**אפיון פרוייקט מערכת שעות**

**ישויות**

יישות User- משתמש

* Id- מפתח
* Name
* Password
* Email
* Role [{ manager, worker} enum Roles]
* מאפיין צל: ListPresences המקשר את כל הרשומות מטבלת נוכחות השייכות למשתמש זה.

ליישות זו 2 בנאים- בנאי אח שמקבל שם ומספר מזהה בלבד

ובנאי נוסף ההמקבל את כל הפרמטרים.

יישות Presence נוכחות

* Id-מפתח
* UserId- מפתח זר לטבלת המשתמשים
* Date
* TimeOfStart
* TimeOfEnd
* Type [enum TypeOfTime{Presence, Disease, Vacation, egnancyHours}]
* Approval מפתח זר לטבלת אישורים

יישות Approvals -אישורים

* Id
* StatusRequet enum Status{InProcessing, approved, notApproved}]]
* FileName
* ContentType
* FileData

**קונטרולרים**

הקונטרולרים בכללם מממשים את פעולות הבסיס הדיפולטיביות, מלבד PresencesController

המממש מלבדן גם מתודת GetByUserIdAsync המקבלת משתמש ומחזירה את כל הנוכחויות הקשורות אליו [יכולתי להפעיל זאת ישירות על המשתמש, נכון רק שאת ההקשרים האלו עשיתי אח"כ....] וכנ"ל מתודת GetByDateAsync המקבלת תאריך ומחזירה את כל הנוכחויות הרלוונטיות.

**Middlewares**

LogMiddleware

מידלוור זה אחראי על כתיבת הלוגים לקובץ logfile.   
הוא כותב ללוג איזו בקשת API נשלחה , שגיאה אם ישנה וסטטוס הבקשה בחזור.

ShabesMiddleware

מידלוור זה בודק האם הבקשה נשלחה בשבת- ואם כן מחזיר אותה ומתעד ללוג את תחילת הבדיקה ואת התוצאה.

TokenMiddlware

מידלוור זה אחראי על חידוש הטוקן עבור כל קריאת API

במידלוור זה שימוש במתודת GenerateToken. המתודה מקבלת טיפוס IEnumerable<Claim> currentUserClaims ומחזירה טוקן חדש מסוג string