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

* הפיצ'ר הראשון הוא קבלת רשימה של פעילות חברים בפרופיל של המשתמש.
* הפיצ'ר השני הוא חיתוך של Checkins לפי המשתמש ו/או החברים שלו.
* הפיצ'ר השלישי הוא משחק של שמות.

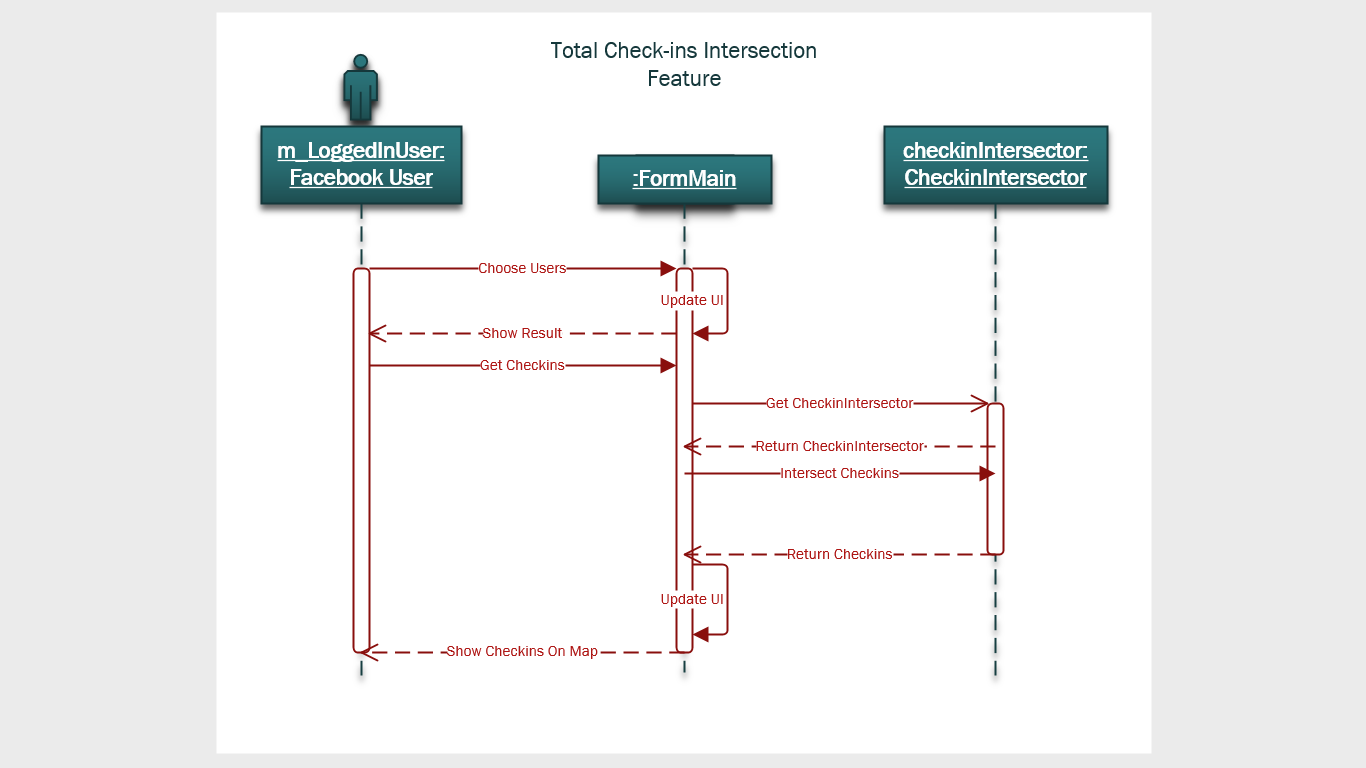
### תבנית מס' 1 – Iterator

* במחלקה CheckinIntersecctor המתודה IntersectCheckins מחזירה אוסף של Checkins אשר משותפים לכמה משתמשים. זהו סוג של הטלה שאנו עושים על אוסף של Checkins, ובנוסף, כפי שהמתודה בנויה עד מטלה 2 המתודה קודם אספה את כל הCheckins ואז החזירה את האוסף. צריך לחשוב על מצב שבו אם למשל הClient ירצה לקבל חלק מהCheckins ואז לעצור. לכן השתמשתי בתבנית Iterator.

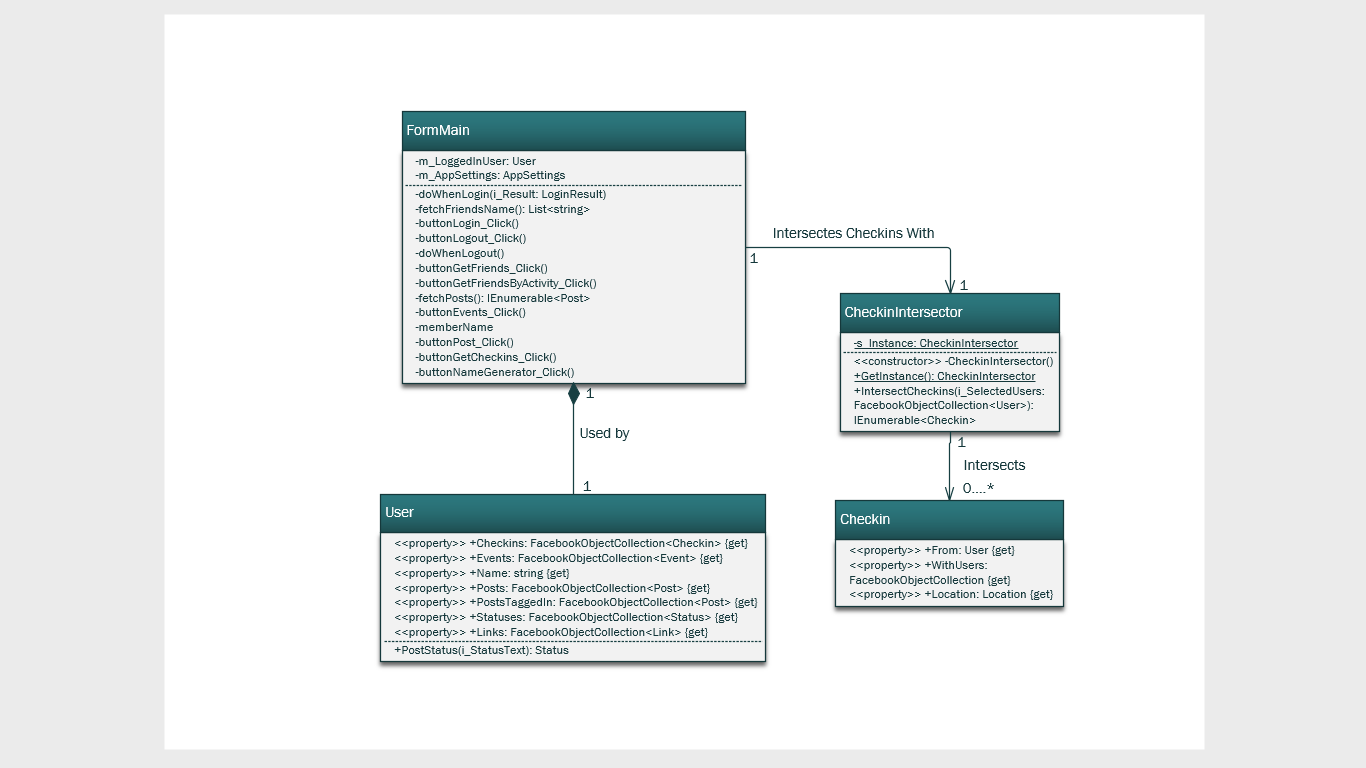
* אופן המימוש:

במחלקה CheckinIntersecctor במתודה IntersectCheckins, בכל מקום שהיה הוספה לאוסף sharedCheckins החלפתי בYield return.

* Sequence Diagram



* Class Diagram



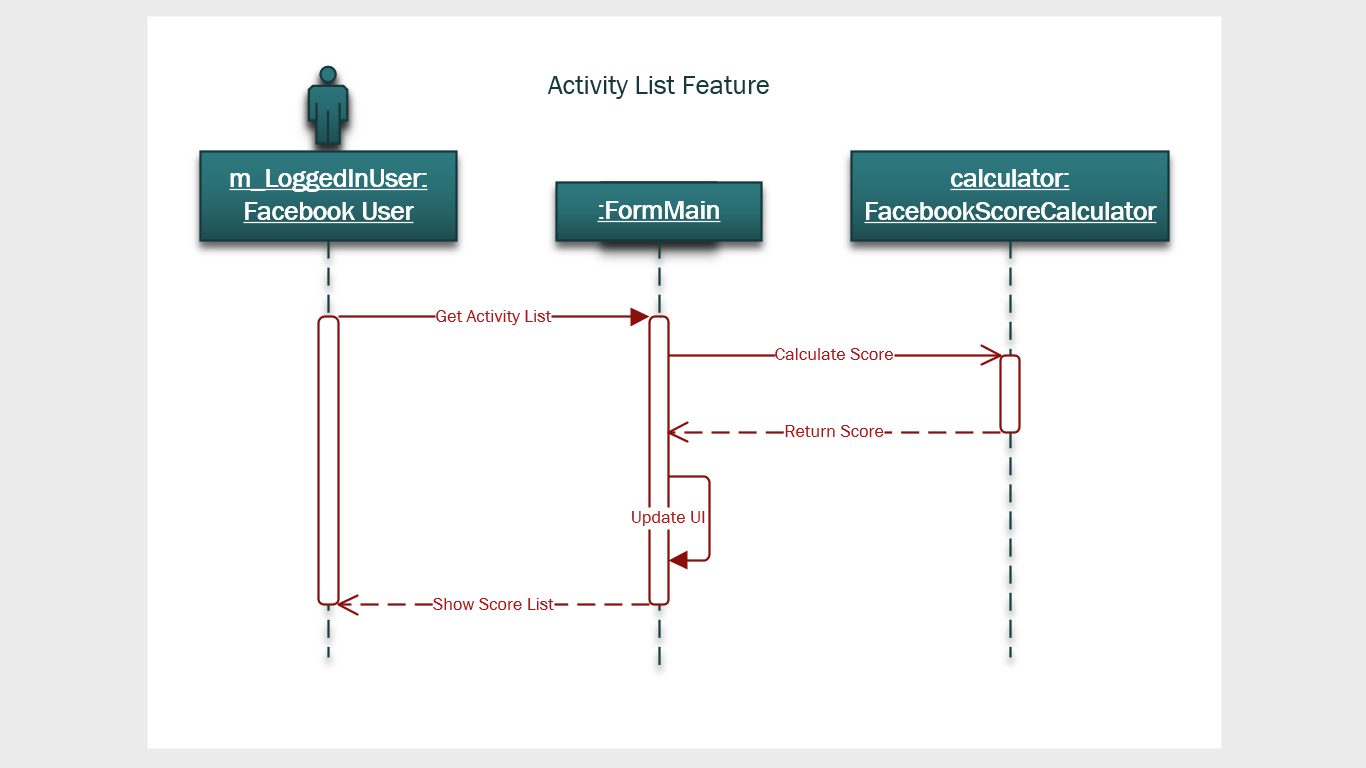
### תבנית מס' 2 – Strategy

* בפיצ'ר רשימת פעילות משתמשים, לפני שאני מציג את התוצאות למשתמש אני קודם ממיין את התוצאות בסדר יורד על פי רמת פעילות באחוזים. וזאת בעזרת מחלקה שמממשת ממשק IComparer. הבנתי שהמחלקה ספציפית מדי עבור המיון הזה, שכן אם ארצה למיין בדרך שונה עלי ליצור עוד מחלקה דומה (לא Reusabiliy). לכן השתמשתי בStrategy.

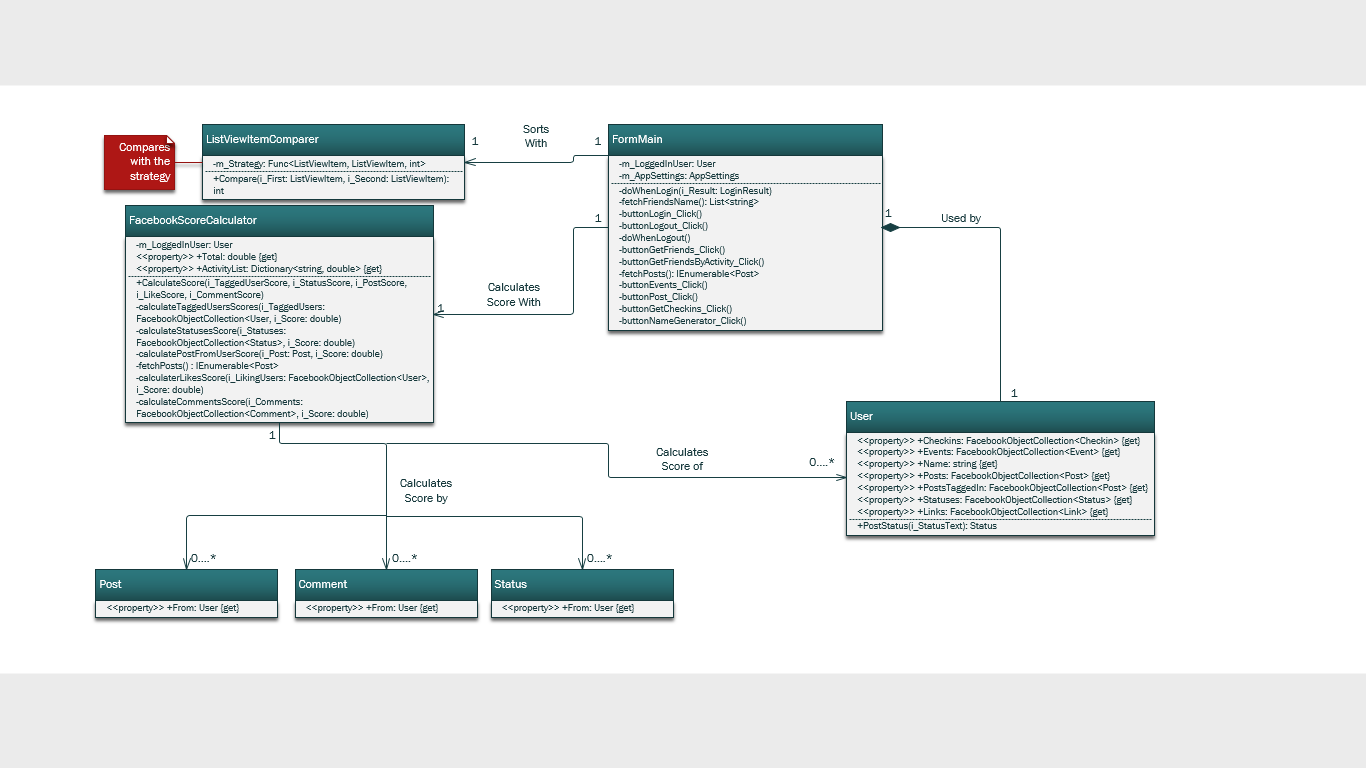
* אופן המימוש:

במחלקה ListViewItemComparerDescending שכרגע נקראת ListViewItemComparer מקבל בConstructor שלו Func<string, string, int> מכיוון שהמתודה Compare צריכה להחזיר int. בתוך המתודה Compare אני מפעיל את הDelegate על שני אובייקטים של ListViewItem.

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Command

* בפיצ'ר של NameGeneration הבחנתי שיש סוג של תפריט בטופס FormNameGames והכפתורים עושים סוג של פעולה שהיא די דומה. כולם מציגים string בתוך TextBox.

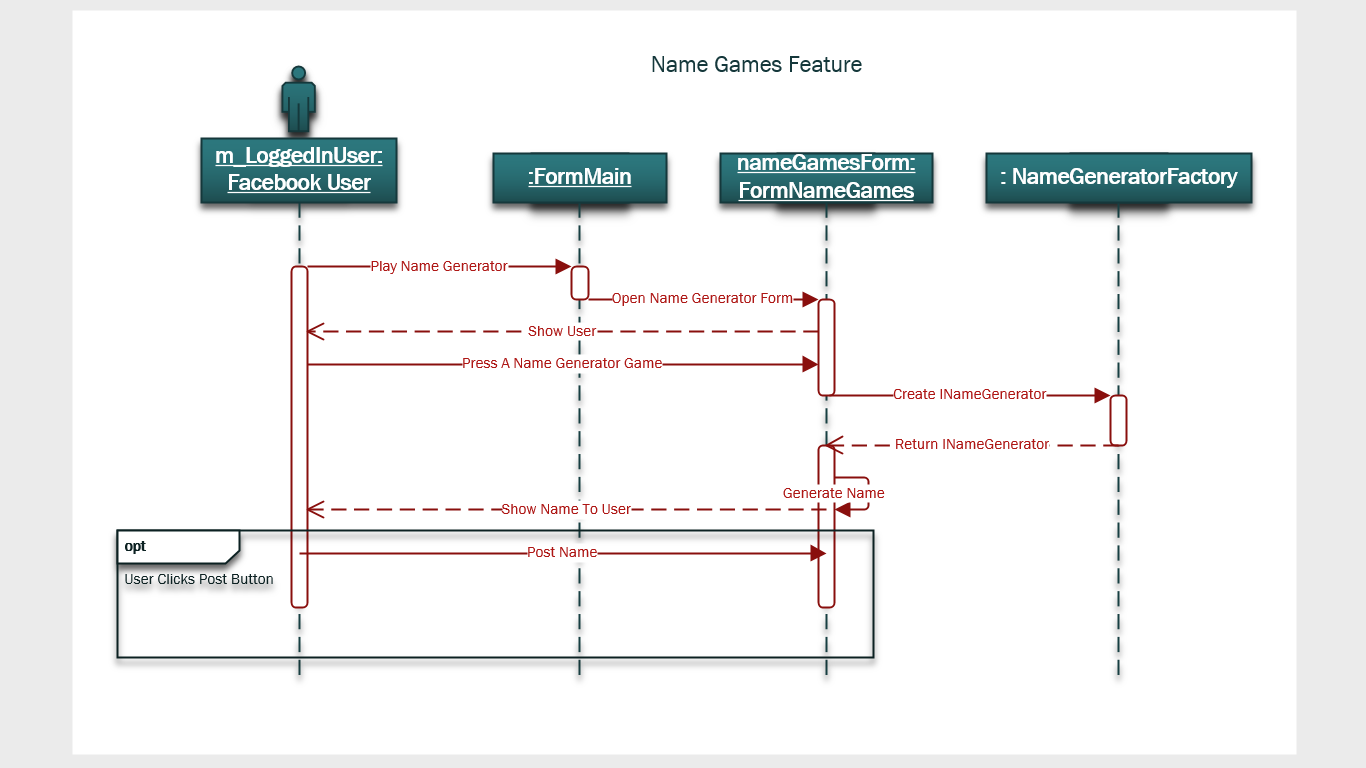
לכן חשבתי על כך שהכפתור הוא הUIItem והמתודות שמג'נרטות שמות הן Commands.

* אופן המימוש:

במקום מחלקות שמממשות את הממשק INameGenerator, יש מחלקה אחת שמממשת INameGenerator והיא נקראת NameGeneratorByFullName. הConstructor מקבל 2 string שהם השם המלא של המשתמש וFunc<string, string, string>. המחלקה הזאת קוראת לDelegate בפונקציה GenerateName. יש לשים לב שאין שבירה של תבנית Factory Method, מכיוון שלמרות שיש רק מחלקה אחת שמממשת INameGenerator, עדיין אפשר ליצור עוד מחלקות שימממשו את הממשק. השוני ביניהם יהיה בסוג הFunc.

בנוסף התבנית Factory Method השתדרגה מכיוון שכרגע אין switch case במתודה Create של הFactory. אם למשל ארצה ליצור INameGenerator עם חתימה שונה אז אצור עוד מתודה Create עם חתימה שונה וכך קיבלנו פולימורפיזם בלי switch case.

* Sequence Diagram



* Class Diagram

