### 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).

**שימוש ב-Data Binding:** נמצא בקוד בעת מילוי רשימת העמודים הלא-פעילים, וגם במילוי הטופס המספק פרטים נוספים על עמוד פייסבוק מסויים.

**שימוש ב-MultiProcessing:** חלון עריכת התמונה רץ בתהליכון נפרד מהחלון הראשי ולכן ניתן להשתמש בהם במקביל במקום להשתמש רק בחלון העריכה, בנוסף, בחירת הכפתור Edit בחלון הראשי כאשר חלון העריכה פתוח מעבירה את התמונה הנוכחית לחלון העריכה בתהליכון. בנוסף, טעינת העמודים הלא-פעילים לרשימה מתבצעת גם היא בתהליכון נפרד ולכן ניתן להמשיך להשתמש בחלון הראשי בזמן הטעינה לרשימה.

### תבנית מס' 1 – Factory Method

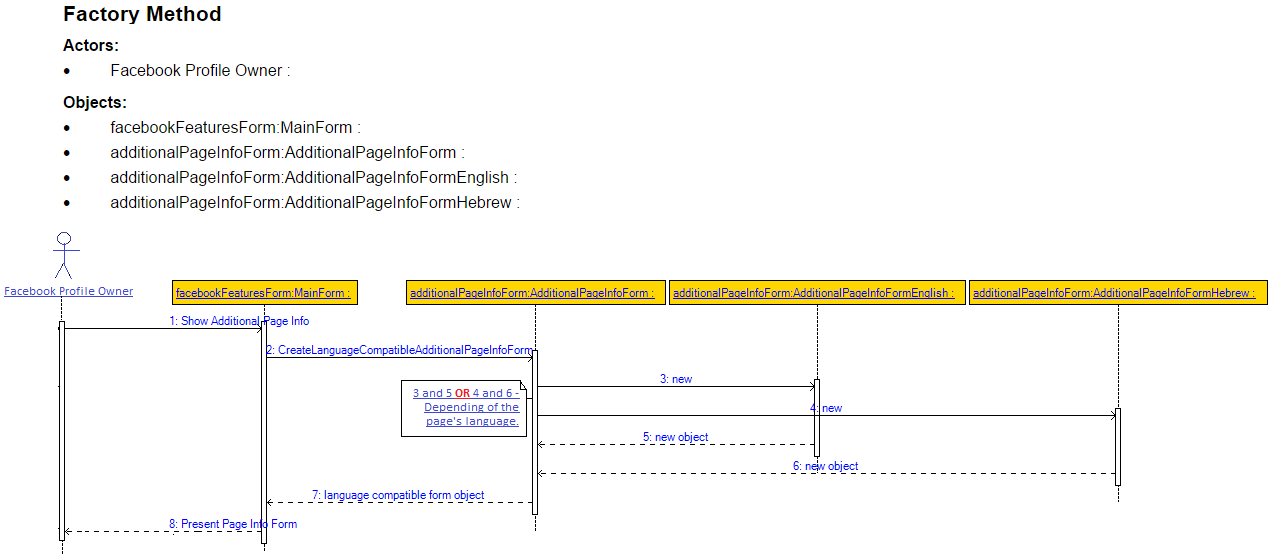
* סיבת הבחירה / שימוש בתבנית:

הוספנו למערכת חלון פרטי עמוד, כדי שהמשתמש יוכל לבחור מתוך רשימת העמודים הלא-פעילים את אחד העמודים ולראות פרטים נוספים לגביו. מכיוון שיש עמודים בעברית ויש עמודים באנגלית, היה צורך ליצור 2 צורות של טופס "פרטי עמוד נוספים" – טופס בשפה האנגלית (שמאל לימין) וטופס בשפה העברית (ימין לשמאל).

* אופן המימוש:

הפונקציה הסטטית CreateLanguageCompatibleAdditionalPageInfoForm במחלקת הבסיס של המשפחה הפולימורפית AdditionalPageInfoForm בודקת את שפת העמוד שעבורו יש לתת פרטים, ומייצרת מחלקת טופס בעל שפה מתאימה (AdditionalPageInfoFormEnglish **או** AdditionalPageInfoFormHebrew ).

* Sequence Diagram



* Class Diagram

\* הערה: הפקדים של WinForms שאנו משתמשים ב-Designer לעיצובם שוכפלו בין רכיבי ה-Contrete מכיוון שאם הם נמצאים רק באבא האבסטרקטי, ה-Designer של VS חוסם את העריכה, ולכן לא הייתה ברירה אלא לשכפלם כדי להשתמש ב-Designer.

במודל זה המחלקה האבסטרקטית (האבא) AdditionalPageInfoForm מתפקדת כחלק ה-Abstraction ושני הבנים (AdditionalPageInfoFormEnglish, AdditionalPageInfoFormHebrew) מתפקדים כ-Concretes. המחלקה MainForm היא ה-Client.



### תבנית מס' 2 –Adapter

* סיבת הבחירה / שימוש בתבנית:

כאשר משתמש מדפדף בתמונות הפייסבוק שלו בחלון הראשי, אנו מעוניינים שהוא יוכל לשנות את התמונה ע"י הפעלת פילטרים על התמונה, ולאחר מכן לערוך את התמונה , ולבסוף להעלות אותה לפייסבוק.

עריכת התמונה לפני העלתה לפייסבוק מתבצעת בחלון FormImageEditing .

אנו מעוניינים להשתמש ב:legacy component

: FacebookWrapper.ObjectMode.User

ולהשתמש בשירותיו על מנת להעלות את התמונה לפייסבוק.

המחלקה ImageEditing Form הינה מחלקה כללית שלא קשורה בהכרח לפייסבוק, וכן היא משתמשת בממשק

ImageUploader כדי להעלות תמונה לשירות כלשהו. בממשק מוגדרת מתודה "UploadImage" שמקבלת אובייקט מסוג Image (של סישארפ) . המחלקה של פייסבוק יכולה לקבל רק מערך של בתים כדי להעלות תמונה על כן הממשקים אינם תואמים ,אין יכולת וגם אין סיבה לממש את תהליך האדפטציה באחת המחלקות (מסיבות של תחזוקה) ולכן היינו צריכים להגדיר את הadapter

אופן המימוש:

המחלקה FormImageEditing משתמשת בממשק של ImageUploader

שמקבלת מופע של ממשק זה בבנאי שלה.

המחקלה שמממשת את ImageUploader היא FacebookImageUploader

והיא משתמשת במחלקה של User של פייסבוק על מנת לבצע את ההמרה מאובייקט תמונה למערך של בתים, באופן שיהיה שקוף לclient (FormImageEditing).

FormImageEditing : Client

ImageUploader: ITarget

ImageFacebookUploader: Adapter

FacebookWrapper.ObjectMode.User : Adaptee

[Comment : We cannot defined in the UML Tool that we use “Tilted arrows” to mention non-instantaneous action like using the facebook servers, so we mentioned it as a comment]

* 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 – Composite

* סיבת הבחירה / שימוש בתבנית:

אנו מעוניינים שהמשתמש יוכל ליצור פילטרים לתמונות על מנת לערוך אותן וכן לשמור פילטרים אלו, וכן גם את היכולת ליצור קבוצות של פילטרים וקבוצות של קבוצות של פילטרים שמהוות סוג של תיקיה.

בנוסף, אנו מעוניינים שהמשתמש יוכל לשנות פילטרים קיימים ולשמור אותם, וגם לשנות קבוצה של פילטרים בבת אחת, ולא פריט פריט.

על כן השתמשנו בComposite pattern

שכן אנו רוצים להשתמש בפריט ובקבוצה של פריטים באופן דומה, ולערוך אותם.

* אופן המימוש:

ImageEditingForm משתמשת במחלקות Ifilter, Filter, FilterGroup.

יצרנו ממשק IFilter שמגדיר, בין היתר, מתודה FilterSet() שמשמעותה לשנות את פילטר הקיים( למשל להוסף לפילטר צבע אדום)

את הממשק מממשים: Filter & FilterGroup

FilterGroup מכיל אגרגציה של IFilters

כך ש filterGroup תוכל להחזיק בתוכה באופן רקורסיבי פילטרים וגם קבוצות של פילטרים, וברגע שמשנים אובייקט אחד, כל התתי קומפוננטות משתנות יחד איתו רקורסיבית, וכן ניתן לשנות קבוצות ותת קבוצות של פילטרים ביתר קלות.

FormImageEditing : Client

IFilter: Component

FilterGroup: Composite

Filter: Leaf

* Sequence Diagram



* Class Diagram

