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

* פוסט עתידי – אופציה לעלות פוסט כעת או בתאריך ושעה עתידיים
* הורדת אלבומים מחשבים – בוחרים אלבום מהפייסבוק ומיקום במחשב והאפליקציה יוצרת תיקייה עם כל תמונות האלבום

### תבנית מס' 1 – singelton

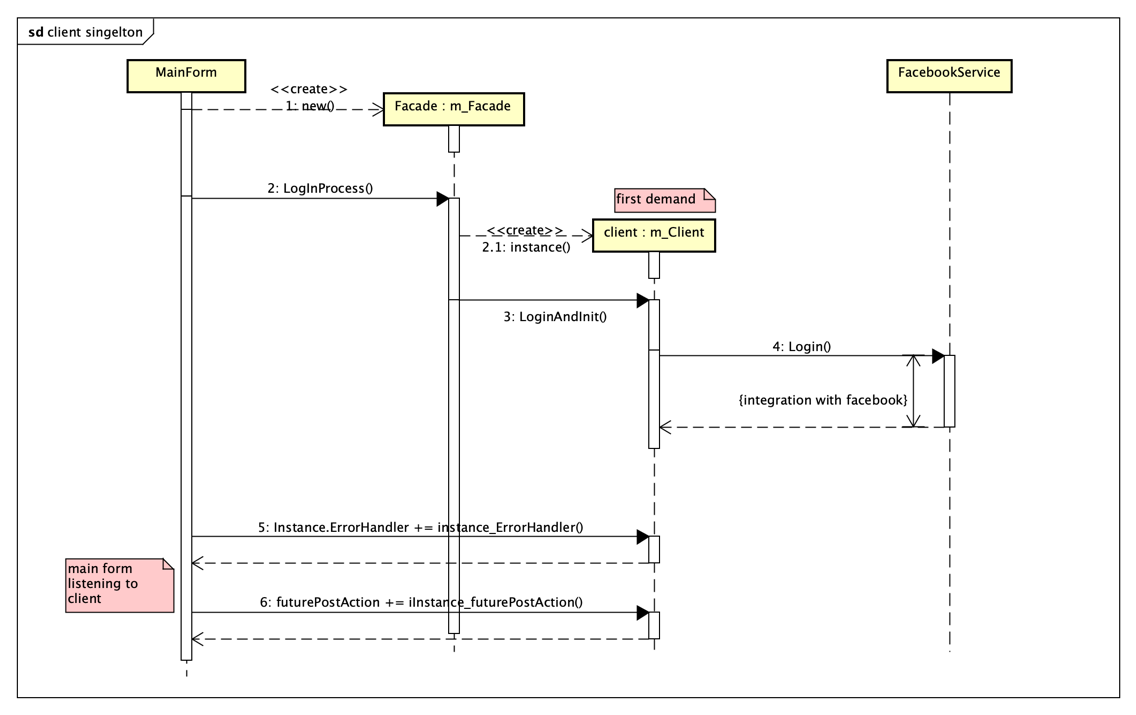
* סיבת הבחירה / שימוש בתבנית:

ראשית, יש לציין שעטפנו את המחלקות הקשורות למשתמש הפייסבוק במחלקה הנקראת client בחרנו להשתמש בsingelton מכיוון שלא אפשרי שיהיה יותר ממשתמש פייסבוק אחד ורצינו למנוע יצירה של משתמש נוסף. בנוסף רצינו שמערכות שונות באפליקציה יוכלו לגשת אליו בצורה נוחה מבלי שהclient בכלל תדע

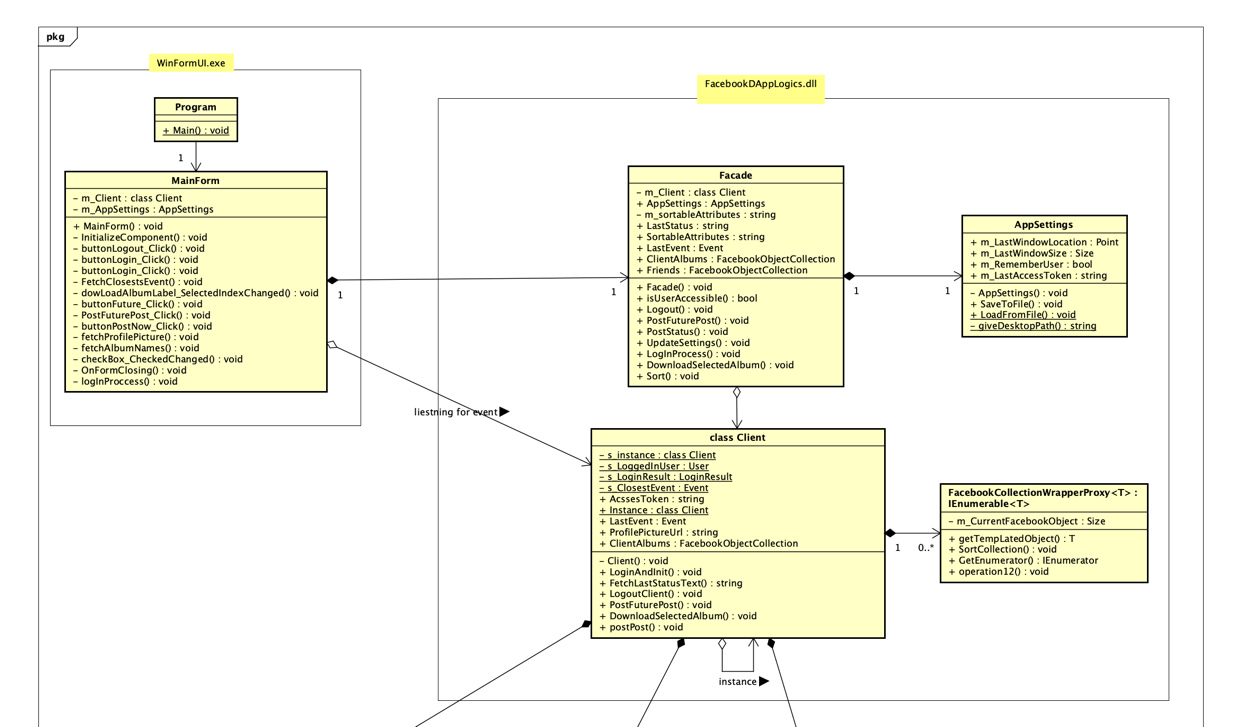
* אופן המימוש:

הסינגלטון ממומש במחלקת הclient בקובץ הdll המכיל את הלוגיקה של האפליקציה בשיטת full-Lazy and safe instantiation

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Facade

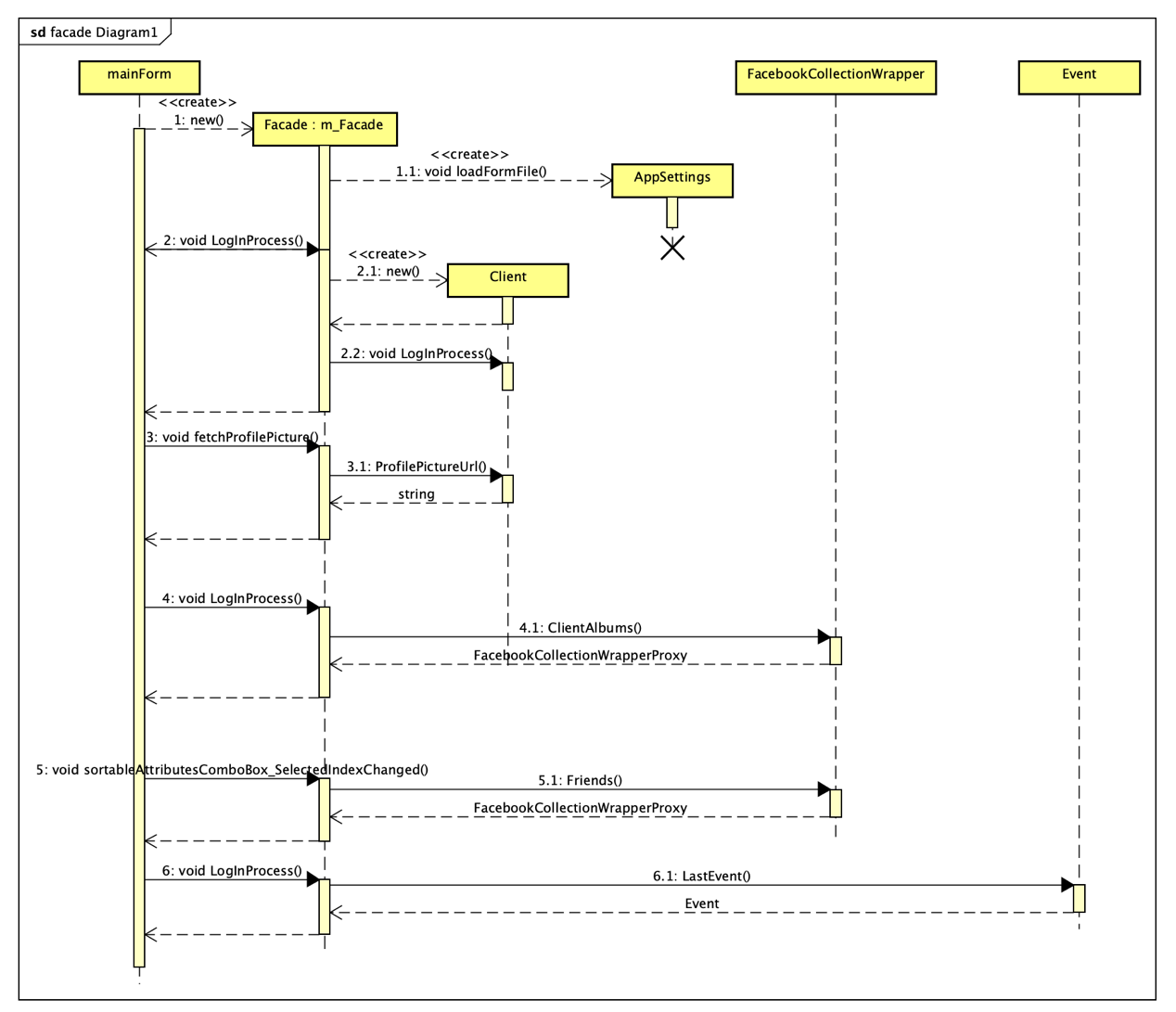
* סיבת הבחירה / שימוש בתבנית:

ייצרנו מחלקת facade על מנת לפשט את השימוש בdll של הלוגיקה ולאפשר לUI גישה ברורה יותר ללוגיקה

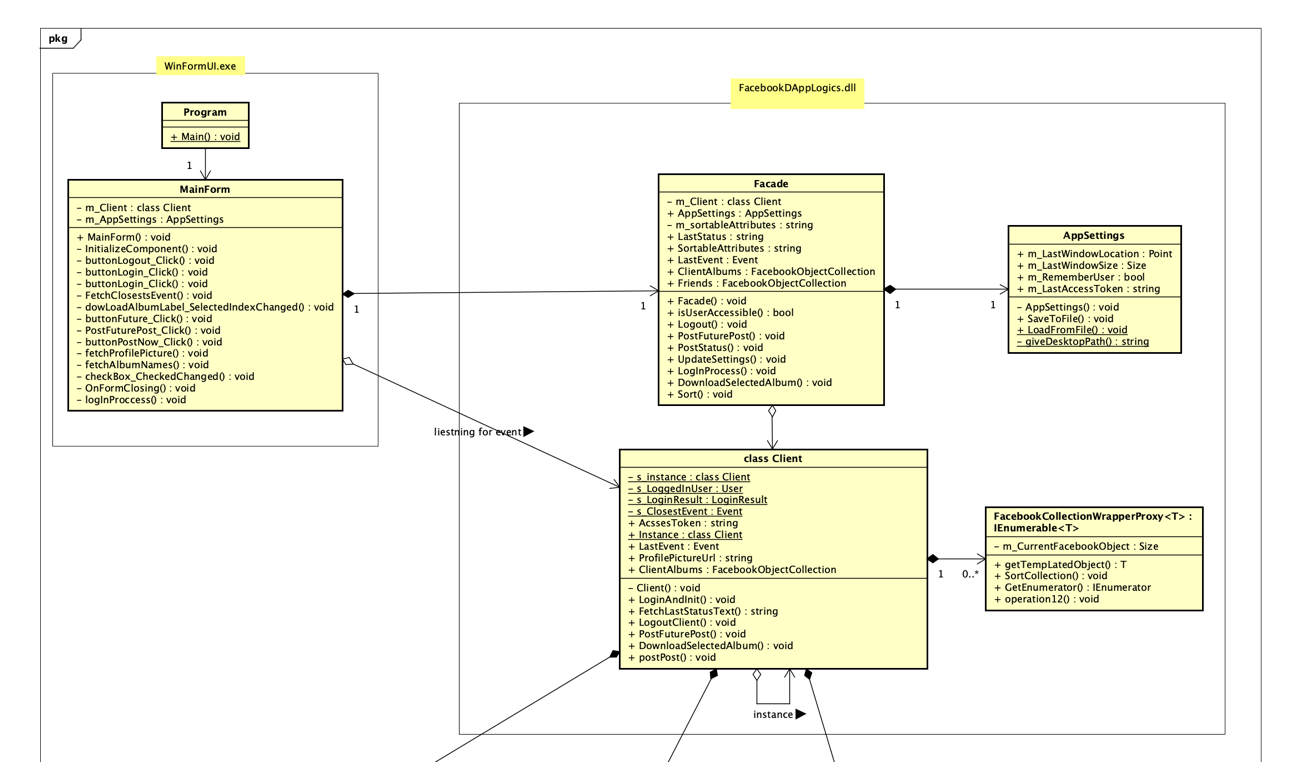
* אופן המימוש:

המחלקה עוטפת את כלל השכבה הלוגית (חוץ מגישה אינפורמטיבית בלבד למחלקות אחרת) מחזיקה properties לאובייקטים ולמידעים השונים בלוגיקה

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Proxy

* סיבת הבחירה / שימוש בתבנית:

רצינו להחליף את מחלקת הfacebookObjectColection במחלקה זהה שמאפשרת פיצר נוסף של מיון נתונים הנקראת FacebookCollectionWrapperProxy

* אופן המימוש:

ה FacebookCollectionWrapperProxyמופיע כמה פעמים בclient לדוגמא  
 בfriendlist וב ClientAlbums

* Sequence Diagram

A diagram of a diagram

Description automatically generated

* Class Diagram

A screenshot of a computer

Description automatically generated