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

* מציאת צ'קאינס בתוך אזור מסוים. האפליקציה מאפשרת למשתמש להגדיר איזור חיפוש גיאורפי מסוים (על ידי הזנת נקודה מרכזית ורדיוס) ולחפש צ'ק-אינס של חבריו בפייסבוק בתוך אזור זה.
* הצע חברים חדשים על פי תמונות או אירועים. זוהי הרחבה של פיצ'ר 'הצע חברים חדשים' של פייסבוק. האפליקציה מחפשת בתמונות של המשתמש ובאירועים שלו אחרי אנשים המתוייגים שם ועדיין אינם חברים של המשתמש ומציעה אותם כחברים שלו.

### תבנית מס' 1 – Template method

* סיבת הבחירה / שימוש בתבנית:

האובייקטים הממליצים על חברים חדשים מכילים את המתודה GetSuggestionsAsync. המתודה היא טמפטית כיוון שהיא מקבלת כפרמטר פונקציה ממיינת. הפונקציה הממיינת היא טמפלטית ומתאימה לכל טיפוס ממוין (צילומים, אירועים וכו') הלקוח של אובייקטים אלו יכול להעביר פונקציות ממינות כרצונו וזאת מבלי לשנות את הקוד של האובייקט הממליץ, כך מתאפשרת גמישות ומשופרת המיינטנביליות.

* אופן המימוש:

ראה בקבצים EventsFriendRecommender.cs, PhotosFriendRecommender.cs במתודה GetSuggestionsAsync.

* Sequence Diagram

ראה למטה בתיאור של התבניות האחרות

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### תבנית מס' 2 – Strategy

* סיבת הבחירה / שימוש בתבנית:

האובייקטים הממליצים על חברים חדשים מכילים את המתודה GetSuggestionsAsync. המתודה היא טמפטית כיוון שהיא מקבלת כפרמטר פונקציה ממיינת.  
שיטת המיון (the strategy) תלויה בלקוח של האובייקטים האלו והם יכולים להחליף אותה כרצונם.

* אופן המימוש:

ראה בקבצים EventsFriendRecommender.cs, PhotosFriendRecommender.cs במתודה GetSuggestionsAsync.

* Sequence Diagram

ראה למטה בתיאור של התבניות האחרות

* Class Diagram

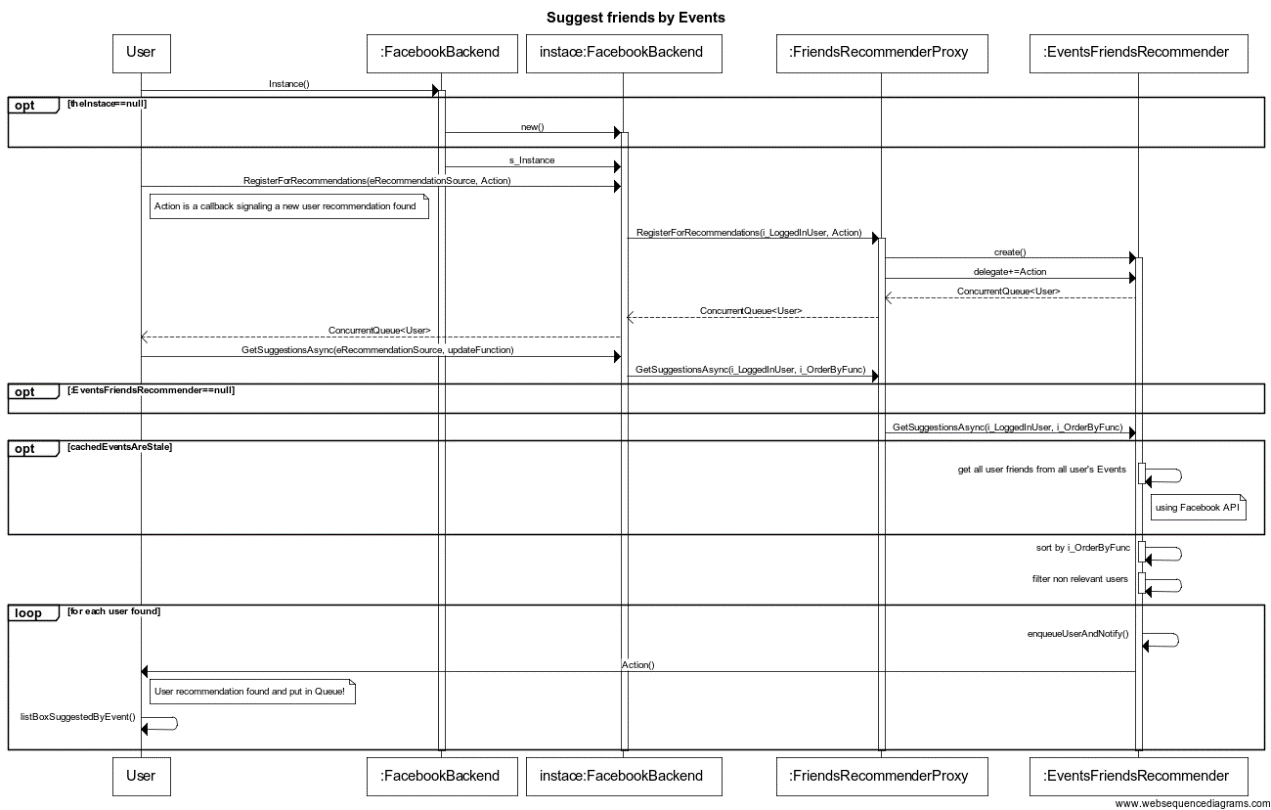
[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### תבנית מס' 3 – Observer

* סיבת הבחירה / שימוש בתבנית:

כדי לשפר ביצועים החלטנו לבצע את הבאת ההמלצות על חברים חדשים מפייסבוק בתהליכון (thread) שונה מזה של ממשק המשתמש. בצורה זו, המשתמש יכול להמשיך ולהפעיל את ממשק המשתמש גם כשהוא מחכה שההמלצות יחושבו. כדי לממש זאת תהליך חישוב ההמלצות מתבצע בצורה אסינכרונית.  
ממשק המשתמש 'נרשם' לאירוע בו נמצאה המלצה חדשה ומחזיק בתור של המלצות. כאשר נמצאת המלצה חדשה היא מוכנסת לתור והאירוע גורם לממשק המשתמש לשלוף אותה מהתור ולהציג אותה.

* אופן המימוש:
* ראה בקבצים EventsFriendRecommender.cs, PhotosFriendRecommender.cs במתודה GetSuggestionsAsync ובקובץ FriendsRecommenderProxy.cs במתודה RegisterForRecommendations
* Sequence Diagram



* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### שימוש בתהליכונים (Threads)

פירטנו את הסיבה לשימוש בתהליכונים בהסבר על השימוש בתבנית Observer.

### שימוש בקישור לנתונים (Data Binding)

[תיאור הסיבה / הצורך בשימוש במערכת שלכם, תיאור השימוש, מציין מקום בקוד]