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

* Internet checker  
  בדיקה אלו אנשים מתעניינים בפוסטים (סטטוסים וכו') שהיוזר מפרסם בפרק זמן ספציפי.   
  התעניינות נבחנת ע"י בדיקה מי מגיב (תגובה/LIKE) לפוסטים.  
  הפיצ'ר נמצא בכרטיסיית Interest Check באפליקציה.
* פילטור חברים ומיילים ע"י פילטרים מקונפגים מראש.  
  ע"י יצירת פילטר מתקדם (קריטריונים: גיל, מין ושייכות לקבוצת חברים)   
  ניתן לראות רשימה מוקטנת של חברים שמקיימים תנאי הפילטר  
  וכן לראות רק שרשורי הודעות ששותף בהן חבר שמקיים תנאי פילטר.  
  את הפילטר מוסיפים בכרטיסיית filters באפלקיציה.  
  את רשימת החברים המצומצמת (והמורחבת) ניתן לראות בכרטיסיית friends באפליקציה.  
  את שרשורי ההודעות המצומצמת (והמורחבת) ניתן לראות בכרטיסיית messages באפליקציה.

### תבנית מס' 1 – Composite

* סיבת הבחירה / שימוש בתבנית:

במערכת ישנם פילטרים בסיסיים שבודקים קיום תנאי אחד ספציפי על קבוצת יוזרים/חברים,  
ופילטר שבודק תנאי שהינו שילוב של מספר פילטרים בסיסיים.  
אופן השימוש של הקלינט בפילטר בסיסי ופילטר המורכב ממספר פילטרים הינו זהה,   
בכולם הוא נותן קבוצת יוזרים ומקבל קבוצה מפולטרת ואין זה מעניין אם הפילטור נעשה ע"י קבוצת פילטרים או פילטר בסיסי.

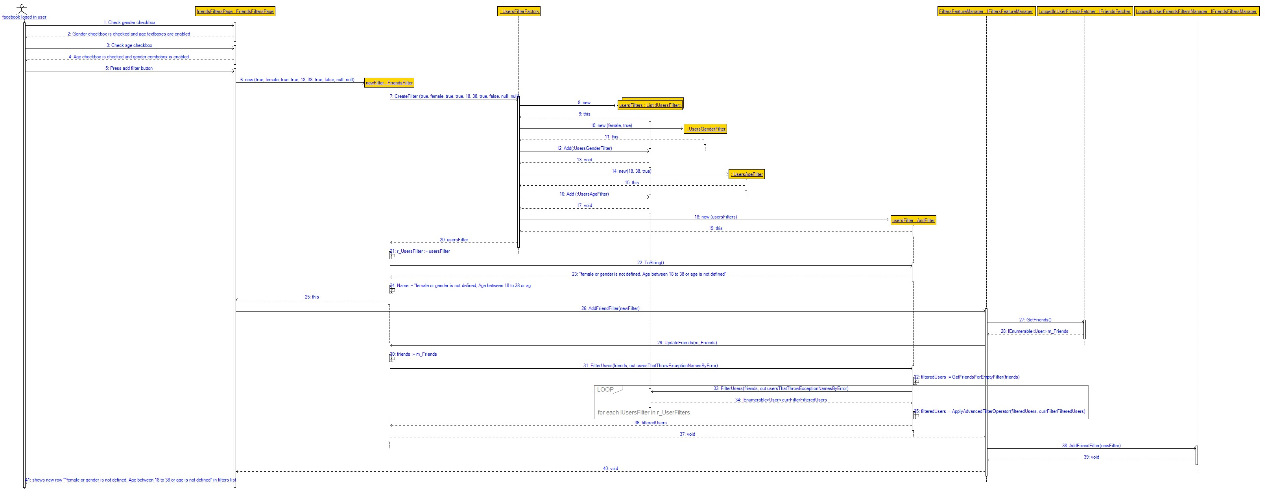
* אופן המימוש:

הוגדר ממשק IUsersFilter אשר מהווה הcomponent של התבנית.

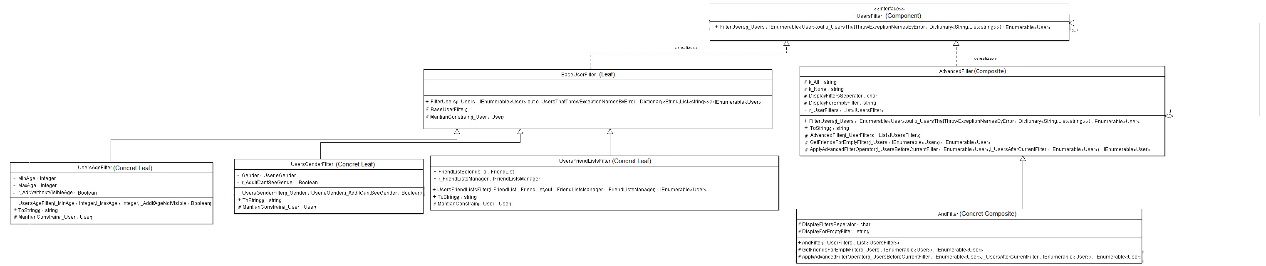
ממשק זה מגדיר את הפונקציה FilterUsers.  
מחלקת הComposite של התבנית הינה AdvancedFilter אשר ממשת את IUsersFilter ומכילה בתוכה רשימה של IUsersFilter.  
העלים הינם UsersAgeFilter , UsersFriendListsFilter ו- UsersGenderFilter אשר כולם יורשים מ- UsersGenderFilter שממש את IUsersFilter.

AdvancedFilter היא מחלקה אבסטרקטית אותה יורשת המחלקה AndFilter. המחלקה נכתבה כמחלקה אבסטרקטית כדי שבעתיד יהיה אפשר להוסיף גם מחלקה כגון OrFilter שגם יירשו מ- AdvancedFilter.

* Sequence Diagram

  
יש קובץ מצורף CompositeSequenceDiagram.jpg

* Class Diagram

  
יש קובץ מצורף CompositeClassDiagram.jpg

### תבנית מס' 2 – Singelton

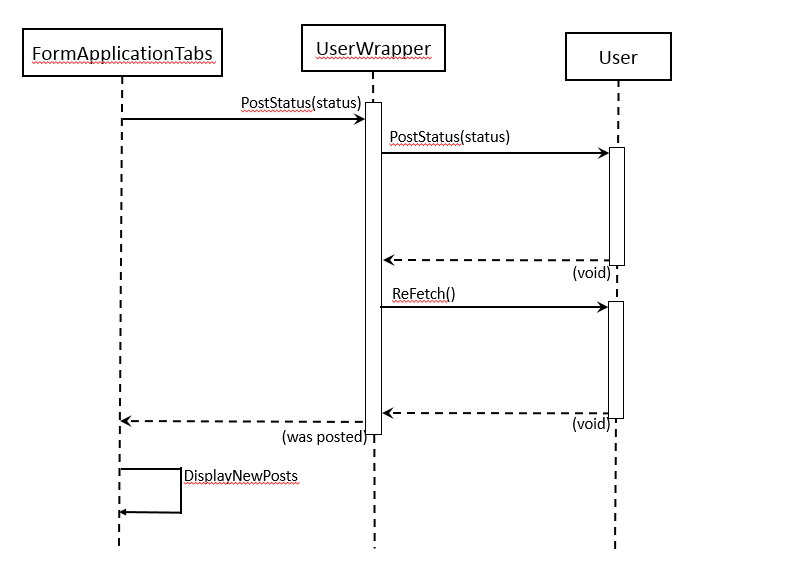
* סיבת הבחירה / שימוש בתבנית:

במערכת לאחר מסך הLOGIN יש יוזר אחד שמחובר. נרצה שכל הרכיבים יפנו באופן יזום לאינסטנס של היוזר הזה במערכת, שכן כל עדכון שיעשה ביוזר (כגון רשימת פוסטים/ רשימת חברים) יצטרך להעשות על היוזר הזה.  
נרצה שהמידע של מי היוזר ופרטי היוזר יהיה מעודכן ומסונכרן בכל הרכיבים שצריכים לפנות ליוזר המחובר.

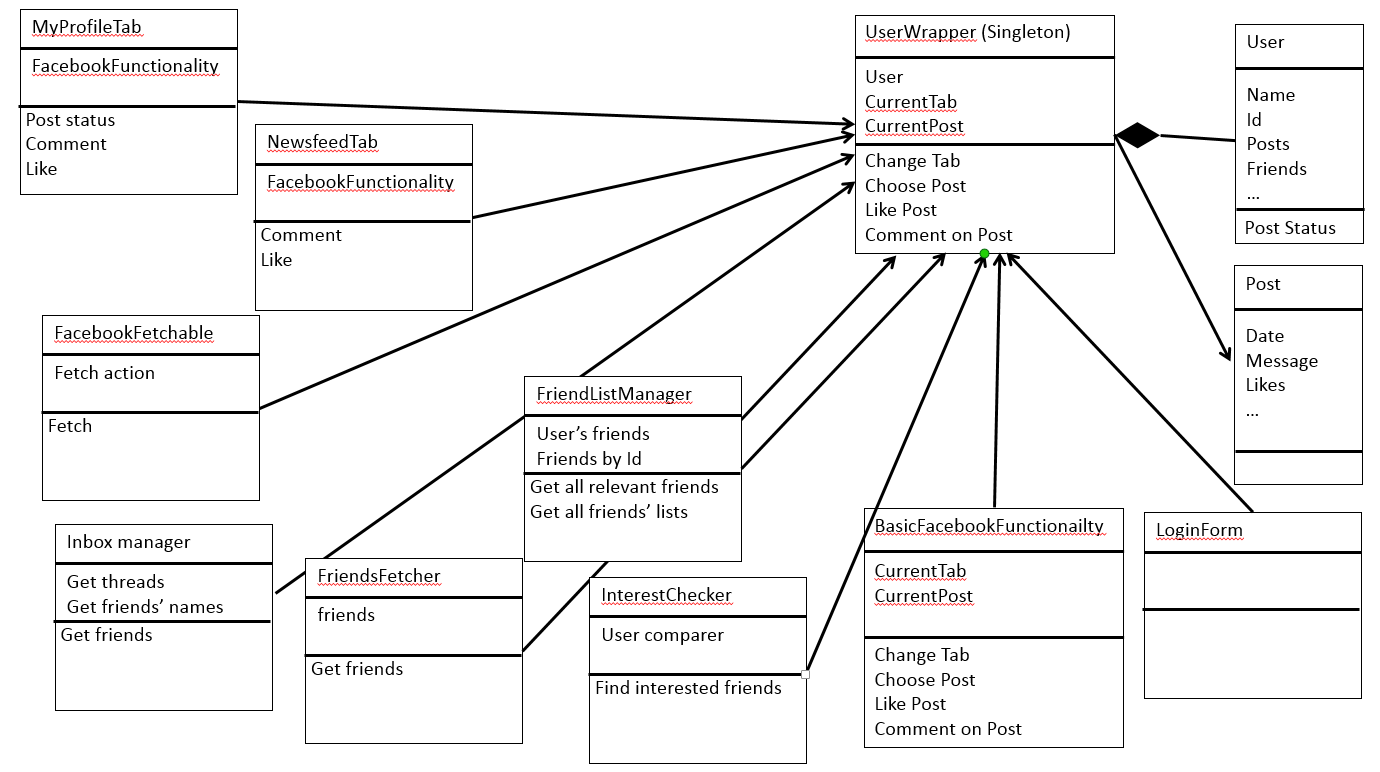
* אופן המימוש:

UserWrapper מממש את התבנית ע"י בנאי פרטי ו-Property סטטי Instance.  
הInstance מומש כך שתהיה נעילה של השדה הפרטי אך ורק אם הוא היה null כשהתרד ניגש לProperty.

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Factory Method

* סיבת הבחירה / שימוש בתבנית:

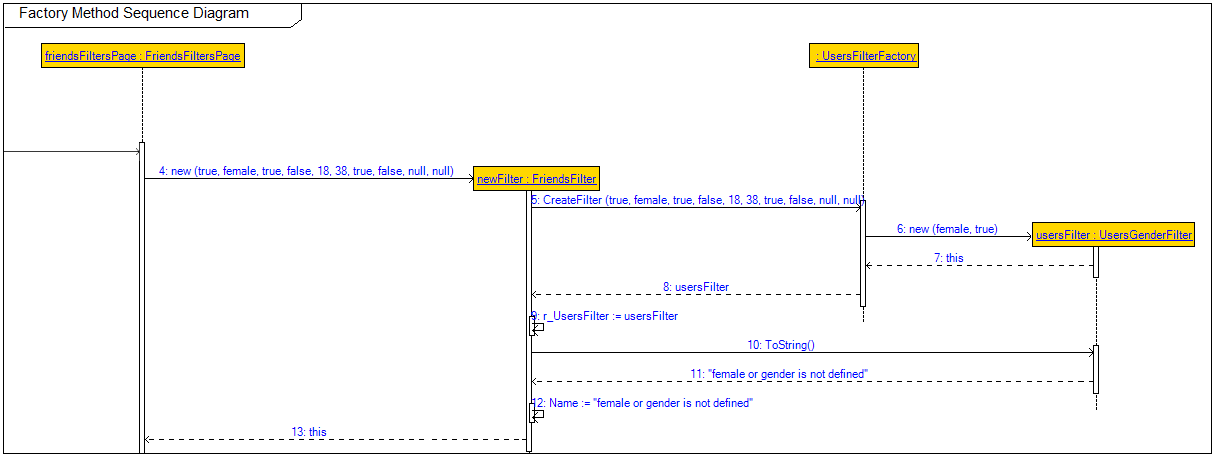
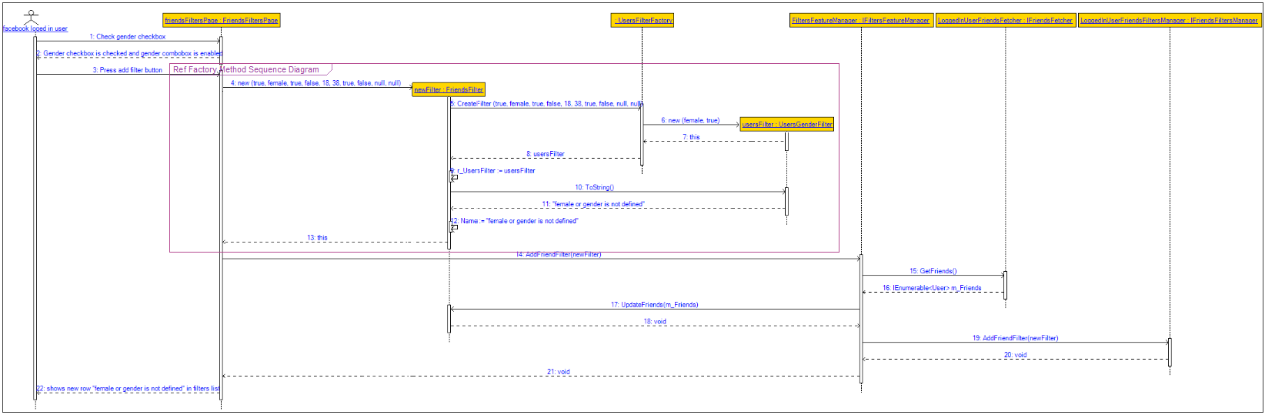
במערכת ישנה משפחה פולימורפית של מחלקות פילטרים. הקלינט יודע ערכי קונפיגורציה שקיבל ממשתמש המערכת, בסופו של דבר הוא משתמש בכל מחלקה מהמשפחה באותו האופן.  
לכן נרצה את היצירה של המחלקות להוציא מהקלינט למחלקה נפרדת שמקבלת מידע ומחזירה מחלקה מתאימה.

הקלינט לא צריך לדעת איזו מחלקה יוצרים, שכן השימוש זהה.

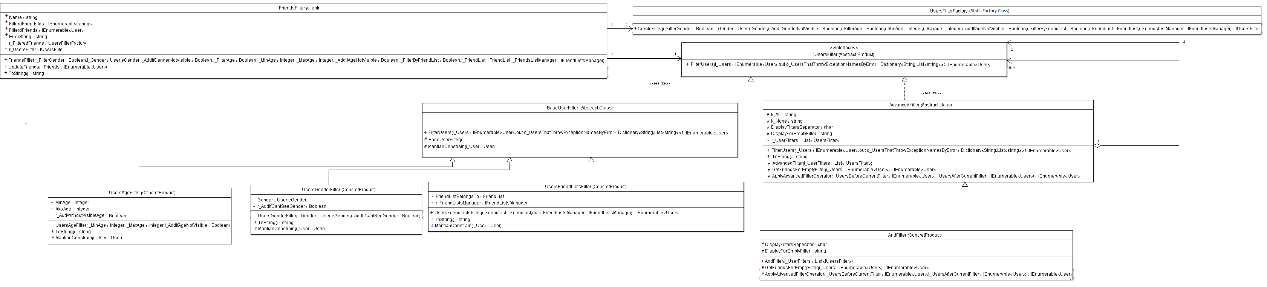
* אופן המימוש:

המחלקה UsersFilterFactory הינה המחלקה הסטטית של תבנית ה Factory Method,   
המתודה הסטטית שמחזירה את האובייקט האבסטרקטי הינה CreateFilter.  
הקלינט הינו FriendsFilter שמחזיק IUsersFilter שזהו הפרודקט האבסטרקטי של התבנית.  
המחלקות שממשות את IUsersFilter הן שתי המחלקות האבסטרקטיות: BaseUserFilter ו- AdvancedFilter. את הראשונה יורשות המחלקות UsersAgeFilter , UsersFriendListsFilter ו- UsersGenderFilter , את השניה יורשת המחלקה AndFilter והן מהוות את הפרודקטס הקונקרטים.

* Sequence Diagram

חלקי – רק המחלקות שקשורות לתבנית:  
  
מלא:  
  
יש קבצים מצורפים:   
FactoryMethodSequenceDiagram.jpg, FactoryMethodFullSequenceDiagram.jpg

* Class Diagram

  
יש קובץ מצורף בשם :  
FactoryMethodClassDiagram.jpg

עבודה אסינכרונית:

הקריאות לפונקציות הבסיס של פייסבוק (Like, Commit, Post) נעשות באופן אסינכרוני, על מנת לא לעצור את ממשק המשתמש בזמן שהקריאה נשלחת לפייסבוק והערכים השמורים במערכת מתעדכנים. את הקריאות מימשנו בעזרת עטיפת הפונקציות הקודמות ב-Task שנוצר ומופעל בלי להמתין לו, מכיוון שאינו מחזיר ערך.

ניתן למצוא את הקריאות האסינכרוניות במחלקה BasicFacebookFunctionality בפונקציות commentAsync, likeAsync ובמחלקה FormApplicationTab בפונקציה buttonSetStatus\_Click.  
  
עבודה עם Data Binding   
הכרטיסיות messages (מחלקה InboxPage) וכרטיסיית friends (מחלקה FriendsPage)   
באפליקציה עובדות עם data binding.  
האובייקטים של הDATAMINDING במחלקה InboxPage הם: iFriendFilterBindingSource, ו-inboxThreadDisplayBindingSource   
האובייקטים של הDATAMINDING במחלקה FriendsPage הם: iFriendFilterBindingSource , ו- userBindingSource.  
השימוש בביינדינג של אובייקט iFriendFilterBindingSource נעשה כדי למלא את friendFilterComboBox ולזהות מתי יש שינוי בבחירת פילטר,   
כאשר בcombobox של הפילטרים משתנה הבחירה הרשימה של החברים/שרשורי הודעות משתנה בהתאם.  
השימוש בביינדינג של אובייקט inboxThreadDisplayBindingSource נעשה כדי שמתי שנבחר שרשור הודעות ברשימת שרשורי הודעות, ההודעה תוצג בפנל הימיני.  
השימוש בביינדינג של אובייקט userBindingSource נעשה כדי שמתי שנבחר חבר/ה ברשימת החברים, התמונה של החבר/ה תוצג בפנל הימיני