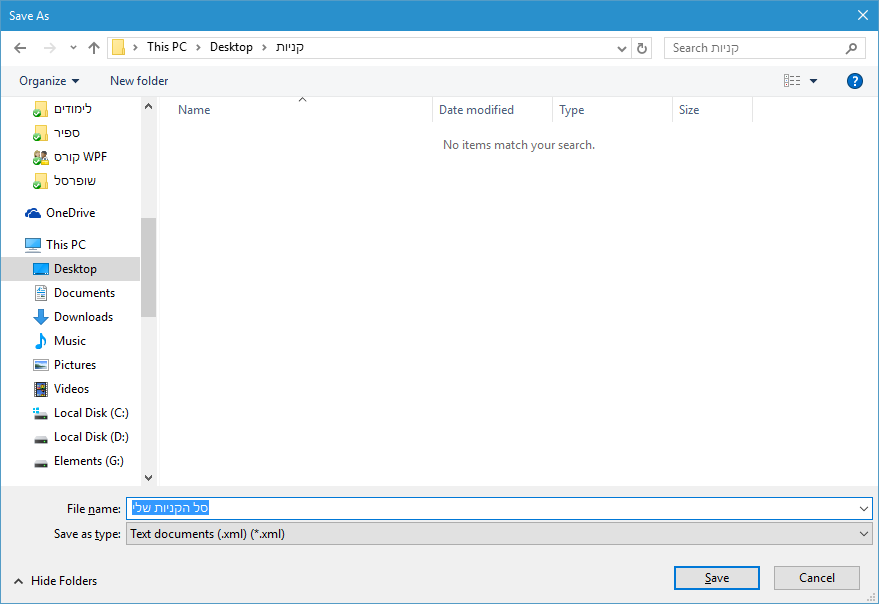


כאן ניתן לראות:

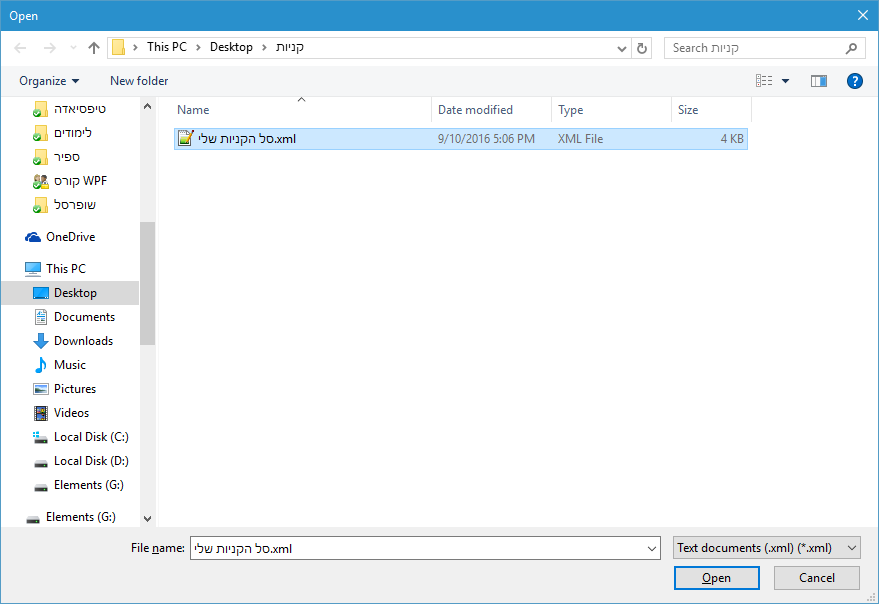
* עץ מוצרים היררכי המכיל את הקטגוריות השונות והמוצרים השונים אותם כוללת האפליקציה וניתן לבנות מהם סל קניות ולקבל את תוצאת המחירים עבור כל רשת. כדי לבחור מוצרים ניתן לסמן את המוצר בתיבת הבחירה (CheckBox) הנמצאת ליד כל שם מוצר, וכמות ע"י הכנסת הכמות הרצויה בתיבת הטקסט הצמודה לשם המוצר.
* ע"י לחיצה על כפתור השוואת המחירים  המערכת תעבד את בחירת המשתמש מעץ המוצרים הנ"ל ותציג את רשימת המוצרים ביחד עם המחיר והכמות בכל אחד מהטבלאות ואת מחיר סל הקניות בהתאם לרשת המתאימה



כפי שניתן לראות, עבור כל סל קניות יסומנו באדום שלושת המוצרים היקרים ביותר ובירוק את שלושת המוצרים הזולים ביותר מסל הקניות עבור כל רשת.



* לאחר בחירת סל המוצרים והחנויות בקטלוג, ניתן לשמור את סל הקניות ע"י לחיצה על לחצן שמירת סל הקניות  , ואז יפתח חלון חדש שיאפשר לשמור איפה שנרצה ובאיזה שם שנרצה את קובץ סל הקניות כדי שנוכל לשחזר אותו בהמשך.



* ע"י לחיצה על לחצן טעינת סל קניות יפתח חלון חדש שבו נוכל לבחור מחדש את הקובץ ששמרנו מבעוד מועד ולטעון את סל הקניות ותוצאתו תוצג כמו לאחר לחיצה על השוואת המחירים.

1. BusinessLogicLayer: שכבת לוגיקת העבודה

תפקיד שכבה זו הוא לקשר בין שכבת ממשק המשתמש (UI) לשכבת הגישה למשאבים (DataAccess) תוך כדי ביצוע מניפולציות על המידע המגיע מה DataAccess והפיכתו לאובייקט המוכר ל UI כך שיוכל לעבוד איתו ולמלא את המידע בצג.

* באתחול התוכניות שכבת ה UI יוצרת אובייקט מסוג LogicManager היושב בשכבת הלוגיקה ודרכו מבקשת את המידע הרלוונטי עבור כל חלקיה.

לאתחול התוכנית הLogicManager יוצר אובייקט מסוג ItemsCatalog (מחלקה הנמצאת בשכבת הלוגיקה) וממלא את האובייקט במידע אודות המוצרים הנכנסים לקטלוג המוצרים בשכבת ה UI ע"י בקשה משכבת ה DataAccess לשאוב את המידע מקובץ XML המתוחזק **ידנית** עבור איזה מוצרים נרצה בכלל להציג בקטלוג.

לאחר קבלת המידע משכבת ה DataAccess ה LogicManager ימלא את הפרטים החסרים באובייקטי ה ItemCatalogData (שמכירה שכבת ה UI )ויעביר את המידע לשכבת ה UI והיא בתורה תמלא את תצוגת קטלוג המוצרים.

* לאחר בחירת סל מוצרים, שכבת ה UI שולחת בקשה לשאר פרטי סל הקניות משכבת הלוגיקה וה LogicManager בתורו יקבל אובייקט מסוג List<ItemCatalogData>, יבצע מניפולציה עבור העברת האובייקט מ ItemCatalogData לאובייקט מסוג ItemGeneralData היושב בשכבת ה DataAccess, וישלח בקשה לשכבת התחתונה כדי שתמלא את הפרטים המלאים.
* עבור שמירת סל המוצרים המנהל יעשה את אותו תהליץ המרת האובייקט מ ItemCatalogData ל ItemGeneralData ובמקום לשלוח את הבקשה למילוי הפרטים מיד ע"י שכבת ה Data הוא ישלח בקשה לשמור את פרטים אלה למילוי במועד מאוחר יותר שבו נבחר לטעון את סל הקניות.

1. DataAccessLayer: שכבת הגישה למשאבים

שכבה זו בתורה, אחראית לחבר בין בקשות מנהל התוכנית למשאבים היושבים על הדיסק בצורת קבצי XML.

* שכבה זו מחזיקה במחלקות ItemGeneralData ו ItemFullData ומשתמשת בסוגים אלה כדי להעביר את המידע הרלוונטי בכל שלב עבור המוצרים.
* בשלב הראשוני נטען רק ה ItemGeneralData המכיל את שם המוצר, ID שקיבל בעת יצירת קובץ ה XML הידני ואת ברקוד הגישה לשלושת הרשתות השונות כדי לשאוב מהן את המידע הנוסף בעת בחירת המוצרים.
* לאחר קבלת סל קניות ממנהת התוכנית, המיוצגת באמצעות רשימה של אובייקטים מסוג ItemGeneralData , השכבה תיגש לקבצי המידע של שלושת הרשתות ותמלא את הפרטים המלאים של כל מוצר ותחזר רשימה של ItemFullData שאותו יוכל המנהל לקבל ולהתאים להמשך תפקוד המערכת.

הפרדת השכבות וביצוע המניפולציה ע"י שכבת הלוגיקה בין המידע שנכנס אליה למידע היוצא לשכבת ה UI מאפשר הפרדה של הטיפוסים המוכרים בין השכבות הקיצוניות ובמקרה של שינוי במידע הנכנס לא תהיה השפעה ביותר מהתאמה בשכבת הלוגיקה ודרך פעולת השכבה העלוינה לא תשתנה, ובהתאם גם שינויים מטיפוסים בשכבת ה UI לא ישפיעו על שכבת ה Data.

* כמו שניתן לראות, הבונוסים שמומשו הם יישום UI גרפי ושמירת סל הקניות עבור טעינה במועד מאוחר יותר.
* בנוסף, הפרוייקט מעוצב בצורת הפרדה ל 3 שכבות, מה שמאפשר ליישם את התוכנית בצורת Client-Server כאשר ב Client תשב שכבת ה UI וב Server ישבו השכבות האחרות (למרות שזה לא מיושם כרגע, המעבר דורש רק את שינוי העברת המידע בין ה UI לשכבת הלוגיקה).