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

* :Events Planner

בשביל להגיע לפיצ'ר הזה: צריך לעשות Login וללחוץ על לשונית "מתכנן אירועים".

בלוח שנה מודגשים תאריכים אשר בהם יש יום הולדת או אירוע אחר של המשתמש.

לחיצה על אחד התאריכים תציג למשתמש את רשימת האירועים באותו יום.

לחיצה על אירוע ספציפי תציג למשתמש מפה עם מיקום האירוע.

הפיצ'ר הזה נועד לעזור למשתמש להיות בעניינים, להיות מעודכן על האירועים הקרובים אליו ומעניק למשתמש אפשרות לנהל את הזמן בצורה קלה ונכונה יותר.

את הפיצ'ר ניתן למצוא בקוד תחת:

FacebookApp.Logic במחלקה EventPlanner

FacebookApp.UI במחלקה AppForm

* Wish Happy Birthday To Your Real Friends:

בשביל להגיע לפיצ'ר הזה: צריך לעשות Login וללחוץ על לשונית "ימי הולדת".

לחיצה על כפתור "לחפש חברים" תציג את רשימת החברים שגם יש להם יום הולדת בחודש הזה והם גם פרסמו פוסט על הקיר של המשתמש ביום ההולדת שלו.

לחיצה על הכפתור "לאחל מזל טוב ליום הולדת" מאפשרת לפרסם פוסט על הקיר של החבר הנבחר לכבוד יום הולדתו.

הפיצ'ר עוזר למשתמש למצוא את חבריו האמיתיים שכותבים לו איחולים ביום הולדתו ולעזור לו להיות מוכן לקראת יום הולדת של חבריו.

את הפיצ'ר ניתן למצוא בקוד תחת:

FacebookApp.Logic במחלקה RealFriendsListFinder

FacebookApp.UI במחלקה AppForm

### תבנית מס' 1 – Strategy

* סיבת הבחירה / שימוש בתבנית:

למשתמש יש אופציה לראות את כל חברים שלו, אך בסדר שהם נקלטים מהשרת. בשביל חיפוש יותר נוח למשתמש של חבר מסוים – נכון למיין את החברים בסדר עולה/יורד.

שני המיונים נבדלים רק בתנאי המיון (שורת קוד אחת שקובעת את סדר המיון). לכן בחרנו להשתמש בתבנית זו, על מנת להימנע משכפול קוד ותמיכה באפשרויות מיון נוספות בעתיד.

* אופן המימוש:
* יצרנו ממשק IUsersSortingStrategy אשר בו מתודה אבסטרקטית בוליינית, שמקבלת שני User-ים ומחזירה true, אם צריך להחליף בניהם במקומות (כלומר אם סדר הנוכחי שלהם במערך לא נכון).
* יצרנו שני מחלקות למימוש הממשק: AZSorter – למיון משתמשים בסדר יורד לפי א"ב, ZASorter – למיון משתמשים בסדר עולה לפי א"ב.
* יצרנו מחלקה UsersSorter – שמכילה אלגוריתם למיון User-ים.

בC’tor של המחלקה מאתחלים את UsersSortingStrategy שהוא מסוג IUsersSortingStrategy, ובו משתמשים על מנת לדעת אם צריך להחליף שני User-ים באלגוריתם.

* בנוסף הוספנו שני כפתורים A->Z וZ->A בסימנית Friends למיון החברים בהתאם

ה"שחקנים" בתבנית:

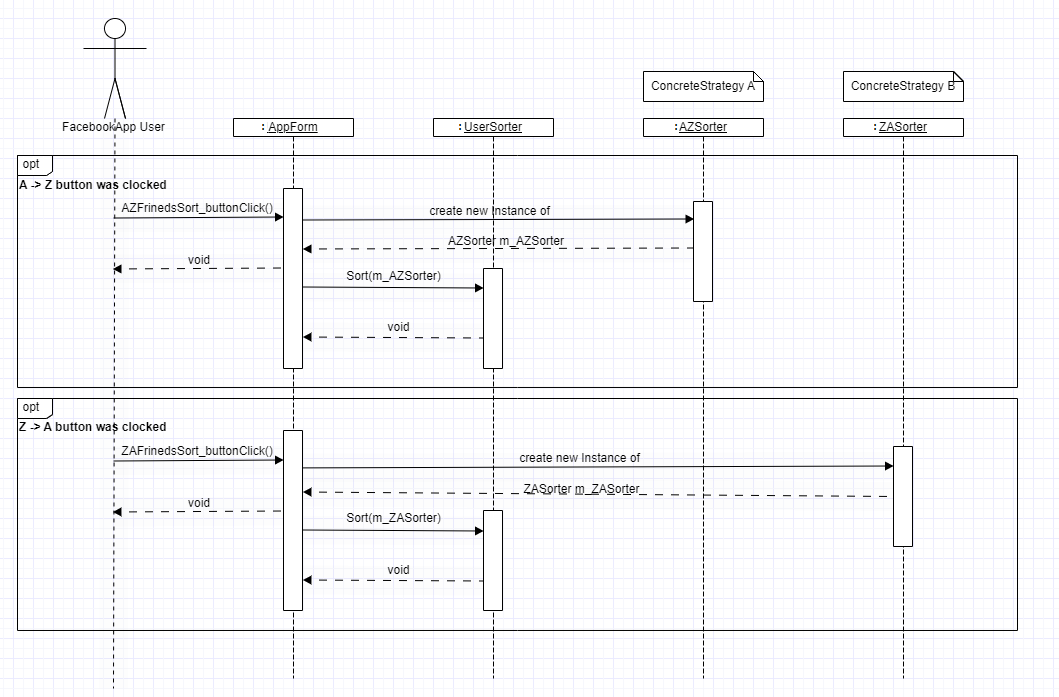
• Context: fetchAndSortFriends method in AppForm.cs

• Strategy: IUsersSortingStrategy

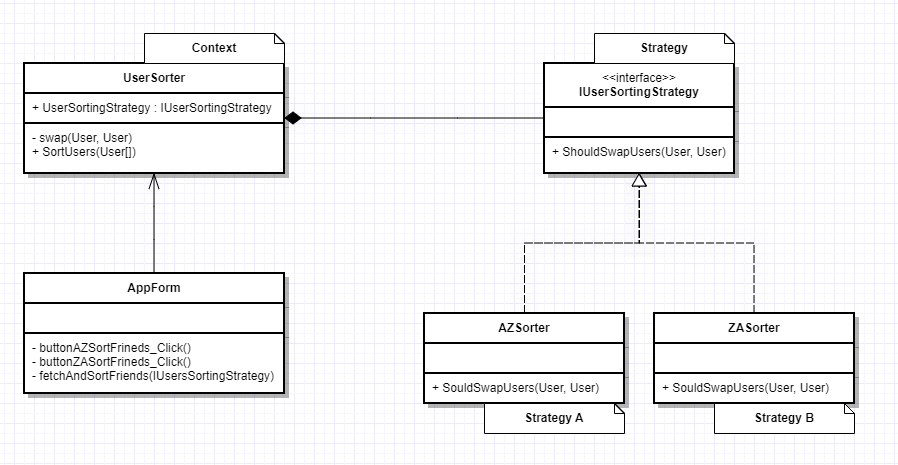
• ConcreteStrategyA: AZSorter

• ConcreteStrategyB: ZASorter

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 –Iterator

* סיבת הבחירה / שימוש בתבנית:

מחלקת User שנמצאת ב-FacebookWrapper נותנת אפשרות לגשת לרפרנס של המופע, כלומר יש אפשרות לבצע שינויים בUser-ים עצמם, וזה לא בטיחותי. לדוגמא, כשמשתמש רוצה לראות רשימת החברים, אנחנו לא רוצים שתהיה לו האפשרות למחוק חברים.

לכן בחרנו לממש תבנית זו וכך לבטח את ההרשאות.

* אופן המימוש:

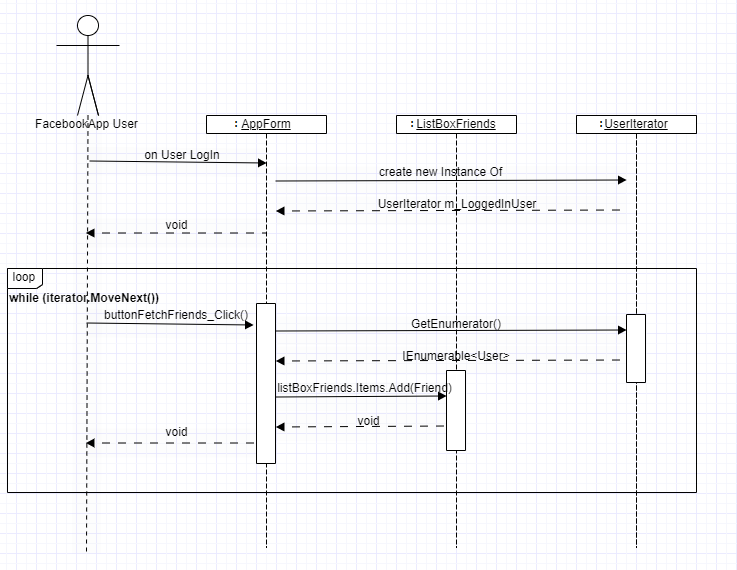
יצרנו מחלקה בשם UserIterator שמתחזה ל-User, אך חושפת רק iterator שיודע לסרוק את האוסף.

המחלקה מכילה את כל השדות הדרושים רק עם Get Property,

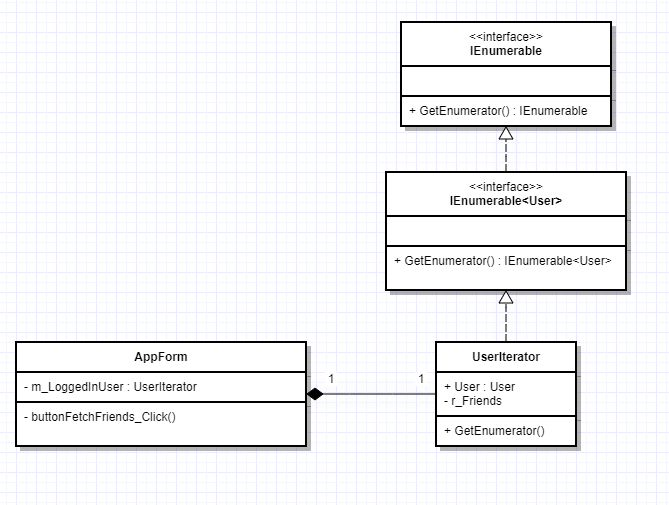
יורשת מ- IEnumerable<User> ומשתמשת בפיצ'ר yield return.

"השחקנים" בתבנית:

* Aggregate: UserIterator
* Iterator: UserIterator:GetEnumerator()
* Client: AppForm, AppLogic
* Sequence Diagram



* Class Diagram



### תבנית מס' 3 –Observer

* סיבת הבחירה / שימוש בתבנית:

הוספנו לאפליקציה שלנו פיצ'ר חדש: הצגת רשימת החברים שחוגגים יומולדת היום. רשימה זו נמצאת על המסך הראשי של האפליקציה.

כאשר מימשנו את הפיצ'ר Wish Happy Birthday To Your Real Friends, שכחלק ממנו צריך להציג רשימת החברים שיש להם יומולדת החודש, הייתה לנו דילמה: כיצד הרשימה תדע מי מהחברים יכנס לרשימה? כמובן שאז פתרנו את הבעיה בצורה שונה, אך עכשיו ניתן לפתור אותה בעזרת תבנית Observer.

במימוש החדש הרשימה תהיה "המקשיבה" וכל אחד מהחברים יהיו "המודיעים". ברגע שמשתמש מתחבר לאפליקציה, כל אחד מחבריו יודיע לרשימה האם הוא חוגג יומולדת היום.

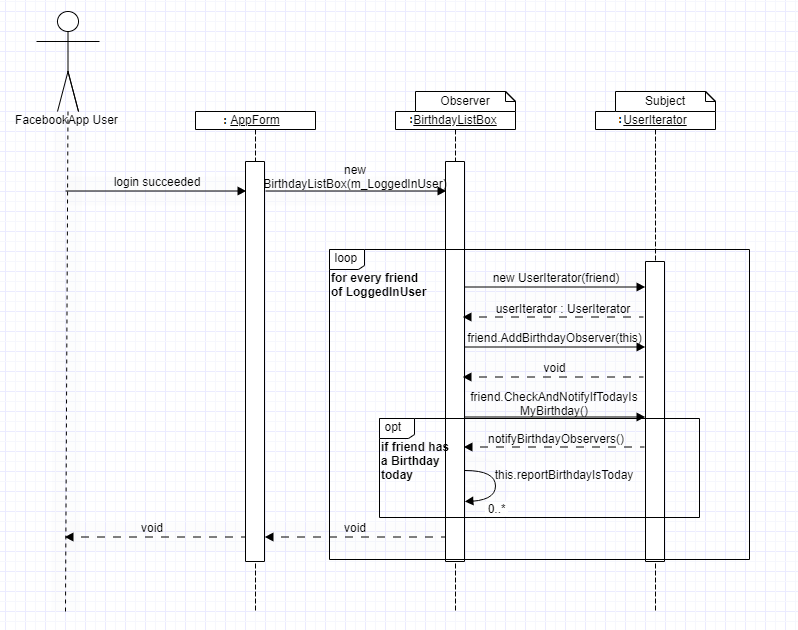
* אופן המימוש:
* יצרנו ממשק IBirthdayObserver ב- Facebook.Logic, אשר בו מתודה void-ית שמקבלת User ומדווחת למקשיב שלמשתמש הזה יש יומולדת היום.
* יצרנו מחלקה BirthdayListBox ב- Facebook.AI היורשת מ- ListBox ומממשת את הממשק, כלומר זאת הרשימה של החברים שחוגגים יומולדת היום.
* הוספנו למחלקה UserIterator שדה IsBirthdayToday מסוג bool
* הוספנו למחלקה UserIterator שדה List<IBirthdayObserver>
* הוספנו למחלקה UserIterator מתודות שמאפשרות למשתמשים להוסיף/למחוק את עצמם מרשימת המודיעים.

"השחקנים" בתבנית:

• Observer: BirthdayListBox

• Subject: UserIterator

* Sequence Diagram



* Class Diagram

