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

**1) פיצ'ר לאיתור מקומות פופולאריים/חברים לפי מקומות בילוי -**

**א)** מספק רשימת מקומות בילוי שבהם הכי הרבה חברים בילו בהם ועשו צ'ק-אין, ומראה אותם לפי סדר הביקוש הגובה ביותר. ניתן בנוסף לפתוח חיפוש של מקום הבילוי במנוע חיפוש לבחירה, באמצעות לחיצה על שם המקום.

**ב)** באמצעות סימון חבר, ניתן למצוא את מקומות הבילוי המשותפים עם חבר זה. גם כאן ניתן לפתוח חיפוש של מקום הבילוי במנוע חיפוש לבחירה, באמצעות לחיצה על שם המקום.

**ג)** מציאת החברים שמבלים באותם מקומות בילוי ומציג אותם לפי כמות המקומות המשותפים.

פיצר זה נמצא ב- PlacesForm .

**2) פיצ'ר לשליחת כרטיס ברכה למייל הפרטי של החברים -**

הפיצ'ר מאפשר לשלוח כרטיס ברכה לחבר אחד או יותר, באמצעות בחירת ברכה מותאמת אישית מהמבחר ברכות המופיעות באפליקציה, כגון: יום הולדת/יום נישואים/גיוס קל ועוד... בנוסף ניתן להוסיף הודעה אישית שתשלח יחד עם כרטיס הברכה למייל של החבר/ים.

* פיצר זה נמצא ב- GreetingForm .
* עבור כל אחד מהפיצרים הנ"ל ישנה תיבת היסטוריה של שימושים השונים בהן, הכוללת זמן הפעולה, סוג הפעולה ותיאור פרטים רלוונטים לגביה.
* ניתן לשמור לקובץ את כל היסטוריית הפעולות הנוכחית, ולבחור באיזה סוג קובץ לשמור את הנתונים. (xml או txt).

### תבנית מס' 1 – [Observer]

* סיבת הבחירה / שימוש בתבנית:

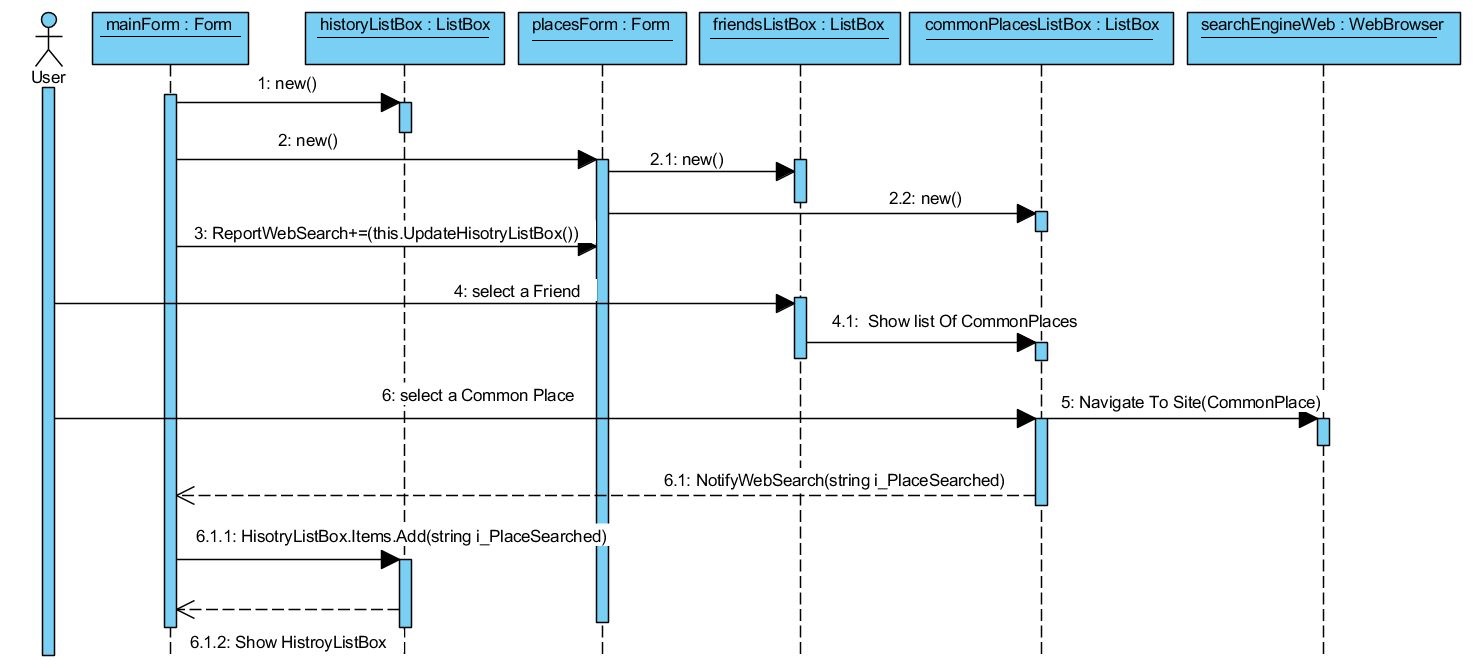
בחרנו בתבנית **Observer**מכיוון שרצינו לאפשר למשתמש לראות את היסטורית השימושים שלו בפעולות שונות שביצע ברכיבים שונים במערכת. על מנת שנרכז את הפעילויות המתרחשות בפיצ'רים השונים, כל אחד מהרכיבים (ה-**Notifiers**) יודיעו כאשר נעשה בהם שימוש ויעבירו את פרטי הפעולה הרלוונטים בהודעה זו. כך **המסך הראשי** (**MainForm**) שהוא ה**Listener**, יעודכן בשינויים ויחזיק רשימה של ההיסטוריה של פעולות המשתמש.

* אופן המימוש:

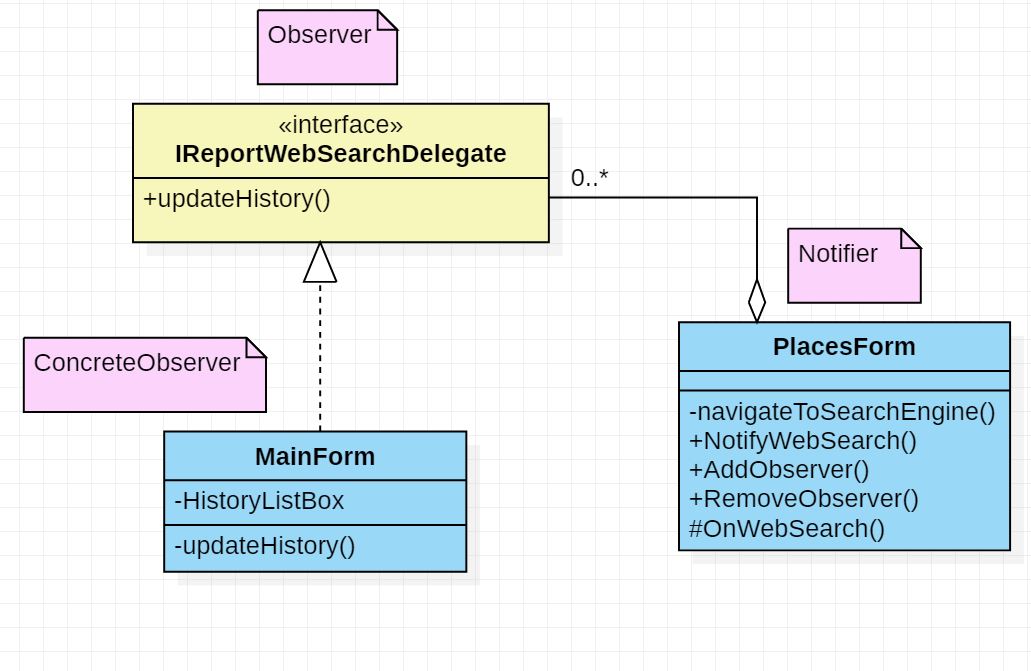
ה-**MainForm** (Concrete Oberver) מאזין ל-**PlacesForm** ול- **GreetingForm**(Notifiers) בכך שנרשם לאירוע שלהם (באמצעות דליגייט): **ReportWebSearch** ו- **ReportGreeingSent**, הם מעדכנים את המאזינים להם באמצעות מתודת **NotifyWebSearch** ו- **NotifyGreetingSent** בהתאמה, עם שליחת מחרוזת **Msg** בה פרטי הפעולה. הירשמות ה- **MainForm** נעשית עם מתודת ה-**updateHistory** שמעדכנת את רשימת ההיסטוריה של הפעולות – **historyListBox** עם ה- Msg שקיבלה.

.

* **Observer** Sequence Diagram



* **Observer** Class Diagram



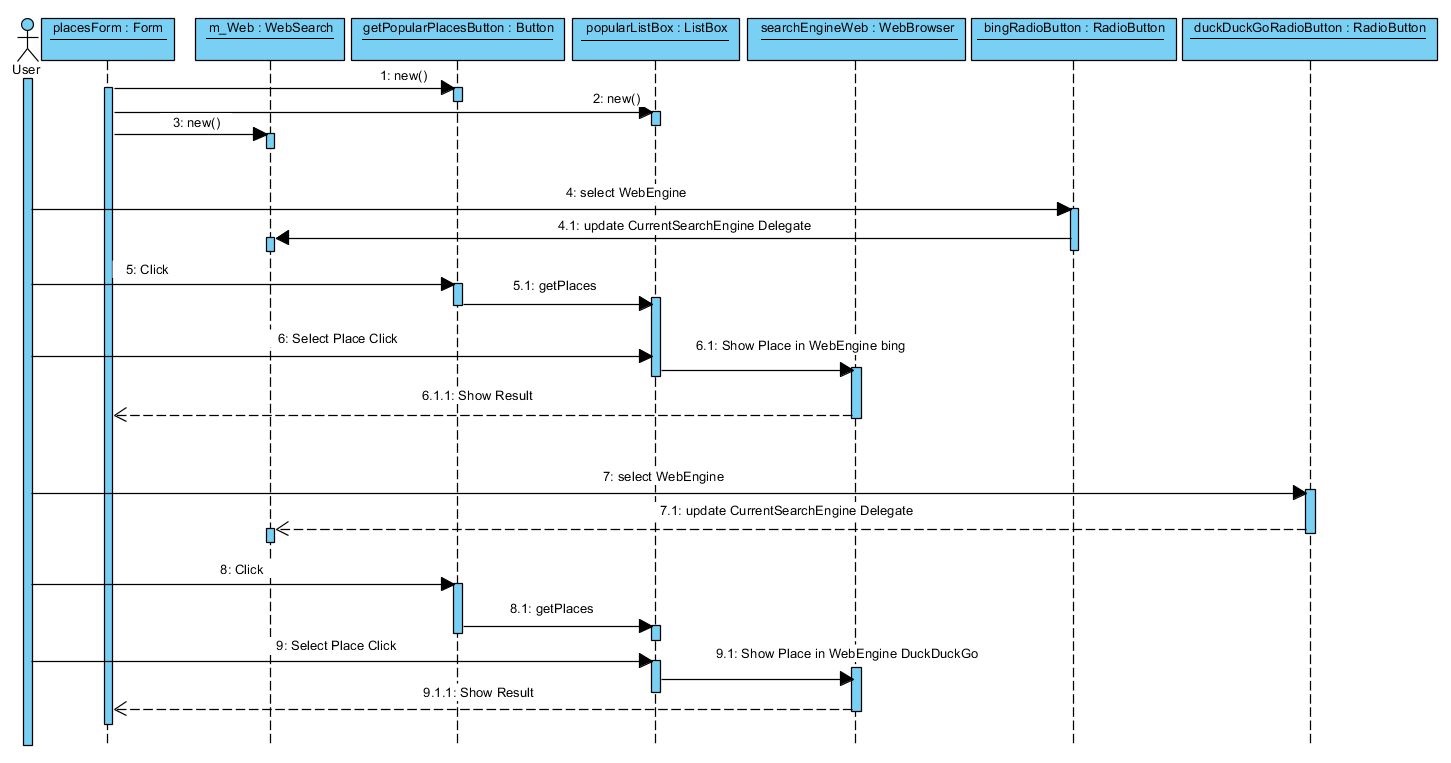
### תבנית מס' 2 – [Strategy]

* סיבת הבחירה / שימוש בתבנית:

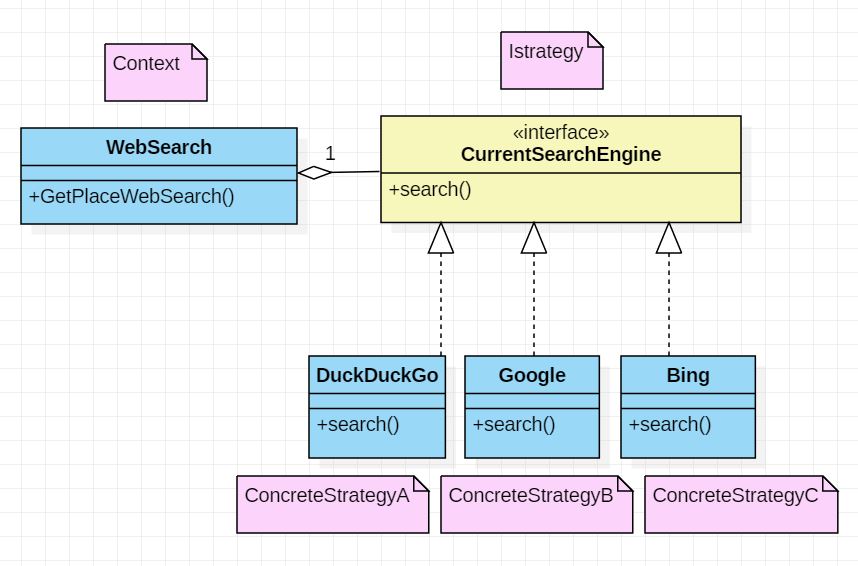
בחרנו בתבנית **Strategy**מכיוון שבפיצ'ר המקומות כאשר המשתמש רוצה לבצע חיפוש לקבלת מידע של מקום, שקיבל באחת מפונקציות המערכת, הוא יכול לבחור את סוג מנוע החיפוש(Concrete strategy) בו הוא מחפש את המקומות שמצא. כך בזמן פעילות האפליקציה, המשתמש יכול לבחור מנועי חיפוש שונים(Google, Bing, DuckDuckGo), ובאמצעות מימוש הפעלת מנגנונים אלו בנפרד יצרנו מבנה פריק, מודולרי **Replacabilly** **וMaintainabiy**. כך שגם בעתיד נוכל לשנות למנגנוני חיפוש נוספים.

* אופן המימוש:

בפיצ'ר המקומות ב- **PlacesForm** ישנו הרכיב WebSearch המחזיק **delegate** למנוע החיפוש הנוכחי **CurrentSearchEngine** (Istrategy), ואחראי לבניית החיפוש המתאים באינטרנט עבור אותו מנוע תמורת מקום שבחר המשתמש וזאת באמצעות המתודה: GetPlaceWebSearch(). המשתמש בוחר את מנוע החיפוש בו יבצע את חיפוש המקומות, באמצעות לחיצה על כפתור רדיו מתאים הנמצא ב - **PlacesForm**. בחירת כפתור הרדיו מעדכנת את מנוע החיפוש הנוכחי באמצעות **lamda expression** למנוע החיפוש הנוכחי(Concrete strategy), שמגיע באופן דיפולטיבי למנוע Google. בעת בקשת החיפוש המערכת משתמשת במצביע הנוכחי למתודת המנוע הרלוונטית.



* **Strategy** Sequence Diagram
* **Strategy** Class Diagram



### תבנית מס' 3 – [Template method]

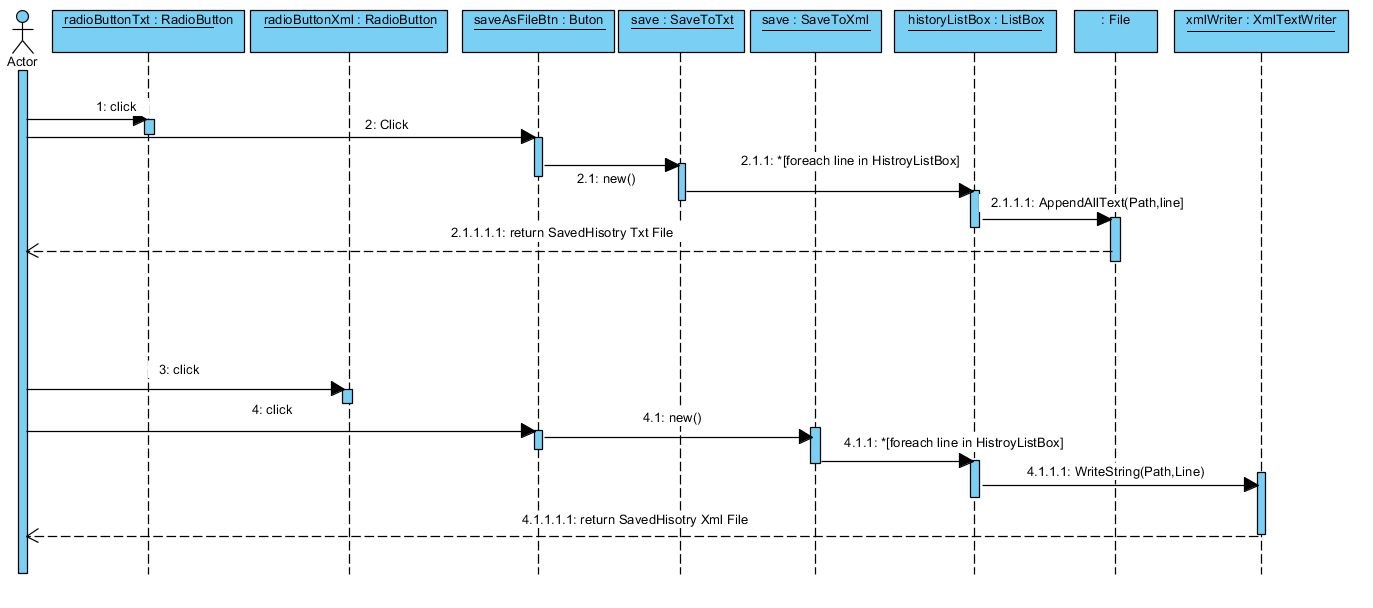
* סיבת הבחירה / שימוש בתבנית:

בחרנו להשתמש בתבנית Template Method מכיוון שכאשר המשתמש שומר למחשב את תיבת ההיסטורייה הנוכחית, הוא יכול לבחור האם להשתמש בשמירה לקובץ xml או לקובץ txt. הרכיב האבסטרקטי **SaveToBase** הוא הבסיס לפעולת השמירה, כלומר בו מתבצעת פתיחת דיאלוג לבחירת מקום השמירה במחשב וכן גם עדכון פרטים הרלוונטים לשמירה. ממנו דאגנו להוציא למתודה נפרדת שהיא **protected abstract**, את פעולת השמירה עצמה של הקובץ (**injection point**). כך ניתן לבחור את סוג הקובץ אליו שומרים, באמצעות מימוש שונה של המתודה האבסטרקטית במחלקות הממשות של השמירה. כך בנינו את מערכת השמירה שלנו יותר replacabily ומודולרית.

* אופן המימוש:

יצרנו רכיב אבסטרקטי **SaveToBase** שהוא הבסיס לפעולת השמירה. לרכיב השמירה יש מתודה – saveHistory()אשר מקבלת כפרמטר listBox שמכיל את היסטוריית השימושים של המשתמש באפליקציית הפייסבוק שלנו, ומהתוכן שלו מבצעת את השמירה לקובץ. בנוסף במתודה זו יש (**injection point**) כלומר שימוש במתודה נוספת: **saveAs()** שהיא מתודה **protected abstract**, מה שאומר שאין לה מימוש ברכיב האב (**SaveToBase**), אך על היורשים ממנו לממש אותה. הרכיבים SaveToXml ו- SaveToTxt הם הבנים אשר יורשים מ- **SaveToBase**ומממשים את המתודה **saveAs()**, כל אחד בצורה שונה לפי סוג הקובץ שלו. SaveToTxt מבצע שמירה באמצעות שימוש במחלקה הסטטית File, ו- SaveToXml מבצע את השמירה באמצעות המחלקה XmlTextWriter.

* **Template Method** Sequence Diagram



* **Template Method** Class Diagram

