### 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:

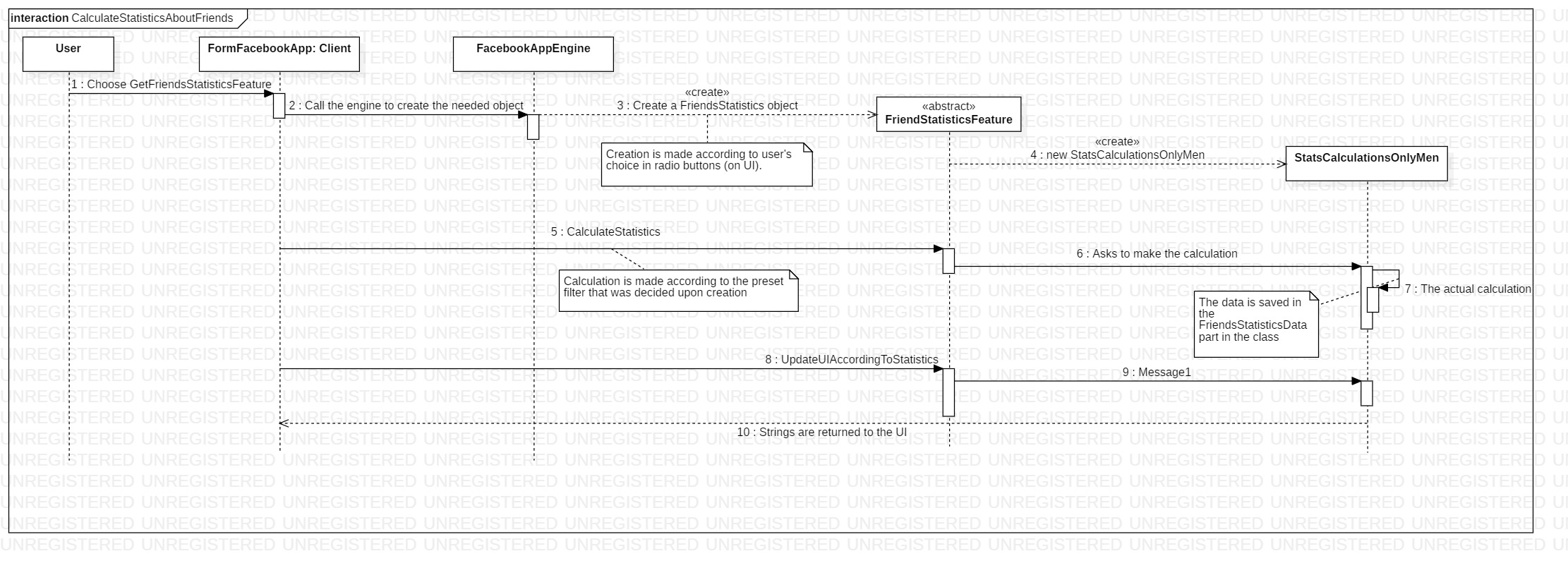
בפיצ'ר Calculate Statistics About Friends , המשתמש יכול לבחור לקבל נתונים המתבססים על רשימת החברים שלו. רצינו שתינתן למשתמש האפשרות לבחור בזמן ריצה מבין אפשרויות לסינון המשתמשים, על מנת להציג לו נתונים המתבססים על רשימת חברים מותאמת ע"פ דרישתו. למשל – קבלת כל הנתונים הסטטיסטיים רק על החברים ממין זכר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

* אופן המימוש:

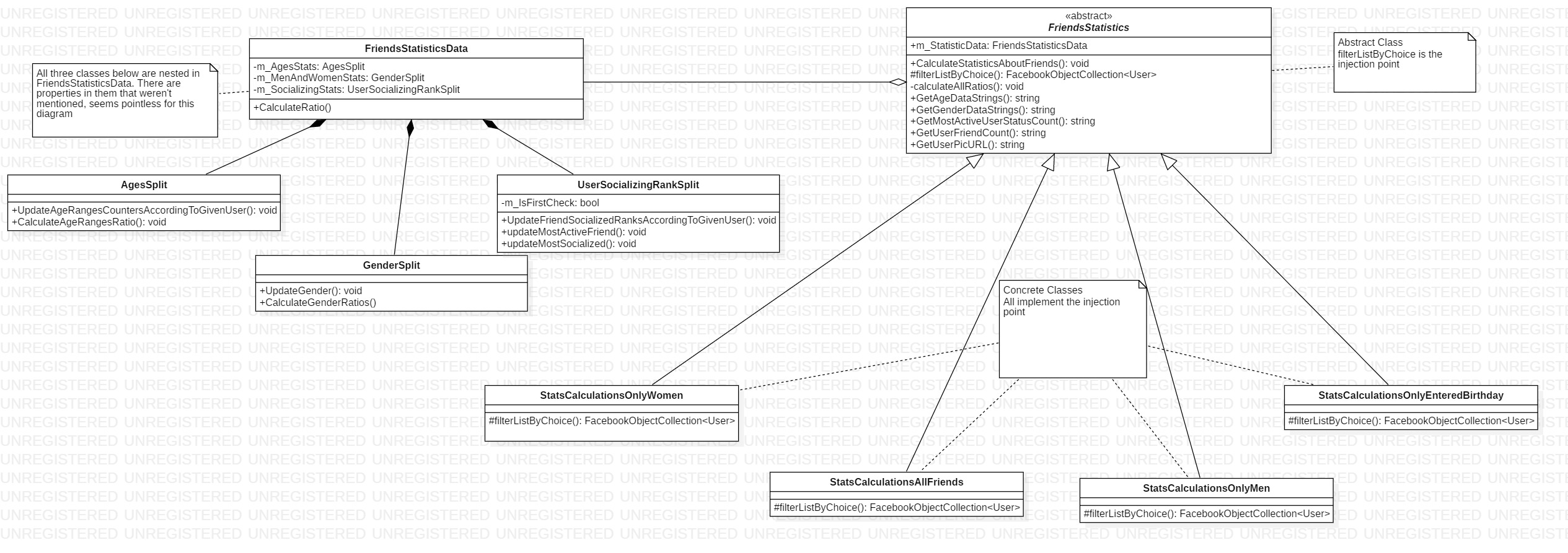
הגדרנו מחלקה אבסטרקטית FriendStatistics אשר מכילה בתוכה רפרנס ל-FriendStatisticsData (האובייקט אשר מכיל את הנתונים הסטטיסטיים עצמם) וכן הגדרנו במחלקה מתודה CalculateStatisticsAboutFrineds אשר אחראית על ביצוע החישובים והזנתם אל האובייקטים המתאימים (מוכלים בתוך FriendStatisticsData). בתוך המתודה הנ"ל , 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

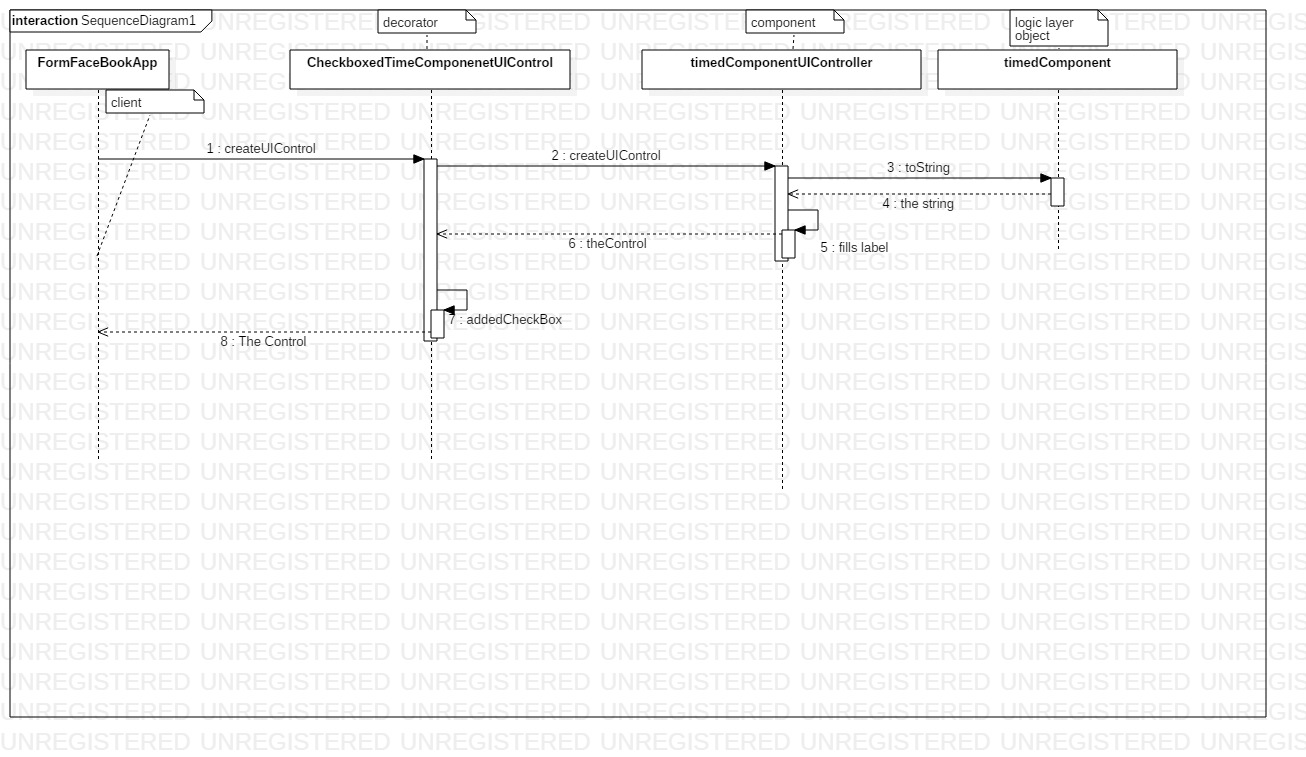
* סיבת הבחירה / שימוש בתבנית:

בפיצ'ר Automate Facebook Tasks אנו יוצרים אובייקט אשר משלב בין טיימר לבחירת המשתמש לבין פעולה "פייסבוקית",אובייקט זה נקרא TimedComponent. ע"מ להציג למשתמש את תוכן ה TimedComponent יצרנו אובייקט חדש אשר נקרא TimedComponentUIControl אשר תפקידו הוא ליהיות פקד ממשק משתמש המציג את הרכיב. בגלל שאנו רוצים לאפשר הוספה של הרחבת האחריות של הפקד בעתיד (לדוג' להוסיף אנימציה כאשר בוצעה הפעולה) נבחר להשתמש בפטרן Decorator .

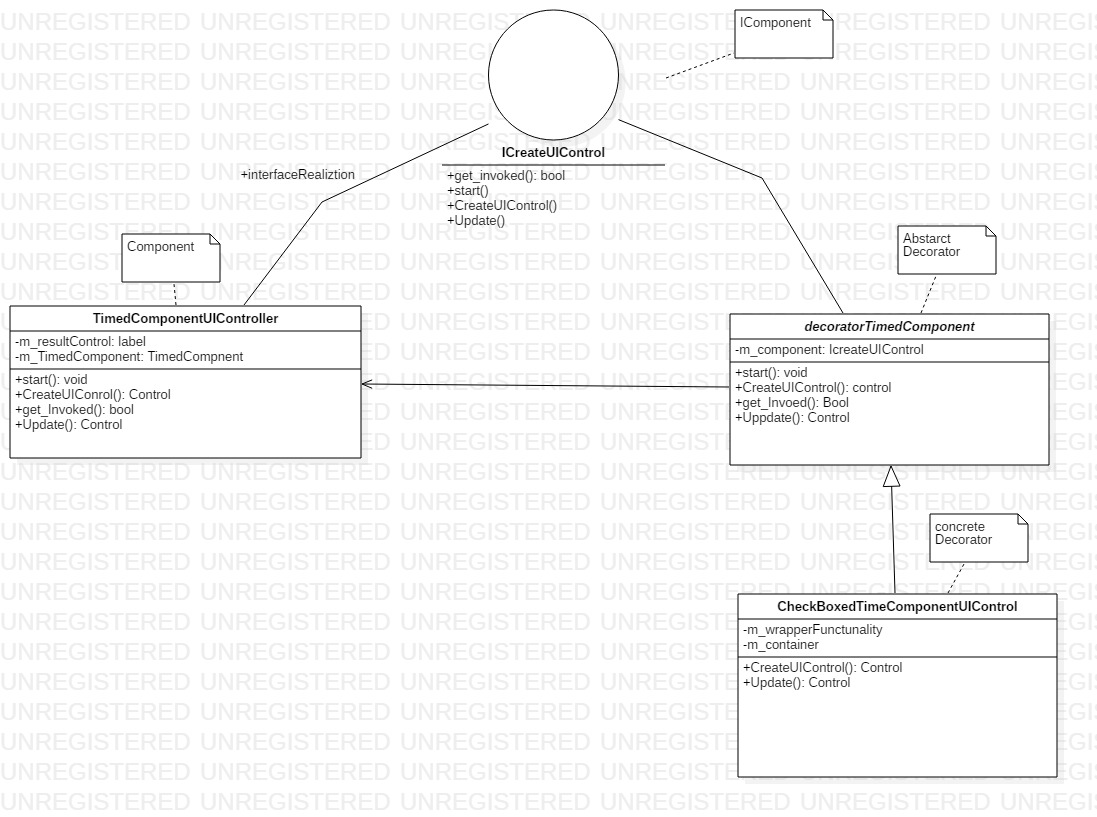
* אופן המימוש:

יצרנו "אינטרפייס" בשם ICrraeteUIControl אשר גם הרכיב הקונקרטי(TimedComponentUIControl ) וגם האובייקט האבסטרקטי(DecortorTimedComponentUIControl ) ממשים. המחלקות היורשות מהאובייקט האבסטרקטי הם אלו הממשות את הדקורציה ע"י הוספת פעולות ביניים ופעולות עוטפות.(השירשור ביצירת האובייקט מתבצע ע"י החזקת ממבר מסוג האינטרפייס אשר בעת האתחול בקונסטרקטור יאותחל הממבר ע"י האובייקט הקונקרטי\דקורציה משום ששניהם ממשים את אותו אינטרפייס ובכך נרחיב את יכולת שימוש חוזר ברכיבים).

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Iterator

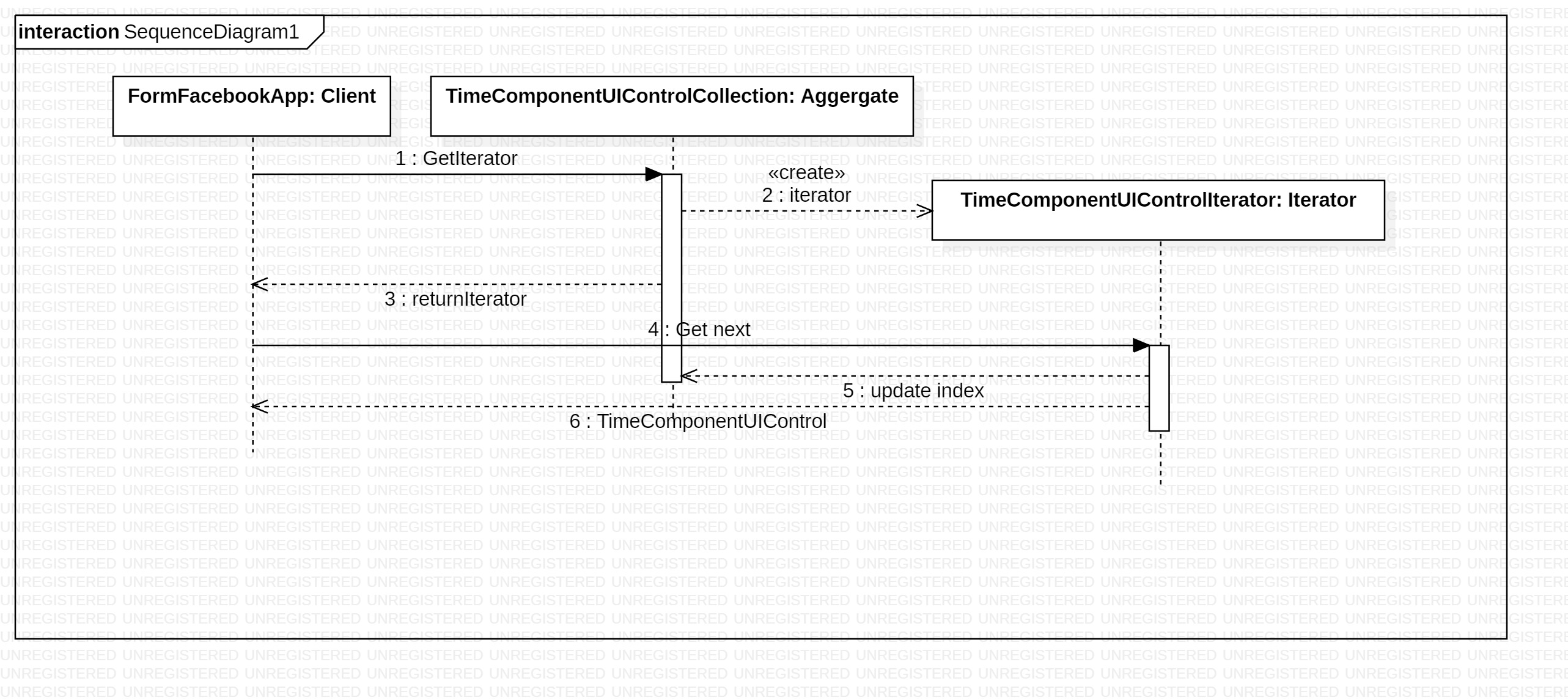
* סיבת הבחירה / שימוש בתבנית:

בהגשה זאת הוספנו פיצ'ר אשר בעקבות לחיצת כפתור מיוחד אשר נמצא בטאב "AutomateFacebookTasks" אשר כתוב עליו "Invoke All",בעת לחיצה מתבצעות כל הפעולות "TimedComponent" אשר אמורות להתבצע בעתיד, לשם ביצוע כל הפעולות אנו צריכים לשמור אותם במבנה נתונים כלשהו משום שאנו רוצים להפריד את מבנה הנתונים מן המשתמש מימשנו איטרטור.

אופן המימוש:

ב"FormFacebookApp" ישנה מתודה " buttonInvokeAll\_Click" אשר עוברת ב לולאת Foreach על האובייקט "TimedComponentUIControlCollection" משום שהוא מממש את הממשק IEnumerable ובמתודה GetEnumrator אנו יוצרים אובייקט "TimedComponentUIControlIterator" אשר מממש את הממשק IEnumartor והוא משמש כאיטרטור של מבנה נתונים כנ"ל.

Sequence Diagram



* Class Diagram

