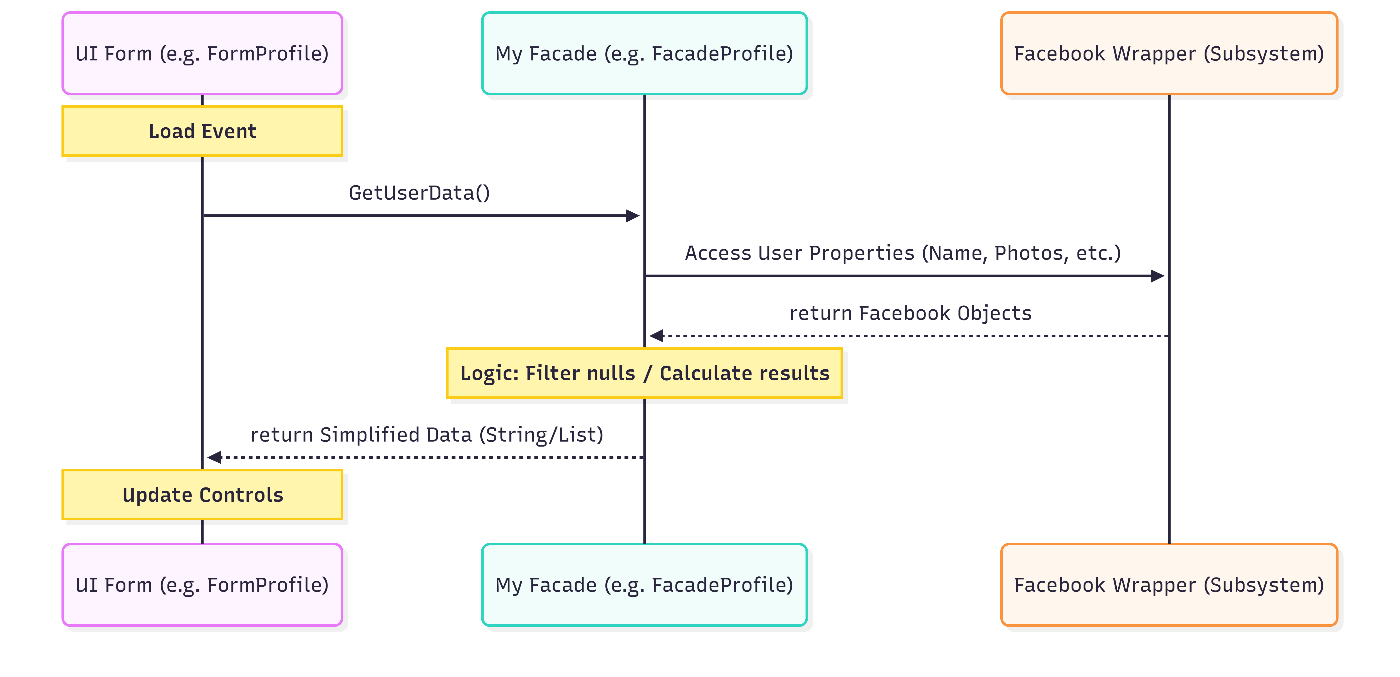
### תיאור קצר של הפיצ'רים שבחרנו לממש בתרגיל הראשון:

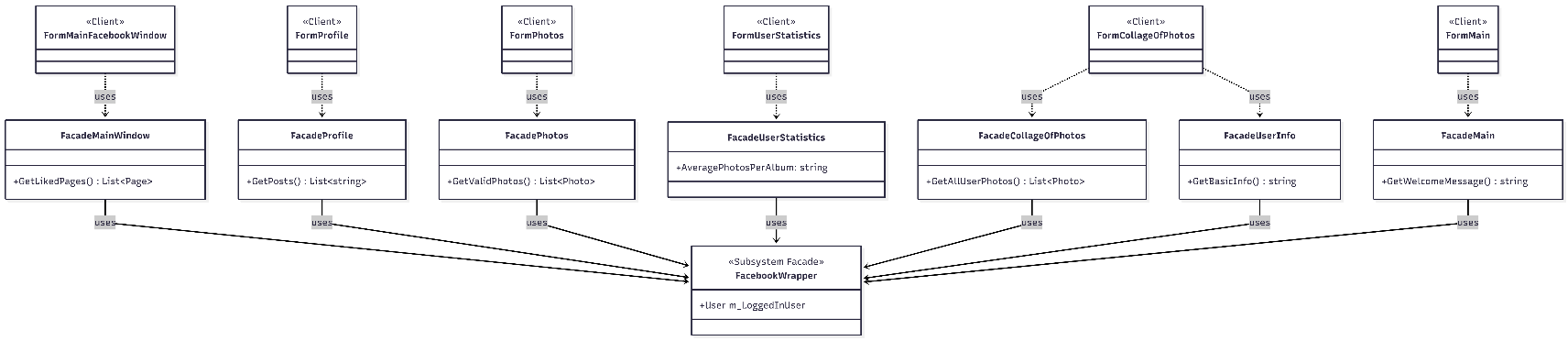
* Collage of moments: קולאז' דינמי של 4 תמונות המתחלפות אוטומטית בכל 2 שניות. עם טעינת הטופס, המערכת שולפת את כל התמונות מכל האלבומים של המשתמש ויוצרת רשימה מרוכזת. טיימר פנימי עובר על הרשימה ומעדכן את תיבות התמונה (PictureBoxes). המשתמש יכול לחזור לתפריט הראשי בכל עת באמצעות כפתור.
* User Statistics: הצגת נתוני סטטיסטיקה על פעילות המשתמש בפייסבוק. נתונים על פוסטים, אלבומים, תמונות ולייקים. סטטיסטיקות מוצגות:כמות חברים, כמות פוסטים,כמות תמונות ממוצעת לאלבום והתמונה שקיבלה הכי הרבה לייקים. הפיצ'ר מאפשר ניווט חזרה לדף הבית או המשך לדפי הפרופיל והתמונות.

### תבנית מס' 1 – Facade

* סיבת הבחירה / שימוש בתבנית: הצורך נבע מהרצון להוציא את הלוגיקה מה Forms למקום מרכזי. ה Facadeמבודד את הטפסים מהפורמט של פייסבוק, כך שאם בעתיד פייסבוק ישנו את מבנה הנתונים שלהם, נצטרך לעדכן רק את מחלקות הfacade בזמן שממשק המשתמש ישאר בלי שינויים.
* אופן המימוש: תבנית ממומשת באמצעות מחלקות FacadeMainWindow FacadeProfile FacadePhotos FacadeUserStatistics FacadeCollageOfPhotos FacadeUserInfo FacadeMainכל מחלקת Facade מחזיקה בתוכה את המופע של המשתמש m\_LoggedInUser ומנהלת מולו את האינטראקציה. היא מספקת לטופס מתודות פשוטות GetLikedPages GetUserName GetAveragePhotosPerAlbum למשל שמחזירות מידע בלי שהטוםס יצטרך להכיר את הלוגיקה. ה-Facade מרכז את כל הלוגיקה בתוכו. הטפסים נשארים רזי ואחראים רק על הצגת הנתונים למשתמש וניהול אירועים בUI.
* Sequence Diagram

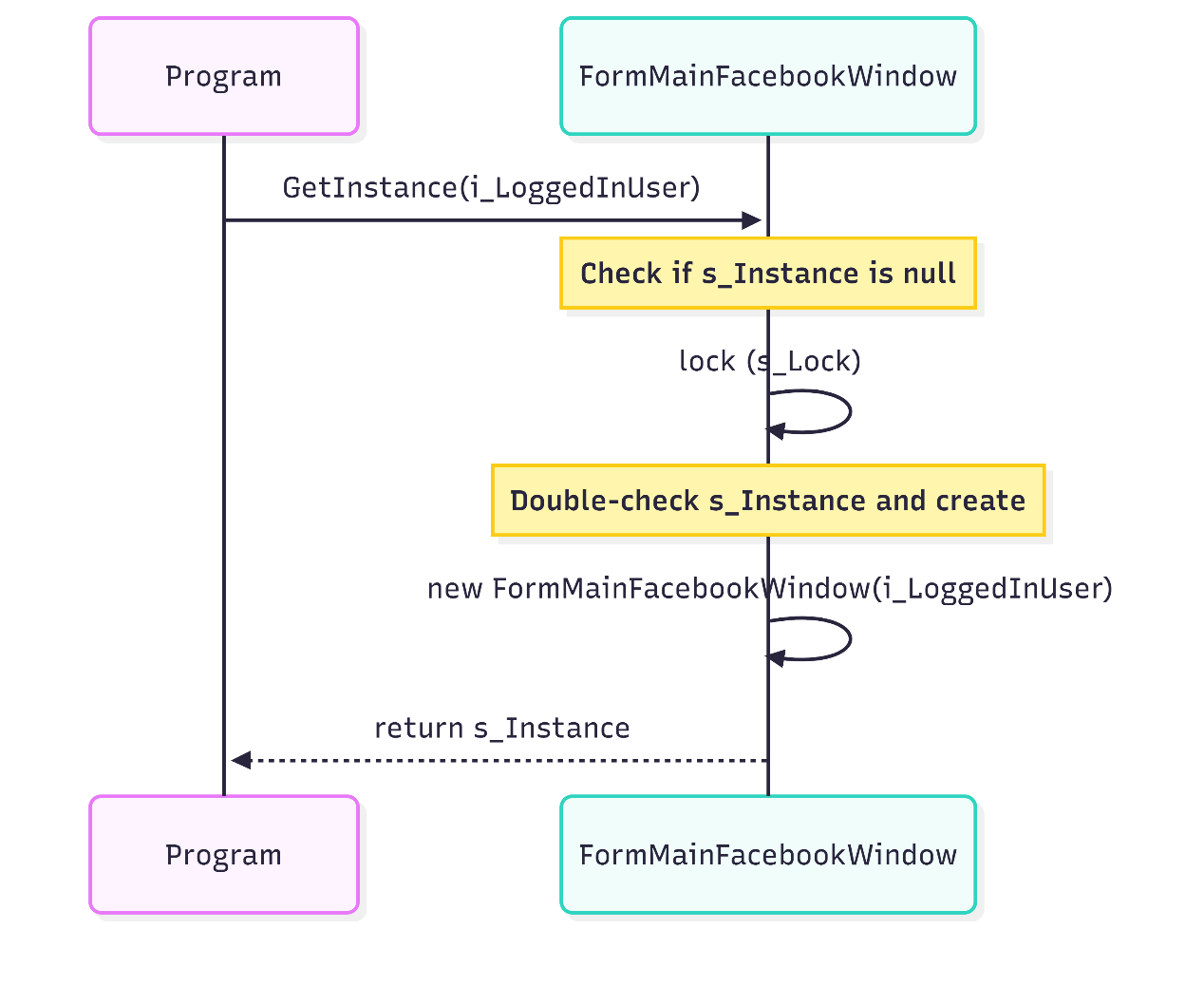


* Class Diagram

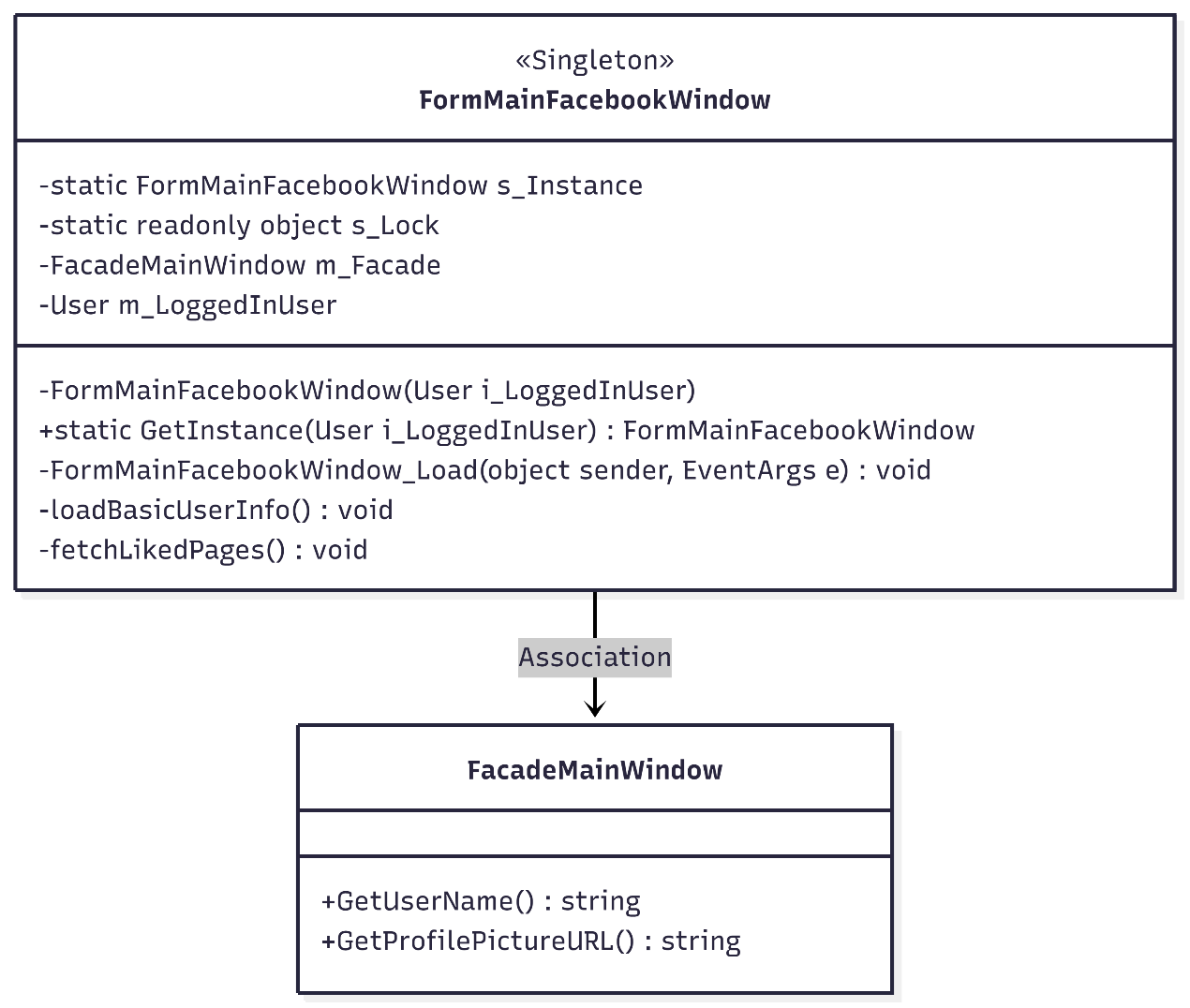


### תבנית מס' 2 – Singleton

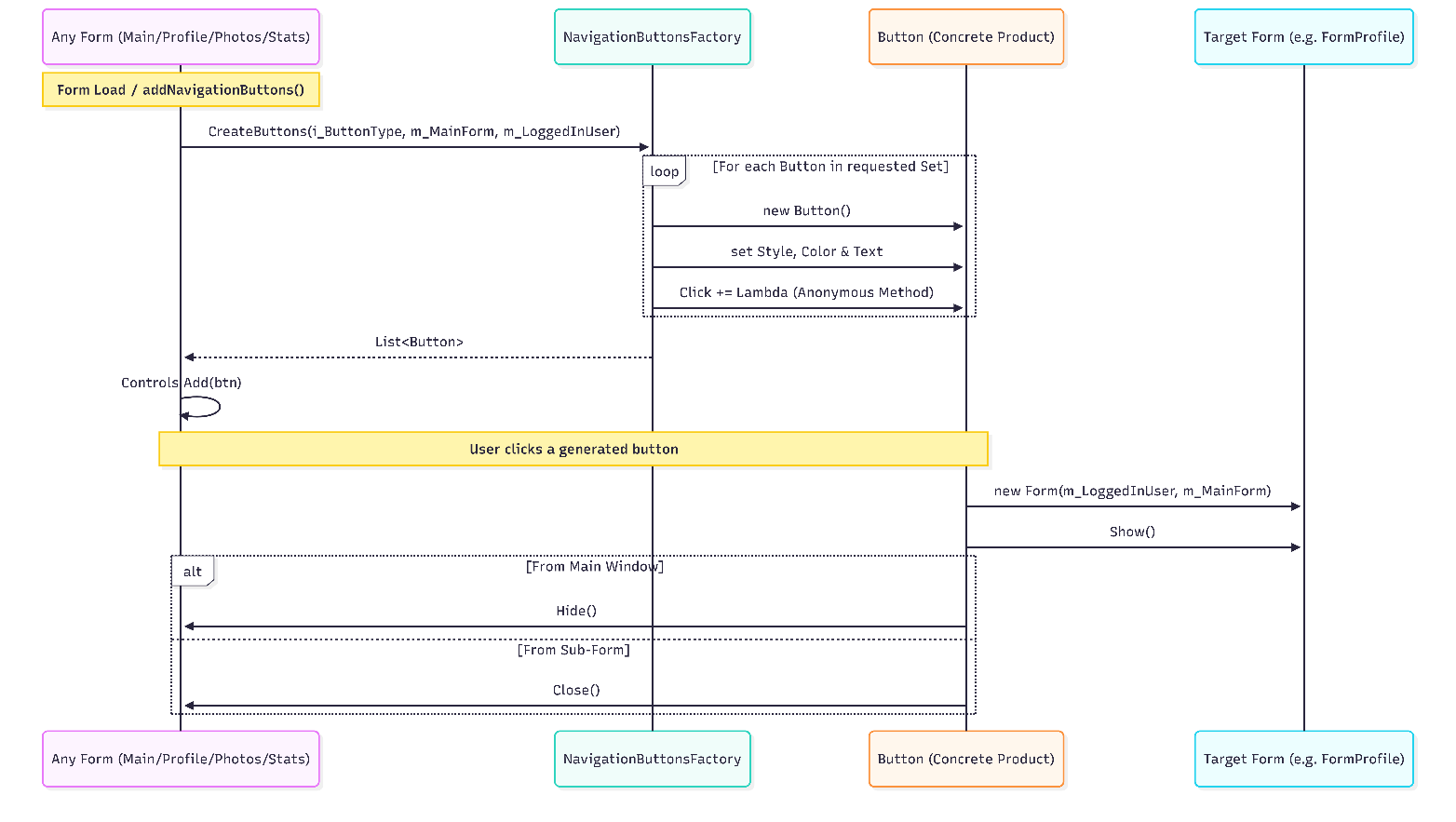
* סיבת הבחירה:השתמשנו בתבנית במחלקת " FormMainFacebookWindow" הצורך נבע מהרצון להבטיח שקיים רק מופע אחד (Instance) של החלון הראשי לאורך כל חיי האפליקציה. מכיוון שחלון זה מחזיק את נתוני המשתמש המחובר (m\_LoggedInUser) ואת ה-Facade הראשי, יצירה של מספר חלונות זהים במקביל עלולה לגרום לחוסר עקביות בנתונים ובעיות בניווט בין הדפים השונים באפליקציה.
* אופן המימוש: ה-Constructor הוגדר כ-private כדי למנוע יצירת אובייקטים מבחוץ באמצעות האופרטור new. הוגדר שדה סטטי פרטי בשם s\_Instance המחזיק את המופע היחיד של המחלקה. הגדרנו מתודה סטטית פומבית GetInstance עם אובייקט נעילה בשם s\_Lock המבטיח שהאובייקט ייווצר פעם אחת בלבד (Thread Safety). בנוסף המחלקה מוגדרת sealed כדי למנוע הורשה מהחלקה.
* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Factory Method

* סיבת הבחירה / שימוש בתבנית: השתמשנו ב Factory Method כדי לנהל את יצירת כפתורי הניווט בדפים השונים.הצורך נבע מהרצון לשמור על עיצוב זהה ולוגיקת ניווט (פתיחה וסגירה של חלונות קודמים) בכל האפליקציה במקום אחד מרכזי. דבר שמקל על תחזוקתיות הקוד.
* אופן המימוש: המימוש נמצא במחלקה הסטטית " NavigationButtonsFactory" המחלקה מכילה את המתודה CreateButtons שמקבלת את סוג סט הכפתורים לפי סוג הדף (אחד מתוך eButtonsTypes) ומחזיקה רשימה של אובייקטי Button. המפעל יוצר את הכפתורים עם עיצוב אחיד ומצמיד לכל כפתור EventHandler הכולל את הלוגיקה המדויקת למעבר בין הטפסים הרלוונטיים. וסגירת הטופס הנוכחי
* Sequence Diagram
* Class Diagram