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

* הצגת פעילות פוסטים – מציג למשתמש כמה זמן עבר מאז העלה פוסט, העלה תמונה או העלה סירטון.
* [תיאור קצר של הפיצ'ר השני]

### תבנית מס' 1 – Static Class Factory

* סיבת הבחירה / שימוש בתבנית:

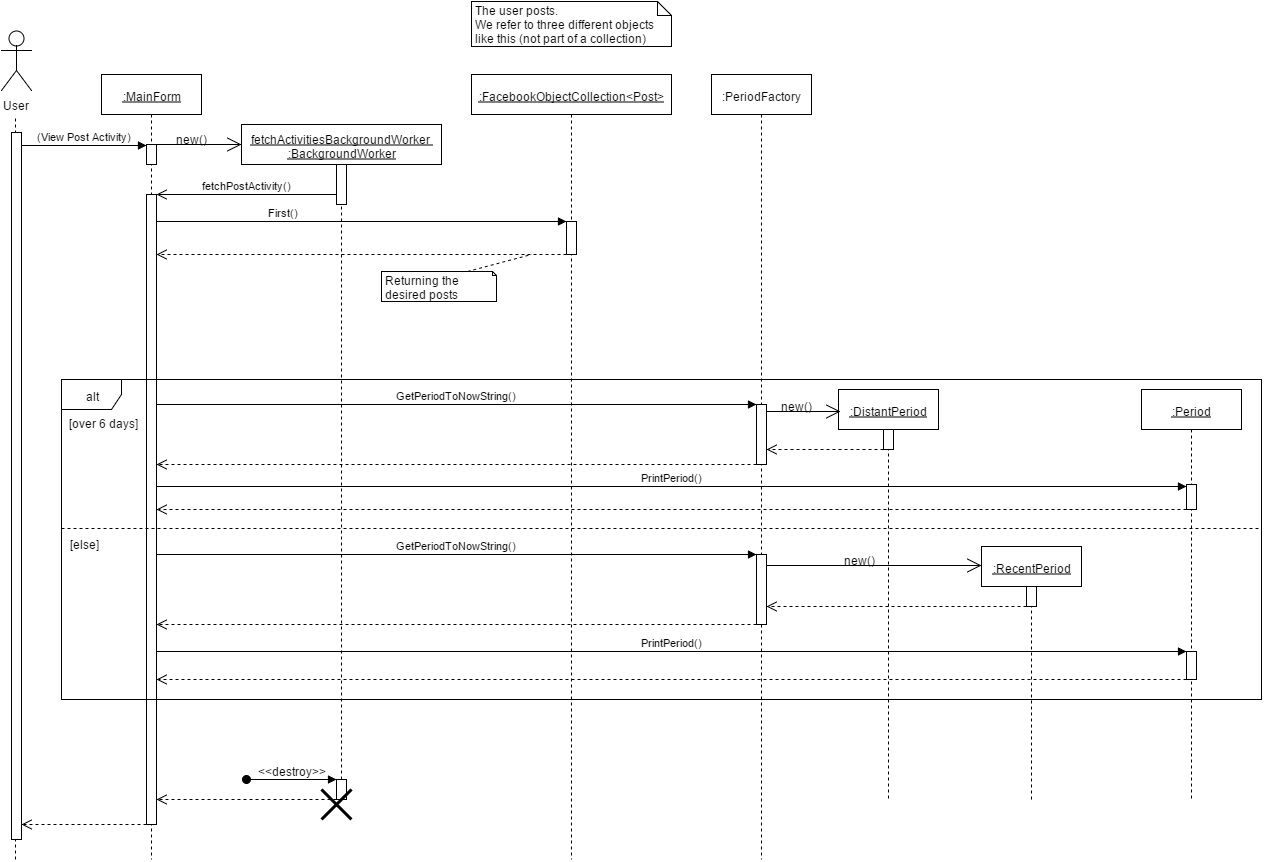
תצוגת זמן שונה עבור תקופת זמן שונות (תקופת זמן קרובה או רחוקה)

* אופן המימוש:

המימוש נמצא בקובץ periodOfTime.cs שמכיל את כל המחלקות המעורבות בתבנית.

אלו המחלקות:

* Period: מחלקה אבסטרקטית שמייצגת תקופת זמן.
* RecentPeriod: מחלקה שיורשת מ-Period ומממשת תצוגה של תקופת זמן קצרה.
* DistantPeriod: מחלקה שיורשת מ-Period ומממשת תצוגה של תקופת זמן ארוכה.
* PeriodFactory: מחלקה שמאפשרת יצירה של תצוגות תקופות שונות, בהתאם לאורכה.
* Sequence Diagram



* Class Diagram

Creator / Concrete Creator

Client

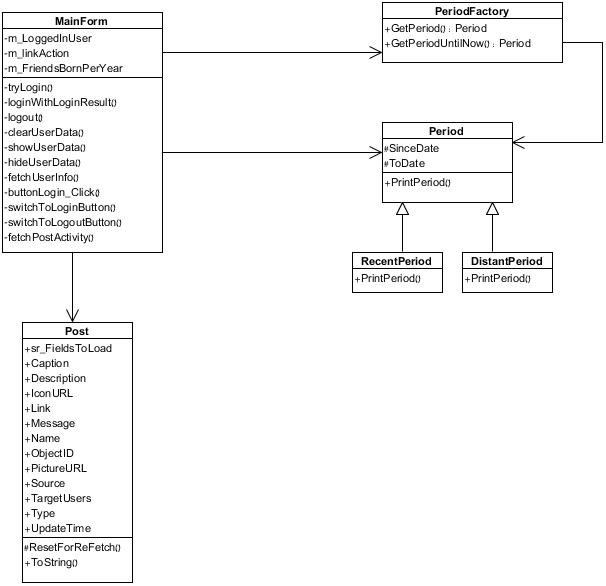
Concrete

Product

Concrete

Product

Product



### 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 – Singleton

* סיבת הבחירה / שימוש בתבנית:

חוסר משמעות למופעים שונים של ה-Director בתבנית ה-Builder בתוכנית שלנו

* אופן המימוש:

מימשנו על המחלקה DirectorBirthdayList בקובץ Builder.cs .

המימוש הינו Thread-safe ובכך מונע יצירה של יותר ממופע אחד כאשר שני thread-ים מנסים ליצור את ה-Instance הראשון של המחלקה.

בנוסף, מומש כך שקטע הקוד הקריטי אינו קורה אלא כאשר עדיין לא נוצר ה-Instance הראשון.

* Sequence Diagram

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

* Class Diagram

Singleton class

Client

