**תיאור קצר על הפיצ'רים שעשינו בתרגיל הקודם:**

* הורדת תמונות, אשר נמצאות באלבום התמונות של הפייסבוק לוקאלי למחשב עם אופציה סינון התמונות בין האלבומים, על ידי הוספה והורדה של תמונות מאלבומים שונים.
* להראות את המזג האוויר על פי המיקום של הפייסבוק. במידה והמיקום לא יאותר, אז המיקום שיוצג הוא יהיה מיקום דיפולטיבי שהוגדר מראש לתל אביב.
* פיצ'ר שחושף את כמות השימושים בכפתורי האפליקציה, ואת כמות זמן השהות באפליקציה.

**תבניות העיצוב הנבחרות באפליקציה:**

1. **Template Method:**

**סיבת הבחירה בתבנית העיצוב**-

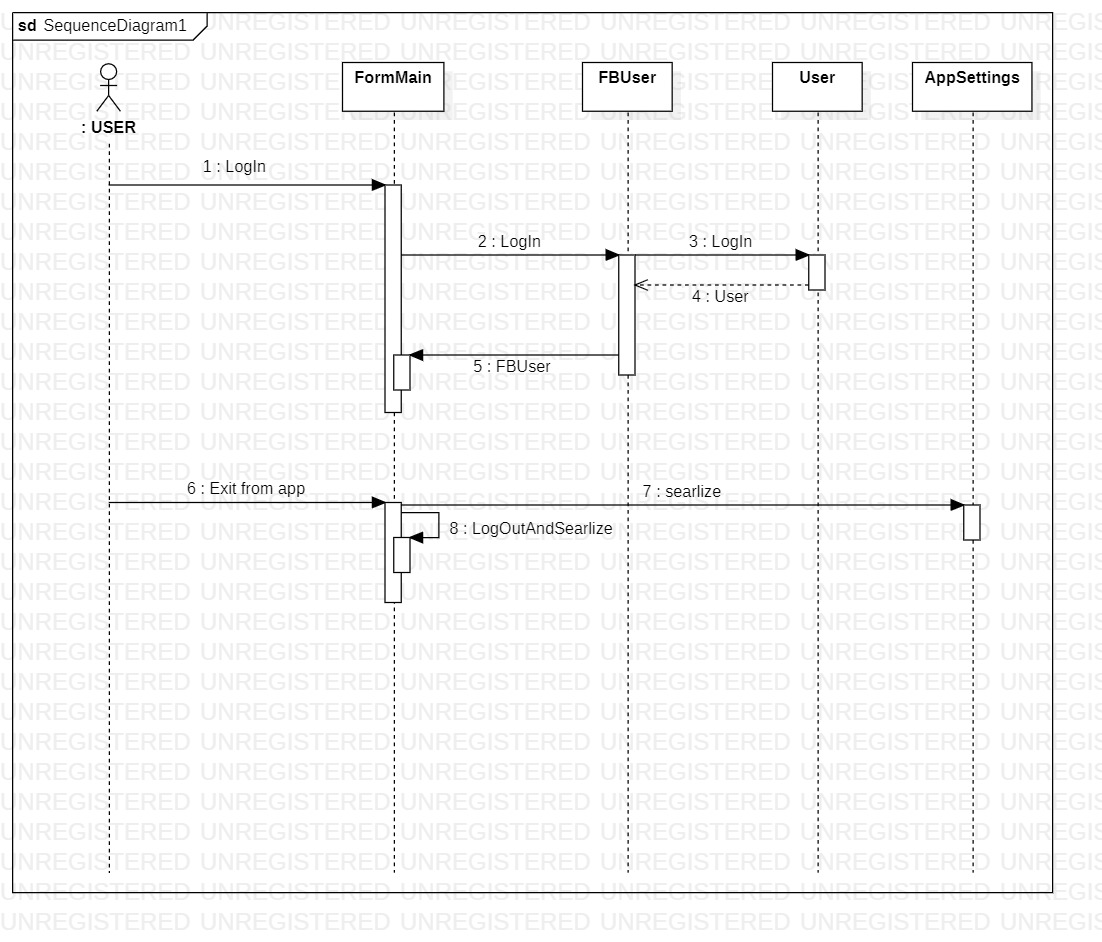
סיבת הבחירה בתבנית עיצוב זו היא בכדי לממש שלד לאלגוריתם של שמירת נתונים של מצב נתון של האפליקציה בזמן בחירת המשתמש לצאת מן האפליקציה. האפשרות של דריסת תתי פעולות אשר נמצאות בתוך האלגוריתם נותנת אופציה שונות למחלקות אחרות אשר רוצות לממש את שמירת המצב. האלגוריתם עצמו לא משתנה, אך תתי הפעולות כן משתנות וכל מחלקה אשר רוצה לשמור מצב נתון תצטרך לדרוס את תתי הפעולות.

**אופן המימוש-**

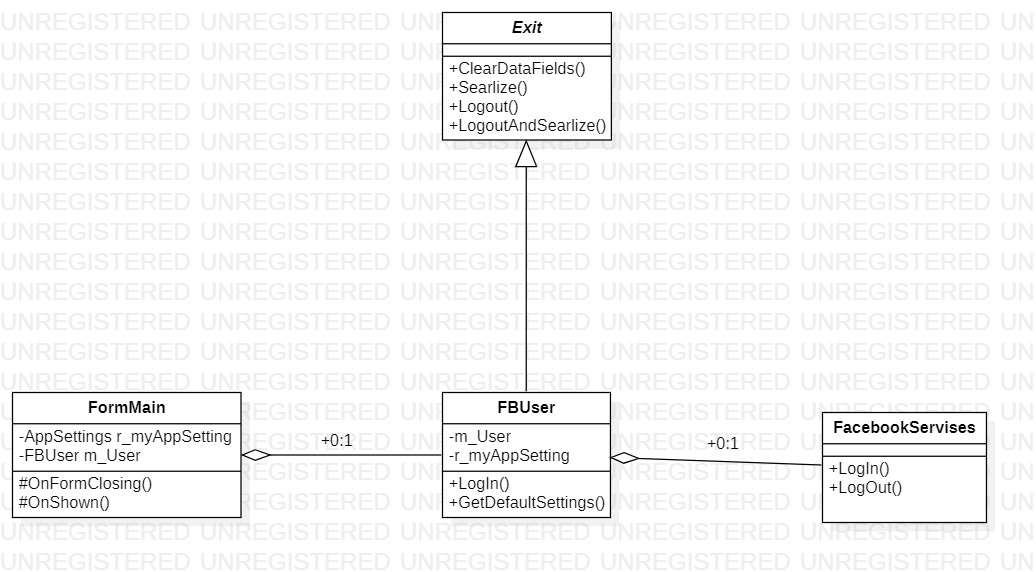
המימוש מתבצע באמצעות הגדרת מחלקה אבסטרקטית של Exit שהוא מגדיר בפנים את הפעולות האבסטרקטיות שצריך לדרוס, וגם הוא מגדיר פעולה אחת של LogOutAndSearlize שהוא הליבה של האלגוריתם. המשתמש בסה''כ צריך לקרוא לפעולה זו אשר תפעיל את כל תתי הפעולות ותעשה את הפעולה של שמירת מצב נתון באפליקציה.

השחקן שלנו הוא FBUser אשר מממש את Exit וה- client משתמש בפעולה של LogOutAndSearlize בשביל לשמור את כל המידע של האפליקציה בזמן סגירת האפליקציה, תכונה זו כאמור מקפיאה מצב נתון ומחזירה אותו לכניסה מחדש לאפליקציה.

**Sequence-diagram:**



**Class diagram:**

****

1. **Iterator:**

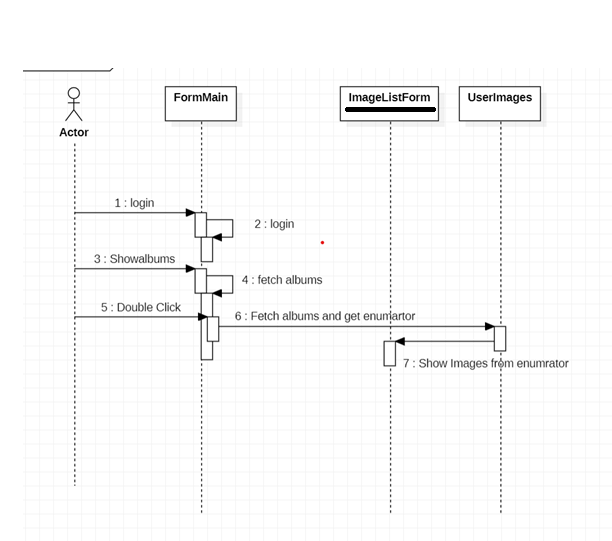
**סיבת הבחירה בתבנית העיצוב**-

סיבת הבחירה בתבנית עיצוב זאת היא בגלל שרצינו להביא אופציה למשתמש לעשות איטרציה על מבנה נתונים באופן אחיד לא משנה באיזה מבנה נתונים בחרנו לשמור את הנתונים. אנחנו רק מספקים למשתמש איטרטור למבנה הנתונים, שיוכל לעבור דרכו ולמצוא את מה שהוא מחפש בו. הסיבה העיקרית היא שאם למשל נרצה מחר לשנות את מבנה הנתונים שבו נשמרים הנתונים, זה לא משהו שאכפת למשתמש כי הוא מקבל רק את האיטרטור שמאפשר לו לגלול בו.

**אופן המימוש-**

UserImages ממש את הממשק של IEnumerable והוא דורס שתי פונקציות של GetEnumerator . אופן המימוש ב UserImages הוא באמצעות לולאת foreach וקבלת אנומרטור של ImageListFormDataBinding.

**Sequence-diagram:**



**Class diagram:**

תמונה שמכילה טקסט, תרשים, קו, תוכנית

התיאור נוצר באופן אוטומטי

3:Strategy

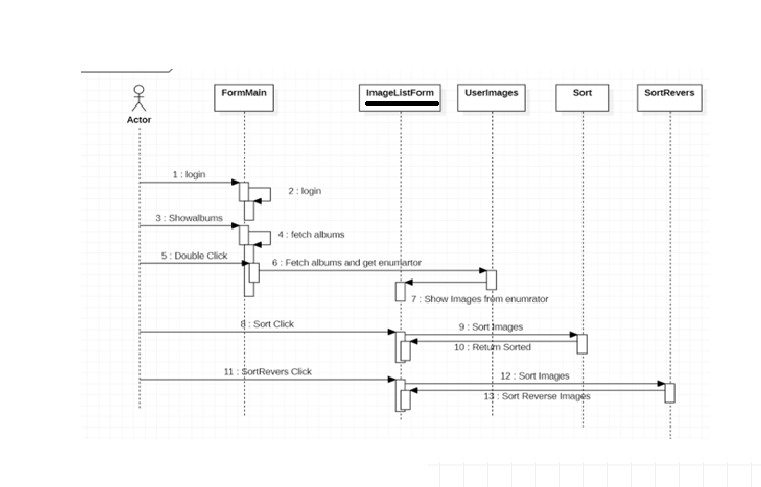
**סיבת הבחירה בתבנית העיצוב**-

סיבת הבחירה בתבנית זאת היא הבאת אופציה למשתמש בזמן ריצה החלפה של אלגוריתמים. בלי השינוי של האובייקט אלא רק את החלפה של אלגוריתמים אחד בשני.

**אופן המימוש-**

הגדרת ממשק של ISTRATEGY של הבאת אופציה לדריסה של פונקציות מי שמצבע את פונקציות הוא IMAGEFORMLIST שהוא מחליף את האלגותמים אשר דורסים את פונקציות הממשק. המימוש עצמו מתבצע באמצעות שתי מחלקות אשר ממשות את ממשק שהם SORT ו SORTREVERSE והם מחולפים בעת הצורך של המשתמש.

**Sequence-diagram:**

****



**Class diagram:**

