### \*יש להוסיף את הDLLS מהלינק הבא על מנת שהקוד יעבוד כשורה:

<https://drive.google.com/open?id=1NMhPlHBqh6FG10IJ0xluNgNUqwVbylf5>

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

* יצירת קורות חיים באמצעות מידע המשתמש מפרופיל הפייסבוק, אפשרות לעריכתן ושמירתן על מחשב המשתמש.

אופציה נוספת, היא שליחת קורות החיים למייל הנתון בפרופיל המשתמש(כתובת הניתנת לשינוי).

* פרסום פוסט בקיר המשתמש, בתאריך ובשעה כלשהי בעתיד, לבחירתו.

### תבנית מס' 1 – [Facade]

* סיבת הבחירה / שימוש בתבנית:

בפיצר קורות החיים שהגשנו בתרגיל הקודם, הייתה אופציה לשליחת קו"ח למייל הרצוי ע"י משתמש האפליקציה.

מימוש האופציה הזו, שהינה חלק אינטגרלי מפיצר זה, אינו איפשר שימוש חוזר

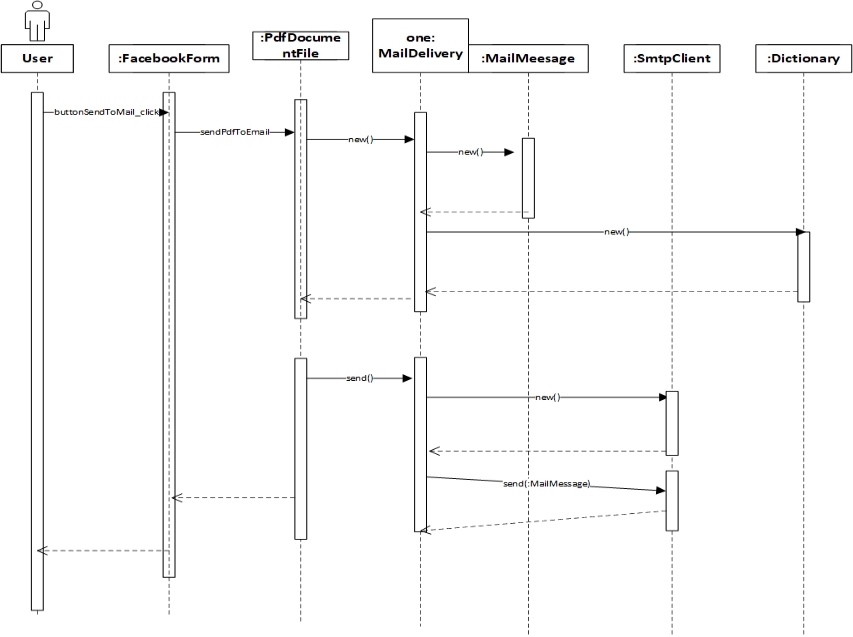
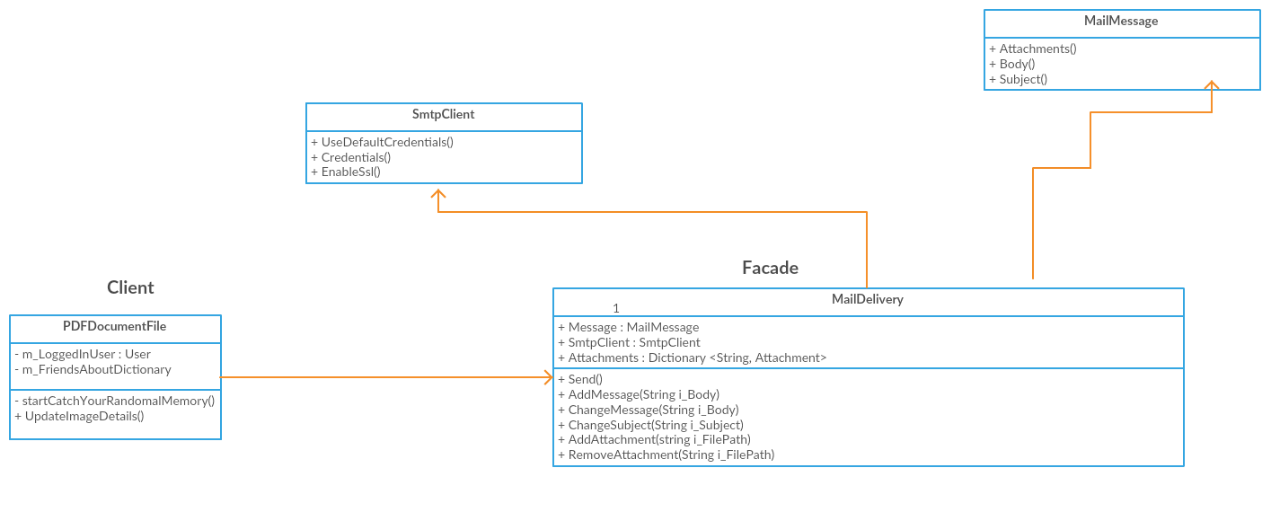
(reuse able בעתיד, כמו כן, הוא חשף את כל תתי הרכיבים אשר חלקם אינם רלוונטיים לקליינט.

לכן, מבחינת נכונות הנדסית, היה מצופה שנממש חלק זה מהפיצ'ר באמצעות תבנית זו, אשר פתרה את כל הבעיות שצויינו וכן , תקל מאוד על תחזוקה עתידית של הקוד והרחבתו.

* אופן המימוש:

את המימוש ניתן למצוא במרחב השמות FacebookAppLogic , במחלקה בשם MailDelivery וניתן למצוא קריאה אליה (קרי –מופע ) , במחלקה PdfDocFile בשיטה בשם SendPdfToEMail.

המימוש כולל קומופוזיציה לתתי הרכיבים אשר דרושים בכדי לאפשר את השירות המצויין לעיל באופן פשוט ונוח עבור קליינט זה וכל קליינט שיצטרך שירות זה.

* Sequence Diagram
* Class Diagram 

### תבנית מס' 2 – [Adapter]

* סיבת הבחירה / שימוש בתבנית:

אחד הפיצרים שלנו הוא יצירת קורות חיים ממידע המאוכסן בפייסבוק בצורת מחלקות ומבני נתונים שונים. עד כה, ביצענו את המרת המידע לטקסט באופן לא תקני מבחינה הנדסית, שכן ביצענו את ההמרה עצמה בתוך . FacebookEngine

בשל כך המחלקה שתפקידה ליצור pdf בלבד תיקשרה באופן ישיר עם מנוע האפליקציה ללא צורך, שכן היא רק מדפיסה טקסט לתוך קובץ PDF.

תבנית זו, שהוספנו בעקבות כל מה שכתבנו, מאפשרת לנו לבצע המרה לטקסט באמצעות מחלקה ייעודית המהווה Adapter, ובכך המחלקה PdfDocumentFile תיקשרה עם ה Adapter-, בכדי לקבל את המידע בפורמט מחרוזתי בממשקיות המתאימה להדפסה, במקום לתקשר עם המנוע באופן ישיר.

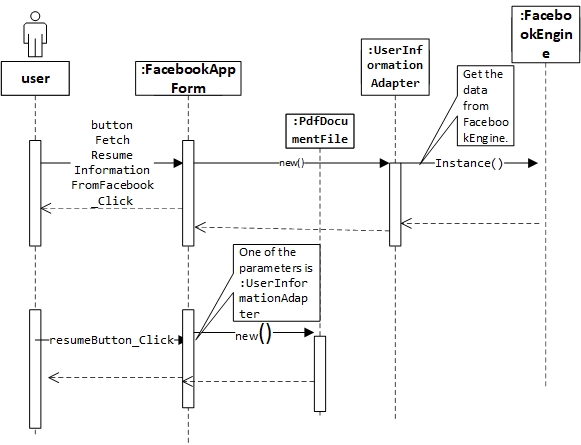
* אופן המימוש:

אופן מימוש תבנית זו הינו פתיחת מחלקה בשם UserInformationAdapter אשר תהווה מתאם (ממשק) בין רכיב המערכת הפנימי במערכת המספק לרכיבי המערכת השונים את המידע, אותו הם צריכים בכדי למלא את תפקידם במערכת.

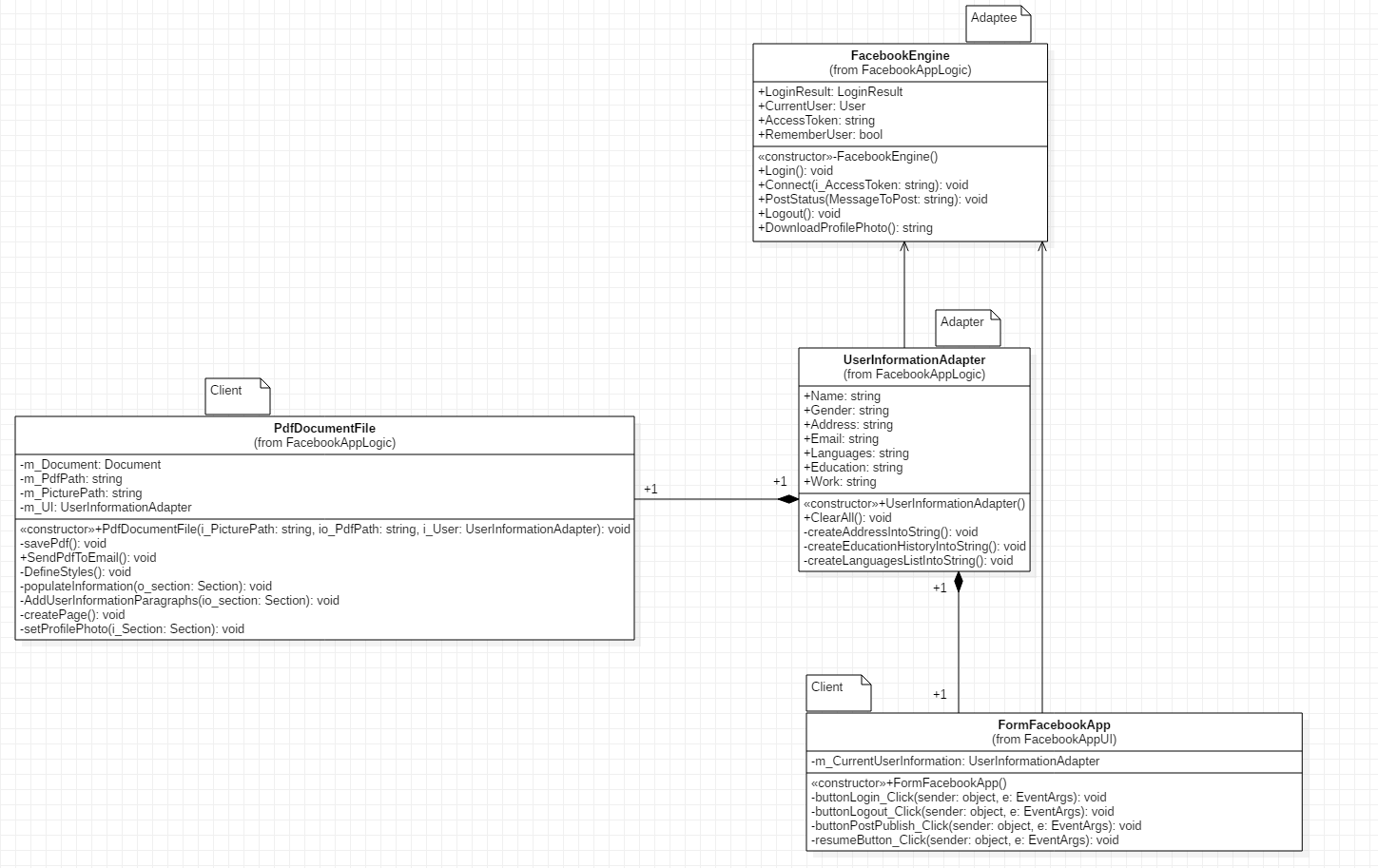
בפרט, לרכיב (המחלקה PDFDocumentFile ) המערכת , אשר יוצרת את טופס הPDF ויוצקת אליו את המידע הניתן מן המערכת.

כאמור, המימוש הינו בתור מחלקת ממשק אשר תגשר בין FacebookEngine לבין PDFDocumentFile, ושמה UserInformationAdapter.

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – [Singleton]

* סיבת הבחירה / שימוש בתבנית:

המחלקה FacebookEngine מתפקדת כמנוע העובד מול שרתי פייסבוק, מחלקה זו עובדת מול מספר רכיבים שונים במקביל, כגון רכיבים הקשורים בממשק משתמש וכן בפיצר יצירת קורות חיים וכדומה. מצב זה היה בעייתי כי זה איפשר שכפולים רבים של המנוע, אשר כל שכפול מתקשר עם שרתי בפייסבוק בפני עצמו, דבר העלול לגרום לבלבול רב בקוד.

בחרנו לממש את המנוע באמצעות תבנית סינגלטון, בכדי שרכיב אחד בלבד יקשר את רכיבי המערכת השונים עם שרתי פייסבוק ויאפשר להם גישה לתכונות המופע היחיד בהן הוא מעוניין. כמו כן מנענו שכפול מופעים אשר עשויים היו לגרום לבלבול למתכנתי האפליקציה בעתיד.

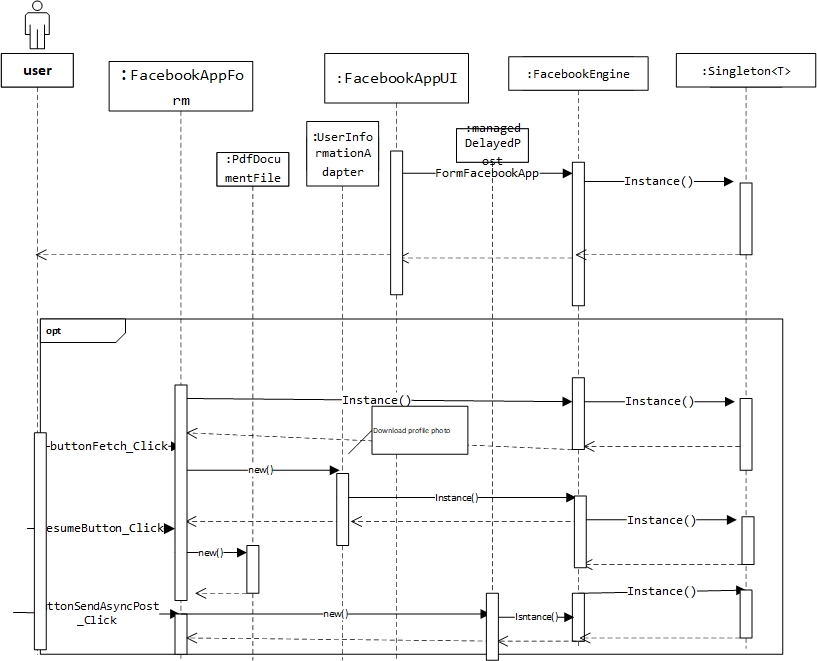
* אופן המימוש:

המחלקה FacebookEngine היא סינגלטון, אשר ממומש באמצעות המחלקה הגלובלית Singleton<T>. הבנאי של FacebookEngine מוגדר כ,PRIVATE בשביל לקבל גישה אל הסינגלטון יש לבקש גישה למופע באמצעות המתודה INSTANCE, אשר מפעילה בתורה את המתודה INSTANCE של Singleton<T>.

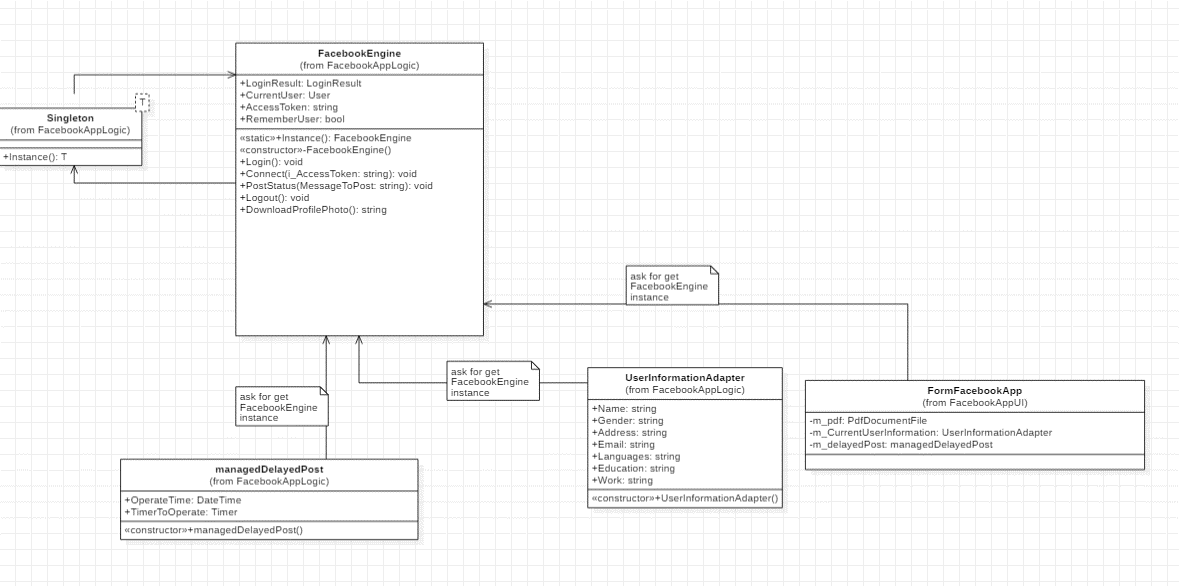
כמו כן, סינגלטון זה תומך בריבוי תהליכים.

* Sequence Diagram

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



* Class Diagram



**עבודה אסינכרונית –**

נעשה שימוש בתכנות א-סינכרוני, בכל טעינת הרכיבים מן השרת המרוחק (פייסבוק) , וזאת בשל רצוננו לא להאט את מהלך טעינת האפליקציה, ו/או הפרעה לריצתה בזמן זה.

המיקום בקוד שבו נעשה שימוש בתכנות א-סינכרוני:

בתוך מחלקת FacebookAppForm:

שיטת fetchLoggedInUserThread המאתחלת תהליך חדש והקוראת לשיטת fetchLoggedInUser, שבתורה קוראת לכל בשיטות המאכלסות את טופס האפליקציה במידע והן :

populateProfilePicture()

populatePosts()

populatePhotos()

populateGroups()

populateFriends()

populateIntro()

חשוב לציין שבכל אחת מהן יש שימוש ב – Invoke()

כמו כן, בפיצר שליחת פוסט מתוזמן שלנו , נעשה שימוש במחלקה System.Timers וזאת בכדי לתזמן את מועד שליחת הפוסט, מבלי להפריע לתהליכון הראשי.

**עבודה עם Data Binding –**

נעשה שימוש ב Two Way Data Binding במחלקה FacebookAppForm בשדות למילוי הפרטים לצורך יצירת קורות חיים עם פרטי המשתמש שנלקחו מהפייסבוק, זה נעשה בעזרת אובייקט המחלקה UserInformationAdapter.

אופן השימוש: שדות FacebookAppForm לוקחות מידע ממופע המחלקה UserInformationAdapte הנמצא ב FacebookAppForm (אשר לקח את המידע מפייסבוק והמירו לטקסט) וכאשר מתבצע שינוי בשדות מופע מחלקת ,UserInformationAdapterהמופע מתעדכן.

לאחר מכן,המופע UserInformationAdapter מועבר למחלקה PdfDocumentFile ושם יוצרים קורות חיים בעזרת המידע של המופע הנ"ל.