

Service Layer

UserService

- um UserManager;
- log ILogger;

+ createNewUser(String email, String password) : json

+ login(String email, String password): json

+ logout(String email) : json

+ isLoggedIn(String email) : json

ResponseInt

+ ReturnValue int;

BoardService

- bm BoardManagement;
- log ILogger;

+ addBoard(string toAddTitle, string userEmail): json

+ removeBoard(string userEmail, string toRemove) : json

+ addTask(string userEmail, string boardName, string title, string description, DateTime dueDate) : json

+ moveTask(string userEmail, string boardName, int columnOrdinal, int taskId): json

+ editTask(string userEmail, string boardName, int columnOrdinal, int taskId, string? newTitle, string? newDesc, DateTime? newDueDate, int numAct) : json

+ setCapacity(string userEmail,string boardName,int limit, int columnIndex) : json

+ getCapacity(string userEmail, string boardName,int columnIndex): json

+ getColumnName(string userEmail, string boardName, int columnIndex): json

+ ListProgressTasks(string client) : json

+ GetColumn(string email, string boardName, int columnOrdinal) : json

+ GetUserBoards(string email) : json

+ JoinBoard(string userEmail, int boardId) : json

+ LeaveBoard(string userEmail, int boardId) : json

+ AssignTask(string email, string boardName, int columnOrdinal, int taskId, string emailAssignee) : json

+ GetBoardName(int boardId) : json

+ TransferOwnership(string currentOwnerEmail, string newOwnerEmail, string boardName) : json

+ LoadData() : json

+ RemoveData() : json

Response<T>

+ ErrorMessage string?;
+ ReturnValue T?;

+ Response(string? errorMessage, T? returnValue): Response

JsonController

+ js JsonSerializerOptions;

+ Serialize<T>(T o): string

+ DeSerialize<T> (string os) : T

UserManagment
- _users List<User>; - log ILog; - userDalController UserDalController; - emailCheck EmailAddressAttribute;
+ GetUser(string email): User
+ createNewUser(string email, string password):bool
+ logIn (string email, string password): bool
+ logOut (string email): bool
+ exist (string email): bool
+ checks(string email): void
+ isValidEmail (string email): bool
+ passwordCheck (string password): bool
+ isLoggedIn (string email): bool
+ LoadData(): string
+ RemoveData(): string

BoardManagement
- boardList List<Board>; - um UserManagement; - boardIdCounter int; - boardDalController BoardDalController; - userBoardDalController UserBoardDalController; - taskDalController TaskDalController; - boardIdCounterDalController BoardIdCounterDalController; - taskCounterDalController TaskCounterDalController; - log ILog;
+ getBoard(User user, string boardName) : Board
+ getBoard(int id) : Board
+ getBoardName(int boardId) : string
+ getColumnName(string userEmail, string boardName, int column) : string
+ getCapacity(string userEmail, string boardName, int columnIndex) : int
+ GetColumn(string email, string boardName, int columnOrdinal): List<Task>
+ setCapacity(string userEmail, string boardName, int limit, int columnIndex): void
+ addBoard(string toAddTitle, string userEmail) : void
+ boardCheck(string boardName) : bool
+ removeBoard(string boardNameToRemove, string userEmail) : void
+ boardExists(string boardName, string userEmail) : bool
+ addTask(string userEmail, string boardName, string title, string description, DateTime dueDate) : Task
+ moveTask(string userEmail, string boardName, int columnOrdinal, int taskId) : void
+ editTask(string userEmail, string boardName, int columnOrdinal, int taskId, string? newTitle, string? newDesc, DateTime? newDueDate, int numAct): void
+ listProgressTasks(string userEmail) : List<Task>
+ JoinBoard(string email, int boardId) : bool
+ LeaveBoard(string email, int boardId) : bool
+ transferOwner(string owner, string newOwner, string boardName) : bool
+ assignTask(string email, string boardName, int columnOrdinal, int taskId, string emailAssignee) : void
+ GetUserBoards(string email) : List<int>
+ LoadData() : string
+ RemoveData() : string

Board
+ boardName string; - backlogList List<Task>; - inProgressList List<Task>; - doneList List<Task>; - owner string; - capacities int[]; - taskIdCounter int; - boardId int; - collaborators List<string>; - log ILog;
+ getCapacity(int columnIndex): int
+ getColumn(int columnOrdinal): List<Task>
+ listInProgressTask(string email): List<Task>
+ getTask(int taskId): Task
+ getTask(int columnOrdinal, int taskId): Task
+ setCapacity(int limit, int columnIndex, BoardDalController boardDalController) : void
+ addTask(string title, string desc, DateTime dueDate, TaskDalController taskDalController) : Task
+ moveTask(int columnOrdinal, int taskId, string email, TaskDalController taskDalController): void
+ editTask(int columnOrdinal, int taskId, string? newTitle, string? newDesc, DateTime? newDueDate, int numAct, string email, TaskDalController taskDalController) : void
+ isMember(string email) : bool
+ addMember(string email) : void
+ removeMember(string email, TaskDalController taskDalController) : bool
+ assignTask(string assignee, string newAssignee, int taskId, int columnOrdinal) : void
+ transferOwner(string newOwner, BoardDalController boardDalController, UserBoardDalController userBoardDalController) : bool

User
+ email string; - password string; - loggedIn bool; - log ILog;
+ login(): void
+ logout(): void
+ passwordMatch (string pass): bool

Task
-title string; -description string; -creationTime DateTime; -dueDate DateTime; -columnIndex int; -taskId int; - assignee string; - boardID int; - log ILog;
+ setIndex(int columnIndex, TaskDalController taskDalController): bool
+ editTitle(string newTitle, TaskDalController taskDalController) : void
+ editDescription(string newDesc, TaskDalController taskDalController) : void
+ editDueDate(DateTime newDueDate, TaskDalController taskDalController): void

Data Access Layer

TaskCounterDalController
-log ILog;
+ Insert(TaskIdCounterDTO taskIdCounterDTO): bool
+ ConvertReaderToObject(SQLiteDataReader reader): DTO

TaskDalController
- log ILog;
+ Insert(TaskDTO task): bool
+ ConvertReaderToObject(SQLiteDataReader reader): DTO

BoardIDCounterDalController
- log ILog;
+ Insert(BoardIdCounterDTO counter): bool
+ ConvertReaderToObject(SQLiteDataReader reader): DTO

BoardDalController
- log ILog;
+ Insert(BoardDTO board): bool
+ ConvertReaderToObject(SQLiteDataReader reader): DTO

UserDalController
- UserBoardName string; - log ILog;
+ Insert(UserDTO user): bool
+ ConvertReaderToObject(SQLiteDataReader reader): DTO

UserBoardDalController
-userBoardTableName string; - log ILog;
+ Insert(UserBoardDTO ub): bool
+ ConvertReaderToObject(SQLiteDataReader reader): DTO

DTO
- _controller DalController;

TaskIdCounterDTO
-counter int; -boardId int;

UserDTO
-email string; -password string;

DalController
- log ILog;
+ Update(long id, string attributeName, string attributeValue, string kind): bool
+ Update(long id, string attributeName, long attributeValue, string kind): bool
+ Update(string id, string attributeName, string attributeValue, string kind): bool
+ Update(int id1, int id2, string attributeName, long attributeValue): bool
+ Update(int id1, int id2, string attributeName, string attributeValue): bool
+ Update(int id1, int id2, string attributeName, DateTime attributeValue): bool
+ Update(string id1, int id2, string attributeName, string attributeValue): bool
+ Update(string attributeName, int attributeValue): bool
+ Select(): List<DTO>
+ Select(string assignee,int boardId): List<DTO>
+ ConvertReaderToObject(SQLiteDataReader reader): DTO
+ Delete(long boardId): bool
+ Delete(string email): bool
+ Delete(long boardId, long taskId): bool
+ Delete(string email, long boardId): bool
+ Delete() : bool

BoardIdCounterDTO
- counter int;

UserBoardDTO
- email string; - boardId int; - isOwner string;

BoardDTO
-boardName string; -owner string; -boardId int; -backLogCapacity int; -inProgressCapacity int; -doneCapacity int;

TaskDTO
-taskId int; -title string; -desc string; -creationTime DateTime; -dueDate DateTime; -boardId int; -columnIndex int; -assignee string;

Design changes:

- Each task now has a unique id in the board.
- Every task has a person assigned to it- an assignee. Once this person leaves a board, his assigned tasks that are not done become unassigned. As a result we added the method assignTask in BoardManagement and the method assign in Task.
- Each board has now a unique id in the system.
- Each user can join and leave existing boards created by someone else, without getting permission. As a result, we added the methods joinBoard and leaveBoard in BoardManagement and in Board.
- Each board owner can transfer the board ownership to another board member. As a result, we added the method transferOwner in BoardManagement and in Board.
- Added DataAccessLayer with their respective DataTransferObjects, responsible for translating objects and variables of our project into objects suitable to be added as entries into our database.
- Added database kanban.db.
- Added DataAccessLayer Controllers for each of the DataTransferObjects, responsible for communicating with the database and manipulating it (adding entries, removing entries).
- Added loadData, responsible for loading missing information from the database into our project, method is found in BoardManagement and UserManagement.
- Added removeData method, responsible for removing entries from the tables of our database and clearing our project (clearing lists, resetting ids etc.), found in BoardManagement and Usermanagement.