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

* :Events Planner

בשביל להגיע לפיצ'ר הזה: צריך לעשות Login וללחוץ על לשונית "מתכנן אירועים".

בלוח שנה מודגשים תאריכים אשר בהם יש יום הולדת או אירוע אחר של המשתמש.

לחיצה על אחד התאריכים תציג למשתמש את רשימת האירועים באותו יום.

לחיצה על אירוע ספציפי תציג למשתמש מפה עם מיקום האירוע.

הפיצ'ר הזה נועד לעזור למשתמש להיות בעיניינים, להיות מעודכן על האירועים הקרובים אליו ומעניק למשתמש אפשרות לנהל את הזמן בצורה קלה ונכונה יותר.

את הפיצ'ר ניתן למצוא בקוד תחת:

FacebookApp.Logic במחלקה EventPlanner

FacebookApp.UI במחלקה AppForm

* Wish Happy Birthday To Your Real Friends:

בשביל להגיע לפיצ'ר הזה: צריך לעשות Login וללחוץ על לשונית "ימי הולדת".

לחיצה על כפתור "לחפש חברים" תציג את רשימת החברים שגם יש להם יום הולדת בחודש הזה והם גם פרסמו פוסט על הקיר של המשתמש ביום ההולדת שלו.

לחיצה על הכפתור "לאחל מזל טוב ליום הולדת" מאפשרת לפרסם פוסט על הקיר של החבר הנבחר לכבוד יום הולדתו.

הפיצר עוזר למשתמש למצוא את חבריו האמיתיים שכותבים לו איחולים ביום הולדתו ולעזור לו להיות מוכן לקראת יום הולדת של חבריו.

את הפיצ'ר ניתן למצוא בקוד תחת:

FacebookApp.Logic במחלקה RealFriendsListFinder

FacebookApp.UI במחלקה AppForm

### תבנית מס' 1 – Adapter Pattern

* סיבת הבחירה / שימוש בתבנית:

כאשר מימשנו את הפיצ'רים הנ"ל, עלה לנו הצורך לקבל את התאריכים בפורמט DateTime. ומכך ש- FacebookAPI מחזיר את התאריכים בפורמט String ואנחנו היינו צריכים להסתכל על היום והחודש של התאריך, נדרשנו לבצע את ההמרה מהטיפוס String ל- DataTime . עשינו זאת באמצעות התבנית Adapter.

* אופן המימוש:

יצרנו interface בשם IConvertableBirthday אשר מצהיר על מתודה ConvertBirthdayToDate שממירה תאריך מ-string ל-DateTime, ומחזירה את התאריך בפורמט החדש.

בנוסף יצרנו מחלקה בשם BirthdayDateAdapterאשר מממשת את הממשק הנ"ל ומחזיקה אובייקט מטיפוס User, שמייצג את החבר שאת תאריך יום הולדתו נרצה להמיר לפורמט DataTime.

שימוש בתבנית ניתן לראות במחלקה EventsPlanner ,במתודה BirthdayStringToDateTime שבה למעשה אנחנו משתמשים ב- Adapter כדי לקבל את המידע בפורמט שנחוץ לנו.

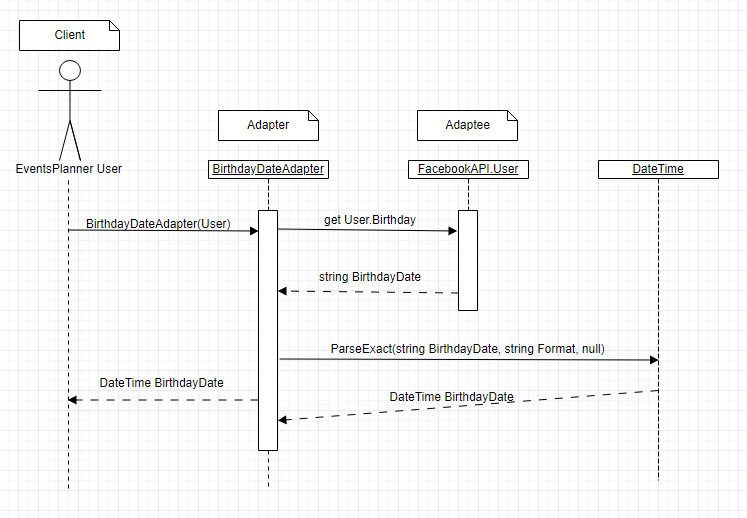
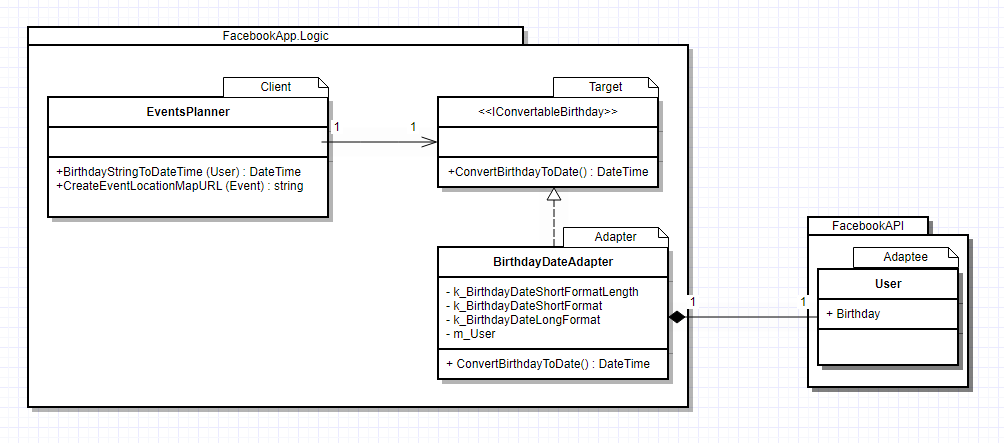
ה"שחקנים" בתבנית:

• Client: Class EventsPlanner

• Adapter: BirthdayDateAdapter

• Adaptee: Class User

• Target: Interface IConvertableBirthday

* Sequence Diagram
* Class Diagram

### תבנית מס' 2 –Singleton Pattern

* סיבת הבחירה / שימוש בתבנית:

בתוכנית שלנו הייתה האפשרות ליצור מספר מופעים של המחלקה AppForm, דבר שלא היה הגיוני.

לכן, בחרנו לממש תבנית זו כדי שיהיה ניתן ליצור מופע אחד ויחיד של המחלקה AppForm.

* אופן המימוש:

יצרנו במחלקה AppForm שדה סטטי s\_AppFormInstance מסוג AppForm שיהיה המופע היחידי של המחלקה.

הוספנו מתודה Create ליצירת המופע היחיד.

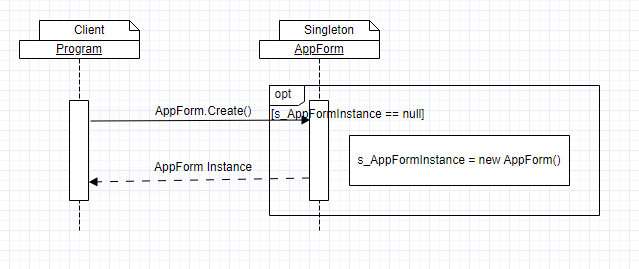
הוספנו מתודה AppFormInstance לשימוש במופע היחיד.

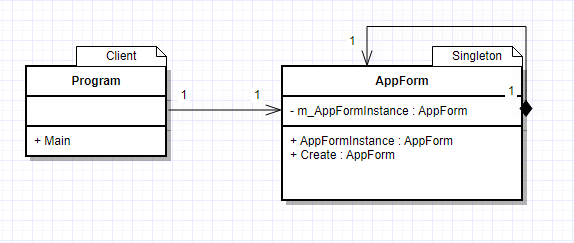
"השחקנים" בתבנית:

• Client: Program

• Singleton: AppForm

* Sequence Diagram



* Class Diagram

### תבנית מס' 3 –Façade Pattern

* סיבת הבחירה / שימוש בתבנית:

בחרנו לממש תבנית זו כדי לאפשר ניהול יותר נוח של הפיצ'רים שלנו.

תבנית זו גם מאפשרת הוספת פיצ'רים חדשים באופן יעיל ותחת אותה קורת גג.

* אופן המימוש:

הוספנו מחלקה בשם AppLogic ובה שני שדות עיקריים כנגד שני הפיצ'רים: m\_EventsPlanner ו- m\_RealFriendsListFinder.

וכן מתודות המאפשרות שימוש מלא בפיצ'רים בקריאה פשוטה אחת.

"השחקנים" בתבנית:

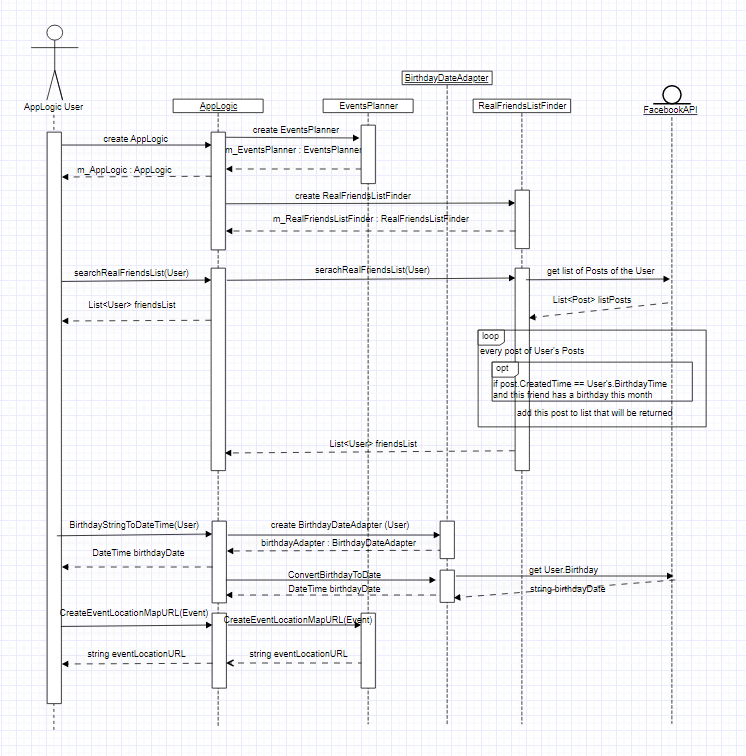
• Client: AppForm

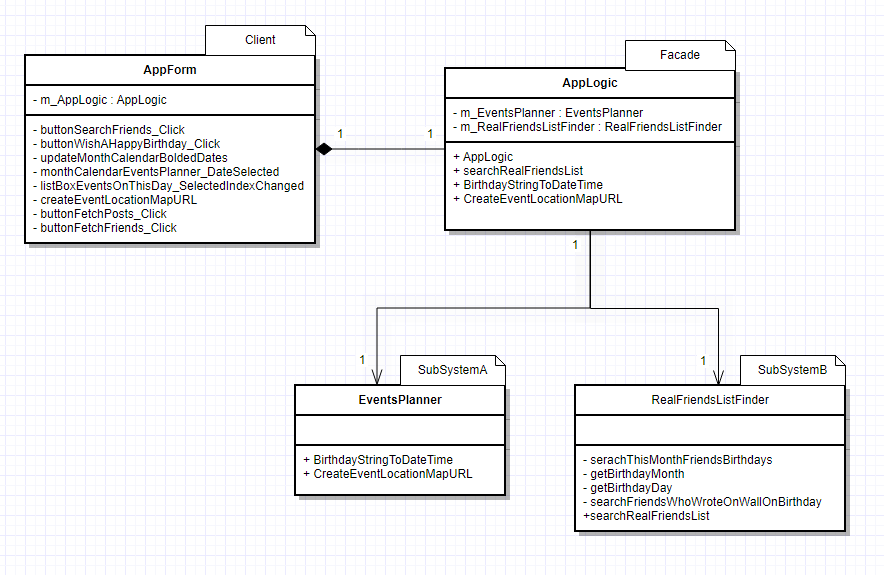
• Façade: AppLogic

• SubSystemA: EventsPlanner

• SubSystemB: RealFriendsListFinder

* Sequence Diagram



* Class Diagram

**שימוש בעבודה אי-סינכרונית:**

בחרנו להשתמש בתכנות א-סינכרוני עבור פעולות שעובדות מול שרת הפייסבוק בטופס הראשי, מכיוון שברגע שפעולה אחת מופעלת ושולחת בקשה לקבלת מידע משרת הפייסבוק, לא ניתן לבצע פעולות נוספות עד שהפעולה הקודמת מסתיימת לגמרי.

ביצענו את המימוש של העבודה הא-סינכרונית על ידי הפעלת המתודות האחראיות על הבאת נתונים משרת הפייסבוק ב-Threads נפרדים.

כאשר משתמש מתחבר לאפליקציה נטענים למסך הראשי הפוסטים שלו וכן מידע כללי נוסף.

הסיבה שבחרנו להשתמש בעבודה אי-סינכרונית עבור הטעינה של המידע היא שהעבודה האי-סינכרונית מאפשרת שטעינת פוסטים לא תקפיא הטענת שאר המידע.

המתודות שמתבצעות ב-threads הנפרדים:

• fetchUserInfo

• fetchPosts

מתודות אלו נמצאות במחלקה AppForm.

יצרנו מתודה בשם fetchUserInfoAndPosts אשר בה מייצרים את ה-threads הנ"ל ומאתחלים אותם וטוענים את כלל המידע למסך הראשי.

בנוסף, הוספנו גם מנעולים לthreads אלו במתודות Create, fetchUserInfo, fetchPosts.

**עבודה עם Data Binding:**

השתמשנו בData Binding בצורת One-Way בפיצ'ר EventsPlanner:

כאשר משתמש בוחר בתאריך בלוח שנה, מוצגים ב-listBoxEvents כל ה-Events של תאריך זה. ובלחיצה על event מוצגת מפה ובה המיקום של ה-event הנבחר.

מימשנו זאת במתודות monthCalendarEventsPlanner\_DateSelected ו-listBoxEventsOnThisDay\_SelectedIndexChanged.