**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.

**21/6/2022**

שינויים ב- service layer:

* במחלקה BoardService:
  + הוספנו את הפונקציה TransferOwnership שמקבלת את המנהל הנוכחי של הלוח ואת המנהל שרוצים להעביר אליו את הלוח.
* במחלקה ColumnService:
  + הוספנו שדה פרטי של BoardController.
  + פונקציה של GetCloumn שמחזירה את העמודה לפי השם שלה.
  + פונקציה של GetColumnNameStr שמחזירה מחרוזת של השם של העמודה (לא ב JSON).
* במחלקה TaskService :
  + הוספנו שדה פרטי של BoardController.
  + הוספנו את הפונקציה של InProgressTasks שמחזירה את כל המשימות שבהכנה עבור המשתמש.
  + שינינו את כלל הפונקציות שיקבלו את השם של העמודה ואת המספר הסידורי של הלוח (במקום את שמו).
* במחלקה UserService :
  + הוספנו פונקציה GetUserBoards שמחזירה רשימה של המספרים שסידוריים של הלוחות שבבעלות אותו המשתמש.
  + הורדנו את הפונקציה InProgressTasks מפני שהעברנו אותה ל- TaskService כי הוא מחובר למחלקה BoardController אבל UserService לא.
* הוספנו מחלקה של ServiceFactory שאחראית על אתחולים של כל המופעים של המחלקה ב- Service Layer.

שינויים ב- business layer:

* במחלקה User:
  + הורדנו את השדה הפרטי ששמר משתנה של UserController.
  + העברנו את הפעולות IsValidPassword , IsValidEmail למחלקה User מהמחלקה UserController.
  + שינינו את הפונקציה login , logout שלא יקבלו את האימייל של המשתמש.
  + העברנו את הפונקציה InProgressTasks למחלקה BoardController.
  + הורדנו את הפונקציה IsExists.
  + העברנו את הפונקציות שקשורות לקשר בין המשתמש והלוחות למחלקה BoardController : GetBoard, JoinBoard , LeaveBoard.
  + הוספנו את הפעולה AddBoard שהוסיפה את המספר הסידורי של הלוח שהיא קיבלה לרשימה של המספרים הסידוריים של הלוחות של אותו המשתמש.
* במחלקה UserController:
  + הורדנו את הפעולה של DeleteUser.
  + הוספנו את הפעולה CheckUser שמחזירה האם המשתמש רשום ומחובר למערכת.
  + הורדנו מהמחלקה את המתודות IsValidPassword , IsValidEmail כי העברנו אותם למחלקה User.
* במחלקה Board:
  + הפכנו את השדה הפרטי columns שישמור מילון של העמודות עם השם של כל אחת כמפתח סידורי, במקום לשמור אותן ברשימה.
  + שינינו את השם של השדה הפרטי ששמר את המספר הסידורי של המשימות בלוח מהשם IDcounter אל counterIDtask.
  + הוספנו פונקציה של AddUser שמאפשרת להוסיף משתמש ללוח כלשהו.
  + הוספנו לפונקציה AdvanceTask משתנה של useremail שמקבלת את המשתמש שאמור להיות assign שלה.
  + הוספנו את הפונקציה AddTask שמוסיפה משימה ללוח.
  + הוספנו את הפונקציה GetColumn שמחזירה עמודה לפי השם שלה.
  + הוספנו את הפונקציה GetColumn שמחזירה עמודה לפי המספר הסידורי שלה.
  + הוספנו את הפונקציה JoinBoard שמאפשרת ללקוח מסוים להצטרף ללוח.
  + הוספנו את הפונקציה LeaveBoard שמאפשרת ללקוח מסוים לעזוב את הלוח.
  + שינינו את הפונקציה ChangeOwner שתקבל פרמטרים של user ואת מחרוזת אימייל שלהם.
  + הוספנו את הפונקציה AssignTask שמאפשרת לשנות את מי שרשום למשימה מסוימת בלוח.
  + הוספנו את הפונקציה UpdateTaskDescription שמאפשרת לשנות את התיאור של אחת המשימה בלוח.
  + הוספנו את הפונקציה UpdateTaskDueDate שמאפשר לשנות את התאריך הסיום של אחת המשימות בלוח.
  + הוספנו את הפונקציה UpdateTaskTitle שמאפשר לשנות את הכותרת של משימה בלוח.
  + הוספנו את הפונקציה InProgressTasks שמחזירה את כל המשימות שבהכנה מתוך הלוח הזה.
* במחלקה BoardController:
  + שינינו את השם של השדה הפרטי ששומר את המספר הסידורים של הלוחות בפרויקט שלנו מהשם IDcounter לשם counterIDboards.
  + הוספנו את הפעולה GetBoardName שמחזירה את השם של הלוח לפי המספר הסידורי שלו.
  + הוספנו את הפעולה JoinBoard שמצרפת משתמש ללוח מסוים.
  + הוספנו את הפעולה LeaveBoard שמאפשר למשתמש לעזוב לוח מסוים.
  + שינינו את החתימה של הפעלוה RemoveBoard כך שהיא תקבל גם את האימייל שלי המשתמש שרוצה להסיר את הלוח.
  + הוספנו את הפעולה TransferOwnership שמעבירה בעלות על לוח ממשתמש מסוים לאחר.
  + הוספנו את הפעולה CheckBoardUser שמקבלת לוח ומשתמש ומחזירה משתנה בוליאני שמציין האם המשתמש קשור ללוח או לא.
  + הוספנו את הפעולה GetBoard שמחזירה לוח לפי השם שלו ומשתמש שרשום אליו.
  + הוספנו את הפעולה GetBoard שמחזירה לוח לפי המספר הסידורי שלו.
  + הוספנו את הפעולה IsBoardNameExists שמקבלת משתמש ושם של לוח ובודקת האם כבר קיים למשתמש לוח עם אותו השם.
  + הוספנו את הפעולה InProgressTask שמחזירה את כל המשימות שבהכנה מתוך כלל הלוחות עבור משתמש מסוים.
  + הוספנו את הפעולה MaxID שמחזירה את המספר המקסימאלי של בלוחות שיש כרגע במאגר נתונים שלנו.
  + הוספנו את הפעולה UBtoUser שמחזירה את המידע מהמאגר נתונים ומקשרת בין כל לוח ומשתמשים שלו, ובין כל משתמש ללוחות שלו.
* במחלקה Column:
  + הוספנו שדה פרטי ששומר את הטבלה של העמודות: columnDTO .
  + הוספנו את הפונקציה AssignTask שמאפשרת לשנות את מי שרשום למשימה מסוימת בעמודה הזאת.
  + הוספנו את הפונקציה UpdateTaskDescription שמאפשרת לשנות את התיאור של אחת המשימה בעמודה הזאת.
  + הוספנו את הפונקציה UpdateTaskDueDate שמאפשר לשנות את התאריך הסיום של אחת המשימות בעמודה הזאת.
  + הוספנו את הפונקציה UpdateTaskTitle שמאפשר לשנות את הכותרת של משימה בעמודה הזאת.
  + הוספנו את הפונקציה InProgressTasks שמחזירה את כל המשימות שבהכנה מתוך בעמודה הזאת.
  + הורדנו את הפונקציה AddTask.
* במחלקה Task:
  + שינינו את השדה assignee להיות EmailAssignee שמען ההבנה.
  + הוספנו שדה פרטי ששומר את המספר המזהה של הלוח BoardID
  + הגדרנו את השדות הפרטיים של המחלקה שאנחנו רוצים שיחזור בפורמט JSON ואלו אנחנו לא רוצים שיופיעו, לפי דרישות העבודה (במחלקה Grading Service).
  + הוספנו את הפונקציה IsAssignee שמחזירה ערך בוליאני האם המשתמש הוא הרשום במשימה הזאת או לא.
  + שינינו את השם של הפעולה AssignTask לשם ChangeAssignee.

שינויים ב- data access layer:

שינינו מחדש הכול.

* הוספנו את המחלקה BoardDTO שהיא אחראית על הטבלה של הלוחות.
  + השדות שלה: עמודה של שם הלוח, עמודה של מנהל הלוח, עמודה של מספר המשימות בלוח, השם של הלוח, בעל הלוח, המספר הסידורי של הלוח, והמספר הסידורי של המשימות בלוח.
  + המתודות שלה: שמירת ומחיקת מידע בטבלה ב- DataBase.
* הוספנו את המחלקה UserDTO שהיא אחראית על הטבלה של המשתמשים.
  + השדות שלה: עמודה של האימייל של המשתמש, עמודה של הסיסמא של המשתמש, האימייל של המשתמש, והסיסמא שלו.
  + המתודות שלה: שמירת ומחיקת מידע בטבלה ב- DataBase.
* הוספנו את המחלקה UserBoardDTO שהיא אחראית על הטבלה של הקשרים בין כל הלוחות לכל המשתמשים.
  + השדות שלה: עמודה של המשתמש, עמודה של הלוח, המספר הסידורי של הלוח, האימייל של המשתמש.
  + המתודות שלה: שמירת ומחיקת מידע בטבלה ב- DataBase.
* הוספנו את המחלקה ColumnDTO שהיא אחראית על הטבלה של הלוחות.
  + השדות שלה: עמודה של השם של העמודה, עמודה של ההגבלה של העמודה, עמודה של המספר הסידורי של הלוח, את המספר הסידורי של הלוח, השם של העמודה, וההגבלה שלה (על משימות).
  + המתודות שלה: שמירת ומחיקת מידע בטבלה ב- DataBase.
* הוספנו את המחלקה TaskDTO שהיא אחראית על הטבלה של הלוחות.
  + השדות שלה: עמודה של הכותרת של השימה, עמודה של התיאור של המשימה, עמודה של המשתמש המוקצה למשימה, עמודה של תאריך ייצור המשימה, עמודה של תאריך סיום המשימה, עמודה של העמודה שבה המשימה, עמודה של המספר המזהה של הלוח שבו המשימה, את המספר המזהה של המשימה, את תאריך הייצור שלה, את התאריך הסיום שלה, את הכותרת שלה, את התיאור שלה, את המשתמש שמוקצה לה, את שם העמודה שבה היא נמצאת, את המספר הסידורי של הלוח שבה היא נמצאת.
  + המתודות שלה: שמירת ומחיקת מידע בטבלה ב- DataBase.
* הוספנו את המחלקה DTO שהיא מחלקה אבסטרקטית. כל המחלקות שרשמתי למעלה מרחיבות אותה. למחלקה 2 שדות פרטיים: המספר הסידורי של העמודה לפי השם שלה, וה- mapper שהוא מחלקה אבסטרקטית.
* הוספנו את המחלקה האבסטרקטית AbstractMapper. היא אחראית על הקישוריות לטבלה.
  + יש לה 2 שדות פרטיים: היא שומרת את המחרוזת שמקשרת אל הטבלאות של המידע שלנו ב- database, והיא שומרת את השם של הטבלה שאנחנו רוצים לעבוד עליה.
  + יש לה את הפונקציות הבאות: עדכון המידע בטבלה, בחירת של מידע מתוך הטבלה, המרה של המידע בטבלה לאובייקט מסוג DTO, ומחיקה של מידע מהטבלה.
  + כל המחלקות שרשומות למטה מרחיבות אותה.
* הוספנו את המחלקה BoardMapper שהיא אחראית על הקישוריות לטבלה של הלוחות ב- SQL.
  + השדות שלה: השם של הטבלה.
  + המתודות שלה: הכנסת מידע לטבלה בשורה חדשה, והמרת המידע שרשומה ב-SQL לאובייקט מסוג DTO.
* הוספנו את המחלקה UserMapper שהיא אחראית על הקישוריות לטבלה של המשתמשים ב- SQL.
  + השדות שלה: השם של הטבלה.
  + המתודות שלה: הכנסת מידע לטבלה בשורה חדשה, והמרת המידע שרשומה ב-SQL לאובייקט מסוג DTO.
* הוספנו את המחלקה UserBoardMapper שהיא אחראית על הקישוריות לטבלה של הקשרים בין הלוחות למשתמשים במערכת ב- SQL.
  + השדות שלה: השם של הטבלה.
  + המתודות שלה: הכנסת מידע לטבלה בשורה חדשה, והמרת המידע שרשומה ב-SQL לאובייקט מסוג DTO, פונקציה המחזירה רשימה של קשרים בין משתמשים ללוחות בהינתן מספר סידורים של לוח.
* הוספנו את המחלקה ColumnMapper שהיא אחראית על הקישוריות לטבלה של העמודות ב- SQL.
  + השדות שלה: השם של הטבלה.
  + המתודות שלה: הכנסת מידע לטבלה בשורה חדשה, והמרת המידע שרשומה ב-SQL לאובייקט מסוג DTO, פונקציה המחזירה רשימה של עמודות של טבלה מסוימת, בהינתן המספר הסידורי של הטבלה.
* הוספנו את המחלקה TaskMapper שהיא אחראית על הקישוריות לטבלה של המשימות ב- SQL.
  + השדות שלה: השם של הטבלה.
  + המתודות שלה: הכנסת מידע לטבלה בשורה חדשה, והמרת המידע שרשומה ב-SQL לאובייקט מסוג DTO, פונקציה המחזירה רשימה של משימות של עמודה מסוימת מתוך לוח מסוים, בהינתן המספר הסידורי של הלוח ושל העמודה הנ"ל.

**25/6/2022**

שינויים ב- service layer:

* במחלקה BoardService:
  + הוספנו את הפונקציה GetBoardName שמחזירה את השם של הלוח כדי שנוכל להציג אותו בשכבה frontend.
* במחלקה ColumnService:
  + הוספנו פונקציה של GetTaskDescription שמחזירה את התיאור של כל המשימות שיש באותה העמודה.
  + הוספנו את הפונקציה GetColumn שמחזירה את רשימה של השמות של המשימות עבור עמודה מסוימת, כדי שנוכל להציג אותן בשכבה frontend.
  + הוספנו את הפונקציה GetTasksIDs שמחזירה את רשימה של המספרים הסידוריים של המשימות עבור עמודה מסוימת, כדי שנוכל להציג אותן בשכבה frontend.
* במחלקה UserService:
  + הוספנו פונקציה של GetUserBoard שמחזירה את השמות של כל הלוחות שיש לאותו הלקוח.
* במחלקה ResponseT:
  + שינינו את המחלקה שתהיה מטיפוס גנרי.
* שינויים ב- DAL ובשכבה Business layer:
  + שינויים לגבי טיפוס החזרת המידה ובדיקה שהוא לא null כלומר "" ונוכל להחזיר את המידע ולהציג אותו גם.
  + בנוסף תיקנו את השאילתה כך שיוחזר גם ה counter של המשימות.