



The changes we made in the Service Layer:

- In **UserService**: we added the functions- GetUserBoards(), loadData() and deleteData()
- In **TaskService**: we added the function AssignTask()- now each task can have an assigned user to it.
- In **BoardFacade**: we added the functions- JoinBoard(), LeaveBoard(), GetBoardName() and TransferOwnership(), LoadData(), and DeleteData()
- We added the class **WrapperFactory** which has a function create() that initializes all the service's layer classes and returns a new WrapperClass, and a function delete() that deleted the data all data.

The changes we made in the Business Layer:

- In **UserFacade**: we added the functions- GetUserBoards() (to get all boards that he is a member at), LoadData() and DeleteData()
- In **User**: we added the field udto: UserDTO, we also added the functions- GetUserBoards(), AddUserToBoard() and DeleteUserFromBoard()
- In **BoardFacade**: we added the functions- JoinBoard(), LeaveBoard(), GetBoardName(), TransferOwnership(), AssignTask(), LoadData(), copyInProgress() and deleteTask()
- In **Board**: we added the fields- bdto: BoardDTO (we connected the classes in the business layer to the appropriate classes in Data Access layer), boardID: int, ownerEmail: string, we also added the functions- GetBoardName(), AssignTask() and TransferOwnership()
- In **Task**: we added the fields- tdto: TaskDTO, emailAssignee: string, we also added the function assignTask()
- In **UserInProgressTasks**: we added the field: uipdto: UserInProgressDTO
- We added a class **TaskToSend** that when the service layer asks for tasks (for example: GetColumn()) it will return a task in a format that does not include all the unwanted fields.

Data Access Layer:

We created UserDTO and UserController, TaskDTO and TaskController, BoardDTO and BoardController. For each of them we added the matching fields and the functions that need to access the data base.