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

* רשימה שמסננת את העמודים שהמשתמש אהב ומראה רק עמודים שלא היו פעילים מספר מסויים של ימים (בין 1 ל-60, לבחירת המשתמש) במטרה להיות מועמדים למחיקה על ידי המשתמש (ניקוי של עמודים לא פעילים מהפרופיל).
* חלון עריכת תמונה אשר מאפשר שימוש בפילטרים גרפיים מובנים ויצירת פילטרים גרפיים חדשים, שמירתם ושימוש חוזר שלהם על תמונות נוספות. אפשרות העלאה של התמונה שנוצרה לפרופיל בפייסבוק (בפרטיות Only Me).

### תבנית מס' 1 – [שם התבנית]

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* 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' 2 – Strategy

* סיבת הבחירה / שימוש בתבנית:

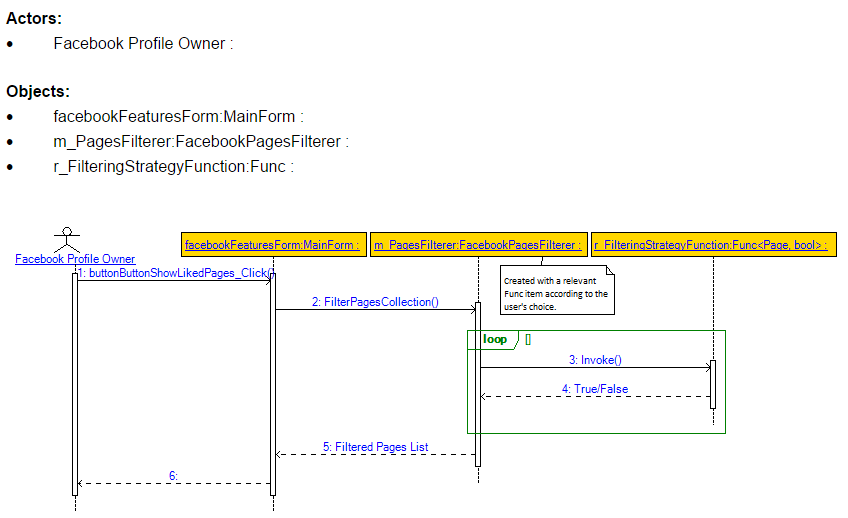
משתמש פייסבוק לעתים ירצה לראות אילו עמודים שהוא "אהב" כדאי לו למחוק מהפייסבוק. לשם כך הוא ירצה לראות רק את חלקם בהתאם לפילטר מסויים. לדוגמה: פילטר שמראה רק את העמודים הלא פעילים, או פילטר המראה רק את העמודים בשפה מסויימת, או פילטר שמראה רק את העמודים שיש להם פחות ממס' לייקים מסויים וכו'. זאת במטרה לדעת אילו עמודים כדאי לו (לפי בחירתו) למחוק.

* אופן המימוש:

המימוש נעשה בעזרת מחלקה בשם FacebookPagesFilterer שה-MainForm מחזיק ממנה אינסטנס. במחלקה יש פונקציה שיודעת לעבור על מקבץ של דפי פייסבוק, ולהחזיר מקבץ חדש רק של הדפים שעברו את הפילטר בהצלחה. את המחלקה יוצרים עם הפונקציה שתהווה את האסטרטגיה של הפילטור (בתור פרמטר - Delegate). נכון לכרגע בקוד יש 4 אסטרטגיות פילטור שונות שאיתן המחלקה יכולה להיווצר.

הקונטקסט הוא ה MainForm-שמשתמש ב-Strategy שהוא FacebookPagesFilterer, הנוצר עם פונקציה ש-MainForm שולח לו כאסטרטגיה שבה הוא ישתמש. האינסטנס של FacebookPagesFilterer שיש בו פונקציה מסויימת, הוא ConcreteStrategy.

* Sequence Diagram



* 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 – Visitor

* סיבת הבחירה / שימוש בתבנית:

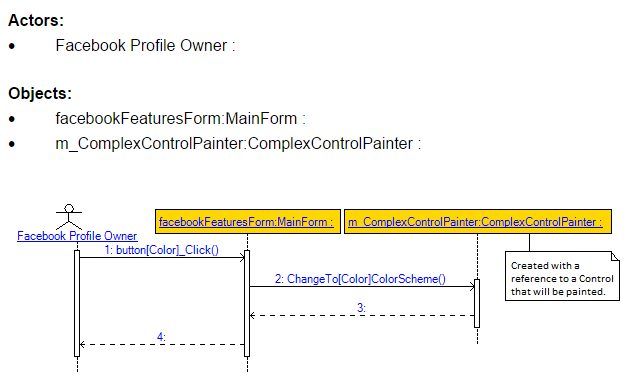
אנו רוצים להנעים את השימוש באפליקציה עבור המשתמש ולכן מאפשרים צביעה של כל אחד מהחלונות באפליקציה בנפרד בסכמת צבעים אחת מתוך 4 אפשרויות.

* אופן המימוש:

ה-Visitor הוא המחלקה ComplexControlPainter. הוא מקבל אובייקט קונטרול ויודע לצבוע את חלקיו השונים לפי סכמת צבעים קבועה. יש שימוש באובייקט של מחלקה זאת בכל אחד מהחלונות שקיימים באפליקציה, ולכן הלוגיקה של צביעת החלון יצאה מחוץ לטופס ספציפי אחד.

ה-Visitor כאן הוא המחלקה ComplexControlPainter אשר יודעת לקבל רפרנס לאובייקט מסוג Control (שהוא הורה של Form) ולצבוע אותו באחת מ-4 ערכות צבע שונות. אנו מחזיקים אובייקט Visitor כזה בכל אחד מהטפסים האופציונליים באפליקציה ומשתמשים באותו קוד שלו בכדי לצבוע כל אחד מהטפסים (המשתמש בוחר כפתור צביעה בטופס והטופס נצבע על ידי ה-Visitor).

* Sequence Diagram



* 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