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

* המשתמש בוחר תקופת זמן כלשהי והאפליקציה מציגה לו רשימת ימי הולדת של חבריו בזמן שבחר ונותנת לו אפשרות לייצא לקובץ של לוח שנה את ימי ההולדת עם ברכה מוכנה לפי שם וגיל.
* הצעת תגובה מותאמת אישית לפוסט "יום הולדת שמח" על קיר המשתמש ביום הולדתו מטרת - הפיצ'ר היא להקל על המשתמש בלהגיב לכל הפוסטים שהוא מקבל ביום הולדתו, ע"י מתן תגובה אישית שנוצרת ע"י מחולל תגובות ומותאמת לכותב פוסט יום ההולדת (עפ"י גיל, מגדר ושם כותב הפוסט).

**שימוש ב Data Binding –**

בקוד נעשה שימוש בData Binding בקלאס FacebookAppLogicFacade שמציג את פרטי חבר הפייסבוק שנבחר בFormMain.

**שימוש ב- MultiThreading**

בקוד נעשה שימוש בMultiThreading בקלאס FormGenerateComment.

בקלאס FormGenerateComment אנו יוצרים ת׳רד נפרד לקבלת כל הפוסטים של המשתמש, מהיות וזוהי בקשה כבדה שלוקחת זמן, יצירת הת׳רד זה מאפשרת לשאר החלונות להישאר פעילים עד קבלת הפוסטים (דבר שלפני הת׳רד לא היה אפשרי).

### תבנית מס' 1 – Singleton

* סיבת הבחירה / שימוש בתבנית:

בחרנו להשתמש בתבנית סינגלטון בקלאס FormMain אשר משמש כ-״יקום״ של האפליקציה שלנו. במימוש של FormMain אנו שומרים כשדה את היוזר המחובר. בישביל להמנע מהעברת היוזר לכל פיצ׳ר בכל פעם שמוסיפים פיצ׳ר לאפליקציה, ניתן יהיה להוסיף את הפיצ׳ר כרגיל ומתוכו לגשת לסינגלטון ולקחת את היוזר המחובר.

לא רצינו לאפשר לייצר עוד FormMain מהיות ורק יוזר אחד יכול להיות מחובר בו זמנית ולכן הסינגלטון פותר לנו גם את זה.

* אופן המימוש:

המימוש נמצא ב FormMain עם Private Constractorו- Double check lock

* Sequence Diagram

Diagram

Description automatically generated

* Class Diagram

Diagram

Description automatically generated

### תבנית מס' 2 – Façade

* סיבת הבחירה / שימוש בתבנית:

בחרנו לממש קלאס בשם FacebookAppLogicFacade אשר ישמש כמעין "שלט" ויקשר בין MainForm לבין כל הפעולות שהוא רוצה לבצע בטופס מבלי לגשת לקלאסים נוספים ובכך לאגד את כל הפעולות לקלאס אחד.

* אופן המימוש:

הקלאס FacebookAppLogicFacade מייצג Façade שבו קיימים מופעים לקלאסים FindAndPrepareBirthdaysFeature ו- GenerateCommentFeature.

כך יוכלו הטפסים להשתמש בפונקציונליות שלהם מבלי לגשת לכל קלאס בנפרד וכך הטפסים לא צריכים להכיר את הקלאסים של הלוגיקה שלהם.

* Sequence Diagram

Diagram

Description automatically generated

* Class Diagram

Diagram

Description automatically generated

### תבנית מס' 3 –Factory Method

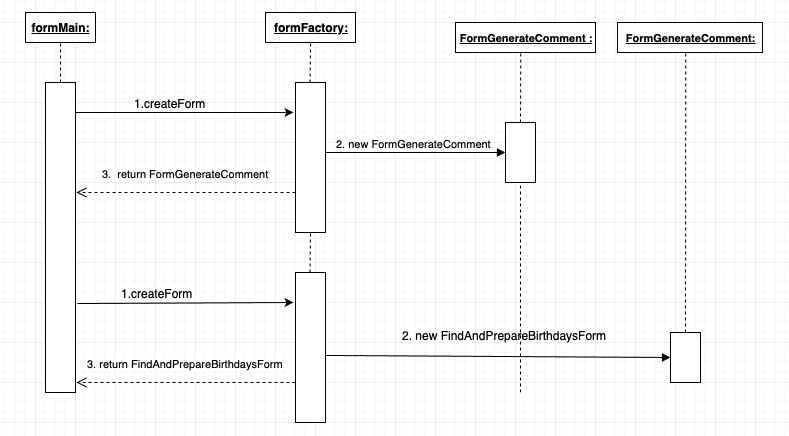
* סיבת הבחירה / שימוש בתבנית:

1. מתן אפשרות לייצור forms חדשים נוספים מסוגים שונים והבאתם למשתמש, מבלי שיצטרך להתעסק בתהליך היצירה והלוגיקה שמאחורי יצירתם.
2. אקטנסיביליות – כשנרצה בעתיד להוסיף forms נוספים לא נצטרך לבקש אותם דרך מחלקה נפרדת, אלא דרך אותה מתודה במחלקה הסטטית שיצרנו.

* אופן המימוש:

1. המימוש במחלקה הסטטית FormFactory.
2. המחלקה מחזיקה מתודה סטטית שנקראת createForm שמקבלת סטרינג – השם של הform שאנו רוצים ליצור ויוצרת form בהתאם.
3. ההגיון שעומד מאחורי המימוש – נרצה כמה סוגי forms באפליקציה (לדוג' בעתיד אולי נרצה לממש עוד פיצרים בforms חדשים וכו') ועקרון חשוב - מימוש יצירת האובייקטים בצד השרת תחת קורת גג אחת.

* Sequence Diagram



* Class Diagram

