Buisness Layer

```
BoardController
Register (email) : void
+ addBoard (email, name) : void
 removeBoard (email, name) : Board
+ addTask (email, boardName, title, description, duaDate) : Task
moveTask (email,boardName, columnOrdinal TaskId) : void
+ changeDueDate (email, boardName, ColumnOrdinal, taskId, dueDate) : void
+ changeTitle (email, boardName ,columnOrdinal, taskld, title) : void
changeDescreption (email, boarName,ColumnOrdinal, taskId, description): void
+ GetColumnName (email,boardname,columnOrdinal):string
+ LimitColunm (email boardname columnOrdinal limit) : void
+ GetColumnLimit (email,boardName,columnOrdinal): int

    GetColumn(email, boardName, columnOrdinal) : List<Task>

InProgressTasks (email) : List<Task>
FindBoard (email, boardName) : Board
BoardsToList (email) : List<Board>
                                                  -dictionary <email,boards>
                                           Board
 name: string
baord : List<Colunm>
idcounter = 1 : int
+ addTask (dueDate, title, description) : Task
+ RemoveTask (id, columnOrdinal) : Task
+ moveTask (id, columnOrdinal) : void
+ changeDueDate (id, columnOrdinal, newdueDate) : void
+ change Title (id, columnOrdinal, title): void
+ change Descreption (id, columnOrdinal, description): void
+GetColumnName (columnOrdinal): string
+LimitColunm (columnOrdinal.limit) : void
+GetColumnLimit (columnOrdinal): int
GetColumn (columnOrdinal) : List<Task>
 GetBoardId(): int
                                       column
 name: string
tasks : Dictionary<int, Task>
columnLimitVerifier : boolean = false
columnLimiter: int
+ limitTasks (int) : void
+ GetColumnLimit(): int
+ addTask (id, dueDate, title, description) : Task
+ removeTask(id) : Task
+ changeDueDate (id, Duedate) : void
+ changeTitle (id, new Title) : void
+ ChangeDescreption (id, newDescription) : void
+ GetColumnName() : string
+ Tasks : List<Task>
                                             dictionary<id,task>
+ creationTime : DateTime
dueDate : DateTime
title : string
 descreption : string
 final static titleLength : int = 50
- final static descreptionLength : int = 300
+ changeDueDate (NewDueDate) : void
+ changeTitle NewTitle) : void
+ changeDescreption (NewDescription) : void
+ GetDueDate() : DateTime
+ GetDescription(): string
+ GetTitle() : string
+ GetId(): int
+ GetCreationTime() : DateTime
 CheckDuedate(dueDate) : void
 CheckDescription(description): void
 CheckTitle(title): void
```

UserController

register (email_pass) : void

► ValidateUserLoggin (email) : void

1 n

PasswordVerifier (pass): void

email: string password : string Isloggedin: boolean login (email, pass): void + logout() : void - IsValidEmail (email) : boolean ValidateUserLoggin(): boolean

- users

login (email, pass): User

logout (email) : void

שינויים מהדיאגרמה הראשונית:

BoardController:

הוספנו שיטה של הוספת Board למילון, הוספנו Getters לשם עמודה, להגבלה של משימות בעמודה, ולמספר עמודה. הוספנו אפשרות להגבלת המשימות בכל עמודה, ואפשרות לראות inProgress משימות שנמצאות כרגע כמו כן, הוספנו 2 שיטות פרטיות על מנת שיקלו את Service השימוש במימוש

Board:

העברנו את השיטה להסרת משימה למחלקה זו, הוספנו Getters שיורשים מהBoardController, והוספנו עוד שיטה פרטית לקבלת הID של Board

Column:

הוספנו מתודה שמחזירה רשימה של כל המשימות, על מנת שיהיה קל יותר לעקוב אחריהן

Task:

הוספנו Getters לדברים הבאים: תאריך סיום, תיאור משימה, ID, כותרת משימה, זמן יצירת המשימה. בנוסף, 3 שיטות פרטיות שמוודאות תקינות של תאריך הגשת משימה, של תיאור משימה ושל כותרת משימה.

<u>UserController:</u>

הוספנו User, הוספנו עדיקת תקינות הסיסמא ל את האפשרות להתנתקות גם מהמחלקה הזו, וגם מהשer

User:

הוספנו מתודה שבודקת את תקינות המייל, שיטה שבודקת את תקינות הסיסמא לפי כל הדרישות, ושיטה שבודקת האם המשתמש מחובר