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

* פיצ'ר ראשון – **Friends Quiz** – פיצ'ר זה בוחר חבר רנדומלי ולאחר מכן יוצר לו חידון, בסוף החידון ניתן לבחור חבר חדש או להפסיק לשחק.
* פיצ'ר שני – **Photos Filter** – פיצ'ר זה מאפשר למשתמש לסנן ולהציג רשימת את רשימת התמונות שלו העונות לקריטריונים הבאים – Publish date, Amount of Likes, Album, Tagged.

### תבנית מס' 1 – [Façade]

* סיבת הבחירה / שימוש בתבנית:

במהלך כתיבת הקוד לאפליקציה נתקלנו בבעיה הבאה – רכיבי ה UI שלנו השתמשו ברכיבים לוגים רבים ואם בעתיד היינו רוצים להוסיף פיצ'ר חדש או לשדרג פיצ'ר קיים היינו צריכים לבצע שינוי ב UI בכדי להתאים אותו. כאן ה Façade נכנס לשימוש ובעזרתו בעצם יצרנו הפרדה בין שכבת ה UI אל הלוגיקה, ה UI יכיר אך ורק את הפונקציות הפשוטות שה Façade מאפשר לו, ה Façade יתקשר עם הרכיבים הלוגים המורכבים וינגיש אותו אל ה UI וכך ובמידה ויתווסף רכיב לוגי חדש הספתו לאפליקציה תרכז את השינוי אך ורק ל Façade ולא ל UI.

* אופן המימוש:

הקוד מומש במחלקות –

Form: MainForm – Façade: MainPage – Logic Class: UserDataManager

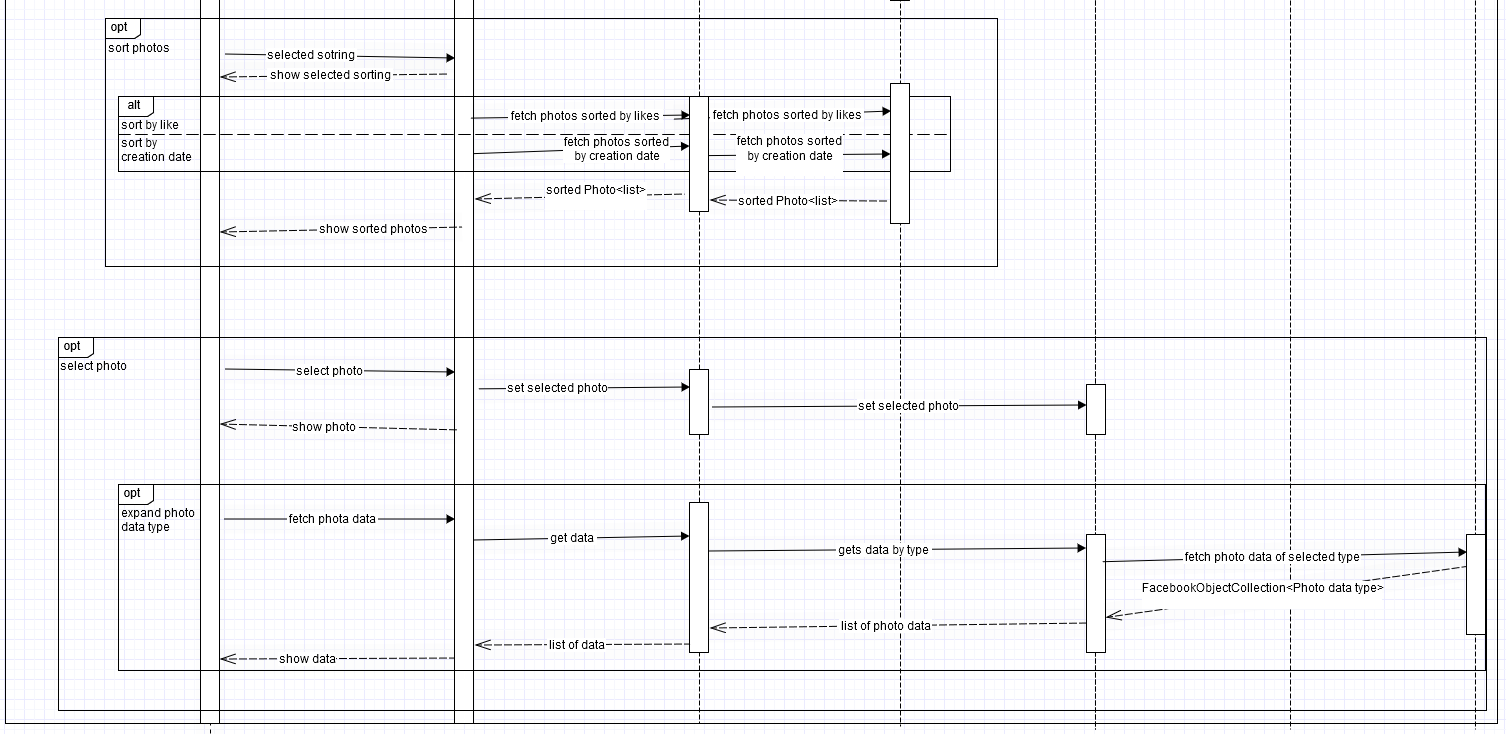
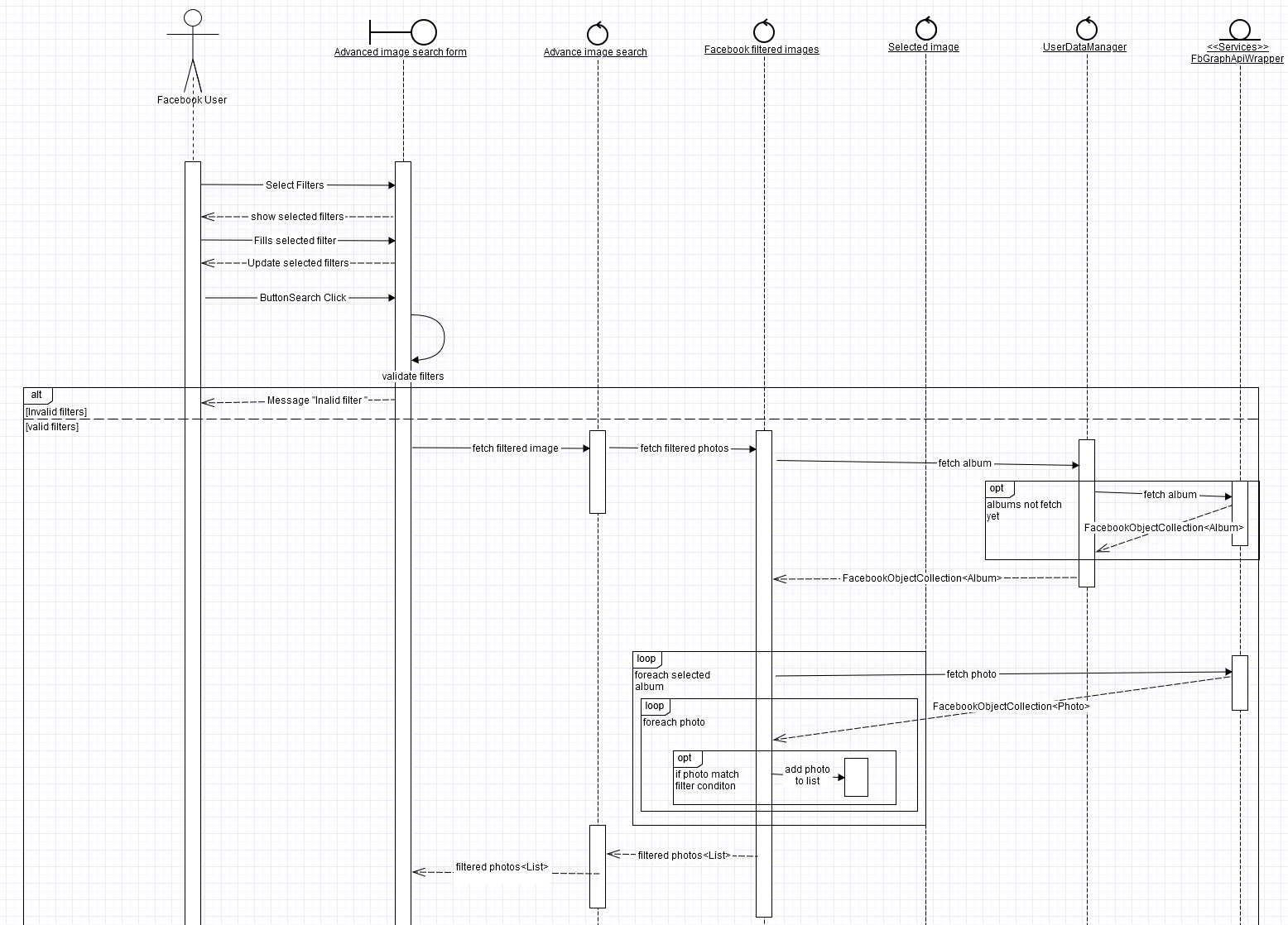
Form: LoginForm – Façade: LoginPage - Logic Class: UserDataManager

Form: FriendsQuizForm – Façade: QuizPage - Logic Class: FacebooksFriends

Form: AdvancedImageForm – Façade: AdvancedImageSearch –

Logic Class: FacebookFilteredImages

* Sequence Diagram



* Diagram

  Description automatically generatedClass Diagram

**ההקבלה בין המחלקות לתבנית:**

Client = FriendsQuizForm, AdvancedImageSearchForm, MainForm, LoginForm

Façade = QuizPage, LoginPage, MainPage, AdvanceImageSearch

Library = FacebookAppLogic

SubSystems = FacebookFilteredImages, SelectedImage, FacebooksFriends,

### תבנית מס' 2 – [Singleton]

* סיבת הבחירה / שימוש בתבנית:

במהלך כתיבת הקוד לאפליקציה נתקלנו בבעיה הבאה – מחלקות רבות עשו שימוש בנתונים המגיעים מפייסבוק, החזקנו אוביקט אשר ניהל את התקשורת עם ה FacebookApiGraphWrapper אך בכל מחלקה כזאת יצרנו מופע חדש של המחלקה המקשרת ובעזרתה שלפנו את הנתונים מהשרת, וגם ביצענו לפעמים את אותן השליפות. את הבעיה הזאת פתרנו בעזרת Singleton, קיצצנו במספר המופעים של המחלקה המתקשרת, וגם יכלנו למנוע קריאות חוזרות לשרת פייסבוק לאסוף data שכבר נמצא ברשותנו.

* אופן המימוש:

הקוד מומש במחלקה – UserDataManager.

הגדרנו את המחלקה כSealed ואת הConstructor שלה כPrivate. אנו מחזיקים מופע סטטי בשם s\_Instance אשר מאותחל עם null. בשימוש הראשון של המחלקה במהלך התוכנית נוצר מופע חדש ו s\_Instance כבר לא null. מעכשיו בכל פעם שמשתמשים במחלקה ומבקשים את ה instance נחזיר את מה שיצרנו. פעולת הGet של הInstance כוללת Double Check-Lock אשר הופכת את המחלקה להיות Thread-Safe. בכל שלב בה מתרחשת גישה לשרת, אותו מידע ששלפנו נשמר כמידע של הInstance. בכל שימוש חוזר, במקום לתקשר עם השרת, נחזיר את המידע אשר נשלף מקודם.

* Sequence Diagram

תהליך יצירה:

Diagram

Description automatically generated

תהליך שימוש כללי

Diagram, schematic

Description automatically generated

* Class Diagram

Diagram

Description automatically generated

**ההקבלה בין המחלקות לתבנית:**

Singleton = UserDataManager

Client = MainPage, LoginPage, AdvanceImageSearch, FacebookFilteredImages, FacebookFirends, QuizPage

### תבנית מס' 3 – [Adapter]

* סיבת הבחירה / שימוש בתבנית:

במהלך כתיבת הקוד לאפליקציה נתקלנו בבעיה הבאה – כאשר אנו מציגים למשתמש את הפוסט שלו אנו מציגים אך ורק את הטקסט אותו הפוסט מכיל, רצינו להציג למשתמש גם את תאריך ושעת כתיבת הפוסט אך לא היה ניתן לעשות זאת מתוך קולקציה וכך נוצרה אי-התאמה בין שני חלקי Legacy. הראשון הוא אובייקט Post והשני הוא פקד התצוגה ב form שלנו (ListBox). השתמשנו בפטרן שלנו בכדי ליצור פורמט חדש בו משתלבים כמה נתונים מהאובייקט ויוצרים מחרוזת חדשה אשר נוכל להציג למשתמש.

* אופן המימוש:

הקוד מומש במחלקה – StatusAdapter.

בעת יצירה של מופע מחלקה עם רפרנס ל Post אותו אנו רוצים להציג נוכל להמיר לפורמט אותו אנו רוצים, המחלקה תממש את הממשק IStatusFormat אשר ממשת את הפונקציה ליצירת הפורמט הרצוי.

* Diagram

  Description automatically generatedSequence Diagram
* Class Diagram

Diagram

Description automatically generated

Client – who use, Adapter – StatusFormatAdapter, Adaptee – Post

### עבודה אסינכרונית

התכנות האסינכרוני אצלנו היה בשימוש כאשר ביצענו פעולות השולפות מידע מהשרת של פייסבוק.

בטופס Main Form כאשר הטופס עולה ומבקש למלא את החלונות/פרטי המשתמש בפונקציה retrieveUserProfileData כל חלק אחר יוצא לת'רד אחר.

בטופס AdvancedImageSearchForm כאשר מחלצים תמונות בעזרת הפילטור מהשרת, וכאשר רוצים למשוך מידע כגון תגובות,תיוגים ולייקים של התמונה הנבחרת.

### עבודה עם Data Binding

את השימוש ב Data Binding השתמשנו כאשר רצינו להציג מידע למשתמש מאובייקטים שונים.

בטופס Main Form כאשר מוצג ליוזר ה Events/Pages/Groups – כל אחד מה ListBox מקושר בעזרת Data Binding אל האובייקט המתאים.