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

* מאפשר למשתמש להודיע לחברי הפייסבוק שלו בעזרת פוסט בכל פעם שהוא מתחיל לשדר באתר Twitch.tv
* מאפשר למשתמש להגיב בצורה גורפת להזמנות אירועים שקיבל בפייסבוק.

### תבנית מס' 1 – Proxy

* סיבת הבחירה / שימוש בתבנית:

המחלקה TwitchAPIWrapper המשמשת לגישה לפונקציונאליות לAPI של Twitch, אמורה לספק רק מידע אודות הנתונים שנאספו, וכן הטופס FormMain אמור רק לבקש אותו ולא להכיר את ההיגיון הרצוי בעקבות המידע שהתקבל.  
המחלקה TwitchForFacebookProxy משמשת כנקודת אמצע המאפשרת למשתמש בטופס לבקש את אותן הבקשות מהAPI, אך מעבירה את ביצוע ההיגיון למחלקה הייעודית לכך.

בכך אנחנו תומכים במיינטיינאביליות התכנית.

* אופן המימוש:

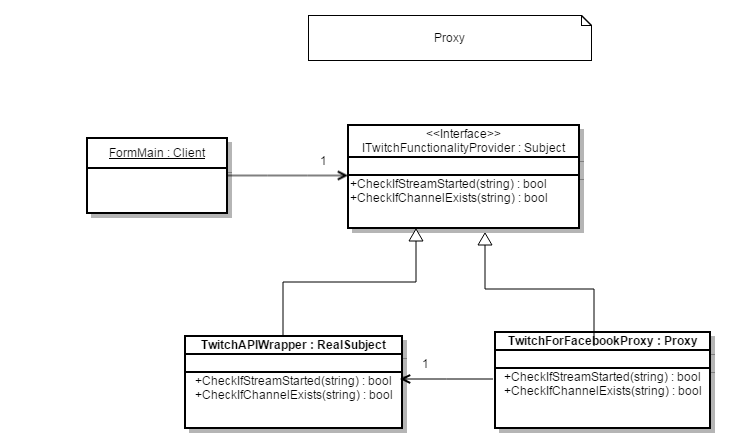
בFormMain בשורה קיימת המתודה AutoPostTwitchUpdate() הנקראת כל זמן קצוב ובודקת אם המשתמש הרצוי התחיל כרגע לשדר.  
שם אנו קוראים למתודה CheckIfStreamStarted(string) של TwitchForFacebookProxy.

המתודה מבצעת הבדיקה דרך הAPI ואז שולחת עדכון לפייסבוק והודעה למשתמש על כך שההודעה נשלחה.

* Sequence Diagram

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

* Class Diagram



### תבנית מס' 2 – Adapter

* סיבת הבחירה / שימוש בתבנית:

עלה לנו הצורך לגשת למשתמש המחובר ביותר מטופס אחד(ספציפית טופס שאינו הטופס בו התחברנו למערכת), אך מכיוון שהAPI של הפייסבוק שסופק לנו לא מכיל אפשרות גישה חוזרת למשתמש המחובר, וכדי למנוע מצב שבו צריך להעביר את הreference לאובייקט המשתמש בין הטפסים, היינו צריכים לשנות את דרך הגישה והשימוש במערכת החיבור.  
בכך שבחרנו בגישה זו אנחנו תומכים במיינטיינאביליות התכנית בכך ששומרים דרך גישה יחידה וקונסיסנטית במקום להעביר הפניות וליצור מקומות נוספים לתחזוקה עתידית.

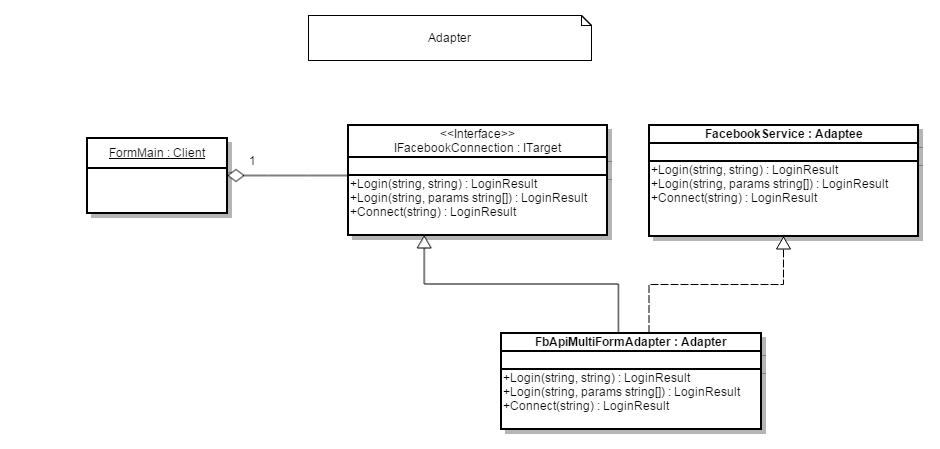
* אופן המימוש:

בנקודות החיבור בקוד(כאשר כפתור הlogin נלחץ והמשתמש מתחבר לפייסבוק), אנו קוראים לפקודות הLogin והConnect של מחלקת הFbApiMultiFormAdapter המממשת את האינטרפייס IFacebookConnection, כל מה שהמחלקה עושה היא לשמור רפרנס של המשתמש המחובר בעקבות חיבור מוצלח, אליו לאחר מכן ניתן לגשת מכל מקום מפני שהמחלקה היא סינגלטונית.

* Sequence Diagram

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

* Class Diagram



### תבנית מס' 3 – Singleton

* סיבת הבחירה / שימוש בתבנית:

היה צורך בייצוג יחיד של מספר מחלקות תחזוקה(FbApiMultiFormAdapter, UserInfo, TwitchForFacebookProxy, TwitchAPIWrapper), כדי ליצור גישה גלובאלית לנתונים מסוימים הרלוונטיים לכל התכנית.

* אופן המימוש:

יצרנו מפעל סינגלטונים(המחלקה SingletonFactory) המאפשרת לנו לבקש לקבל גישה לעותק היחיד עבור סוג מסוים בו אנו רוצים להשתמש.  
המפעל מחזיר לנו את העותק היחיד, ואם אינו קיים, יוצר אותו בLazy Creation.

* Sequence Diagram

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

* Class Diagram

### C:\Users\Mor\Downloads\Singleton_Class_Diagram.png

### תבנית מס' 4 – [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