**תיאור קצר של הפיצ'רים שבחרנו לממש בתרגיל הקודם:**

* Post Filtering - Enables prioretizing and filtering new feed posts
* Canned Posts - Allows creating posts from templaces

**שימוש ב-Threadים**

לאחר ההמרה ל-DataBinding צמצמנו את כל הקוד שמדבר ישירות עם FB לכדי שורות מעטות. לכן רק היה צורך להפריד ל-Thread נפרד בעת החיבור ל-Data Source.

מקום בקוד: FormMain::buttonLogIn\_Click()

**שימוש ב-Data Binding**

מקום בקוד: FormMain, FormEditFilterGroup

**תבנית מס' 1 – Factory Method**

* סיבת הבחירה / שימוש בתבנית:

קיימת משפחה פולימורפית (כל ממשי ה-(IPostFilter של סוגי פילטרים. בממשק רצינו לאפשר למשתמש לתת את סוג הפילטר ומידע ומזה ליצור אוביקט קונקרטי מתאים של פילטר.

* אופן המימוש:

הגדרנו את PostFilterFactory שהוא בעצם אחרי על בניית האוביקטים. יצרנו גם Attribute הנקרא RegisterAttribute שהוא מוגדר בתוך PostFilterFactory זה מאפשר למשתמש להגדיר איזה סוג פילטר מגדירה הclass-. ב-constructor הסטטי של PostFilterFactory אני משתמשים ב-reflection כדי למצוא את כל המחלקות שמשתמשות ב-attribute לעיל. מחלקות אלו הן אלו שה-PostFilterFactory ייצר.

בקוד:

Facebooky/PostFilterFactory.cs

Facebooky/IPostFilter.cs

Facebooky/RegexPostFilter.cs

Facebooky/WordPostFilter.cs

Facebooky/PostFilterGroup.cs

* Sequence Diagram



* Class Diagram



**תבנית מס' 2 – Composite**

* סיבת הבחירה / שימוש בתבנית:

שמנו לב שלפעמים נוח להגיד בפילטרים קבוצה בתוך קבוצה. בעזרת שימוש ב-Composite כל השימוש ב-Filter וכן פעולות הסריאליזציה והדסריאליזציה יכולות להעשות באופן פולימורפי ולהקל על כתיבת הקוד ומרחיבה את היכולות של אירגון הפילטרים.

* אופן המימוש:

ממישנו זאת בכך שממימשנו את הממשק IPostFilter גם ב-PostFilterGroup ובכך הפכנו אות הקבוצה והפריט להיות פולימורפים.

בקוד:

Facebooky/PostFilterFactory.cs

Facebooky/IPostFilter.cs

Facebooky/RegexPostFilter.cs

Facebooky/WordPostFilter.cs

Facebooky/PostFilterGroup.cs

* Sequence Diagram



* Class Diagram



**תבנית מס' 3 – Composite**

* סיבת הבחירה / שימוש בתבנית:

לאחר שהמרנו את הקוד להשתמש ב-Data Binding לא יכלנו יותר לשלוט בתוכן הרשימה של ה-New Feed זאת מכיוון שהיא קיבלה את התוכן שלה מה-Data Source וזה לא מכיר את המנגון שלנו. זה בעצם גרם לכך שאחד הפיצ'רים שלנו פשוט לא עבד.

* אופן המימוש:

לכן ממישנו Proxy בין ה-Data Source ל-List Box שמציג את התוכן ב-Data Source לאחר פילטור ומיון של המנגנון שלנו. כך אנו יכולים להציג מפולטרת אבל עדיין להשתמש באותו Data Source.

בקוד:

Facebooky/ProxyDataSource.cs

Facebooky/FormMain.cs

* Sequence Diagram



* Class Diagram

