

הצדקות לשינויים - Milestone 1

בתחילת העבודה על הDESIGN החלטנו להוסיף לUSER, בקר לוחות שינהל עבורו את הלוחות. לאחר שעות קבלה הבנו כי הוא מיותר במקום הזה כי הוא יכול לממש רק פונקציונליות של USER. לכן החלטנו להמיר את הBOARDCONTROLLER לאובייקט בעל משמעות שיחזיק בתוכו את כל הלוחות במערכת. הפונקציונליות של הBOARD שהייתה בUSER עברה לBOARD עצמו ולBOARDCONTROLLER. כך הגענו לCOHESION גבוה משום שכל אובייקט דואג לפונקציונליות שלו. כמו כן הCOUPLING נמוך דרסטית מהDESIGN הראשוני- זאת מכיוון שניתקנו את הקשר כמעט לחלוטין בין הפונקציונליות של הUSER לשל הBOARD.

ServiceLayer:

:BoardService

1. נוסף שדה UserController על מנת לאפשר גישה לפעולות מה-BusinessServer.
2. הועברו פעולות AddBoard, RemoveBoard מה-UserService, בעקבות השינוי שנעשה ב-BusinessServer, במסגרתו פעולות אלה הועברו ל-BusinessController.
3. GetTasksInProgress הועברה מה-UserService, בעקבות השינוי שנעשה ב-BusinessServer, במסגרתו פעולה זו הועברה ל-BusinessController.

:Task Service

boardController, UserController נוספו מכיוון והם מאפשרים גישה לBusiness Layer

:Service Factory

נוצר כדי לאחד בין שלושת המחלקות וגם כדי לייצר יחידות של controllerים עבור התוכנה.

:Grading Service

הפונקציות נוצרו כדי לאפשר תרגום של columnOrdinal לname.

:Response

מאפשר החזרת מידע על מקרים שונים שמתרחשים בקוד.

:Task To Send

נוצר כדי לאפשר תרגום נכון בהתאם לדרישות של אובייקט task לjson.

BusinessLayer:

:User

הפעולה login הועברה מהcontroller לכאן כי זה עוזר לcohesion שלהם- כי הuser דואג בעצמו לבצע login ולא controller דואג לזה, לכן זה גורם לcohesion גבוה.

:Column

הפונקציות נוספו לכאן כי הן דואגות לפונקציונליות של עמודה ומאפשרות לבצע שינויים בcolumn בפרויקט.

:BoardController

קיבל את כל הפונקציונליות שהייתה לUSER בנוגע לBOARD הוספת לוחות, מחיקת לוחות וקבלת ה"הטאסקים" של "יוזר" מסוים.

:Board

הפונקציה הממירה את המספר לשם עברה לGRADING SERVICE כי זה לא פונקציונלי ולא קשור למימוש.

התווספה פונקציונליות של הזזת ה"טאסק", ופונקציות החזרת הטאסטים בעמודות.