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

* הפיצ'ר הראשון שבחרנו לממש הוא פיצ'ר שמנתח סטטסיטקה לגבי תמונות באלבומים או פוסטים לפי כמות מקסימלית של לייקים.

לפיצר יש טופס משלו, ובלחיצה על כפתור ה- Top Fan נפתח חלון הפיצ'ר בו מוצגות שלוש אפשרויות לקבלת רשימה ממויינת של כלל החברים. בנוסף, הפיצ'ר מספק אפשרות סינון של חברים לפי מין וטווח גילאים. לאחר שבחר נשלחים הנתונים למחלקה שבודקת את הסטטיסטיקה הדרושה ומציגה אותו למשתמש.

* הפיצ'ר השני שבחרנו לממש הוא פיצ'ר החוקר את פעילויות המשתמש. בלחיצה על כפתור ה- Covid-19 נפתח חלון הפיצ'ר בו מוצג כפתור לקבלת רשימה של חברים שעל פי פייסבוק שהו בקרבת המשתמש בתקופת השבועיים האחרונים. זאת באמצעות חקירה מעמיקה של תמונות שתוייג המשתמש בתקופת הזמן, סימון Check Inשתוייג/העלה המשתמש באותה התקופה ומחזיר את החברים שתויגו עמו.

### תבנית מס' 1 – Factory Method

(קישור נוסף ל-Class Diagram ו- Sequence Diagram באיכות טובה יותר: <https://f2h.io/676ikv96shk2> )

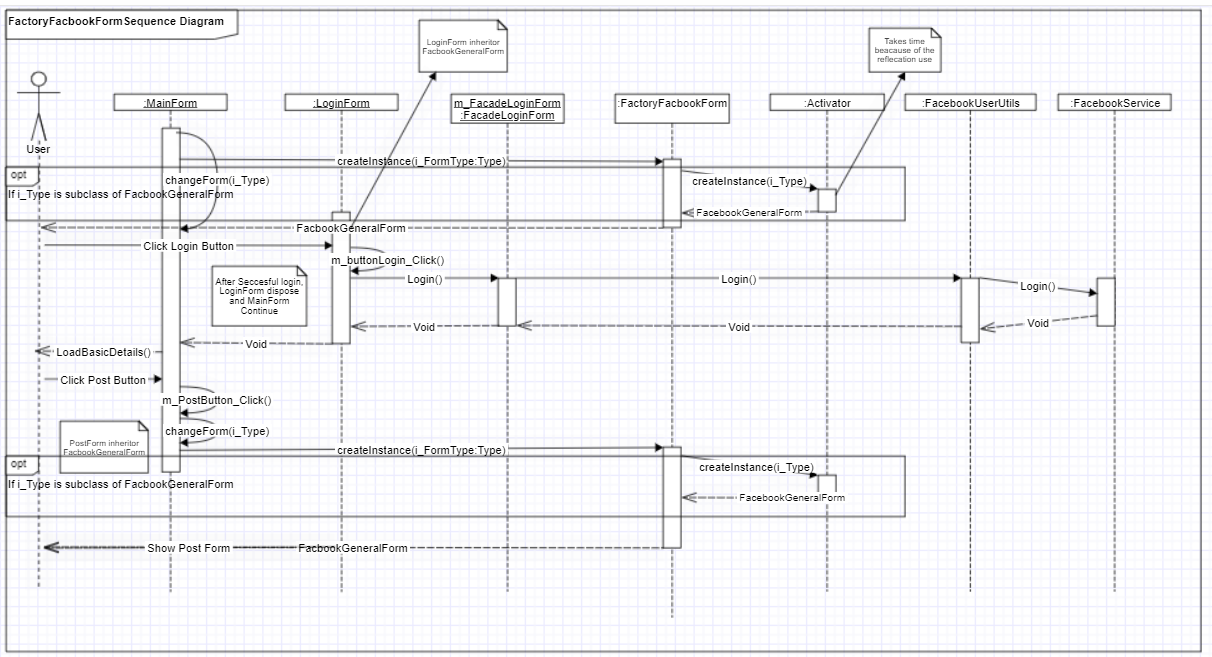
* סיבת הבחירה / שימוש בתבנית:

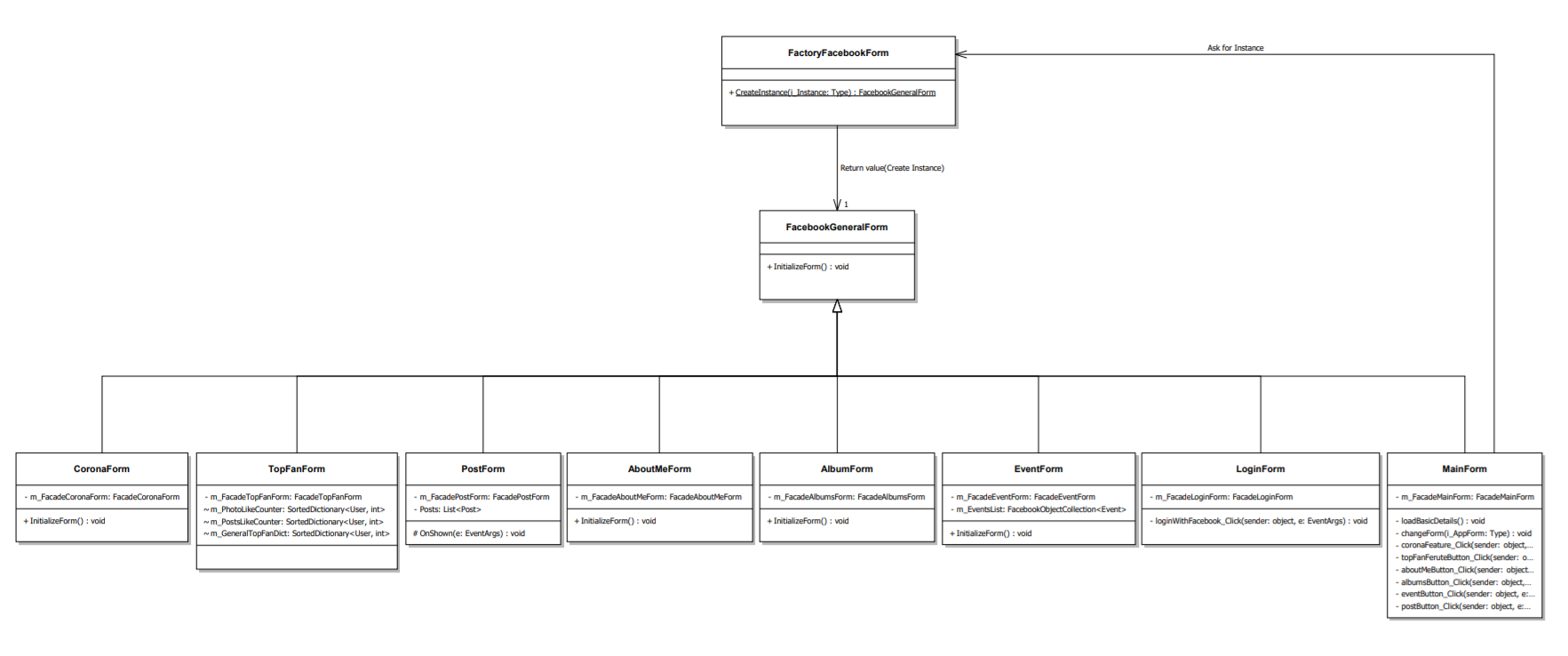
בחרנו להשתמש בתבנית הנ"ל מכיוון שישנה בתוכנית משפחה פולימורפית (FacebookGeneralForm) של טפסים שהיצירה שלהם נעשת ב-Client.

* אופן המימוש:

המימוש נוצר באמצעות מחלקה חדשה שאחראית על יצירת הטפסים המחלקה החדשה נקראת FactoryFacebookForm ובתוכה מטודה אחת סטטית שמקבלת כפרמטר Type ובודקת האם הפרמטר שקיבלנו הוא subclass של המחלקה .FacebookGeneralForm במידה וכן, המחלקה יוצרת מופע שלו באמצעות המחלקה Activator (שימוש ב-refelction ) וכתוצאה מכך המחלקה לא מכירה את האוביקטים שהיא יוצרת.

ניתן לראות כי המחלקה MainForm היא המחלקה המייצגת את תפקיד ה – Client וכן באמצעות שימוש במטודה changeForm() מתבצעות בקשות החלפת הטפסים.

* Sequence Diagram
* Class Diagram



* על פי התבנית המחלקה המקבילה ל - Product היא: FacebookGeneralForm
* על פי התבנית המחלקות המקבילות ל - ConcreteProduct הן: LoginForm, EventForm, AlbumsForm, PostForm, AboutMeForm, TopFanForm, CoronaForm.

### תבנית מס' 2 –Pattern Façade

(קישור נוסף ל-Class Diagram ו- Sequence Diagram באיכות טובה יותר: <https://f2h.io/ixsyl1wcycax>)

* סיבת הבחירה / שימוש בתבנית:

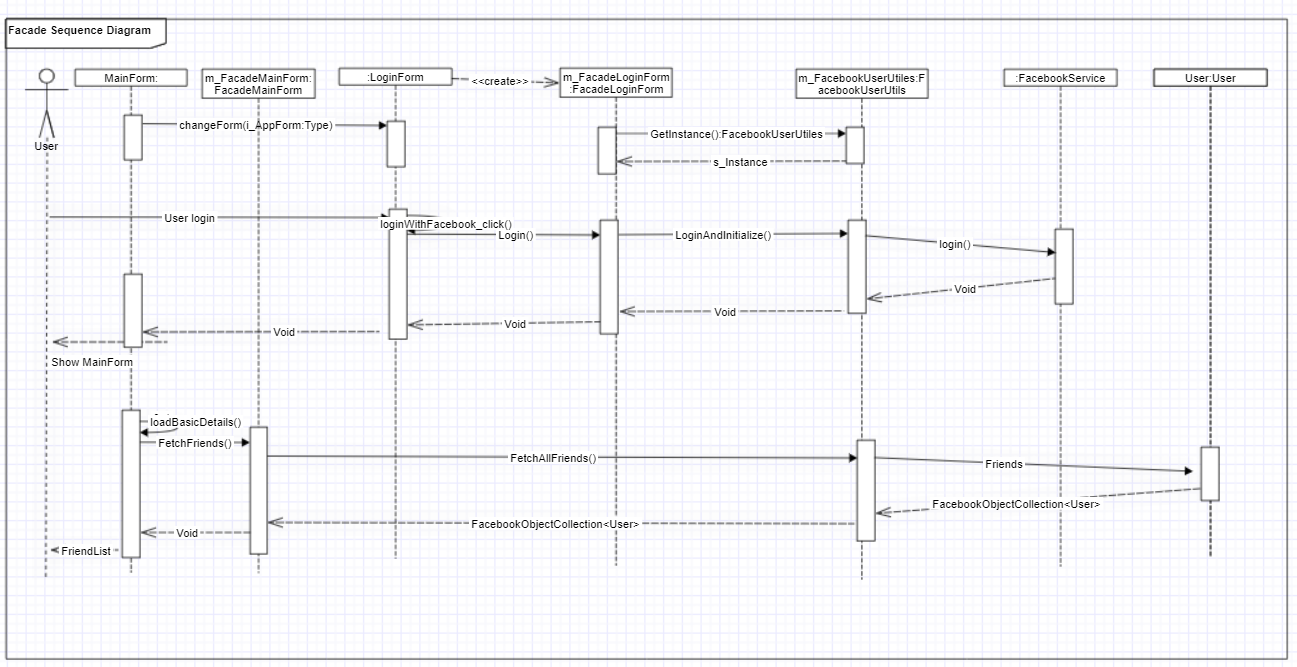
בחרנו לממש את התבנית Façade pattern (שקוף) על מנת לענות לצורך של:

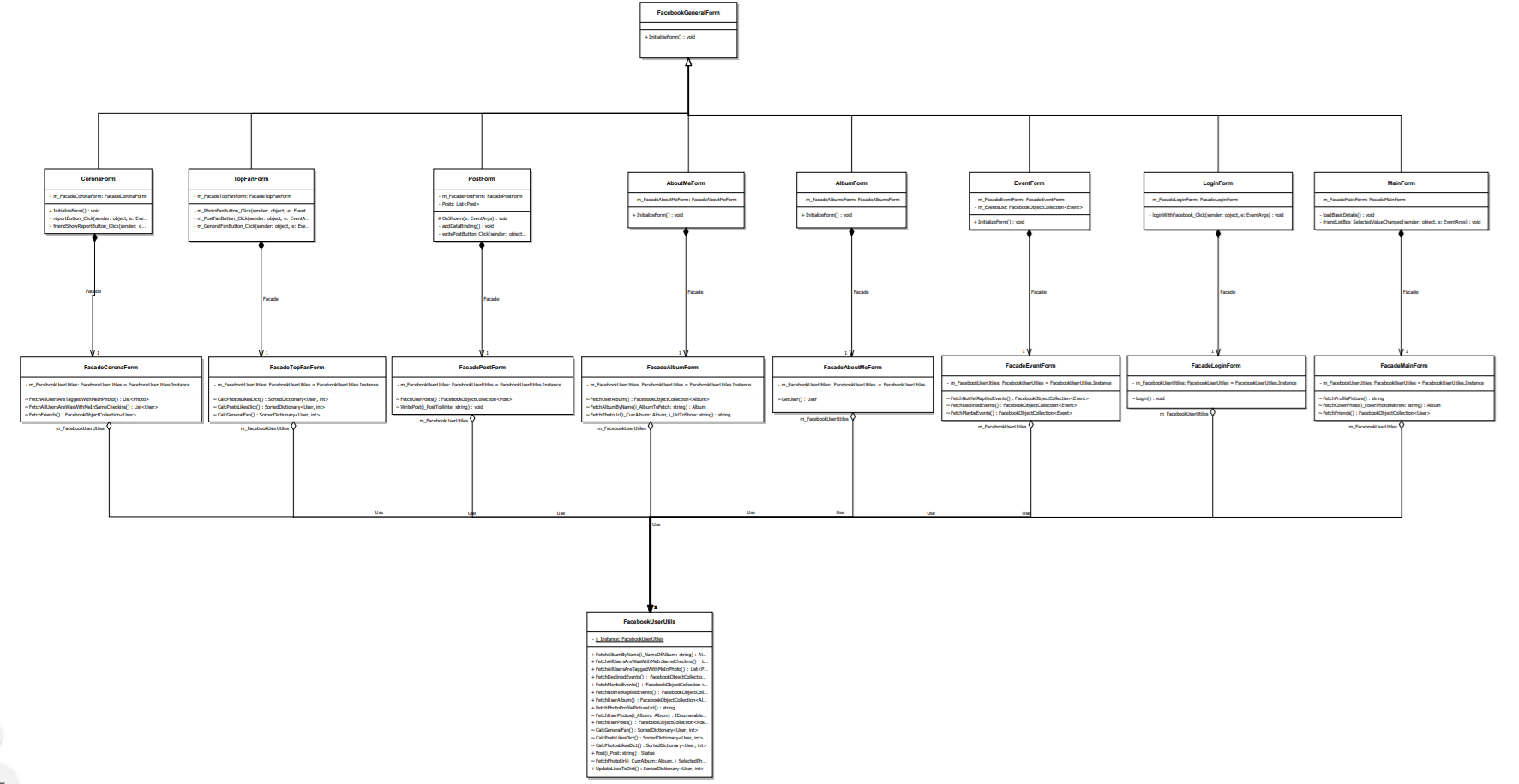
* הפרדת מחלקות הלוגיקה ממחלקות ממשק המשתמש.
* בשל שימוש ב- Data Binding נוצר צורך לעבד את הנתונים מהלוגיקה. כתוצאה מכך, המחלקה שמייצגת את ה-Facade "משטחת את הלוגיקה" כדי שממשק המשתמש יוכל להשתמש בנתונים בצורה נוחה יותר.
* הפעלת מניפוצליות שונות על המידע המתקבל ממחלקות הלוגיקה. לעיתים היה צורך לעשות מניפוציות שונות כמו סידור ומיון במידע המתקבל ורק לאחר מכן להעביר מידע רלוונטי לממשק המשתמש.
* אופן המימוש:

לכל טופס בתוכנית יש Facade משלו. זאת כדי לממש בסיס לארכיטקטורת MVVM:

* המחלקה FacadeAlbumsForm משמשת Facade עבור המחלקה (טופס)

AlbumsForm המשמשת כ-Client.

* המחלקה FacadeEventForm משמשת Facade עבור המחלקה (טופס) EventForm המשמשת כ-Client.
* המחקלה FacadeTopFanForm משמשת Facade עבור המחלקה (טופס) TopFanForm המשמשת כ-Client.
* המחלקה FacadePostForm משמשת Facade עבור המחלקה (טופס) PostForm המשמשת כ-Client.
* המחלקה FacadeMainForm משמשת Facade עבור המחלקה (טופס)   
  MainForm המשמשת כ-Client.
* FacadeAboutMeForm המשמשת Facade עבור המחלקה (טופס)  
   AboutMeForm המשמשת כ-Client.
* FacadeLoginForm משמשת Facade עבור המחלקה (טופס)  
   LoginForm המשמשת כ-Client.
* FacadeCoronaForm המשמשת Facade עבור המחלקה (טופס)  
   CoronaForm המשמשת כ-Client.
* Sequence Diagram



* Class Diagram

### תבנית מס' 3 – Singleton Pattern

(קישור נוסף ל-Class Diagram ו- Sequence Diagram באיכות טובה יותר: <https://f2h.io/ozgzjy6a7eim> )

* סיבת הבחירה / שימוש בתבנית:

בחרנו להשתמש בתבנית הנ"ל מכיוון שתכונית ישנה מחלקה FacebookUserUtils האחראית על הלוגיקה, מחלקה זאת עושה פעולות לוגיות על האוביקטים שנמצאים בתוך ה- Facbook dll .

כתוצאה מכך, אם ניצור יותר ממופע אחד של המחלקה ונשנה בצורה בלתי תלויה את מצבם של המופעים השונים זה יכול לפגוע במערכת כיוון שלא נדע איזה מצב רלוונטי ברגע נתון,

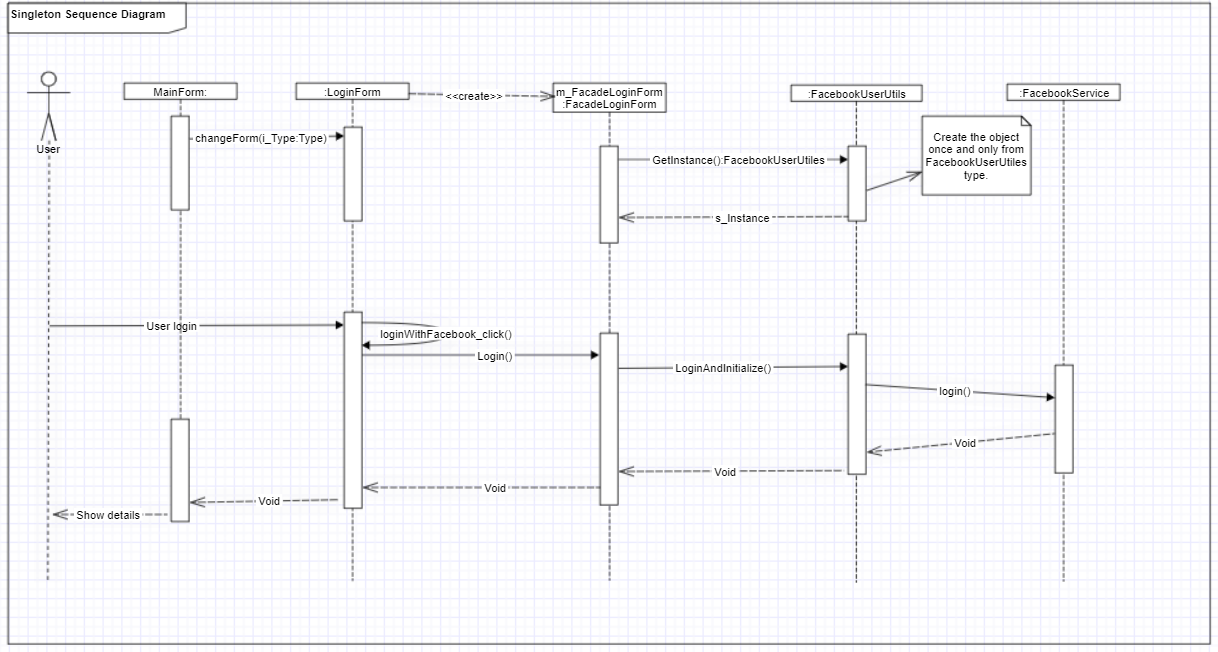
ולכן כאשר אנו רוצים להישאר את אותו המידע בדיוק בכל המופעים, נובעת הסיבה שלא נצטרך ליצור יותר ממופע אחד.

בנוסף מכיוון שמחלקה זו אחראית על הלוגיקה, יש לאפשר לכל מחלקה את השימוש בה, ביוזמתה ללא צורך העברתה בין המחלקות ודאגה על שמירת המידע.

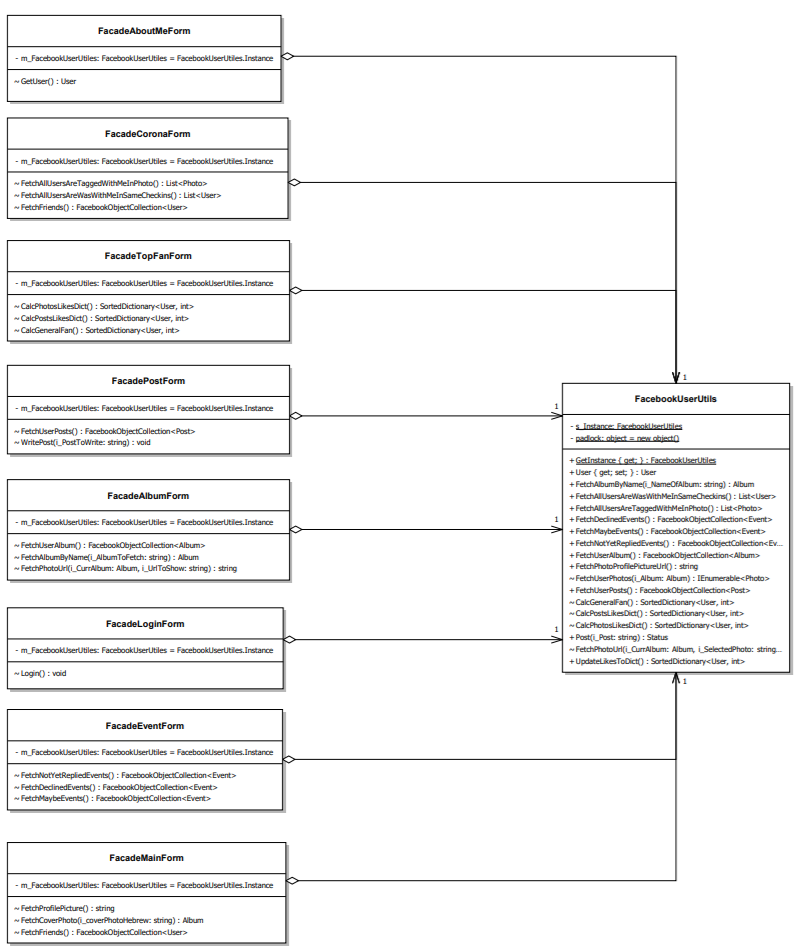
אופן המימוש:

* המחלקה FacebookUserUtils אחראית לייצר מופע אחד שלה בדיוק הכולל הגנה מפני Threads. בתוך המחלקה קיים DataMember סטטי s\_Instance שבאמצעות

Properties סטטי בשם Instance שתפקידו לבדוק אם כבר קיים מופע של ה – s\_Instance ואם יש, להחזירו או אם אין, אז לייצר אחד כזה.



* Sequence Diagram
* Class Diagram



### Threads:

* במחלקה MainForm במטודה OnShown בשורה 32 בקוד נעשה שימוש בעבודה אסינכורית.

העבודה האסינכורית מפעילה מטודה בשם LoadBasicDetails שדואגת לאתחל את כל המידע הנחוץ למשתמש ומידע הבסיסי שלו בחלון ה- MainForm ולכן, יש צורך לעבוד בצורה אסינכורית מכיוון שתהליך זה לוקח זמן ותוקע את ממשק המשתמש עד שיסתיים.

* במחלקהPostForm במטודה onShown בשורה 35 בקוד נעשה שימוש בעבודה אסינכורית.

העבודה אסינכורית מפעילה מטודה בשם FetchUserPostהמטודה הזאת דואגת להביא את הפוסטים של המשתמש.

לאחר שהביא את הפוסטים של המשתמש היא מחברת אותם לDataBinding

תהליך זה הוא דבר שלוקח זמן מכיוון שהפנייה לשרתי פייסבוק הוא דבר שלוקח זמן ולכן העדפתי להשתמש בעבודה אסינכורית כדי שממשק המשתמש יהיה פנוי.

* כמו כן, השתמשנו במטודה Invoke ובתוכה בLambadaExpresstion שמפעילה Action עבור פעולות עבודה עם ממשק המשתמש ופקדי הטפסים כדי שנוכל לפנות ולשלוט בהם בכל תהליכון:
  + במחלקה MainForm בשורה 55 על מנת לעדכן את מיקום וצורת תמונת הפרופיל.
  + במחלקה MainForm בשורה 77 על מנת להציג את רשימת החברים ב ListBox המתאימה.
  + במחלקה TopFanForm בשורות 54 ו-90 על מנת להציג את תוצאות חיפוש החברים הרלוונטית ב ListBox.
  + במחלקה CoronaForm בשורה 74 על מנת להציג מידע שעבר עיבוד ל ListBox.

### Data Binding:

* במחלקה AlbumsForm נעשה שני שימושים ב Data Binding :
  + בשורה 29 נעשה שימוש Data Binding בכדי לקשר את האלבומים מה- Facade
  + בשורה 39 נעשה שימוש Data Binding בכדי לקשר את התמונות מהאלבום הנוכחי שכרגע הcurrent של Data Binding של האלבומים מצביע עליו.
* במחלקה EventForm נעשה שימוש ב Data Binding לפי בחירת סוג האירוע:
* בשורה 48 נעשה שימוש Data Binding בכדי לקשר את ה- MaybeEvent   
  מה- Facade.
* בשורה 53 נעשה שימוש Data Binding בכדי לקשר את ה- NotYetReplayedEvent מה- Facade.
* בשורה 58 נעשה שימוש Data Binding בכדי לקשר את ה- DeclinedEvent   
  מה- Facade.
* במחלקהAboutMeForm נעשה שימוש ב Data Binding :
* בשורה 24 נעשה שימוש Data Binding בכדי לקשר את המידע הנוכחי על המשתמש מה- .Facade
* במחלקה PostForm נעשה שימוש ב Data Binding :
* בשורה 30 נעשה שימוש Data Binding בכדי לקשר את הפוסטים של המשתמש מה- Facade.