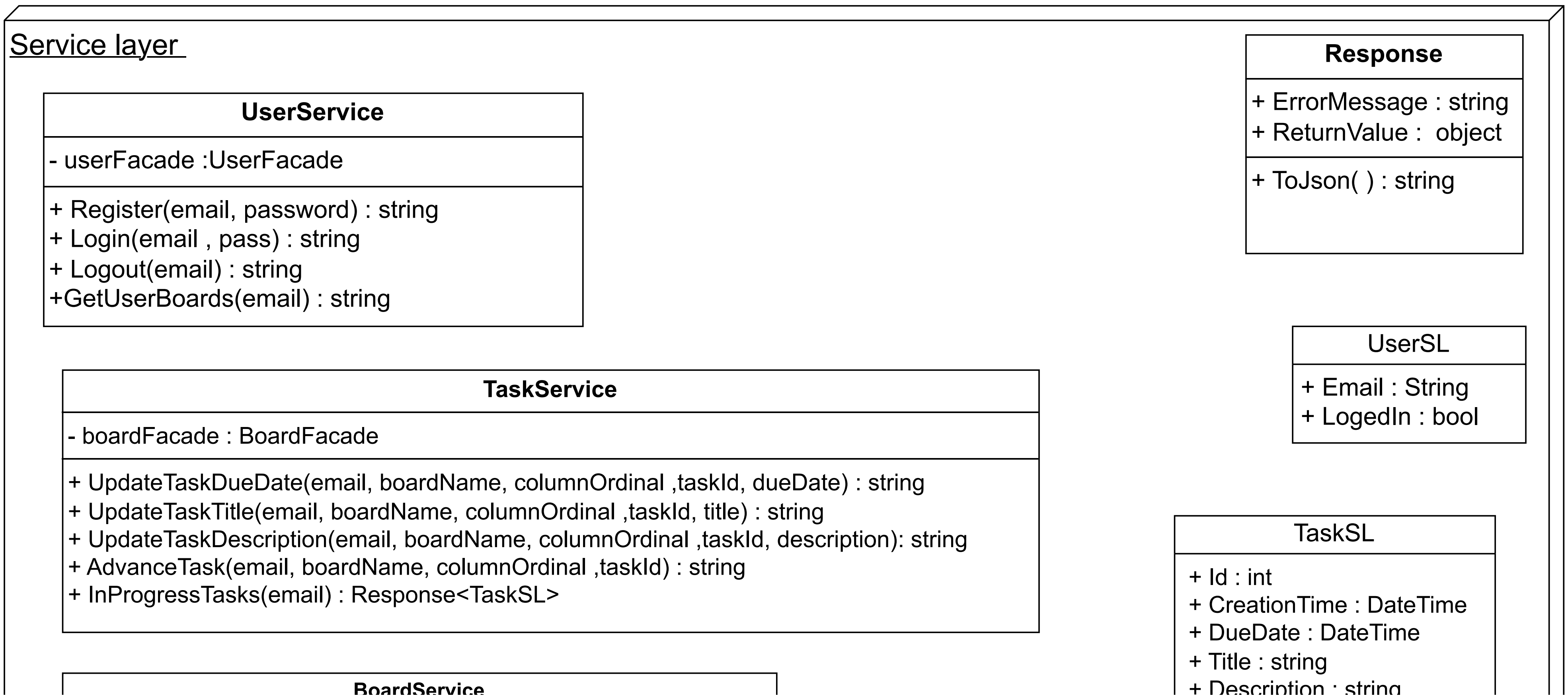


Service layer

UserService

- userFacade :UserFacade

+ Register(email, password) : string
+ Login(email , pass) : string
+ Logout(email) : string
+GetUserBoards(email) : string

TaskService

- boardFacade : BoardFacade

+ UpdateTaskDueDate(email, boardName, columnOrdinal ,taskId, dueDate) : string
+ UpdateTaskTitle(email, boardName, columnOrdinal ,taskId, title) : string
+ UpdateTaskDescription(email, boardName, columnOrdinal ,taskId, description): string
+ AdvanceTask(email, boardName, columnOrdinal ,taskId) : string
+ InProgressTasks(email) : Response<TaskSL>

BoardService

Response

+ ErrorMessage : string
+ ReturnValue : object

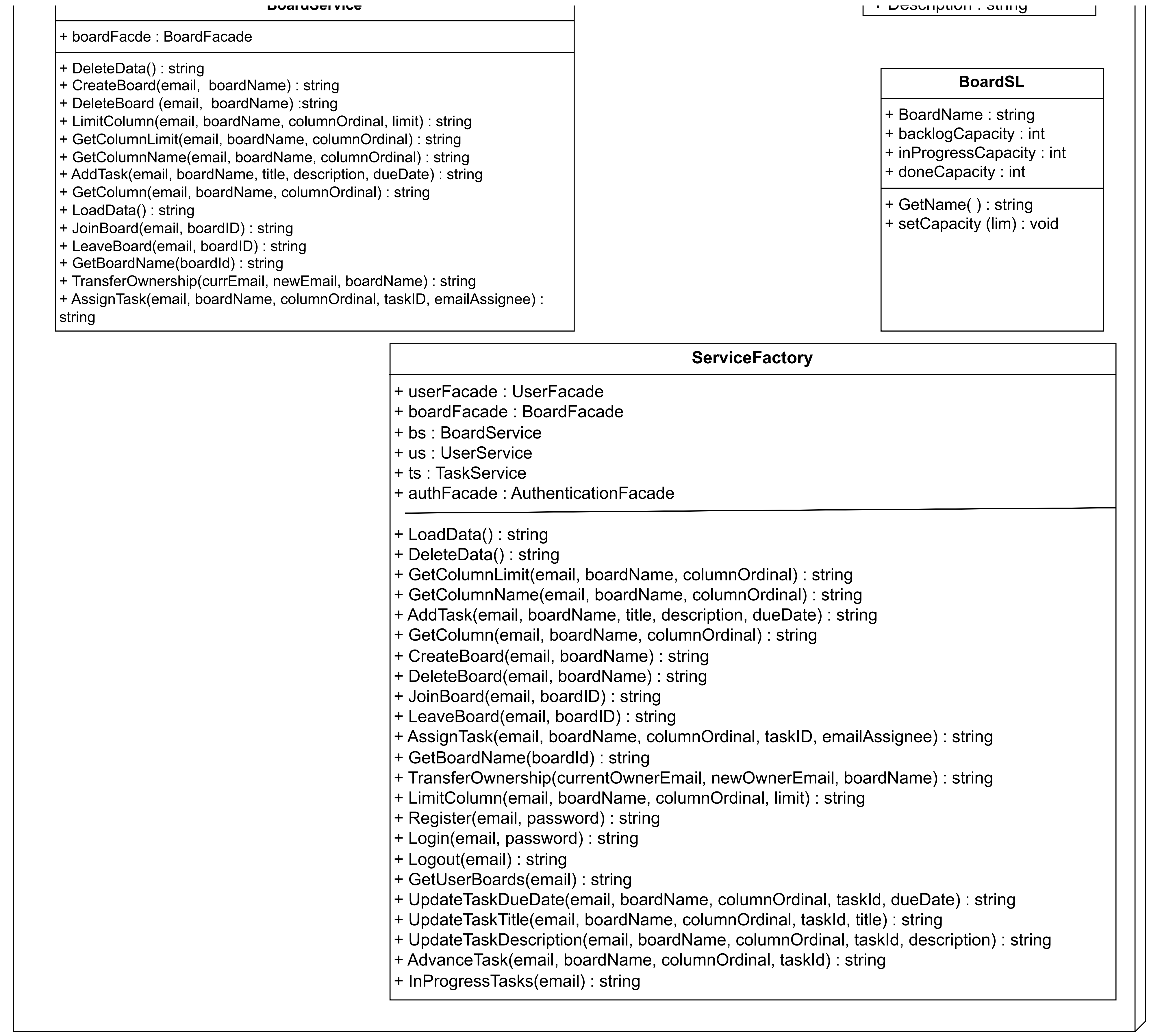
+ ToJson( ) : string

UserSL

+ Email : String
+ LogedIn : bool

TaskSL

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



BoardService

+ boardFacde : BoardFacade

+ DeleteData() : string
+ CreateBoard(email, boardName) : string
+ DeleteBoard (email, boardName) :string
+ LimitColumn(email, boardName, columnOrdinal, limit) : string
+ GetColumnLimit(email, boardName, columnOrdinal) : string
+ GetColumnName(email, boardName, columnOrdinal) : string
+ AddTask(email, boardName, title, description, dueDate) : string
+ GetColumn(email, boardName, columnOrdinal) : string
+ LoadData() : string
+ JoinBoard(email, boardID) : string
+ LeaveBoard(email, boardID) : string
+ GetBoardName(boardId) : string
+ TransferOwnership(currEmail, newEmail, boardName) : string
+ AssignTask(email, boardName, columnOrdinal, taskID, emailAssignee) : string

+ Description : string

BoardSL

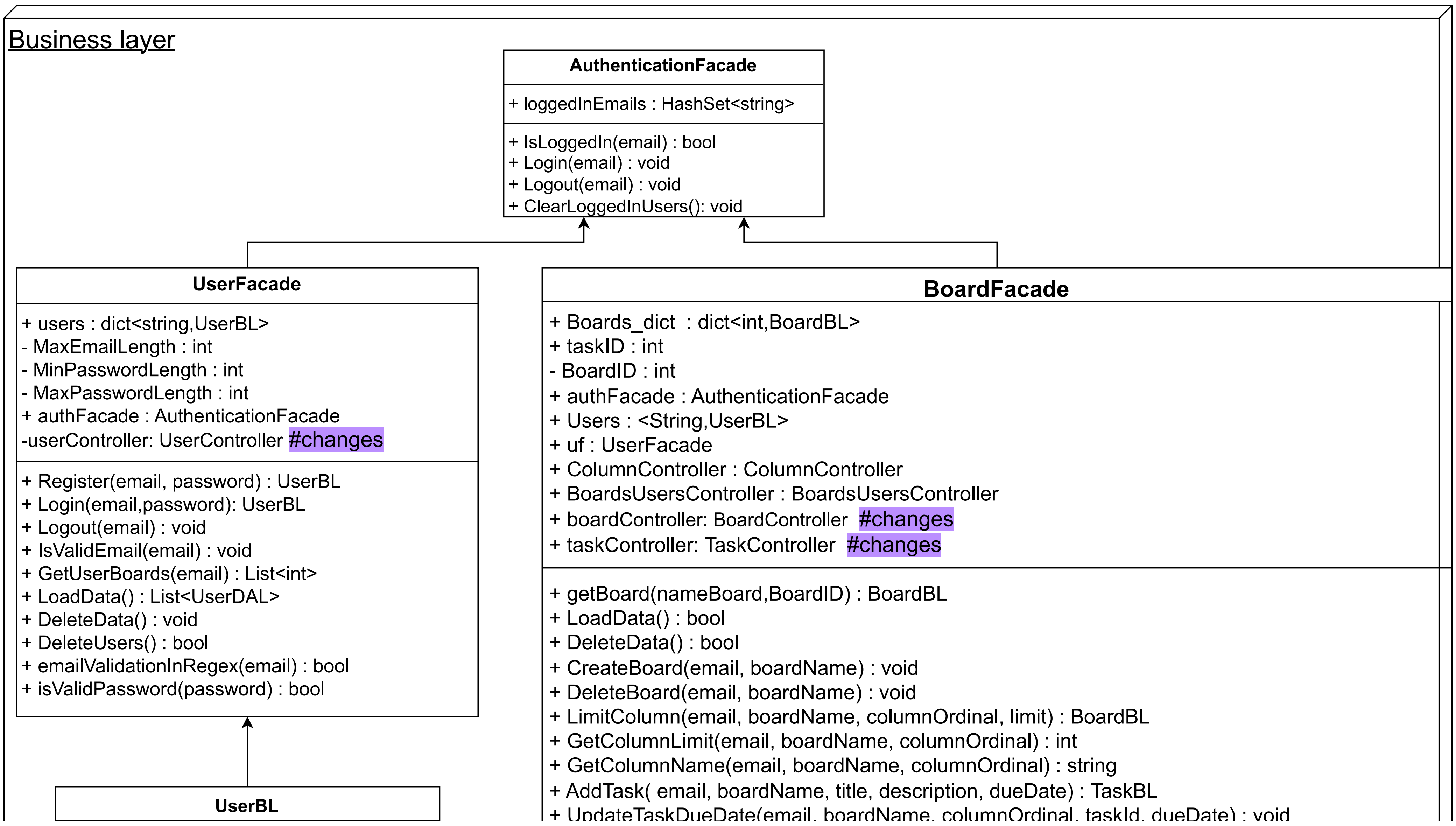
+ BoardName : string
+ backlogCapacity : int
+ inProgressCapacity : int
+ doneCapacity : int

+ GetName( ) : string
+ setCapacity (lim) : void

ServiceFactory

+ userFacade : UserFacade
+ boardFacade : BoardFacade
+ bs : BoardService
+ us : UserService
+ ts : TaskService
+ authFacade : AuthenticationFacade

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



Business layer

AuthenticationFacade

+ loggedInEmails : HashSet<string>

+ IsLoggedIn(email) : bool
+ Login(email) : void
+ Logout(email) : void
+ ClearLoggedInUsers(): void

UserFacade

+ users : dict<string,UserBL>
- MaxEmailLength : int
- MinPasswordLength : int
- MaxPasswordLength : int
+ authFacade : AuthenticationFacade
-userController: UserController #changes

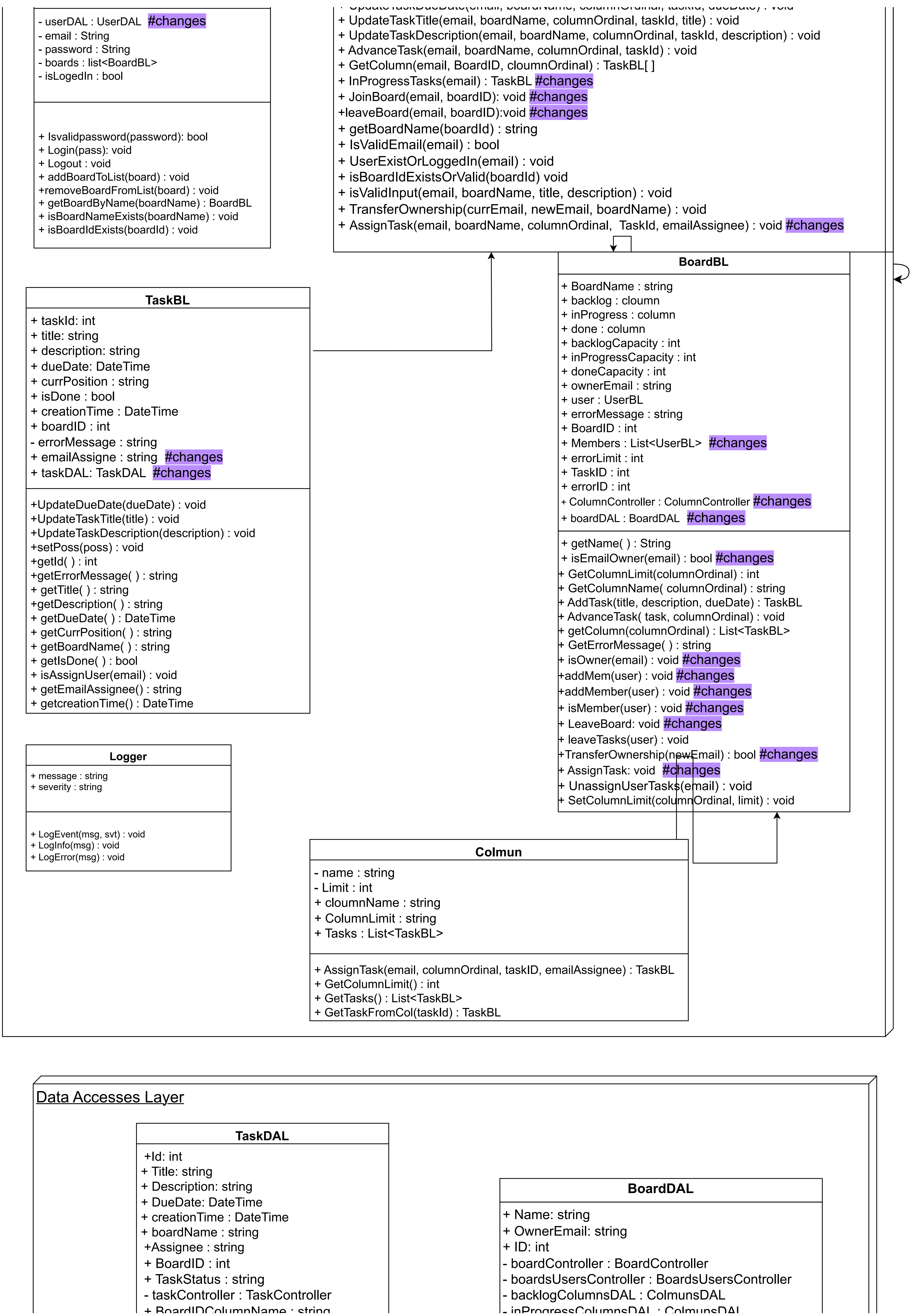
+ Register(email, password) : UserBL
+ Login(email,password): UserBL
+ Logout(email) : void
+ IsValidEmail(email) : void
+ GetUserBoards(email) : List<int>
+ LoadData() : List<UserDAL>
+ DeleteData() : void
+ DeleteUsers() : bool
+ emailValidationInRegex(email) : bool
+ isValidPassword(password) : bool

UserBL

BoardFacade

+ Boards_dict : dict<int,BoardBL>
+ taskID : int
- BoardID : int
+ authFacade : AuthenticationFacade
+ Users : <String,UserBL>
+ uf : UserFacade
+ ColumnController : ColumnController
+ BoardsUsersController : BoardsUsersController
+ boardController: BoardController #changes
+ taskController: TaskController #changes

+ getBoard(nameBoard,BoardID) : BoardBL
+ LoadData() : bool
+ DeleteData() : bool
+ CreateBoard(email, boardName) : void
+ DeleteBoard(email, boardName) : void
+ LimitColumn(email, boardName, columnOrdinal, limit) : BoardBL
+ GetColumnLimit(email, boardName, columnOrdinal) : int
+ GetColumnName(email, boardName, columnOrdinal) : string
+ AddTask(email, boardName, title, description, dueDate) : TaskBL
+ UpdateTaskDueDate(email, boardName, columnOrdinal, taskId, dueDate) : void



- userDAL : UserDAL #changes
- email : String
- password : String
- boards : list<BoardBL>
- isLogedIn : bool

+ Isvalidpassword(password): bool
+ Login(pass): void
+ Logout : void
+ addBoardToList(board) : void
+removeBoardFromList(board) : void
+ getBoardByName(boardName) : BoardBL
+ isBoardNameExists(boardName) : void
+ isBoardIdExists(boardId) : void

+ UpdateTaskDueDate(email, boardName, columnOrdinal, taskId, dueDate) : void
+ UpdateTaskTitle(email, boardName, columnOrdinal, taskId, title) : void
+ UpdateTaskDescription(email, boardName, columnOrdinal, taskId, description) : void
+ AdvanceTask(email, boardName, columnOrdinal, taskId) : void
+ GetColumn(email, BoardID, cloumnOrdinal) : TaskBL[ ]
+ InProgressTasks(email) : TaskBL #changes
+ JoinBoard(email, boardID): void #changes
+leaveBoard(email, boardID):void #changes
+ getBoardName(boardId) : string
+ IsValidEmail(email) : bool
+ UserExistOrLoggedIn(email) : void
+ isBoardIdExistsOrValid(boardId) void
+ isValidInput(email, boardName, title, description) : void
+ TransferOwnership(currEmail, newEmail, boardName) : void
+ AssignTask(email, boardName, columnOrdinal, TaskId, emailAssignee) : void #changes

TaskBL

+ taskId: int
+ title: string
+ description: string
+ dueDate: DateTime
+ currPosition : string
+ isDone : bool
+ creationTime : DateTime
+ boardID : int
- errorMessage : string
+ emailAssigne : string #changes
+ taskDAL: TaskDAL #changes

+UpdateDueDate(dueDate) : void
+UpdateTaskTitle(title) : void
+UpdateTaskDescription(description) : void
+setPoss(poss) : void
+getId( ) : int
+getErrorMessage( ) : string
+ getTitle( ) : string
+getDescription( ) : string
+ getDueDate( ) : DateTime
+ getCurrPosition( ) : string
+ getBoardName( ) : string
+ getIsDone( ) : bool
+ isAssignUser(email) : void
+ getEmailAssignee() : string
+ getcreationTime() : DateTime

BoardBL

+ BoardName : string
+ backlog : cloumn
+ inProgress : column
+ done : column
+ backlogCapacity : int
+ inProgressCapacity : int
+ doneCapacity : int
+ ownerEmail : string
+ user : UserBL
+ errorMessage : string
+ BoardID : int
+ Members : List<UserBL> #changes
+ errorLimit : int
+ TaskID : int
+ errorID : int
+ ColumnController : ColumnController #changes
+ boardDAL : BoardDAL #changes

+ getName( ) : String
+ isEmailOwner(email) : bool #changes
+ GetColumnLimit(columnOrdinal) : int
+ GetColumnName(columnOrdinal) : string
+ AddTask(title, description, dueDate) : TaskBL
+ AdvanceTask(task, columnOrdinal) : void
+ getColumn(columnOrdinal) : List<TaskBL>
+ GetErrorMessage( ) : string
+ isOwner(email) : void #changes
+addMem(user) : void #changes
+addMember(user) : void #changes
+ isMember(user) : void #changes
+ LeaveBoard: void #changes
+ leaveTasks(user) : void
+TransferOwnership(newEmail) : bool #changes
+ AssignTask: void #changes
+ UnassignUserTasks(email) : void
+ SetColumnLimit(columnOrdinal, limit) : void

Logger

+ message : string
+ severity : string

+ LogEvent(msg, svt) : void
+ LogInfo(msg) : void
+ LogError(msg) : void

Colmun

- name : string
- Limit : int
+ cloumnName : string
+ ColumnLimit : string
+ Tasks : List<TaskBL>

+ AssignTask(email, columnOrdinal, taskID, emailAssignee) : TaskBL
+ GetColumnLimit() : int
+ GetTasks() : List<TaskBL>
+ GetTaskFromCol(taskId) : TaskBL

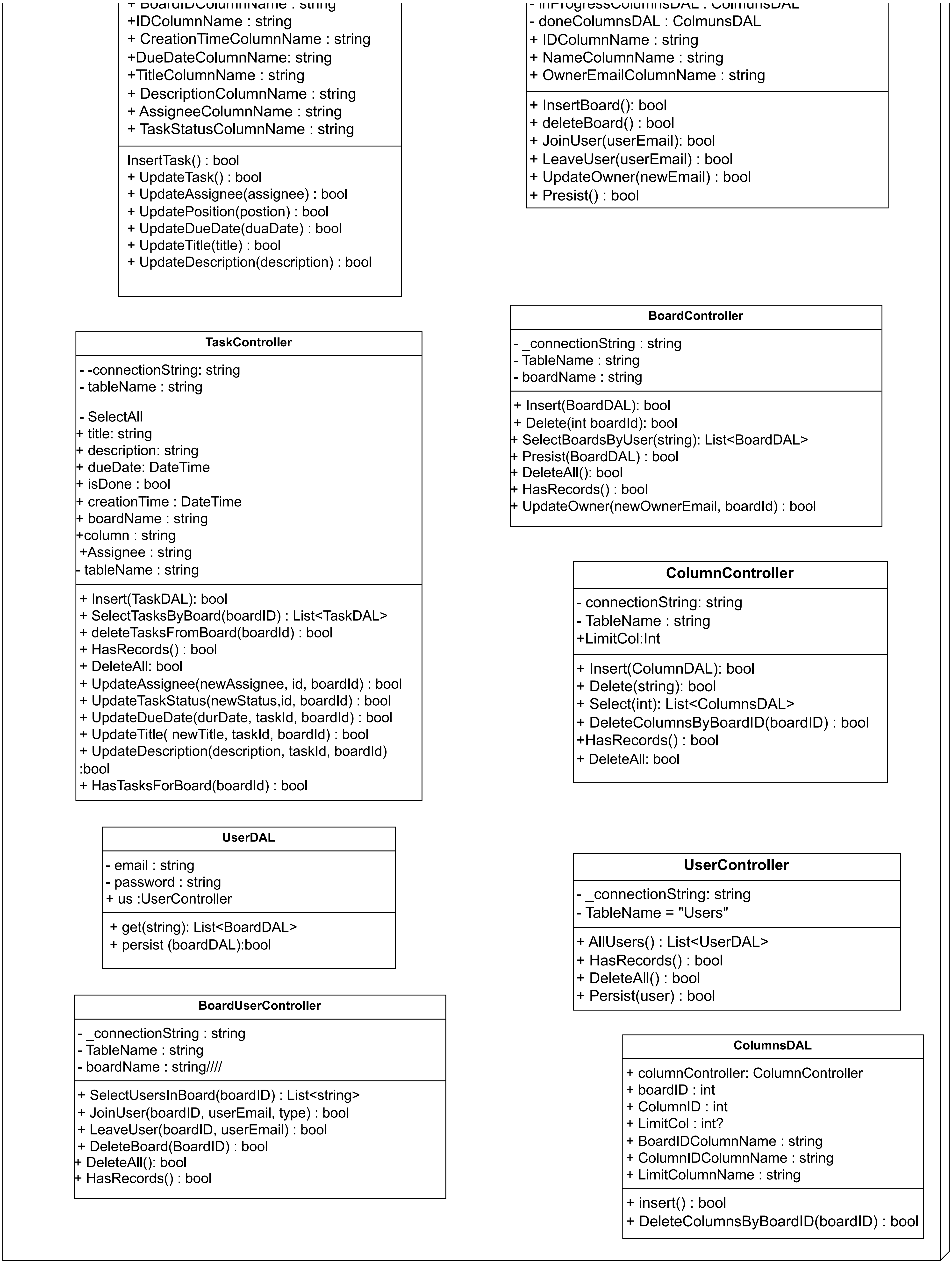
Data Accesses Layer

TaskDAL

+Id: int
+ Title: string
+ Description: string
+ DueDate: DateTime
+ creationTime : DateTime
+ boardName : string
+Assignee : string
+ BoardID : int
+ TaskStatus : string
- taskController : TaskController
+ BoardIDColumnName : string

BoardDAL

+ Name: string
+ OwnerEmail: string
+ ID: int
- boardController : BoardController
- boardsUsersController : BoardsUsersController
- backlogColumnsDAL : ColmunsDAL
- inProgressColumnsDAL : ColmunsDAL



+ BoardIDColumnName : string
+IDColumnName : string
+ CreationTimeColumnName : string
+DueDateColumnName: string
+TitleColumnName : string
+ DescriptionColumnName : string
+ AssigneeColumnName : string
+ TaskStatusColumnName : string

InsertTask() : bool
+ UpdateTask() : bool
+ UpdateAssignee(assignee) : bool
+ UpdatePosition(postion) : bool
+ UpdateDueDate(duaDate) : bool
+ UpdateTitle(title) : bool
+ UpdateDescription(description) : bool

- inProgressColumnsDAL : ColmunsDAL
- doneColumnsDAL : ColmunsDAL
+ IDColumnName : string
+ NameColumnName : string
+ OwnerEmailColumnName : string

+ InsertBoard(): bool
+ deleteBoard() : bool
+ JoinUser(userEmail): bool
+ LeaveUser(userEmail) : bool
+ UpdateOwner(newEmail) : bool
+ Presist() : bool

BoardController

- _connectionString : string
- TableName : string
- boardName : string

+ Insert(BoardDAL): bool
+ Delete(int boardId): bool
+ SelectBoardsByUser(string): List<BoardDAL>
+ Presist(BoardDAL) : bool
+ DeleteAll(): bool
+ HasRecords() : bool
+ UpdateOwner(newOwnerEmail, boardId) : bool

TaskController

- -connectionString: string
- tableName : string

- SelectAll
+ title: string
+ description: string
+ dueDate: DateTime
+ isDone : bool
+ creationTime : DateTime
+ boardName : string
+column : string
+Assignee : string
- tableName : string

+ Insert(TaskDAL): bool
+ SelectTasksByBoard(boardID) : List<TaskDAL>
+ deleteTasksFromBoard(boardId) : bool
+ HasRecords() : bool
+ DeleteAll: bool
+ UpdateAssignee(newAssignee, id, boardId) : bool
+ UpdateTaskStatus(newStatus,id, boardId) : bool
+ UpdateDueDate(durDate, taskId, boardId) : bool
+ UpdateTitle(newTitle, taskId, boardId) : bool
+ UpdateDescription(description, taskId, boardId) :bool
+ HasTasksForBoard(boardId) : bool

ColumnController

- connectionString: string
- TableName : string
+LimitCol:Int

+ Insert(ColumnDAL): bool
+ Delete(string): bool
+ Select(int): List<ColumnsDAL>
+ DeleteColumnsByBoardID(boardID) : bool
+HasRecords() : bool
+ DeleteAll: bool

UserDAL

- email : string
- password : string
+ us :UserController

+ get(string): List<BoardDAL>
+ persist (boardDAL):bool

UserController

- _connectionString: string
- TableName = "Users"

+ AllUsers() : List<UserDAL>
+ HasRecords() : bool
+ DeleteAll() : bool
+ Persist(user) : bool

BoardUserController

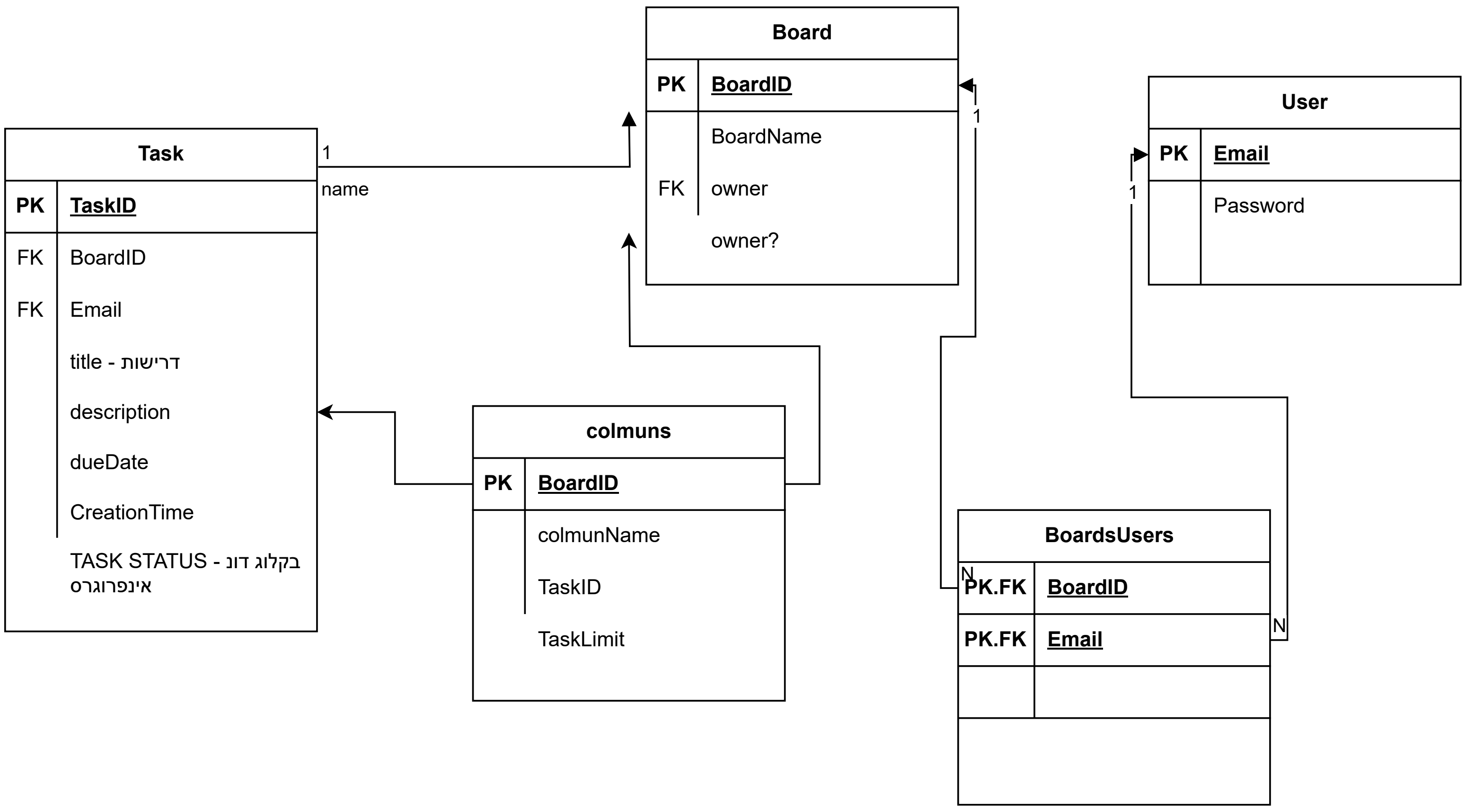
- _connectionString : string
- TableName : string
- boardName : string////

+ SelectUsersInBoard(boardID) : List<string>
+ JoinUser(boardID, userEmail, type) : bool
+ LeaveUser(boardID, userEmail) : bool
+ DeleteBoard(BoardID) : bool
+ DeleteAll(): bool
+ HasRecords() : bool

ColumnsDAL

+ columnController: ColumnController
+ boardID : int
+ ColumnID : int
+ LimitCol : int?
+ BoardIDColumnName : string
+ ColumnIDColumnName : string
+ LimitColumnName : string

+ insert() : bool
+ DeleteColumnsByBoardID(boardID) : bool


Task
PK TaskID
FK BoardID
FK Email
title - דרישות
description
dueDate
CreationTime
TASK STATUS - בקלוג דונ אינפרוגרס
1
name
Board
PK BoardID
BoardName
FK owner
owner?
1
User
PK Email
Password
1
colmuns
PK BoardID
colmunName
TaskID
TaskLimit
BoardsUsers
N
PK.FK BoardID
PK.FK Email
N