

**תיאור קצר של הפיצ'רים שבחרנו לממש:****1. Top three profile pictures**

- היכן למצוא בקוד:

במחלקה FormMain המתודה buttonTopThreePhotos\_Click() פותחת את הטופס של הפיצ'ר, במחלקה FormTopThreePhotos ממומש ה-UI של הפיצ'ר, ובמחלקה TopThreePhotosFeature ממומשת הלוגיקה של הפיצ'ר.

- הסבר על הפיצ'ר:

המשתמש יכול לצפות בשלושת תמונות הפרופיל הכי פופולריות שלו, כלומר התמונות שמכילות את מספר התגובות הרב ביותר ומי מהחברים שלו (זכר ונקבה) הגיב לו הכי הרבה. במידה ואין למשתמש תמונות פרופיל כלל או שיש לו פחות משלוש המשתמש יבחין בכך בטופס. בנוסף, בטופס מוצג מספר התגובות שיש לכל תמונה וישנה אפשרות לראות את התגובות עבור כל תמונה (ע"י לחיצה על תמונה מסויימת). בעקבות הרשאות פייסבוק לא ניתן לגשת לתגובות החברים של המשתמש, לכן יוצגו רק תגובות המשתמש עצמו על התמונות.

**2. Birthday in a week**

- היכן למצוא בקוד:

במחלקה FormMain המתודה OnShown() מפעילה את הפיצ'ר. במחלקה FormBirthdayInAWeek ממומש ה-UI של הפיצ'ר ובמחלקה BirthdayInAWeekFeature ממומשת הלוגיקה שלו.

- הסבר על הפיצ'ר:

בעת כניסתו של המשתמש לטופס הראשי יוצג טופס בו יצטרך לסנן את החברים לפי מגדר. כתוצאה מכך, תופיע הודעה המכילה את שמות החברים שלו אשר חוגגים יום הולדת בדיוק בעוד שבוע מהתאריך הנוכחי, לפי המגדר שהמשתמש בחר לסנן. בנוסף, בטופס ישנה גישה לאתר ממנו אפשר לקנות מתנות יומולדת.

מגישים: [Maya Tal], [Lidar Ben-Dor]

**תבנית מס' 1 – [Factory Method]**

- סיבת הבחירה / שימוש בתבנית:

ניתן לראות כי ישנה משפחה פולימורפית של טפסים שיורשת מהמחלקה Form, שמונה את הטפסים המוצגים למשתמש באפליקציה. בנוסף ישנו מכנה משותף בין כל הטפסים, בעת יצירת הטופס אנו משתמשים ב- LoggedInUser.

- אופן המימוש:

Client: FormMain ○

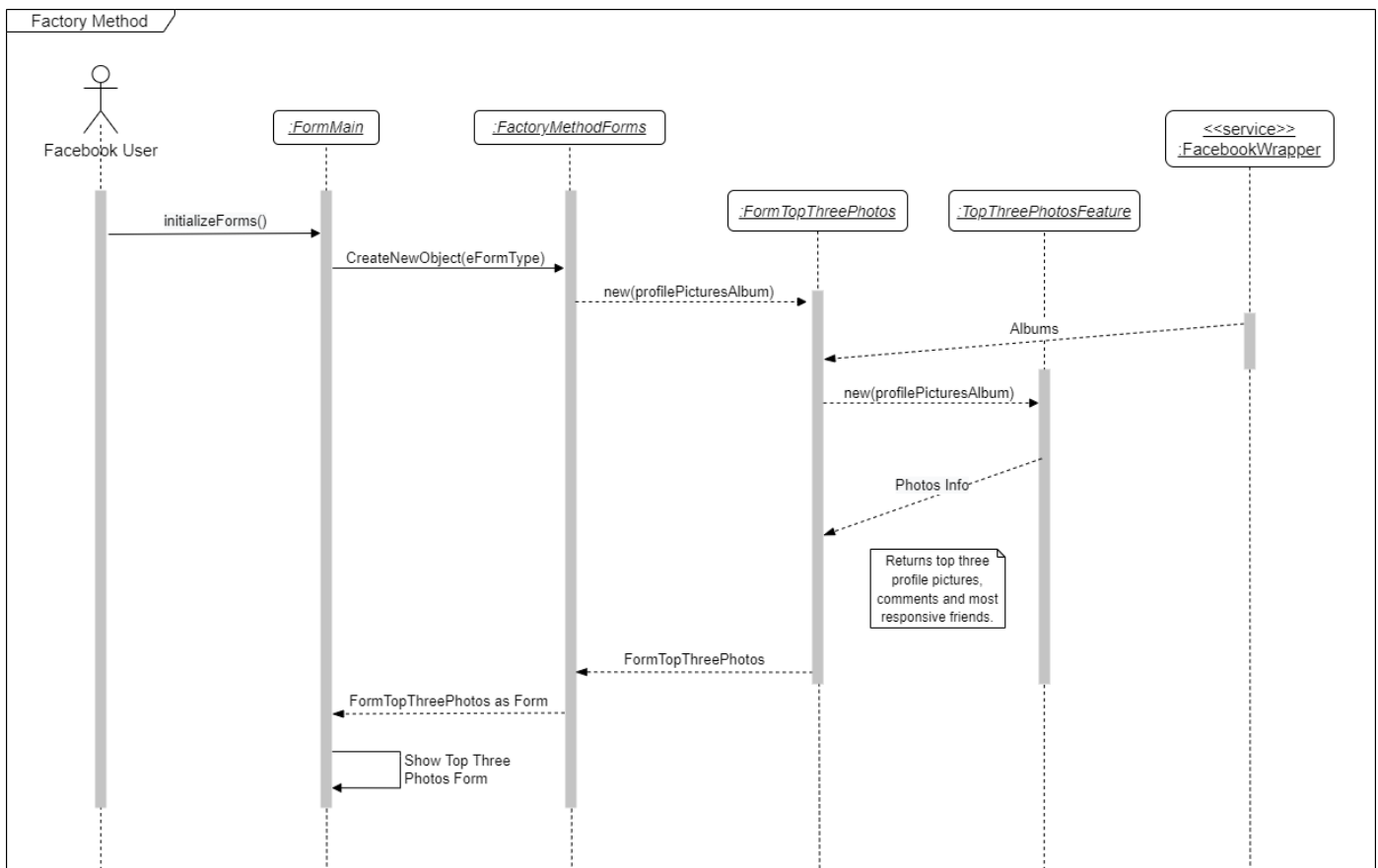
Factory: FactoryMethodForms ○

Products: Facebook Forms ○

המחלקה FormMain מחזיקה טפסים שונים המאותחלים ע"י המתודה CreateNewObject, שנמצאת במחלקה FactoryMethodForms. המתודה מקבלת שני פרמטרים: enum המייצג את סוג הטופס והמשתמש (LoggedInUser). המתודה הנ"ל מחזירה טיפוס מסוג Form זה מתאפשר משום שכל הטפסים יורשים מהמחלקה Form, כלומר זוהי משפחה פולימורפית.

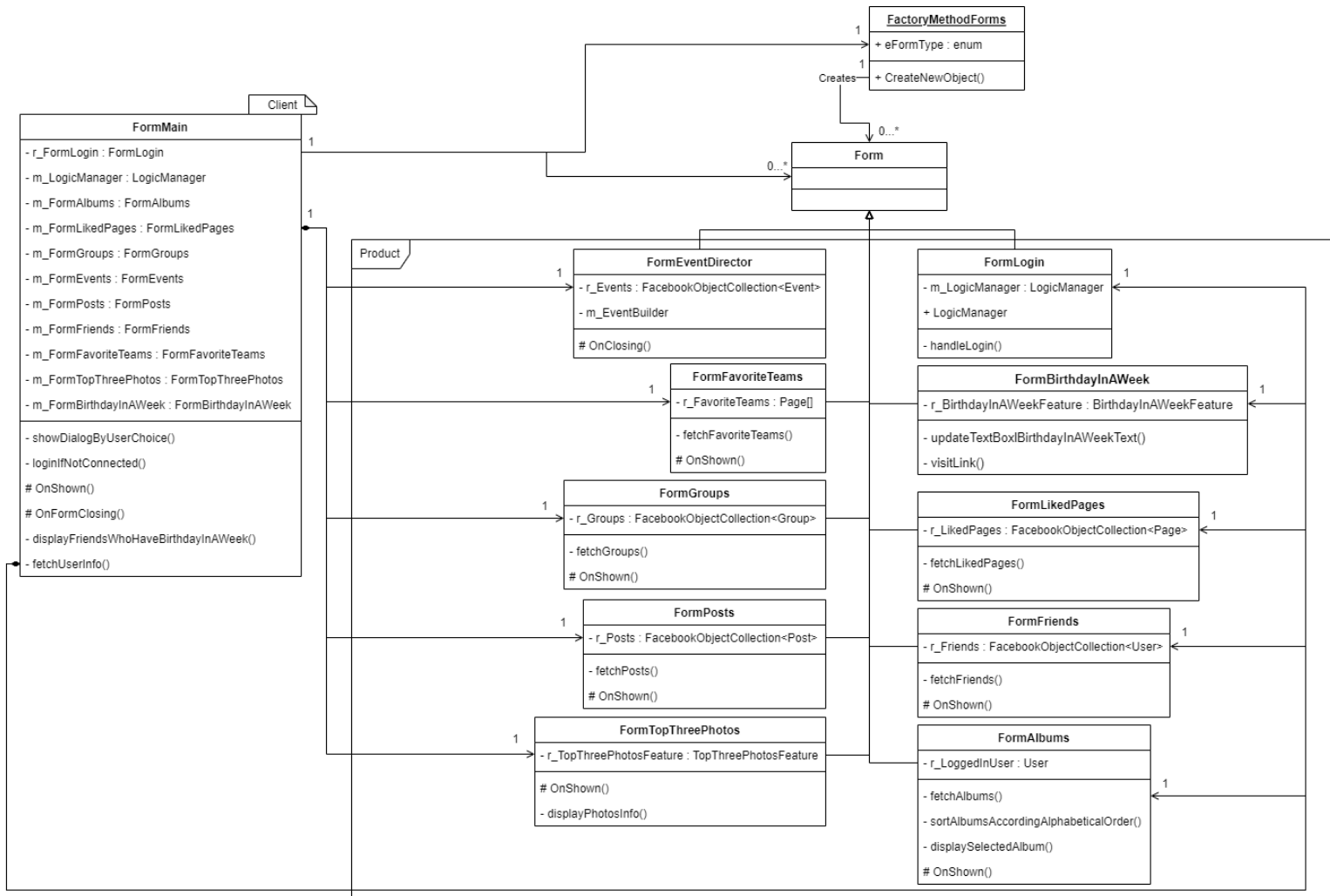
**Sequence Diagram**

בחרנו להתמקד בתרחיש המורכב ביותר, בו מייצרים טופס המכיל את שלושת התמונות הפופולריות ביותר של המשתמש.



מגישים: [Lidar Ben-Dor], [Maya Tal]

## Class Diagram •



## תבנית מס' 2 – [Facade]

- סיבת הבחירה / שימוש בתבנית:

השתמשנו בתבנית עבור הפיצ'ר שמראה את שלושת תמונות הפרופיל הפופולריות ביותר של המשתמש (TopThreePhotos).

רצינו לחשוף את המשתמש רק למידע הרלוונטי אליו בנוגע לפיצ'ר ולהסתיר ממנו את הלוגיקה שעומדת מאחורי תפעול הפיצ'ר.

- אופן המימוש:

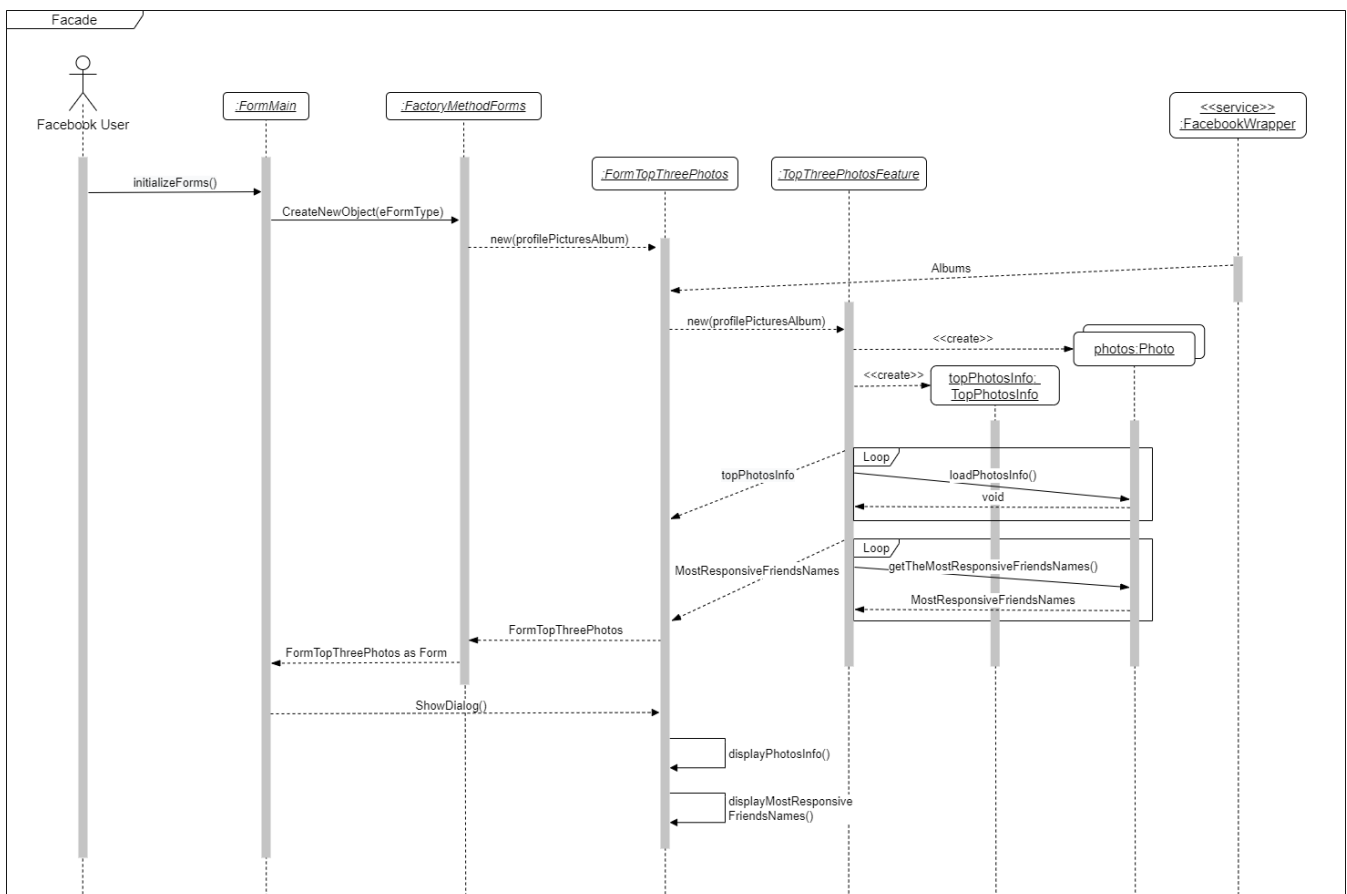
Client: FormTopThreePhotos ○

Facade: TopThreePhotosFeature ○

Sub-System Classes: TopPhotosInfo, Facebook Warpper ○

המחלקה TopThreePhotosFeature מכילה את הלוגיקה שעומדת מאחורי הפיצ'ר הנ"ל, אליה המשתמש אינו חשוף. מחלקה זו נעזרת במחלקות TopPhotosInfo, Facebook Warpper. המחלקה פונה ל-Facebook Warpper על מנת לקבל את מאגר התמונות של המשתמש ומוצאת את שלושת תמונות הפרופיל הפופולריות ביותר שלו. בנוסף, אנו עוברים על רשימת החברים שהגיבו למשתמש על התמונות ומתוך רשימה זו מוצאים את החברים המגיבים ביותר (זכר ונקבה). המחלקה FormTopThreePhotos מהווה את הממשק עם המשתמש ומקבלת את התוצר מהמחלקה TopThreePhotosFeature.

## Sequence Diagram





## תבנית מס' 3 – [Builder]

- סיבת הבחירה / שימוש בתבנית:

רצינו שחווית המשתמש באפליקציה תהיה נוחה ביותר, לכן איפשרנו למשתמש לסנן את האירועים שהוא נוכח בהם לפי תאריכים ומספר משתתפים. מכאן, התבנית הנ"ל עוזרת לנו לבנות את האובייקט המורכב שמכיל את כל האירועים של המשתמש לפי קריטריונים מסויימים.

- אופן המימוש:

Client: FormMain ○

Director: FormEventDirector ○

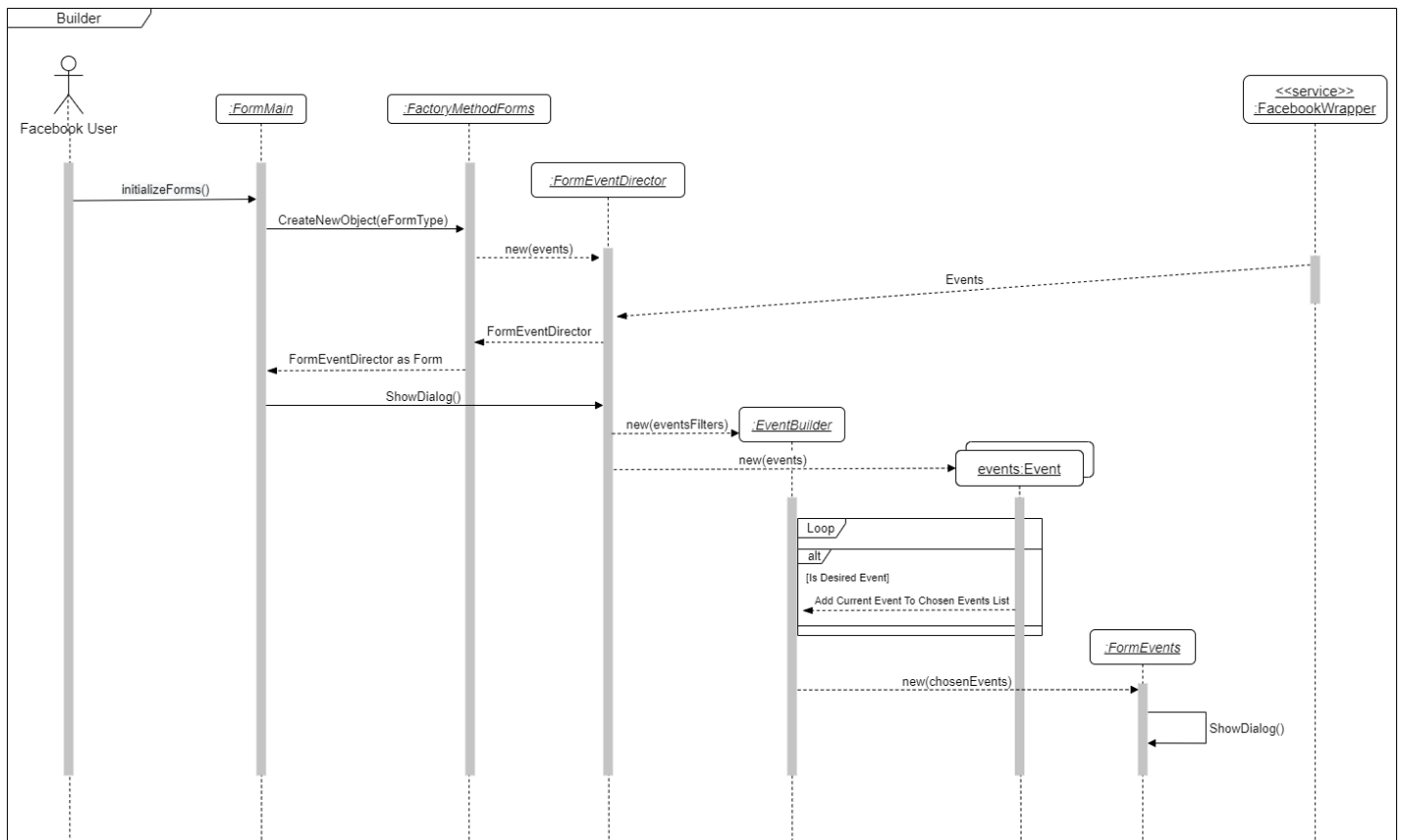
Builder: EventBuilder ○

Product: FormEvents ○

באמצעות המחלקה FormMain המשתמש יכול לגשת לאירועים שבהם הוא נוכח. כלומר המשתמש מנותב לטופס FormEventDirector, שם המשתמש מתבקש למלא פרטים לגבי המסננים לפיהם יוצגו לו האירועים- תאריך התחלה, תאריך סיום ומספר המשתתפים.

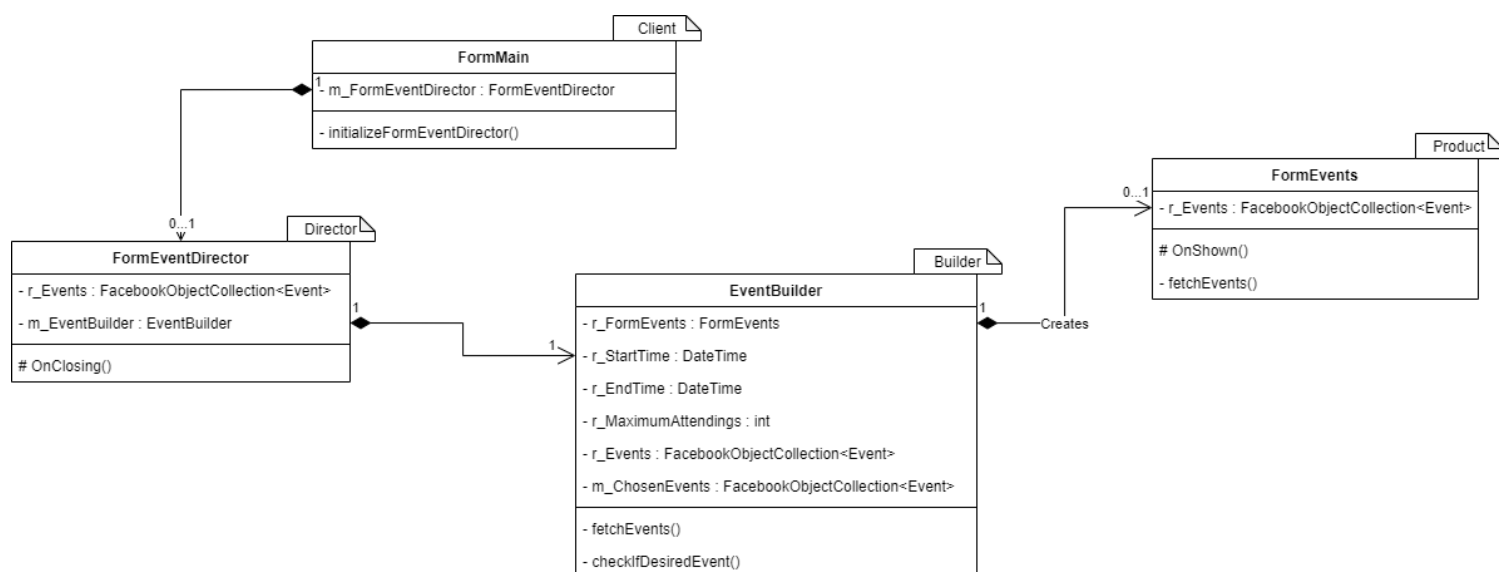
המחלקה EventBuilder מקבלת את הקריטריונים לסינון מהמחלקה FormEventDirector, והיא מאגדת את כל האירועים הרלוונטים למשתמש, כלומר מסננת את האירועים ממאגר האירועים של המשתמש. לבסוף, נוצר הטופס FormEvents שבו מוצגים כל האירועים הרלוונטים למשתמש.

## Sequence Diagram



מגישים: [Maya Tal], [Lidar Ben-Dor]

## Class Diagram •



## עבודה א-סינכרונית:

- סיבת השימוש:  
האפליקציה שפיתחנו פונה לשרתים של פייסבוק על מנת לקחת מידע על המשתמש. עקב כך שישנו הרבה מידע, הבאתו לוקחת זמן רב ובעת פתיחת האפליקציה המשתמש נאלץ להמתין פרק זמן יחסית ארוך עד לעלייתה.  
לכן, כאשר אנו משתמשים בעבודה אסינכרונית אנו ניגשים לשרתים במקביל, כלומר הבאת המידע לוקחת פחות זמן זה תורם לחווית המשתמש.  
אופן השימוש והיכן השתמשנו:
- במחלקה `FormMain`, בטופס הראשי, במתודה `initializeForms()`, יצרנו עבור כל טופס `Thread` חדש.  
בנוסף, בכל טופס כשקראנו למתודה `fetch` - שמביאה את המידע הרלוונטי לטופס הנוכחי - ייצרנו `Thread` חדש. במתודה `fetch` השתמשנו במתודה `invoke`.

## עבודה עם Data Bindings:

- אופן השימוש והיכן השתמשנו:  
השתמשנו ב-`Data Bindings` במחלקות:
  - `FormFavoriteTeams`
  - `FormGroups`
  - `FormFriends`
  - `FormEvents`
  - `FormLikedPages`
  - `FormAlbums`בכל אחת מהמחלקות לקחנו את המידע הספציפי שרצינו להציג למשתמש באמצעות המידע שנמצא ב-`Data Source`, והצגנו אותו בטופס.



## תבנית מס' 4 – [Strategy]

- סיבת הבחירה / שימוש בתבנית:

בפיצ'ר "Birthday in a week" ישנה אפשרות למשתמש לסנן את החברים שלו לפי מגדר. כלומר אפשרויות הסינון של המשתמש בעלות בסיס משותף שהוא בחירת מגדר החברים, אך המימוש של הסינון מתבצע באופן שונה. התבנית מאפשרת לנו להשתמש באלגוריתם הסינון הנכון בזמן ריצה.

- אופן המימוש:

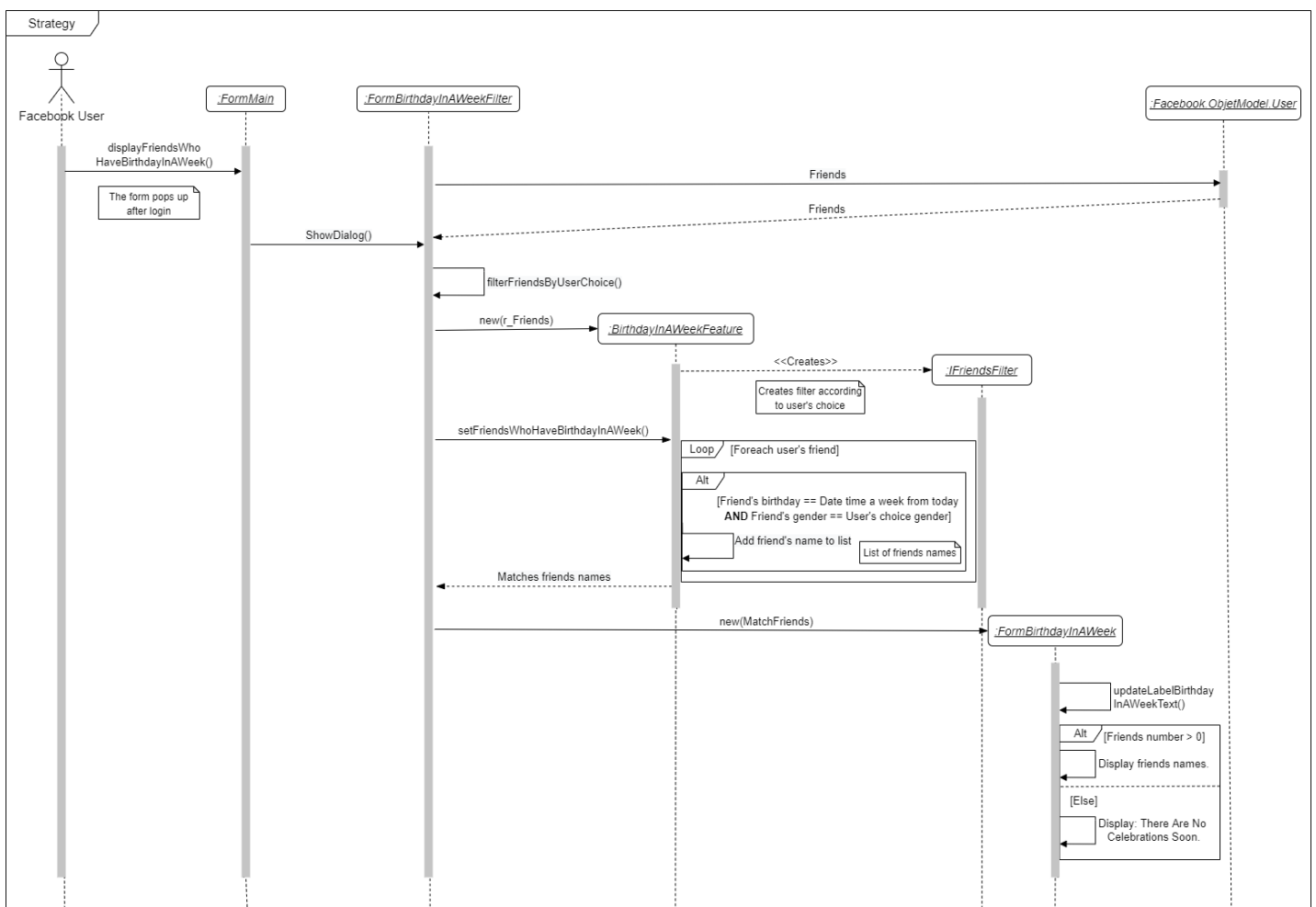
Strategy: IFriendsFilter ○

Concrete Strategy: FemaleFilter, MaleFilter, FemaleAndMaleFilter ○

Context: BirthdayInAWeekFeature ○

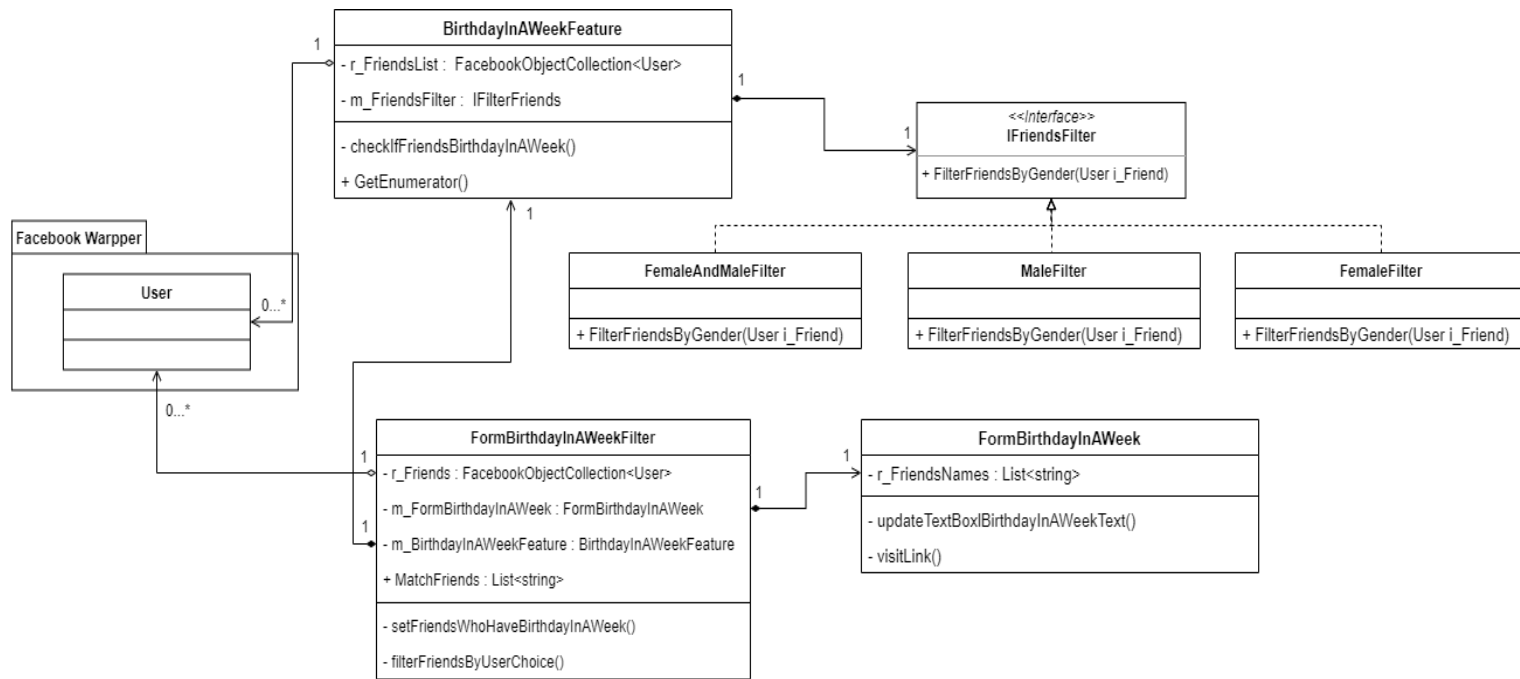
במחלקה BirthdayInAWeekFeature מוחזרים כל החברים שעברו את הסינון, באמצעות המחלקה IFriendFilter המכילה מתודה "FilterFriendByGender" - שהיא מסננת את החברים לפי המגדר שהמשתמש בחר. המתודה ממומשת באופן שונה בכל אחת מהמחלקות: FemaleFilter, MaleFilter, FemaleAndMaleFilter.

## Sequence Diagram



מגישים: [Lidar Ben-Dor], [Maya Tal]

## Class Diagram •



מגישים: [Maya Tal], [Lidar Ben-Dor]

**תבנית מס' 5 – [Iterator]**

- סיבת הבחירה / שימוש בתבנית:

במחלקה BirthdayInAWeekFeature שנמצאת בחלק הלוגי, קיים אוסף של חברים, ונרצה להחזיר למשתמש את רשימת החברים לאחר סינון. אנו רוצים לתת גישה למשתמש לסרוק את אוסף החברים מבלי לחשוף את מבנה הנתונים של אחד מהאובייקטים וללא תלות במימוש מבנה הנתונים שלהם, על מנת שבעתיד תהיה לנו האפשרות לבצע שינויים במחלקה הנ"ל ללא שינוי הקוד אצל המשתמש.

- אופן המימוש:

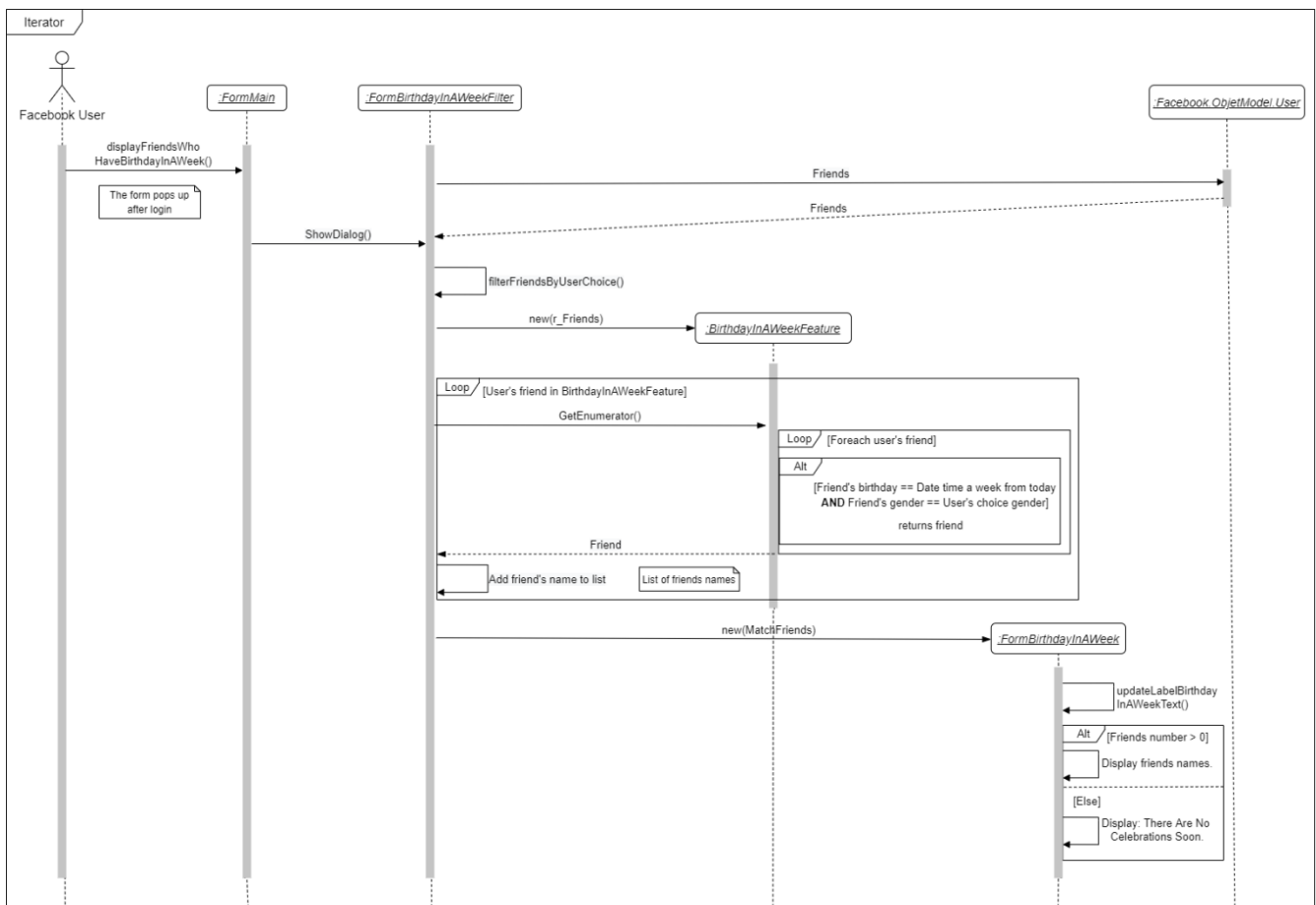
Client: FormBirthdayInAWeekFilter ○

Aggregate: BirthdayInAWeekFeature ○

האיטרטור מסופק ע"י רשימת החברים המוחזקת במחלקה

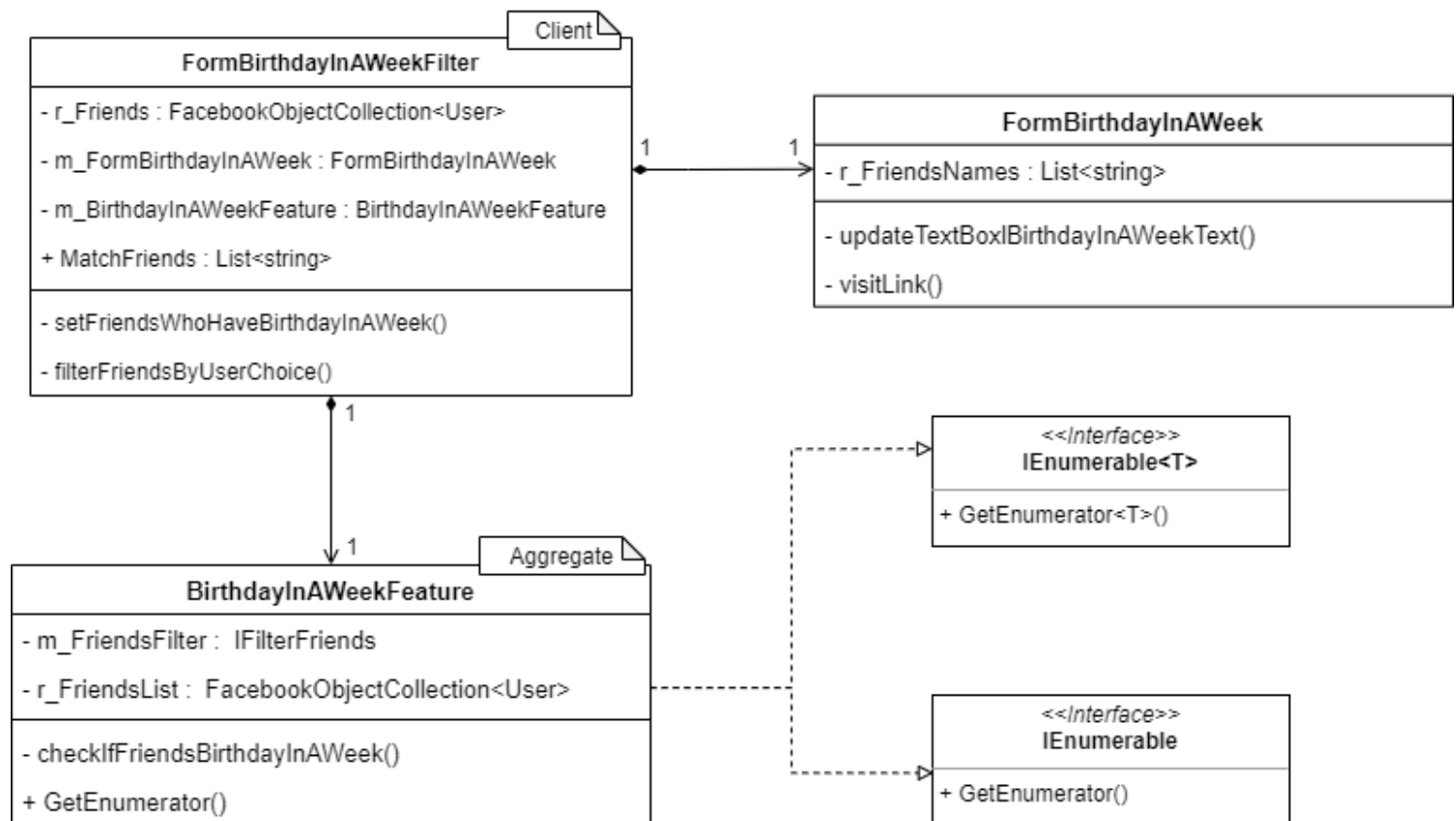
BirthdayInAWeekFeature

במחלקה FormBirthdayInAWeekFilter המשתמש בוחר כיצד לסנן את החברים שלו, ואנו עוברים בלולאת foreach על האובייקט BirthdayInAWeekFeature שבממשים Filter Iterator. המימוש מתבצע בכך שאנו עוברים על רשימת החברים של המשתמש במתודה GetEnumerator() ובכל איטרציה בודקים האם החבר עומד בקריטריונים הנ"ל. במידה וכן נבצע yield return שיחזיר את ה- iterator.

**Sequence Diagram**

מגישים: [Maya Tal], [Lidar Ben-Dor]

## Class Diagram •



## תבנית מס' 6 – [Observer]

- סיבת הבחירה / שימוש בתבנית:

בטופס הראשי ישנה אפשרות למשתמש לשנות את רקע הטופס. ברגע שהמשתמש לוחץ על רקע מסויים אנו רוצים להודיע לטופס הראשי שהמשתמש רוצה לשנות לרקע הנבחר. התבנית מאפשרת לאובייקטים שונים להרשם כמקשיבים לאירוע, ולכן הטופס הראשי יוכל להקשיב לבחירת המשתמש בנושא הנ"ל. אם בעתיד יהיו אובייקטים נוספים שירצו להרשם כמקשיבים לאירוע, בעזרת התבנית השינוי בקוד יהיה מינורי והם רק יצטרכו להירשם לאירוע.

- אופן המימוש:

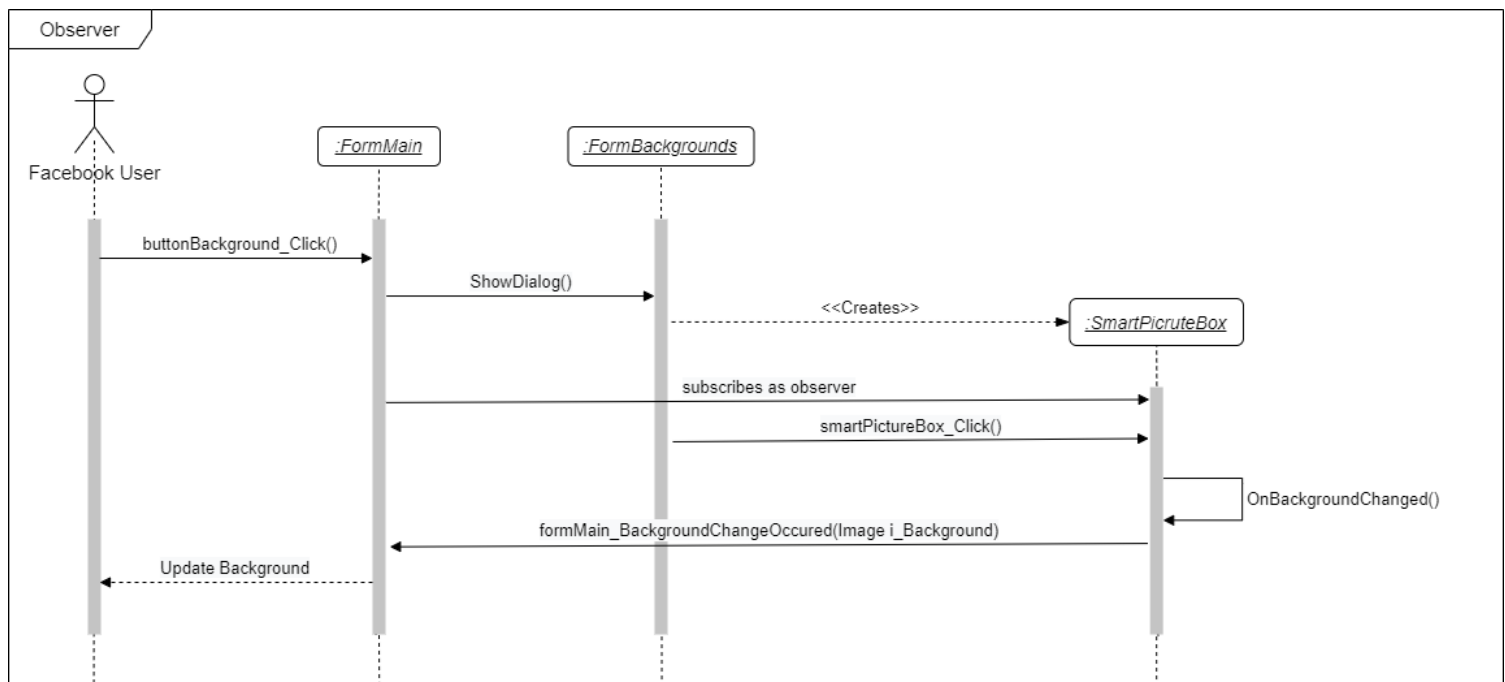
Subject: PictureBox ○

Concrete Subject: SmartPictureBox ○

Observer: FormMain ○

יצרנו את המחלקה SmartPictureBox אשר יורשת מהמחלקה PictureBox, בה יצרנו אירוע BackgroundChanged מסוג Action<Image>. כאשר המשתמש יבחר רקע, נודיע באמצעות האירוע הנ"ל לכל המקשיבים שהמשתמש רוצה לשנות לרקע שבחר. במחלקה FormMain (הטופס הראשי) המשתמש יכול להחליט שהוא משתנה רקע לטופס. בנוסף, במחלקה זו יצרנו אובייקט מסוג SmartPictureBox והטופס הראשי נרשם לאירוע שלו.

## Sequence Diagram



מגישים: [Lidar Ben-Dor], [Maya Tal]

## Class Diagram •

