### 

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

* **Guess Who Game:**

בכדי להתחיל משחק זה יש ללחוץ על ה:Tab GuessWho , ופיצ'ר זה יופעל.

יצרנו פיצ'ר באפליקציה שהוא בעצם משחק, על מנת להעניק למשתמש חווית משחק עם המידע האישי שאנו מביאים על היוזר מFacebook.

על המשתמש לנחש את שם חברו המופיע בתמונה מטושטשת שלרשותו 7 חיים בלבד.

ראשית, על המשתמש לבחור איזה סוג אפקט תמונה יטשטש את תמונת חברו שעליו לנחש – Gamma, Tranparency or Blur. לאחר מכן, יתחיל המשחק עם סוג אפקט התמונה הנבחר. במהלך המשחק לרשות המשתמש לקחת עצה (Hint), כשכל עצה מבהרת במקצת את תמונת החבר המטושטשת ומורידה למשתמש חיים אחד. אם המשתמש מצליח לנחש את שם חברו כאשר אינו ביזבז את כל שבעת החיים שלרשותו המשתמש ניצח.

* **Events Center**

הוספנו פיצ'ר עריכת איוונט

בטאב שבחלון המרכזי, כאשר היוזר ילחץ על תאריך מסוים בלוח השנה, יופיעו בליסט בוקס של האיוונטים, כל האיוונטים הרלוונטים שחלים בתאריך הזה

לחיצה על אחד האיוונטים תפתח חלון חדש שבו יופיעו נתונים מסוימים מהאירוע, כאשר כל נתון ניתן לעריכה, כאשר המשתמש ילחץ על OK, העריכה תשמר והנתונים החדשים יופיעו בלוח הראשי

**דאטה ביינדינג:**

ממשנו את הדאטה ביינדינג manually, בעריכת אירוע.

כאשר המשתמש לוחץ על OK בחלון העריכה EditEventForm האירוע שנערך ישתנה ברשימת האירועים של החלון הראשי, כמו כן חלון העריכה יציג את הנתונים של האירוע.

### תבנית מס' 1 – MethodFactory

* סיבת הבחירה / שימוש בתבנית:

ראשית, השתמשנו בתבנית זו בכדי להימנע משיכפול קוד, בכל פעם שנרצה ליצור אובייקט מהמשפחה הפולימורפית של ImageFilter. בתבנית זו, שאנו מייצאים את תהליך יצירת האובייקטים לפונקציה נפרדת גנרית, נוכל לזמן פונקציה זו בכל פעם שנרצה ליצור אובייקט מהמשפחה הפולימורפית של ImageFilter .

שנית, מטרת התבנית היא להכמיס ולספק ממשק פשוט ליצירת אובייקט קונקרטי של ImageFilter מתוך המשפחה הפולימורפית שלו. בצורה זו הClient, לא צריך להכיר את לוגיקת היצירה של איזה אובייקט קונקרטי של ImageFilter ליצור, ולוגיקת יצירת האובייקט "ארוזה" במקום אחד מה שמקל על תחזוקת המערכת בעתיד.

* אופן המימוש:

**השחקנים בתבנית:**

**MethodFactory:** ImageFactory

מחלקת האבא הינה: ImageFilter.

כאשר הנורשים הינם: ImgTransparency, ImgBlur, ImgGamma.

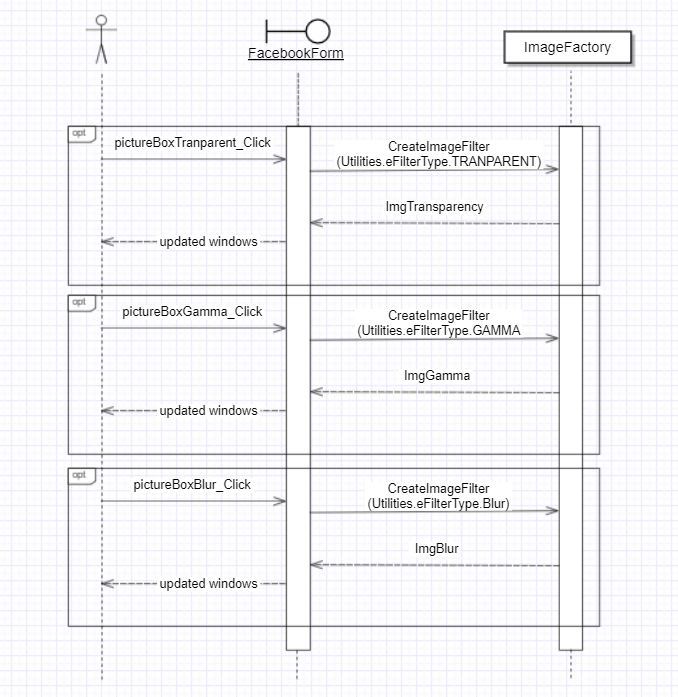
יצרנו משפחה פולימורפית כמופיע לעיל.

תחילת המשחק, בפני המשתמש מוצגות 3 תמונות שונות כאשר כל תמונה עם האפקט הייחודי שלה, תמונה אחת לכל אפקט. על המשתמש לבחור את סוג אפקט התמונה על ידי לחיצה על התמונה עם סוג האפקט המבוקש. לאחר מכן, מופעלת הפונקציה CreateImageFilter עם enum (eFilterType), סוג האפקט.

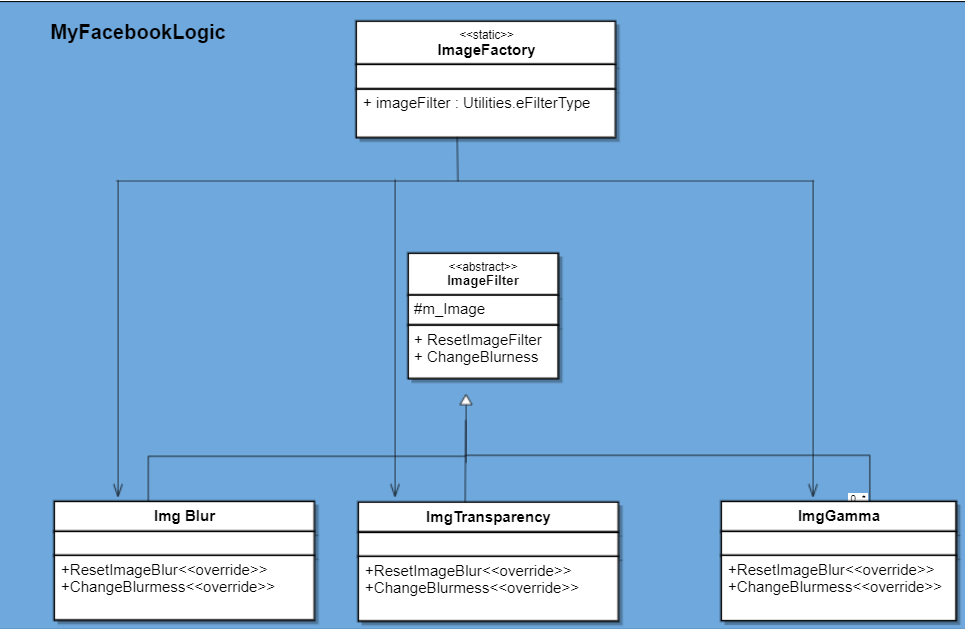
לאחר מכן, נוצר במחלקת ImageFactory התמונה עם האפקט הנדרש, ומופיעה לפני המשתמש לתחילת המשחק.

כמו כן, המתודה CreateImageFilter מקבלת eFilterType המשמש לבחירה של הטיפוס של האובייקט הקונקרטי הנבחר על ידי המשתמש.

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Facade

* **סיבת הבחירה :**

בחרנו להשתמש בתבנית זו בכדי לפשט את גישת המשתמש לממשק ולכמס רכיבים לוגיים מהממשק בשביל תחזוקה עתידית.

כמו כן, מחלקה זו מאפשרת הפרדה ברורה בין רכיבי הלוגיקה ובין רכיבי הUI ומאחדת את כל רכיבי הלוגיקה דבר העוזר לנוחות השימוש, עושה את המחלקות גנריות יותר (דבר התורם לReuse בעתיד) ומאפשרת צירוף רכיבי לוגיקה בצורה קלה יותר.

**אופן המימוש:**

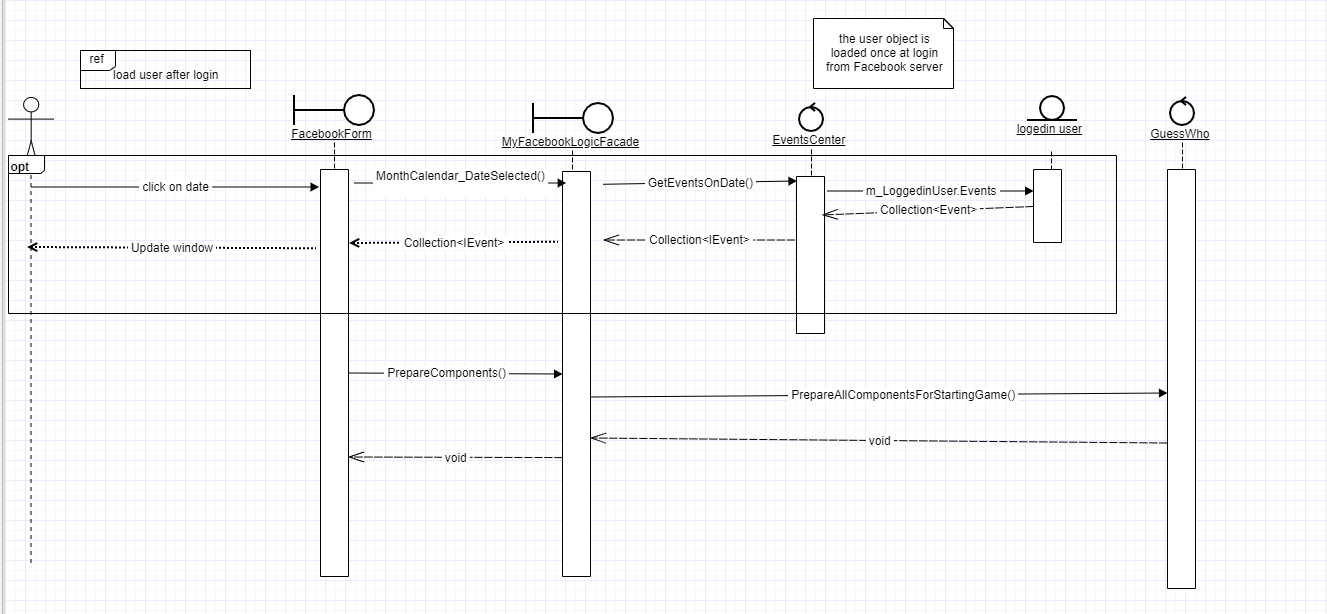
**השחקנים בתבנית:**

* :ClientFacebookForm
* Facade: MyFacbookLogicFacade
* SubsystemA: GuessWho
* EventsCenter :SubsystemB

הלקוח מחזיק מופע אחד של הFacade המכיל בתוכו את מופעי הפיצ'רים.

כאשר יש צורך בהפעלה או תפעול של פיצ'ר מסוים, הלקוח יפנה לאובייקט הנ"ל אשר יפעיל את הלוגיקה ע"י תפעול הsubSystems המתאים.

* Sequence Diagram



* 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 – Proxy

* סיבת הבחירה / שימוש בתבנית:

התאמת אובייקט Event לצרכים שלנו, הוספת היכולת לערוך.

אובייקט איוונט לא ניתן לעריכה, רצינו לתת למשתמש את החוויה שהוא יכול לערוך את האירועים שלו, לשם כך יצרנו את eventProxy שידמה את event ויוסיף את היכולת לערוך את השדות הרלוונטים עבורנו.

* אופן המימוש:

יצרנו מחלקה בשם eventProxy שיורשת מIEvent ומחזיקה ממבר Event שאליו היא רוצה להתחקות.  
כעת, כל קליינט שירצה להשתמש בEvent יחזיק משתנה של IEvent שנותן לו אותה חוויה כמו Event רגיל ובנוסף היכולת לערוך שדות מסוימים ולהעביר את האיוונט לtoString שמחזיר את ערכי השדות מופרדים עם דלמיטר.

IEvent – מכיל את השדות ש"מעניינות" את האפליקציה שלנו

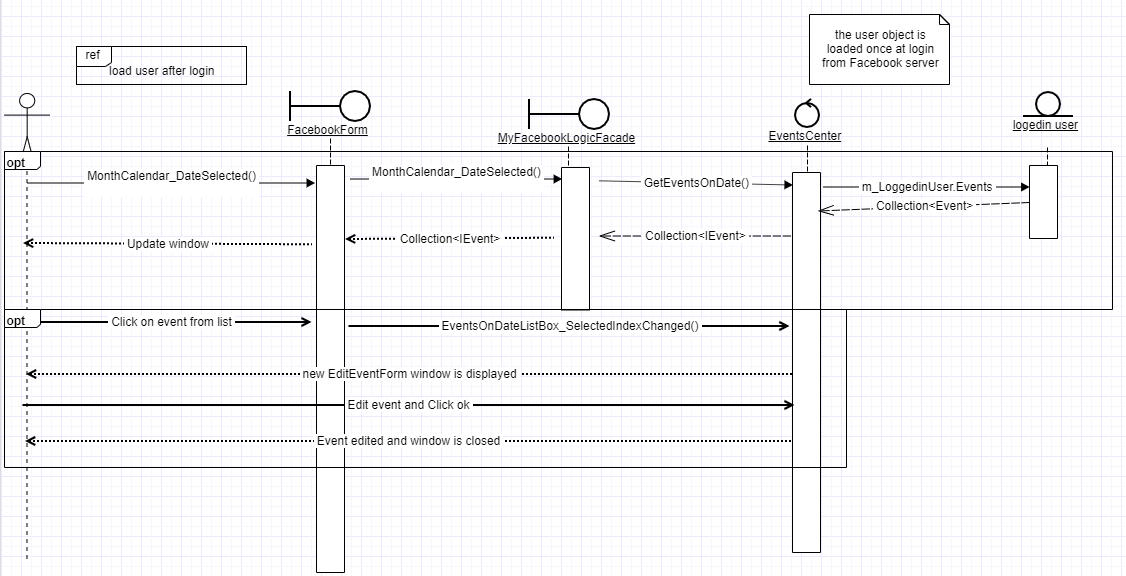
קליינטים:

EditEventForm

FacebookForm

EventsCenter

* Sequence Diagram



knfljfdnjfdbdjkvfsjגדהנלגדנהדלח

\*המרובע הלבן עם המסגרת הכתומה מייצג קליק של היוזר על איוונט מרשימת האיוונטים.

* Class Diagram

