### 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 – Singleton

* סיבת הבחירה / שימוש בתבנית:

השתמשנו בתבנית על המחלקה FacebookBackend. רצינו לאכוף יצירת אינסטנס בודד שלה מכל מקום באפליקציה משתי סיבות. האחת, דרכב עושים לוגאין לפייסבוק ורצינו למנוע את האפשרות לייצר שני אובייקטים כאלו ולייצר race conditions ובעיות אחרות שיכולות לנבוע משני נסיונות לוג-אין נפרדים\ מקבילים. השנייה, היא מהווה נקודת כניסה של האפליקציה לשירותים שהיא מקבלת מפייסבוק ורצינו לאכוף קיום יחיד של המופע שלה מסיבות caching (ראה בהמשך).

* אופן המימוש:

המימוש מבוצע על ידי בנאי פרטי ו double check lock במחלקה FacebookBackend במתודה Instance.

* Sequence Diagram

ראה בהמשך דיאגרמות המפרטות גם את פעולת התבנית הזו.

* 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 – Proxy

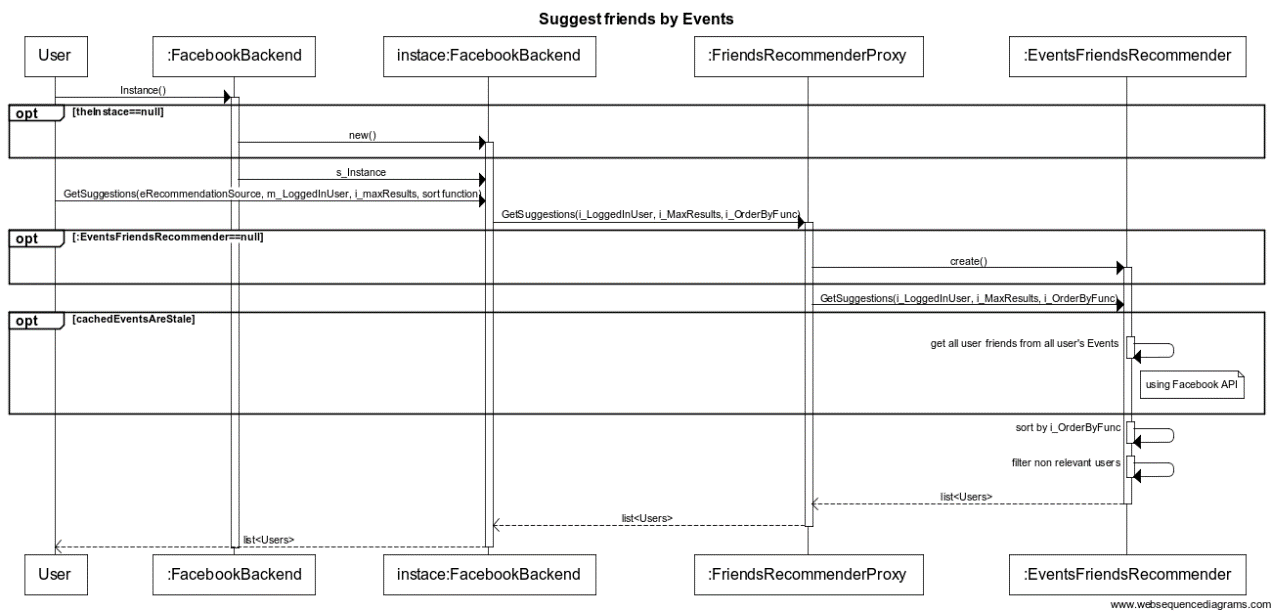
* סיבת הבחירה / שימוש בתבנית:

הבאת התמונות והאירועים של המשתמש מפייסבוק דורשת זמן (התהליך מתבצע במחלקות PhotosFriendsRecommender ו EventsFriendsRecommender) ולכן נחלטנו להכניס מנגנון caching פשוט, המבוסס על זמן, כדי לשפר את ביצועי האפליקציה. הכנסת המנגנון גרמה לכך ששני אובייקטים אלו יכילו מידע (state) ש'יקר' לייצר אותו ולכן החלטנו להשתמש Proxy המממש את אותו הממשק על מנת להפריד את יצירת האובייקט היקרה מהשימוש באובייקט ועל ידי כך להפוך אותה ל'עצלנית'- "lazy proxy".

* אופן המימוש:

המימוש מופיע במחלקה FriendsRecommenderProxy. המחלקה FacebookBeckend מחזיקה רפרנס למופע שלה והיא בעצמה יוצרת מופעים של EventsFriendsRecommender ושל PhotosFriendsecommender במקרה הצורך. מחלקות אלו מוגדרות internal בפרויקט.

* 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

### תבנית מס' 3 – Facade

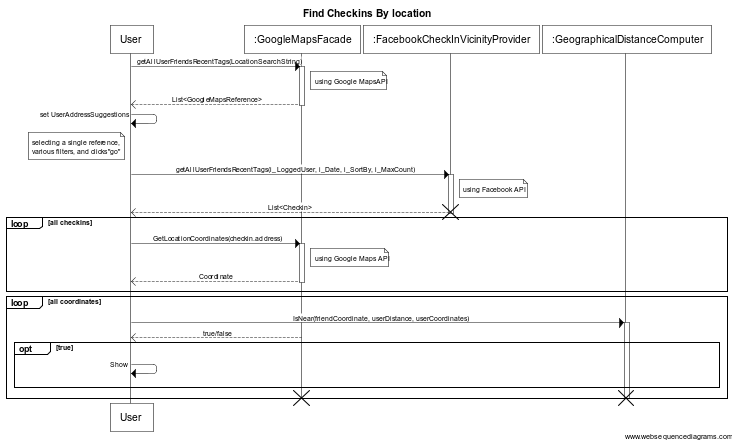
* סיבת הבחירה / שימוש בתבנית:

המחלקה GoogleMapsFacade היא למעשה Facade המיועד לפשט את העבודה מול השירותים שאפשר לקבל משני API נפרדים המהווים חלק מGoogle maps: Places details, Places autocomplete. היא מסתירה מהמשתמש את העבודה הכרוכה ביצירת בקשות HTTP לשירותים האלו וביצוע Parsing של התשובות המתקבלות מהם.

* אופן המימוש:

המימוש כתוב במחלקה GoogleMapsFacade

* 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