תיאור קצר של הפיצ'רים שבחרנו לממש בתרגיל הקודם:

- מציאת שידוך פוטנציאלי: (Find Match Feature)
  הפיצ'ר מאפשר למצוא התאמות עבור דייטים פוטנציאליים בשביל המשתמש על ידי הזנת טווח גילאים ומגדר מועדף.
  בנוסף, הדייטים הפוטנציאליים ממוינים בסדר יורד על בסיס מספר הדפים האהובים המשותפים שיש בין המשתמש והדייט הפוטנציאלי.

- מציאת פוסט ותמונת השנה: (Find Greatest Of The Year)
  הפיצ'ר מאפשר למשתמש לבחור שנה ולקבל בחזרה את התמונה והפוסט (לפי השנה שהוזנה) שקיבלו הכי הרבה תגובות מהחברים שלו.

תבנית מס' 1 – Singleton

- <u>סיבת הבחירה / שימוש בתבנית:</u>

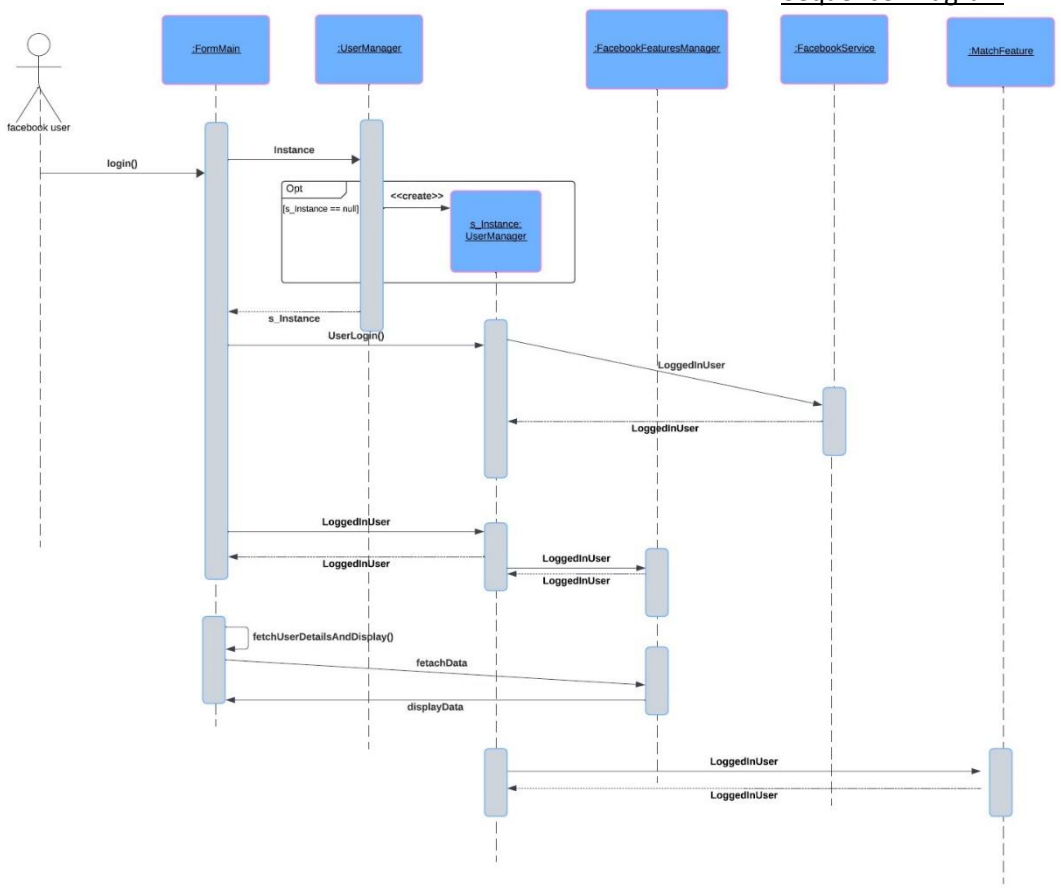
באפליקציה שלנו מימשנו מחלקה בשם UserManager, תפקידה הוא לנהל את המשתמש במערכת שלנו אל מול שרתי פייסבוק. המחלקה הזו מאפשרת פעולות התחברות והתנתקות מפייסבוק ובנוסף מחזיקה מידע מפייסבוק אודות המשתמש (ה-User) המחובר כעת למערכת.
FormMain עושה שימוש במחלקה UserManager, כדי לבצע התחברות או התנתקות ומחלקות שונות במערכת עושות שימוש במחלקה כדי לשלוף נתונים על המשתמש המחובר למערכת.
כחלק מאפיון האפליקציה קיים משתמש אחד בלבד המחובר למערכת, לכן לא רצינו לאפשר יצירה של יותר ממופע יחיד של המחלקה (אם יהיה יותר ממופע 1 זה לא תקין). בנוסף רצינו לאפשר לרכיבים שונים במערכת כולל רכיבים עתידיים, לעשות שימוש במופע היחיד של המחלקה ביוזמתם.
התבנית Singleton Pattern עונה לצרכינו מהאפיון, לכן בחרנו לממש את המחלקה UserManager כמחלקה סינגלטונית.

- <u>אופן המימוש:</u>

בחרנו לממש באופן המלא של סינגלטון, על ידי המימוש הבא:
הסתרה של בנאי המחלקה, כלומר: private ctor כדי שלא יהיה ניתן לייצר מופעים מחוץ למחלקה.
הגישה למופע של המחלקה נעשה באמצעות רפרנס private static למופע היחיד של המחלקה, שנקרא s_Instance.
הגדרת המחלקה כ- sealed כדי לעזור בהתלבטות שלא יהיה ניתן לרשת מהמחלקה.
המימוש הוא Thread safe על ידי שימוש במנעול שעוטף את שלב יצירת האובייקט.

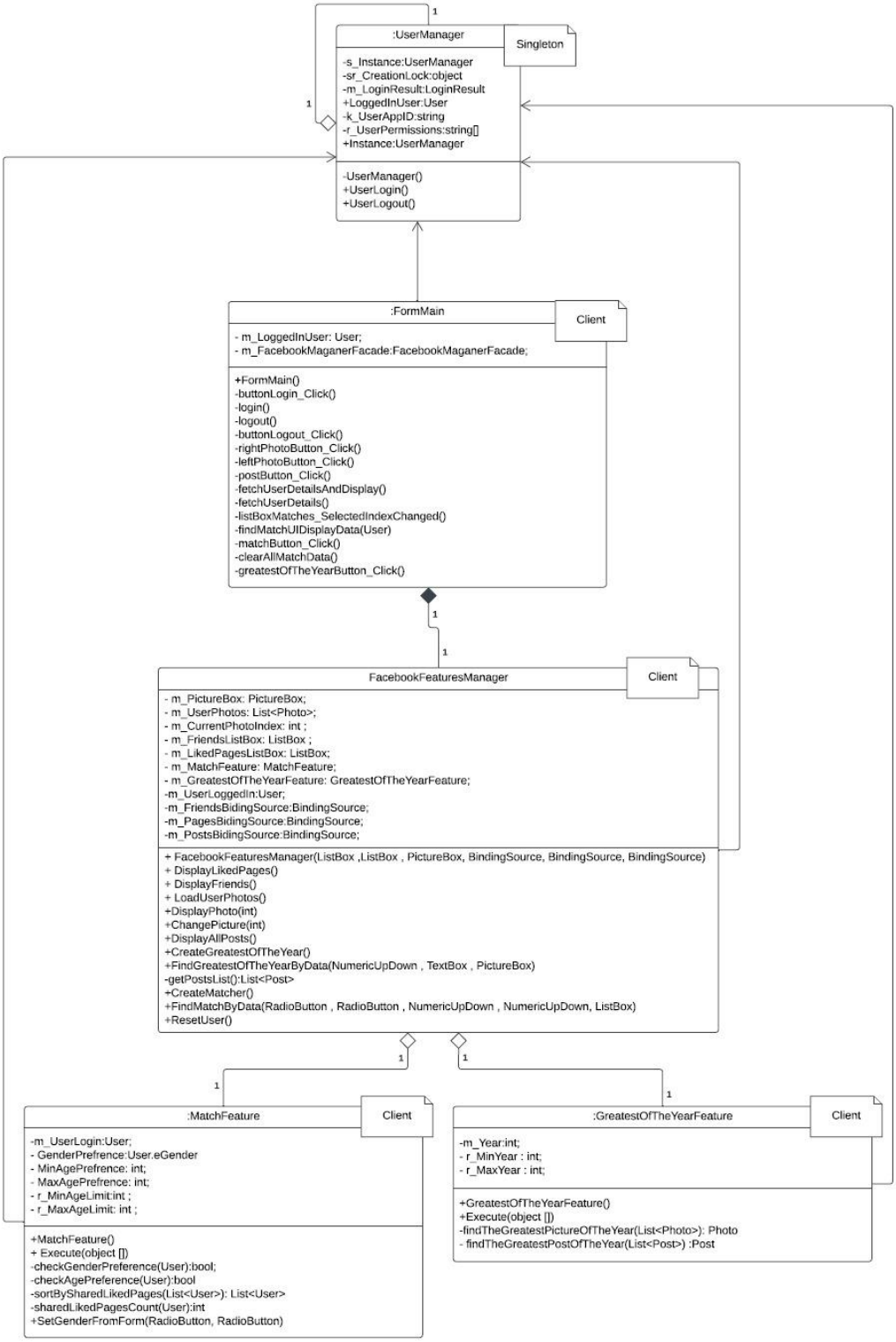
- ניתן למצוא את המימוש בקוד תחת התיקייה Singleton.

- <u>Sequence Diagram</u>



- Class Diagram

Singleton: UserManager
Client: FormMain
Client: FacebookFeatureManager
Client: MatchFeature
Client: GreatestOfTheYearFeature



תבנית מס' 2 – Factory

- <u>סיבת הבחירה:</u>

באפליקציה שלנו מימשנו 2 פיצ'רים שונים :
1. MatchFeature – פיצ'ר בו המשתמש יכול למצוא שידוך פוטנציאלי על בסיס העדפות מתוך רשימת החברים שלו.
3. GreatestOfTheYearFeature – פיצ'ר בו המשתמש יכול לראות את הפוסט והתמונה הפופולריים של שנה מסויימת.

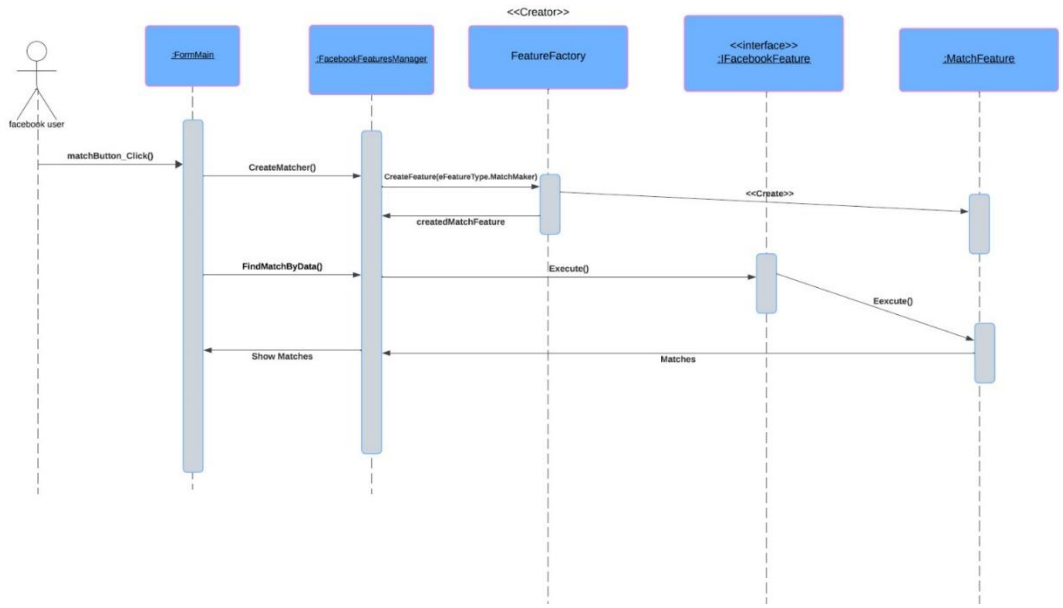
המחלקות הממשות את שני הפיצרים יורשות מ IFacebookFeatureלכן מהוות משפחה פולימורפית, בנוסף במהלך השימוש באפליקציה נרצה לבחור וליצור אובייקטים מהמשפחה.
לכן מהאפיון נובע שנרצה להשתמש בדיזיין פטרן Factory Method כדי לבצע את מלאכת הבחירה והיצירה של אובייקטים מהמשפחה במודל ולא ב-Client.
באופן זה, הוצאנו את מלאכת הבחירה והיצירה של הטפסים למקום אחד בקוד במודל, במקום שיהיה מפוזר ב – Client.
אם בעתיד נרצה להוסיף פיצ'ר חדש לאפליקציה נצטרך רק לבצע שינויים קלים במחלקת FeaturesFactory שאחראית על בחירת וייצור הפיצ'רים.

- <u>אופן המימוש:</u>

בחרנו לממש בתצורת Static Factory Class שהיא תצורה reality driven, אקסטנסבילית ותחזוקתית, לכן תצורה טובה מבחינה הנדסית.
יצרנו מחלקה סטטית בשם FeaturesFactory, שאחראית על מלאכת הבחירה והיצירה של הפיצ'רים השונים.
במחלקה מימשנו מתודה סטטית CreateFeature, המקבלת כפרמטר enum מסוג eFeatureType ומכיל את סוגי הפיצ'רים השונים אותם ניתן לייצר.
המתודה יוצרת את הפיצ'ר המתאים לפי ה-enum ומחזירה רפרנס אליו כאינטרפייס IFacebookFeature.
ה- Client משתמש במחלקה FeaturesFactory ובמתודה CreateFeature בכדי לייצר את הפיצ'ר הרצוי.

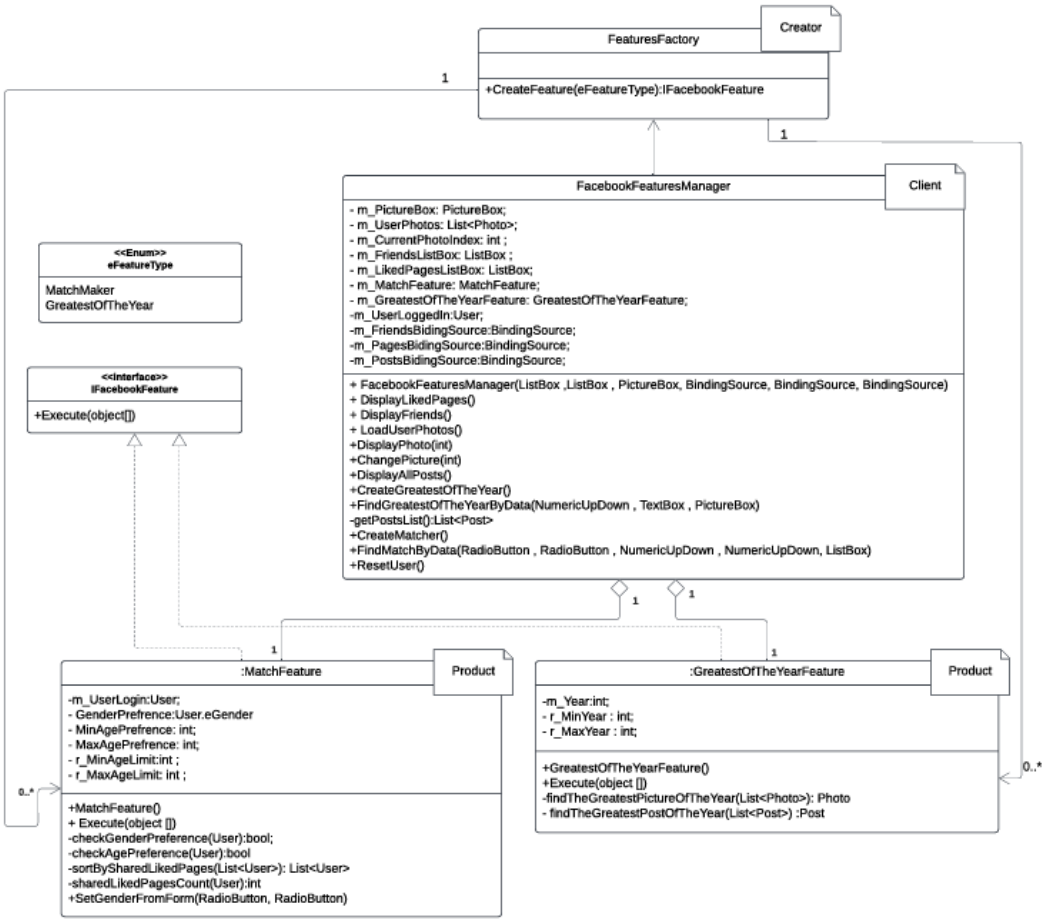
- ניתן למצוא את המימוש בקוד תחת התיקייה Factory.

- <u>Sequence Diagram</u>



- <u>Class Diagram</u>

Creator: FeatureFactory
Client: FacebookFeatureManager
Product: MatchFeature
Product: GreatestOfTheYearFeature



תבנית מס' 3 – Façade

- <u>סיבת הבחירה / שימוש בתבנית:</u>

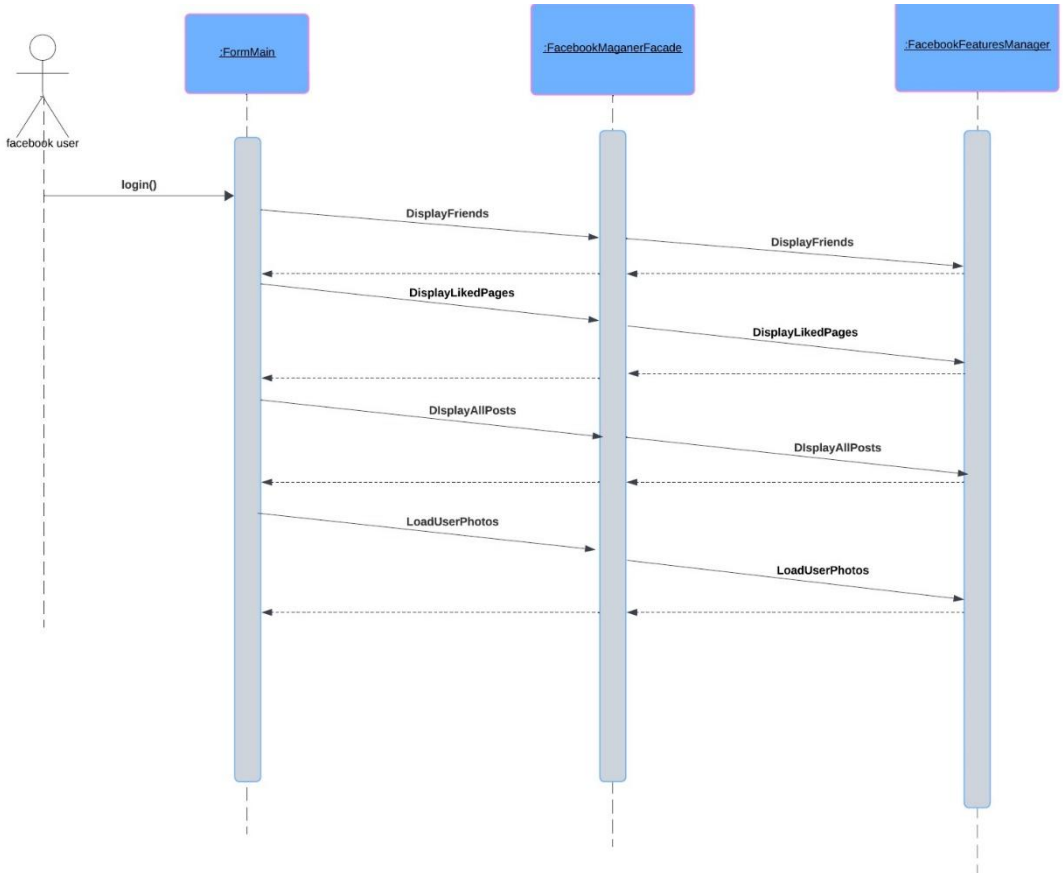
ה-Facade ממומש באמצעות המחלקה FacebookMaganerFacade, אשר מספקת ממשק פשוט ואחיד לשימוש בפונקציות של מחלקת FacebookFeaturesManager. מחלקה זו אחראית לרכז את כל הקריאות לפונקציות הקשורות לניהול חברים, עמודים, תמונות, פוסטים, והתאמות אישיות, ומאפשרת למשתמשים במערכת לגשת לפונקציונליות המלאה של ניהול התכונות הללו מבלי להתעסק בפרטי המימוש הפנימיים.
המטרה היא להסתיר את הפרטים המורכבים של FacebookFeaturesManager ולספק ממשק ברור וידידותי למשתמש.

- <u>אופן המימוש:</u>

Facade מספק מתודות פשוטות כגון DisplayFriends, LoadUserPhotos, ו-DisplayAllPosts.
מתודות אלו משמשות כשכבת הפשטה מעל המתודות המפורטות במחלקת FacebookFeaturesManager.

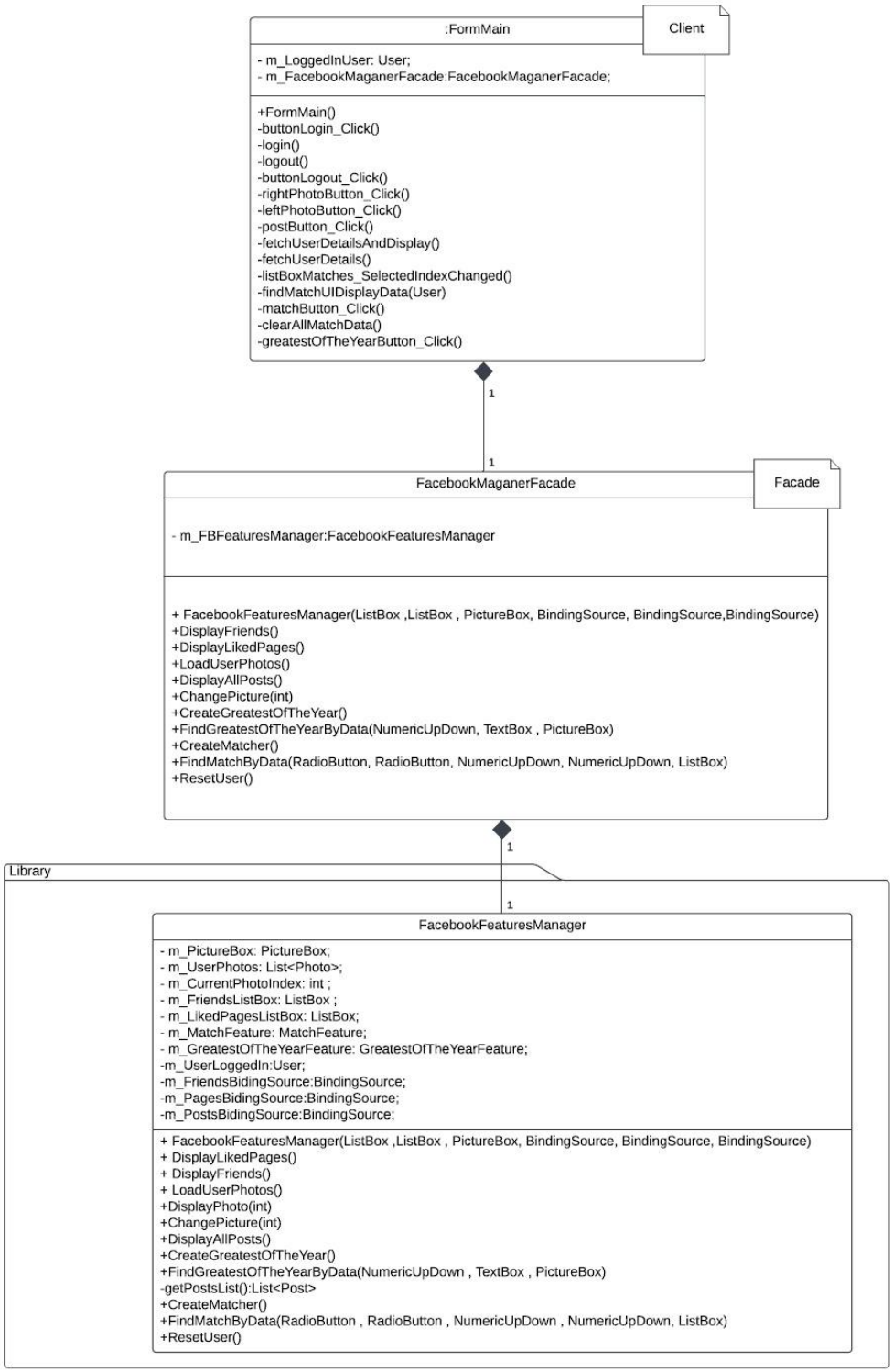
- ניתן למצוא את המימוש בקוד תחת התיקייה Facade.

- <u>Sequence Diagram</u>



- <u>Class Diagram</u>

Client: FormMain
Façade: FacebookManagerFacade



עבודה אסניכרונית עם ממשק משתמש:

מימשנו במחלקת FormMain של האפליקציה, השימוש בתכנות אסינכרוני ממחיש את הצורך לעבד נתונים רבים משרתי פייסבוק במקביל תוך שמירה על חוויית משתמש חלקה ומהירה. כל פעולה שדורשת פנייה לשרת מתבצעת ב Thread-נפרד, מה שמאפשר ביצוע מקבילי של מספר פעולות ללא חשש לתקיעות או השהיות בממשק המשתמש.

העבודה האסינכרונית ב- FormMain כוללת את הפעולות הבאות:

- שליפת נתוני המשתמש המחובר והצגתם בממשק.
- טעינת פרטים נוספים כמו פוסטים, אלבומים, עמודים שאהב, חברים וכו'...
- פרסום פוסטים חדשים ועדכונם ברשימת הפוסטים של המשתמש.

הבחירה בשימוש בתכנות אסינכרוני נבעה מהצורך להבטיח שהממשק משתמש ישאר תגובתי ויעיל, בזמן שביצוע הפעולות השונות ובעיקר הפעולות הארוכות של משיכת מידע משרתי פייסבוק לא יהווה עומס על האפליקציה. זה מאפשר למשתמש להמשיך לנווט באפליקציה ולבצע פעולות נוספות בזמן שנתונים נטענים ברקע, מה שמשפר את חוויית השימוש ומזרז את התגובה של האפליקציה.

עבודה עם Data binding:

השתמשנו ב-Data Binding במחלקת ה FacebookFeaturesManager בצורה שבאה לידי ביטוי בהצגת רשימת החברים, הצגת הדפים שאהב, והצגת הפוסטים שהמשתמש פרסם.