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

* הצגת חמשת התמונות של המשתמש שקיבלו הכי הרבה לייקים.
* הצגת רשימה של אנשים מפורסמים שנולדו באותו תאריך (בלי שנה) עם המשתמש.

תוספת:

* הצגת חמשת התמונות של המשתמש שקיבלו הכי הרבה תגובות.

### תבנית מס' 1 – Strategy

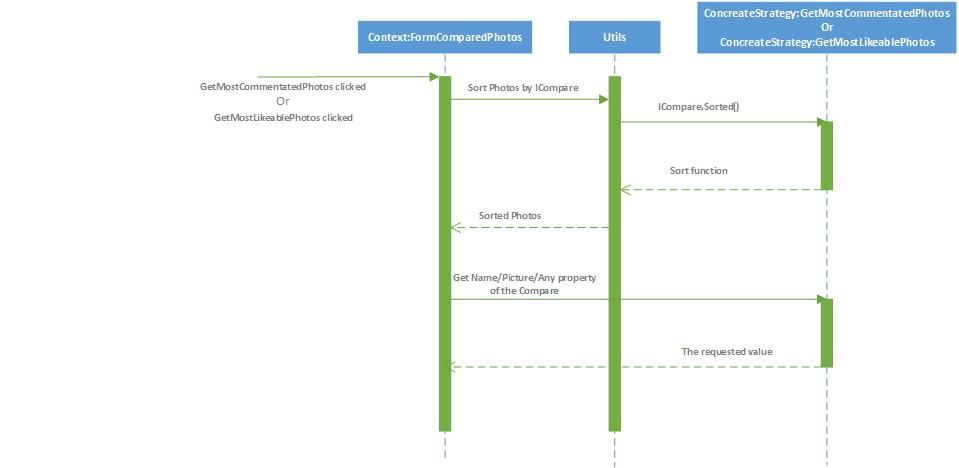
* סיבת הבחירה / שימוש בתבנית:

במערכת יש קוד אלגוריתמי של השוואה בין אובייקטים על מנת לייצר את הפיצ'ר של חמשת התמונות (או עם הכי הרבה לייקים או הכי הרבה תגובות). לכן, ייצרנו Strategy שבעזרתו אנחנו יכולים להוסיף את ההשוואות לפי החלטת המשתמש.

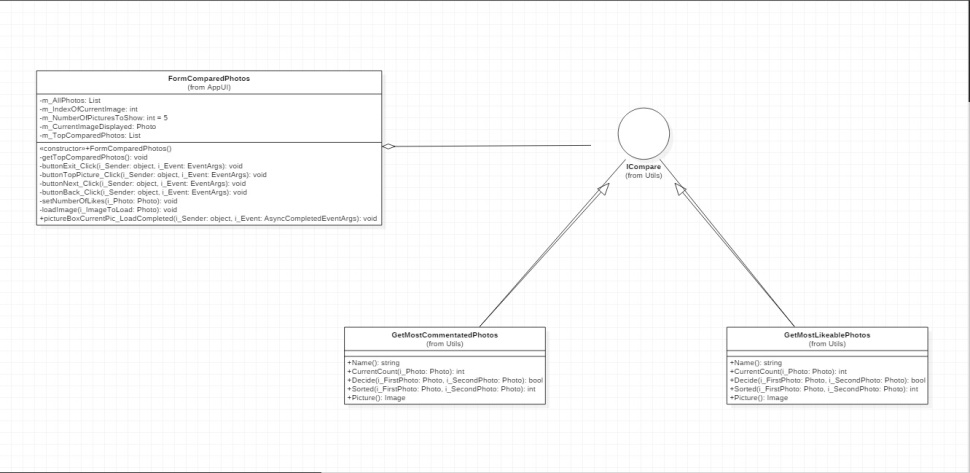
* אופן המימוש:

הוספנו ל- FormFb הכללי שלנו החזקת רפרנס ל interface שנקרא ICompare. כך עבור הטופס הספציפי FormComparedPhotos יש לנו את הרפרנס הרלוונטי. ב- interface שלנו ICompare יש את כל החתימות של המתודות הרלוונטיות בהן נעשה שימוש בתוכן הקשור לאובייקט ההשוואה. עבור כל לוגיקת השוואה ייצרנו מחלקה הממשת את ICompare. יש שתיים כאלה- GetMostCommentatedPhotos ו- GetMostLikeablePhotos. בכל אחת ממומשת לוגיקת ההשוואה הנדרשת ובנוסף ממומשות מספר תוספות הנחוצות במהלך השימוש ב- ICompare.

* Sequence Diagram



* Class Diagram



FormComparedPhotos - מקביל ל- Context ב- Pattern.

ICompare - מקביל ל- Strategy ב- Pattern.

המחלקות - GetMostCommentatedPhotos ו- GetMostLikeablePhotos הן ה- ConcreateStrategy ב- Pattern.

### תבנית מס' 2 – Template method

* סיבת הבחירה / שימוש בתבנית:

נרצה לשמור על תבנית אחידה למימוש כפתורים באפליקציה שלנו כאשר לכל כפתור עיצוב ולוגיקת ייצור דומים. השוני בין הכפתורים יבוא לידי ביטוי בהתנהגויות ספציפיות של כל כפתור.

* אופן המימוש:

כתבנו מחלקה אבסטרקטית שנקראית ButtonFBTemplate – היא מכילה תבנית כללית של יצירת כפתור facebook כפי שהיינו רוצים שיראה ע"י מימוש מתודות כלליות (יצירת רקע, עיצוב גנרי, נוכל להכניס פה דברים נוספים כאוות נפשנו מבלי לגשת לכל כפתור בנפרד). המתודה האבסטרקטית תהיה setUniqueProperties – כאשר נרצה להוסיף התנהגות ייחודית לכל כפתור.

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### תבנית מס' 3 – Command

* סיבת הבחירה / שימוש בתבנית:

נרצה לממש תפריט UI כלשהו עבור הפיצ'רים שבנינו. התפריט לבחירתנו יהיה כפתורי פייסבוק שהוגדרו מראש ובתגובה ללחיצה על כל אחד מהם, תופעל פונקצית תגובה מתאימה.

* אופן המימוש:

הוגדר ממשק שנקרא ICommand שמכיל פונקציה אחת Execute(). כל פונקצית תגובה שנרצה, תממש את הממשק הנ"ל והיא בעצם תפעיל את הפקודה שתקבל מראש ב constructor שלה (לדוגמא המחלקה WhoWasBornOnMyBirthdayCommand). במחלקה FeatureReciever מוגדרות הפקודות שנרצה לממש ושם בעצם אנחנו עושים את ההצמדה למחלקות שמימשו את הממשק ICommand. (לדוגמא, נפעיל את ה constructor של WhoWasBornOnMyBirthdayCommand במחלקה הזאת וכך נגדיר לה את הפעולה שהיא בעצם צריכה לעשות). כעת, על מנת להפעיל את פונקצית התגובה, סה"כ נבצע פנייה ישירה באופן הבא r\_FeatureReceiver. WhoWasBornOnMyBirthday.Execute().

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם