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

**פיצ'ר 1- Finder Dating- Dating Feature**

על מנת להפעיל פיצ'ר זה יש ללחוץ על הטאב Finder Dating בחלון הראשי של האפליקציה.

פיצ'ר זה למעשה הינו אפליקציית היכרויות המבוססת על החברים לאפליקציה בפייסבוק, המתשמש יכול לפלטר את בן/בת הזוג ע"פ מגדרם וע"פ עיר מגוריהם. בלחיצה על כפתור Find My Matches! האפליקציה תפלטר ותציג למשתמש רק את החברים הרלוונטים שעברו את הפילטור ע"פ בקשת המשתמש.

תחילה, האפליקציה תציג למשתמש את שמות החברים שנמצאו כהתאמה ברשימה, לאחר מכן המשתמש יוכל ללחוץ על חבר מהרשימה ולהציג את אודותיו המתארים את שמו, תמונת הפרופיל שלו, לינק ישיר לחשבון הפייסבוק שלו, אימייל, תאריך לידה ומידע שכתב בחשבון הפייסבוק שלו בחלונית About.

* המנגון שמפלטר את החברים זורק אקספשן, כי תכונת הIntersting in לא נגישה יותר מפייסבוק.

המחלקות המעורבות בפיצ'ר זה:

* User- המכיל את המידע עבור המשתמש ובין היתר את המידע הרלוונטי על חבריו לאפליקציה.

**פיצ'ר 2- Guess My Name Feature-**

על מנת להפעיל פיצ'ר זה יש ללחוץ על הטאב Guess My Name! בחלון הראשי של האפליקציה.

אנו למעשה מייצרים חווית משחק אינטרקטיבית עם המשתמש עם מידע שייבאנו מחשבון הפייסבוק של המשתמש המחובר.

כאשר המשתמש ילחץ על הכפתור Roll a friend! יאותחל סיבוב ותוצג למסך תמונה רנדומלית של חבר מחבריו לפייסבוק, על המשתמש לזהות את שמו המלא של החבר.

למשחק הוספנו מנגנון ניקוד וחיים שיצור יותר מתח אצל המשתמש.

בתחילה, המשתמש יתחיל עם ניקוד 0 וחיים מלאים (18 פסי חיים בHealth Bar).

המשתמש יכול לנסות לנחש את שם החבר, לבקש רמז או לוותר על הסיבוב.

חוקי המשחק הן:

* תשובה נכונה תזכה את המשתמש ב10 נקודות לניקוד.
* שלוש תשובות נכונות ברציפות יזכו את המשתמש בפלוס 3 למד החיים.
* תשובה שאינה נכונה תוריד למשתמש 6 ממד החיים.
* אם המשתמש בחר לוותר על הסיבוב הוא יפסיד 3 ממד החיים.

כאשר המשתמש ילחץ על הכפתור HINT יוצג לו רמז שמיוצר מהמערכת, מנגנון הרמז מג'נרט את שם החבר ומשתיל בחלק מהאותיות בשמו באופן רנדומלי קווים תחתונים (Underline) ומציג את הג'ינרוט על המסך.

למשל אם לחבר קוראים Yossi Cohen המנגנון יכול לייצר את הרמז: Y\_s\_i C\_\_en.

כמו כן, בעת לחיצה על הכפתור Give Up יוצג למשתמש שמו של החבר שלא זכר את שמו.

בעת לחיצה על הכפתור Check your guess המערכת תבדוק אם המשתמש צלח בניחוש או לא ותפיק פלט בהתאם ע"פ חוקי המסך.

הפיצ'ר הינו not case sensitive.

המחלקות המעורבות בפיצ'ר זה:

* User- המכיל את המידע הרלוונטי אודות המשתמש וחבריו לפייסבוק.
* FacebookObjectCollection- קולקציה שמכילה את חבריו של המשתמש
* RandomNumbersGenerator- מחלקה שתפקידה העיקרי לייצר קבוצת מספרים רנדומלים ע"פ כמות מספרים וטווח המספרים. השתמשנו במחלקה זו בפיצ'ר על מנת לקבל קבוצה של מספרים שיהוו כאינדקסים לאותיות בשם החבר אותם נרצה לחשוף כאשר המשתמש יבקש רמז, בכך בכל לחיצה על רמז ג'ונרט רמז שונה למשתמש.
* StringBuilder- נעזרנו בעת הרכבת השם המרומז לבנייה ויצור הרמז.

### תבנית מס' 1 – Strategy

* סיבת הבחירה / שימוש בתבנית:

בחרנו לממש את תבנית Strategy משום שרצינו לאפשר לקליינט למיין את ה Matches בפיצ'ר הדייטינג בכל מיני דרכים, כך שיוכל להוסיף סוגי מיון שונים בעתיד בצורה מיינטנביליט.

* אופן המימוש:

FacebookDatingFeatureמחזיק כעת אובייקט מסוג IComparer, אשר מהווה את הInjection Point למחלקה, כך שכל פעם שנרצה לממש מתודולוגיית מיון שונה, ניתן לאובייקט זה את האובייקט מיון החדש.

מימשנו שתי מתודולוגיות מיון ,AscendingUserNameComparer DescendingUserNameComparer אשר מופעלות לפי בחירת המשתמש.

בעלי התפקידים בתבנית:

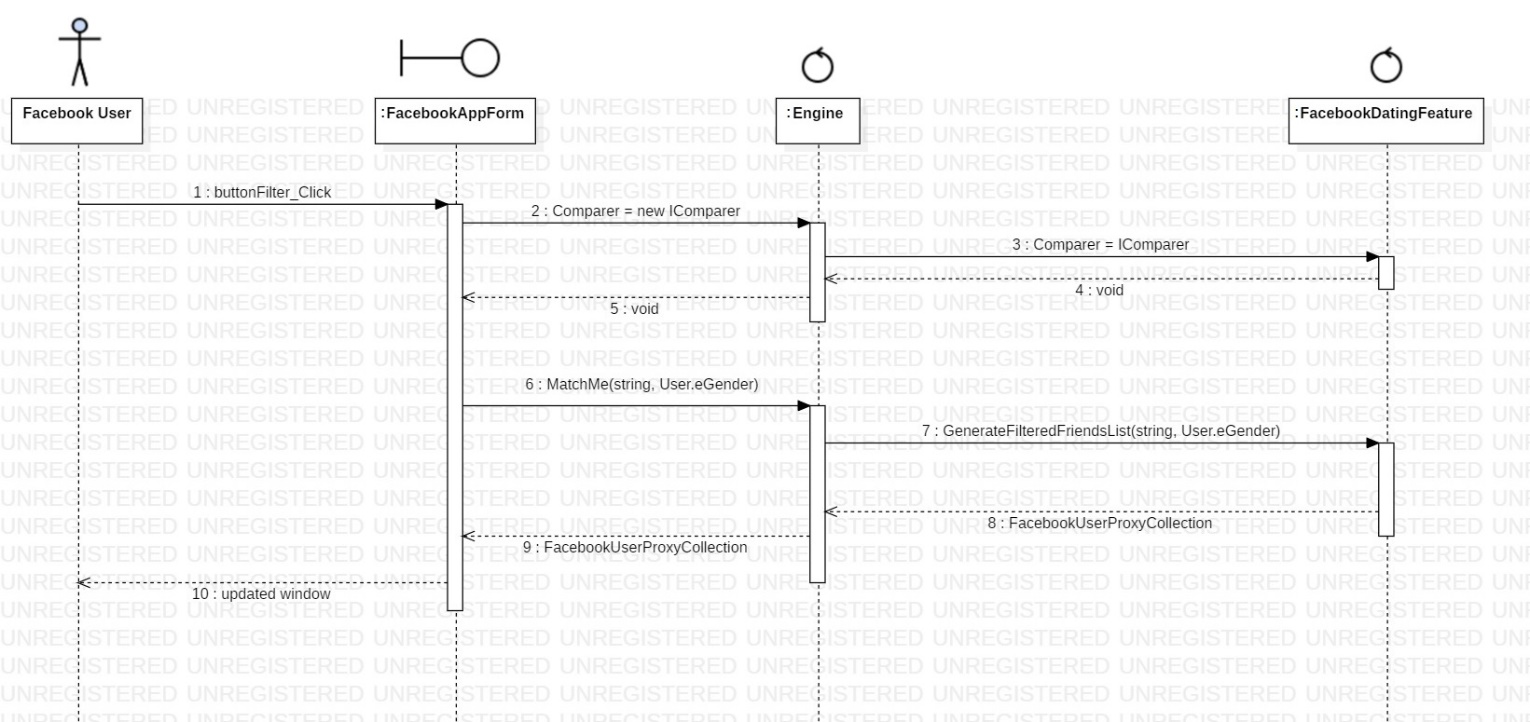
Context - FacebookDatingFeature

Strategy – IComparer

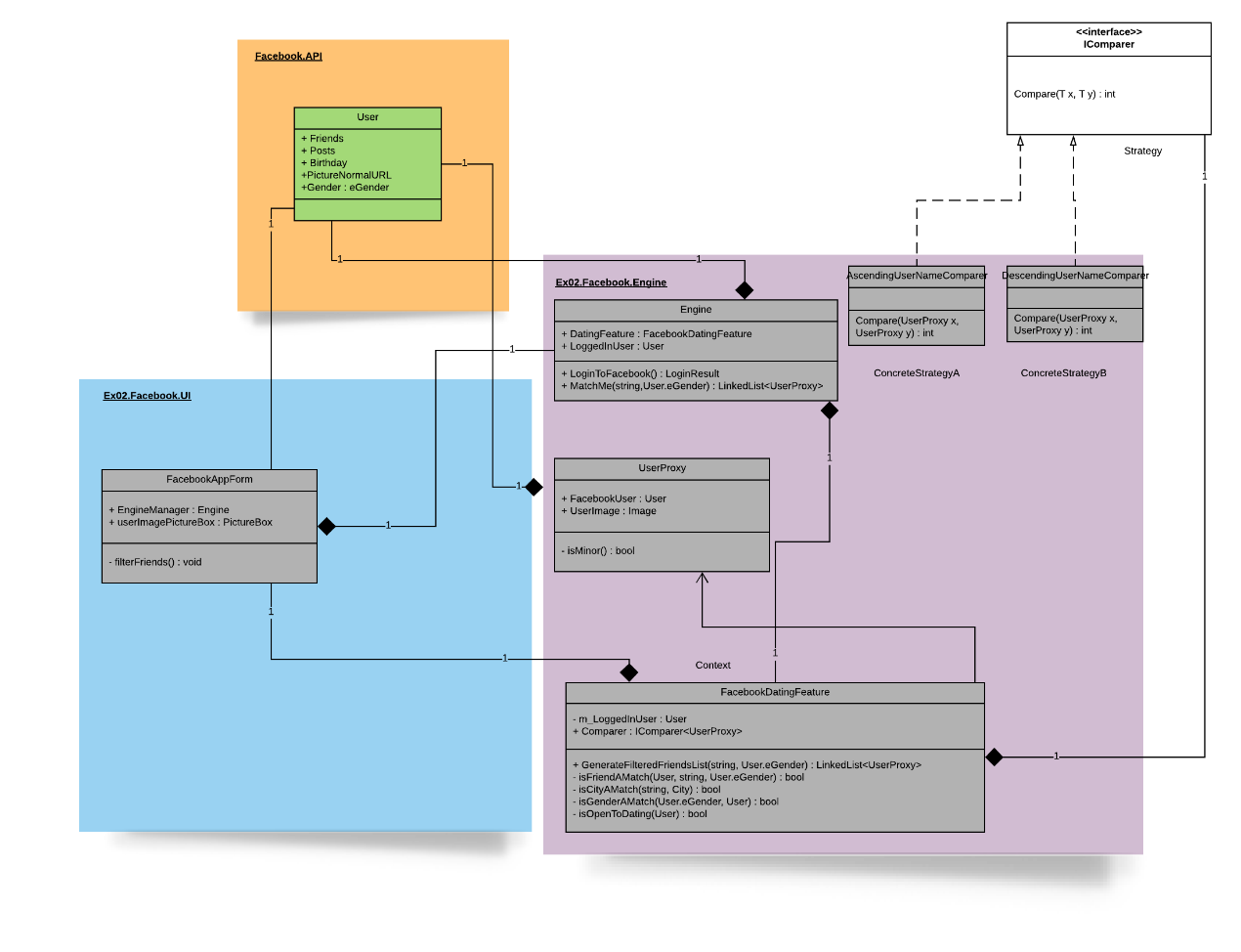
ConcreteStrategyA - AscendingUserNameComparer

ConcreteStrategyB - DescendingUserNameComparer

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Observer

* סיבת הבחירה / שימוש בתבנית:  
  בחרנו לממש את תבנית Observer משום שרצינו ליצור decoupling בין ה model ל view. התבנית מאפשרת למודל ליידע את רכיבי ה UI על שינויים בתוצאת המשתמש(Score, Health) מבלי הצורך להכיר אותם באופן ממשי.

בנוסף, אם בעתיד ברכיב כלשהו במערכת יהיה הצורך להיות מעודכן לגבי שינויים ב Score או ב Health, נוכל לעשות זאת בצורה קלה מבלי לשנות את הקוד ב model על ידי רישום כ Observer.

* אופן המימוש:

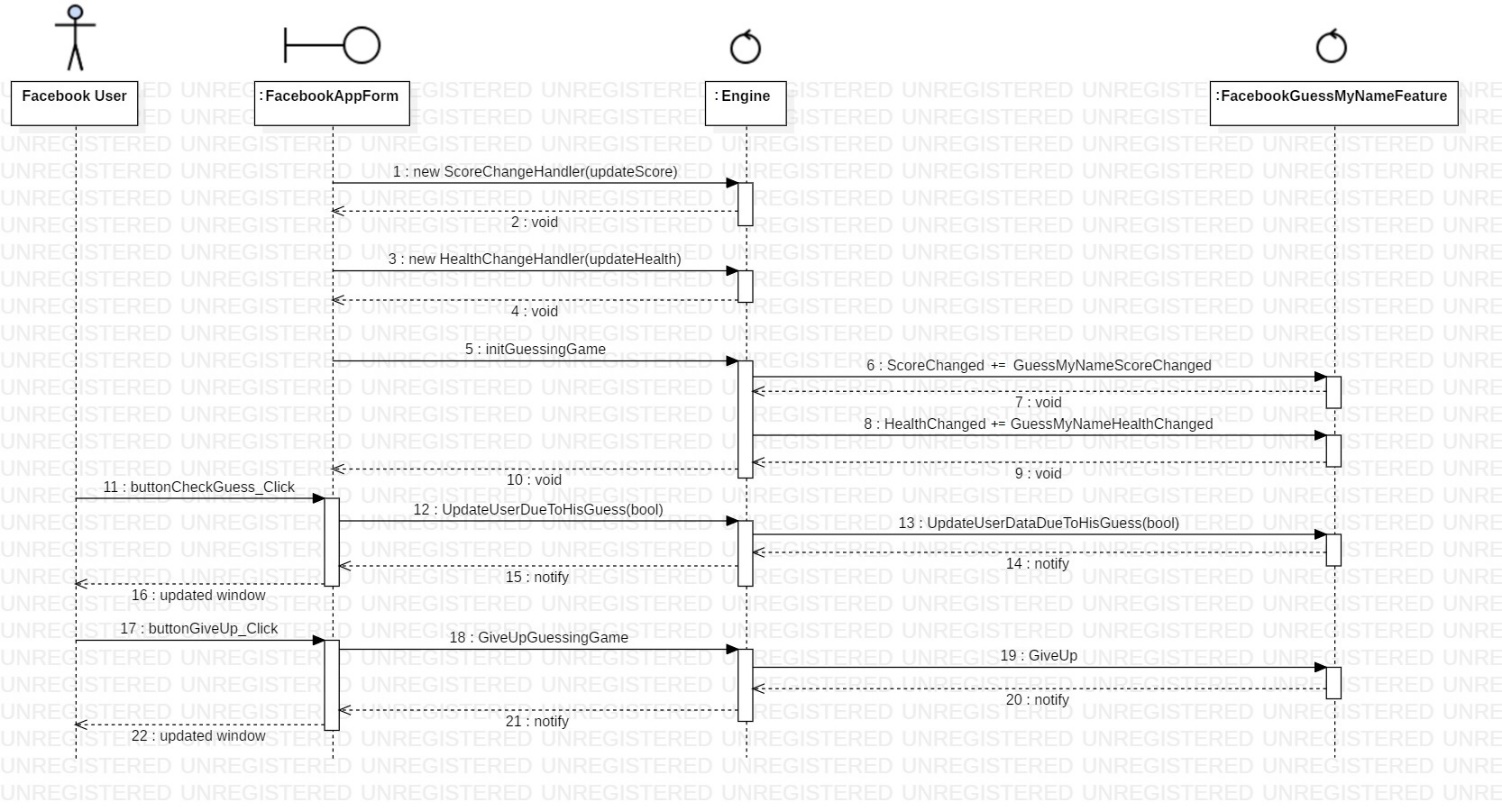
בפיצ'ר GuessNyName כאשר משתנה ה Score או ה Health, הוא מיידע את כל הObservers שלו על כך מבלי להכיר אותם באופן מפורש.

ה FacebookAppForm נרשם כ Observer לשינויים עבור Score ו Health , ובכך כל שינוי שיתבצע בהם במהלך המשחק, יעדכן גם את FacebookAppForm שבהתאם יעדכן את רכיבי ה UI של Score ו Health.

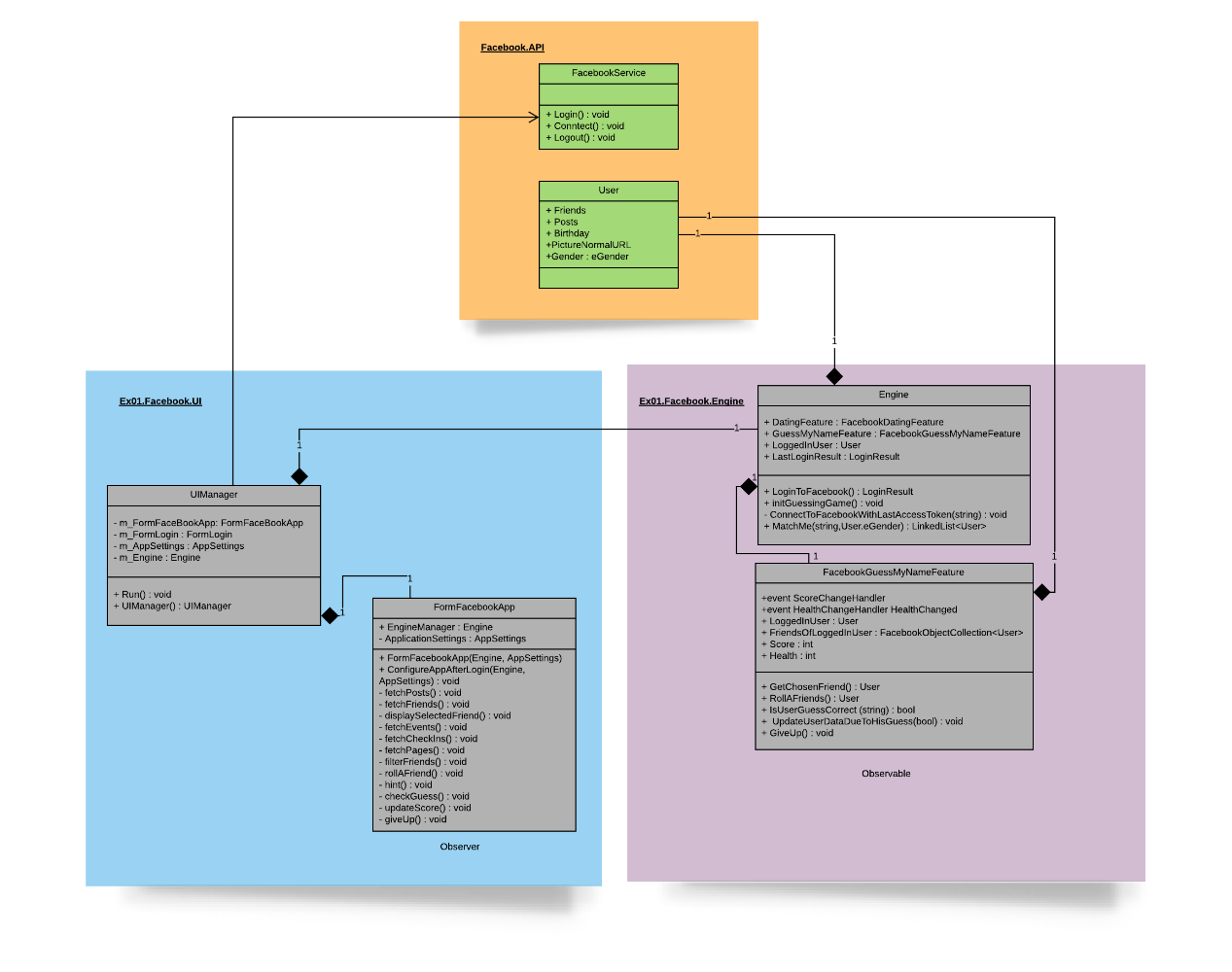
Observer: FacebookAppForm

Observable : GuessMyNameFeature

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Iterator

* סיבת הבחירה / שימוש בתבנית:

במימוש הפיצ'ר FacebookDatingFeature אנו מחזיקים אוסף של אובייקטים מסוג FacebookUserProxyCollection (במתודה GenerateFilteredFriendsList) אשר מחלקות הUI מושכות ממנו מידע על מנת להציגו למשתמש.

כדי לפשט את המעבר על האוסף השתמשנו בתבנית iterator ובכך יצרנו ממשק פשוט המאפשר מעבר על האוסף גם ללא הצורך לדעת באיזה מבנה נתונים מדובר.

כמו כן, רצינו להקל על תחזוקה עתידית של הקוד כך שלא נהיה תלויים במבנה נתונים ספציפי, ואם בעתיד נרצה להחליף את מבנה הנתונים, נוכל לעשות זאת ללא שינוי הקוד בקליינט, וזה בדיוק מה שמציעה תבנית Iterator.

* אופן המימוש:

כתבנו מחלקה נוספת FacebookUserProxyCollection המחזיקה מופע מסוג ICollection של UserProxy.

מחלקה זו מימשה את ה Interface IEnumerable ולפיכך את המתודה GetEnumerator, אשר עוברת על כל ה UserProxy בCollection ומבצעת yield return עבור כל UserProxy שכזה.

השימוש ב- Iterator מופיע במחלקה FacebookAppForm במתודה filterFriends בעת הגדרת הקולקציה כ DataSource, אשר תוביל לשימוש באיטרטור על מנת להציג את קולקציית החברים ב UI(לדוגמא ב ListBox).

בעלי התפקידים בתבנית:

Client: FacebookAppForm

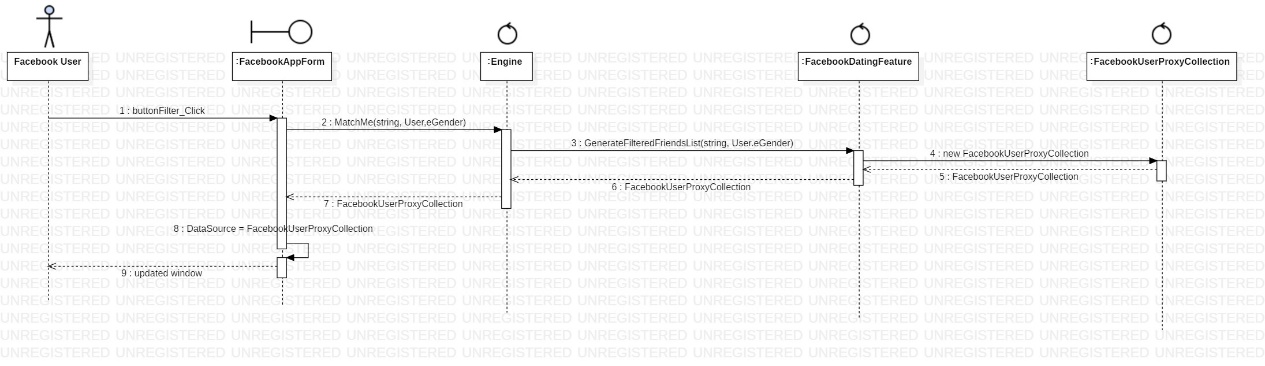
Aggregate: IEnumerable

ConcreteAggregate : FacebookUserProxyCollection

Iterator : IEnumerator

ConcreteIterator : IEnumerator<UserProxy>

* Sequence Diagram



* Class Diagram

