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

* שליחת אימייל עצמי למען תזכורת
* הצגה של רשימת החברים הממוינת לפי כמות הלייקים שכל חבר נתן למשתמש

### תבנית מס' 1 – [Singleton]

* סיבת הבחירה / שימוש בתבנית:

רצינו שיווצר מופע יחיד של שליחת האימייל מכיוון שהיוזר ממנו נשלח המייל הוא יחיד לאורך כל התוכנית וגם נתונים נוספים(כגון: server port, smtp server…)לכן לא נרצה לשכפל אותו.

* אופן המימוש:

האובייקט EmailSender מכיל רפרנס לעצמו, מכיל instance property כך שבפעם הראשונה שנקרא אליו, הוא ייצור לנו אובייקט(ע"י private ctor) שעטוף במנעול

אשר יוודא שהאובייקט ייוצר פעם אחת בלבד.

כל קריאה נוספת ל instance תחזיר את האובייקט שנוצר בפעם הראשונה.

* Sequence Diagram

A diagram of a diagram

Description automatically generated

* Class Diagram

A diagram of email mailing

Description automatically generated

### תבנית מס' 2 – [Adapter]

* סיבת הבחירה / שימוש בתבנית:

השתמשנו ב Adapter עבור אובייקט שיצרנו(Birthday)

Birthday מקבל DateTime בעת היצירה שלו ולכן יצרנו Adapter בשם BirtdayAdapter

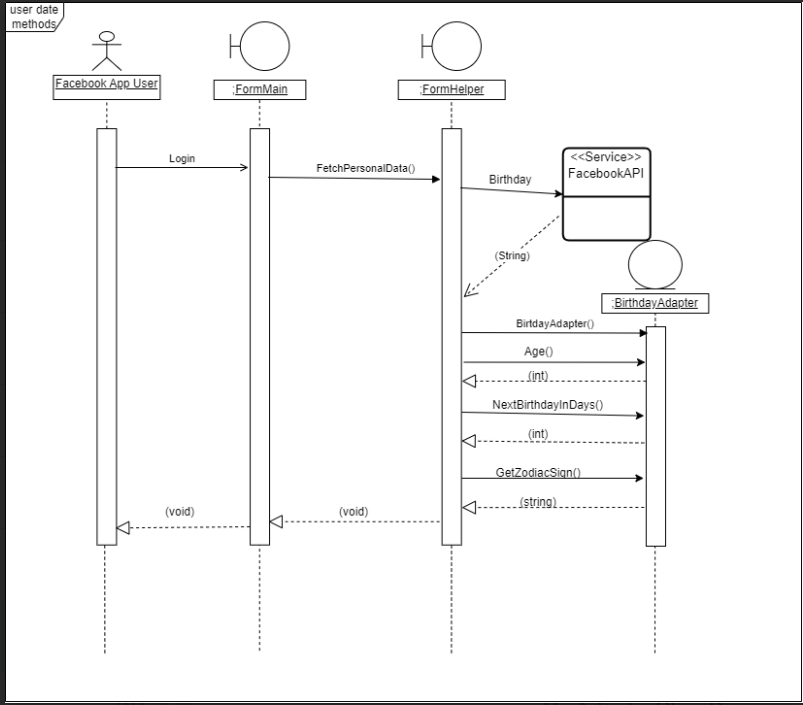
על מנת להדפיס מידע אודות תאריך הלידה של המשתמש.

* אופן המימוש:

יצרנו אובייקט BirthdayAdapter שמקבל תאריך כ- string (בפורמט שמוחזר מ- FacebookAPI) , הופך את ה- string ל- DateTime ובעזרתו יוצר אובייקט מסוג Birthday שמוחזק בקומפוזיציה.

לשניהם אותם מתודות ושימוש זהה.

* Sequence Diagram



* Class Diagram

A diagram of a computer

Description automatically generated

### תבנית מס' 3 – [Proxy]

* סיבת הבחירה / שימוש בתבנית:

יצרנו שני סוגי Proxy עבור האובייקטים (User, Page) שגיא סיפק לנו.

בחרנו להשתמש ב Proxy הנ"ל על מנת לצמצם גישות מיותרות לשרת ולחסוך בביצועים.

* אופן המימוש:

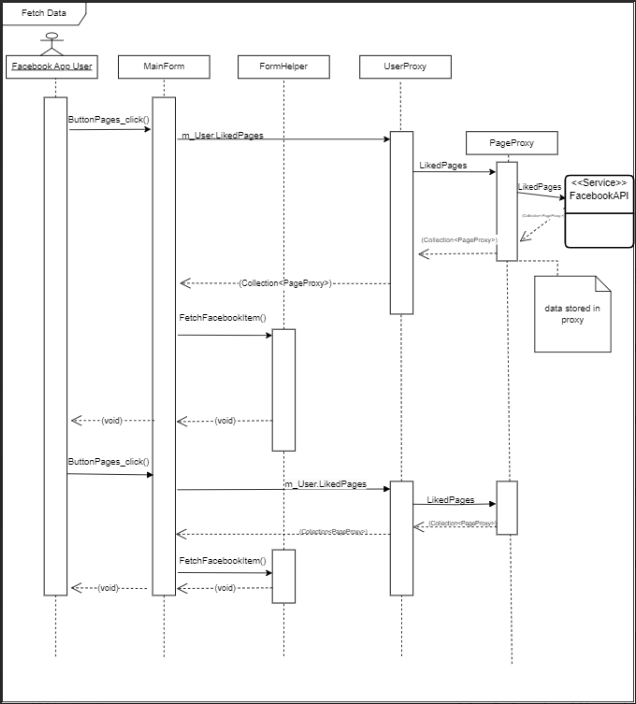
יצרנו אובייקט Interface עם properties מסויימים שחייבים להיות בכל אחד מהאובייקטים שהרחבנו אותם.

בנוסף כל אחד מ Proxy ירש מן ה-Interface (על מנת שנוכל להשתמש בהם בצורה דומה)

והוא יכיל את האובייקט המקורי (User,Page) תלוי באיזה מימוש מדובר ו Members נוספים שכל אחד מהם יהיה קולקציה של ObjectModel אחר.

עבור כל קולקציה שכזאת יש Property באותו שם של האובייקט המקורי(למשל LikedPages) כך שבכל פעם לאחר שיבוצעו 5 לחיצות של בקשת מידע, הוא ינסה לגשת

* Sequence Diagram



* Class Diagram

A screenshot of a computer

Description automatically generated

### עבודה אסינכרונית עם ממשק המשתמש

יש לנו עבודה אסינכרונית במקומות שבהן יש קריאות לשרת, במקום לחכות לשרת מאחר וקריאות לשרת לוקחות זמן, את קריאות אלו יצרנו ב-thread נפרד על מנת שהמשתמש יוכל לבצע פעולות נוספות במקום להמתין לתגובת השרת.  
את הקוד ניתן למצוא במחלקה FormMain בעיקר במתודות Button\_Click, CompleteUIFromFacebookData.

### עבודה עם Data Binding

מימשנו עבודה זו באזור הראשי, label המכיל את פרטי האלבום שעליו לחץ המשתמש(בתוך ה-ListBox המתאים) באמצעות Data Binding אשר מכיל כ-data source אובייקט מסוג Album.

אופן המימוש מכיל 2-way Data Binding כך שהמשתמש יוכל לערוך את המידע דרך TextBox ו TimePicker .

באזור ה-Profile, label המכיל את פרטי הקבוצה עליה לחץ המשתמש(בתוך ה- ListBox המתאים)

באמצעות Data Binding אשר מכיל כ- data sourceאובייקט מסוג Group.

אופן המימוש מכיל 2-way Data Binding כך שהמשתמש יוכל לערוך את המידע דרך TextBox.

שני המימושים הנ"ל נמצאים ב- FormMain.