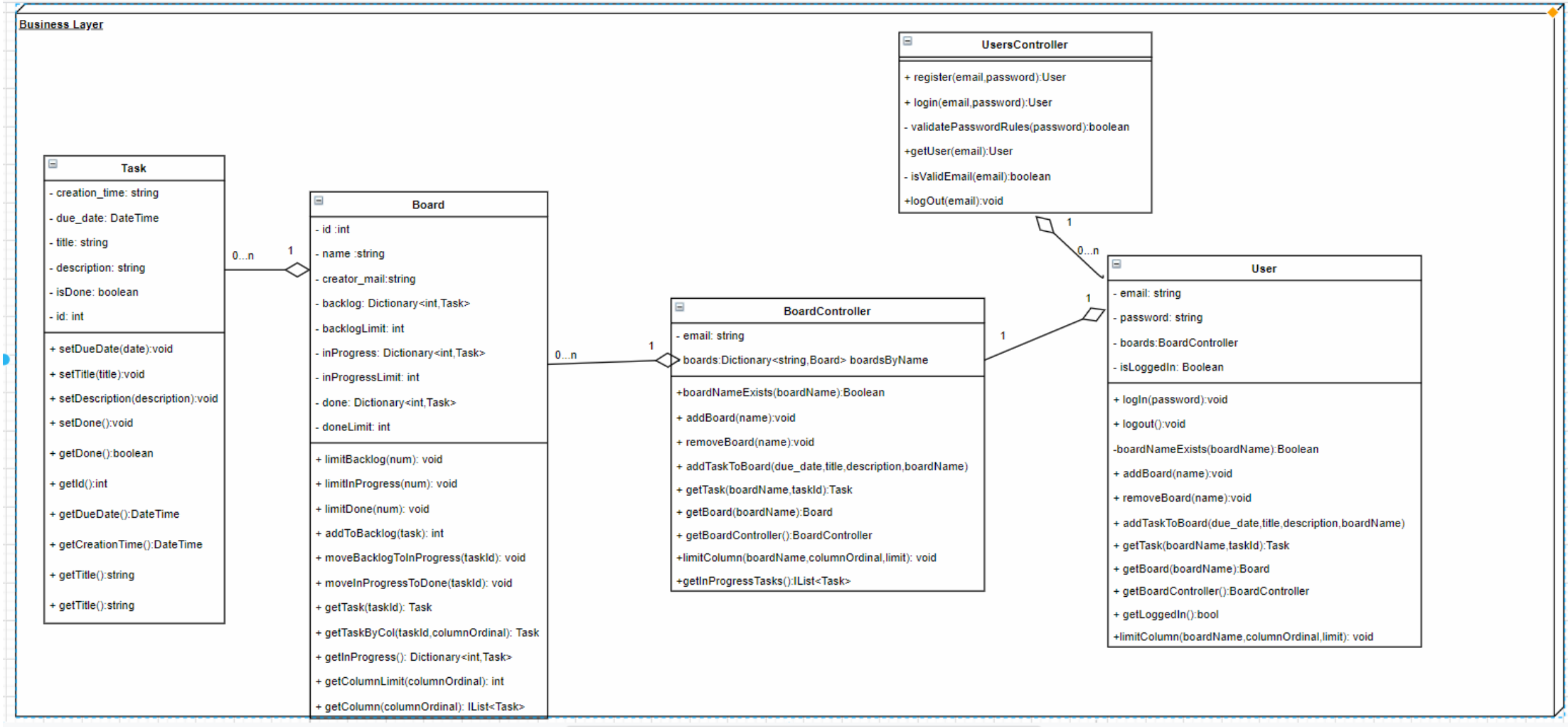
דיאגרמה מעודכנת-



<u>שינויים מהדיאגרמה המקורית:</u>

*המחלקה Task:

-השדות שמערבים תאריכים(תאריך ההגשה ותאריך היצירה) הפכו להיות מטיפוס DateTime כיוון שקיים טיפוס נוח ושימושי עבור תאריכים ב#c

-התווספו getters לכל השדות כיוון שנדרשנו לכך במהלך העבודה- למשל בשכבת הservice כדי ליצור Task מהטיפוס של שכבה זו.

*המחלקה Board:

-השדות של הcolumns הפכו מרשימה למילון כדי לאפשר גישה מהירה לכל Task בהינתן הId שלה.

- התווספו getters לכל השדות כיוון שנדרשנו לכך במהלך העבודה- למשל בשכבת הservice כדי ליצור IList של Tasks מהטיפוס של שכבה זו(זה שהוגדר על ידי סגל הקורס).

*המחלקה UserController:

-הוספנו מתודה פרטית לבדיקת חוקיות מייל(פונקציונאליות שהייתה חסרה בדיאגרמה המקורית). כמו כן- מתודת getUser נהפכה לפומבית עקב צורך בשכבת הService שעובדת מול האובייקט UserController וזקוקה למשתמש כדי לבצע חלק מהמשימות. כמו כן, הוספנו מתודה logout פומבית כדי לאפשר התנתקות בשכבת הService.

*המחלקה User:

-העברנו את "סמכות" הניהול של הBoards למחלקה חדשה בשם BoardController, לכן מילון המשימות עבר אל מחלקה זו. הסיבה לכך היא שאנו שואפים לייצר כמה שפחות תלות בין המשתמש ללוחותיו על מנת שבעתיד, במידה ונידרש לפתח פיצ'רים כגון שיתוף לוח בין משתמשים שונים, נוכל לנצל את המודולריות של ניהול הלוחות ולממש בקלות יחסית את אותם פיצ'רים חדשים.

-המתודה getBoardController התווספה בשל הצורך להפריד את ניהול הלוחות והמשתמשים גם בשכבת הservice ולאפשר גם לה להיות מודולרית ככל הניתן.

-המתודה getLoggedIn והשדה שהיא מחזירה התווספו כיוון שעקב ההפרדה בין המשתמש ללוחות נדרש לוודא בעזרתם האם המשתמש אכן מחובר כשהוא שואף לבצע שינוי בלוח כלשהו, מאוד קריטית לפונקציונאלית המערכת.

-הערה חשובה: כפי שניתן לראות, קיימת מעין חזרה על חלק מהמתודות שיופיעו בBoardController, מתודות אלה היוו חלק מהמימוש ההתחלתי שלנו, בטרם ההפרדה, לשכבת הביזנס. לא נעשה בהן שימוש בעבודה(עבדנו ישירות מול הBoardController הרלוונטי) אך בחרנו להשאיר אותן למקרה שנצטרך לבצע שינוי בעיצוב התוכנה בהמשך הפרויקט שיגרור שימוש בהן/הידוק התלות בין לוחות למשתמשים.

*המחלקה BoardController(לא הופיעה בדיאגרמה ההתחלתית):

-כפי שהוסבר קודם לכן, מחלקה זו אחראית לניהול הלוחות של משתמש מסוים, המתודות בה הוגדרו לאחר התחשבות במה שנדרש מאיתנו לממש במחלקה Service ומתוך ההבנה שכדאי לנו, כמתכנתים, לייצר כמה שפחות תלות בין המחלקות(בפרט- משתמש ולוחותיו) על מנת שהתוכנית תהיה מודולרית ונוחה לשינויים עתידיים.

-עבור כל מתודה בשכבת הService אותה נדרשנו לממש הגדרנו מתודה מתאימה במחלקה זו, ודאגנו שהBoardService בשכבת הService ידע לעבוד בכל עת מול הBoardController המתאים(מילון שמתאים בין מייל של משתמש לבורד קונטרולר הרלוונטי). באופן זה הקטנו גם בשכבת הService, עד כמה שניתן, את התלות בין משתמש ללוחותיו.