**Desktop Facebook application**

**Basic user experience -** הדף הראשי של האפליקציה מספק למשתמש חווית פייסבוק בסיסית. האפליקציה מציגה את תמונת הפרופיל ותמונת הנושא של המשתמש שהתחבר לאפליקציה. ניתן לראות מידע בסיסי אודות המשתמש: שם מלא, תאריך לידה, כתובת אימייל, עיר מגורים, מין, סטטוס מערכת יחסים, מקום בו הוא לומד ומקום בו הוא עובד.  
בנוסף קיימים 2 תאים נוספים המציגים את רשימת החברים של המשתמש (עם אופציה לחיפוש חבר ברשימה) ואת הפוסטים שפרסם בקיר שלו.  
בדף הראשי גם קיימת למשתמש אופציה לפרסם פוסט חדש על הקיר שלו (במידה והפרסום לא מצליח תופיע הודעה מתאימה עבור המשתמש). במידה והמשתמש מעוניין להתנתק מהאפליקציה, קיים כפתור התנתקות במסך הראשי.

בנוסף מימשנו בדף הראשי פיצ'ר נוסף - בעת לחיצה על הכפתור Fetch Albums יפתח חלון נוסף עם רשימת האלבומים של המשתמש. בעת לחיצה על שם אלבום מסוים, יופיעו על החלון פרטים על האלבום והתמונות המופיעות בו, עם אופציה למיון התמונות לפי קטגוריות שהוגדרו מראש.

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

* **פיצ'ר ראשון – חיפוש זכרונות בפייסבוק לפי תאריך מסוים/טווח תאריכים ותוכן (פוסטים,תמונות, אירועים, קבוצות, חברויות שנוצרו).**פיצ'ר זה מאפשר למשתמש אופציה לראות את הזכרונות שלו בפייסבוק בדומה לפיצ'ר שכבר קיים, אך השדרוג הוא שכאן המשתמש יכול לבחור בתאריך ספציפי או בטווח תאריכים (בחירה לפי יום, חודש, שנה), ובנוסף הוא יכול לסנן לפי התוכן שהוא מעוניין לראות (תמונות שהעלה בתאריכים האלה, פוסטים שפרסם, אירועים שהשתתף בהם, קבוצות שהצטרף אליהם).   
  למשתמש יש אופציה לראות רק אחת מהאופציות, לבחור כמה מהן או לראות את כולן.

בנוסף, כאשר המשתמש בוחר לראות תמונות, קיימת אפשרות בלחיצה כפולה על אחת התמונות לפתוח את התמונה לצפייה בחלון גדול יותר.

בכל רגע נתון, קיימת האופציה לחזור לחלון הראשי של האפליקציה ע"י לחיצה על כפתור החזרה.  
  
מיקום בקוד: המחלקה MemoriesSearch ללוגיקה והטופס FormMemories.

* **פיצ'ר שני – שאלון חברות – עד כמה אתה מכיר את חבר שלך?**פיצ'ר זה מאפשר למשתמש לבצע שאלון שהוגדר מראש על אחד מחבריו בפייסבוק (קיימת אופציה לבחור חבר ספציפי או לתת לאפליקציה לבחור חבר בצורה רנדומלית).  
  לאחר בחירת החבר יופיעו על גבי המסך תמונת הפרופיל שלו ו – 6 שאלות הקשורות לנתונים שהחבר שיתף בפייסבוק (לדוגמא: מתי פרסם פוסט בפעם האחרונה?).  
  במידה ואחד הנתונים המתקשרים לתוכן השאלות לא הוגדר ע"י המשתמש/שליפת הנתונים לא מתאפשרת על ידי Graph API השאלה לא תוצג, והנקודה תינתן באופן אוטומטי למשתמש. בסיום השאלון, עם לחיצה על כפתור ה – Submit האפליקציה מחשבת את הניקוד (כל שאלה שנענתה נכון שווה נקודה), ומציגה אותו על המסך בצירוף תמונת הפרופיל של החבר.  
  המשתמש יכול לבחור באופציה לפרסם את השתתפותו בשאלון והניקוד שקיבל על גבי הקיר של חברו.

לאחר מכן, המשתמש יכול לחזור למסך השאלון ולשחק פעם נוספת (או לחלופין לחזור ממסך השאלון למסך הראשי).  
  
מיקום בקוד – המחלקות FriendShipTester, QuestionInitializer ללוגיקה   
והטפסים: FriendshipTestQuestions,FormFriendshipTest .

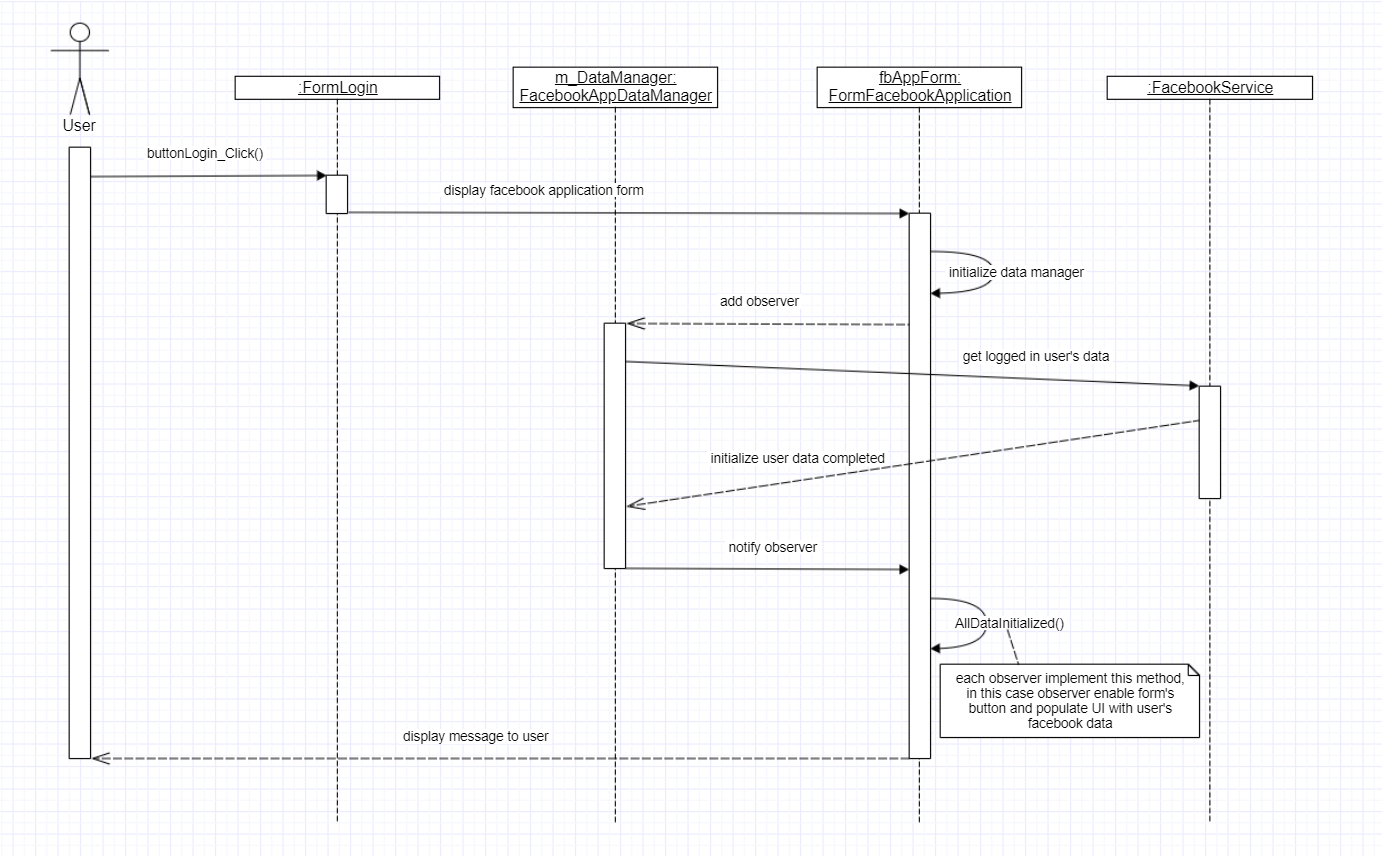
### תבנית מס' 1 – Obserber

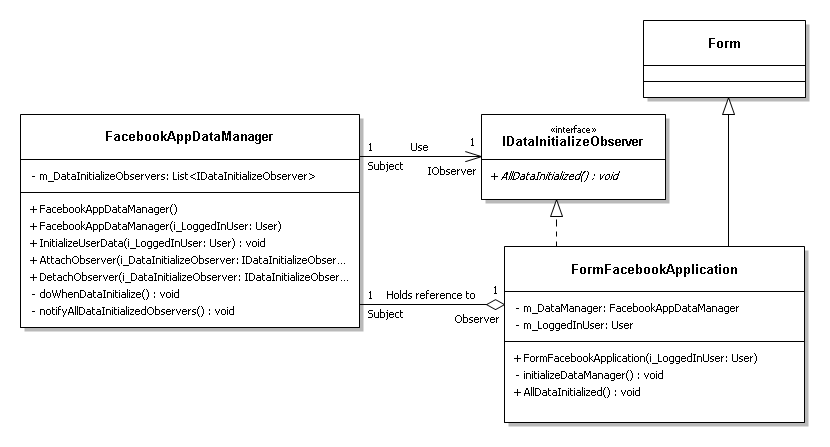
* סיבת הבחירה / שימוש בתבנית:

בעת העלאת החלון הראשי של האפליקציה, בחרנו לבצע את טעינת המידע לחלון (פרטי המשתמש, פוסטים, רשימת חברים ועוד) בצורה אסינכרונית. בזמן טעינת המידע המשתמש יכול לבצע פעולות ברקע ובנוסף, אין באפשרותו ללחוץ על הפקדים (כפתורי הפיצ'רים, כפתור פרסום הפוסט וכפתור ההתנתקות) עד שהמידע נטען והמשתמש מחובר למערכת. לכן, יש צורך להודיע למשתמש ברגע שהמידע מסיים להיטען, ובנוסף לשנות את מצב הפקדים לEnable על מנת שהמשתמש יוכל להתחיל שימוש באפליקציה ובפיצ'רים.   
בשביל לממש זאת בחרנו בתבנית observer.

* אופן המימוש:

יצרנו ממשק IDataInitializeObserver המגדיר את המתודה AllDataInitialize אותה כל מחלקה המעוניינת להאזין צריכה לממש. המחלקה FacebookAppDataManager האחראית לאתחול הנתונים ברשימות עבור המשתמש שהתחבר למערכת מחזיקה רשימה של מאזינים ומממשת שתי מתודות המאפשרות להוסיף או להוריד מאזין מהרשימה.   
כאשר משלימים את פעולת אתחול המידע מופעלת המתודה notifyAllDataInitializedObservers אשר מודיעה לכל מאזין שהצטרף לרשימה לבצע את המימוש שלהם עבור המתודה AllDataInitialize.

* Sequence Diagram
* Class Diagram



Subject – FacebookAppDataManager

Observer – FormFacebookApplication

IObserver – IDataInitializeObserver

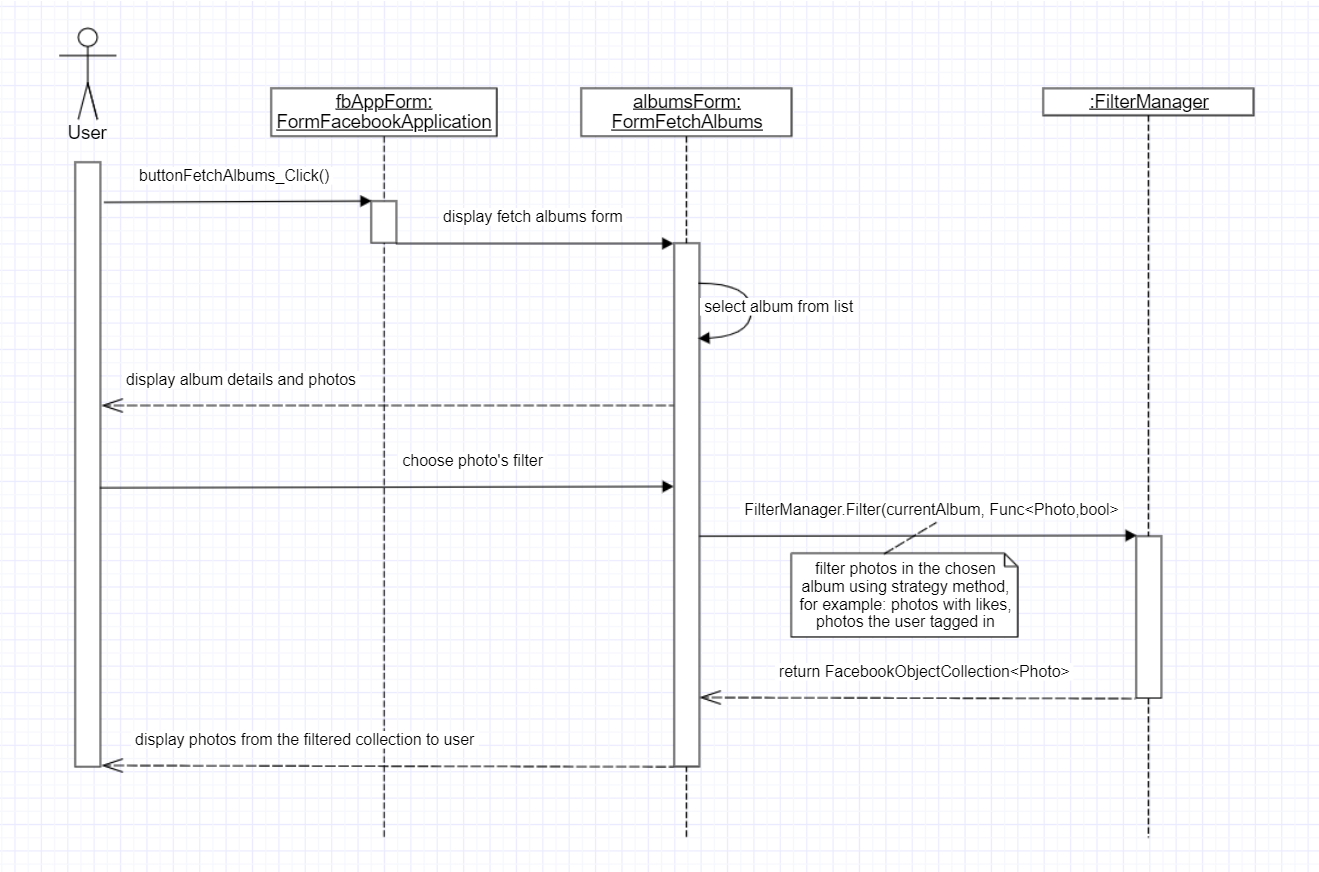
### תבנית מס' 2 – Strategy Method

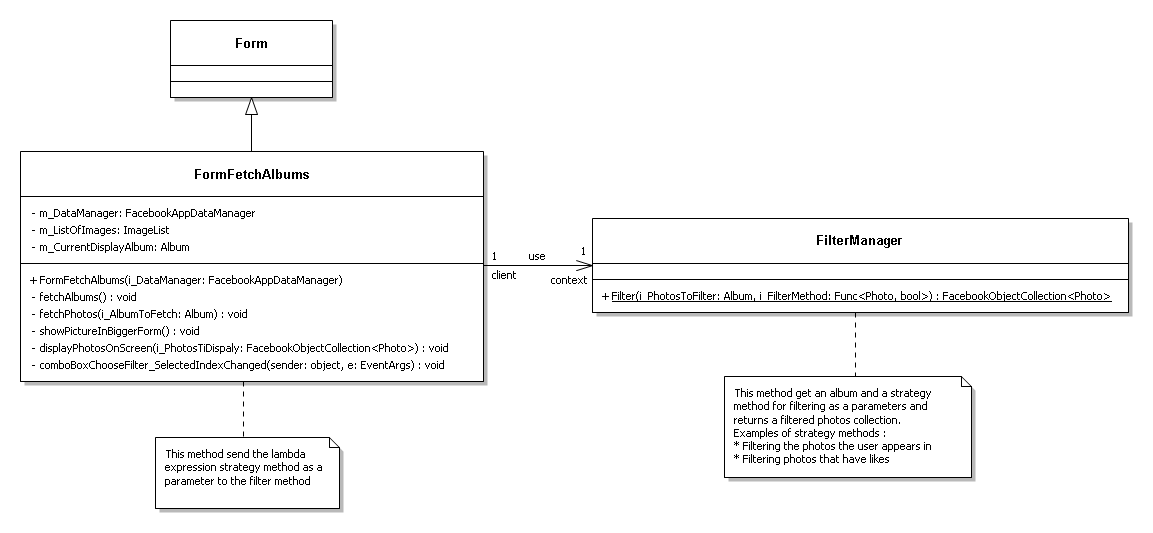
* סיבת הבחירה / שימוש בתבנית:

במערכת שלנו מימשנו פיצ'ר שבו בעת לחיצה על הכפתור Fetch Albums יפתח חלון נוסף עם רשימת האלבומים של המשתמש. כאשר המשתמש ילחץ על שם אלבום מסוים, יופיעו על החלון פרטים על האלבום והתמונות המופיעות בו. רצינו להוסיף את האופציה של סינון התמונות באלבום מתוך רשימת אופציות סינון.  
כדי להימנע משכפול קוד ולאפשר הוספה של אופציות סינון נוספות בעתיד בחרנו להשתמש בתבנית strategy method.  
בחירה זו תאפשר לנו לממש פעם אחת מתודה שתקבל את האלבום הנבחר ומשתמשת באסטרטגיית המיון (בצורה של Func<Photo, bool> ) שנבחרה כדי להחזיר אוסף תמונות "מסונן".

* אופן המימוש:

יצרנו מחלקה סטטית בשם FilterManager אשר בה מימשנו את המתודה Filter. המתודה מקבלת את האלבום שהמשתמש בחר ואת אסטרטגיית הסינון (שנקבעת לפי בחירת המשתמש מרשימת אופציות הסינון) בתור Func<Photo,bool>. המתודה מחזירה FacebookObjectCollection<Photo> (אוסף של תמונות להצגה המקיימות את אסטרטגיית הסינון). השימוש בStrategy Method באופן הנ"ל חוסך את הצורך לממש ממשק המגדיר את אופן המיון ומאפשר לשלוח את אסטרטגיית המיון כlambda expression כאשר האסטרטגיה פשוטה.

* Sequence Diagram
* Class Diagram



Client – FormFetchAlbums

Context – FilterManager

Strategy – Func<Photo,bool> (lambda expression)

### תבנית מס' 3 – Iterator

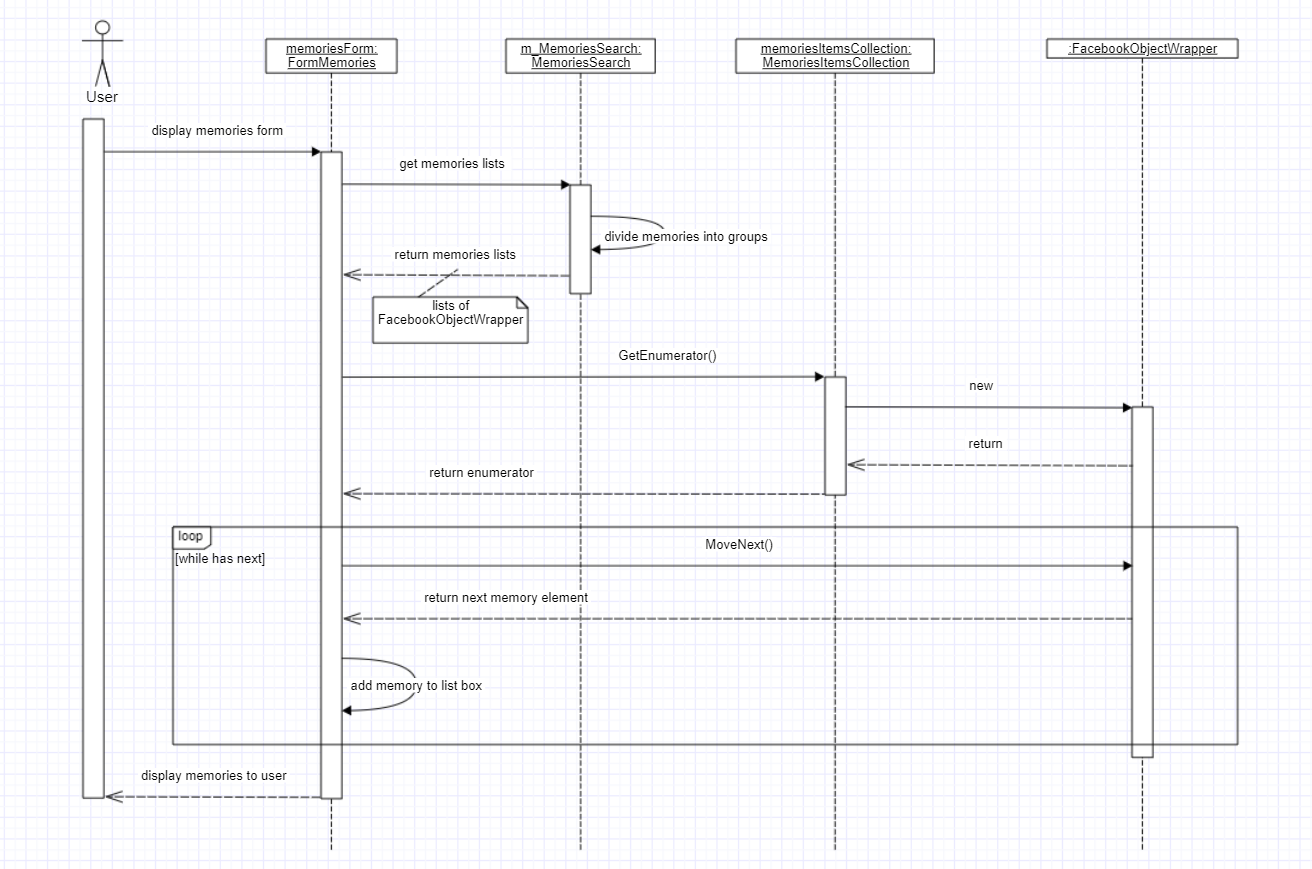
* סיבת הבחירה / שימוש בתבנית:

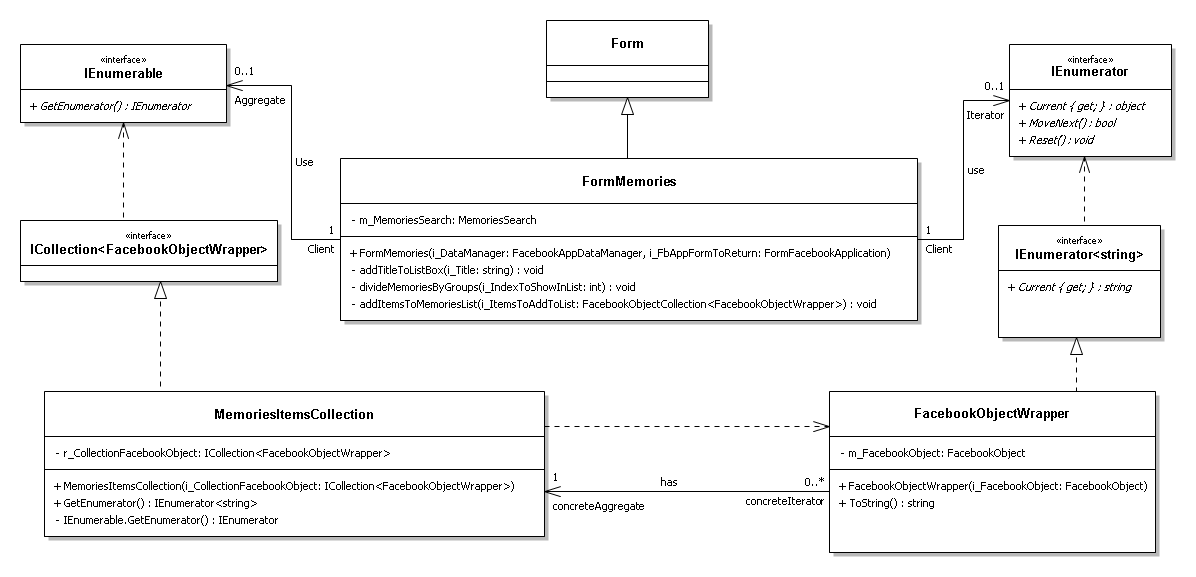
בפיצ'ר שלנו של חיפוש הזכרונות אנו סורקים את הרשימות שאותחלו במחלקה FacebookAppDataManager, על מנת למצוא זכרונות הנמצאים בטווח התאריכים אותם בחר המשתמש. נרצה לאפשר סריקה של נתוני המשתמש ללא הצורך להכיר את מבנה הנתונים בו הם מאוכלסים ובלי לאפשר את האופציה לשנותם.

אנו רוצים להציג למשתמש רק את הנתונים הרלוונטים עבורו ולא את רשימות האובייקטים המלאות.   
לשם כך השתמשנו בתבנית Iterator.

* אופן המימוש:

יצרנו מחלקה חדשה FacebookObjectWrapper העוטפת את FacebookObject ומממשת עבורו את המתודה ToString (באמצעות override). במימוש זה, נחזיר את הנתונים הרלוונטים שאותם נרצה להציג על המסך עבור המשתמש.  
בנוסף יצרנו את המחלקה MemoriesItemsCollections, אשר מממשת את IEnumrable. במחלקה זאת, אנו מחזיקים אוסף של FacebookObjectWrapper שאותו נסרוק ב-Client בעזרת המתודה GetEnumrator, מבלי לחשוף את מבנה הנתונים ומבלי לאפשר לו לבצע שינויים וכך נציג את הנתונים הרלוונטים עבור כל אובייקט הנמצא בו.

* Sequence Diagram
* Class Diagram



Client – FormMemories

Iterator – IEnumerator

ConcreteIterator – FacebookObjectWrapper

Aggregate – IEnumerable

ConcreteAggregate – MemoriesItemsCollection