# דיאגרמות

דיאגרמת מסד הנתונים (Data Base Diagram)



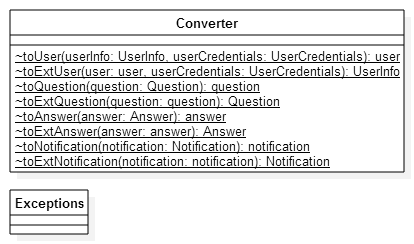
## דיאגרמת מחלקות

הערות כלליות עבור דיאגרמות המחלקות

הדיאגרמות יוצרו בעזרת StarUML2.

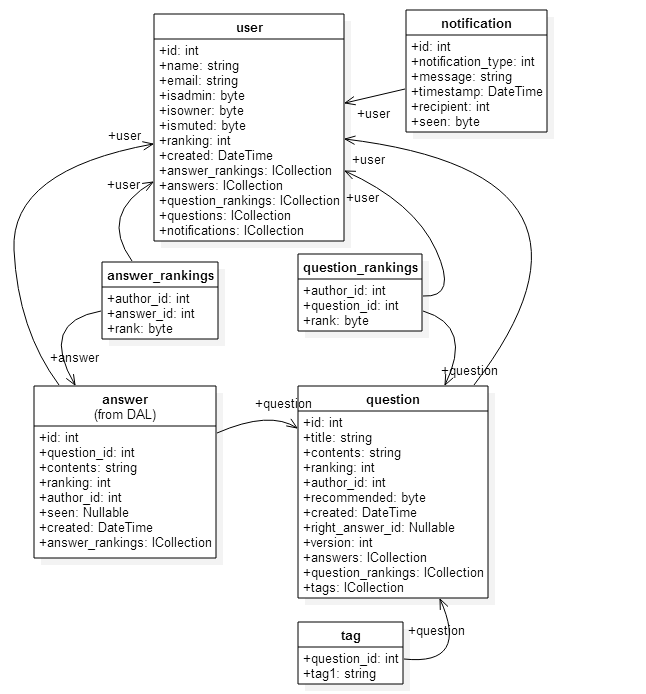
1. הסימן + מסמן גישה ציבורית – public
2. הסימן – מסמן גישה פרטית – private
3. הסימן ~ מסמן גישה פנימית – internal

### מחלקות כלליות בשכבת ה-DAL



### מחלקות הגישה לבסיס הנתונים בשכבת ה-DAL

מחלקות אלו נוצרו אוטומטית ע"י ה-EntityFramework. הדיאגרמה מציגה את הקשרים בינהן. המחלקות שנוצרו ע"י ה-EntityFramework מכילות שיטות רבות לגישה למאגר הנתונים, אשר אינן מופיעות בדיאגרמה זו.



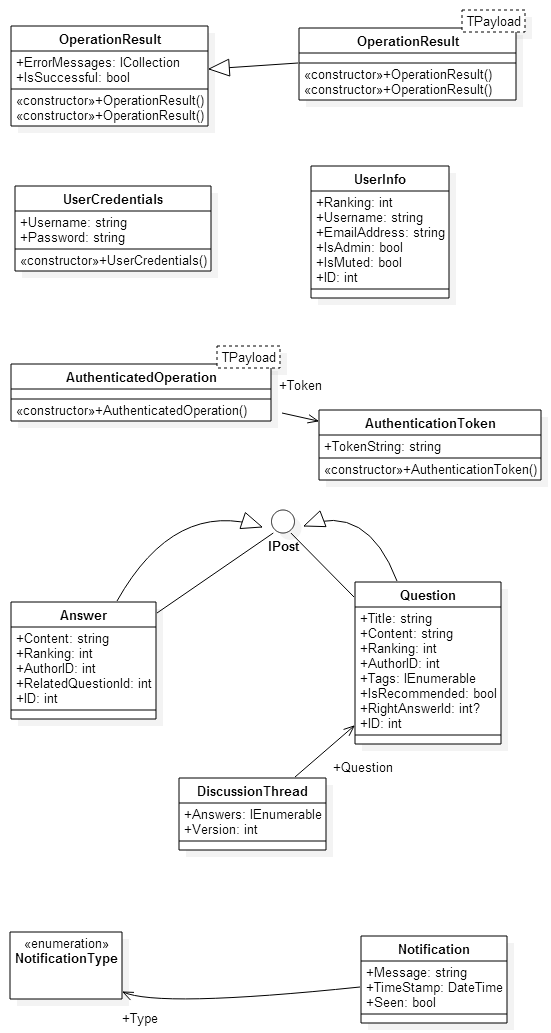
### מחלקות הפעילות בשכבת ה-DAL

מחלקות אלו הן המחלקות שחשופות לשכבת ה-BL לביצוע הפעילות מול מאגר הנתונים.



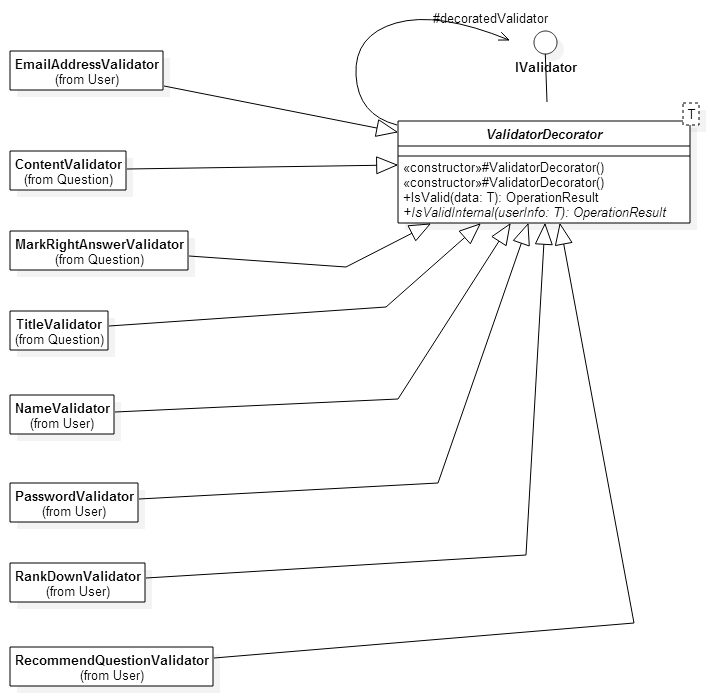
### מחלקות ספריית Common

ספריות אלו כוללות את האובייקטים הבסיסיים במערכת. כל השכבות האחרות משתמשות באובייקטים אלו.



### מחלקות מרחב Common.Logic

במרחב זה קיימים מרחבי User ו-Question המכילים את המימושים של ממשק IValidator בעזרת ירושה מ-ValidatorDecorator המאפשר מימוש של ולידציות לתצוגה ול-BL עבור כלל הפעולות שניתן לבצע במערכת.  
כל המחלקות מממשות את IsValidInternal כפי שמוגדר, ולכן המתודה לא מוצגת בדיאגרמה

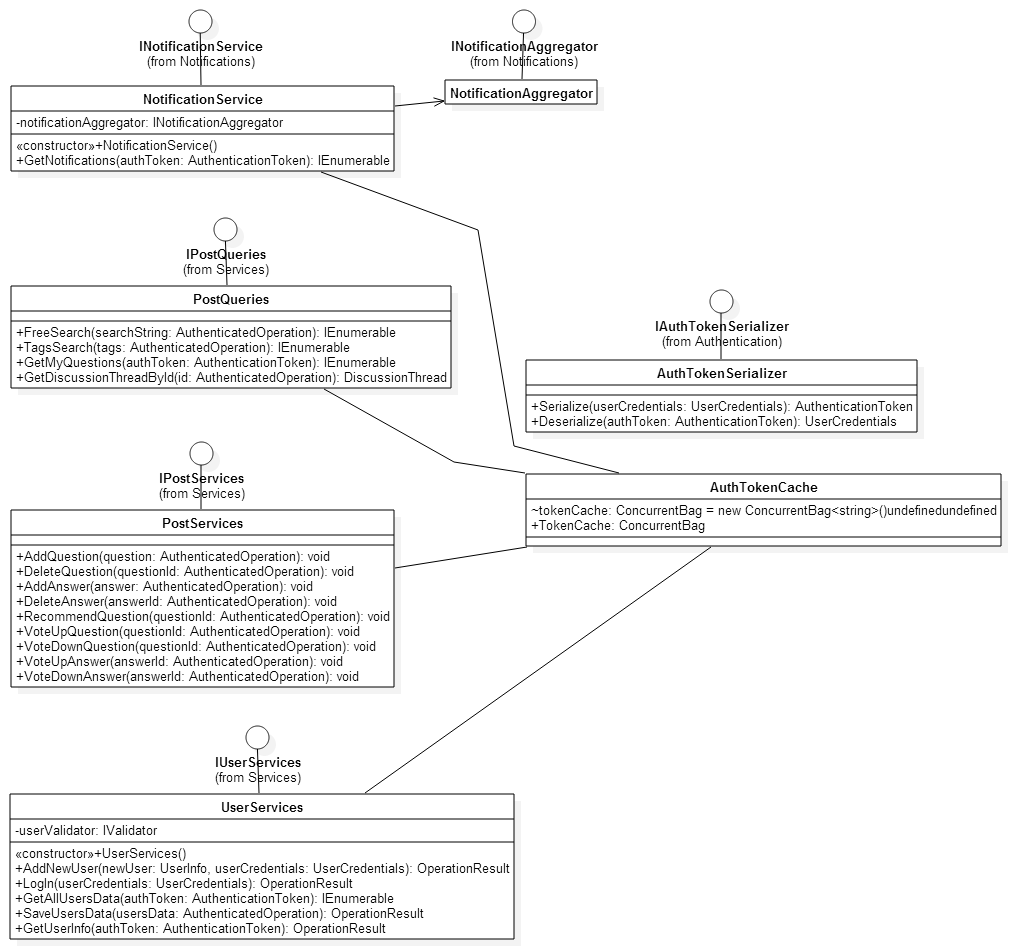


### מחלקות שכבת ה-BL

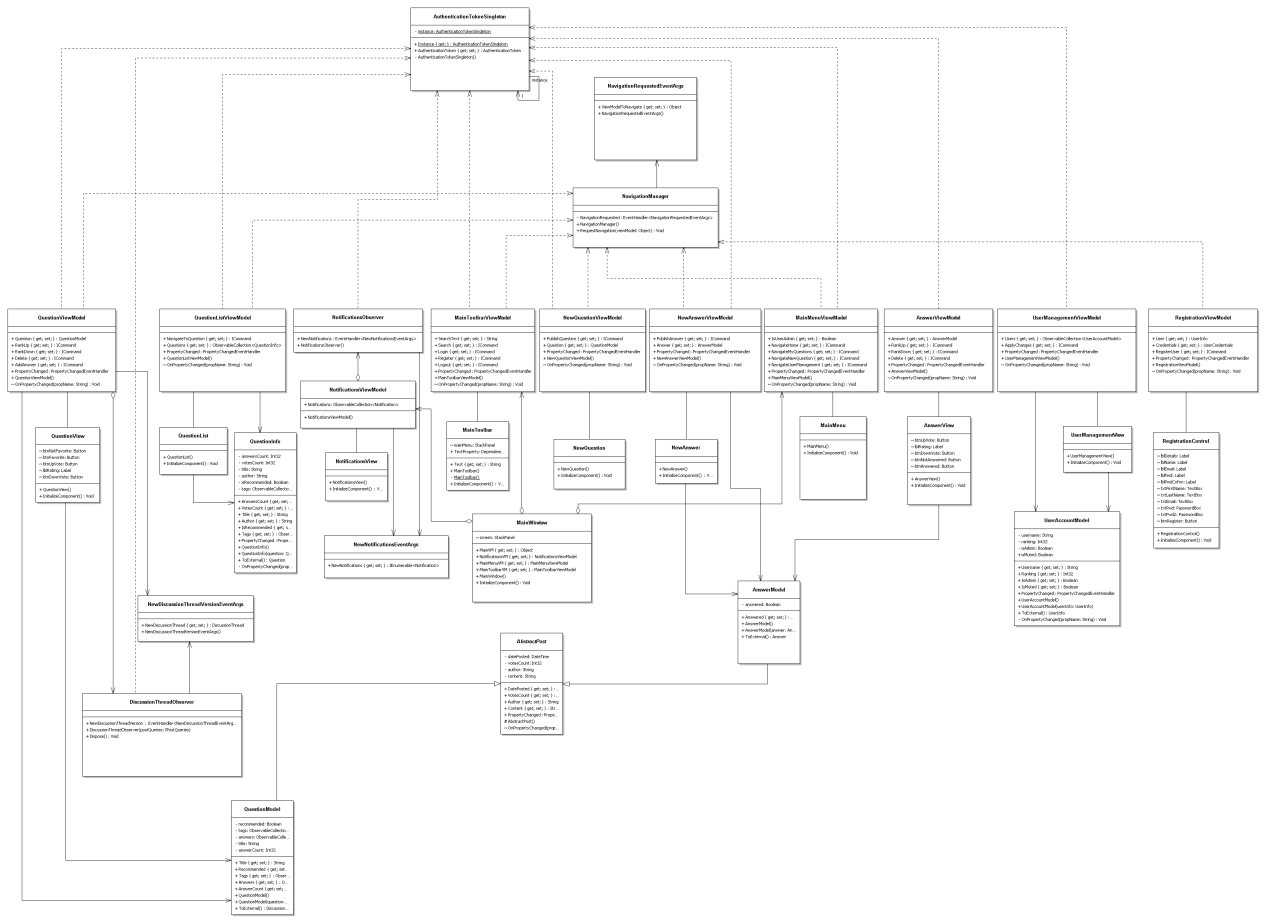
בשכבה זו נעשה שימוש ב-Services כדי לממש את הפעולות המבוצעות במערכת.

כלל האובייקטים בשכבה ממשים את הממשקים שהוגדרו בשכבת Common, אך מוצגים כאן

כל המחלקות בשכבה זו משתמשות במתודות לאחר שקיבלו הרשאה – כלומר ברשותן AuthenticationToken מתאים



### מחלקות בשכבת התצוגה

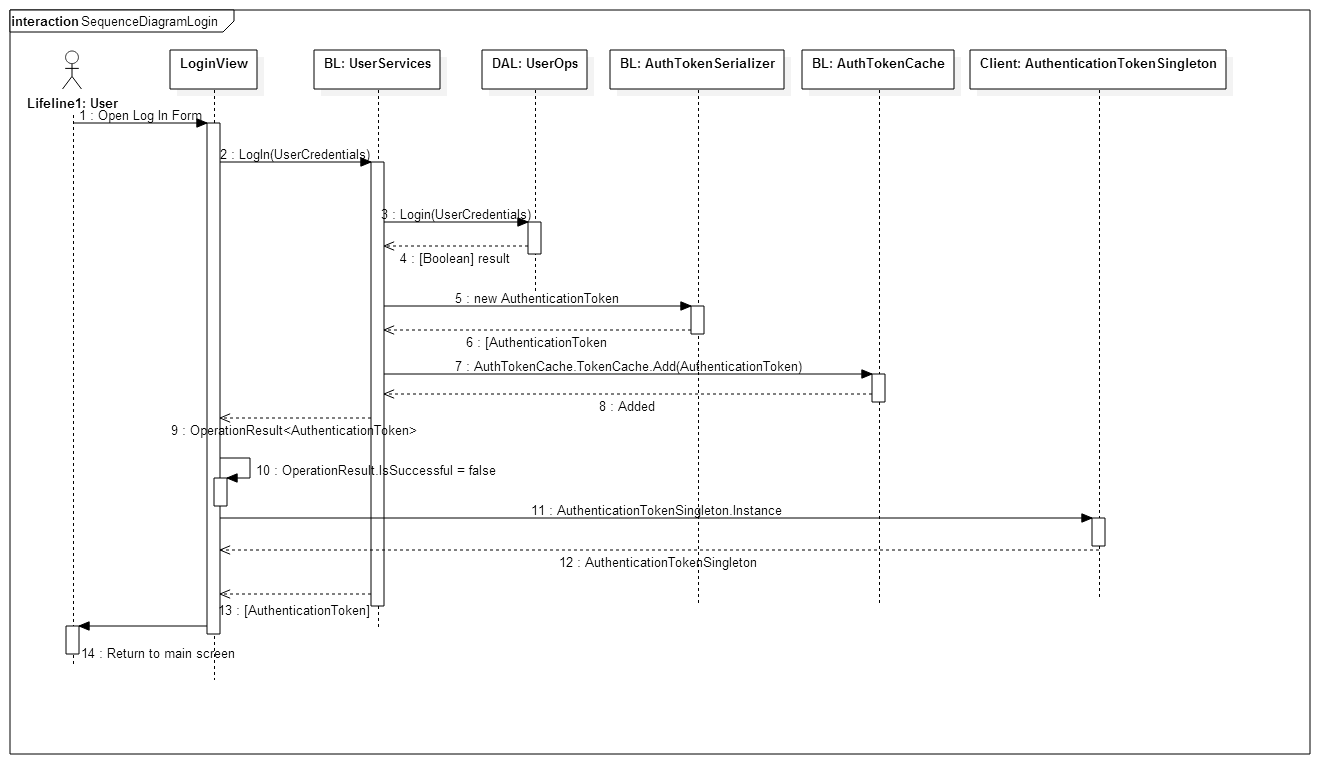


## דיאגרמות רצף - Sequence Diagram

דיאגרמות הרצף המפורטות מטה יתארו חלק מהתהליכים המרכזיים במערכת. התיאור הוא כללי וממצה ומטרתו להסביר את הלוגיקה בפעולות במערכת. בפועל יהיו הבדלים ושינויים בין הדיאגרמות והמימוש בפועל, שכן המימוש, בעיקר בשכבת התצוגה, הינו מורכב ביותר מחד, אבל עקבי וזהה ומתבסס על עקרונות ה-MVVM מאידך.  
כדי שהדיאגרמות יהיו בהירות ויסבירו את תהליכי העבודה, הן יכילו את תהליך העבודה המשמעותי, ולא יצמדו לדקויות בהעברת הודעות לרכיבי .NET WPF ודומיהם.

### דיאגרמת רצף עבור תהליך Login

עם הכניסה למערכת המשתמש יקליק על כפתור Login ב-MainToolBar. בתגובה המערכת תציג את מסך Login בו יקליד המשתמש את שמו ואת סיסמתו למערכת. השם והסיסמא יבדקו מול מאגר הנתונים בעזרת ה-DAL. במידה והשם והסיסמא נכונים, יווצר AuthenticationToken אשר ישמר ב-cache ב-BL כדי לחסוך בדיקות חוזרות מול מאגר הנתונים עבור כל פעולה. ה-Token יוחזר לשכבת הפרזנטציה, וישמר כ-singleton ב-client. מסך ה-login יסגר, והשליטה תוחזר למשתמש במסך הראשי.  
בכל שלב בדרך בו תהיה שגיאה, תוחזר הודעת השגיאה למשתמש במסך ה-login כדי לאפשר לו לתקן ולנסות שוב את התהליך.



### דיאגרמת רצף תהליך ולידציה של משתמש (האם המשתמש הנוכחי ביצע login מסודר למערכת)

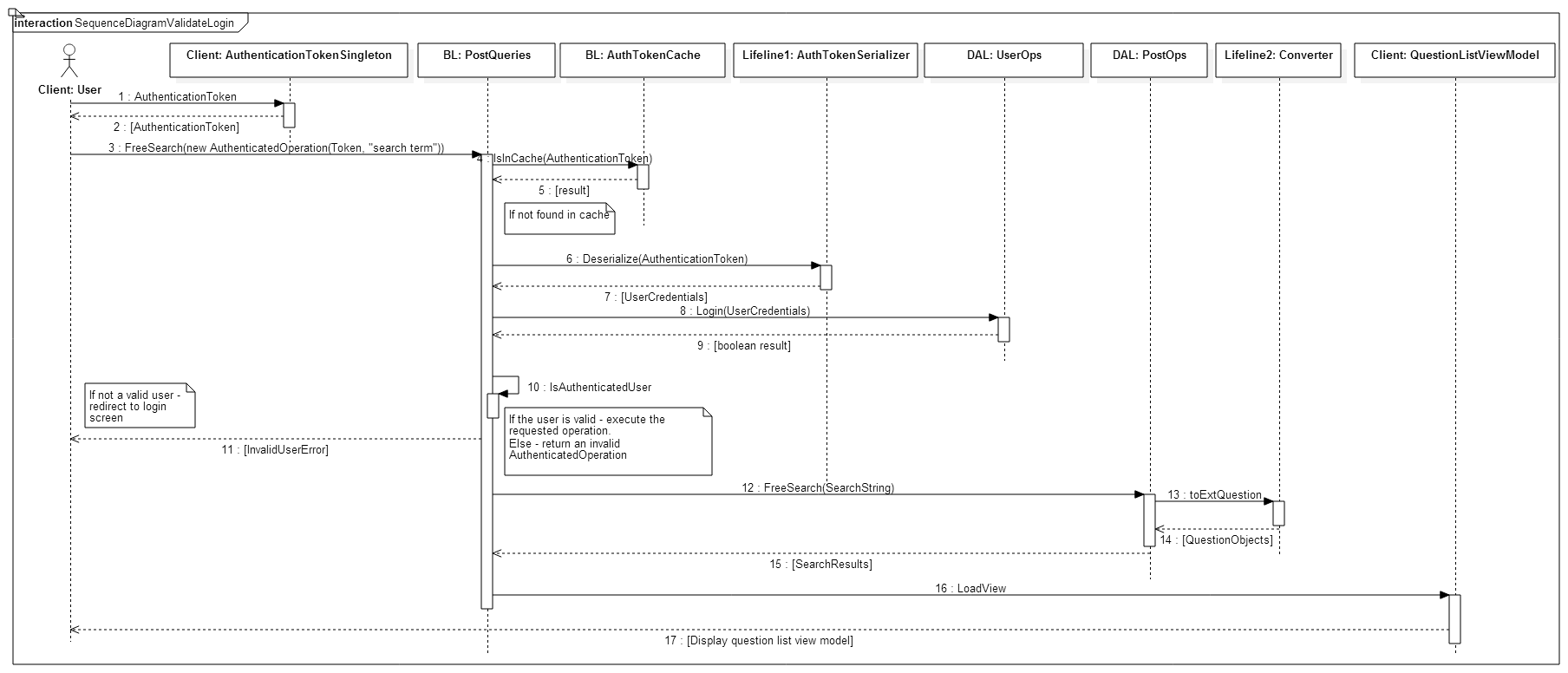
כל בקשה לפעולה של המשתמש במערכת מתחילה בבדיקה האם המשתמש מחובר למערכת.

שכבת התצוגה שומרת את ה-AuthneticationToken של המשתמש ב-singleton בצד האפליקציה.

בצד שכבת ה-BL, ה-Token-ים נשמרים ב-cache.

כל פעולה כוללת פרמטר של AuthneticationToken הנשלח מצד ה-Client עם שאר הפרמטרים של ה-operation. בצד ה-BL הם מושווים ל-Token שב-cache, אם קיים, או מושווים דרך ה-DAL לנתונים שבמאגר הנתונים.  
אם קיימת התאמה, המשתמש מזוהה, וניתן להמשיך ולבדוק האם הוא מורשה לביצוע הפעולה הספציפית, או פשוט לבצעה.

דיאגרמה זו מתארת תהליך ביצוע חיפוש טקסט חופשי במערכת, כיוון שזוהי פעולה פשוטה, אותה מורשים כל המשתמשים לבצע. רק לאחר סיום תהליך הולידציה יבוצע החיפוש בפועל, התוצאות ישלפו ממאגר הנתונים, יומרו לאובייקטים המוכרים (ב-common) וישלחו חזרה לשכבת התצוגה שתציג את ה-view המתאים



### דיאגרמת רצף כתיבת תשובה

דיאגרמה זו באה להדגיש את ההתנהגות בעת הוספת עדכון למערכת, אשר גורר שינויים רבים ב-BL ובעקבותיו במאגר הנתונים, אליהם המשתמש אינו מודע.

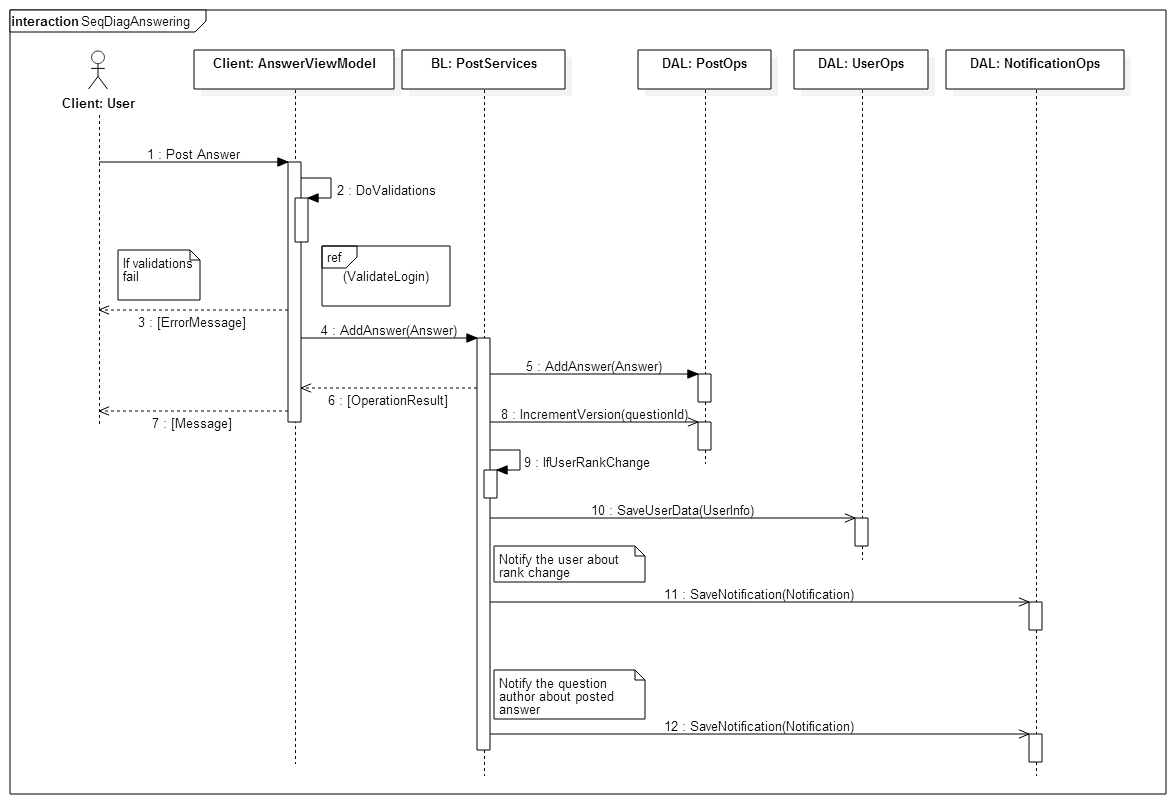
כאשר משתמש כותב תשובה לשאלה ושולח אותה למערכת, המערכת מוודאה שהמשתמש אכן מחובר למערכת (מוסבר בדיאגרמה הקודמת), ומעבירה את התשובה ל-BL שמירה ב-DAL.

בנוסף, מתבצעים התהליכים הבאים:

1. ולידציות לתוכן ההודעה ב-PL.
2. בדיקה האם הגיע הזמן להעלות את דירוג המשתמש במערכת, והעלאת הדירוג במידה וכן
3. יצירת התראה עבור כותב השאלה
4. העלאת ה-version של השאלה.

אחרי ולידציה מוצלחת, למעט שמירת השאלה, שאר הפעולות יכולות להתבצע בצורה אסינכרונית.

אותו תהליך בדיוק קורה בכל פעולה של המשתמש במערכת – בעת שינוי דירוג של שאלה, בעת המלצה על שאלה, כאשר תשובה מסומנת ע"י מרצה כנכונה. דיאגרמה זו מהווה template עבור כל אותן פעולות, שההבדל היחידי בינן הוא הנתונים שהמשתמש מעדכן.



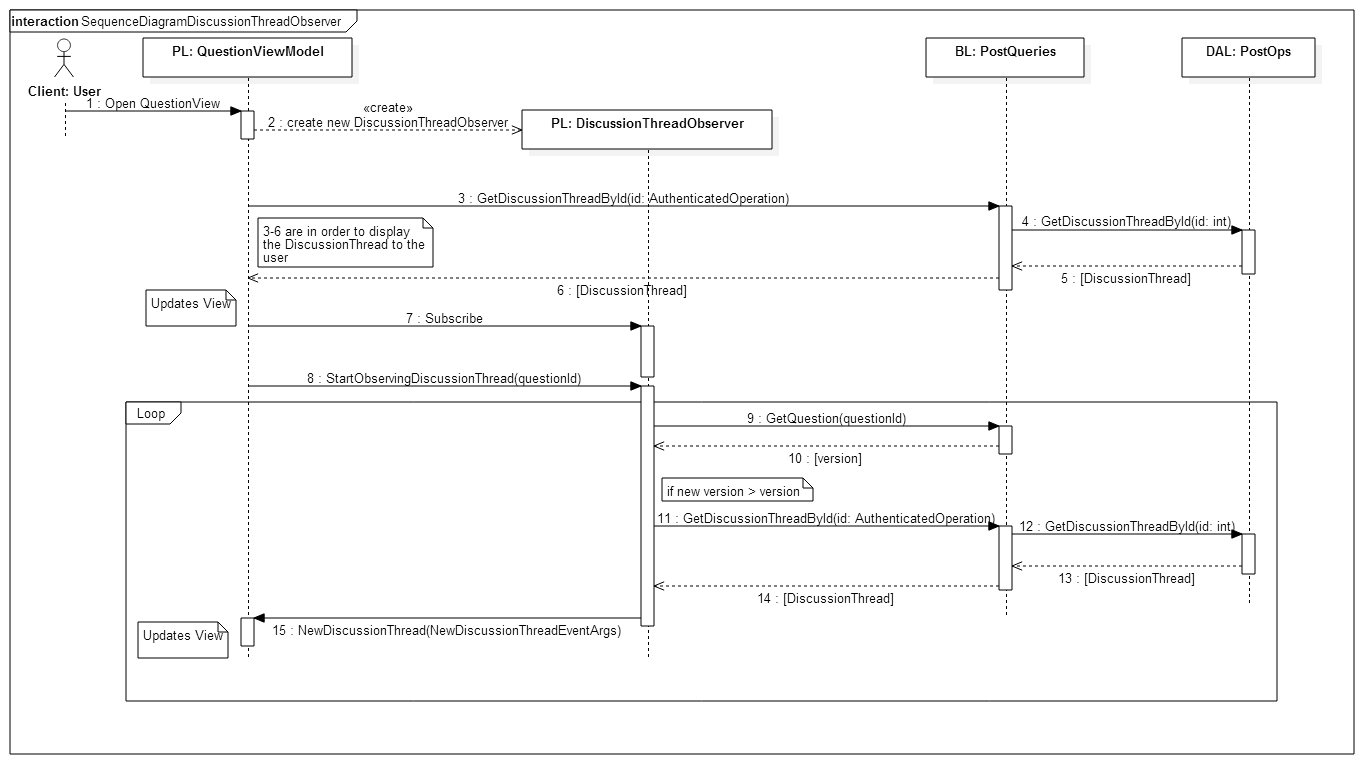
### דיאגרמת רצף מנגנון עדכון התצוגה

דיאגרמה זו ממחישה את פעולתו של מנגנון ההתרעות במערכת. בעת צפיה במסך כלשהו, פועל ברקע מנגנון ההתרעות. במידה ו-version של שאלה (ולכן של כל ה-DiscussionThread) משתנה, מנגנון ההתרעות דוחף את השינויים למקומות הרלוונטיים במסך, כדי למנוע מצב בו משתמש שמשתהה במסך מסויים לא ידע שהנתונים שמוצגים בו כבר אינם עדכניים.

מנגנון ההתרעות פונה כל פרק זמן קבוע ל-DAL בבקשה לקבלת גרסת השאלה העדכנית. במידה והיא שונה מהגרסה הנוכחית, הוא שולף את פרטי ה-DiscussionThread העדכניים ומעדכן את התצוגה.

הרצף הוא כזה – ה-QuestionViewModel נרשם ל-Event של ה-DiscussionThreadObserver. ואז קורא ל- StartObservingDiscussionThread כשהוא מעביר לו את מזהה השאלה שצריך לנטר.

ה-DiscussionObserverThread מריץ Thread שבלולאה שולף מה-DAL את נתוני השאלה ובודק אם ה-Version השתנה, ואם כן, הוא שולף את נתוני ה-DiscussionThread ממאגר הנתונים ומקפיץ את ה-Event



בנוסף, ובמקביל, מנגנון ההתרעות שולף התרעות כלליות עבור המשתמש, במידה והן קיימות הוא מציג אותן בפאנל ההתרעות. כאן, הוא שולף את ההתרעות בהתאם לחשבון המשתמש – במידה ומדובר בסטודנט הוא שולף התרעות רק עבור המשתמש, במידה ומדובר במרצה הוא שולף התרעות עבור מרצה.

