### 

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

* מתזמן פוסטים - פיצ'ר הנותן אפשרות לתזמן העלאת פוסטים לזמן עתידי
* סטטיסטיקות על עמודים - פיצ'ר הנותן סטטיסטיקות על העמודים שלהם המשתמש עשה לייק

### תבנית מס' 1 – Singleton

* סיבת הבחירה / שימוש בתבנית:

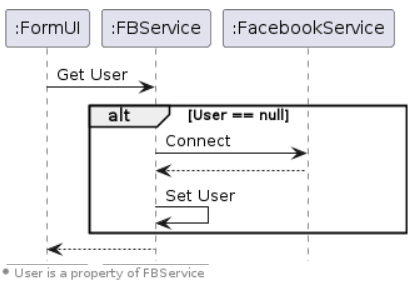
בחרנו בדיזיין פאטרן זה כיוון שאין לנו צורך ביותר ממופע אחד של המחלקה, ויותר ממופע אחד יוצר לנו בעיה (בעיית אבטחה - יווצר טוקן חדש, שלא יהיה בשימוש, שמספק גישה למשתמש איתו הלקוח מתחבר)

בחרנו להשתמש בפיצ'רים של השפה ולממש את הפאטרן באמצעות מחלקה סטטית. מטרת השימוש של המחלקה הינה תקשורת בין ממשק המשתמש לשרת הנתונים של פייסבוק.

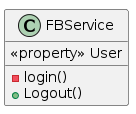
* אופן המימוש:

המחלקה הסטאטית FBService. אופן קבלת הInstance, שבמקרה שלנו הוא User, הוא ע"י גישה ל-property בשם User.

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Adapter

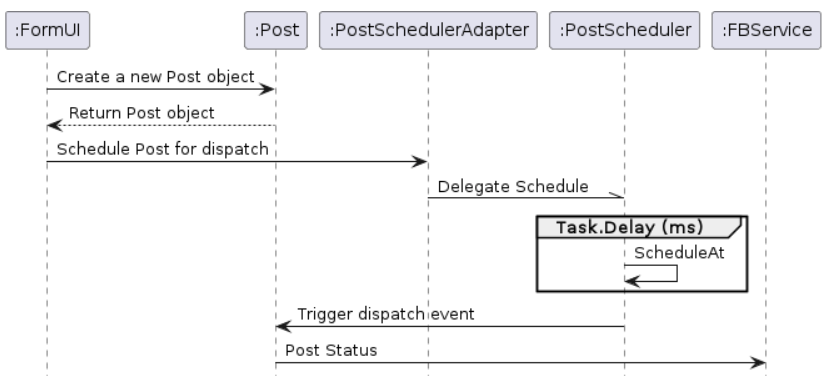
* סיבת הבחירה / שימוש בתבנית:

בחרנו בדיזיין פאטרן זה כיוון שיש לנו רכיב Legacy בשם PostScheduler, שרצינו להתאים לממשק רחב כללי יותר בשם IScheduler שמטרתו לתזמן פרסומים, לאו דווקא פוסטים פשוטים (לדוגמא - תזמון העלאת תמונות).

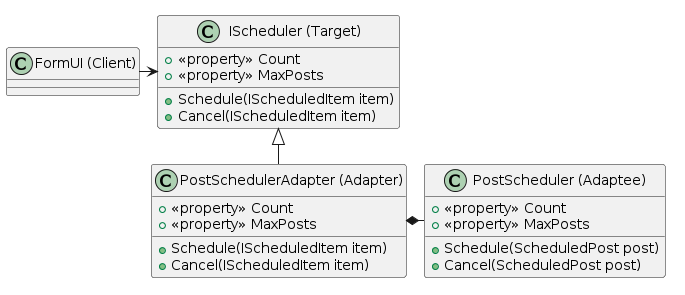
* אופן המימוש:

יצרנו ממשקים חדשים בשם IScheduler, IScheduledItem שמהווים ממשק כללי לפריטים מתוזמנים.  
המחלקה ScheduledPost מממשת את הInterface של IScheduledItem, ומחלקה חדשה בשם PostSchedulerAdapter המשמשת כמתאם בין הממשק הכללי למחלקה הקונקרטית PostScheduler. במימוש זה השתמשנו בקומפוזיציה.  
Client - המחלקה FormUI  
Target - הממשק IScheduler  
Adapter - המחלקה PostSchedulerAdapter  
Adaptee - המחלקה PostScheduler

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Proxy

* סיבת הבחירה / שימוש בתבנית:

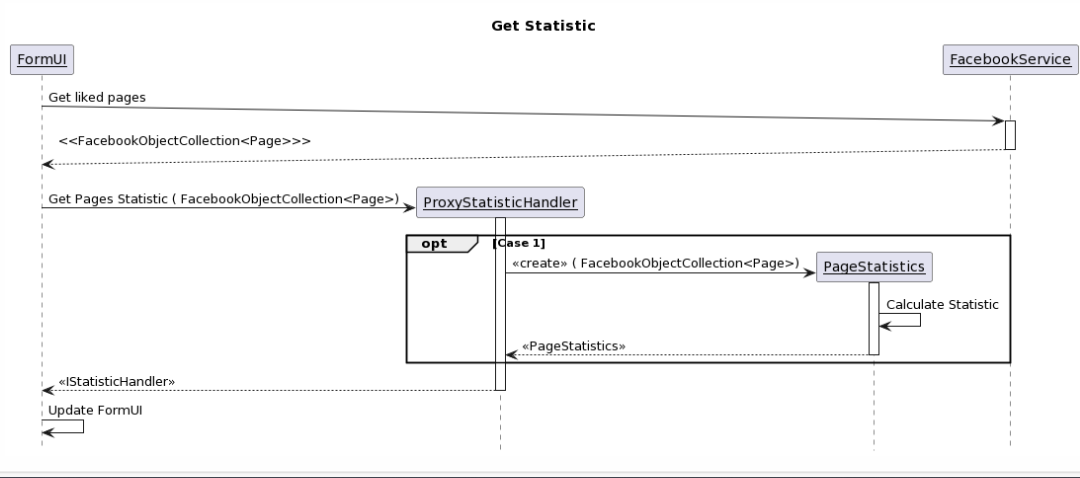
בחרנו בדיזיין פאטרן זה כיוון שיש לנו רכיב Legacy בשם Statistic , שכרגע עובד עם רשימות עמודים אהובים בפייסבוק ובעתיד אם נרצה להוסיף לכך שהרכיב יוכל לעבוד עם רשימת קבוצות נרצה להוסיף עוד פונקציונליות לStatistic אך בלי לפגוע בStatistic של עמודים.

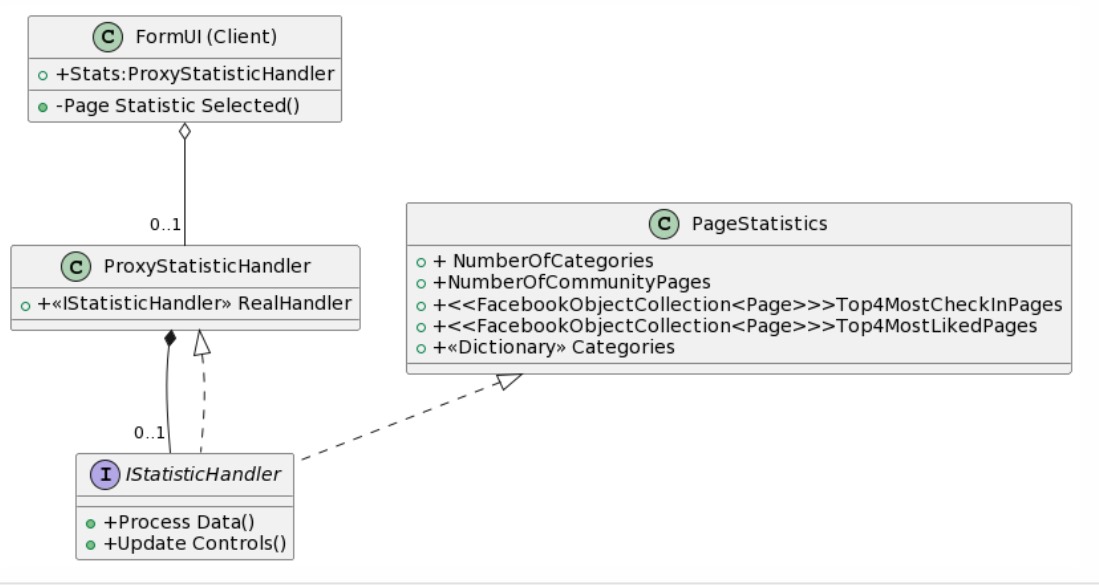
* אופן המימוש:

יצרנו ממשק חדש בשם , IStatisticHandler שמהווה ממשק כללי לסטטיסטיקה.  
המחלקה PageStatistic מממשת את הInterface של IStatisticHandler.

מחלקה חדשה בשם ProxyStatisticHendler משמשת כמחברת בין הממשק הכללי למחלקה הקונקרטית PageStatistic. במימוש זה השתמשנו בקומפוזיציה.

* Sequence Diagram



* Class Diagram 

**עבודה אסינכרונית עם ממשק משתמש:**

השתמשנו בThreads בקריאות לAPI של פייסבוק, דוגמאות לכך בקוד ניתן לראות בתוך המתודות:

* linkLabelFetchNewsfeed\_LinkClicked
* linkLabelFetchPosts\_LinkClicked
* linkLabelFetchPhotos\_LinkClicked

מטרת השימוש היא לא לחסום למשתמש את האפשרות להשתמש באפליקציה **בזמן ההמתנה לתשובה מהשרת** של Facebook. כך שהמשתמש יכול להמשיך בפעילותו עד **קבלת התשובה מהשרת**.  
כל לחיצה על כל כפתור Fetch באפליקציה יוצר Thread חדש, ומתבצע Invoke בנפרד, בהתאם לקומפוננטה המאחסנת את המידע.

**עבודה עם Data Binding:**

אכלוס כל המידע ב-List Boxes ברחבי האפליקציה מתבצע באמצעות Data Binding. דוגמאות לכך בקוד ניתן לראות בתוך המתודות:

* fetchNewsFeed
* fetchPosts
* fetchComments