### 

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

* **החבר החופר ביותר** – הפעלת [A1] פיצ'ר זה מבוצעת על ידי לחיצת הכפתור "Show most digging friend", לאחר לחיצה על הכפתור, יוצג למסך חלון חדש בו יופיע פרטים על החבר שנמצא החופר ביותר, לפי כמות הפוסטים שהוא פרסם בשנה האחרונה.
* **בדיקת קורונה סביבי:**

נמצא במחלקה Covid19SickLocation בפרויקט Ex01.ApplicationEngine

מעלה רשימה של מקומות מאומתים של חולה קורנה השוואה אל מול רשימת הצ'קאין של המשתמש, בחירה ידנית של מקומות להשוואה (גם אצל החברים) מול הרשימה והזנת ושמירת מקומות מאומתים לעתיד.

### תבנית מס' 1 – Object Proxy

* סיבת הבחירה / שימוש בתבנית:

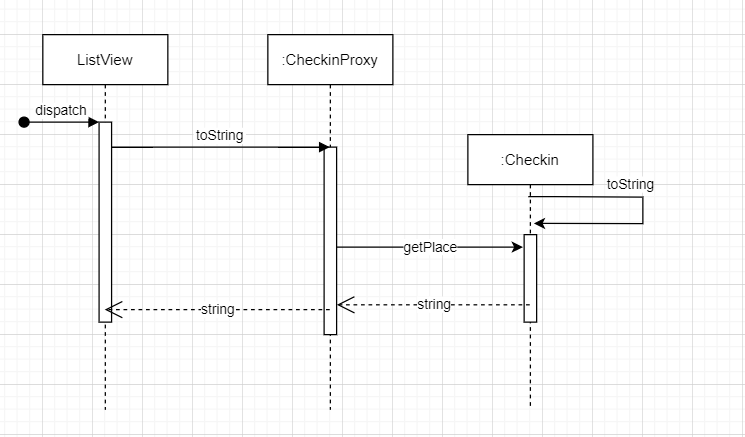
על מנת להציג את האוסף של האובייקט Checkin בתוך ListView כך שה-ListView יציג Deep Data בתור DisplayMember, ומבלי שה-UI יצטרך לבצע שינויים בקוד.

כיוון שמדובר ב-Legacy Component, עלה צורך ב- CheckinProxy, אשר ייתחזה לאובייקט המקוריי ויאפשר את הנ"ל.

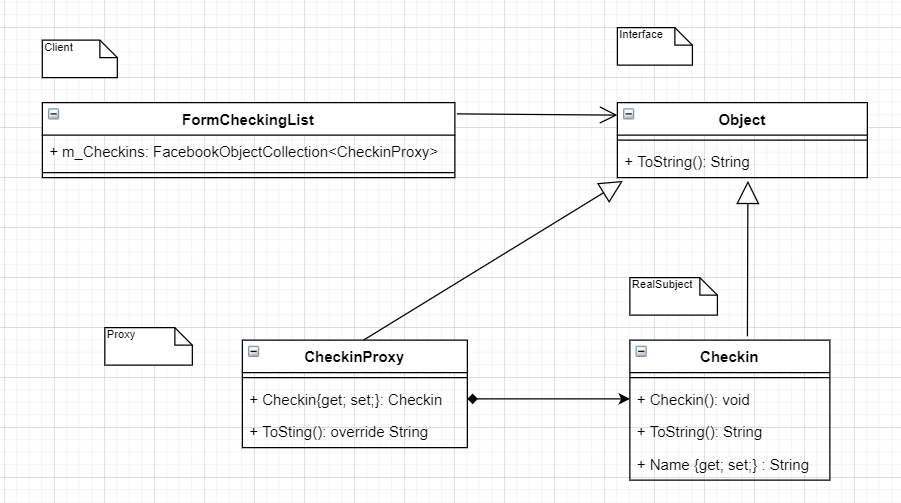
* אופן המימוש:

הגדרת המחלקה נמצאת ב-Ex01.ApplicationUI Project, בקובץ נפרד CheckinProxy.cs. אובייקט מסוג CheckinProxy מחזיק Reference לאובייקט Checkin, דורס את המתודה ToString, וממש כזאת שתחזיר Deep Data (Checkin.Place.Name).

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Facade

* סיבת הבחירה / שימוש בתבנית:

1. Facebook.dll ו-FbGraphAPI.dll הן ספריות צד שלישי שאין לנו גישה לקוד שלהם. הספריות דורשות התאמות לצרכינו, ואיננו מעוניינים שההתאמות יבוצעו ב-Client. ה-Facade מסייע לנו בביצוע התאמות עבור ה-Client.
2. אנו מעוניינים שה-Client יעבוד עם רכיב אחד, שהוא ה-Façade, אשר מרכז עבורו את הנגישות לdll-ים.  
    ה-Facade יחשוף ל-Client רק את המתודות הרלוונטיות מתוך Facebook.dll ו-FbGraphAPI.dll, מבלי שה-Client ידע כי מדובר בכמה רכיבים שונים, ובכך ה-Facade מקל עבור ה-Client את ההתמצאות בקוד.

* אופן המימוש:

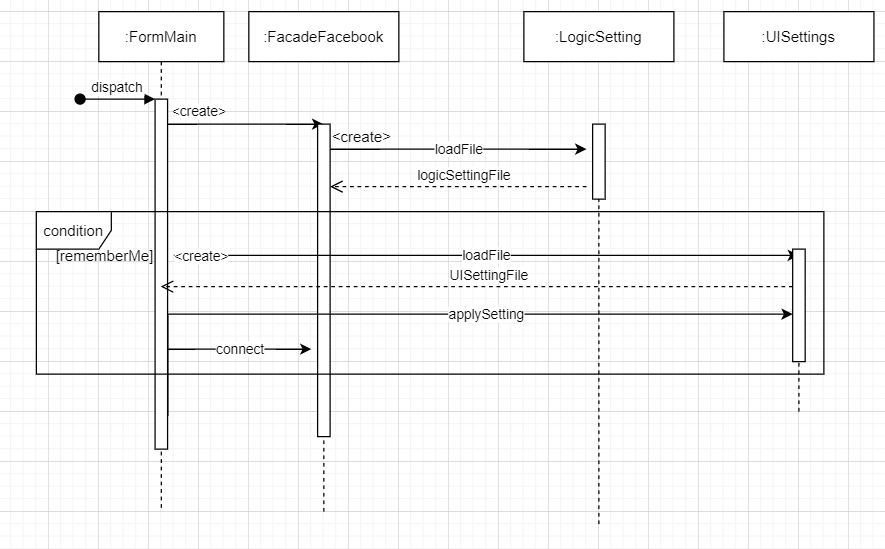
הגדרת המחלקה נמצאת ב-FacadeFacebook Project בקובץ FacadeFacebook.

ה-Properties של המחלקה הם:

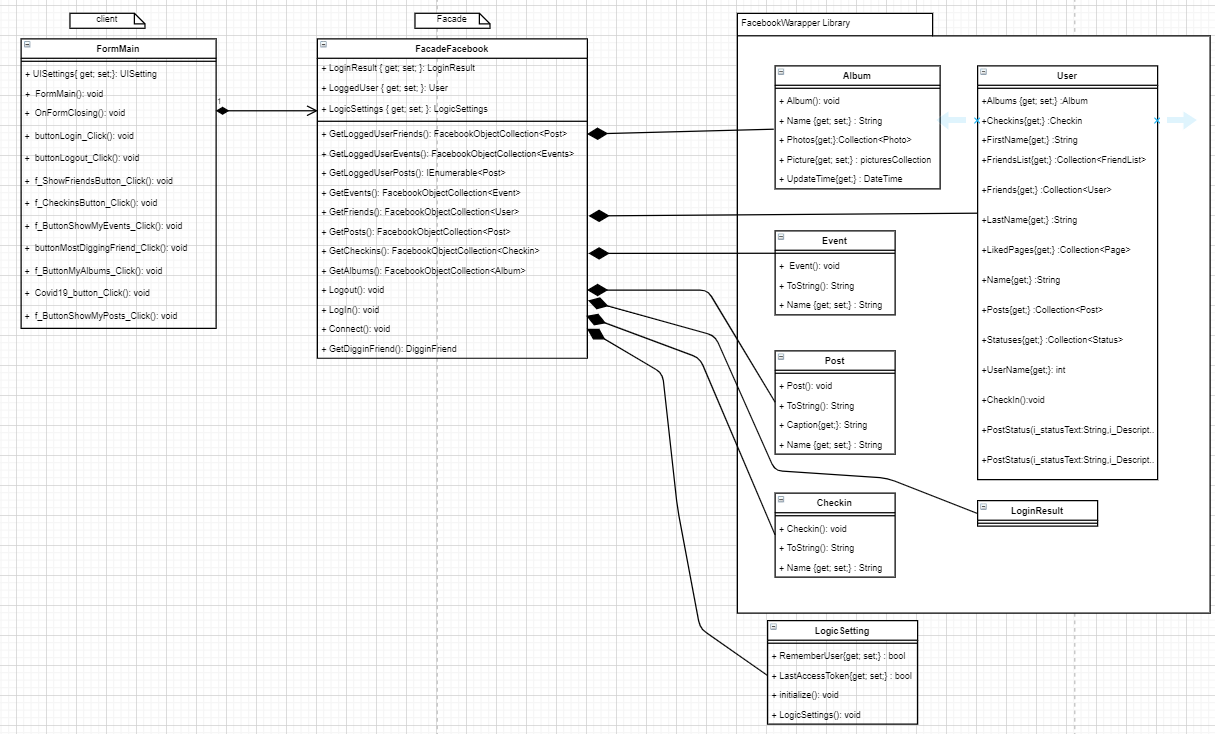
* User
* LogicSettings LogicSettings
* LoginResult

שעליהם ה-Facade מבצע מניפולציות (בדמות כל מיני מתודות) על מנת לשטח ולהקל את העבודה של ה- Client עם האובייקטים הלוגים הללו

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Singleton

* סיבת הבחירה / שימוש בתבנית:

מעוניינים שה-Façade Pattern אותו תיארנו, יקיים שני דברים:

1. יהיה יחיד במערכת. (Facade אחד ל-Client אחד)
2. ה-Client יוכל לגשת אליו מכל מקום.

סינגלטון פותר בדיוק את הצרכים האלה.

* אופן המימוש:

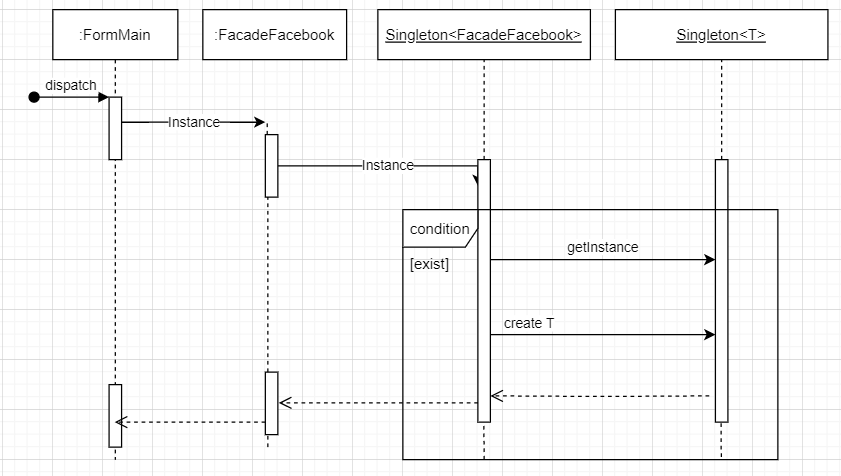
ב-FacadeFacebook Project מוגדרות המחלקות SingletonException ו- Singleton בקבצים SingletonException.cs ו- Singleton.cs בהתאמה.

SingletonException – מימוש Exception ייעודי עבור יצירת הסינגלטון.

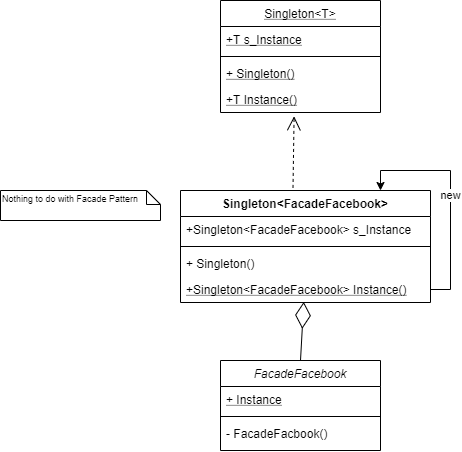
Singleton – מחלקה גנרית, אשר עושה שימוש ב-Static Constructor, DoubleCheckLocking ו-Reflection על מנת שיהיה Thread safe, Fully Lazy עם שילוב Exception.

ב-Client (של ה-Pattern) – ב-FacadeFacebook Project מוגדרת המחלקה FacadeFacebook בקובץ FacadeFacebook.cs. המחלקה FacadeFacebook מממשת Static Property אשר מחזיר את ה-Instance Static Property (שהוגדר גנרי במחלקה Singleton), מסוג FacadeFacebook

* Sequence Diagram



* Class Diagram



**עבודה אסינכרונית –**

מקומות בקוד ה-ui, בקובץ Ex01.ApplicationUI.FormMain.cs בהם מימשנו עבודה אסינכרונית:

* שורה 223
* שורה 229
* שורה 235
* שורה 241
* שורה 247
* שורה 253

בכל המקומות הנ"ל מימשנו עבודה אסינכרונית מאותה הסיבה - לאפשר לאפליקציה להמשיך להיות איטרקטיבית בו בזמן שהפעולות הנדרשות מתבצעות

בנוסף, השימוש ב-Threads נוספים אפשרו לנו לשלב אנימציה של Loading, המשפרת את חווית המשתמש בפעולותיו השונות על המסך.

אופן המימוש:

לחיצת כפתור מתוך הכפתורים הפותחים חלון חדש ------> הפעלת אנימציית loading על ידי ה-main thread באירוע הלחיצה -----> יצירת Thread חדש אשר יטען טופס חדש, ייקח כמה שייקח



לאחר שהטופס נטען, בו בזמן האפליקציה נשארת איטרקטיבית  
ה-Thread שיצרנו יבקש ע"י ה-Main Thread  
מה-Main Thread לסיים  
את האנימציה של ה-Loading  
באמצעות המתודה Invoke   
של המחלקה Control.

Data Binding -

המקומות בקוד בהם עשינו שימוש ב-Data Binding הם:

* Ex01.ApplicationUI.FormFriendList at Line 21
* Ex01.ApplicationUI.FormMain at Line 22
* Ex01.ApplicationUI.FormMain at Line 69
* Ex01.ApplicationUI.FormAlbum at Line 18
* Ex01.ApplicationUI.FormCheckinList at Line 29
* Ex01.ApplicationUI.FormEventList at Line 25
* Ex01.ApplicationUI.FormMosiftDiggingFriend at Line 24