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

* יצירה\עריכת פוסטים מוכנים מראש:

המשתמש יכול ליצור פוסטים מוכנים מראש ב editor ,לערוך אותם,למחוק ולהוסיף ולהשתמש בהם כtemplate ולפרסם בפוסט

* יצירת יום הולדת לחבר: אפשרות ליצור יום הולדת לחבר קרוב ויום הולדת לחבר רחוק על פי בחירת המשתמש. (פרסום הודעת מזל טוב ואלבום משותף – עבור חבר קרוב ועבור חבר רחוק יהיו 2 פוסטים שונים . עבור החבר הקרוב יותר הודעה רגשית יותר ועבור החבר הרחוק פחות). בנוסף הופסנו בהצגת יום הולדת של החבר (באיזור החברים באפליקציה) לוח שנה לועזי וכאשר לוחצים על תאריך הוא נותן את הלוח השנה העברי. בנוסף לכך תקפוץ ,כאשר בוחרים חבר מהרשימה בlistbox את תאריך היומהולדת העברי שלו (וגם יסומן הלועזי בלוח)

### תבנית מס' 1 – Facade

* סיבת הבחירה / שימוש בתבנית:

השתמשנו בתבנית זאת כפסאד "שקוף" על מנת להשתמש בכל נושא ההתחברות לפייסבוק ,קבלת access token , פרטי המשתמש (שם המשתמש, פרטיו ,הקבוצות שהוא משוייך אליהן, הדפים שעשה להם "לייק" וכו') . בשימוש בפסאד ,הפרדנו את הui מהמודל כך שבעתיד שנרצה לשנות את לוגיקת החיבור לפייסבוק ומתודות פנימיות נצטרך לשנות רק במקום אחד ולא גם במודל וגם בui. יצרנו LoginFacade אשר לו יש גישה אל facbookservices, facebookwraper ,AppSettings. על מנת להסתיר מהמשתמש את כל הגישה מול ממשקים של שמירת הקובץ והפייסבוק ונתנו לפסאד להיות אחראי על כך.

* אופן המימוש:

השחקנים: LoginFacade class , FacebookWrapper, AppSettings,LoginForm,MainForm

יצרנו מחלקתLoginFacade אשר מחזיקה בAppSettings ,LoginUser, LoginResult בתור חברי מחלקה והוא זה שמשתמש בהם ומממש את הלוגיקה של התחברות לפייסבוק,התנתקות, שליפת המידע,שמירת המידע (במידה והמשתמש בחר בזאת) של פרטי התחברות להתחברות אוטומטית .

הפסאד עושה את ההתחברות לפייסבוק באמצעות מתודה ובעצם כאשר המשתמש בוחר להתחבר הוא משתמש בפסאד בתור קופסא שחורה של התחברות ולא חשוף למה שהפסאד עושה מאחורי הקלעים (משתמש במחלקות הפייסבוק על מנת להתחבר ולקבל אקסס טוקן,ואובייקט של user .

2 הclient הם: MainForm,LoginForm אשר מחזיקים בfacade ומשתמשים בפונקציונליות שלו מבלי להיחשף לעבודה שלו מאחורי הקלעים.





### תבנית מס' 2 - Proxy

* סיבת הבחירה / שימוש בתבנית:

בחרנו להשתמש בתבנית זאת כפרוקסי יורש (class proxy) עבור הפיצ'ר של הלוח שנה העברי.

הלוח שנה עברי מתנהג כמו לוח שנה רגיל (גרגוריאני) אך עם התוספת של הלוח השנה העברי. הלוח השנה העברי "מתחקה" לreal subject שהוא לוח שנה חודשי רגיל.

* אופן המימוש:

ה"שחקנים" בתנית הזו : real subject: MonthCalendar (שיורש מcontrol (והפרוקסי הוא HebrewDateDetails שיורש מהreal subject.

המשתמש רוצה לקבל בעת בחירה בלוח שנה של יום בחודש (והוא מתנהג כמו monthCalander אך מציג לו את התאריך העברי.





### תבנית מס' 3 – Builder

* השתמשנו בתבנית זו בשימוש הפיצ'ר של "פרסום ברכת יומהולדת ויצירת אלבום" – הסיבה שהשתמשנו בפיצ'ר זה משום שהוא מורכב מכמה חלקים ויש חשיבות לסדר הפעולות. אם נרצה בעתיד להוסיף עוד משימות לבנייה,נוכל לשמור על רצף לוגי נכון ולוגיקת הבנייה מתבצעת ברכיב שמיועד לכך.

לכל סוג יומהולדת נוכל ליצור בילדר ייחודי ופולימורפי ולהרחיב את הפעילות בעתיד.

* אופן המימוש:

השחקנים: BirthDayManager אשר בתפקיד ה director ומחזיק בקומפוזיציה את ה אינטרפייס של BirthdayActivityBuilder : מחזיק 2 מתודות : BuildAlbum,BuildStatus

הדיירקטור אחרי על 2 הבניות המרכזיות : בניית סטטוס של מזל טוב ובניית אלבום משותף. הדיירקטור בונה את הפעולות מבלי להיות אחראי עליהן כך שאם נרצה בעתיד לשנות לוגיקה ולהוסיף לא תהיה בעיה לעשות את זה ולא נצטרך לעשות את זה דרכו אלא דרך מי שאחראי לזה.

BirthdayActivity שמממש את BirthDayActivityBuilder: אחראי על הלוגיקה של פרסום סטטוס "מזל טוב" על ה"קיר" של החבר ובניית האלבום המשותף.

CloseFriendBirthdayActivityBuilder וFarFriendBirthdayBuilder משמשים כconcreate builders

שניהם מממשים את BirthdayActivityBuilder ובונים את האלבום והסטטוס בצורה היחודית להם .





### עבודה אסינכרונית

בתוכנית שלנו השתמשנו בעבודה אסינכרונית בכמה שימושים:

בשליפת מידע מהפייסבוק של המשתמש בתוכנית בכלל הפיצ'רים:

Fetch albums

Fetch groups

Fetch friends

Fetch liked pages

Fetch posts

ע"י יצירת thread חדש לכל אחת מהמתודות אשר שולפות את המידע

וinvoke עבור פעולות של הפקדים שלא מאפשרים גישה עבור thread שהוא לא ה main thread שיצר אותם.

הסיבה שהשתמשנו בעבודה אסינכרונית עבור פעולות אלא,משום שלא רצינו "לתקוע" את הUI עבור המשתמש כאשר אנחנו מבקשים שליפה משרת הפייסבוק, ושהשתמש יוכל לעשות עוד פעולות עד שהui יוצג עבורו.

### עבודה עם Data Binding

בתוכנית שלנו השתמשנו בdata binding עבור הצגת הקבוצות של המשתמש והתמונת נושא של הקבוצה.

לצורך כך השתמשנו בהצגת הקבוצות של המשתמש.

השתמשנו ב DataBindingWizard בצור אוטומטית כך שבחרנו להציג את התמונה של המשתמש כאשר הוא בוחר בlistbox של הקבוצות את הקבוצה.

השורת קוד היחידה שכתבנו הייתה במתודה fetchGroups והיא: groupBindingSource.DataSource = LoginFacade.LoginUser.Groups;

על מנת לקשר בין הdata source והנתונים של הקבוצה של המשתמש.

בנוסף גם בfetchFriends השמשנו בזה.כאשר הList