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

* פיצ'ר 1: קולאז' תמונות

המשתמש יכול להשתמש בפיצ'ר על ידי לחיצה על אופציית ה-Collage שנמצאת בתפריט הצד של האפליקציה.

ברגע שהמשתמש לוחץ על הכפתור Fetch all photos, מוצגות לו כל התמונות המשויכות למשתמש הפייסבוק שלו ב-Listbox, שמתוכו הוא יכול לסמן עד 4 תמונות (על ידי לחיצה על ctrl + left mouse click כדי לסמן או לבטל סימון תמונה), ובסיום הבחירה ללחוץ על הכפתור Make collage, שלאחר לחיצה עליו מוצג לו קולאז' מהתמונות אותן הוא בחר.

הפיצ'ר ממומש במחלקה FormCollage.

* פיצ'ר 2: סינון של אלבומי תמונות לפי פרמטרים

המשתמש יכול להשתמש בפיצ'ר על ידי לחיצה על אופציית ה- Albums בתפריט הצד של האפליקציה.

ברגע שהמשתמש לוחץ על Fetch all albums, מוצגים לו כל האלבומים המשויכים למשתמש הפייסבוק שלו ב- Listbox.

לאחר מכן המשתמש יכול לבחור את אחת מן האופציות של הסינון (לפי שנת יצירת האלבום או לפי מינימום כמות התמונות באלבום) וללחוץ על הכפתור Filter albums כדי לראות את האלבומים מסוננים לפי בחירתו.

הפיצ'ר ממומש במחלקות: FormAlbum, AlbumsFilter, IAlbumFilter, AlbumFilterByYearCreater, AlbumFilterByMinAmountOfPhotos.

### תבנית מס' 1 – Observer

* **סיבת הבחירה / שימוש בתבנית:**

השתמשנו בתבנית במחלקה FacebookUserManager. מחלקה זו מחזיקה את לוגיקת ההתחברות למערכת וכן את המשתמש המחובר, ובמחלקה FacebookForm בה מוצגת האפליקציה הראשית עם תפריט המיוצג על ידי כפתורים.

השתמשנו בתבנית observer כך שההכפתורים בתפריט בFacebookForm יהיו Enabled רק כאשר התבצע אירוע של login מוצלח של משתמש לאפליקציה, כמו כן שהמידע הבסיסי של המשתמש שמתחבר יוצג לאחר התחברות מוצלחת של משתמש.

על ידי שימוש בתבנית, חסכנו שימוש במשתני flag בוליאנים ופישטנו את התהליך של ההתחברות ואת לוגיקת "הדלקת" כפתורי התפריט הראשי.

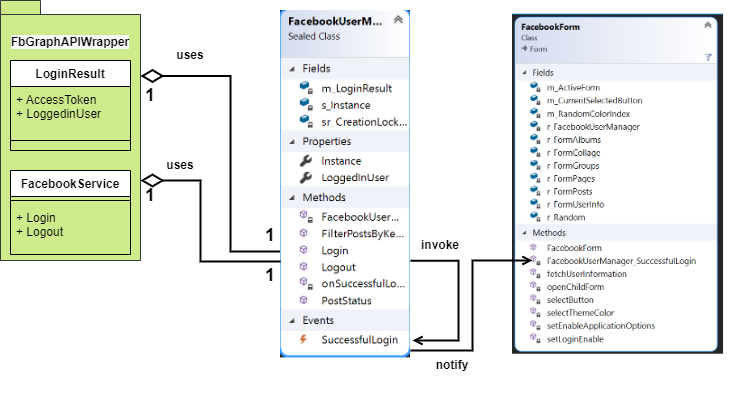
בנוסף תבנית זו מעניקה גם קלות תחזוקה עתידית של האפליקציה – אם בעתיד נרצה להוסיף לוגיקה נוספת שמתבצעת לאחר login מוצלח, נוכל לעשות זאת בקלות על ידי הוספת "מאזין" חדש לאירוע התחברות המוצלחת שיצרנו.

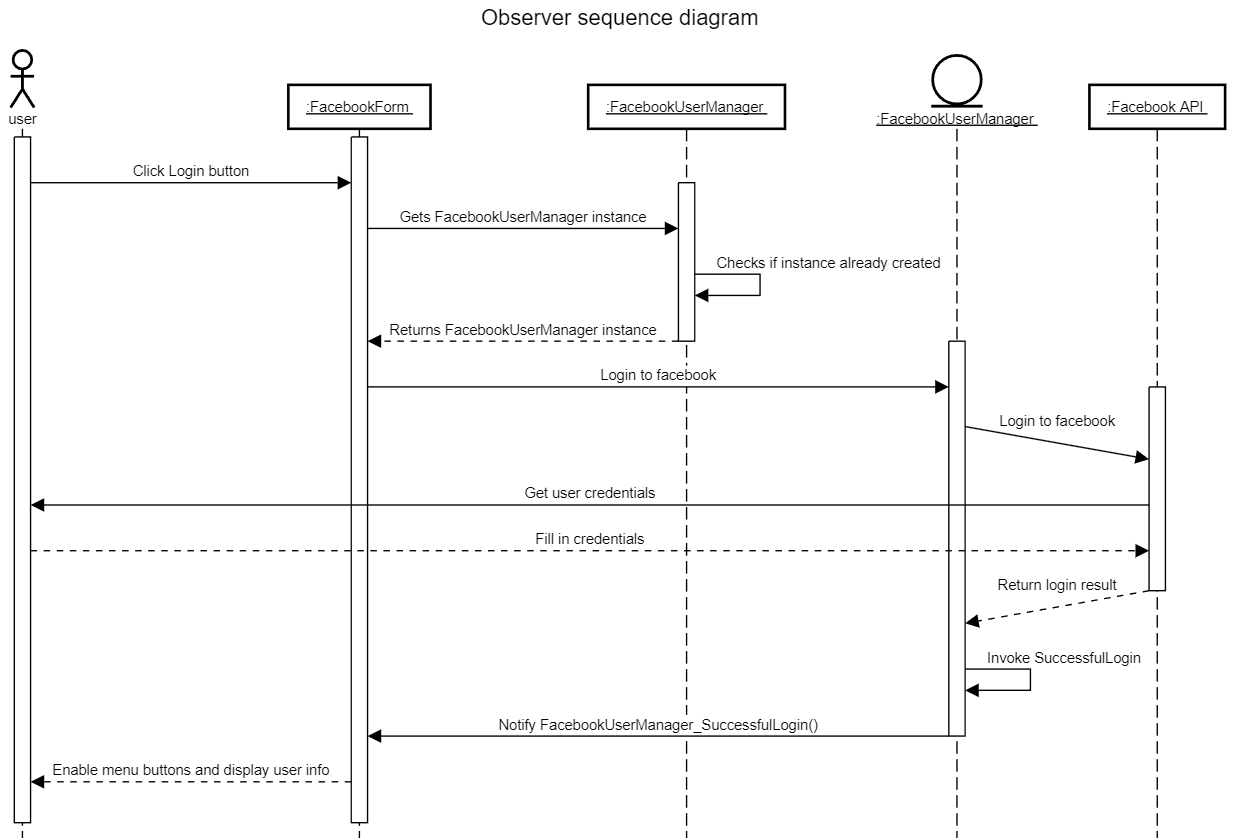
* אופן המימוש:

יצרנו אירוע חדש במחלקה FacebookUserManager בשם SuccessfulLogin ומתודה בשם onSuccessfulLogin שעושה invoke לאירוע כאשר מתבצע login מוצלח (במתודה Login באותה המחלקה).

כמו כן, הוספנו מתודה במחלקה FacebookForm בשם FacebookUserManager\_SuccessfulLogin המקשיבה לאירוע SuccessfulLogin וכאשר מתבצע invoke לSuccessfulLogin היא מופעלת ומטריגה את המתודות המטפלות בהצגת המידע הבסיסי של המשתמש ו-"הדלקת" כפתורי התפריט.

* Class diagram



* Sequence diagram

### תבנית מס' 2 – Strategy

* **סיבת הבחירה / שימוש בתבנית:**

השתמשנו בתבנית זו בפיצ'ר סינון האלבומים לפי פרמטרים.

שמנו לב שכאשר אנחנו רוצים להשתמש בסינון אלבומים לפי פרמטרים שונים, נוצר שכפול של לוגיקת הסינון, כאשר ההבדל היחיד הוא הנקודה בה אנחנו מחליטים האם להכניס אלבום מסוים לרשימה המסוננת או לא על פי פרמטר מספרי כלשהו. פתרנו בעיה זו על ידי שימוש בתבנית Strategy – בנקודת השוני יצרנו נקודת הזרקה של לוגיקת סינון אלבום שמיוצגת על ידי הInterface IAlbumFilter ומימשנו מחלקות שונות המממשות את ממשק זה כדי "להזריק" את הלוגיקה המסננת.

על ידי שימוש בתבנית זו מנענו שכפול קוד והקלנו על תחזוקתיות עתידית – אם בעתיד נרצה לסנן את האלבומים לפי פרמטרים מספריים נוספים, נוכל לעשות זו בקלות על ידי הוספת מחלקה חדשה המממשת את הממשק IAlbumFilter.

* **אופן המימוש:**

יצרנו את הממשק IAlbumFilter, ובו המתודה הבוליאנית FilterAlbum המקבלת כפרמטרים אלבום פייסבוק ופרמטר סינון מספרי כלשהו, ומחזירה True אם האלבום עבר את הסינון או False אם הוא לא עבר את הסינון.

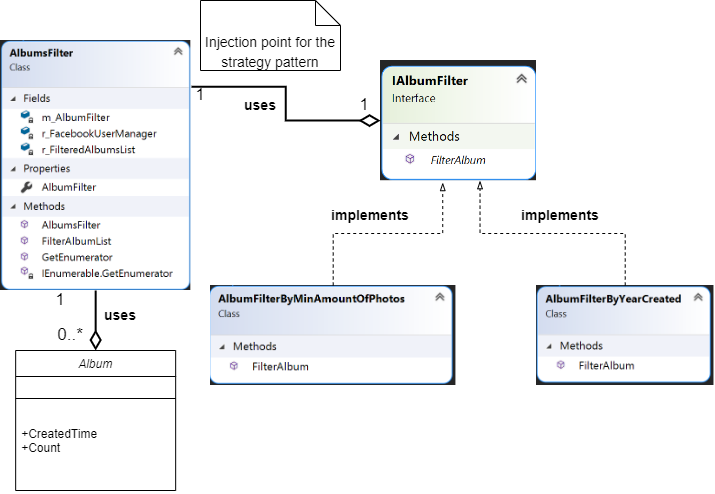
יצרנו שתי מחלקות הממשות את IAlbumFilter – AlbumFilterByMinAmountOfPhotos ו- AlbumFilterByYearCreated, אחת מחזירה אמת אם האלבום מכיל לפחות את כמות התמונות לפי הפרמטר ושקר אחרת.

השנייה מחזירה אמת אם האלבום נוצר בשנה שמיוצגת על ידי הפרמטר המספרי ושקר אחרת.

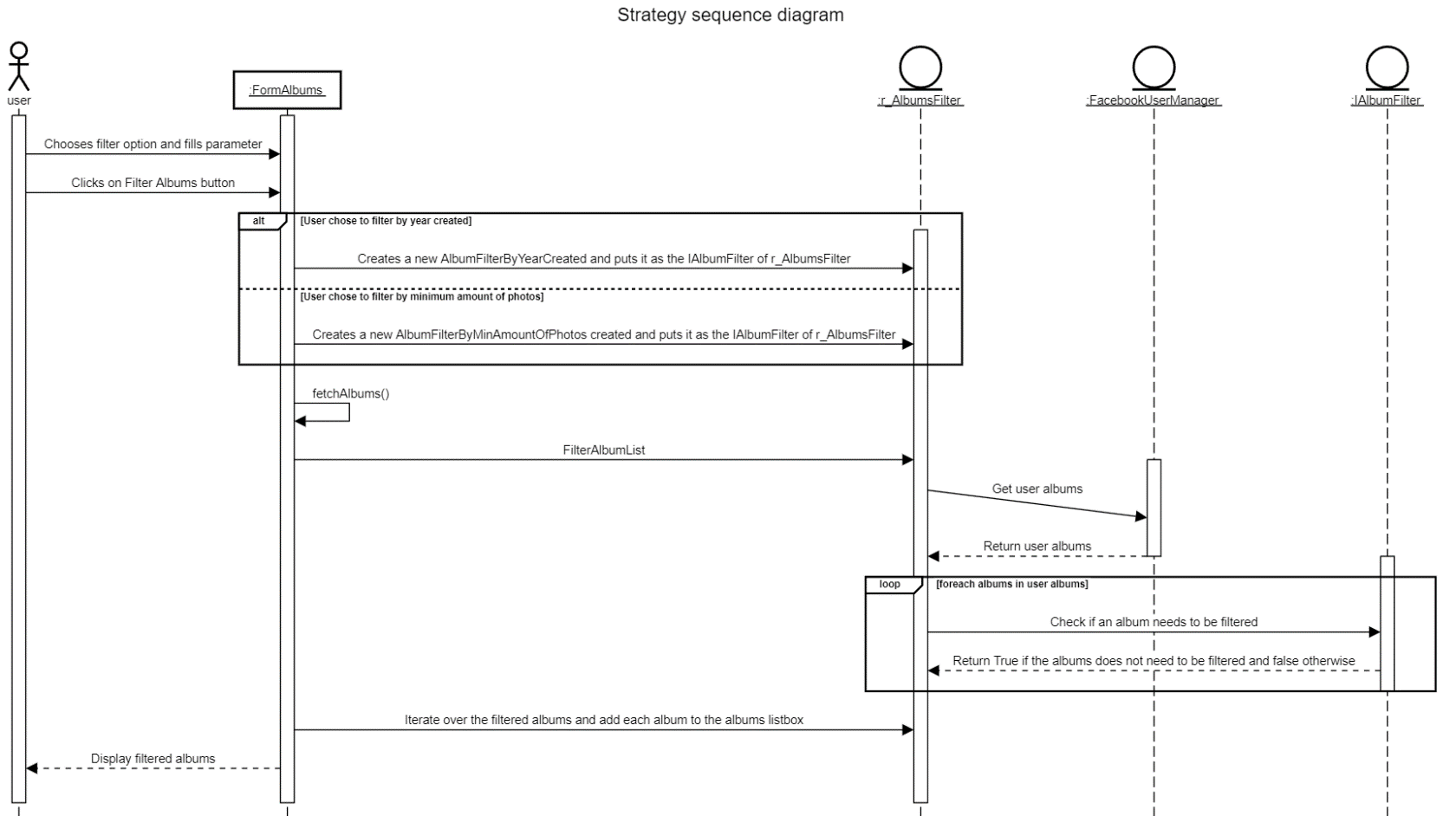
בנוסף יצרנו את המחלקה AlbumsFilter לה שדה מסוג IAlbumFilter, ומתודה משם FilterAlbumList המקבלת פרמטר מספרי המשמש לסינון, עוברת על רשימת האלבומים שנטענו מהמשתמש, ועבור כל אלבום בודקת האם הוא צריך להיות מסונן או לא על ידי המתודה הממומשת על ידי הIAlbumFilter שנמצא כרגע במחלקה AlbumsFilter.

כאשר המשתמש בוחר בפרמטר הסינון ולוחץ על FilterAlbums מופעלת המתודה buttonFilterAlbums\_Click ושם לפי פרמטר הבחירה נוצר IAlbumFilter מתאים לשדה המתאים במחלקה AlbumsFilter, והמתודה fetchAlbums מופעלת, שהיא משתמשת במתודת הסינון של AlbumsFilter כדי לסנן את האלבומים המתאימים מרשימת האלבומים של המשתמש המחובר.

* Class diagram



* Sequence Diagram



### תבנית מס' 3 – Iterator

* סיבת הבחירה / שימוש בתבנית:

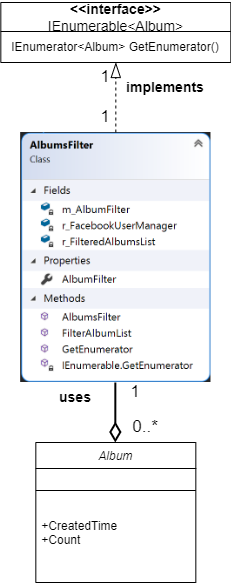
השתמשנו בתבנית זו עבור המחלקה AlbumsFilter על מנת לרוץ על אוסף האלבומים שהיא מחזיקה בקלות, ובאופן שמאפשר תחזוקתיות עתידית קלה – אם בעתיד נרצה לשנות את סוג האוסף שמחזיק את האלבומים, נוכל לעשות זאת בקלות מבלי שנצטרך לשנות את לוגיקת הריצה על האלבומים.

* אופן המימוש:

הגדרנו שהמחלקה AlbumsFilter מממשת את הממשק IEnumerable<Album> ומימשנו בה את המתודות GetEnumarator המחזירות את הערך של GetEnumrator של אוסף האלבומים.

השתמשנו ביכולות של התבנית בריצה באמצעות foreach על כל האלבומים של מופע כלשהו של AlbumsFilter.

* Class diagram



* Sequence diagram

