service

BoardService

- bc:boardController

+ LimitColumn(email: string, boardName: string, columnOr
+ GetColumn(email: string, boardName: string, columnOrd
+ RemoveBoard(email: string, BoardName: string): Json
+ AddBoard(email: string, name: string): Json

userService

- uc:userController

+ Register(email: string, password: strin
+ LogIn(email: string, password: string):
+ LogOut(email: string): Json
+ ValidateUserLoggin(email: string): Voi
+ getUser(email:string):json

Response

- ErrorMessage:string
- ErrorOccured:bool

~ Response(value:T,msg:string)

board

- name:String
-creatorEmail:string
-columns:Array[3] column

Response<T>

+ value :T

-Response<T>(value:T,n
~ FromValue(Value:T):R
~ FromError(msg:string):

TaskService

- bc : BoardController
-taskIdCounter : int

+ AdvanceTask(email: string, boardName: string, columnOrdinal: int, taskId: int): Js
+ UpdateTaskTitle(email: string, boardName: string, columnOrdinal: int, taskId: int, t
+ UpdateTaskDescription(email: string, boardName: string, columnOrdinal: int, task
+ UpdateTaskDueDate(email: string, boardName: string, columnOrdinal: int, taskId:
+removeTask(email:string, boardName:string,ColumnOrdinal:int,taskId:int):Json
+AddTask(email:string, boardName:string,title:string, decription:string, dueDate :Da
+ InProgressTasks(email: string): Json
getTasks(email : string, boardName : string, columnOrdinal :int) : Json

task

+ Id: int
+ CreationTime: DateTime
+ Title: string
+ Description: string
+ DueDate: DateTime

user

- email:String
- password:String

Business Layer

boardController

- Map<name,List<board>> boards
+ TASK_DESCRIPTION_MAX_LEN: int = 300
+ TASK_TITLE_MAX_LEN: int = 50

+ getBoard(boardName: string, email: string): Board
+ AddBoard(email: string, name: string): boolean
+ RemoveBoard(email: string, name: string): boolean

task

- creationTime: DateTime
- taskID: int
- title: string
- description: string
- dueDate: DateTime

+ UpdateTaskDescription(description: string): void
+ UpdateTaskDueDate(dueDate: DateTime): void
+ UpdateTaskTitle(title: string): void
+ getters and setters...

each email is unice but the id is for convinient

user

- email: string
- password: string
//-boards:Map<string:board
- isLogedIn:boolean

+ login(pass: string):boolea
+ logout(): void
+ validatePassword(pass: s

board

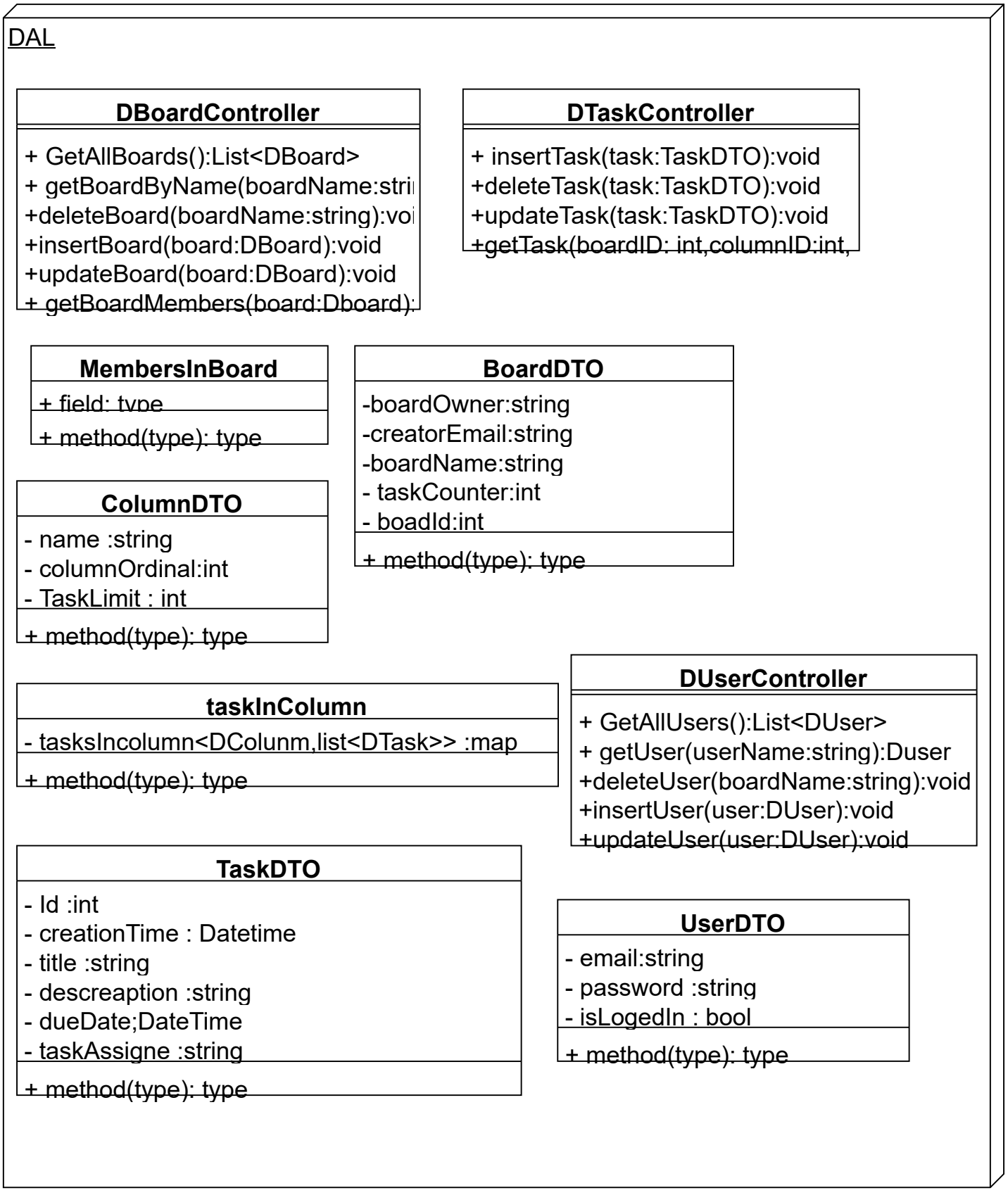
-cloumns:Array[3] column
- CreatorEmail: string
- Name: string

+ getColumnTasks(columnOrdinal: int): List<task>
+ removeBoard(boardName:string):boolean
+ inProgressTasks(): List<task>
+ getColumn(columnOrdinal: int): column
+ limitColumn(columnOrdinal: int, limit: int): void
+ getColumnName(columnOrdinal: int): string
+ advancedTask(columnOrdinal: int, taskId: int):boolean
+ getColumnLimit(columnOrdinal: int): int
+ endTask(title: string, description: string, dueDate: DateTime): task

Column

- ColumnOrdinal: int
- TaskLimit: int
- tasks:List<task>

+ getTask(int id) : task
+ addTask(task fields...):boolean
+ getTasks():List<task>


DAL
DBoardController
+ GetAllBoards():List<DBoard>
+ getBoardByName(boardName:stri
+deleteBoard(boardName:string):voi
+insertBoard(board:DBoard):void
+updateBoard(board:DBoard):void
+ getBoardMembers(board:Dboard):
DTaskController
+ insertTask(task:TaskDTO):void
+deleteTask(task:TaskDTO):void
+updateTask(task:TaskDTO):void
+getTask(boardID: int,columnID:int,
MembersInBoard
+ field: type
+ method(type): type
BoardDTO
-boardOwner:string
-creatorEmail:string
-boardName:string
- taskCounter:int
- boadId:int
+ method(type): type
ColumnDTO
- name :string
- columnOrdinal:int
- TaskLimit : int
+ method(type): type
taskInColumn
- tasksIncolumn<DColunm,list<DTask>> :map
+ method(type): type
DUserController
+ GetAllUsers():List<DUser>
+ getUser(userName:string):Duser
+deleteUser(boardName:string):void
+insertUser(user:DUser):void
+updateUser(user:DUser):void
TaskDTO
- Id :int
- creationTime : Datetime
- title :string
- descreaption :string
- dueDate;DateTime
- taskAssigne :string
+ method(type): type
UserDTO
- email:string
- password :string
- isLogedIn : bool
+ method(type): type