**תיעוד שינויים ב- design:**

**19/5/2022**

שינויים ב- service layer:

* בכל פונקציה במחלקות ב- serviceLayer הוספנו משתנים בהתאם לפונקציות בבדיקות שלנו. למשל הוספנו מייל כדי שבכל רגע נתון נדע בנתונים של איזה משתמש אנחנו מסתכלים.
* הוספת הפונקציה addBoard במחלקה UserService, מפני שביצירת לוח חדש אנחנו רוצים לוודא שהלקוח רשום ומחובר למערכת.
* הוספת הפונקציה removeBoard במחלקה UserService, מפני שבביצוע הסרת לוח אנחנו רוצים לוודא שהלקוח רשום ומחובר למערכת, כדי לבצע את הפעולה.
* הוספת משתנים לכלל הפונקציות במחלקה ColumnService שכן אם נרצה לעבוד על עמודה ספציפית אנחנו צריכים למצוא אותה לפי הלקוח, ולפי הלוח שהעמודה משויכת אליו.
* הוספת שדה פרטי של UserController למחלקות BoardService ולמחלקה TaskService כדי שנוכל לדעת בכל רגע נתון את המצב של המשתמש במערכת – האם הוא מחובר ויכול לבצע פעולות או לא.
* העברת הפונקציות addBoard והפונקציה removeBoard מהמחלקה BoardService אל המחלקה UserSrevice מפני שהלוחות נוצרים שם ונשמרים אצל המשתמש ברשימה של לוחות פרטית שהיא מחזיק.
* במחלקה Response שמרנו את המשתנה errorMassage והמשתנה returnValue כדי שנוכל להחזיר במחרוזת את הג'ייסון עם הודעת שגיאה מתאימה, או ערך ההחזרה המתאים בהתאם לכל פונקציה.

שינויים ב- business layer:

* הורדת המחלקה BoardController מפני שאנחנו שומרים לכל משתמש מילון עם הלוחות שהוא יצר בשדה פרטי במחלקה שלו. כל הפונקציות שניתן לעשות על הלוחות נעשות במחלקות שמממשות את המשתמש או את הלוח.
* הורדת השדה columnID במחלקה Column מפני שכאשר אנחנו ניגשים לעמודה מסוימת בלוח, אנחנו ניגשים אליה דרך ב- Board, ולכן בהכרח אנחנו יודעים באיזה עמודה אנחנו.
* במחלקה Column הוספנו מתודה של getTask שמחזירה לנו משימה מתוך העמודה, ואת המתודה של isExist שמחזירה האם המשימה קיימת בעבודה שלנו. הן משמשות מתודות שונות ולכן אחת מהן מחזירה את המשימה אם נמצאה, והשנייה מחזירה ערך בוליאני של אמת או שקר בהתאם לשאלה האם המשימה נמצאה או לא בהתאם.
* במחלקה UserController הוספנו את הפונקציה isValidEmail שבודקת האם כתובת המייל שהקבלה תקינה.
* במחלקה UserController הוספנו את הפונקציה isLogIn שקבלת מייל של לקוח מסוים ומחזירה האם הוא מחובר למערכת או לא. אנחנו משתמשים בה כמעט בכל הפונקציות שמשנות דברים מצד הלקוח.
* הוספת שדה של מילון במחלקה User שנקרא boards ששומר את כל הלוחות שהלקוח יצר.
* הסרת השדה של UserController מהמחלקה User שכן ה- UserController שומר מילון של משתמשים, אבל משתמשים לא צריך לשמור קישור לכלל המשתמשים במערכת.
* הורדת הפונקציה של isValidPassword מהמחלקה של User שכן היא נמצאת ה- UserController והיה כפילות של הפונקציה בקוד.
* הוספת המתודות של לוח אל User שכן אפשר לבצע שינויים על לוחות (להוסיף לערוך לחפש ולמצוא) במחלקה של user כי היא מחזירה מילון של הלוחות של הלקוח.

**2/6/2022**

שינויים ב- service layer:

* העברת הפונקציות של הוספת לוח והורדת לוח מה userservice אל boardservice.
* הוספת פונקציות של loadData ו- deleteData מהמחלקות של userservice boardservice.
* הוספת פונקציה returnJson לכל אחת מהמחלקה כדי לחסוך כפילויות קוד בכל בדיקה.
* פונקציות ChangeOwner JoinBoard LeaveBoard GetBoardName למחלקה boardservice.
* פונקציות assignTask במחלקה taskService.

שינויים ב- business layer:

* הוספנו boardController.
* הוספנו פונקציות changeOwnerבמחלקה board.
* במחלקה user הוספנו את הפונקציות joinBoard, leaveBoard.
* הוספנו שדה של usercontroller במחלקה boardController ששומר את כל המשתמשים לכל הלוחות.
* שמרנו רשימה של יוזרים לכל לוח ששומרת את המשתמשים שלו, ומי הבעלים של הלוח הזה.
* הוספנו פונקציות של loadData, deleteData במחלקות קונטרולרים.
* הוספנו במחלקה task שדה פרטי ששומר לי הרשום אלו (assignee).
* הוספנו שדות של DTO שמקשרים כל מחלקה למחלקה המתאימה לה ב- dataAccessLayer.

שינויים ב- data access layer:

* הוספנו את המחלקה userDTO שתהיה אחראיות על המידע שנשמר על כל המשתמשים.
* הוספנו את המחלקה boardDTO שתהיה אחראיות על המידע שנשמר על כל הלוחות במערכת.
* הוספנו את המחלקה taskTDO שתהיה אחראיות על המידע שנשמר על המשימות בלוחות.
* הוספנו את המחלקה UserBoardDTO שתהיה אחראיות על קישור הלוחות למשתמשים.