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

* אלגוריתם למציאת הזמן הטוב ביותר לפרסום פוסט לפי שעה שבה ממוצע הלייקים הוא הגבוה ביותר.
* סינון מתקדם בהצגה של פוסטים לפי מקום, כמות לייקים וכמות תגובות.

### תבנית מס' 1 – Singleton

* סיבת הבחירה / שימוש בתבנית:

בחרנו בתבנית זו עבור המחלקה LoggedinUserData אשר מחזיקה את פרטי המשתמש.

ההחלטה התקלה בעקבות הסיבה שבכל רגע נתון באפליקציה יכול להיות רק משתמש אחד שמחובר ובמופע שלו נעשה שימוש בחלקים שונים באפליקציה.

* אופן המימוש:

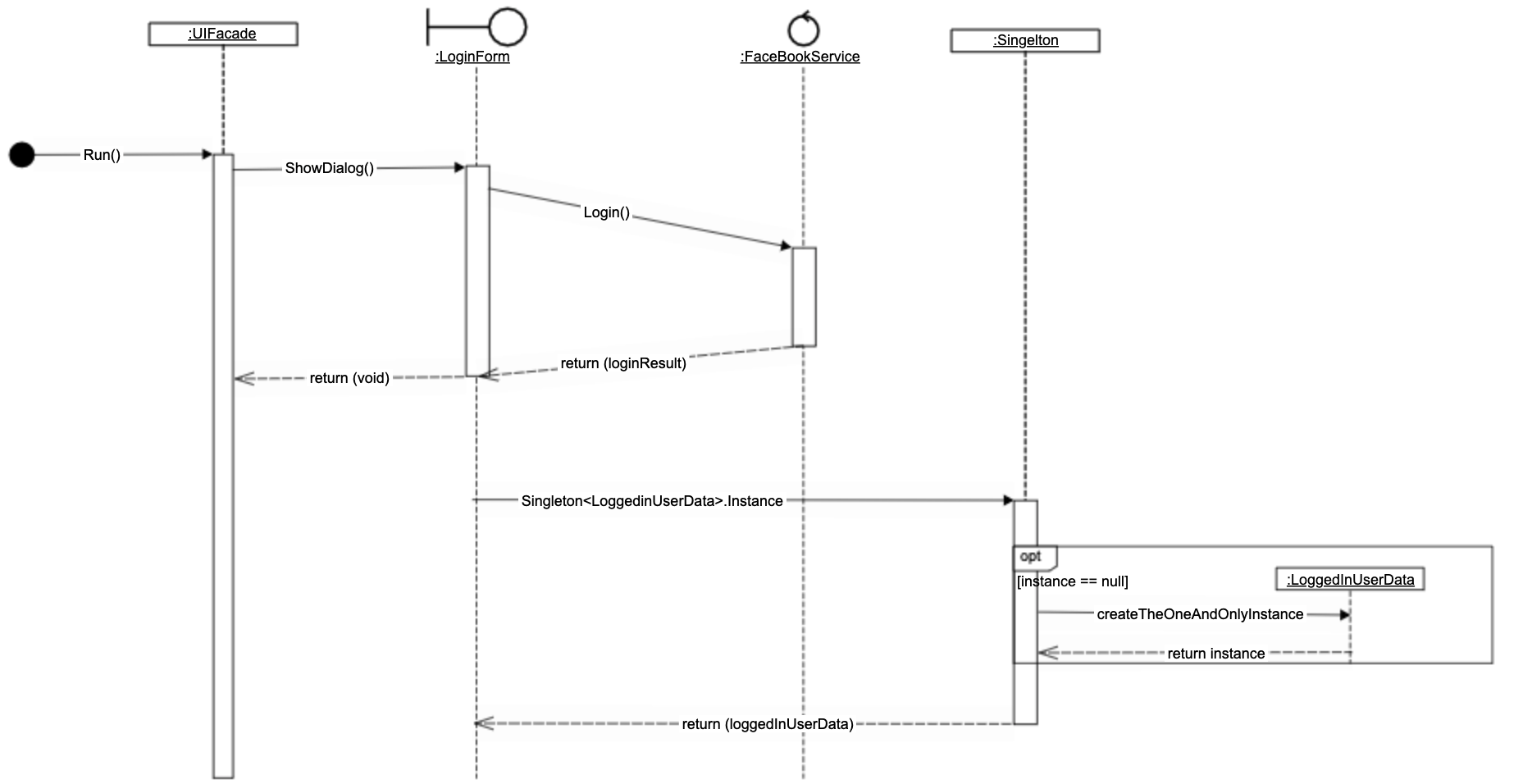
אופן המימוש נעשה על ידי הסתרת הבנאי של המחקלה LoggedinUserData עם

private modifier ובעזרת מחלקה נוספת טמפלית Singleton<T> המבצעת

double-check lock כאשר פונים למופע היחיד.

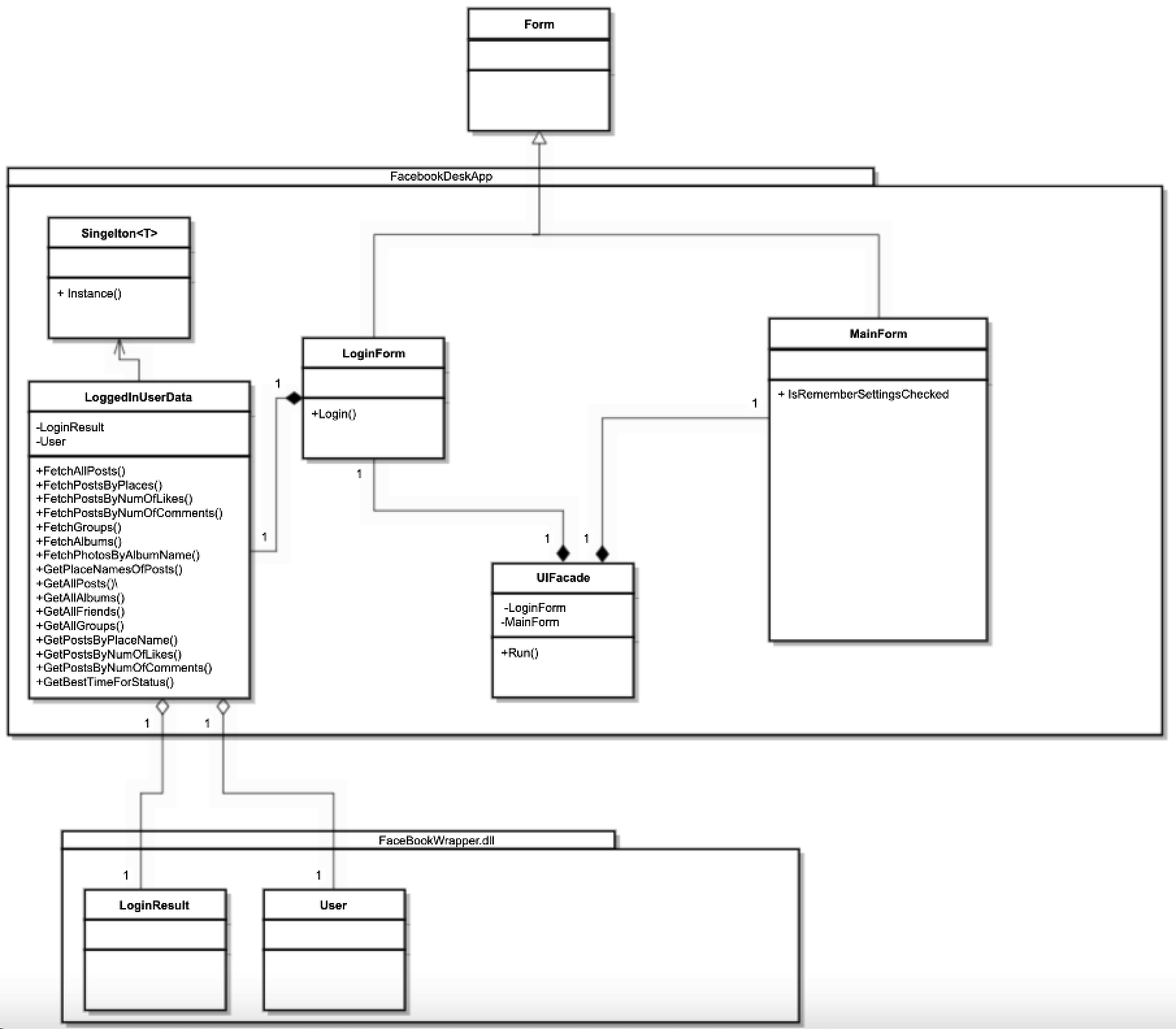
השימוש הראשון במופע קורה בטופס LoginForm במתודה Login() כאשר מתבצעת אליו השמה של פרטי המשתמש שהתקבלו לאחר ההתחברות.

* Sequence Diagram



הערה: המחלקה FaceBookServiceProxy לא מצויין בדיאגרמה, מפני שרצינו להתמקד ב – Singelton.

* Class Diagram



### תבנית מס' 2 – Facade

* סיבת הבחירה / שימוש בתבנית:

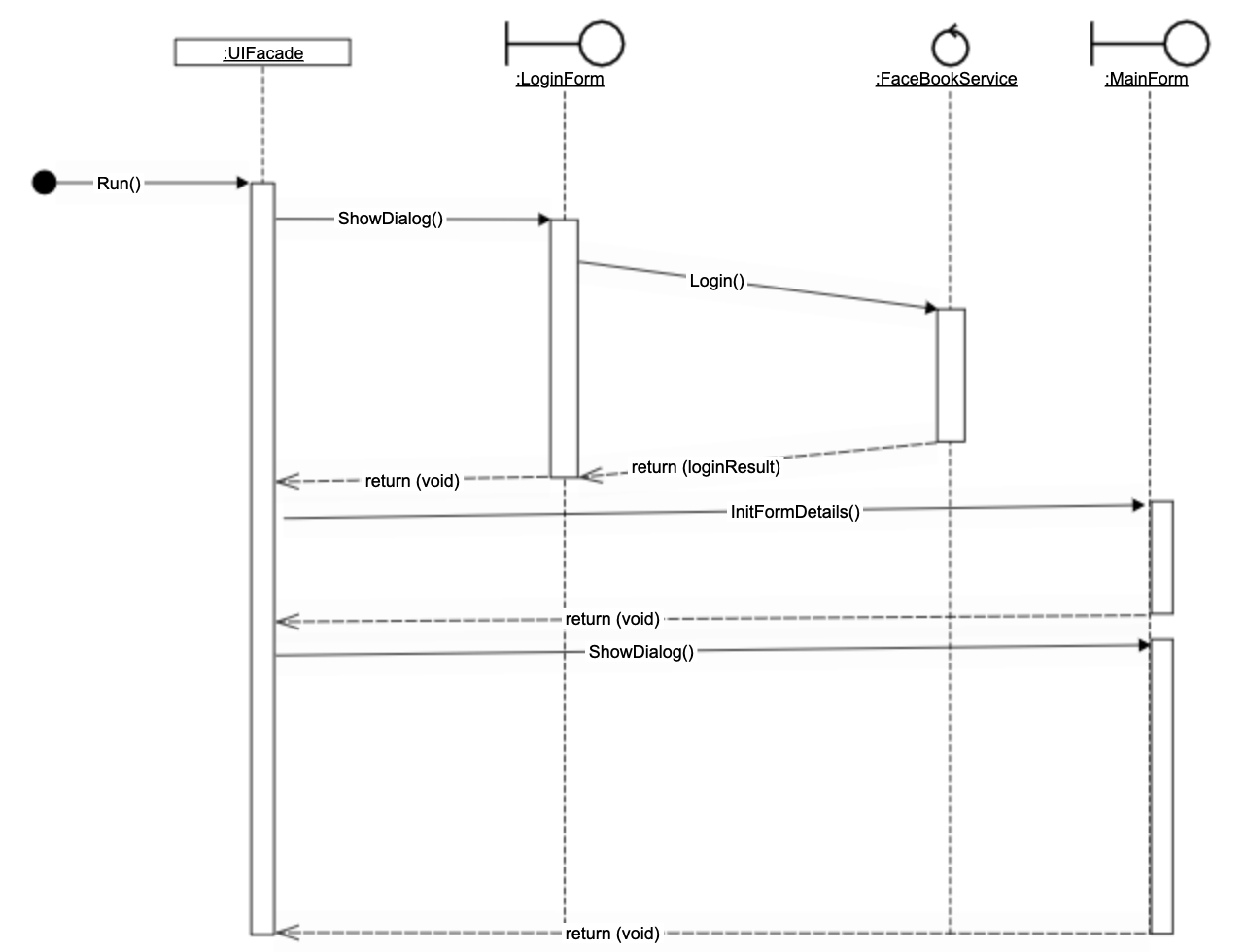
הפעלת האפליקציה מתבצעת על ידי הצגת טפסים, איתחולם בסדר מסוים וטעינה/שמירה של נתוני המשתמש בשלבים ספציפיים במהלך הריצה. על מנת להקל על ה-Client בחרנו לפשט עבורו את תהליך ההרצה ולממש מחלקה UIFacade אשר מרכזת את כך הטפסים וחושפת מתודה Run() המבצעת כל הפעולות הנ"ל ובסדר הנכון שלהן.

* אופן המימוש:

המימוש במחלקה UIFacade המכילה את מופעי הטפסים של LoginForm ו- MainForm וכן מכילה מתודה Run() הדואגת להרצה נכונה ותקינה של האפליקציה. כמו כן מטפלת בתהליך סגירה של MainForm (יציאה מהאפליקציה) ובודקת אם יש צורך בתהליך סגירה מיוחד

(Remember Me)

* Sequence Diagram



הערה: המחלקה FaceBookServiceProxy לא מצויין בדיאגרמה, מפני שרצינו להתמקד ב – Singelton.

* Class Diagram

### 

### תבנית מס' 3 – Proxy

* סיבת הבחירה / שימוש בתבנית:

רצינו להוסיף פונקציונליות נוספת למתודות Login() ו- Logout() של המחלקה FacebookService המאפשרות למשתמש באפליקציה לבקש ממנה שתזכור אותו בהתחברות הבאה על ידי סימון Remember Me בפקד ה- Checkbox שב- MainForm.

* אופן המימוש:

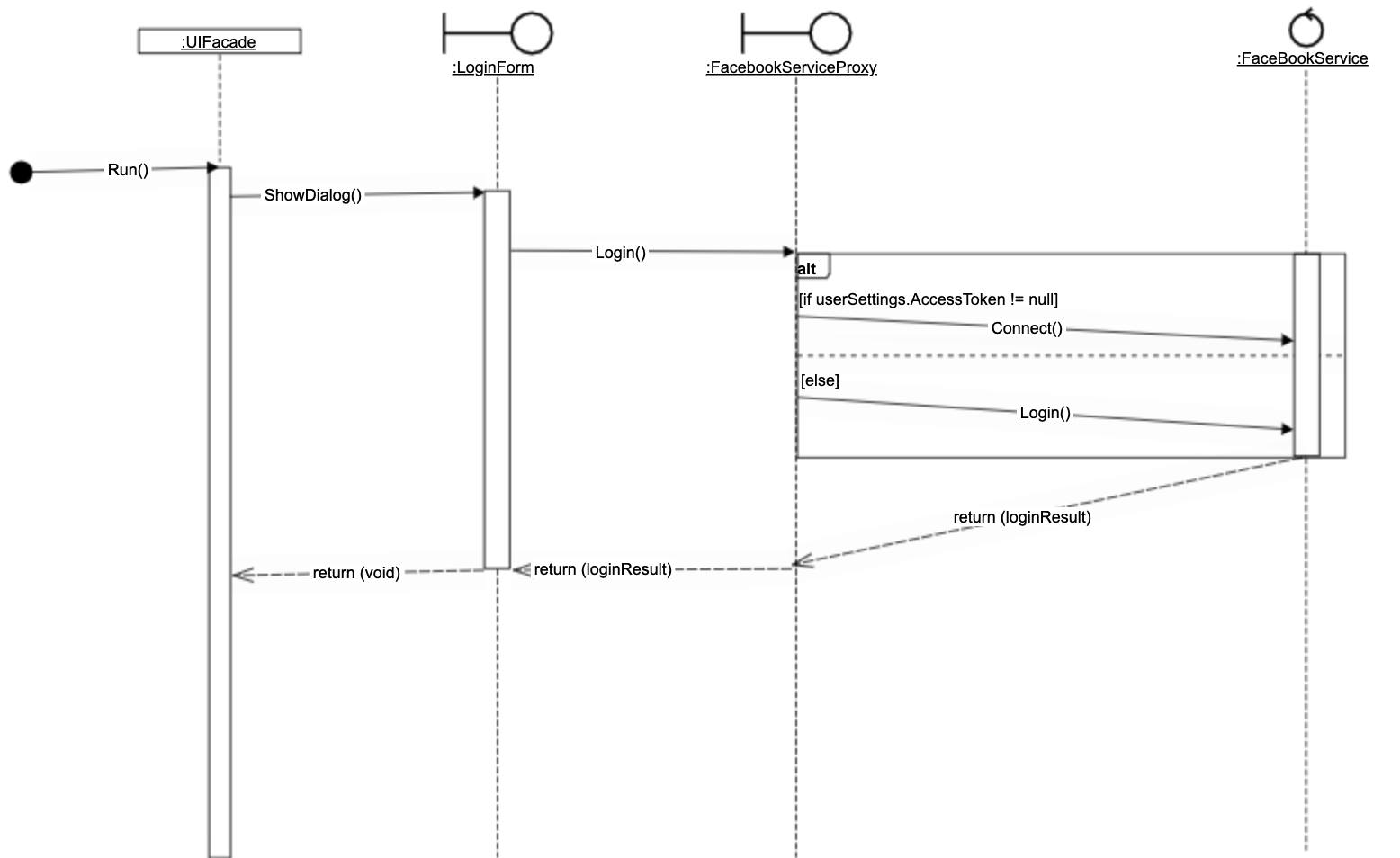
מחלקה חדשה בשם FacebookServiceProxy היא בתפקיד ה-Proxy והיא מממשת את כל המתודות של המחלקה FacebookService ובנוסף מבצעת את הצעדים הבאים:

במתודה Login() לפני ביצוע ההתחברות מתבצע ניסיון טעינה של קובץ UserDetails.xml ובעזרת XmlSerializer נוצר אובייקט מסוג UserSettings. אם אובייקט זה מכיל AccessToken אז מתבצע Login עם AccessToken זה, אחרת המשתמש הנוכחי רשאי להתחבר בעצמו.

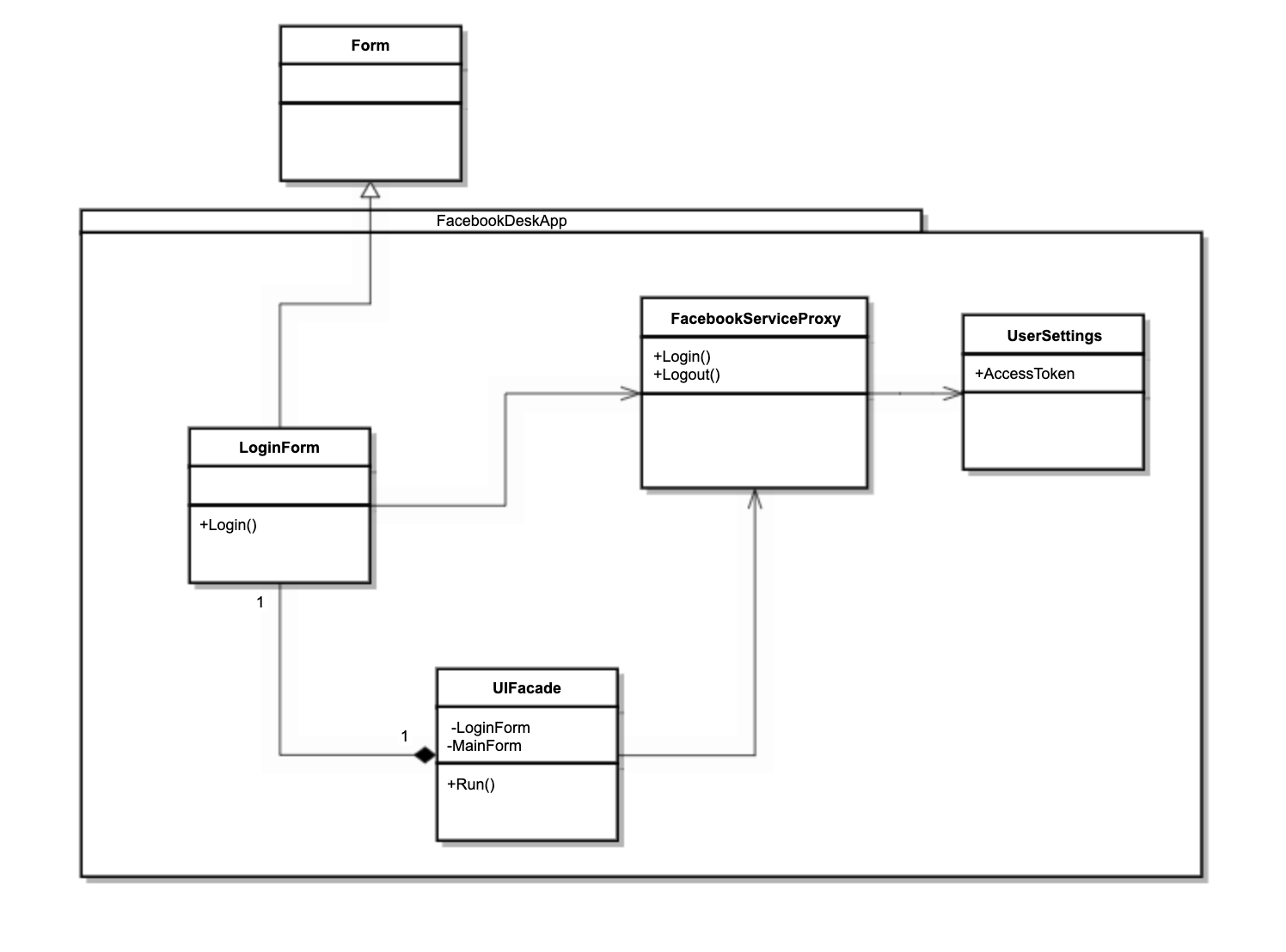
במתודה Logout() מתבצעת שמירה של נתוני המשתמש בקובץ UserDetails.xml.

אם המשתמש לא ביקש שהאפליקציה תזכור אותו- לא ישמר קובץ UserDetails.xml בסגירת האפליקציה.

* Sequence Diagram



* Class Diagram



### עבודה אסינכרונית

בעליית הטופס MainForm מתבצע איתחול שבו נתונים של המשתמש מאוכלסים בתוך פקדים בכל תתי הדפים שלו (לדוגמא מעבר על כל האלבומים והוספתם כל אחד מהם לפקד ListBox).

כאשר מספר הנתונים גדול פעולות אלו עלולות לקחת זמן ופקדים מסויימים עלולים להיות לא מוכנים לשימוש לפני שפקדים אחרים מוכנים. לכן עבור חווית המשתמש, החלטנו לפצל אחריות זו לטרדים שונים ובכך נוצרת תחושת עלייה חלקה יותר והמשתמש מרגיש פחות עיכובים עד שהטופס מוכן במלואו.

המתודות setPostsPage, setAlbumsPage, setFriendsPage, setGroupsPage מופעלות על ידי המתודה InitFormDetails של MainForm כך שכל אחת רצה בטרד שונה. הסיבה לבחירה בהן היא שזמן הריצה שלהן מושפע מכמות הנתונים (תמונות, פוסטים, חברים) שמכיל המשתמש הנוכחי, דבר שמשתנה בין משתמש אחד לאחר.

### עבודה עם Data Binding

קיים שימוש ב- DataBinding בהצגה של חלק מהנתונים בחלון MainForm.

ישנו קישור בין פוסט שנבחר על ידי המשתמש ב-ListBox לבין הפקדים שמציגים את הפרטים שלו.

ישנו קישור בין חבר שנבחר על ידי המשתמש ב-ListBox לבין הפקדים שמציגים את הפרטים שלו.

ישנו קישור בין תמונה שנבחרת על ידי המשתמש ב-ListBox לבין פקד PictureBox שמציג אותה.

ישנו קישור בין קבוצה שנחברת על ידי המשתמש ב-ListBox לבין הפקדים שמציגים את הפרטים שלה.

כלל הקישורים מתצעים במתודות הבאות:

listBoxPhotos\_SelectedIndexChanged

listBoxPosts\_SelectedIndexChanged

listBoxFriends\_SelectedIndexChanged

listBoxGroups\_SelectedIndexChanged