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

* [תיאור קצר של הפיצ'ר הראשון]
* [תיאור קצר של הפיצ'ר הראשון]

### תבנית מס' 1 – [שם התבנית]

* 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' 2 – [שם התבנית]

* 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 – [שם התבנית]

* 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

### עבודה אסינכרונית:

בחרנו להשתמש בעבודה עם multithreading בעת טעינת ה posts של המשתמש. את השימוש ב multithreading ניתן למצוא במחלקה UserProfileForm תחת הפונקציה fetchUserPostAsync. מטרת הפונקציה היא לטעון את ה posts האחרונים של המשתמש ולהציג אותם. הצגת ה post כוללת יצירת userControl חדש עבור כל post שמכיל בתוכו מכלול של פקדים לצורך ביצוע פעולות על ה post. השימוש בעבודה אסינכרונית ממומש באופן הבא:  
הפעולה fetchUserPostAsync מייצרת thread חדש האחראי לקבל את ה posts מפייסבוק , לייצר עבור כל אחד את ה userControl שלו ולהוסיף אותו ל panel המתאים ב UI. על מנת להוסיף את ה userControl של כל post ל panel הראשי של ה posts היה עלינו להשתמש בפונקציה invoke של ה panel על מנת שנוכל לשנות את פקד ה UI מ thread שהוא לא ה thread שיצר אותו.  
בחרנו להשתמש בעבודה אסינכרונית במקרה זה על מנת לאפשר למשתמש להמשיך להשתמש באפליקציה מבלי לחכות עד שכל ה posts שלו ייטענו. בעזרת multithreading שיפרנו את חווית השימוש של המשתמש.