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

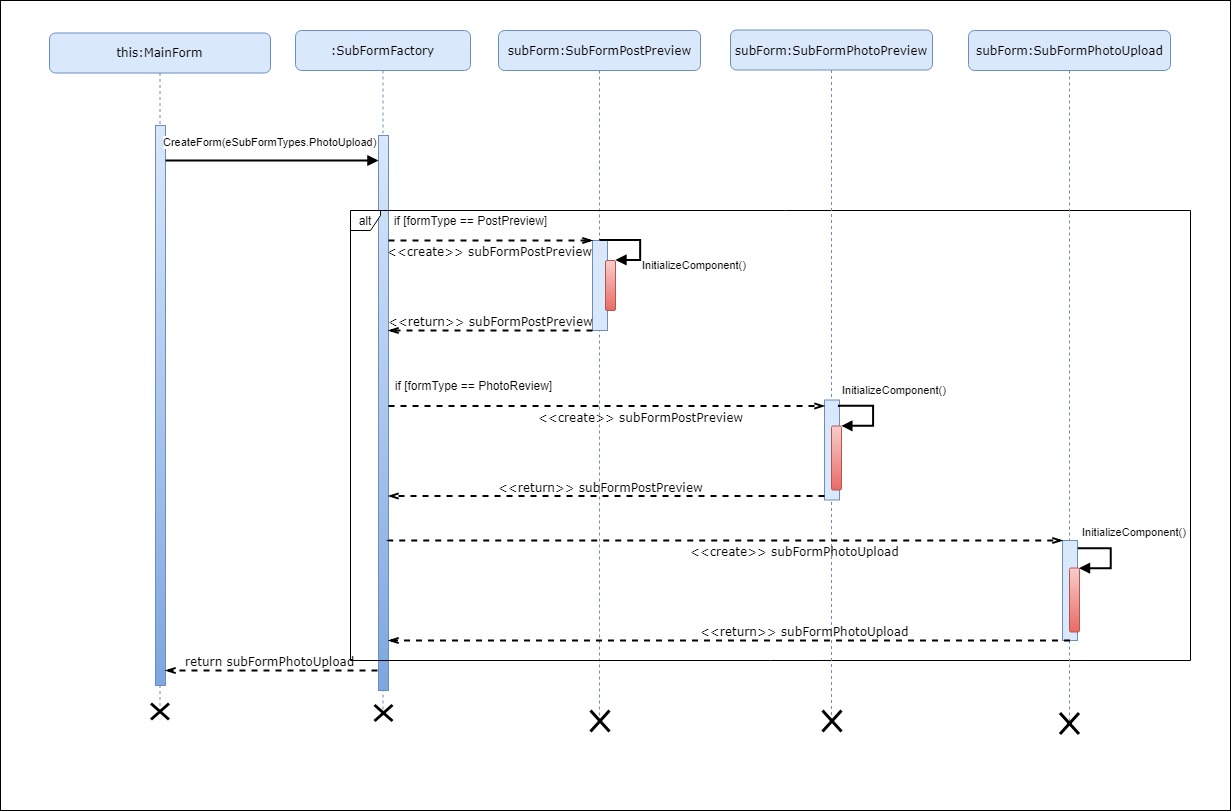
* Tagged Places – מפה שעל גביה מוצגים מיקומים של המשתמש על פי סוג המידע המחובר למיקום.
* Top Liked Photo Feature – פרמידה של 6 תמונות שזכו למירב ה"לייקים" של המשתמש, ורשימת 5 החברים המפרגנים ביותר ב"לייקים".

### תבנית מס' 1 – Factory Method

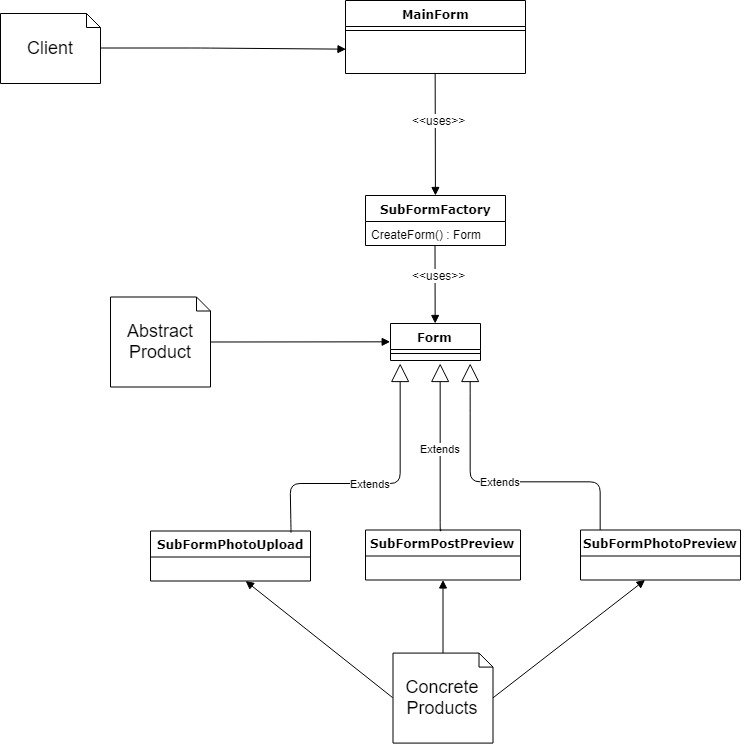
* סיבת הבחירה / שימוש בתבנית

השימוש בתבנית זו נעשה על ידי ניהול יצירת טפסים (Forms) נוספים במערכת שכן הם מהווים משפחה פולימורפית. בחרנו בתבנית זאת כדי למנוע מצב בו חלק יצירת טפסים במערכת מפוזר וחוזר על עצמו בקוד של ה-Client. מכאן שאם בעתיד נרצה להוסיף טפסים נוספים במערכת, נוכל להשתמש בתבנית זאת בהתאם ל-Context מסויים.

* אופן המימוש:
  + יצרנו את המחלקה "SubFormFacroty" אשר מכילה Enum שמציין את סוגי הטפסים שניתן לייצר להם מופע. כמו כן במחלקה קיימת המטודה הסטטית "CreateForm".
  + המטודה הסטטית "CreateForm" מקבלת בתור ה-"Context" את סוג הטופס אותו המערכת רוצה ליצור, והיא מחזירה מופע חדש של הטופס המבוקש.
  + המחלקה "SubFormFacroty" משמשת כ-Factory Method, והמחלקות "SubFormPhotoPreview", "SubFormPhotoUpload" ו-"SubFormPostPreview" הינן הטפסים שניתן ליצור להן מופע באמצעות ה-Factory Method.
* Sequence Diagram

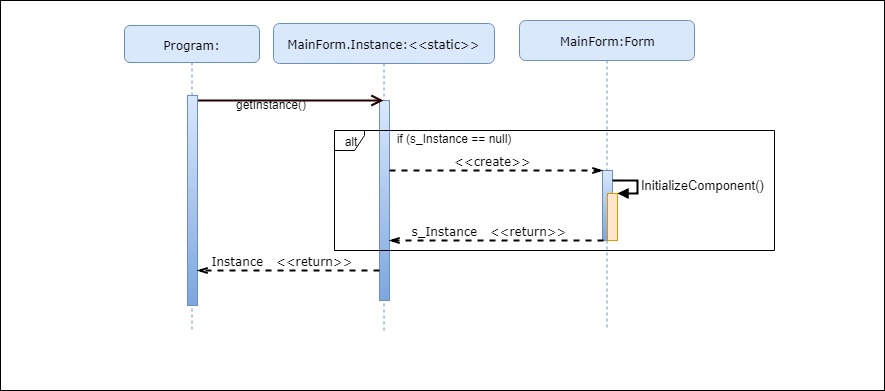


* Class Diagram

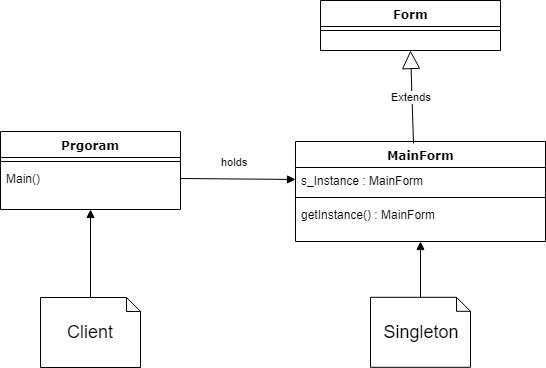


### תבנית מס' 2 – Singelton

* סיבת הבחירה / שימוש בתבנית: ממשנו את המחלקה "MainForm", אשר משמשת כטופס הראשי, כ-Singleton. הסיבה לבחירה היא שבמערכת זו קיים טופס אחד בלבד שתפקוד המערכת נקבע לפיו. מצב בו קיימים מספר טפסים (Forms) עלול לגרום נזק לתפקוד ולשימוש במערכת.
* אופן המימוש:
  + המחלקה "MainForm" כעת מוגדרת כ-"sealed", זאת על מנת שלא יהיה ניתן לרשת ממנה (אם יהיה ניתן לרשת ממחלקה זו, אזי יהיה ניתן ליצור object שהוא "MainForm").
  + ה-"constractor" מוגדר כעת כ-"private", זאת על מנת שלא יהיה ניתן ליצור מופע של ה-object מחוץ למחלקה.
  + הוספת "property" שמוגדר כ-"static" ואתחולו כ-"null".
  + הוספת "method" שמוגדרת כ-"static", אשר מחזירה את המופע של המחלקה. בתוך ה-"method" מתבצעת בדיקה האם התבצע אתחול של המחלקה טרם הקריאה ל-"method". אם כן, אזי מוחזר המופע של המחלקה שכבר נוצר טרם הקריאה, אחרת מתבצע אתחול של מופע חדש של המחלקה באמצעות ה-"constractor" של המחלקה, שהגישה אליו ניתנת אך ורק מתוך המחלקה.
* Sequence Diagram



* Class Diagram



### 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

במערכת ביצענו שימוש בעבודה אסינכרונית ע"י שימוש ב-threads נוספים אשר כל אחד מהם מפעיל טופס משני (תצוגה מקדימה ומפורטת של פוסטים, תצוגה מוגדלת של תמונות והעלאת תמונה חדשה לאלבום).

העבודה האסינכרונית במערכת נמצאת במחלקה "MainForm", בה באמצעות threads נעשת הפעלה של הטפסים המשניים, וכמו כן במחלקה "SubFormPhotoUpload" על מנת להפעיל File Dialog. השימוש בעבודה אסינכרונית מעניק ממשק עשיר יותר לשימוש במערכת ויכולות נוספות עבור המשתמשים.

### Data Binding

במערכת ביצענו שימוש ב-Data Binding בהצגת הפרטים של המשתמש אשר מחובר למערכת. הצגת פרטי המשתמש נמצאת בטופס הראשי של המערכת במחלקה "MainForm".

באמצעות השימוש ב-Data Binding ניתן לאגד את הנתונים המוצגים על המשתמש ביחידת לוגית דינמית אשר מתעדכנת במידה ומשתנים פרטים מסויימים.