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

* Automate Facebook Tasks – למשתמש ישנה האפשרות לתזמן פעולות פייסבוקיות (העלאת תמונה, סטטוס או לינק) והפעולות יתבצעו במועד שנקבע על ידי המשתמש מראש.
* Shick O Shook – הגרלת חבר מרשימת החברים והצגת תמונה שלו ופרטים אודותיו. ניתן להחליף בין תמונותיו בלחיצה על התמונה, לבסוף ניתן לדרג "שיק" או "שוק" על החבר. לחיצה על אחד מהכפתורים הנ"ל תעלה סטטוס ובו רשומה הודעה בנוגע לבחירת היוזר.
* Calculate Statistics About Friends – המשתמש מקבל נתונים סטטיסטיים אודות רשימת החברים שלו (החבר הכי פעיל, יחס גברים\נשים, גילאים וכו').

### תבנית מס' 1 – Template Method

* סיבת הבחירה / שימוש בתבנית:

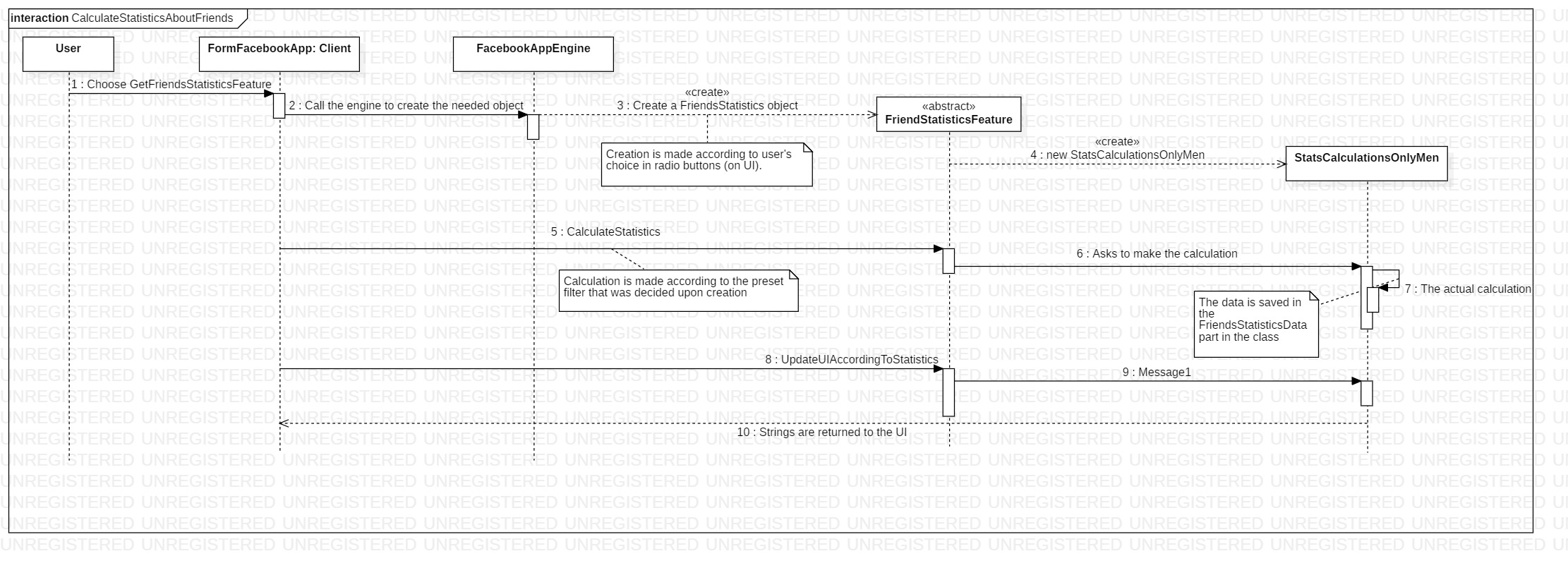
בפיצ'ר Calculate Statistics About Friends , המשתמש יכול לבחור לקבל נתונים המתבססים על רשימת החברים שלו. רצינו שתינתן למשתמש האפשרות לבחור בזמן ריצה מבין אפשרויות לסינון המשתמשים, על מנת להציג לו נתונים המתבססים על רשימת חברים מותאמת ע"פ דרישתו. למשל – קבלת כל הנתונים הסטטיסטיים רק על החברים ממין זכר וכו'.  
בפועל – רשימת החברים עליה מתבצע איסוף הנתונים, משתנה בהתאם לבחירת המשתמש ובכך מושגת התוצאה הנדרשת.

* אופן המימוש:

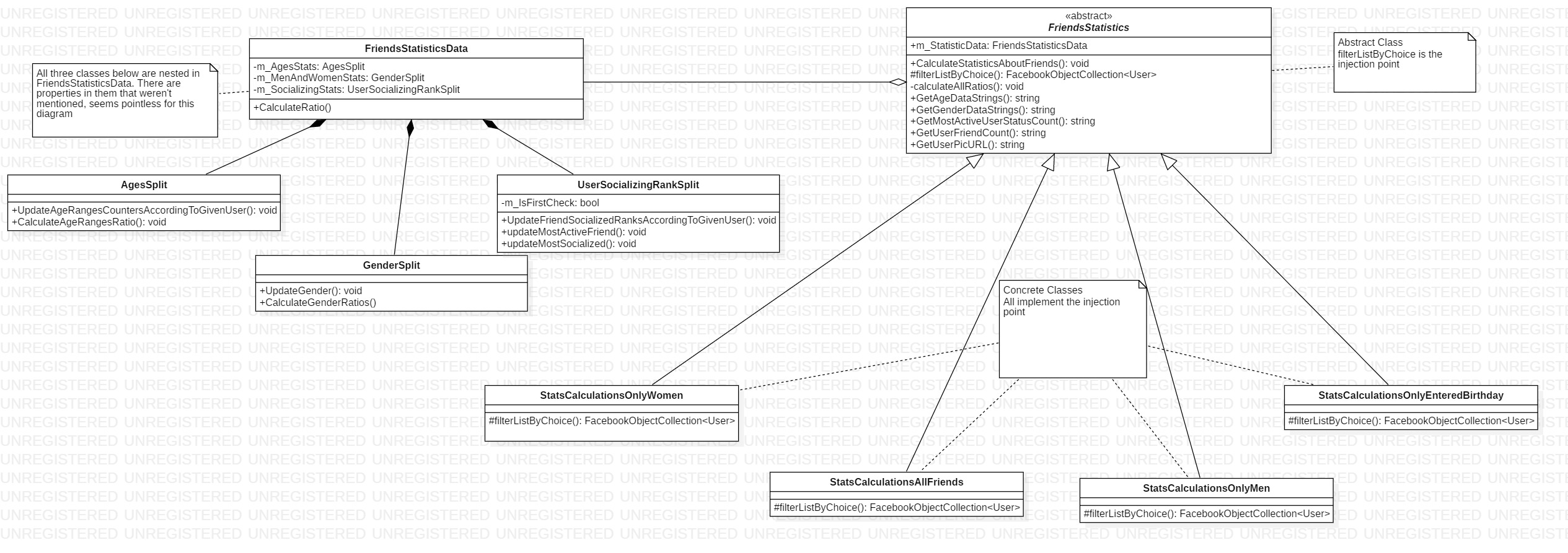
הגדרנו מחלקה אבסטרקטית FriendStatistics אשר מכילה בתוכה רפרנס ל-FriendStatisticsData (האובייקט אשר מכיל את הנתונים הסטטיסטיים עצמם) וכן הגדרנו במחלקה מתודה CalculateStatisticsAboutFrineds אשר אחראית על ביצוע החישובים והזנתם אל האובייקטים המתאימים (מוכלים בתוך FriendStatisticsData). בתוך המתודה הנ"ל , ישנה שורת injection point – שורה בה מתבצעת קריאה למתודה אבסטרקטית filterListByChoice אשר אחראית לעשות מניפולציה על רשימת החברים ע"פ בחירת המשתמש (למשל: סינון הגברים בלבד).  
הגדרנו מחלקות StatsCalculationsOnlyMen, StatsCalculationsOnlyWomen, StatsCalculationsAllFriends, StatsCalculationsOnlyEnteredBirthday אשר יורשות מהמחלקה האבסטרקטית וכל אחת מהן מציעה מימוש שונה עבור המתודה filterListByChoice.

בשכבת ה-UI, המתמש בוחר (באמצעות RadioButtons) באיזה סוג פילטור הוא מעוניין וברגע שהוא לוחץ על הכפתור GetStats נוצר האובייקט הרלוונטי (אחד מ4 המחלקות היורשות) ומתבצעות הפעולות הנדרשות.

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Decorator

* סיבת הבחירה / שימוש בתבנית:

בפיצ'ר Automate Facebook Tasks אנו יוצרים אובייקט אשר משלב בין טיימר לבחירת המשתמש לבין פעולה "פייסבוקית" מתוזמנת מראש (אוביקט מסוג TimedComponent).

רצינו לממש פקד ממשק משתמש באופן כזה שניתן יהיה להרחיב את תוכנו ו\או הדברים אשר מתרחשים בעת ביצוע הפעולה (למשל: הוספת אנימציה). משום כך, בחרנו לממש את תבנית Decorator במקרה הנ"ל.

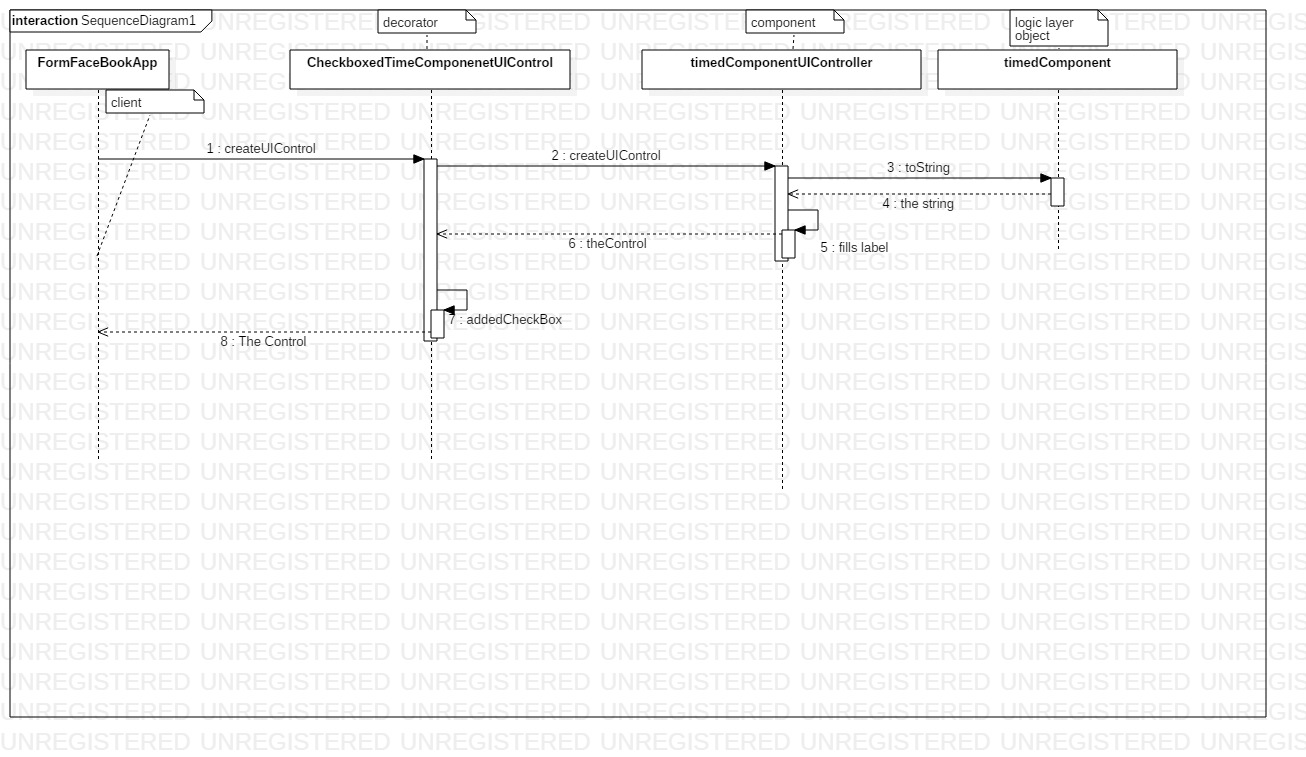
* אופן המימוש:

יצרנו ממשק בשם IControl אשר גם הרכיב הקונקרטי(TimedComponentUIControl) וגם האובייקט האבסטרקטי(DecortorTimedComponentUIControl ) מממשים.

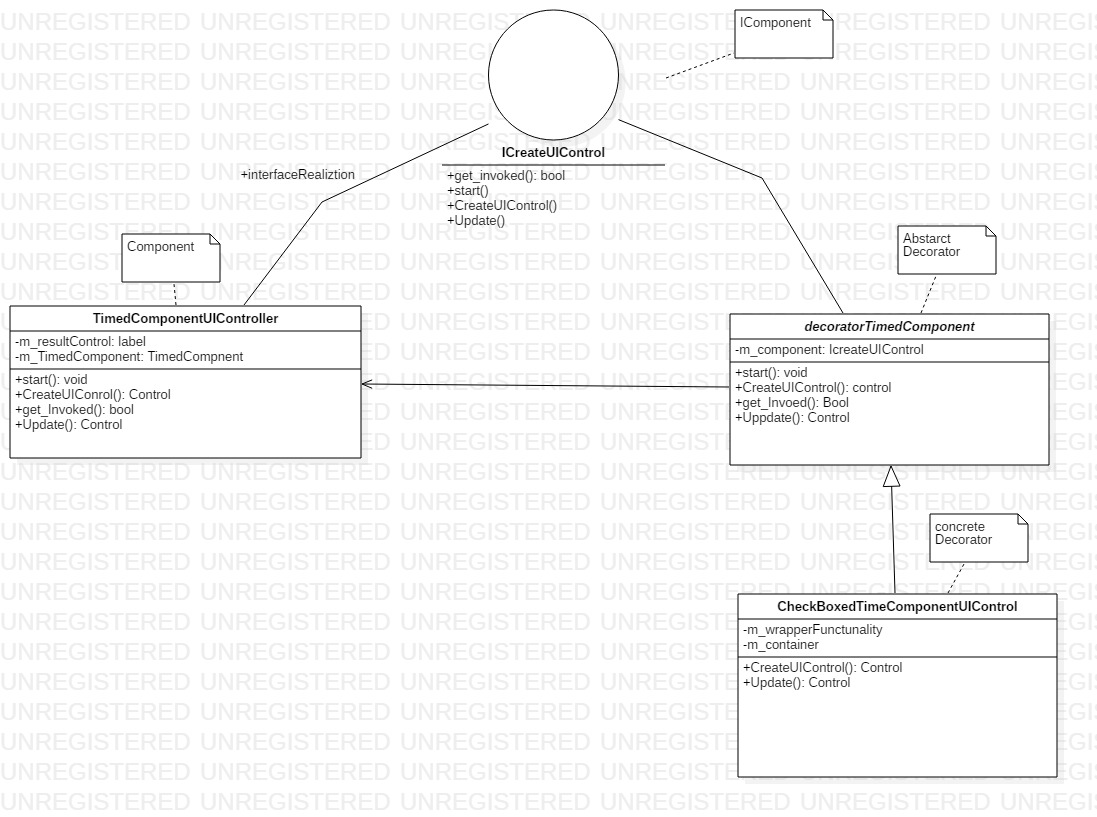
כל מחלקה אשר תירש מהאובייקט האבסטרקטי תממש את הדקורציה על ידי הוספת פעולות ביניים, פעולות עוטפות וכו'.

כמו כן, ניתן לשרשר אובייקטים בעת היצירה שלהם (כמו בצל). ביצענו זאת על ידי החזקת member מסוג IControl במחלקה DecoratorTimedComponentUIControl. בעת אתחול אובייקט מהסוג הזה, בקונסטרור מתבצעת השמת האובייקט הנוכחי ב-base class ובכך ניתן לעטוף את הרכיבים ולבצע בהם שימוש חוזר.

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Iterator

* סיבת הבחירה / שימוש בתבנית:

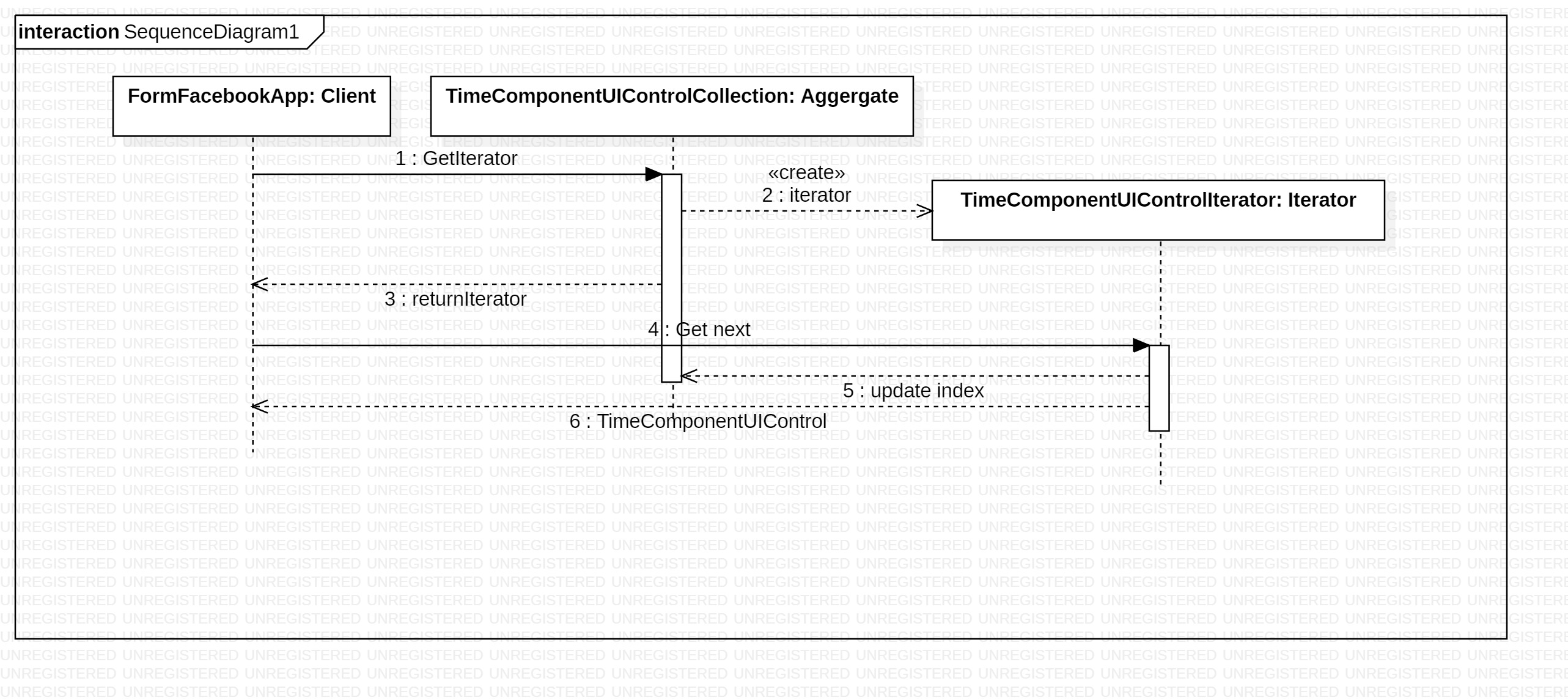
בפיצ'ר Automate Facebook Tasks, הוספנו למשתמש את האפשרות לבצע את כל הפעולות הרשומות בלחיצת כפתור (במקום להמתין לזמן הביצוע שנקבע מראש). האפשרות מוצגת למשתמש באמצעות הכפתור InvokeAll אשר נמצא בפיצ'ר. בעת לחיצה על הכפתור מתבצעות כל פעולות ה-"TimedComponent" אשר נוספו לרשימה על ידי המשתמש והן מיועדות להתבצע בעתיד.

לשם ביצוע כל הפעולות ברצף, רצינו לשמור את יצוג הפעולות במבנה נתונים כלשהו ולהפריד את מבנה הנתונים מהמשתמש, אך לתת למשתמש את האפשרות לעבור על כל מבנה הנתונים. בכדי לקבל את ההפרדה הנ"ל ולתת למשתמש את היכולת לרוץ על מבנה הנתונים, מימשנו את התבנית Iterator.

* אופן המימוש:

ב"FormFacebookApp" ישנה מתודה " buttonInvokeAll\_Click" אשר עוברת בלולאת Foreach על הקולקציה "TimedComponentUIControlCollection" והקולקציה הנ"ל מממשת את הממשק IEnumerable. במתודה GetEnumrator אנו יוצרים אובייקט TimedComponentUIControlIterator אשר מממש את הממשק IEnumartor והוא האיטרטור המוחזר אל ה- Client.

Sequence Diagram



* Class Diagram

