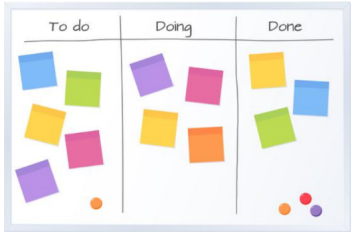


Presentation Layer



Service Layer

BoardService
- log: ILog - boardController: BoardController - userController: UserController - jsonConverter: JsonConverter
+ LoadData(): void + CreateBoard(email: string, boardName: string): json + DeleteBoard(email: string , boardName: string): json + GetColumn(email: string, boardName: string, columnOrdinal: int): json + GetColumnName(email: string, boardName: string, columnOrdinal: int): json + GetColumnLimit(email: string, boardName: string, columnOrdinal: int): json + LimitColumn(email: string, boardName: string, columnOrdinal: int, limit: int): json + AddUser(email: string, boardID: int): json + RemoveUser(email: string, boardID: int): json + SetOwner(currentOwnerEmail: string, newOwnerEmail: string, boardName: string): json

JsonConverter
- properties : JsonSerializerOptions
+ ToSerialize<T>(element: T): json + ToDeSeralize<T>(json: string): T

Response
+ ErrorMessage: string + ReturnedValue: <T>

ResponseInt
+ ReturnedValue: int

TaskService
- log: ILog - boardController: BoardController - userController: UserController - jsonConverter: JsonConverter
+ AddTask(email: string, boardName: string, title: string, description: string, dueDate: DateTime): json + MoveTask(email: string, boardName: string, columnOrdinal: int, taskID: int): json + UpdateTaskTitle(email: string, boardName: string, columnOrdinal: int, taskID: int, newTitle: string): json + UpdateTaskDescription(email: string, boardName: string, columnOrdinal: int, taskID: int, newDescription: string): json + UpdateTaskDueDate(email: string, boardName: string, columnOrdinal: int, taskID: int, newDueDate: DateTime): json + AssignTask(email: string, boardName: string, columnOrdinal: int, taskID: int, emailAssignee: string): json

UserService
- userController: UserController - jsonConverter: JsonConverter - log: ILog
+ LoadData(): void + Register(email: string, password: string): json + Login(email: string, password: string): json + Logout(email: string): json + InProgressTasks(email: string): json + GetUserBoardsID(email: string): json + GetUserBoards(email: string): json

ServiceFactory
- log: ILog - boardController: BoardController - userController: UserController + jsonConverter: JsonConverter + userService: UserService + boardService: BoardService + taskService: TaskService
+ LoadData(): json + DeleteData(): json

Business Layer

BoardController
- _boardId: int - boards: Dictionary<int, Board> - boardMapper: BoardMapper - boardIdCounterMapper: BoardIdCounerMapper - taskMapper: TaskMapper - log: ILog
+ LoadAllBoards(): void - LoadBoard(boardID: int, boardName: string, _users: LinkedList<string>, _bIList: LinkedList<Task>, _ipList: LinkedList<Task>, _doneList: LinkedList<Task>, _bILimit: int, _ipLimit: int, _doneLimit: int, ownerEmail: string, taskCouner: int): void + CreateBoard(boardName: string, email: string): void + DeleteBoard(boardName: string, email: string): void - GetBoard(boardName: string, email: string): Board - GetBoard(id: int): Board + JoinBoard(boardID: int, email: string): void + LeaveBoard(boardID: int, email: string): void + GetColumn(email: string, boardName: string, columnOrdinal: int): LinkedList<Task> + GetColumnName(email: string, boardName: string, columnOrdinal:int): string + GetColumnLimit(email: string, boardName: string, columnOrdinal: int): int + LimitColumn(email: string, boardName: string, columnOrdinal: int, limit: int): void + SetOwner(currentOwnerEmail: string, newOwnerEmail: string, boardName: string): void + AddTask(email: string, boardName: string, title: string, description: string, dueDate: DateTime): void + MoveTask(email: string, boardName: string, columnOrdinal: int, taskID: int): void + EditTaskTitle(email: string, boardName: string, columnOrdinal: int, taskId: int, newTitle: string): void + EditTaskDescription(email: string, boardName: string, columnOrdinal: int, taskId: int, newDescription: string): void + EditTaskDueDate(email: string, boardName: string, columnOrdinal: int, taskId: int, newDueDate: DateTime): void + AssignTask(email: string, boardName: string, columnOrdinal: int, taskId: int, emailAssignee: string): void + DeleteData(): void

UserController
- users: Dictionary<string, User> - log: ILog - boardController: BoardController - userMapper: UserMapper
+ LoadAllUsers(): void - LoadUser(email: string, password: string): void + CreateUser(email: string, password: string): void - CheckEmail(email: string): void + Login(email: string, password: string): void + Logout(email: string): void + IsLoggedInUser(email: string): void + GetUser(email: string): User + GetUserBoardsID(email: string): LinkedList<int> + GetUserBoardsID(email: string): LinkedList<Board> + GetInProgress(email: string): LinkedList<Task> + DeleteData(): void

User
- boardController: BoardController + Email: string - password: string + IsLoggedIn: bool - MAX_PASS_LENGTH: int - MIN_PASS_LENGTH: int
+ Login(password: string): void + Logout(): void + CheckPassword(password: string): bool + GetUserBoardsID(): LinkedList<int> + GetUserBoards(email: string): LinkedList<Board> + GetInProgressTask(): LinkedList<Task>

Board
- id: int - taskIdCounter: int + Name: string - owner: string - users: LinkedList<string> - BIList: LinkedList<Task> - IpList: LinkedList<Task> - DoneList: LinkedList<Task> - BILimit: int - IpLimit: int - DoneLimit: int - BACKLOG_COLUMN_ORDINAL: int - INPROGRESS_COLUMN_ORDINAL: int - DONE_COLUMN_ORDINAL: int - boardMapper: BoardMapper
+ GetColumn(columnOrdinal: int, email: string): LinkedList<Task> + GetColumnName(columnOrdinal: int, email: string): string + GetColumnLimit(columnOrdinal: int, email: string): int + EditLimit(columnOrdinal: int, limit: int, email: string): void + AddUser(userEmail: string): void + RemoveUser(userEmail: string): void + SetOwner(currentOwner: string, newEmail: string): void + ContainsUser(userEmail: string): bool + AddTask(dueDate: DateTime, title: string, description: string, email: string, taskMapper: TaskMapper): void + GetTask(taskID: int, columnOrdinal: int): Task + GetTask(taskID: int, taskList: LinkedList<Task>): Task + MoveTask(columnOrdinal: int, taskID: int, email: string): void + AssignTask(email: string, emailAssignee: string, columnOrdinal: int, taskID: int): void + EditTaskTitle(email: string, newTitle: string, columnOrdinal: int, taskID: int): void + EditTaskDescription(email: string, newDescription: string, columnOrdinal: int, taskID: int): void + EditTaskDueDate(email: string, newDate Time: DateTime, columnOrdinal: int, taskID: int): void - CheckLegalColumnEdit(columnOrdinal: int): void

Task
- id: int - creationTime: DateTime - title: string - description: string - dueDate: DateTime - assignee: string - TITLE_LENGTH: int - DESCRIPTION_LENGTH: int - taskMapper: TaskMapper - log: ILog
+ InsertTask(boardID: int): void + AssignTask(boardID: int, emailAssignee: string): void + EditTaskTitle(email: string, newTitle: string, columnOrdinal: int): void + EditTaskDescription(email: string, newDescription: string, columnOrdinal: int): void + EditTaskDueDate(email: string, newDueDate: DateTime, columnOrdinal: int): void + MoveTask(boardID: int, columnOrdinal: int): void

Data-Access Layer

BoardDTO
+ IdColumnName: string - _id: int + NameColumnName: string - _name: string + BILimitColumnName: string - bILimit: int + IpLimitColumnName: string - _ipLimit: int + DoneLimitColumnName: string - _doneLimit: int + OwnerEmailColumnName: string - _ownerEmail: string + BoardIdTaskCounterColumnName: string - _boardIdTaskCounter: int

TaskDTO
+ IdColumnName: string - _id: int + BoardIdColumnName: string - _boardId: int + CreationTimeColumnName: string - _creationTime: DateTime + TitleColumnName: string - _title : string + DescriptionColumnName: string - _description: string + DueDateColumnName: string - _dueDate: DateTime + AssigneeColumnName: string - _assignee: string + ColumnPlaceColumnName: string - _column : int

BoardMapper
- TableName: string - boardUserMapper: BoardUserMapper
+ SelectAllBoards(): List<BoardDTO> + SelectAllUsers(id: int): LinkedList<string> # ConvertReaderToObject(reader: SQLiteDataReader): DTO + InsertBoard(board: BoardDTO): bool + InsertUser(email: string, id: int): bool + DeleteUser(email: string, id: int): bool + DeleteBoard(id: int): bool + DeleteBoardUsers(): bool

TaskMapper
+ TableName : string
+ SelectAllTasks(): List<TaskDTO> + SelectAllTasks(id: int): List<TaskDTO> + SelectAllAssigneeTasks(id: int, email: string): List<TaskDTO> # SelectTasksByAssignee(id: int, email: string): List<DTO> # SelectTasks(id: int): List<DTO> # ConvertReaderToObject(reader: SQLiteDataReader): DTO + InsertTask(task: TaskDTO): bool + UpdateTasksAssignee(id: int, email: string, assignee: string): bool + DeleteTasksByBoardID(id: long): bool

BoardUserMapper
- TableName: string
+ SelectAllBoardUsers(): List<BoardUserDTO> + SelectAllBoardUsers(id: int): LinkedList<BoardUserDTO> # ConvertReaderToObject(reader: SQLiteDataReader): DTO + InsertBoardUSer(boardUser: BoardUserDTO): bool

Mapper*
- _connectionString: string - _tableName: string - _log: ILog
# Select (id: int) :List<DTO> # Select (): List<DTO> # ConvertReaderToObject(reader: SQLiteDataReader): DTO + Update(id: long, attributeName: string, attributeValue: string): bool + Update(emailId: string, attributeName: string, attributeValue: string): bool + Update(id: long, attributeName: string, attributeValue: long): bool + Update(emailId: string, attributeName: string, attributeValue: long): bool + Update(boardId: long, id: long, attributeName: string, attributeValue: string): bool + Update(boardId: long, id: long, attributeName: string, attributeValue: DateTime): bool + Update(boardId: long, id: long, attributeName: string, attributeValue: long): bool + Delete(id: long): bool + Delete(id: long, boardId: long): bool + Delete(emailId: string, boardId: long): bool + DeleteData(): bool

BoardIdCounterMapper
+ TableName : string
+ SelectBoardCounterId(): BoardIdCounterDTO # ConvertReaderToObject(reader: SQLiteDataReader): DTO + InsertBoardIdCounter(idCounter: BoardIdCounterDTO): bool

UserMapper
+ TableName : string
+ SelectAllUsers(): List<UserDTO> # ConvertReaderToObject(reader: SQLiteDataReader): DTO + InsertUser(user: List UserDTO): bool

DTO
+ _mapper : DalMapper

BoardUserDTO
+ EmailColumnName: string - _email: string + IdColumnName: string - _Id: int

UserDTO
+ EmailColumnName: string - _email: string + PasswordColumnName: string - _password: string

BoardIdCounterDTO
+ NextIDColumnName: string - _nextID: int