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

1. פיצ'ר המאפשר למשתמש לסנן את חבריו לפי אפשרויות מובנות מראש, הכוללות סינון לפי מין, חברים שדוברים את אותה השפה, חברים אשר נולדו באותו החודש, וחברים אשר הפרש השנים בגיל עם המשתמש הוא X (ניתן לבחירה באפליקציה). המשתמש בוחר מתוך ארבע האפשרויות את הסינון המבוקש, כאשר חלקם כוללים תתי-בחירות (כגון שפה מתוך הרשימה או בחירת שנה עבור ההפרש), ואז מוצגת למשתמש רשימה של החברים הרלוונטיים לפי הסינון שנבחר (בעת לחיצה על שם מסוים מתוך הרשימה, מופיעה התמונה של אותו חבר שנבחר).
2. פיצ'ר שמטרתו לתת למשתמש המלצות מחבריו בפייסבוק לטובת ביקור בארץ ובעולם. מכיל רשימה עם כל הערים שבהן גרים חבריו, כאשר בעת בחירה בעיר מסוימת תוצג למשתמש רשימה נוספת של כל החברים שגרים באותה העיר (לחיצה על חבר מסוים תראה את תמונת הפרופיל שלו). בעת בחירת העיר מתוך הרשימה, מקבל המשתמש מידע אודות נתוני מזג האוויר והמצב הנוכחי בעיר (באמצעות api חיצוני של חיזוי מזג אוויר), כגון: טמפרטורה, אחוזי לחות, זמני שקיעה וזריחה. בנוסף, יוצג למשתמש לינק לאתר ויקיפדיה, שם יוכל לקבל מידע נוסף אודות העיר שבה מעוניין לבקר.

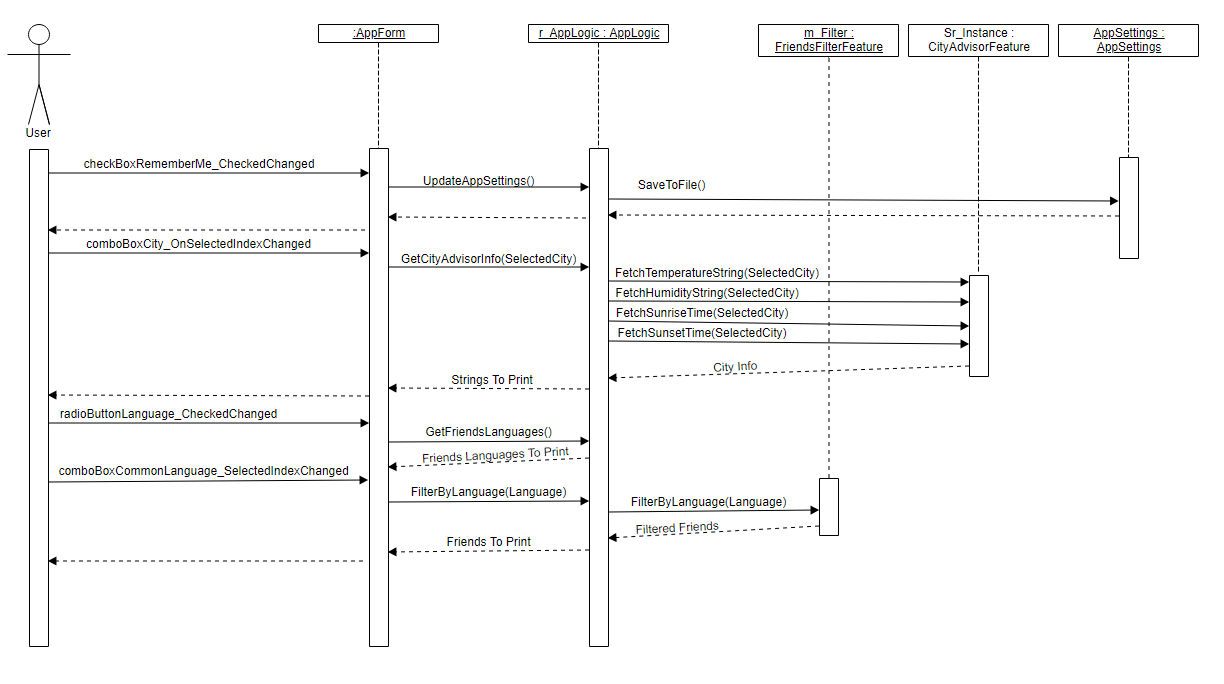
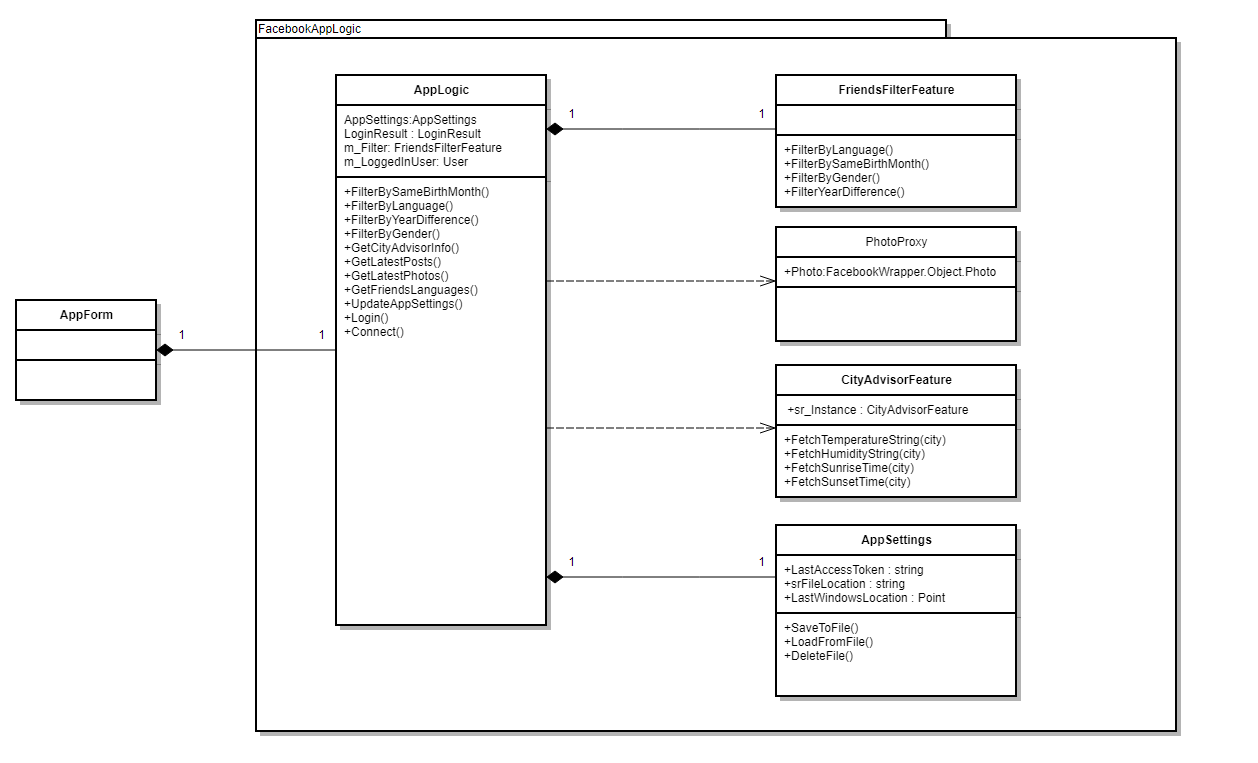
### תבנית מס' 1 – Facade:

* **סיבת הבחירה / שימוש בתבנית:**

באפליקציה ישנה חלוקה בין החלק של הלוגיקה לבין ממשק המשתמש. החלק הלוגי מחולק למספר מחלקות, כאשר כל אחת מממשת חלקים נפרדים באפליקציה. על מנת להקל על השימוש בחלק הלוגי, החלטנו להשתמש בFacade.

* **אופן המימוש:**

המחלקה AppLogic אחראית לפנות לכל שאר המחלקות ולאגד במקום אחד (לחשוף) את כל המתודות הרלוונטיות עבור ממשק המשתמש, כך שלממשק המשתמש ישנו רק רפרנס לFacade ובאמצעותו הוא משיג את הפונקציונליות הנדרשת.

* **Sequence Diagram:**
* **Class Diagram:**
* **ההקבלה בין המחלקות בתבנית:**

Client = AppForm

Façade = AppLogic

Library = FacebookAppLogic

SubSystems = FriendsFilterFeature, AppSettings, CityAdvisorFeature, PhotoProxy

### תבנית מס' 2 – Proxy:

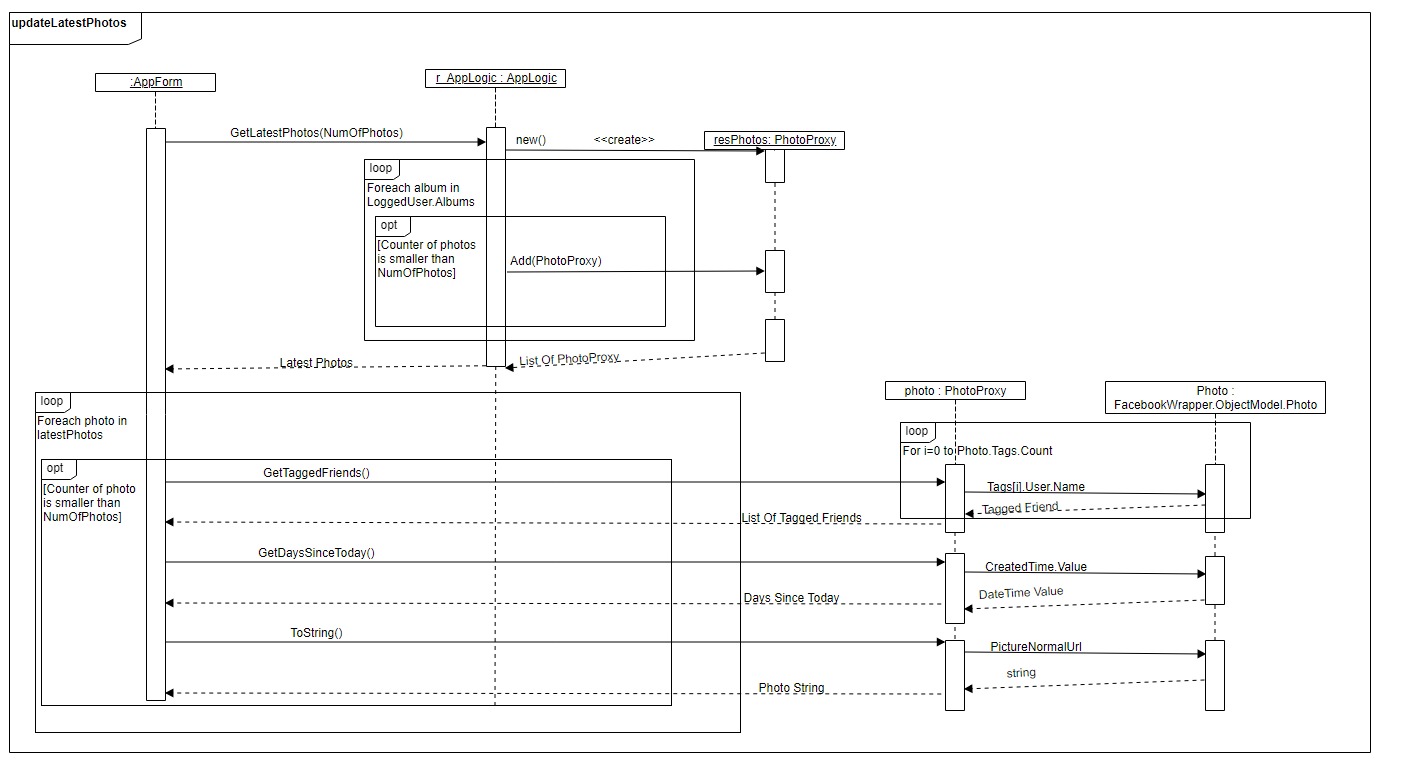
* **סיבת הבחירה / שימוש בתבנית:**

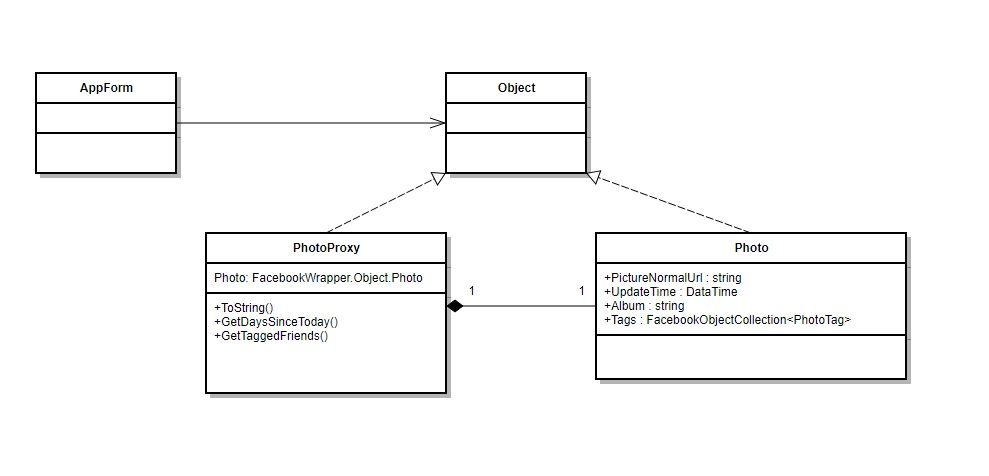
בעמוד הראשי של האפליקציה ישנו שימוש בתמונות העדכניות ביותר של המשתמש. עבור כל תמונה ממשק המשתמש מציג את התמונה עצמה, תאריך היצירה והאנשים המתוייגים בה. במימוש שהיה עד כה לאובייקט עצמו לא היה מספיק מידע על מנת לספק את הנדרש, ולכן באמצעות PhotoProxy הוספנו את הפונקציונליות הנוספת שנדרשה, ובכך חסכנו מצד ממשק המשתמש להיות זה שיממש אותה.

* **אופן המימוש:**

עד כה התמונות היו מיוצגות כאובייקט מטיפוס FacebookWrapper.Object.Photo, וכעת עטפנו אותו במחלקה PhotoProxy (באמצעות רפרנס) ומימשנו בתוכה את כל הפונקציונליות הנוספת הנדרשת מצד ממשק המשתמש. כך ברגע שהוא פונה לFacade על מנת לקבל את התמונות, הן מגיעות בצורת PhotoProxy ומראש מספקות את הפונקציונליות הנדרשת.

* **Sequence Diagram:**



* **Class Diagram:**
* **ההקבלה בין המחלקות בתבנית:**

Client = AppForm

Proxy = PhotoProxy

RealSubject = Photo

Interface = Object

### תבנית מס' 3 – Singleton:

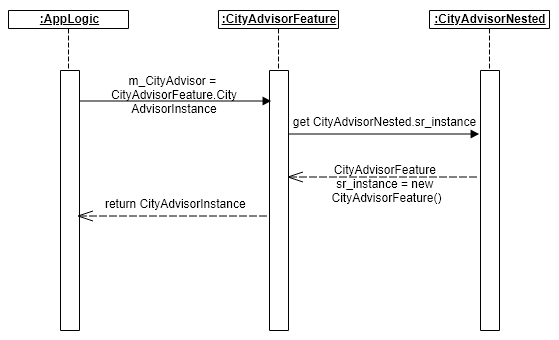
* **סיבת הבחירה / שימוש בתבנית:**

רצינו להכיר טוב יותר את שיטות המימוש השונות עבור התבנית הידועה הנ"ל, ולנסות ולתרגל זאת בעצמנו על-מנת להבין את היתרונות והחסרונות של כל שיטה. מטרת השימוש במקרה שלנו הייתה ליצור מופע יחיד של מחלקת ה-CityAdvisorFeature, כך שבכל שינוי של העיר המבוקשת יקבל המשתמש את אותו המופע, ללא צורך ביצירת מופע חדש של המחלקה. המוטיבציה היא בין היתר מניעת האפשרות של יצירת מופעים מיותרים של המחלקה, שכן בכל מקרה באפליקציה שלנו ניתן להציג מידע רק עבור עיר אחת בכל פעם.

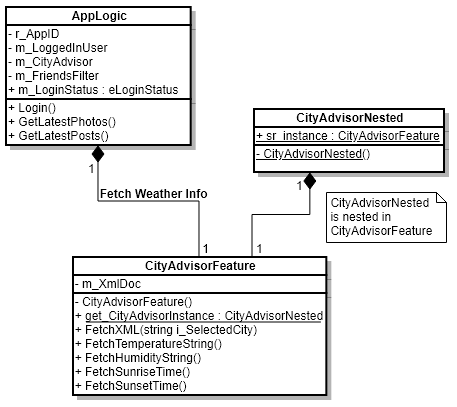
* **אופן המימוש:**

את התבנית מימשנו כאמור במחלקה CityAdvisorFeature, באופן fully-lazy thread-safe וללא שימוש במנעולים. תחילה, הפכנו את המחלקה להיות internal sealed, ומימשנו לה בנאי private חסר-פרמטרים שיחליף את הבנאי ה-public הדיפולטיבי. כמו כן יצרנו מחלקה מקוננת בשם CityAdvisorNested בעלת גישה private, עם בנאי סטטי, שהיא זו שמחזיקה בפועל את המופע היחיד של המחלקה הראשית, אשר יהיה סטטי בעצמו. בכך יצרנו מצב שבו בפנייה ראשונה של מחלקת הלוגיקה לאותו מופע סטטי של המחלקה CityAdvisorFeature, מופעל הבנאי ה-private של המחלקה באמצעות המחלקה המקוננת שיכולה לגשת אליו, ואז כדי ליצור את המופע של המחלקה אנחנו מוכרחים לעבור דרך הבנאי הסטטי במחלקה המקוננת, שאליו ניתן להיכנס פעם אחת בלבד, ורק בעת פנייה למשתנה הסטטי שאותו רוצים לאתחל. כעת, בכל פנייה חוזרת למופע זה, יוחזר אותו מופע שנוצר כאן באופן בטוח ו"עצל" כפי שתיארנו, ללא יצירת מופעים נוספים.

* **Sequence Diagram:**



* **Class Diagram:**



* **ההקבלה בין המחלקות בתבנית:**

Singleton = CityAdvisorFeature

Singleton Nested Class = CityAdvisorNested

### עבודה אסינכרונית:

* **היכן בקוד נעשה שימוש:**

השימוש נעשה במחלקה AppForm המייצגת את החלק של ממשק המשתמש באפליקציה.

בתוכה המימוש היה במתודות הבאות:

updateLatestPhotos, updateLatestPosts, loadFriendsList, loadCityAdvisorInfo, fetchAlbums (כל המתודות האחראיות על הבאת מידע כבד מהשרת של פייסבוק).

* **אופן המימוש:**

עבור כל אחת מהמתודות שצויינו לעיל, הפעלנו Thread נפרד שיהיה אחראי לטפל בהן. בתוך כל מתודה, במקומות אשר ישנו שימוש בפקדים הפעלנו את המתודה Invoke על מנת לאפשר לThreads השונים להפעיל פונקציונליות אשר קיימת רק בMain Thread. לצורך כך בתוך Invoke השתמשנו בLambda Expressions מכיוון שהיה מדובר על פעולה אחת בכל אחד מהם, ובכך נחסך הצורך מפונקציות נפרדות + התאפשרה גישה לאובייקטים מקומיים בתוך כל אחת מהמתודות.

* **מדוע השתמשנו:**

כל אחת מהמתודות שהפעלנו באמצעות Thread נפרד אחראית על הבאת מידע מהשרתים של פייסבוק (תמונות, פוסטים , אלבומים, רשימת חברים). כל גישה לשרתים כרוכה בהמתנה של כמה שניות, אשר בהן האפליקציה ''קופאת'' ושאר התהליכים לא ממשיכים, כלומר נוצר מצב שבו המשתמש פותח את האפליקציה והכל תקוע. על מנת למנוע זאת, ובעצם לאפשר לuser להשתמש בכל הפונקציות של האפליקציה אשר כן הצליחו להיטען בכל נקודת זמן, ישנו צורך בThreads. כמו כן, כל מתודה מופעלת בThread נפרד בכדי שלמשתמש יהיה מוצג המידע שנטען עד כה, ולא יצטרך לחכות לכל התהליכים ה''איטיים'' להסתיים.

### עבודה עם Data Binding:

* **היכן בקוד נעשה שימוש:**

השימוש נעשה בטאב 'My Albums', שבו ניתן לייבא בלחיצה על כפתור 'Fetch Albums' את כל האלבומים של היוזר המחובר באפליקציה, ולקבל פרטים על כ"א מהם ע"י בחירה של אלבום ספציפי ב-ListBox.

### Test Users:

הסיסמא לכל המשתמשים היא : Test1234

שמות המשתמשים הם:

[elliot\_pddykpa\_alderson@tfbnw.net](mailto:elliot_pddykpa_alderson@tfbnw.net)

[hannah\_tvexyzn\_baker@tfbnw.net](mailto:hannah_tvexyzn_baker@tfbnw.net)

[walter\_idmbcqb\_white@tfbnw.net](mailto:walter_idmbcqb_white@tfbnw.net)

[dolores\_vanvirn\_abernathy@tfbnw.net](mailto:dolores_vanvirn_abernathy@tfbnw.net)

[rick\_nnqhwnk\_sanchez@tfbnw.net](mailto:rick_nnqhwnk_sanchez@tfbnw.net)