Presentation Layer

Service Layer

UserService

+ changePassword(username: string , oldP: string, newP: string): json

+ createUser(username: string, password: string): json

+ login(username: string, password: string): json

+ getBoard(BoardName: String): json

+ getInProgress(): json

+ logout(userEmail: string):json

Board

+ createBoard(Name: string, userEmail: string)

+ addTask(Name: string, userEmail: string,title: String, Description: String, DueDate: DateTime): json

+ nextState(email: string, boardName: string, TaskID: int): json

+ deleteBoard(Name: string, userEmail: string)

TaskService

+ EditTitle(email:string, boardName:string, taskId:int, newTitle:string): json

+ EditDueDate(email:string, boardName:srting, taskId:int, newDueDate:DateTime): json

+ EditDescription(email:stringl, boardName:string, taskId:int, newDescription:string): json

---

Visibility

+ public

- private

protected

Business Layer

UserController

- users: Dictionary<string, User>

- loggedIn: Collection

+ CreateUser(password: string, email: String): void

+ DeleteUser(email: string): void

+ UserExists(email: string): boolean

+ GetUser(email: string): User

+ validatePassword(user: User, password: String): boolean

+ login(email: String, password: String): void

+ logout(user: User): void

User

+ USERNAME: string

- password: string

- Boards: Collection

- TasksInProgress: Collection

+ ChangePassword(oldP: string, newP: string): void

+ ValidatePassword(password: string): bool

BoardBuissnes

- tasks(Dictionary<int,Task>

- InProgress(List<>)

+ Name(String)

- maxTasks(int [])

- numTasks(int[])

+ addTask(title: String, Description: String, DueDate: String): void

+ changeState(Task): void

+ ChangeState(int taskId):void

+ getters

+SetMaxTask(newMaxTask:int, whichBoard:int):void

+GEtColList(int columnO):List

+GetNameOrdinal(int coulumnO):string

+GetInProgress():List

+ setters

Task

- Title(String)

- DueDate(String)

+ CREATION DATE(DateTime)

- Description(newBody: string): void

- State(int)

+ TaskID(int)

+ EditTitle(newTitle: string): void

+ EditDueDate(newDate: String): void

+ EditDescription(newBody: string): void

+ Getters

+ Setters

BoardController

- Boards: Dictionary<string,Dictionary<string, Board>

+UserHasAnyBoard(userEmail:string):bool

+ createBoard(userEmail:string,boardName:string): void

+ UserHasThisBoard(userEmail:string, boardName:string):bool

+ deleteBoard(userEmail:string ,boardName: String): void

+ GetAllInPrograss(userEmail:string):List<Task>

+ getBoard(userEmail:string ,boardName: String): Board

Association

Inheritance

Realization / Implementation

Dependency

Aggregation

Composition

Aggregation: child can exist without the parent

Composition: child cannot exist without the parent

From Wikipedia

https://en.wikipedia.org/wiki/Class_diagram