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

* **חיפוש פוסטים לפי תאריך** – האפליקציה מאפשרת סינון וחיפוש פוסטים ע"פ תאריך.
* **חיפוש חברים ע"פ HomeTown** - האפליקציה מאפשרת סינון מתוך רשימת החברים את החברים אשר עיר המגורים שלהם היא אותה העיר אשר מוגדרת למשתמש המחובר לאפליקציה.

### תבנית מס' 1 – Singleton

* סיבת הבחירה / שימוש בתבנית:

בחרנו לממש את המחלקה AppManager כמחלקה סינגלטונית כיוון שלאפליקציה שלנו נדרש אך ורק מנהל אחד ויחיד אשר יכיל בתוכו את ה Form's השונים ביניהם LoginForm, AppSettingsForm.במידה והאפשרות ליצור 2 ומעלה instances של המחלקה הייתה קיימת, זהו פוטנציאל לבאגים כיוון שיחד איתה יווצרו הפורמים השונים אשר המחלקה מכילה, אין זה הגיוני להחזיק כמה מופעים של SettingsForm כמו כן LoginForm. בנוסף זהו בזבוז משאבים.

* אופן המימוש:

יצרנו מחלקה – AppManager שמטרתה לנהל את הform's השונים במערכת ולהגביל את מספר מנהלי האפליקציה ל1.

Access modifier של המחלקה שונה ל private. המחלקה הינה Sealed כי לא נרצה שתהיה אפשרות לרשת מהמחלקה (כי במידה וכן אז כשניצור אובייקט מהמחלקה היורשת אז נפעיל את הבנאי של הנורש והוא private).

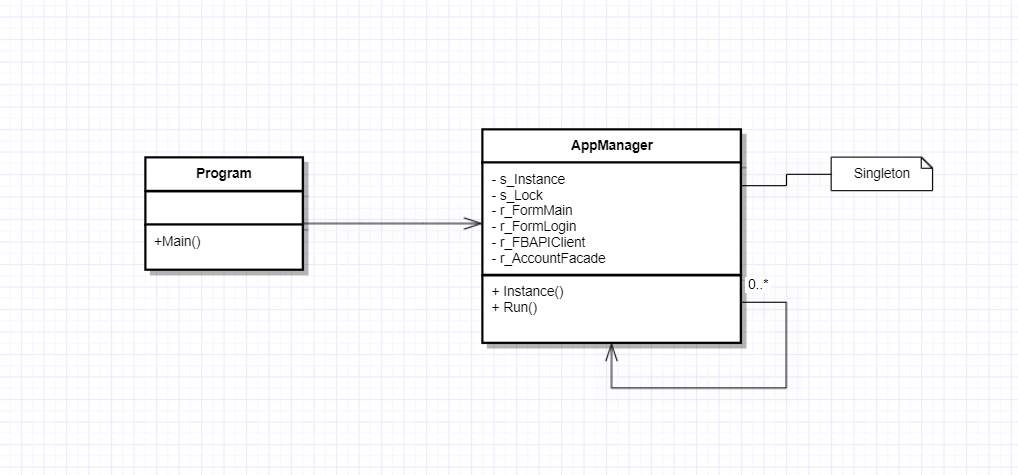
המחלקה מחזיקה שדה סטטי –

private static AppManager s\_Instance = null;

שדה זה מאותחל באופן Lazy רק במידה ונוצרת בקשה לקבל את הinstance.  
היווצרות האובייקט נוצרת בתוך getter של הפרופרטי Instance באמצעות שיטת double-check locking מה שהופך את היצירה ל thread-safe.

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

### תבנית מס' 2 – Facade

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### תבנית מס' 3 – FactoryMethod

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם