

Service layer

UserService
- userFacade :UserFacade
+ Register(email, password) : Response<UserSL> + Login(email , password) : Response<UserSL> + Logout(email) : Response<UserSL> + LoadData : Response<UserSL> #changes

TaskService
- boardFacade : BoardFacade
+ UpdateTaskDueDate(email, boardName, columnOrdinal ,taskId, dueDate) : string + UpdateTaskTitle(email, boardName, columnOrdinal ,taskId, title) : string + UpdateTaskDescription(email, boardName, columnOrdinal ,taskId, description): string

Response
+ ErrorMessage : string + ReturnValue : object
+ ToJson( ) : string

+ AdvanceTask(email, boardName, columnOrdinal ,taskId) : string  
+ InProgressTasks(email) : Response<TaskSL>  
+ AddTask(email, boardName, title, description, dueDate) : void #changes  
+ MoveTask(email, boardName, taskId) : void #changes  
+ AssignTask(email, boardName, taskId, assignee) : string #changes

#### UserSL

+ Email : String  
+ LoggedIn : bool

#### TaskSL

+ id : int  
-,+ creationTime : DateTime  
-,+ dueDate : DateTime  
-,+ Title : string  
-,+ description : string  
+ currPosition : string  
+ status: String  
+ isDone: bool  
+ boardName : string

#### BoardSL

+ BoardName : string  
+ backlogCapacity : int  
+ inProgressCapacity : int  
+ doneCapacity : int  
+ ID : int  
  
+ GetName( ) : string  
+ setCapacity (lim) : void  
+ getID :Int

#### BoardService

+ boardFacde : BoardFacade  
  
+ AddTask(email, boardName, title, description, dueDate) : string  
+ CreateBoard(email, boardName):Response<BoardSL>  
+ DeleteBoard (email, boardName) :Response<BoardSL>  
+ LimitColumn(email, boardName, columnOrdinal, limit) : string  
+ GetColumnLimit(email, boardName, columnOrdinal) : string  
+ GetColumnName(email, boardName, columnOrdinal) : string  
+ GetColumn(email, boardName, columnOrdinal) : Rseponse<TaskSL>

#### ServiceFactory

+ userFacade : UserFacade  
+ boardFacade : BoardFacade  
+ bs : BoardService  
+ us : UserService  
+ ts : TaskService  
+ authFacade : AuthenticationFacade

## Business layer

#### AuthenticationFacade

+ loggedInEmails : HashSet<string>  
  
+ IsLoggedIn(email) : bool  
+ Login(email) : void  
+ Logout(email) : void

#### UserFacade

+ users : dict<string,UserBL>  
- MaxEmailLength : int  
- MinPasswordLength : int  
- MaxPasswordLength : int  
+ authFacade : AuthenticationFacade  
- boardController: BoardController #changes  
-userController: UserController #changes  
-taskController:TaskController #changes  
-columnController: ColumnController #changes  
+ boardFacade: BoardFacade  
  
+ Register(email, password) : UserBL  
+ Login(email,password): UserBL  
+ Logout(email) : UserBL  
+ IsValidEmail(email) : bool

#### BoardFacade

+ Boards\_dict : dict<int,BoardBL>  
+ taskID : int  
+ authFacade : AuthenticationFacade  
+ Users : <String,UserBL>  
- boardController: BoardController #changes  
- taskController: TaskController #changes  
  
+ getBoard(nameBoard,BoardID) : BoardBL  
+ CreateBoard(OwnerEmail, boardName,BoardID) : BoardBL  
+ DeleteBoard(OwnerEmail:String, BoardID) : bool  
+ LimitColumn(email, BoardID, columnOrdinal, limit) : BoardBL  
+ GetColumnLimit(email, BoardID, columnOrdinal) : int  
+ GetColumnName(email, BoardID, columnOrdinal) : string  
+ AddTask( email, BoardID, title, description, dueDate) : TaskBL  
+ UpdateTaskDueDate(AssigneEmail, BoardID, columnOrdinal, taskId, dueDate) ; bool  
+ UpdateTaskTitle(AssigneEmail, BoardID, cloumnOrdinal, taskId, title) : ReturnThis  
+ UpdateTaskDescription(AssigneEmail, BoardID, cloumnOrdinal, taskId, description) : bool  
+ AdvanceTask(AssigneEmail, BoardID, cloumnOrdinal, taskId) : bool  
+ GetColumn(email, BoardID, cloumnOrdinal) : ReturnThis  
+InProgressTasks(AssigneEmail) : ReturnThis #changes  
+ SetOwner(email,OwnerEmail):bool #changes  
+ JoinBoard(email , OwmerEmail,BoardID): bool #changes  
+leaveBoard(email):bool #changes  
+AssignTask(Email, BoardID TaskID?) : bool #changes

#### ReturnThis

+ List : List<TaskBL>  
+ ErrorMsg : string

#### UserBL

- userDAL : UserDAL #changes  
- email : String  
- password : String  
- boards : list<BoardBL>  
- isLogedIn : bool  
- loginErrorMessage: string  
- logoutErrorMessage : string  
- registerErrorMessage : string

+ Isvalidpassword(password): bool  
+ Login(pass): bool  
+ Logout : void

#### BoardBL

+ BoardID : int  
+ BoardName : String  
+ backlog : Colmun  
+ in progress :Colmun  
+ done : Colmun



