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

* פיצ'ר ראשון: מציג את האלבומים של היוזר שהתחבר, ומתוך אחד האלבומים ניתן לבחור תמונה, לערוך בצייר ולפרסם אותה באלבום.
* פיצ'ר שני: מאפשר להפעיל את מצלמת האינטרנט של היוזר שהתחבר ולהעלות תמונה ממצלמת האינטרנט ל- timeline.
* פיצ'ר שלישי: (התווסף עקב שימוש בתבניות עיצוב Façade, Factory) מאפשר ליוזר שהתחבר להודיע על "מצב רגשי" שמח, עצוב, כועס ולעדכן זאת ב- facebook.

### תבנית מס' 1 – [Singleton]

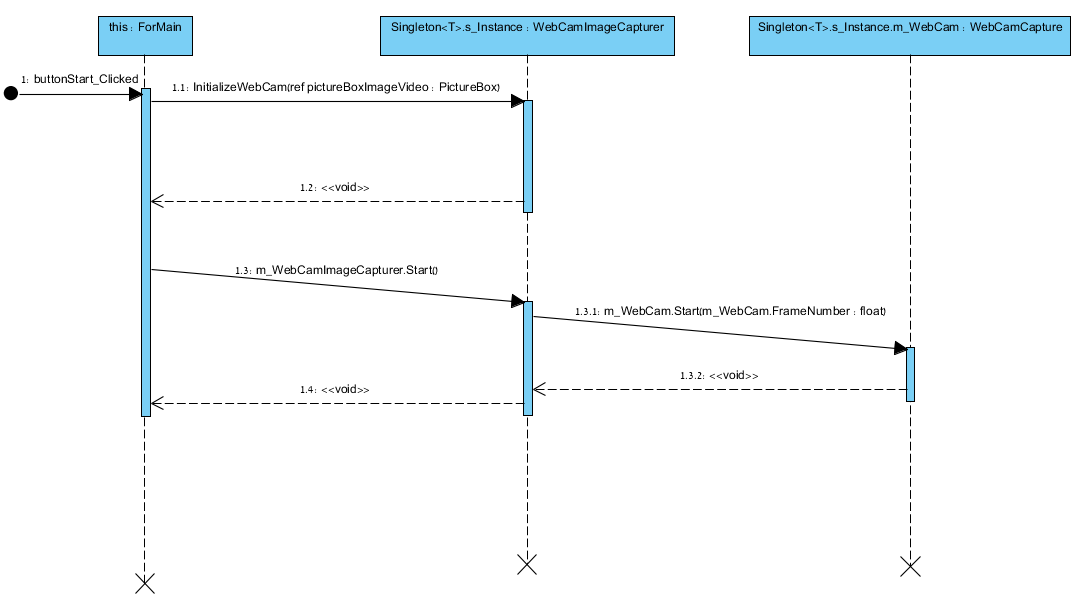
* סיבת הבחירה / שימוש בתבנית:

במערכת יש את מחלקת WebCamImageCapturer שאחראית על הפעלת המצלמה, לכידת תמונות, שמירתם, צילומי וידאו. לכן נרצה שבמהלך התוכנית יווצר מופע יחיד והוא יהיה נגיש למחלקות אשר יידרשו להשתמש במופע זה למשל FormMain. במילים אחרות קיימת גישה יזומה למופע היחיד לכל מי שמבקש להשתמש במערכת. יתר על כן היצירה מתבצעת On-Demand כלומר במתודה buttonStart\_Click כלומר רק כאשר מבקשים להפעיל את מצלמת האינטרנט (וכך איפשרנו מצב של 0 מופעים כלומר יצירה מיותרת של האובייקט במידה ולא רצינו להפעיל את מצלמת האינטרנט). בנוסף אין אפשרות ליצור מספר אובייקטים ממחלקה זו וכך אנו מונעים באגים פוטנציאליים במערכת.

* אופן המימוש:

המימוש של התבנית הוא באמצעות יצירת מחלקה גנרית שנקראת Singleton, בהתחשבות ב-Multi Threading בשיטת Double check Lock. במחלקת WebCamImageCapturer העברנו את ה- C'tor לפרטי והוספנו Propery סטטי וציבורי (public) בשם Instance שמחזיר את המופע היחיד.

* Sequence Diagram





* Class Diagram



### תבנית מס' 2 – [Facade]

* סיבת הבחירה / שימוש בתבנית:

בפיצ'ר השלישי ניתן להגדיר באיזה מצב רגשי המשתמש נמצא כרגע ולעדכן זאת בפייסבוק. את העדכון אנו עוטפים במספר פעולות כגון: פרסום תמונה עם ציטוט שמתאים למצב הרגשי, הפעלת שיר שתואם את מצב הרוח (שמח, עצוב, כועס). תבנית זו מפשטת למשתמש את התהליך בעדכון בפייסבוק, במקום לחפש שיר המתאים למצב רוח ולפרסם תמונה עם סמיילי וציטוט מתאים ניתן לעשות זאת בלחיצת כפתור 1.

* אופן המימוש:

הגדרנו מחלקה בשם EmotionsManager והיא מספקת את השירותים הבאים:

פתיחת שיר מתאים OpenEmotionalSong, פרסום תמונה וציטוט postEmotionPicture (שירותים אלו הם פרטיים) ופרסום מצב רגשי PostEmotionalState והוא ציבורי.

* Sequence Diagram

**Participants:**

Class FormMain is the **Client**

Class EmotionManager is the **Façade**

Classes EmotionalPictureCreator, User and MessageCreator are the **Subsystems**



* Class Diagram



### תבנית מס' 3 – [Factory]

* סיבת הבחירה / שימוש בתבנית:

הפיצ'ר השלישי מאפשר יצירה של 3 מצבים רגשיים, היצירה של האובייקט שמנהל את המצב הרגשי לא מתנהלת אצל הלקוח כלומר FormMain אלא היצירה מוטלת על המחלקה EmotionMangerFactory. בנוסף באמצעות המחלקה הנ"ל אין הכרות ציקלית בין המחלקה האבסטרקטית EmotionManager לבין היורשות שלה ולכן המערכת מודולרית.

* אופן המימוש:

הגדרנו enum שכולל את מצבי הרוח הקיימים במערכת ומחלקה סטטית בשם EmotionMangerFactory שמציעה את האפשרות של היצירה מתודה סטטית בשם Create שמקבלת את המצב הרגשי בו נמצא המשתמש ובהתאם אליו מחזירה את האובייקט שמנהל את המצב הרגשי הנ"ל.

* Sequence Diagram

**Participants:**

Class FormMain is the **Client**

Class EmotionManager is the **Product**

Classes AngryEmotionManager, HappyEmotionManager and SadEmotionManager are the **ConcreteProduct**

Class EmotionManagerFactory is the **Creator**



* Class Diagram



### עבודה אסינכרונית

אנחנו משתמשים ב Thread בכל הפקדים שמבצעים פעולה ארוכה כדי שהטופס יהיה שמיש למשתשמש.

באירוע לחיצה על כפתור ההתחברות Log In השימוש ב- Thread בעזרת מתודה אנונימית ב- fetchUserInfo בתוך ה- Thread יש שימוש ברכיב ה- UI - picture\_smallPictureBox, textBoxStatus, listBoxAlbumsNames לכן מבצעים שימוש ב- delegate שמקשר בין הפקדים ל- Thread.

באירוע לחיצה על הלינק שמציג את ה- News Feed השימוש ב- Thread בעזרת מתודה אנונימית ב- fetchNewsFeed בתוך ה- Thread יש שימוש ברכיב ה- UI - listBoxNewsFeed לכן מבצעים שימוש ב- delegate שמקשר בין הפקד ל- Thread.

באירוע לחיצה על הלינק שמציג את החברים השימוש ב- Thread בעזרת מתודה אנונימית ב- fetchFriends בתוך ה- Thread יש שימוש ברכיב ה- UI - listBoxFriends לכן מבצעים שימוש ב- delegate שמקשר בין הפקד ל- Thread.

באירוע לחיצה על הלינק שמציג את ה- Checkins השימוש ב- Thread בעזרת מתודה אנונימית ב- fetchCheckins בתוך ה- Thread יש שימוש ברכיב ה- UI - listBoxCheckins לכן מבצעים שימוש ב- delegate שמקשר בין הפקד ל- Thread.

באירוע לחיצה על כפתור שפותח Form של תמונות באלבום הנבחר השימוש ב- Thread בעזרת מתודה אנונימית ב- FetchPhotosFromAlbum בתוך ה- Thread יש שימוש ברכיב ה-UI - listBoxPictures לכן מבצעים שימוש ב- delegate שמקשר בין הפקד ל- Thread.

### שימוש ב- Data Binding

השימוש ב Data Binding נעשה בפיצ'ר 2, הגדרנו Object Data Source עבור מחלקת Photo.

כשה- FormAlbumPictures נפתח מופעלת מתודה FetchPhotosFromAlbum שמעדכנת שה- Data Source של רשימת התמונות הוא קולקציית התמונות באלבום שנבחר.