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

* **CheckIn Reccomandations**

פיצ'ר זה מאפשר למשתמש לקבל המלצות מהאינטרנט על סימוני הCheckIn שלו.

ניתן לחפש המלצות לפי ארבע קטגוריות: תמונות, וידאו, מזג אוויר, הצעות למסעדות.

לשם מימוש הפיצ'ר, יש בדף הבית של האפליקציה כפתור מעבר לפיצ'ר.

לאחר לחיצה על הכפתור – יוצג דף ייעודי ובו תוצג למשתמש רשימת הCheckIn שלו, כמו כן ישנם 4 כפתורי חיפוש לפי הקטגוריות.

המשתמש יבחר CheckIn עליו ירצה לקבל המלצות, כמו כן ייבחר קטגוריה. כעת יוצגו למשתמש תוצאות החיפוש בתוך הדפדפן של האפליקציה, לפי שם הCheckIn והקטגוריה שבחר.

המשתמש יכול להחליף קטגוריה ו/או את הCheckIn המסומן. ישנה אפשרות לחזור לתפריט הראשי.

* **FaceLiker**

פיצ'ר זה מאפשר למשתמש לקבל מידע וסטטיסטיקה על סימוני הלייקים שביצעו חבריו, לתמונות שלו.

לשם מימוש הפיצ'ר, יש בדף הבית של האפליקציה כפתור מעבר לפיצ'ר.

לאחר לחיצה על הכפתור – יוצג דף ייעודי ובו תוצג למשתמש רשימה של חבריו אשר ממויינת לפי כמות הלייקים שביצעו לתמונותיו של המשתמש.

המשתמש יבחר חבר מהרשימה אשר ירצה לקבל מידע וסטטיסטיקה אודות סימוני הלייק שביצע החבר.

המידע וסטטיסטיקה המוצגים למשתמש הם:

* שם החבר + תמונת הפרופיל שלו.
* סה"כ לייקים שבוצעו ע"י החבר.
* התמונה הראשונה עליה ביצע החבר לייק. (לפי תאריך).
* התמונה האחרונה עליה ביצע החבר לייק. לפי תאריך).
* התמונה עם הכי הרבה לייקים עליה ביצע החבר לייק. (סה"כ לייקים כולל משתמשים אחרים).
* התמונה עם הכי מעט לייקים עליה ביצע החבר לייק. (סה"כ לייקים כולל משתמשים אחרים).

המשתמש יכול לבצע פעולת ביטול חברות (Unfriend) עם החבר המסומן.

המשתמש יכול לשנות את החבר המסומן, עליו מקבל מידע. ישנה אפשרות לחזור לתפריט הראשי.

### תבנית מס' 1 – Visitor

* **סיבת הבחירה / שימוש בתבנית:**

הרגשנו שחווית המשתמש אינה מספקת, רצינו לתת תחושה שהאפליקציה "חיה", מגיבה לפעולות המשתמש ולכן בחרנו לממש תגובה אשר משנה את המצב הנוכחי של האובייקט. תגובה בדמות "הבהוב" כאשר המשתמש מקיש על פעולה מסוימת באפליקציה.

רצינו לממש תגובה זאת עבור לחיצה על label שמבצע פעולה וגם עבור לחיצה על Button שמעביר לטופס אחר.

בחרנו לממש את תבנית VISITOR ממספר סיבות.

על מנת למנוע שכפול קוד בין מחלקות שונות,שיפור תחזוקה עתידית של התגובה ובנוסף למנוע הכרות ציקלית בין המחלקה האחראית על הפעולה לבין המחלקה עליה מתבצעת הפעולה.

* **אופן המימוש:**

זיהינו כי המחלקה Control מהווה בסיס פולימורפי עבור המחלקות Label ,Button.

מימשנו מחלקה ControlBlinkera שתהווה VISITOR ובה מתודה Blink אשר מקבלת אובייקט מסוג Control ומשנה את מצבו הנוכחי ("מהבהב").

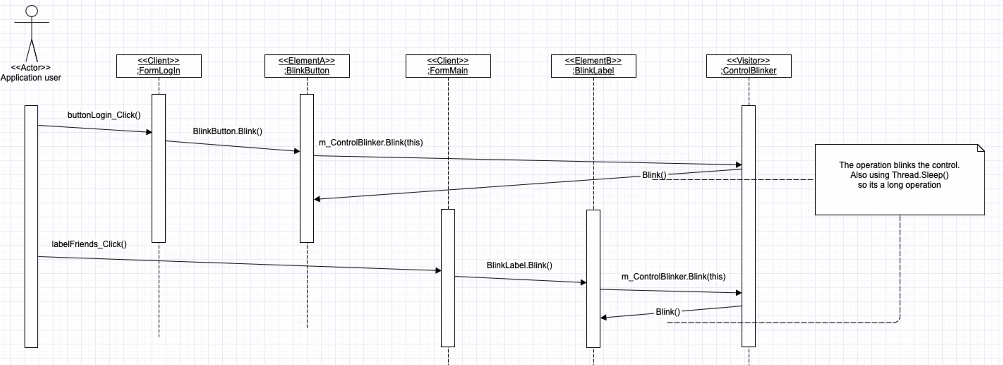
בנוסף מימשנו את המחלקות BlinkLabel, BlinkButton שמהוות אלמנטים ויורשות מהמחלקות Label, Button. כמו כן כל אחת מחזיקה בקומפוזיציה את הVISITOR.

בכל אחת מהאלמנטים מימשנו מתודה Blink אשר מפעילה את מתודת הBlink של הVISITOR שהיא מחזיקה בקומפוזיציה.

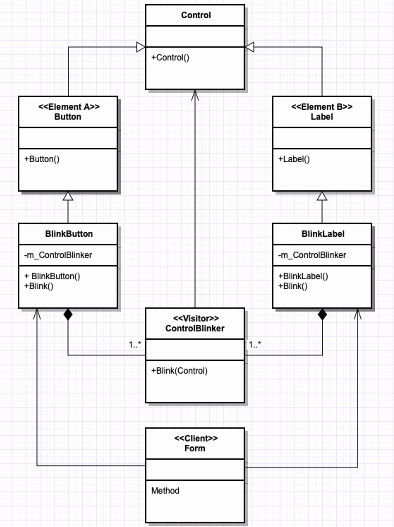
* **השחקנים בתבנית:**

1. ControlBlinker **:Visitor**
2. **:Element A** BlinkButton
3. **:Element B** BlinkLabel
4. **:Clinet** FormLogIn, FormMain

* Sequence Diagram



* Class Diagram



### תבנית מס' 2 – Observer

* **סיבת הבחירה / שימוש בתבנית:**

בחרנו ב Pattern Observer כי רצינו לממש מנגנון שבו Form Settings מודיע לשאר ה Forms במערכת שהמשתמש שינה את הצבע של האפליקציה או שינה את הFont של האפליקציה.

* **אופן המימוש:**

מימוש ה Notifier :

מימשנו FormSettings שבו יש שני Action אחד המקבל Color והשני מקבל Font ואז בכל

פעם שהמשתמש משנה את ה Color או את הFont, נפעיל Invoke ל Action המתאים.

מימוש הListeners :

כדי להימנע משכפול קוד בכל אחד מה Form שיש באפליקציה מימשנו מחלקה FormThatListen אשר יורשת מ Form ואז כל המחלקות שבעבר ירשו מ Form ירשו ממנה.

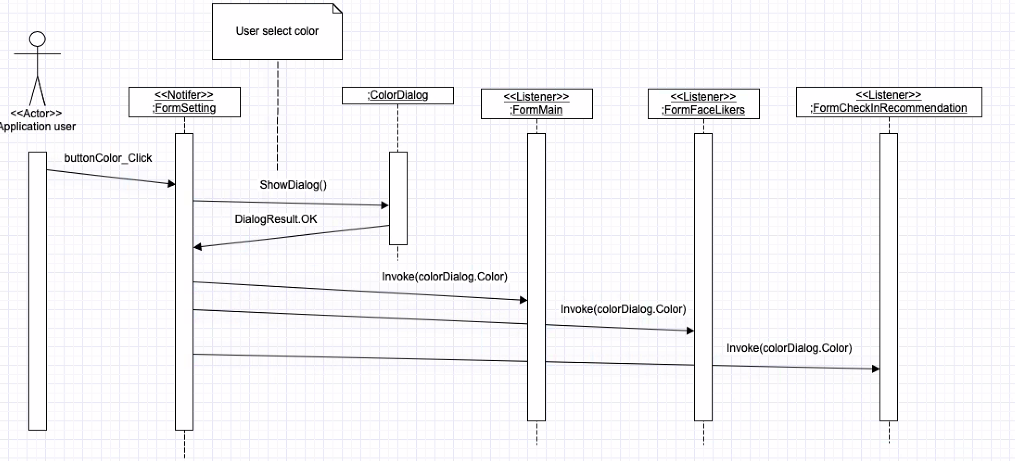
במחלקה FormThatListen מימשנו מתודה fontChanged אשר מעדכנת את הFont של המחלקה היורשת וגם את המתודה colorChanged אשר משנה את הצבע רקע של המחלקה היורשת.

* **השחקנים בתבנית:**

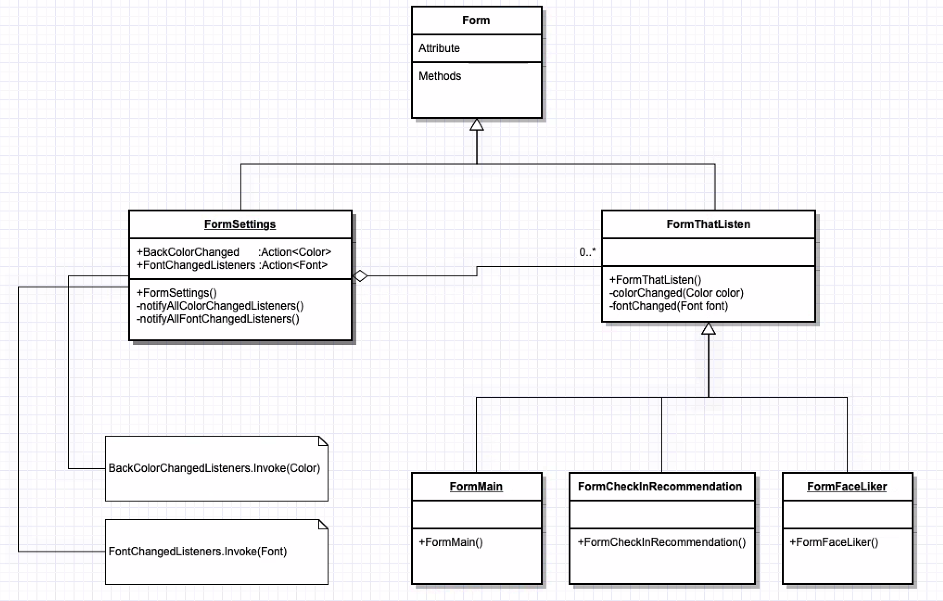
**Notifier**: FormSettings

:**Listeners** FormMain, FormFaceLikers , FormCheckInRecommendation

* Sequence Diagram



* Class Diagram



### תבנית מס' 3 – Strategy

* **סיבת הבחירה / שימוש בתבנית:**

זיהינו כי בפיצ'ר FaceLiker יש לנו מיון אוסף של אובייקטים מסוג LikerData שלו יש תכונות רבות. בחרנו למיין לפי התכונה totalLikes .

בחרנו להשתמש בתבנית Strategy על מנת לאפשר בעתיד למיין לפי שיטה אחרת או תכונה אחרת.

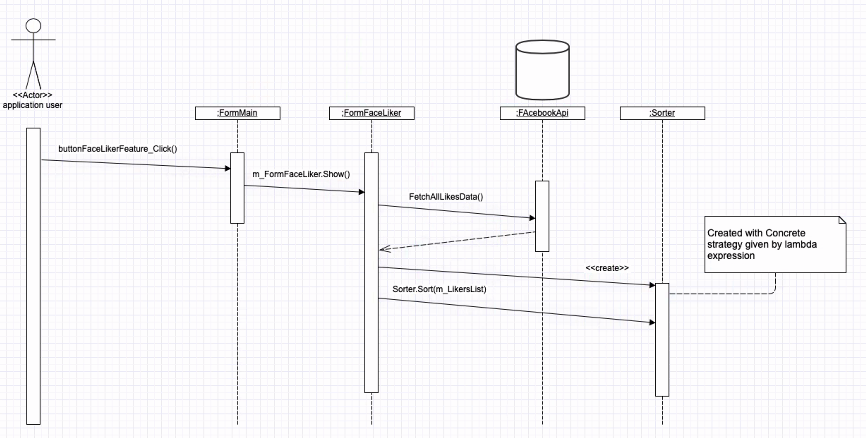
השימוש תבנית יחסוך לנו שכפול קוד, יקל וישפר את התחזוקה הדרושה וגם יקטין את הפוטנציאל לבאגים.

* **אופן המימוש:**

מימשנו מחלקה Sorter שבה יש אלגוריתם מיון שבתוכו יש "חור" – Injection point אשר מצפה לקבל את מתודת האסטרטגיה למיון, מתודה זאת צריכה לקבל שני אובייקטים מסוג LikerData ולהחזיר תשובה בוליאנית, לפי התשובה זאת המיון מתבצע.

במחלקה FormFaceLikers, במתודה sortLikesData, העברנו לSorter את אסטרטגיית המיון בעזרת Lambda expression, הפעלנו את הSorter ובכך מיינו מערך של אובייקטים מסוג LikerData לפי האסטרטגיה שרצינו.

* Sequence Diagram



* Class Diagram

