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

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

הפיצ'ר הראשון שהוספתי הוא חיפוש המעריצים הגדולים ביותר שלך.

* פיצ'ר זה נועד למוצא את שלושת המעריצים הגדולים ביותר של המשתמש, כלומר: שלושת האנשים אשר הכי פעילים בחשבון הפייסבוק של המשתמש. הן מבחינת כמות הלייקים והן מבחינת התגובות בפוסטים ובתמונות שהעלה המשתמש. קיימים 4 פרמטרים עליהם מבוסס מדד ההערצה: לייק על תמונה, תגובה על תמונה, לייק על פוסט ותגובה על פוסט. בררת המחדל היא: ניקוד של 1 עבור לייק או תגובה לתמונות ו-0 עבור לייק תגובה לפוסט. ניתן לשנות את בררת המחדר ולתת לכל פרמטר ערך בטווח בין 0-100. ניקוד זה מהווה את מידת החשיבות אשר המשתמש מייחס לכל אחד מהמאפיינים הללו, כלל שהניקוד למאפיין גבוה יותר ביחס לניקוד של המאפיינים האחרים כך הוא מקנה ערך גבוה יותר במדד הערצה.  
    
  עבור מימוש פיצ'ר זה יצרתי FORM חדש ובו גישה למחלקה בחלק הלוגי של הקוד בשם BestFan. אשר בה מחושב עבור כל משתמש שהגיב ו/או עשה לייק ולבסוף מחזיר את 3 המעריצים הגדולים ביותר של אותו היוזר כאשר ניתנת גם האופציה להמשיך ולשאול על מעריצים נוספים לפני הסדר.

הפיצ'ר השני שהוספתי הוא המראה(The Mirror) – הצגת תחומי העיינין של המשתמש באמצעות מילות מפתח.

* פיצ'ר זה נועד להציג בפני המשתמש את תחומי העניין אשר פייסבוק מאתר עבורו בהתבסס על מילות מפתח המצויות בפוסטים ובשמות התמונות שלו. בפני המשתמש ניתנת האפשרות להחליט מה הוא מספר האזכורים אשר החל ממנו יוצג התחום כתחום העניין של המשתמש. ככול שהדירוג שהזין המשתמש עבור מדד הרגישות לתחומי העניין שלו גבוהה יותר כך המסקנות יהיו יותר מדויקות יותר עבור תחומי העניין שלו.  
    
  עבור מימוש זה יצרתי FORM חדש ובו גישה למחלקה בחלק הלוגי בשם PersonalityAnalyzer אשר מחלקה זו מהווה את החלק של זיהוי תחום העניין והסקת המסקנות אשר מתממשקת עם DATABASE אשר יצרתי בשם WordsDatabase הכולל מילון עם מילות מפתח אשר ממפות לסט של מילים שמצביעות על עניין בתחום מסוים משמעויות עבור מילים אלו (עבור מילה מפתח אחת יכול להיות כמה תחומי עניין/ תפוסי התנהגות) .

יכולת חדשה שהוספתי היא ((Night Mode – הפיכת האפליקציה לחשוכה יותר לטובת סיוע לבריאות העיניים בשעות הלילה וכן הפיכה חזרה למצב רגיל.

פיצ'ר זה מאפשר למשתמש לבחור האם ברצונו לשנות את הרקע של התמונה למצב חשוך יותר וכן לבטל פעולה זו.

עובר מימוש זה יצרתי כפתור ב- MainMenuForm אשר מפעיל את מטודת invokeModePropertyChanged במחלקת ToggleNightMode אשר לה מקשיבים כלל ה- Forms וכל Form קובע כיצד ייראה מצב הלילה שלו.   
בנוסף במחלקה זו קיימת המטודה הסטטית ChangeObjectColor אשר מטרתה לפשט למתכנת (המשתמש באופצית מצב הלילה) את החלפת הרקע של Panel,Button,Form .

### תבנית מס' 1 – Observer

* סיבת הבחירה / שימוש בתבנית:

תבנית זו היא תבנית התנהגותית והיא נותנת פיתרון כאשר יש תלות של אובייקט אחד לרבים כך שאם וכאשר האובייקט האחד משנה את מצבו הוא מידע את כל המאזינים לו שמצבו השתנה וכך למעשה כל אחד מהמאזינים לו יכול לממש פעולה שונה לאותה שינו מצב שקרה.  
זאת כאשר אובייקט אחד יודע כי קיימים אובייקטים אשר מאזינים לו מבלי להניח הנחות לגבי אותם אובייקטים ואודות כמותם ובכך אנו מונעים צימוד בינהם.  
במקרה שלנו שימוש זה טבעי ל- Night Mode שיצרתי אשר אנו צריכים לאפשר בעת לחיצה על כפתור זה שכל מופע שקיים כרגע מול המשתמש וצריך להכירו ולדעת שעליו להשתנות למצב לילה או לבטל מצב זה. כאשר כל אובייקט אשר מקשיב לאובייקט PropertyChanged שקיים במחלקת ToggleNightMode ישנה את מצבו בהתאם להגדרות אשר נקבעו על ידי המשתמש בתבנית זו בין אם זה Button Panel , או .Form כרגע זה רק עניין של שינוי צבעים שונים לרקע אך בעתיד תבנית זו תאפשר מגוון רחב של אפשרויות כאשר כל אובייקט יכול להחליט כיצד עליו להגיב, במקרה זה שינינו את צבע הרקע של האובייקט.

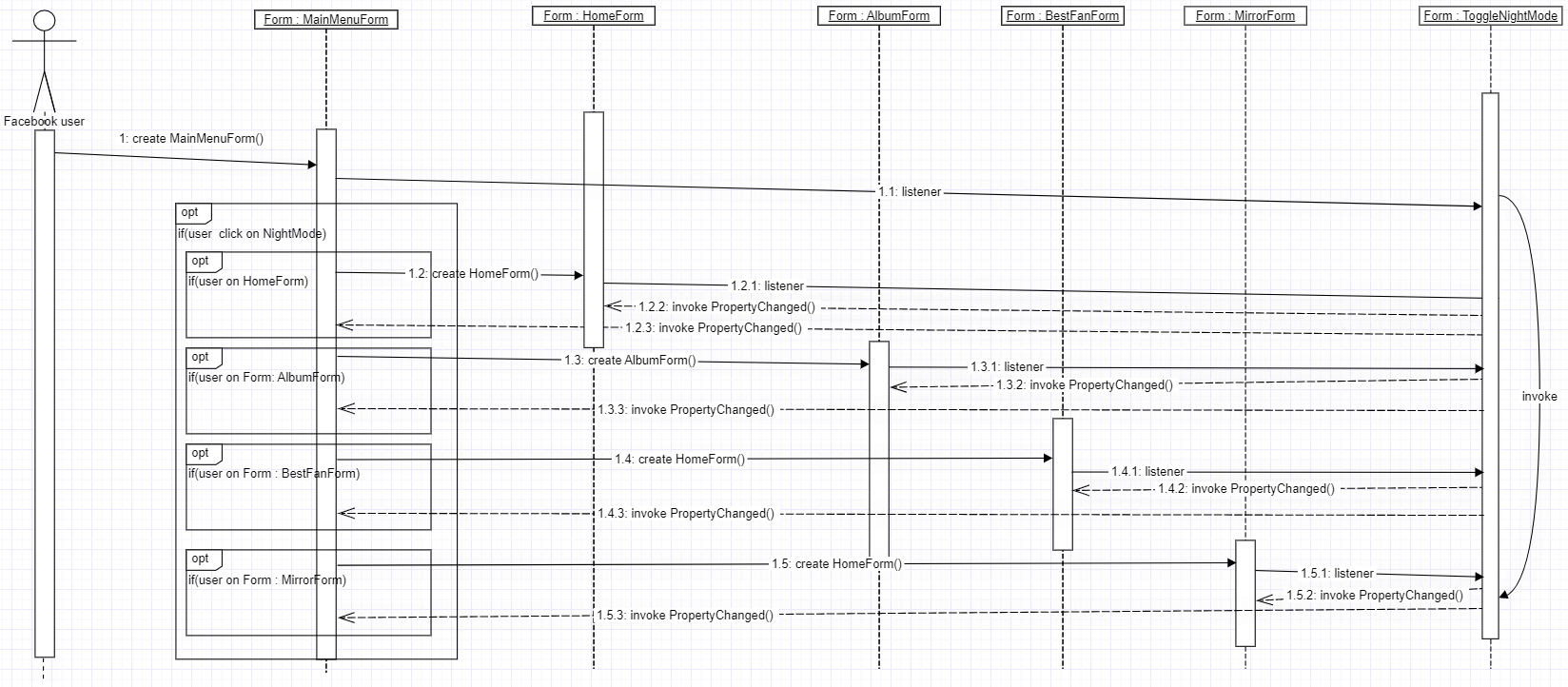
* אופן המימוש:

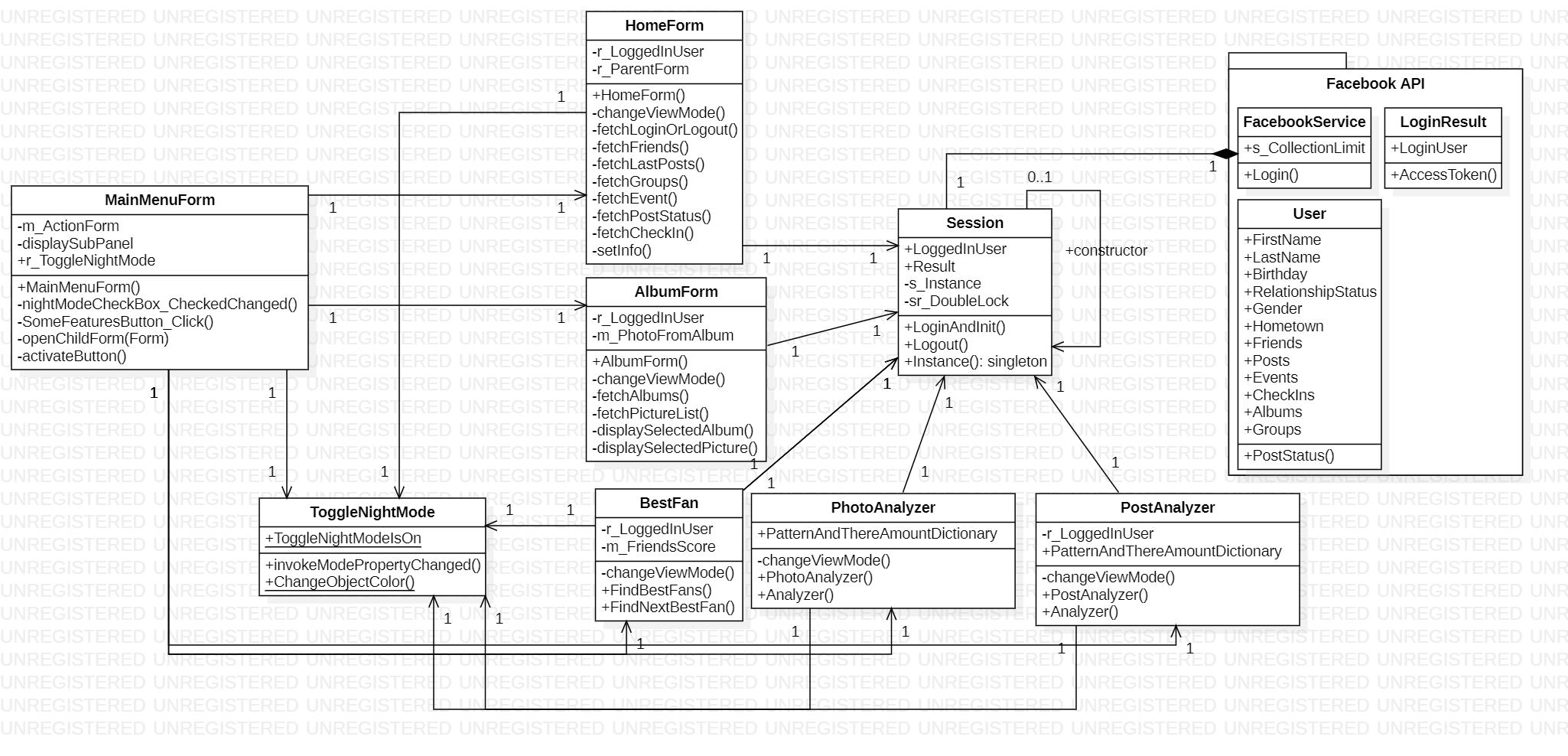
תבנית זו מייושמת במחלקתToggleNightMode אשר מציעה למשתמש בה 2 אפשרויות:

1. להאזין לשינוי שמתבצע לאובייקט ModePropertyChanged
2. פונקציה סטטית בשם: ChangeObjectColor אשר תפקידה לפשט את פעולת החלפת הצבע לאותם אובייקטים אשר הגדיר לו יוצר ה- Form .

האובייקט של מחלקה זו מוחזק במחלקת MainMenuForm בשם: r\_ToggleNightMode ומועבר לכל Form חדש שנוצר אשר מבצעה האזנה אליו על מנת לדעת האם פעולת ה- Toggle בוצעה ובמידה וכן עליו לשנות את צבע הרקע של אותו ה- Panel,Button,Form.   
פעולת ה- invoke מבוצעת כאשר המשתמש לוחץ על כפתור ה- NightMode ונקראת למטודה nightModeCheckBox\_CheckedChanged אשר קוראת למטודת invokeModePropertyChanged שמבצע את פעולת ה- invoke לאובייקט ModePropertyChanged אשר מוחזק במחלקתToggleNightMode ולאחר מכן כלל המאזינים לאותה אובייקט מבצעים את תפקידים בהתאם להגדרתם.

* Sequence Diagram



* Class Diagram

הקבלה ל- Observer:

listeners: MainMenuForm, HomeForm, AlbumForm, BestFanForm, MirrorForm  
Creator: public class ToggleNightMode  
Invoke calls: MainMenuForm

### תבנית מס' 2 – Strategy

* סיבת הבחירה / שימוש בתבנית:

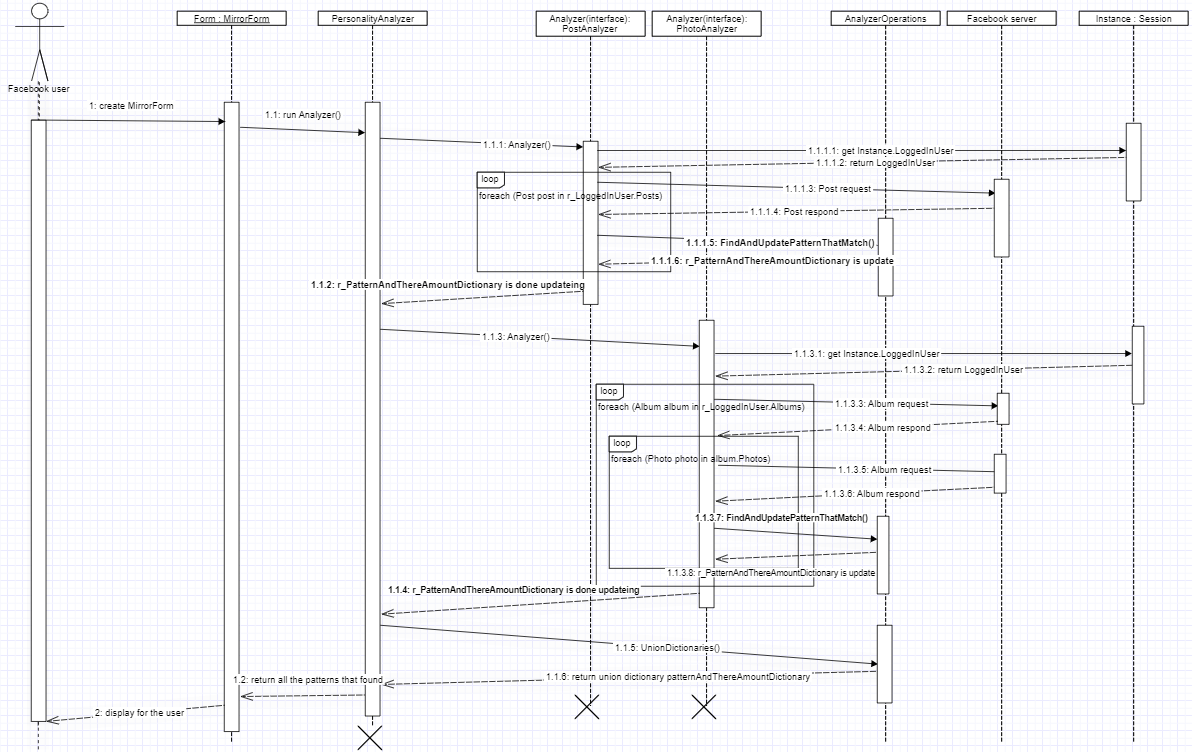
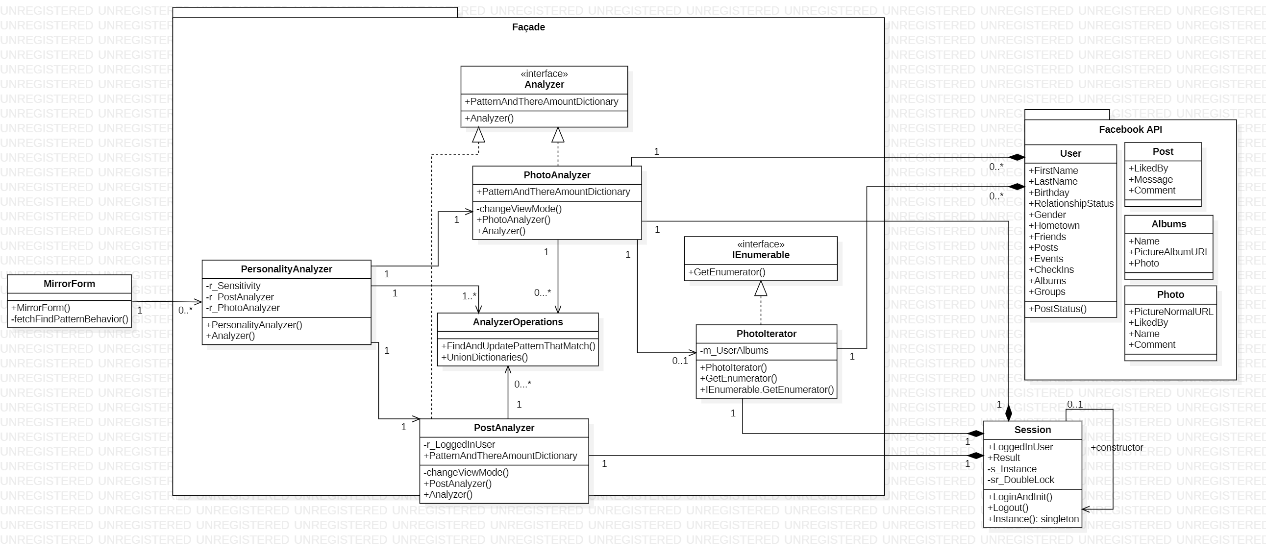
תבנית זו היא תבנית התנהגותית והיא באה לתת פיתרון כאשר האלגוריתם משתמש במידע שאיננו רוצים שיהיה חשוף ללקוח. נוסף על כך תבנית זו מאפשרת לנו לבטל סדרה של תנאים לפעולות ובמקום לבצע תנאים רבים ניתן לפשט זאת בעזרת תבנית זו. נוסף על כך תבנית זו מאפשרת לנו לשנות את האלגוריתם ללא תלות במשתמשי המחלקה.

מסיבות אלו שימוש במחלקה זו תורם לפעולת ה-Analyzer שלי בכך שפעולות של מחלקות אלו אינן נחשפות בפני המשתמשים במחלקה מכיוון שאין אנו רוצים לחשוף פעולות ואלגוריתמים מסויימים. בנוסף בעזרת תפיסה זו אנו מאפשרים הוספה קלה של Analyzer-ים נוספים עתידיים אשר יכולים לשמש אותנו, זאת מבלי להוסיף תנאים רבים כאשר כל מחלקה שמשתמשת ב- interface של Analyzer בעלת התנהגות שונה וכך בדרך זו נוכל לעבוד בקלות עם מספר רב של התנהגויות זאת תוך כדי אי תלות במשתמש כיוון שהוא חשוף רק ל- interface לכן תבנית זו יושמה על פעול ה-Analyzer אשר הלקוח שלה הוא מחלקת PersonalityAnalyzer

* אופן המימוש:

מימשתי את תבנית זו בתוך Ex03.Logic.Facadeבמחלקת Analyzer אשר היא מהווה את ה- interface של התבנית ואת המחלקות PhotoAnalyzer ו- PostAnalyzer בתור ה-ConcreteStrategy.

כאשר ב- interface Analyzer קיים הצורך ליישם את שתי המטודות הבאות Analyzer ו- PatternAndThereAmountDictionary אשר מיושמות במחלקות PhotoAnalyzer ו- PostAnalyzer והלקוח של תבנית זו היא מחלקת PersonalityAnalyzer אשר משתמשת בשני המטודות של Analyzer.

* Sequence Diagram
* Class Diagram

הקבלה ל- Strategy:

Client: PersonalityAnalyzer  
Interface: public interface Analyzer  
Implements: PhotoAnalyzer, PostAnalyzer

### תבנית מס' 3 –Iterator

* סיבת הבחירה / שימוש בתבנית:

תבנית זו היא תבנית התנהגותית והיא מאפשרת לנו ליצור תבנית שבה נוכל לגשת לאיברי קבוצה הנתונה מבלי לדעת מהו מבנה הנתונים ומבלי לחשוף את אותו המבנה אשר מאחסן את איברי הקבוצה. זאת בעזרת הגדרה מראש של סט פעולות אפשריות. בנוסף תבנית זו מאפשרת ממשק אחיד לעבודה עם מבני נתונים אשר מכילים בתוכם רצף של אובייקטים ובמידה ובעתיד מבנה נתונים זה ישתנה אין הלקוח צריך להכיר בשינוי זה.

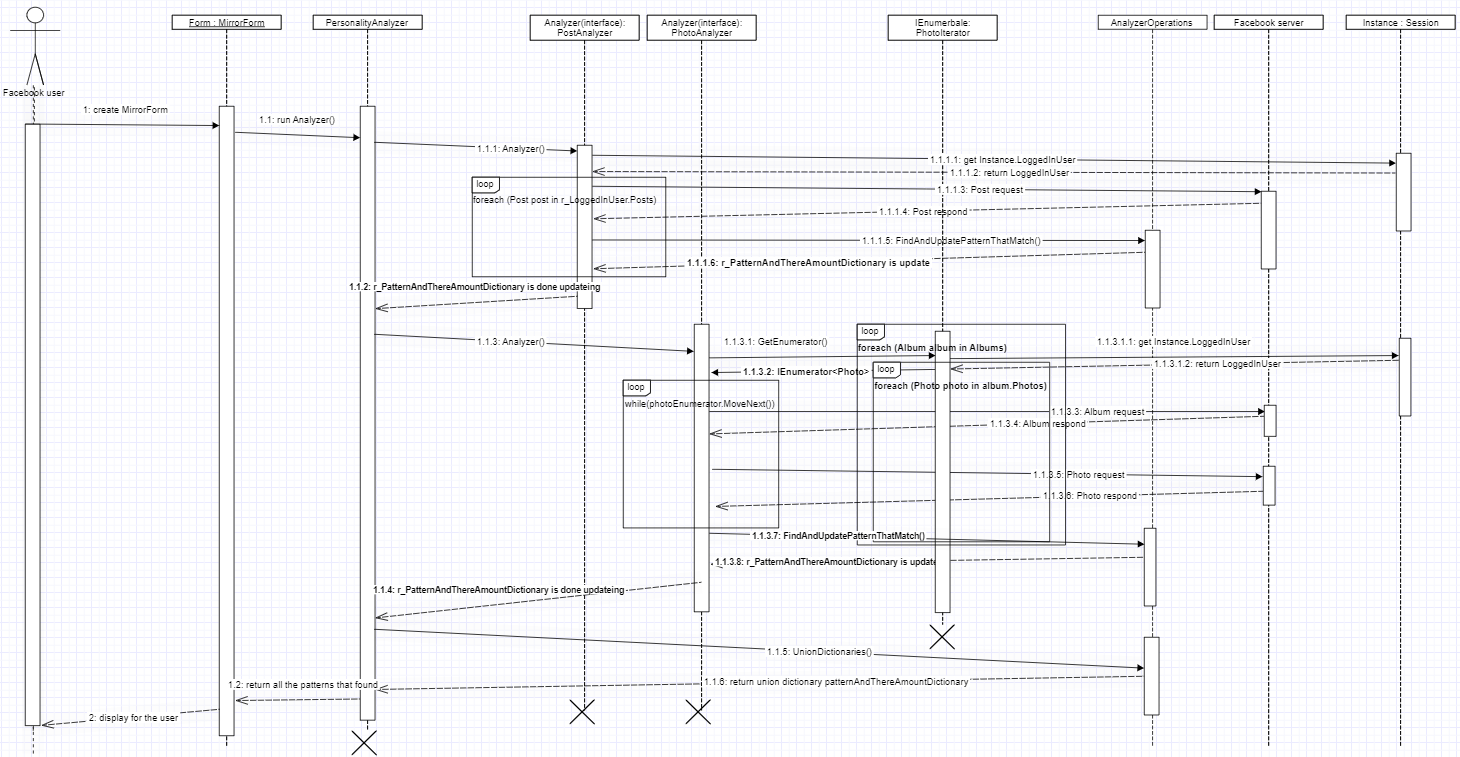
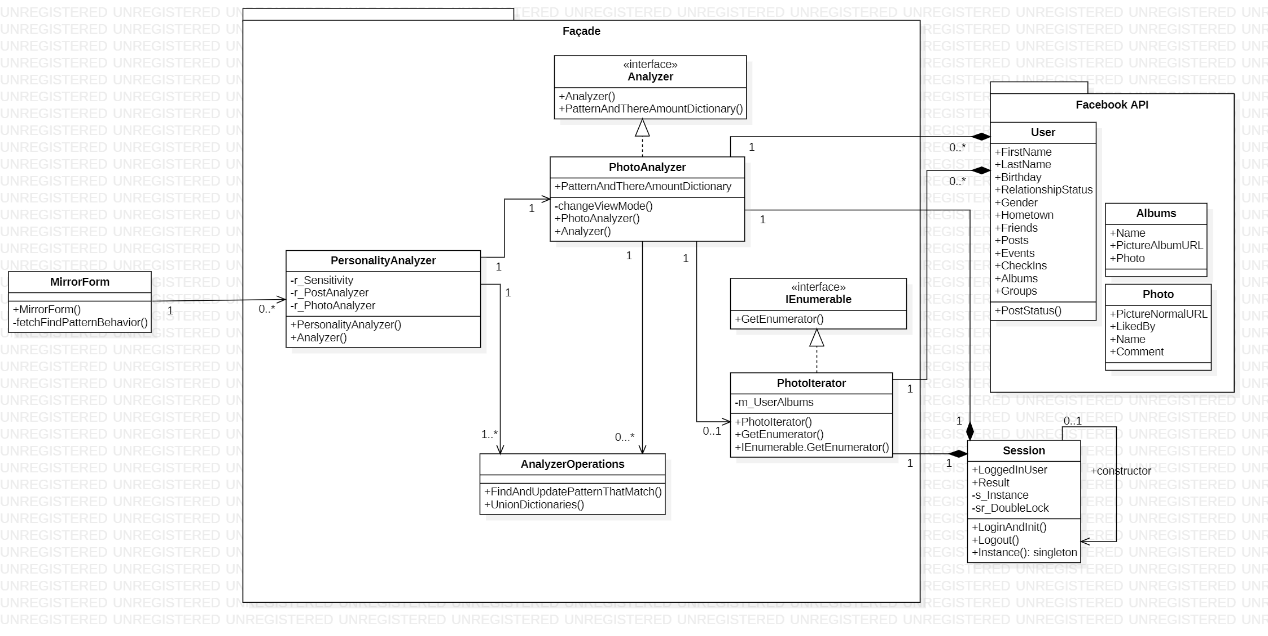
תבנית זו משמשת אותנו במעבר על תמונות מתוך אלבומים, זאת מבלי לדעת מהו מבנה הנתונים הנתון אשר מחזיק את האלבומים והתמונות. תבנית זו מאפשרת לנו גישה לתמונות מתוך האלבומים בעזרת ממשק אחיד ופשוט תוך כדי אפשרות עתידית לשינוי מבנה הנתונים הקיים ללא צורך כלל בשינוי בקוד.

* אופן המימוש:

את תבנית זו יצרנו במחלקת PhotoIterator ב-Ex03.Logic.Facade

והיא מיישמת את מחלקת IEnumerable עם IEnumerable<Photo> כאשר מחלקת PhotoAnalyzer (אשר היא הלקוח) תפנה ל- PhotoIterator על מנת לעבור על התמונות מתוך האלבומים באמצעות איטרטור ומחזירה את התוצאה בעזרת yield return .

בעזרת האיטרטור משתמש PhotoAnalyzer(המשתמש) בממשק אחיד ופשוט של MoveNext ו- Current כל זאת על מנת לנתח את התמונות. מימוש זה מאפשר לנו עתידית לשנות את מבנה הנתונים באופן חופשי זאת ללא חשיבות עדכון המשתמש על שינוי מבנה נתונים זה ומבלי לחשוף את מבנה הנתונים.

* Sequence Diagram
* Class Diagram

הקבלה ל- Iterator:

Client: PhotoAnalyzer  
Creator: public class PhotoIterator  
Implements: IEnumerable<Photo>