Service Layer

+ TaskService

- bf: BoardFacade
- Ilog: log
- ~ TaskService(boardFacade: BoardFacade): TaskService
- + AddTask(email: string,boardName: string, title: string,description: 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
- + UpdateTaskDescription(email: string,boardName: string, columnOrdinal: int, taskId: int, description: string): string
- +AdvanceTask(email: string, boardName: string, colmnOrdinal: int, taskId: int): string

+ BoardService

- ~ bf: boardFacade <<readonly>>
- Ilog: log
- ~ BoardService(userFacade: UserFacade): BoardFacade
- + CreateBoard(email: string, name: string): string
- + DeleteBoard(string email, string name): string
- + LimitColumn(email: string, boardName: string, columnOrdinal: int, limit: int): string
- + GetColumnLimit(email: string, boardName: string, colmnOrdinal: int): string
- + GetColumnName(email: string, boardName: string, columnOrdinal: int): string
- + GetColumn(email: string, boardName: string, columnOrdinal: int): string
- + DeleteTask(email: string, boardName: string, columnOrdinal: int, taskId: int): string

+ UserService

- ~ uf : userFacade <<readonly>>
- Ilog: log
- ~ UserService(): UserService
- + Register(email: string, password: string): string
- + Login(email: string, password: string): string
- + Logout(email: string): string
- + ChangePassword(email: string, oldPassword: string, newPassword: string): string
- + InProgressTasks(email: string): string

+ Response

- + ErrorMessage: string
- + ReturnValue: Object

+ TaskToSend

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

+ WrapperService

- + us: userService
- + bs: boardService
- + ts: taskService
- ~ Ilog: log <<static>> <<readonly>>

~ BoardFacade

- userFacade: UserFacade
- + CreateBoard(email: string, name: string): void
- + DeleteBoard(string email, string name): void
- + LimitColumn(email: string, boardName: string, columnOrdinal: int, limit: int): void
- + GetColumnLimit(email: string, boardName: string, columnOrdinal: int): int
- + GetColumnName(email: string, boardName: string, columnOrdinal: int): string
- + GetColumn(email: string, boardName: string, columnOrdinal: int): List<Task>
- + AddTask(email: string,boardName: string, title: string,description: string, dueDate: DateTime): void
- + UpdateTaskDueDate(email: string, boardName: string, columnOrdinal: int, taskId: int, dueDate: DateTime): void
- + UpdateTaskTitle(email: string,boardName: string, columnOrdinal: int, taskId: int, title: string): void
- + UpdateTaskDescription(email: string,boardName: string, columnOrdinal: int, taskId: int, description: string): void
- + DeleteTask(email: string, boardName: string, columnOrdinal: int, taskId: int): void
- + AdvanceTask(email: string, boardName: string, columnOrdinal: int, taskId: int): void

~ UserFacade

- users: Dictionary<string,User>
- IN PROGRESS COLUMN ORDINAL: int
- + Register(email: string, password: string): User
- + Login(email: string, password: string): User
- + Logout(email: string): void
- + ChangePassword(email: string, oldPassword: string, newPassword: string): void
- + IsLoggedIn(email: string): bool
- + IsLoggedIn(email: string): bool
- + GetUser(email): User
- + GetBoardsNames(email): List<string>

Business Layer

~ User

- a Char Value: int <<const>>
- A_Char_Value: int <<const>>
- A_Char_Value: int <<const>>
- one_Char_Value: int <<const>>
- NUMBERS: int <<const>>
- LETTERS: int <<const>>
- boards: Dictionary<string, Board>
- email: string
- password: string
- loggedIn: bool
- + AddBoard(boardName:string): void
- + RemoveBoard(boardName:string): void
- + GetBoard(boardName:string): Board
- + GetTasks(columnOrdinal: int): List<Task>
- + ChangePassword(password: string): void
- CheckPassword(): void
- + IsLoggedIn(): bool
- + Login(password: string): bool
- + Logout(): void
- + Authenticate(password: string): bool
- + GetBoardsNames(): List<string>

~ Board

- idCounter: int
- columns count: int
- columns: List<Columns>
- DEFAULT COLUMNS: string[] << readonly>>
- LIMITLESS_VALUE: int <<const>>
- DEFAULT_LIMIT: int <<const>>
- + AddTask(title:string, description:string, DueDate:DateTime): void
- + LimitColumn(columnOrdinal: int, limit: int): void
- + GetColumnName(columnOrdinal: int): string
- + GetColumnLimit(columnOrdinal: int): int
- + GetColumn(columnOrdinal: int): List<Task>
- + UpdateTaskDueDate(columnOrdinal: int, taskId:int, dueDate: DateTime): void
- + UpdateTaskDueDate(columnOrdinal: int, taskId:int, title: string): void
- + UpdateTaskTitle(columnOrdinal: int, taskId: int, title: string): void
- + UpdateTaskDescription(columnOrdinal: int, taskId: int, description: string): void
- + DeleteTask(columnOrdinal: int, taskId: int): void
- + AdvanceTask(columnOrdinal: int, taskId: int): void

~ Task

- Id: int
- CreationTime: DateTime
- Title: string
- Description: string
- DueDate: DateTime
- MAX_TITLE_LENGTH: int <<const>>
- MAX_DESCROPTION_LENGTH: int <<const>>
- + UpdateTaskDueDate(DateTime DueDate): void
- + UpdateTaskTitle(string title): void
- + UpdateTaskDescription(string description): void

~ Column

- + Tasks: List<Task>
- + ColumnName: string
- + Limit: int