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

* **Covid-19 Post :**

FormCorona– בעקבות וירוס הקורונה, הוספנו אפשרות של העלאת פוסט בו התוכן קבוע עם האשטאג #CoronaPost, פיצ'ר זה מאפשר למשתמש אשר חלה בקורונה לעדכן את כל החברים שהוא היה בקרבתם בתקופת השבועיים האחרונים מהיום בו הוא השתמש בפיצ'ר.

הפיצ'ר בודק את כל האנשים שהיו בקרבת היוזר בשבועיים האחרונים על פי תיוגים בפוסטים, תמונות וצ'ק אין שהתבצעו בתקופת השבועיים הללו.

**היכן למצוא** :בחלון הראשי שנפתח (FormFacebook) לאחר שמבצעים התחברות, ניתן לראות בשורה השלישית אייקון שרשום עליו 'Covid-19 Post ', לאחר לחיצה על אייקון זה, יפתח חלון המכיל את התוכן של הפוסט ואפשרות לפרסום.

- פיצ'ר זה יושב תחת המחלקה FormCorona בפרויקט ועושה שימוש במחלקות של FacebookWrraper כגון Post, Checkin, Friends וכו'.

* **Matching By Zodiac:**

FormMatch – (שינינו את הפיצ'ר מההגשה הראשונה בעקבות ביקורת הבודק).

פיצ'ר זה מציג למשתמש את כל החברים שלו שמתאימים לו על פי המזל שלו.

הפיצ'ר מסנן את החברים המתאימים לפי מזל ולפי העדפות המשתמש: גברים, נשים או שניהם. פיצ'ר זה ממומש במחלקה FormMatch ועושה שימוש במחלקות : PanelToUser, PanelWomen, PanelMen, PanelBsexual כאשר הפנלים מממשים תצוגת מידע עבור יוזר בודד בחברים של המשתמש, השוני בין הפנלים הוא בעיצוב שלהם כל אחד מעוצב בצבעים שונים כך שכל בחירה של המשתמש תקבל תצוגה ייחודית.

תוספים שהפיצ'ר משתמש בהם:

* הפיצ'ר משתמש במחלקה ProxyFriends על מנת לייעל את העבודה מול השרת של פייסבוק ולהשתמש במאגר זמני של רשימת חברים, שנוצר במערכת שלנו.
* הפיצ'ר עושה שימוש במחלקה FacadeMatch שאחראי על הבאת רשימת החברים שמתאימים ע"פ מזל ומין למשתמש מרשימת החברים שלו.

**היכן למצוא** :בחלון הראשי שנפתח (FormFacebook) לאחר שמבצעים התחברות, ניתן לראות בשורה השלישית אייקון שרשום עליו 'Find Match ', לאחר לחיצה על אייקון זה, יפתח חלון השואל את המשתמש מה ההעדפה המינית שלו.

- פיצ'ר זה יושב תחת המחלקה FormMatch בפרויקט ועושה שימוש במחלקה של FacebookWrraper :User.Friends

### עבודה אסינכרונית - Threads:

רוב המתודות בפרוייקט מקבלות מידע משרת פייסבוק , הבאת המידע ועיבודו מכבידה על הפרוייקט ויוצרת פרקי זמן ארוכים לטעינת מתודה \ העלאת חלון ,

על כן השתמשנו בעבודה אסינכרונית במקרים בהם היה צורך לקיצור זמני ההמתנה למידע. שימוש ביכולת של השרת לקבל מספר פניות במקביל, ובנוסף שימוש בטרדים במקביל אפשר למשתמש את היכולת לבצע כמה תהליכונים, כך שהמסך לא נתקע, והמשתמש יכול לזיז את חלון התצוגה בזמן קבלת הנתונים, או לבצע פעולות נוספות בזמן שמתבצעת גישה לשרת הפייסבוק.

ניתן לראות שימוש בעבודה אסינכרונית תחת תת פרוייקט UI , במקומות הבאים :

* + בכל אחד מהForm הבאים כאשר מבצעים fetch למידע משרת הפייסבוק : Album ,Birthday, Checkin, Corona, Posts (Multi-Thread with UI)
  + בנוסף בחלון הראשי – FormFacebook כאשר יוצרים את כל החלונות שמוצגים כאייקונים.

### עבודה עם Data Binding:

ניתן לראות את המימוש עבור - Data binding Two-Way תחת פרוייקט UI , במחלקות הבאות :

* FormEvents
* FormGroups
* FormFriends

על מנת לבצע את הקישור בין הנתונים לבין ה-UI, עבור כל חלון מהנ"ל יצרנו פאנל שבו יושב ה-Binding Source, וקישרנו אותו לרשימה בטופס, נפרט את ה-Data שהוספנו אל כל חלון :

1. FormEvents : (eventBindingSource)

עבור כל אירוע שנמצא ברשימת האירועים של היוזר ניתן לראות כאשר נלחץ עליו:

* + תמונת Cover של האירוע.
  + שם האירוע.
  + מס' משתתפים.
  + מס' מעוניינים להשתתף.
  + תיאור האירוע.
  + עיר שבה האירוע מתקיים.
  + לינק לאירוע.
  + תאריך התחלה וסיום.

1. FormGroups: (groupsBindingSource)

עבור כל קבוצה שהיוזר שייך אלייה מרשימת הקבוצות, כאשר נלחץ עלייה נראה:

* + תמונת הקבוצה.
  + שם הקבוצה.
  + תיאור על הקבוצה.
  + לינק לעמוד הפייסבוק.
  + תאריך אחרון שבה הקבוצה התעדכנה.

1. :FormFriends (friendsBindingSource )

עבור כל חבר ברשימת החברים של היוזר ניתן לראות כאשר נלחץ עליו:

* + תמונת פרופיל.
  + תאריך יום הולדת.
  + לינק לפייסבוק.
  + שם החבר.

### תבנית מס' 1 – [Pattern Builder]

**סיבת הבחירה / שימוש בתבנית:**

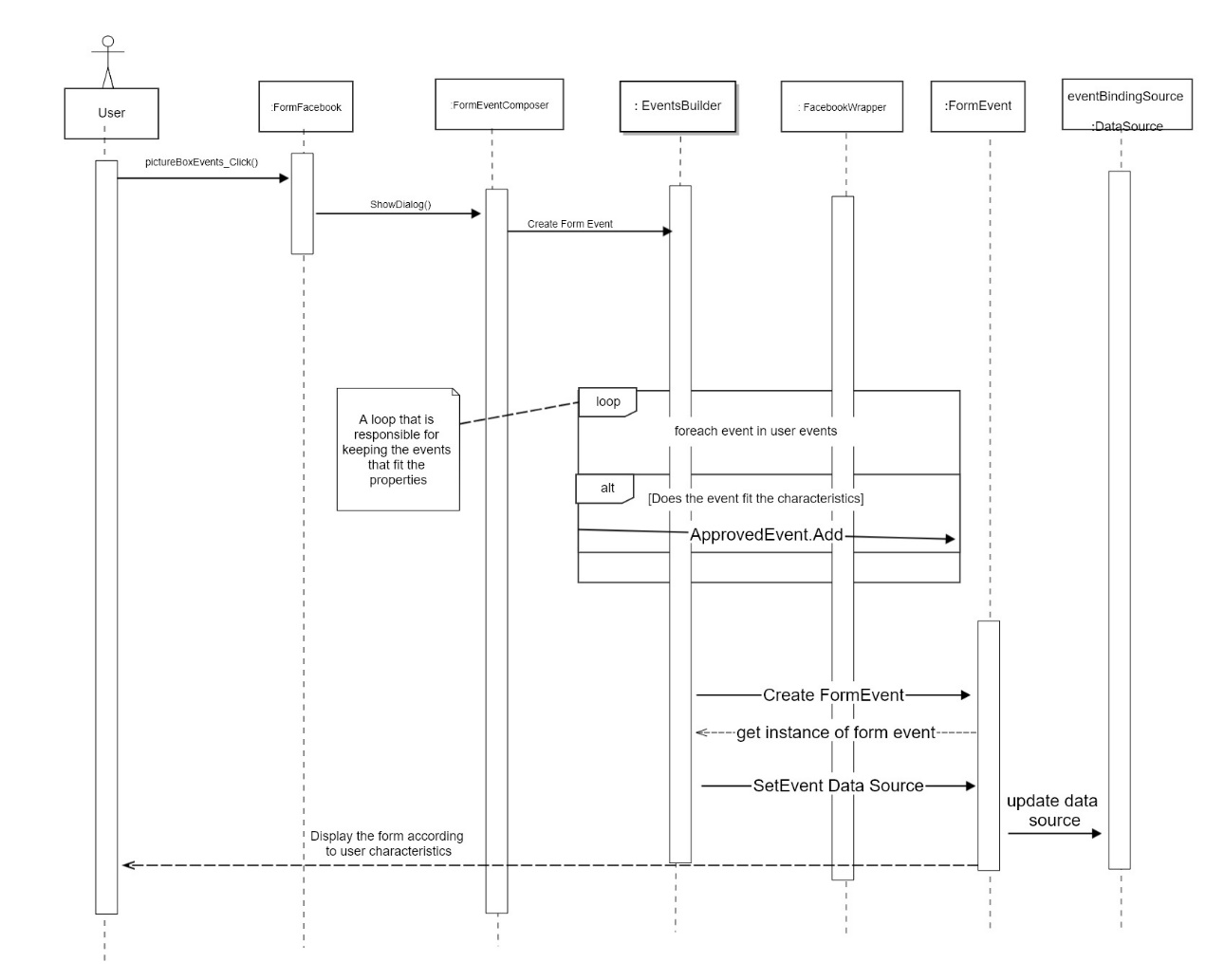
* הצורך בתבנית: החלטנו לממש את התבנית בטופס "אירועים" מהסיבה שאנו חיים בתקופת הקורונה, לאנשים חשוב להיות באירועים שלא מכילים מס' רב של משתתפים, בנוסף ניתנת למשתמש האפשרות לסנן אירועים שאינם רלוונטיים עבורו ע"פ תאריך.

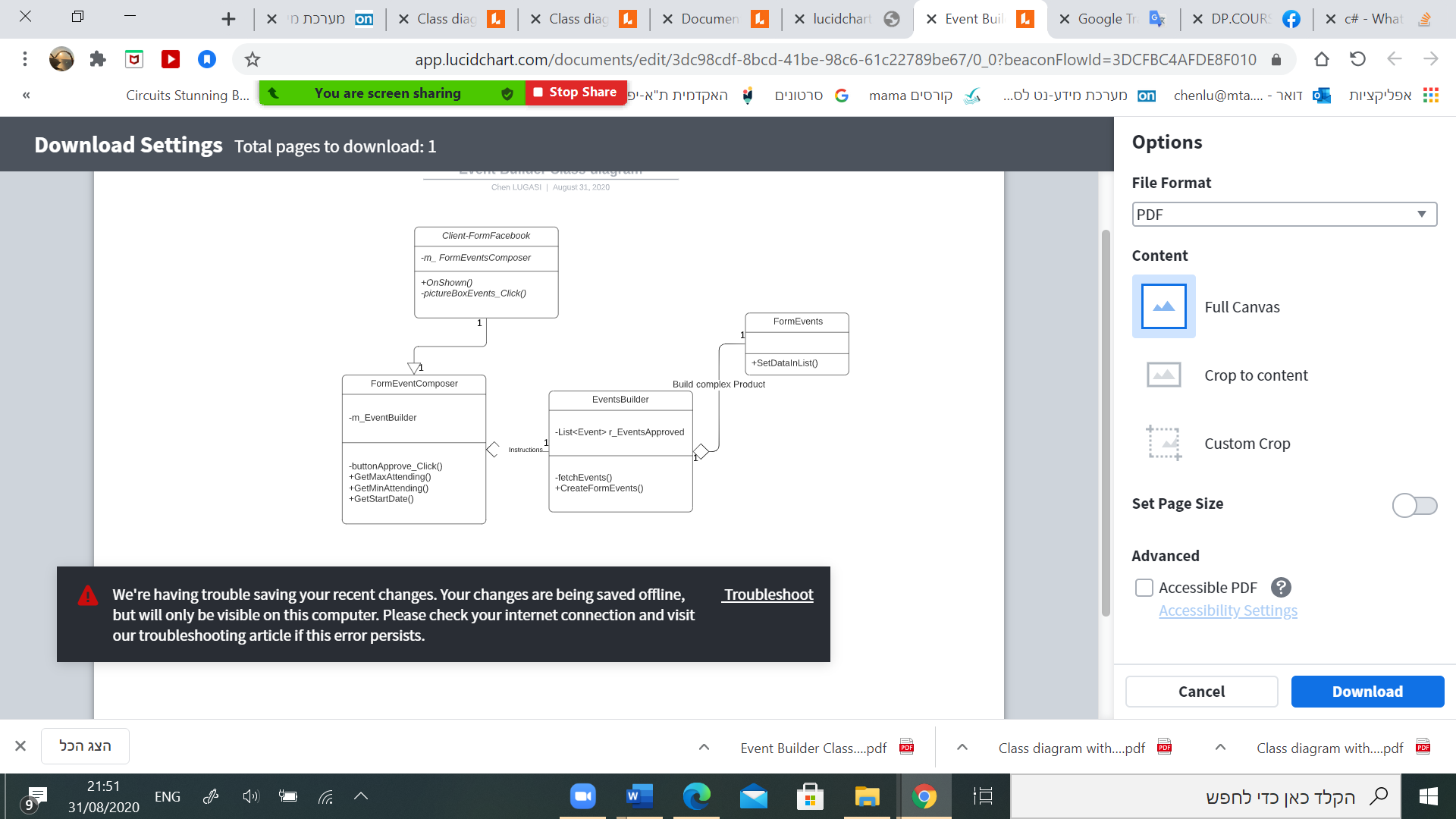
**אופן המימוש:**

* ה**Client -** הוא משתמש הפייסבוק היוזר, הוא מיוצג ע"י בחירה של אירועים בטופס הראשי – FormFacebook ,
* ה-**Composer** שמקבל את הנתונים והדרישות מהלקוח, ממומש במחלקהFormEventComposer .
* ה- **Builder** מקבל את הדרישות מהComposer, הוא זה שמבצע את הסינון ואורז הכל במקשה אחת, כל זאת מתרחש במחלקה EventsBuilder
* ה-**Complex Product** הוא המוצר המוגמר שמתקבל ע"י הBuilder ומיוצג בחלון – FormEvents.

הטופס FormEventComposer שואל את המשתמש על סמך איזה פרמטרים הוא רוצה לראות את רשימת האירועים, הפרמטרים שהחלטנו להציג ועל פיהם לאפשר את סינון הם:

* תאריך התחלת האירוע.
* מס' מקסימלי של משתתפים.
* מס' מינימלי של משתתפים.
* Sequence Diagram



* Class Diagram

### תבנית מס' 2 – [Factory Method]

**סיבת הבחירה / שימוש בתבנית:**

השתמשנו ב design pattern factory method על מנת ליצור מופע של משפחה פולימורפית היורשת מpanel משפחה זו משמשת אותנו לצורך הבדל בין הבחירות השונות מבחינת מגדר בטופס ההיכרות - FormMatch.

הצורך נבע מהרצון לייחד כל בחירה מגדרית בצבע שונה וצורה ייחודית. על מנת שיהיה בולט לעין של המשתמש ההבדלים בין הבחירות השונות.

**אופן המימוש:**

ה- **Client** מיוצג ע"י הטופס FormMatch , כלומר זה מגולם כאשר משתמש בוחר במגדר שבו הוא מעוניין.

עבור כל חבר של המשתמש הנמצא כמתאים לו על פי גלגל המזלות, הטופס FormMatch מייצר פאנל ייחודי למגדר שלו , יצירת הפאנל מתבצעת דרך מחלקה ייעודית אשר היא מגלמת את ה**-Factory Method ,** הנקראת PanelFactoryMethods ,תחת המתודה - CreatePanel , מתבצע סיווג לפי מגדר, ויצירה של ה**Product** , שהוא מיוצג כפאנל (תת מחלקה ).

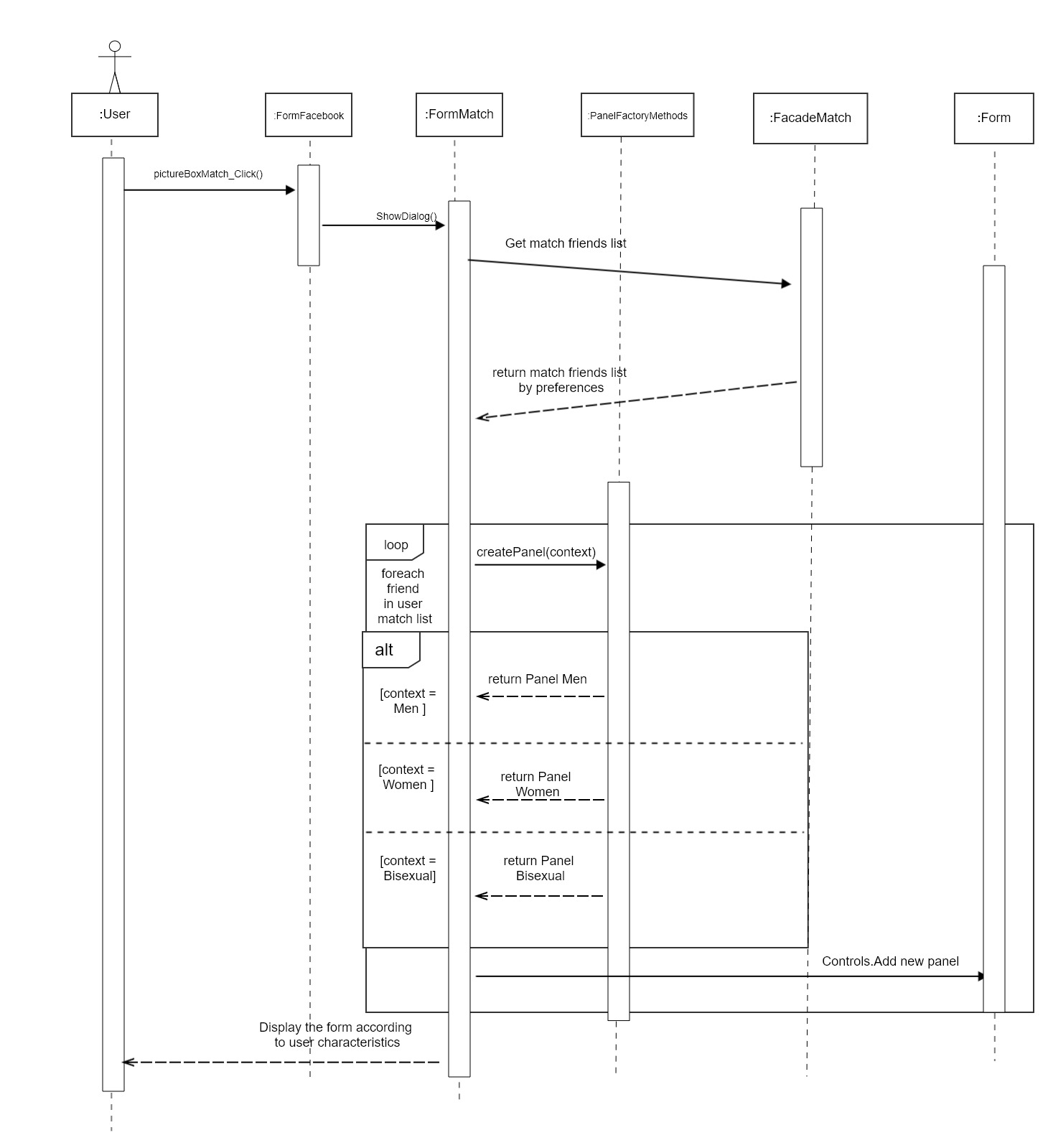
רצינו ליצור שוני בין בחירת המגדרים השונים וליצור פאנל עבור יוזר שירכז את המידע החשוב עליו :

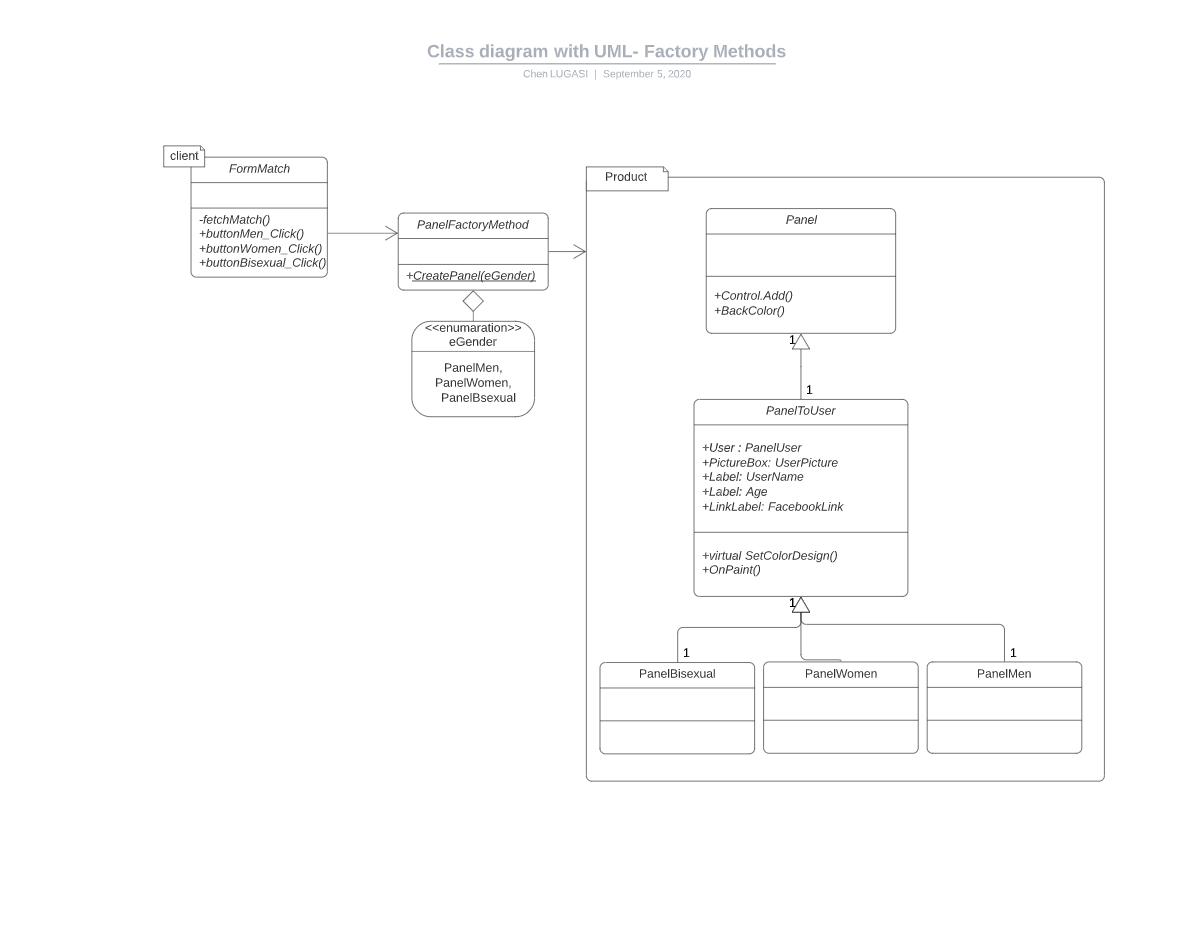
* תמונה
* שם
* גיל
* לינק לפייסבוק

לאחר שיצרנו את פאנל זה, שאר הפאנלים ירשו ממנו וישנו את העיצוב של הפאנל לפי המגדר אליו הם הותאמו ע"י דריסת המתודה שאחראית על עיצוב הפאנל שקיבלו בירושה.

המחלקה PanelFactoryMethods מקבלת בחירת מגדר ולפיה יודעת להחזיר את הפאנל המתאים למגדר זה.

המחלקות שמהוות את המשפחה הפולימורפית הן:

* PanelMen
* PanelWomen
* PanelBisexual
* Sequence Diagram
* Class Diagram



### תבנית מס' 3 – [Façade + Caching proxy]

**סיבת הבחירה / שימוש בתבנית:**

התבניות ממומשות בעיקר לטובת הטופס FormMatch , אשר צורך רשימת חברים מסוננת וממוינת על פי מזלות ועל פי מגדר.

Façade – מימשנו facade על מנת לפשט, לשטח ולהקל על המשתמש, לבצע פעולות שבפועל מבוצעות בתתי מערכות "מאחורי" הרכיב הזה -הוא מסתיר את הפעולות ומידע לא הכרחי מהמשתמש, הפסד אחראי בתוכנית על הכנת רשימת החברים של המשתמש כך שתהיה מסוננת על פי מגדר, ובנוסף מסוננת על פי גלגל המזלות, כך שהמשתמש מקבל רק את החברים שמתאימים לו ולא רואה את כל הפעולות מאחורי הקלעים.

Caching proxy – מימשנו Proxy ש"מתחזה" ל-Friends של המשתמש, ה-Proxy שומר את רשימת החברים של המשתמש בעת יצירתו ובנוסף מבצע חלוקה של רשימות חברים לפי מגדר.

זאת על מנת שכאשר המשתמש ירצה לעשות שימוש ברשימת החברים שוב ושוב, במידה ורשימת החברים לא השתנתה , הוא ישתמש ברשימה שמסופקת מהפרוקסי וכבר קיימת, אין צורך לבצע פנייה מחודשת לשרת הפייסבוק ולעבד את המידע מחדש כדי לקבל את תוצאות הנדרשות, הדבר הנ"ל חוסך את תעבורת המידע ברשת.

**אופן המימוש:**

בחרנו לשלב את שני הפטרנים Façade ו proxy.

ה- **Client** שדרך ה-FormFacebook בוחר לראות את ההתאמות שלו בפיצ'ר החדש המוצג לו, מבצע פתיחה לטופס FormMatch. טופס זה מבקש מהמשתמש לבחור נטייה מינית, ולאחר בחירתו מתבצעת פנייה אל **הפסד - FacadeMatch** לטובת הבאת רשימת החברים לאחר כל התהליכים בצורה מסוננת ומתאימה לפיצ'ר ולבחירת המשתמש.

הפסד מבצע כמה פעולות שמוסתרות מהמשתש :

1. מבצע פנייה לProxy על מנת לקבל את רשימת החברים מסוננת לפי מגדר שהמשתמש בחר.
2. מבצע פנייה למחלקה שנקראת – ZodiacData , אשר אחראית להביא לו את הסוג מזל של המשתמש, ושל חבריי המשתמש.
3. מבצע בדיקה עבור כל חבר של המשתמש האם הוא \ היא נמצאים מתאימים למשתמש על פי גלגל המזלות.
4. מחזיר רשימת חברים שמתאימים על פי המזל , או רשימה ריקה במידה ולא נמצאו חברים מתאימים.

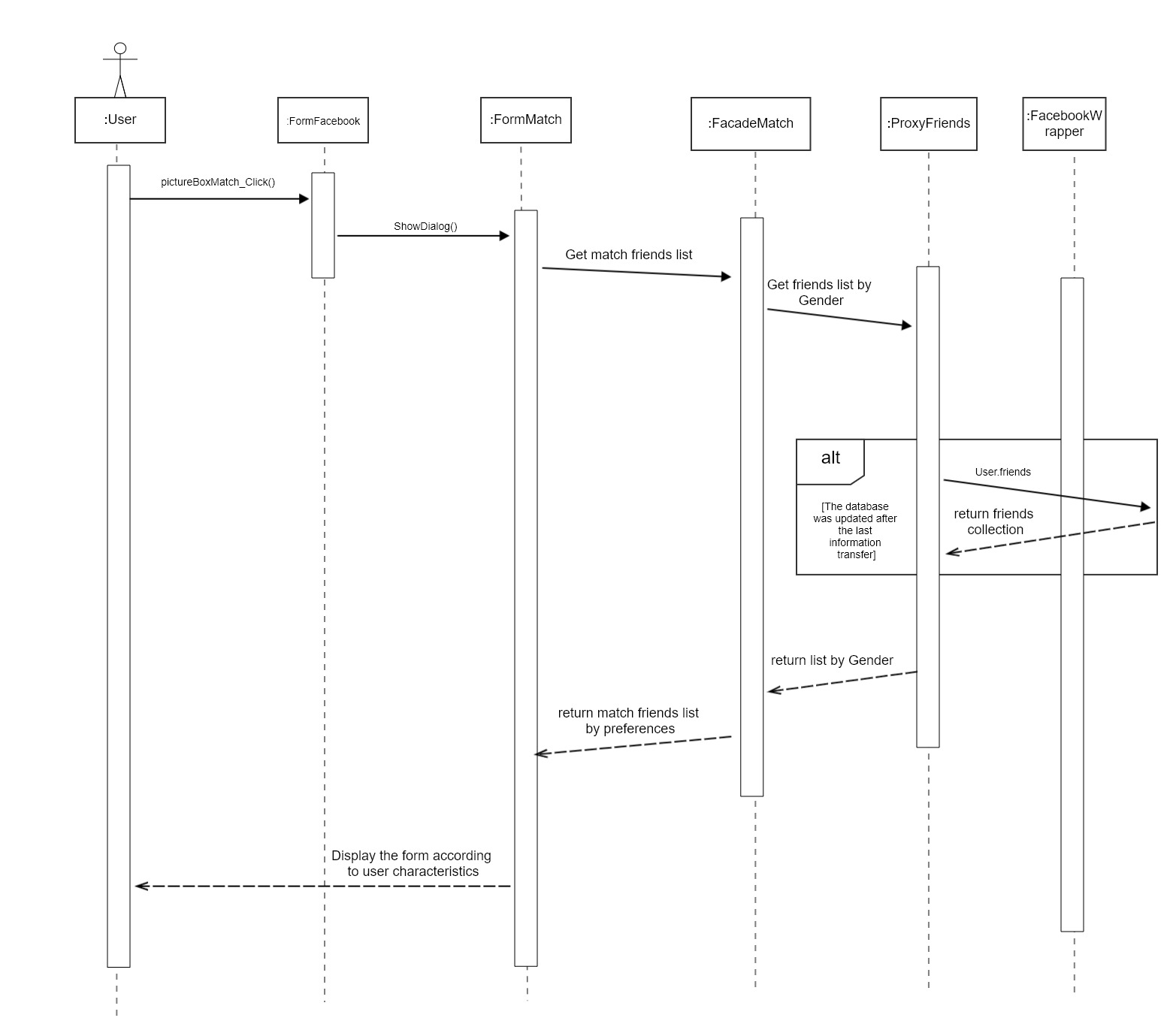
**פרוקסי**- **ProxyFriends** , תחילה נוצרות שלוש רשימות ( כל החברים, חברים ממין זכר, חברים ממין נקבה) , הפרוקסי בעת יצירתו בפעם הראשונה , פונה אל שרת הפייסבוק ומייבא ממנו את החברים של המשתמש, תוך כדי מיונם ע"פ מגדר.

במידה ומתבצעת יותר מפנייה אחת לפרוקסי, המחלקה מבצעת בדיקה האם רשימת החברים השתנתה ( כלומר האם נוספו \ נמחקו חברים), במידה וכן, המחלקה פונה שוב לשרת לקבלת המידע המעודכן, אחרת מחזירה את הרשימה שכבר מחזיקה ללא צורך לפנייה מחודשת לשרת.

בסוף התהליך, המחלקה מחזירה את רשימת החברים הרלוונטית שדרשו ממנה.

שתי המחלקות נמצאות בחלק הלוגי (תחת Logic) של המערכת כיוון ששתיהן לא מתייחסות לממשק המשתמש אלא מתעסקות בקוד המתבצע "מאחורי הקלעים" עבור כל ממשק שהוא, שמות המחלקות:

* Façade = FacadeMatch
* Caching proxy = ProxyFriends

**Sequence Diagram**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**Class Diagram**