Client - Presentation Layer

Service Layer

SystemService
- uf: UserFacade
- bf: BoardFacade
- userService: UserService
- boardService: BoardService
- taskService: TaskService
- log: ILog

+ Register(email: string, password: string): string
+ Login(email: string, password: string): string
+ Logout(email: string): string
+ LimitColumn(email: string, boardName:string, column: int, limit: int): string
+ GetColumnLimit(email: string, boardName:string,columnOrdinal: int): string
+ GetColumnName(email: string, boardName: string ,columnOrdinal: int): string
+ AddTask(email: string, boardName:string, title: string ,description: string, dueDate: DateTime): string
+ AddTask(email: string, boardName:string, title: string, dueDate: DateTime): string
+ UpdateTaskDueDate(email: string, boardName: string, columnOrdinal: int ,taskId:int,dueDate: DateTime): string
+ UpdateTaskTitle(email: string, boardName: string, columnOrdinal: int ,taskId:int, title: string): string
+ UpdateTaskTitle(email: string, boardName: string ,columnOrdinal: int, taskId:int, title: string): string
+ UpdateTaskDescription(email: string, boardName: string, columnOrdinal: int, taskId:int, description: string): string
+ AdvanceTask(email: string, boardName:string, title, taskId:int, column: int): string
+ GetColumn(email: string, boardName: string ,columnOrdinal: int): string
+ CreateBoard(email: string, name: string): string
+ DeleteBoard(email: string, name: string): string
+ InProgressTasks(email: string): string
+ GetUserBoards(email : string): string
+ JoinBoard(email :string , boardId :int) : string
+ LeaveBoard(email : string,boardId : int) : string
+ GetBoardName(boardId : int) : string
+ TransferOwnership(currentOwnerEmail : string, newOwnerEmail : string, boardName :string) : string
+ AssignTask(email : string,boardName :string, columnOrdinal :int ,taskID : int, emailAssignee : string) : string

TaskService
- bf: BoardFacade
- uf: UserFacade
- i:ILog

+ AddTask(email: string, boardName:string, title: string ,description: string, dueDate: DateTime): string
+ AddTask(email: string, boardName,title: string, dueDate: DateTime): string
+ AdvanceTask(email: string, boardName:string,title, taskId:int, column: int): string
+ UpdateTaskDueDate(email: string, boardName : string, columnOrdinal: int ,taskId:int,dueDate: DateTime): string
+ UpdateTaskDescription(email: string, boardName: string, columnOrdinal: int, taskId:int, description: string): string
+ UpdateTaskTitle(email: string, boardName: string, columnOrdinal: int, taskId:int, title: string): string
+ AssignTask(email : string,boardName :string, columnOrdinal :int ,taskID : int, emailAssignee : string) : string

BoardService
- bf: BoardFacade
- uf: UserFacade

+ CreateBoard(email: string, name: string): string
+ DeleteBoard(email: string, name: string): string
+ ShowInProgress(email: string): string
+ GetLimitOnColumn(email: string, boardName:string,column: int): string
+ SetLimitOnColumn(email: string, boardName:string,column: int, limit: int): string
+ AdvanceTask(email: string, boardName: string ,columnOrdinal: int,taskId: int): string
+ GetColumn(email: string, boardName: string ,columnOrdinal: int): string
+ GetUserBoards (email : string): string
+ JoinBoard (email :string , boardId :int) : string
+ LeaveBoard(email : string,boardId : int) : string
+ GetBoardName(boardId : int) : string
+ TransferOwnership(currentOwnerEmail : string, newOwnerEmail : string, boardName :string) : string

UserService
- uf: UserFacade
- log : ILog

+ Register(email: string, password: string, ...) : string
+ Login(email: string, password: string): string
+ Logout(email: string): string
+ Logout(email: string): string
+ CheckLoggedIn(email: string): string

Response
+ ReturnedValue: object
+ ErrorMessage: string

+ ResToString(): string
+ IsErrorOccured(): bool

```
JSON
{
  ErrorMessage: string,
  ReturnValue: string
}
```

→ 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

Visibility
+ public
- private
protected
~ internal

Business Layer

BoardFacade
- usersBoards: Dictionary<string, Dictionary<string, Board>>
- boards : Dictionary<int, Board>
- boardIdCounter : int
- taskIdCounter :int
- log: ILog

~ CreateBoard(email: string, name: string): void
~ DeleteBoard(email: string, name: string): void
~ ShowInProgress(email: string): ArrayList
~ GetBoard(email: string, boardName: string): Board
~ AddTask(email: string, boardName: string, title: string, dueDate: DateTime ,description: string): void
~ AddTask(email: string, boardName: string, title: string, dueDate: DateTime): void
~ AdvanceTask(email: string, boardName: string,column: int, taskId: int): void
~ SetLimitOnColumn(email: string, boardName: string, columnId: int, limit: int): void
~ GetLimitOnColumn(email: string, boardName: string, columnId: int): int
~ UpdateTaskDuedate(email: string, boardName: string, columnOrdinal: int, taskId: int, dueDate: DateTime):void
~ UpdateTaskDescription(email: string, boardName: string, columnOrdinal: int, taskId: int, description: string):void
~ UpdateTaskTitle(email: string, boardName: string, columnOrdinal: int, taskId: int, title: string):void
~ CheckUserHasABoard(email: string): void
~ CheckUserHasABoard(email: string): void
~ CheckBoardNameExists(email: string, boardName: string): void
~ CheckForExceptions(email: string, boardName: string): void
~ GetColumnName(email: string, boardName:string, column: int): string
~ GetColumn(email: string, boardName: string, columnOrdinal: int): List<Task>
~ GetUserBoards (email : string) : List<int>
~ JoinBoard(email :string , boardId :int) : void
~ LeaveBoard (email :string,boardId : int) : void
~ GetBoardName(boardId : int) : string
~ TransferOwnership(currentOwnerEmail : string, newOwnerEmail : string, boardName :string) : void
~ AssignTask(email : string,boardName :string, columnOrdinal :int ,taskID : int, emailAssignee : string) : void

UserFacade
- users: Dictionary<string, User>
- log : ILog

+GetUser(email: string): User
~ ChangePassword(email : string, oldPass: string, newPass: string): void
~ Register(email: string, password: string): void
~ CheckEmail(email: string): void
~ Login(email: string, password: string): string
~ Logout(email: string): void
~ CreateUser(email: string,password: string): void
~ CheckLoggedIn(email: string): void

User
- email: string
- password: string
- loggedin: bool
- MIN_PASSWORD_LENGTH: int (readonly)
- MAX_PASSWORD_LENGTH: int (readonly)
- log: ILog

~ Login(password: string): void
~ Logout(): void
~ IsLoggedIn(): bool
~ GetPassword(): void
~ ChangePassword(newPassword: string): void
~ CheckPassword(password: string): void

Board
- boardId : int
- boardName: string
- owner : string
- limitCol: int[]
- columnsNum: int
- columns: Dictionary<int, Task>[]
- NoLimit: int
- backlog: int
- inProgress: int
- done: int
- log: ILog

-CanAddTo(column: int): bool
~ AddTask(title: string, dueDate: DateTime ,description: string, taskId: int): void
~ AddTask(title: string, dueDate: DateTime , taskId: int): void
~ AdvanceTask(taskId: int, column: int): void
~ CheckIdExis(taskId: int, columnId: int): void
~ SetDescription(taskId:int ,columnId: int ,desc: string): void
~ SetTitle(taskId:int ,columnId: int ,title: string): void
~ SetDueDate(taskId:int ,columnId: int ,duedata: DateTime):void
~ SetLimitOnColumn(columnId: int, limit: int): void
~ GetLimitOnColumn(columnId: int): int
~ ShowInProgress(email : string):ArrayList
~ CheckValid(column: int): void
~ GetColumnName(column: int): string
~ GetColumn(column: int): List<Task>
~ GetId() : int
~ CheckOwner(email : string) : void
~ CheckMember(email : string) : void
~ UnassignTasks(email : string) : void
~ UnassignTasksInCol(email : string , column : int) : void
~ AddMember(email : string) : void
~ RemoveMember(email : string) : void
~ GetName() : string
~ GetOwner() : string
~ TransferOwnership(newOwnerEmail : string) : void
~ AssignTask(email : string,boardName :string, columnOrdinal :int ,taskID : int, emailAssignee : string) : void

Task
+ Title: string { get; set; }
+ Description: string || null { get; set; }
+ DueDate: DateTime { get; set; }
+Id: int { get; set; }
+CreationTime: DateTime { get; set; }
-assignee : string
- TITLE_LENGTH: int (readonly)
- DESCRIPTION_LENGTH: int (readonly)
- log: ILog

~ SetDescription(desc: string): void
~ SetDuedate(duedate: DateTime): void
~ SetTitle(title: string): void
~ CheckTitle(title: string): void
~ CheckDuedate(duedate: DateTime): void
~ CheckDescription(desc: string): void
~ CheckForExceptions(title: string, duedate: DateTime, desc: string): void
+ ToString(): string
~ GetAssignee() : string
~ SetAssignee(email : string) : string
~ checkAssignee(email : string) : void
~ Assign(email : string, emailAssignee : string) : void
~ IsAssigned(email : string) : bool

Changes in our design:

Added to the BoardFacade a new Dictionary that contains the boards with the boardId as key

Added to the BoardFacade a new boardcounter: int

Added to the Board field an owner : string field

Added to the Board field of members : List<string>

Added to the Board field a boardId : int

Added to the Task fields: assignee : string field

We also added to our design the following methods to support the new requirements:

isAssigned() in Task

CheckAssignee() in Task

CheckOwner() in Board

GetUserBoards() in BoardFacade

JoinBoard() in BoardFacade

LeaveBoard() in BoardFacade

UnassignTasks() in Board

UnassignTasksInCol() in Board

AssignTask() in BoardFacade,Board,Task

TransferOwnership()

GetName() in Board

GetOwner() in Board