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

1. Like Rated:

הפיצ'ר הוא לגבי תמונות באלבומים.

אנו נציג למשתמש את התמונה הכי פופולרית שקיבלה את הכי הרבה לייקים ואת הכוזבת ביותר שקיבלה הכי פחות לייקים.

וכן את התמונה הכי מדוברת שעליה יש את מספר התגובות הרב ביותר, והתמונה עם הכי פחות תגובות.

בנוסף נציג לו סטטיסטיקה לגבי מספר הלייקים והתגובות שהוא קיבל בממוצע עבור התמונות, ואת תאריך ההעלאה שלהן.

כמו כן, ניתן בפיצ'ר לבצע סינון לפי השנים בה המשתמש העלה את התמונות לFacebook לטווח מסויים להשוות לפיו.

2. Posts Counter:

הפיצ'ר הוא לגבי פוסטים שהמשתמש שלח.

נציג למשתמש גרף בו מופיע מספר הפוסטים שהמשתמש פרסם בכל חודש בשנה, בגרף יש את חודשי השנה על ציר ה-X ועל צריך ה-Y מספר פוסטים.

ניתן לבצע סינון לפי השנה בו פורסמו הפוסטים או וגם לפי החודש שבו הם פורסמו.

בנוסף אפשר לצפות בכמות פוסטים שנתית לפי חודשים בלבד.

\*את הפיצ'רים ניתן למצוא בקוד בפרויקט UI -> Forms -> Features.

### תבנית מס' 1 – Command

* סיבת הבחירה / שימוש בתבנית:

השתמשנו ב-command design pattern כדי לעטוף את כל המידע הרלוונטי להצגת טופס חדש על המסך בתוך אובייקט יחיד בעל מטודת הפעלה פשוטה.

אם בעתיד נרצה מכמה חלונות שונים, או מדרכים שונות להפעיל את הטפסים הנ"ל, אז נוכל להשתמש בcommand של הטופס שאנו מבקשים להפעיל מבלי להתעסק בלוגיקת ההצגה של הטופס שוב.

* אופן המימוש:

יצרנו אינטרפייס ICommand שמכיל מטודת הפעלה.

את האינטרפייס יורשת מחלקה קונקרטית RelayCommand המחזיקה action להפעלה, שאותו מקבלת באיתחול המופע.

כל כפתור להצגת טופס חדש משתמש ב-command ספציפי בשבילו. וההפעלה של ה command מתבצעת בלחיצת הכפתור.

וכפתורים בעלי אותו command יכולים להימצא בכמה מקומות שונים בקוד.

\* ניתן למצוא בפרויקט UI -> תיקיית Command.

* Sequence Diagram

Diagram, schematic

Description automatically generated

* Class Diagram

Diagram

Description automatically generated

ה-Reciver במקרה שלנו הוא ה FormFactory.

\* במקרה שלנו הטופס MainForm הוא ה Invoker בדיזיין, אך אינו משתמש באגרגציה, וכן CommandManager הוא ה Client, שיוצר ושומר בו את הפקודות השונות כמחסן (repository).

### תבנית מס' 2 – Iterator

* סיבת הבחירה / שימוש בתבנית:

בחרנו להשתמש בתבנית הנ"ל מכיוון שישנו אוסף של תמונות.

* אופן המימוש:

המחלקה Photos מממשת ממשק IEnumerable<Photo>, ומחזיקה אוסף תמונות.

מימוש המטודה GetEnumerator<Photo> סה"כ מחזירה yield של האוסף.

\* ניתן למצוא בפרויקט Logic -> Photos.

* Sequence Diagram

Diagram, schematic

Description automatically generated

* Class Diagram

Diagram

Description automatically generated

### תבנית מס' 3 – Strategy

* סיבת הבחירה / שימוש בתבנית:

אם בעתיד נרצה להוסיף דרך התחברות חדשה (למשל זיהוי תווי פנים), אז ניצור סטרטג'י חדש להתחברות ונוסיף אותו לדרכי ההתחברות למערכת מבלי לשנות שום דבר בקוד הקיים.

* אופן המימוש:

יצרנו ממשק ILoginStrategy ובו מטודה TryLogin שמחזירה LoginResult.

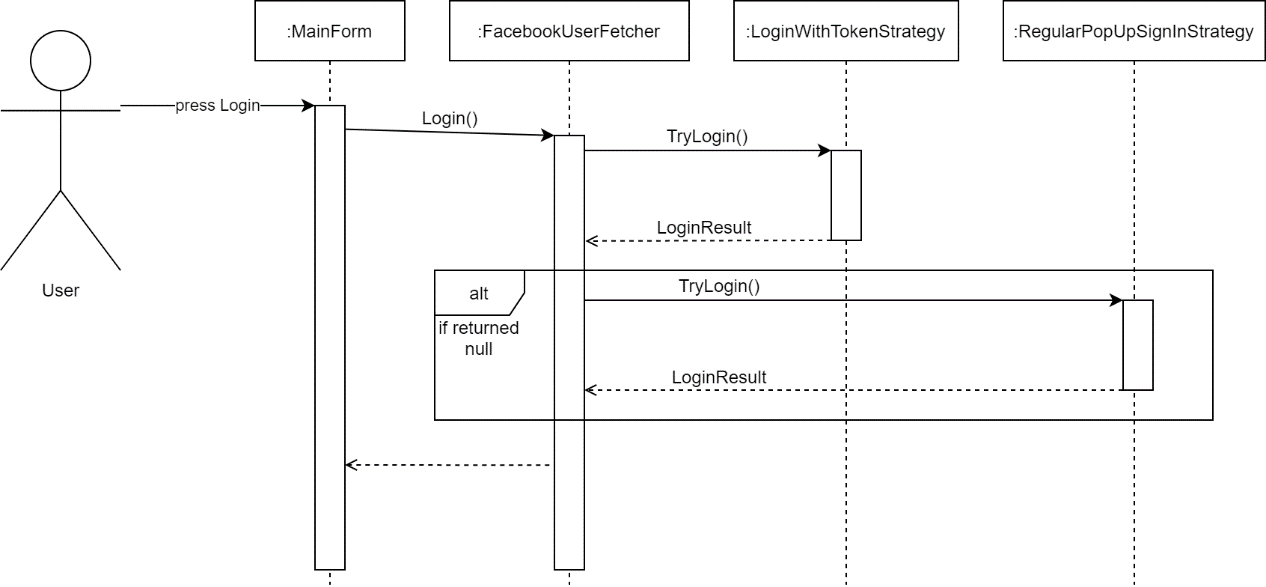
השתמשנו ברשימה List<ILoginStrategy> שכאשר נתחבר למערכת, נפעיל את דרכי ההתחברות בסדר שמבוקש כך שנוכל לקבל התחברות אוטומטית (במידה ואפשר) עוד לפני מסך ה login של שם המשתמש והסיסמא.

כל אלגוריתם התחברות מממש את הממשק ומוסיף אותו לרשימה (שנמצאת ב FacebookUserFetcher דרך מטודה Add).

\* ניתן למצוא בפרויקט Logic -> תיקיית Strategy.

וכן את הרשימה והוספת ה strategy בבנאי FacebookUserFetcher שנמאצ בפרויקט Logic.

* Sequence Diagram



* Class Diagram

