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

* **Random friend liked pages:** From a list of friends, who are also using this application, pick randomly one friend and see 10 pages that this friend likes. The pages are shown as a grid of pictures 2 \* 5.
* **Find friends by city:** This feature helps the user to see in his list of a friends, who are also using this application, only the friends in a specific location (provided by the user).

### תבנית מס' 1 – Caching Proxy

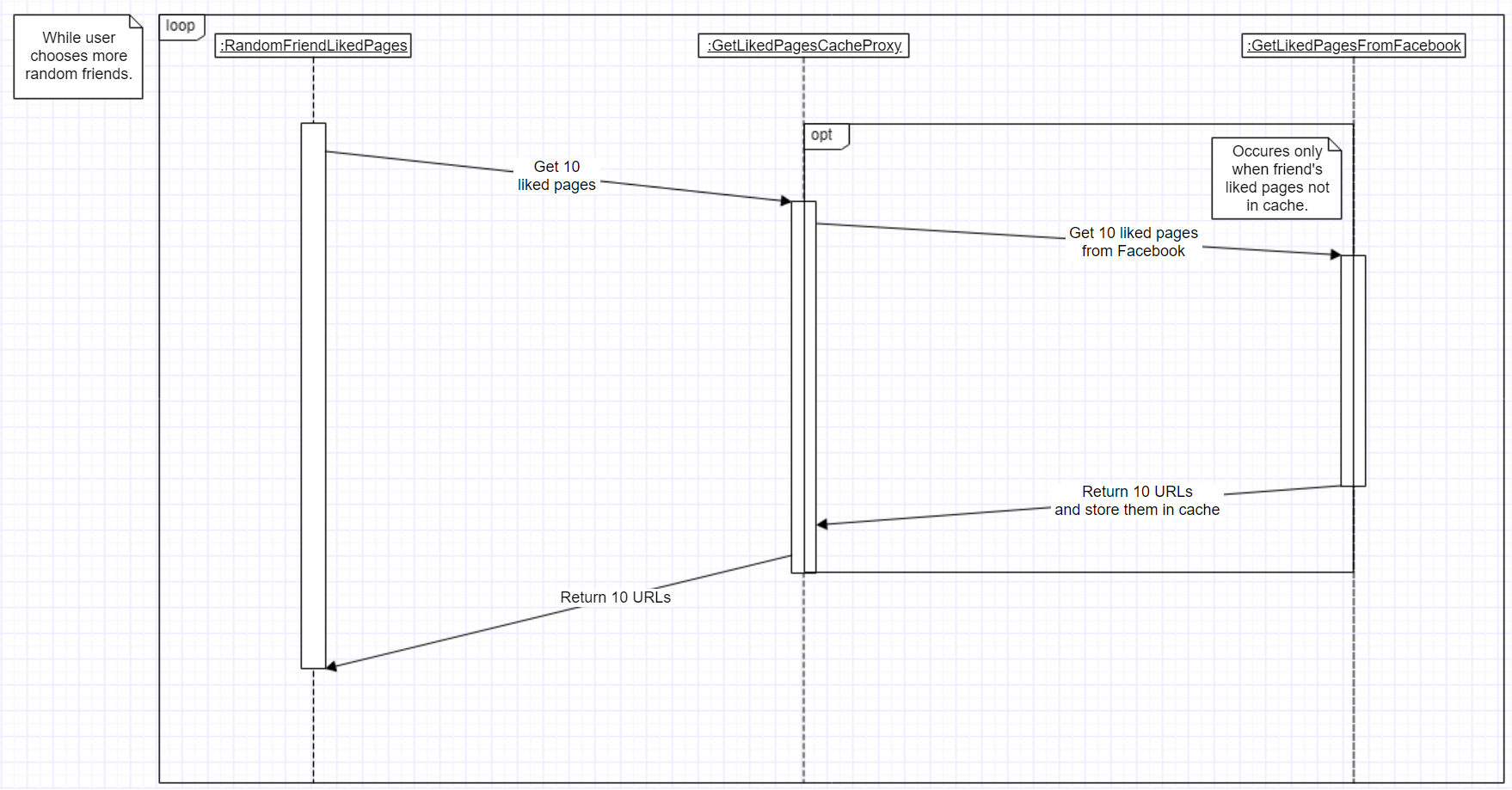
* סיבת הבחירה / שימוש בתבנית:

We’ve decided to implement Caching Proxy pattern for RandomFriendLikedPages feature.  
When we get our randomized friend info about his liked pages (picture of each page), we would like to store that data for possible next time (if user gets same random friend in the same session). That gives us the option to show same pictures even if user had lost connection to server (only the pictures that are stored in cache) and it gives faster results (time efficiency) to user which improve user experience.

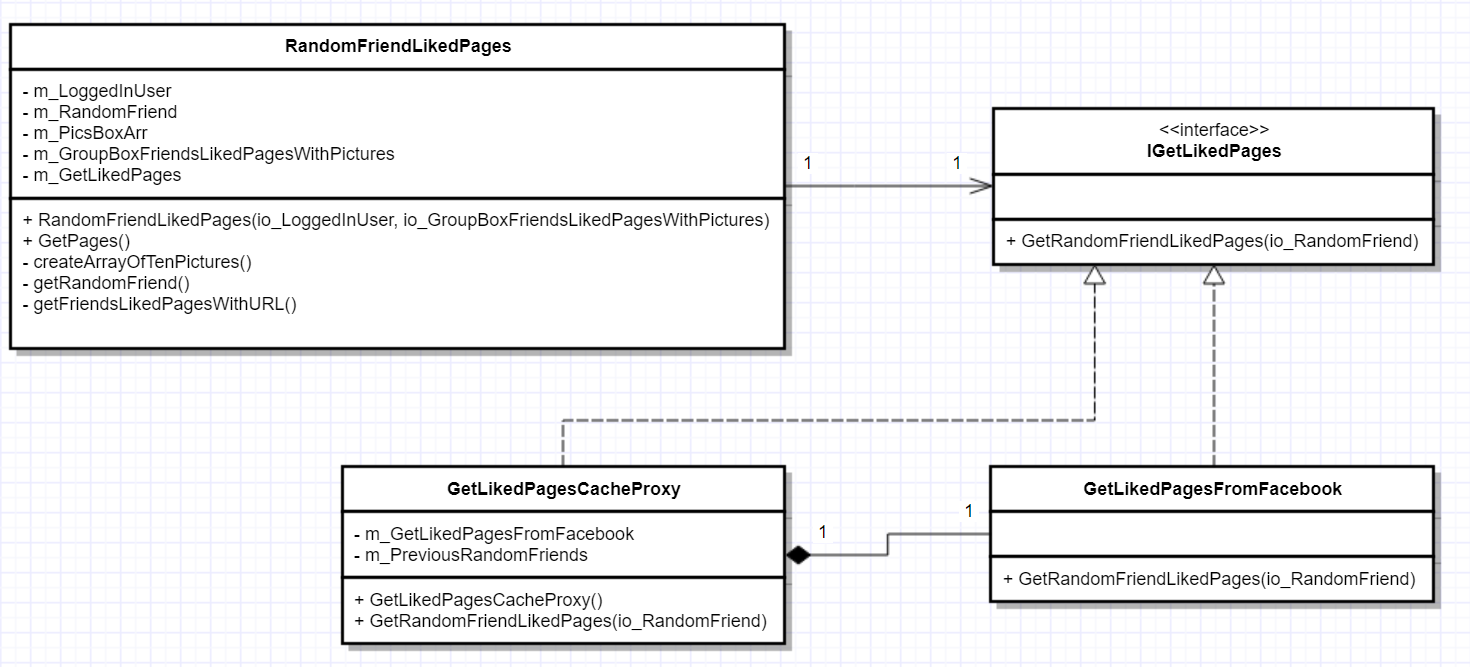
* אופן המימוש:

**Relevant classes:** IGetLikedPages, GetLikedPagesFromFacebook, GetLikedPagesCacheProxy, RandomFriendLikedPages.

* Sequence Diagram



* Class Diagram

**  
Client:** RandomFriendLikedPages.  
**Subject:** IGetLikedPages.  
**Real Subject:** GetLikedPagesFromFacebook.  
**Proxy:** GetLikedPagesCacheProxy.

### תבנית מס' 2 – Factory Method

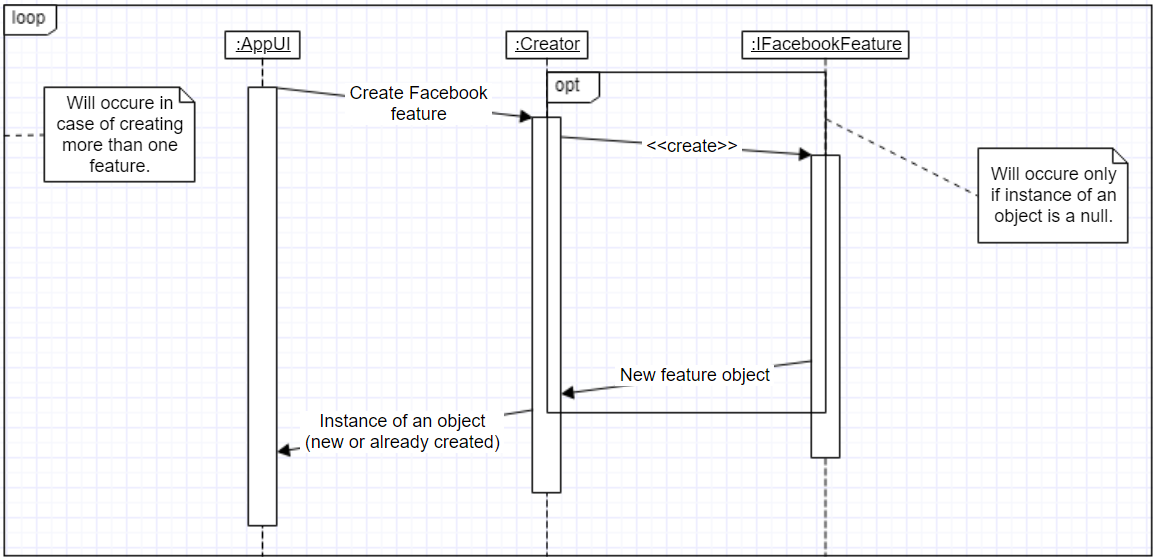
* סיבת הבחירה / שימוש בתבנית:

We’ve decided to implement Factory Method pattern in case of additional future features.  
Instead of writing the whole code over again in case of new feature, we can just implement IFacebookFeature and add new functions. Every concrete creator contains a singleton pattern because we don’t want to create multiple instances of same objects. Also, FriendsByRequest feature is a form (hence it is inheriting from class Form), so we’ve created a wrapper (FriendsByRequestWrapper) that allows us to hold this feature as property and in the same time implement IFacebookFeature interface.

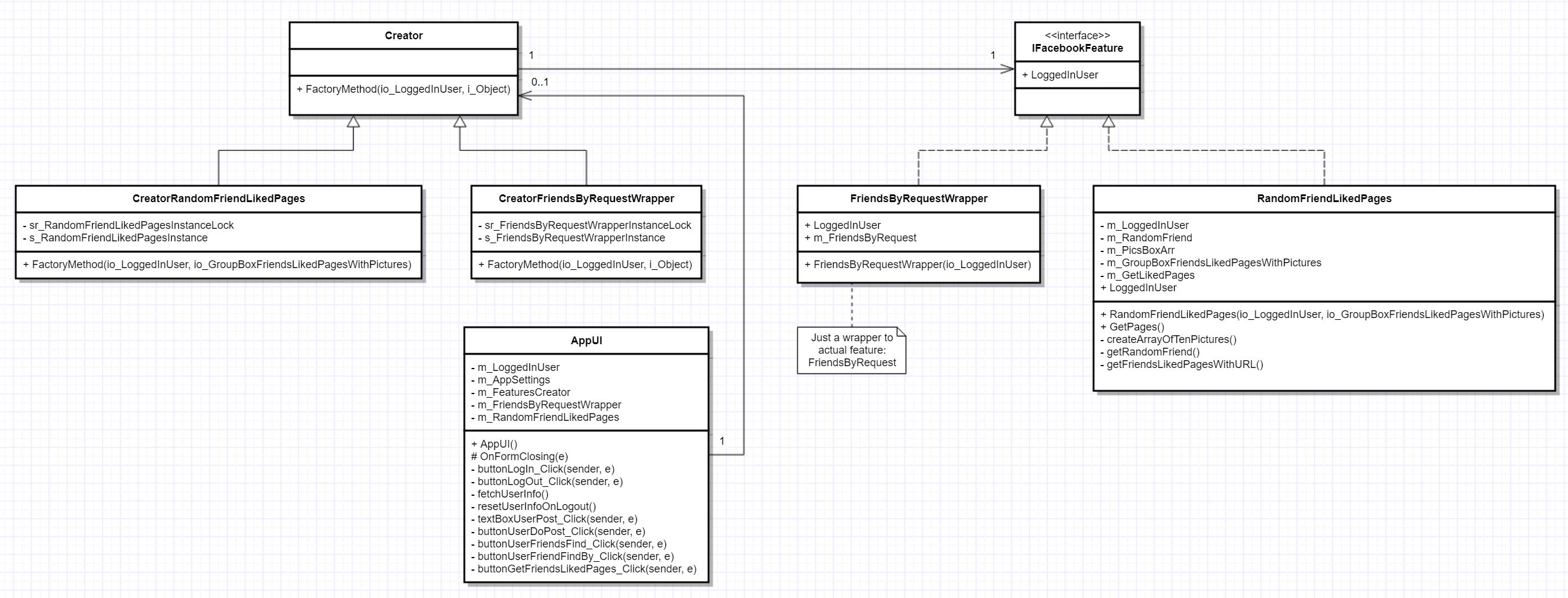
* אופן המימוש:

**Relevant classes:** IFacebookFeature, RandomFriendLikedPages, FriendsByRequestWrapper, Creator, CreatorRandomFriendLikedPages, CreatorFriendsByRequestWrapper, AppUI.

* Sequence Diagram



* Class Diagram

  
  
**Product:** IFacebookFeature.  
**Concrete Products:** RandomFriendLikedPages, FriendsByRequestWrapper.  
**Creator:** Creator.  
**Concrete Creators:** CreatorRandomFriendLikedPages, CreatorFriendsByRequestWrapper.  
**Client:** AppUI.

### תבנית מס' 3 – [שם התבנית]

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם