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

* Tagged Places – מפה שעל גביה מוצגים מיקומים של המשתמש על פי סוג המידע המחובר למיקום.
* Top Liked Photo Feature – פרמידה של 6 תמונות שזכו למירב ה"לייקים" של המשתמש, ורשימת 5 החברים המפרגנים ביותר ב"לייקים".

### תבנית מס' 1 – Factory Method

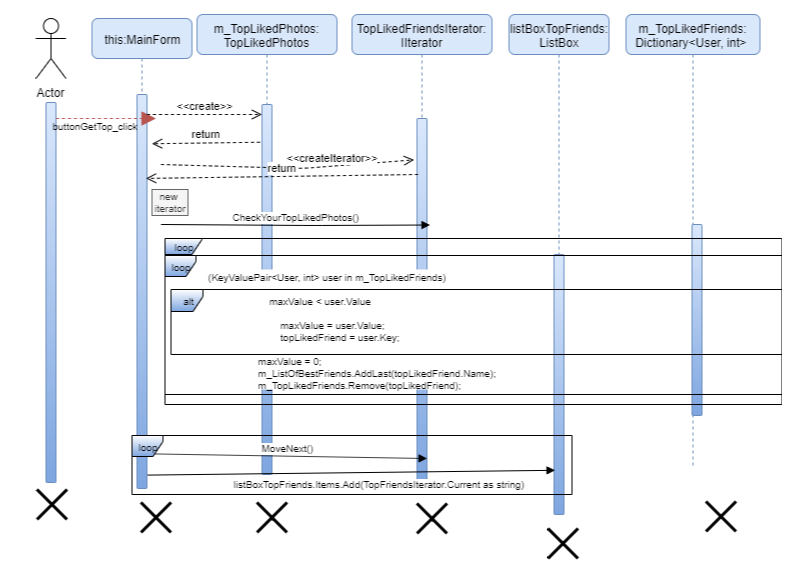
### תבנית מס' 2 – Iterator

* סיבת הבחירה / שימוש בתבנית:

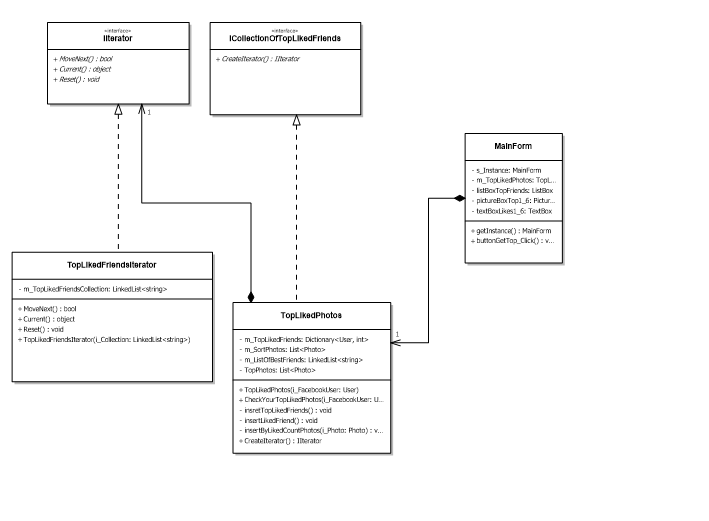
בחרנו להשתמש בתבנית זו במימוש הפיצ׳ר Top Liked Photos על מנת לספק ממשק למעבר על אוסף פריטים (החברים "המפרגנים ביותר" של משתמש) אשר נמצא באובייקט שמאגד אותו (המחלקה שמממשת את הפיצ׳ר TopLikedPhotos), ללא תלות במבנה הנתונים שמשמש לשמירתם.

בעזרת מימוש תבנית זו, אם בעתיד נרצה להחליף את מבנה הנתונים שבו נשמרת רשימת החברים הטובים, נוכל לעשות זאת על ידי שינוי מבנה הנתונים בלבד וללא שום שינוי נוסף בקוד שנכתב קודם לכן.

* אופן המימוש:
  + בנינו ממשק IIterator בעל שלוש חתימות:
    - bool MoveNext() – האם ניתן להתקדם במעבר על מבנה הנתונים, תוך התקדמות על מבנה הנתונים.
    - object Current { get; } – מחזיר את הפריט הנוכחי תוך כדי המעבר על מבנה הנתונים.
    - Void Reset() – איפוס המעבר על מבנה הנתונים.
  + בנינו את המחלקה TopLikedFriendsIterator שמממשת את הממשק IIterator ומכילה Data Member שמשמש כמבנה הנתונים שמכיל את רשימת החברים המפרגנים ביותר במסגרת הפיצ׳ר.
  + בנינו ממשק ICollectionOfTopLikedPhotos בעל חתימה של מתודה אחת IIterator CreateIterator().
  + המחלקה TopLikedPhotos שמממשת את הפיצ׳ר, מממשת את הממשק ICollectionOfTopLikedPhotos, כלומר מאתחלת מופע חדש של האובייקט TopLikedFriendsIterator אשר מאפשר לבצע מעבר על מבנה הנתונים שמכיל את רשימת החברים המפרגנים של המשתמש (שקשורה לפיצ׳ר (TopLikedPhotos , ללא תלות במבנה הנתונים שנבחר.
* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Observer