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

* הפיצ'ר הראשון FindOptionalForRelationship מוצא על נתוני החברים שלי בפייסבוק את הרשימה של האנשים האופציונליים להתכבתבות ברגע זה אם עברו את הבדיקה לפי הנתונים שלהם ויצאו שהם אופציונליים לקשר כלשהו בשבילי.
* ברגע שנלחץ על הכפתור Google Selected Friend הדפדפן של האינטרנט יבצע בדיקה בגוגל על שם החבר המסומן ברשימת החברים

### תבנית מס' 1 – Observer Pattern

* סיבת הבחירה / שימוש בתבנית:

בחרנו בתבנית זו כדי להחליף את התפקידים במערכת שלנו לצורך שימוש עתידי,

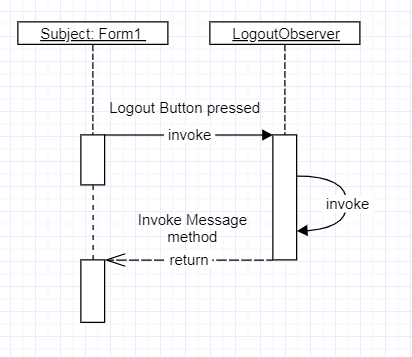
במקום שהקוד יהיה כתוב במתודה של לחיצת כפתור הLogout ואז בעתיד במידה ונצטרך להוסיף פיצ'רים חדשים – נצטרך לשנות את הקוד.

רצינו שכל מי שירצה להשתמש בשירותיו של כפתור זה פשוט יצטרך להירשם כמאזין שלו והוא יעביר את המידע ויפעיל אותם ברגע שילחץ.

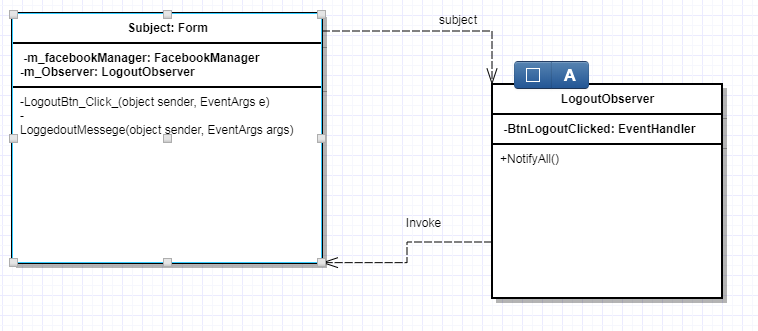
* אופן המימוש:

ממישנו את תבנית זו באמצעות הכלים שנתנו לנו ב C# 3 באמצעות הDELEGATE, כאשר במחלקת הFORM רשמנו את מי שרוצה להאזין לכפתור הLogout וברגע שלוחצים עליו הוא מודיע למאזין Class LogoutObserver והוא מודיע לכל מי שרוצה כרגע או ירצה בעתיד להשתמש במידע שכפתור הLogout נלחץ.

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Strategy Pattern

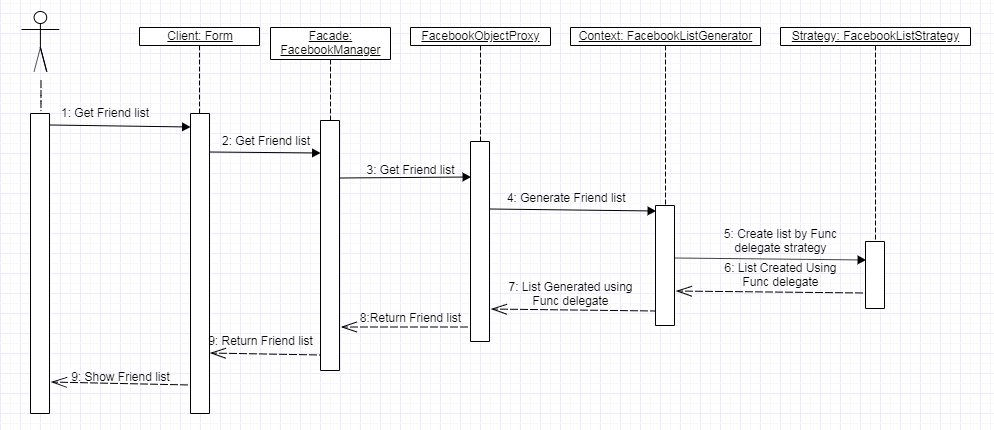
* סיבת הבחירה / שימוש בתבנית:

בתכנית שלנו היינו צריכים להשתמש ברשימות שונות, שלכל אחת מהן יש שימוש במדותה שונה להשגת הרשימה, לכן השתמשנו ב Strategy Pattern כדי שנוכל לבצע החלפה של השימוש במתודות באופן פשוט וריוסבילי(reusable).

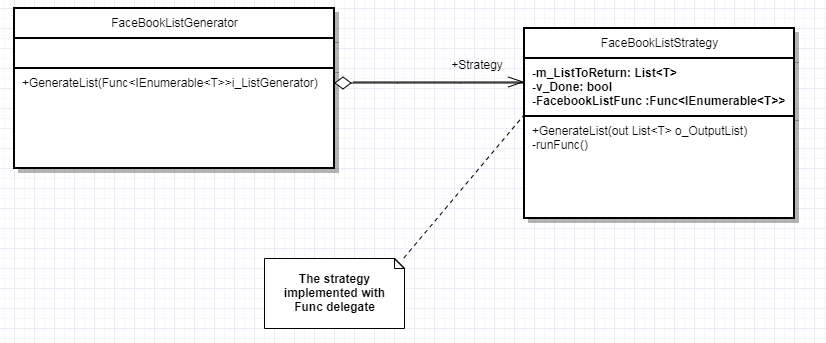
* אופן המימוש:

בתכנית שלנו השתשמנו במחלקת FacebookListGenerator שמשמשת כContext בתבנית הStrategy ומחלקה זו משתמשת שירותיה של מחלקת הFacebookListStrategy שמחליפה באופן יזום את ה"אלגוריתמים" או במקרה שלנו את המתודות השונות שיש להשתמש בהן כדי לקבל את הרשימה הנכונה.

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Iterator Pattern

* סיבת הבחירה / שימוש בתבנית:

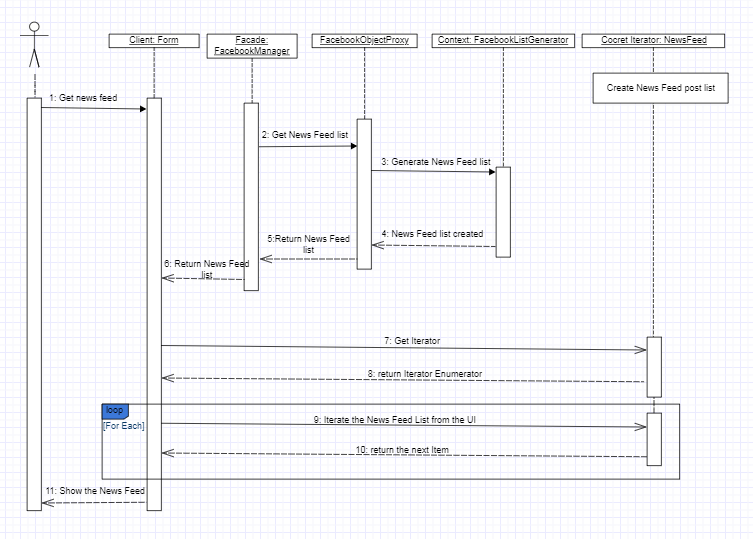
הסיבה שבגינה השתמשנו בתבנית זו היא בשביל להעביר את היכולת לClient לבצע חיפוש על רשימת הNews Feed מבלי לדעת כלום על מבנה הנתונים ועל המימוש שלה,

כדי שבעתיד תהיה לנו היכולת לבצע שינויים במחלקת הNews Feed ללא שינוי הקוד והתכנית עדיין תהיה תקינה.

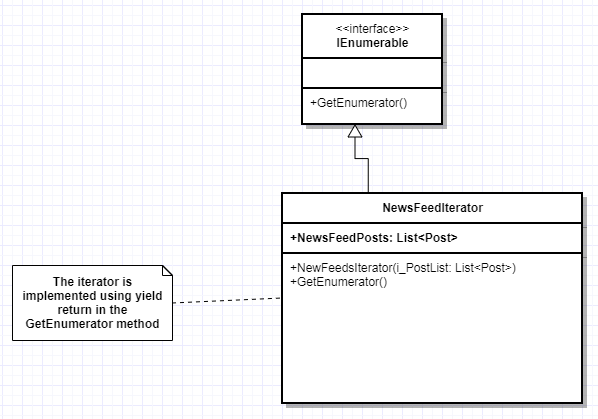
* אופן המימוש:

יצרנו מחלקה בשם NewFeedsIterator שממשת את IEnumerable interface בעזרת מתודה GetEnumerator ושימוש ב yield return מה שנותן את האפשרות מחוץ למחלקה לבצע סריקה על רשימה זו באמצעות שימוש ב foreach (לדוגמה), מבלי לדעת על מבנה הנתונים שבה ממושת הרשימה, כך שבעתיד במידה ונרצה לשנות את מבנה הנתונים הנ"ל, נעשה זאת מבלי לשנות שום דבר בקוד הקיים מחוץ למחלקה זו.

* Sequence Diagram



* Class Diagram



### Full application class diagram

