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

* אינפורמציה סטטיסטית : פיצ'טר שאוסף ומציג למשתמש מידע סטיטסי על הפעילות שלו כגון : כמה לייקים קיבל המשתמש על פוסטים, מה הפוסט שקיבל הכי הרבה לייקים, מה האלבום שקיבל הכי הרבה ליקיים, מה התמונה שקיבלה הכי הרבה לייקים
* מוד מסיבה : פיטצ'ר שנותן למשתמש אפשרות לעשות מסיבה בעמוד הפרופיל שלו. הפיטצ'ר משמיע למשתמש מוזקית מסיבות עדכנית וגורם לעמוד הפרופיל להתחיל לרקוד (משנה צבעים מזיז פקדים)

### תבנית מס' 1 – Factory methood

* סיבת הבחירה / שימוש בתבנית:

במערכת שלנו יש משפחה פולימורית של איברים שעוטפת אובייקטים של פייסבוק שיש להם property של תמונה. ואנו מעוניינים להוציאר את לוקיגת הייצירה מה- client

* אופן המימוש:

מומשה מחלקה GridItemFactory שמספקת מתודה סטטאית שמקבל facebookCollection<T> ובונה קולקציה של המשפחה הפולימורפית שעוטפת את האיברים של פייסבוק

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* 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:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* 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 – Builder

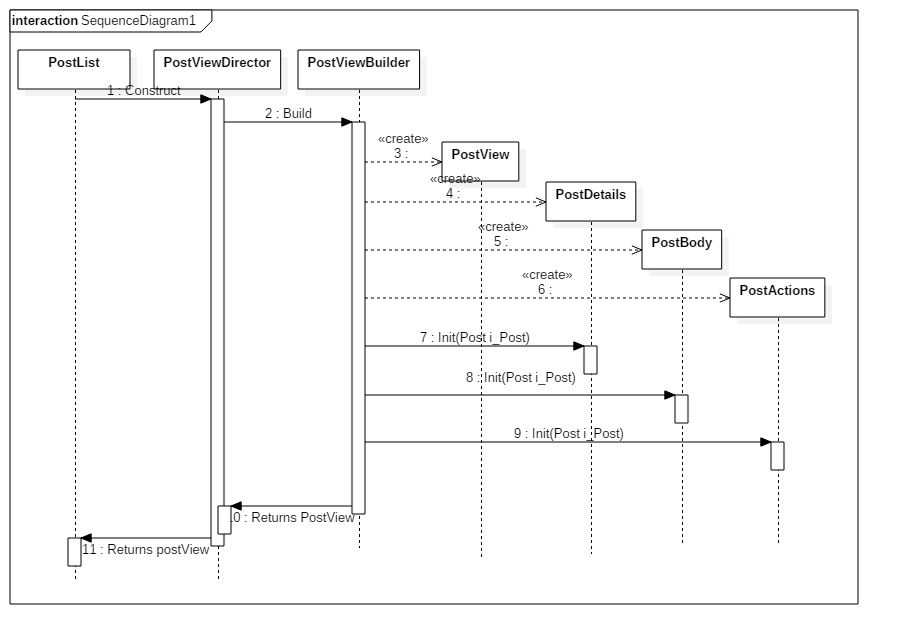
* סיבת הבחירה / שימוש בתבנית:

במערכת שלנו, הפוסט מורכב מכמה תתי רכיבים, מעצם כך שמתבצע רצף "הרכבה" של מספר רכיבים לרכיב אחד, התבנית המתבקשת תהיה Builder לארוז את לוגיקת האריזה.

* אופן המימוש:

מומשו שתי המחלקות PostsViewBuilder ו PostsViewDirector כמצויין בPattern כשאר הדיירקטור אחראי על קבלת ההזמנה, ולהעביר אותה לבילדר עצמו ששם יורכב הרכיב המבוקש ויוחזר ל"לקוח", במקרה שלנו השימוש בדיירקטור נעשה במחלקה PostsList

* Sequence Diagram



* Class Diagram

