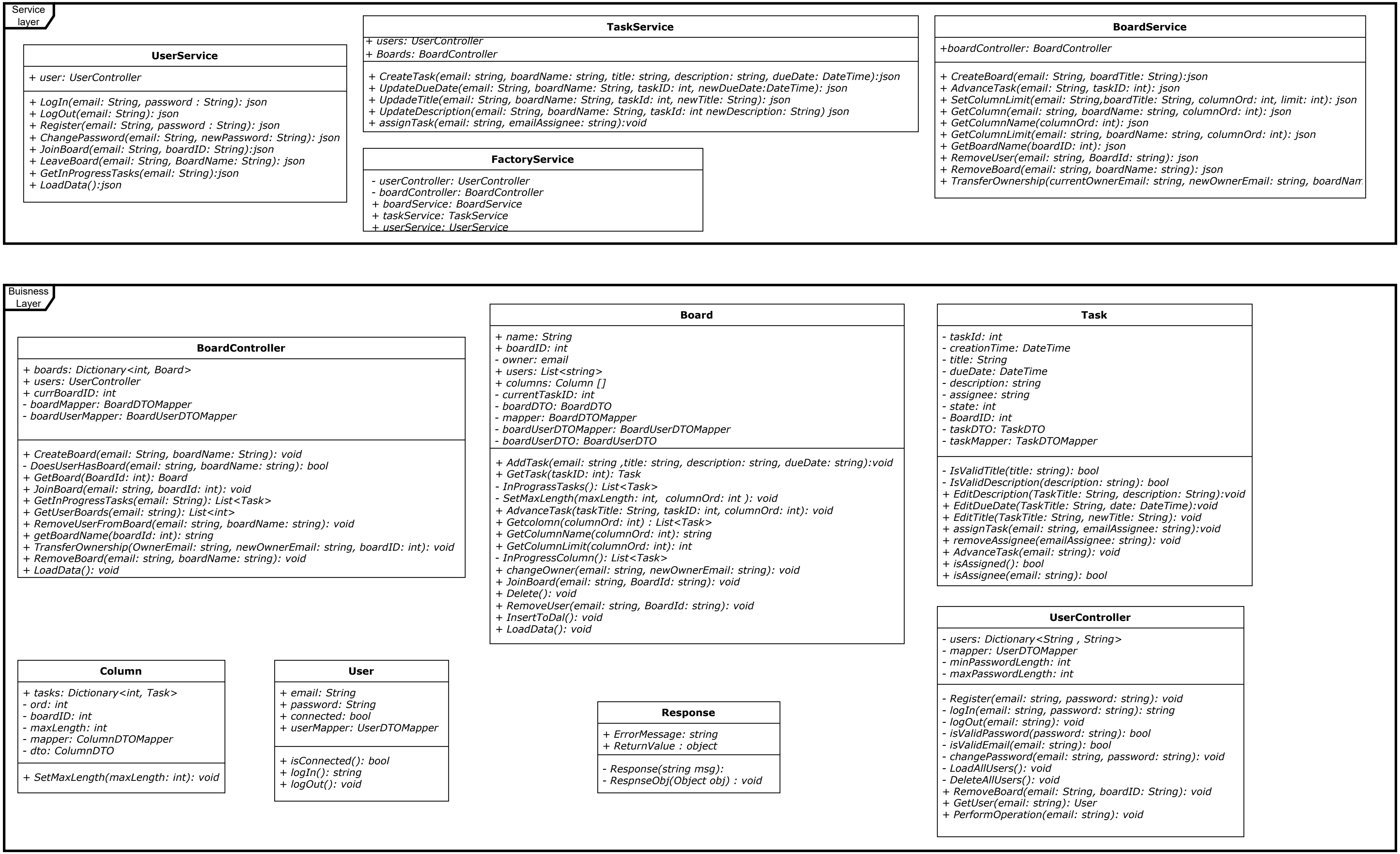


Service layer

**UserService**

+ user: UserController

+ LogIn(email: String, password : String): json
+ LogOut(email: String): json
+ Register(email: String, password : String): json
+ ChangePassword(email: String, newPassword: String): json
+ JoinBoard(email: String, boardID: String):json
+ LeaveBoard(email: String, BoardName: String): json
+ GetInProgressTasks(email: String):json
+ LoadData():json

**TaskService**

+ users: UserController
+ Boards: BoardController

+ CreateTask(email: string, boardName: string, title: string, description: string, dueDate: DateTime):json
+ UpdateDueDate(email: String, boardName: String, taskID: int, newDueDate:DateTime): json
+ UpdadeTitle(email: String, boardName: String, taskId: int, newTitle: String): json
+ UpdateDescription(email: String, boardName: String, taskId: int newDescription: String) json
+ assignTask(email: string, emailAssignee: string):void

**FactoryService**

- userController: UserController
- boardController: BoardController
+ boardService: BoardService
+ taskService: TaskService
+ userService: UserService

**BoardService**

+boardController: BoardController

+ CreateBoard(email: String, boardTitle: String):json
+ AdvanceTask(email: String, taskID: int): json
+ SetColumnLimit(email: String,boardTitle: String, columnOrd: int, limit: int): json
+ GetColumn(email: string, boardName: string, columnOrd: int): json
+ GetColumnName(columnOrd: int): json
+ GetColumnLimit(email: string, boardName: string, columnOrd: int): json
+ GetBoardName(boardID: int): json
+ RemoveUser(email: string, BoardId: string): json
+ RemoveBoard(email: string, boardName: string): json
+ TransferOwnership(currentOwnerEmail: string, newOwnerEmail: string, boardNam

Buisness Layer

**BoardController**

+ boards: Dictionary<int, Board>
+ users: UserController
+ currBoardID: int
- boardMapper: BoardDTOMapper
- boardUserMapper: BoardUserDTOMapper

+ CreateBoard(email: String, boardName: String): void
- DoesUserHasBoard(email: string, boardName: string): bool
+ GetBoard(BoardId: int): Board
+ JoinBoard(email: string, boardId: int): void
+ GetInProgressTasks(email: String): List<Task>
+ GetUserBoards(email: string): List<int>
+ RemoveUserFromBoard(email: string, boardName: string): void
+ getBoardName(boardId: int): string
+ TransferOwnership(OwnerEmail: string, newOwnerEmail: string, boardID: int): void
+ RemoveBoard(email: string, boardName: string): void
+ LoadData(): void

**Board**

+ name: String
+ boardID: int
- owner: email
+ users: List<string>
+ columns: Column []
- currentTaskID: int
- boardDTO: BoardDTO
- mapper: BoardDTOMapper
- boardUserDTOMapper: BoardUserDTOMapper
- boardUserDTO: BoardUserDTO

+ AddTask(email: string ,title: string, description: string, dueDate: string):void
+ GetTask(taskID: int): Task
- InPrograssTasks(): List<Task>
- SetMaxLength(maxLength: int, columnOrd: int ): void
+ AdvanceTask(taskTitle: String, taskID: int, columnOrd: int): void
+ Getcolomn(columnOrd: int) : List<Task>
+ GetColumnName(columnOrd: int): string
+ GetColumnLimit(columnOrd: int): int
- InProgressColumn(): List<Task>
+ changeOwner(email: string, newOwnerEmail: string): void
+ JoinBoard(email: string, BoardId: string): void
+ Delete(): void
+ RemoveUser(email: string, BoardId: string): void
+ InsertToDal(): void
+ LoadData(): void

**Task**

- taskId: int
- creationTime: DateTime
- title: String
- dueDate: DateTime
- description: string
- assignee: string
- state: int
- BoardID: int
- taskDTO: TaskDTO
- taskMapper: TaskDTOMapper

- IsValidTitle(title: string): bool
- IsValidDescription(description: string): bool
+ EditDescription(TaskTitle: String, description: String):void
+ EditDueDate(TaskTitle: String, date: DateTime):void
+ EditTitle(TaskTitle: String, newTitle: String): void
+ assignTask(email: string, emailAssignee: string):void
+ removeAssignee(emailAssignee: string): void
+ AdvanceTask(email: string): void
+ isAssigned(): bool
+ isAssignee(email: string): bool

**Column**

+ tasks: Dictionary<int, Task>
- ord: int
- boardID: int
- maxLength: int
- mapper: ColumnDTOMapper
- dto: ColumnDTO

+ SetMaxLength(maxLength: int): void

**User**

+ email: String
+ password: String
+ connected: bool
+ userMapper: UserDTOMapper

+ isConnected(): bool
+ logIn(): string
+ logOut(): void

**Response**

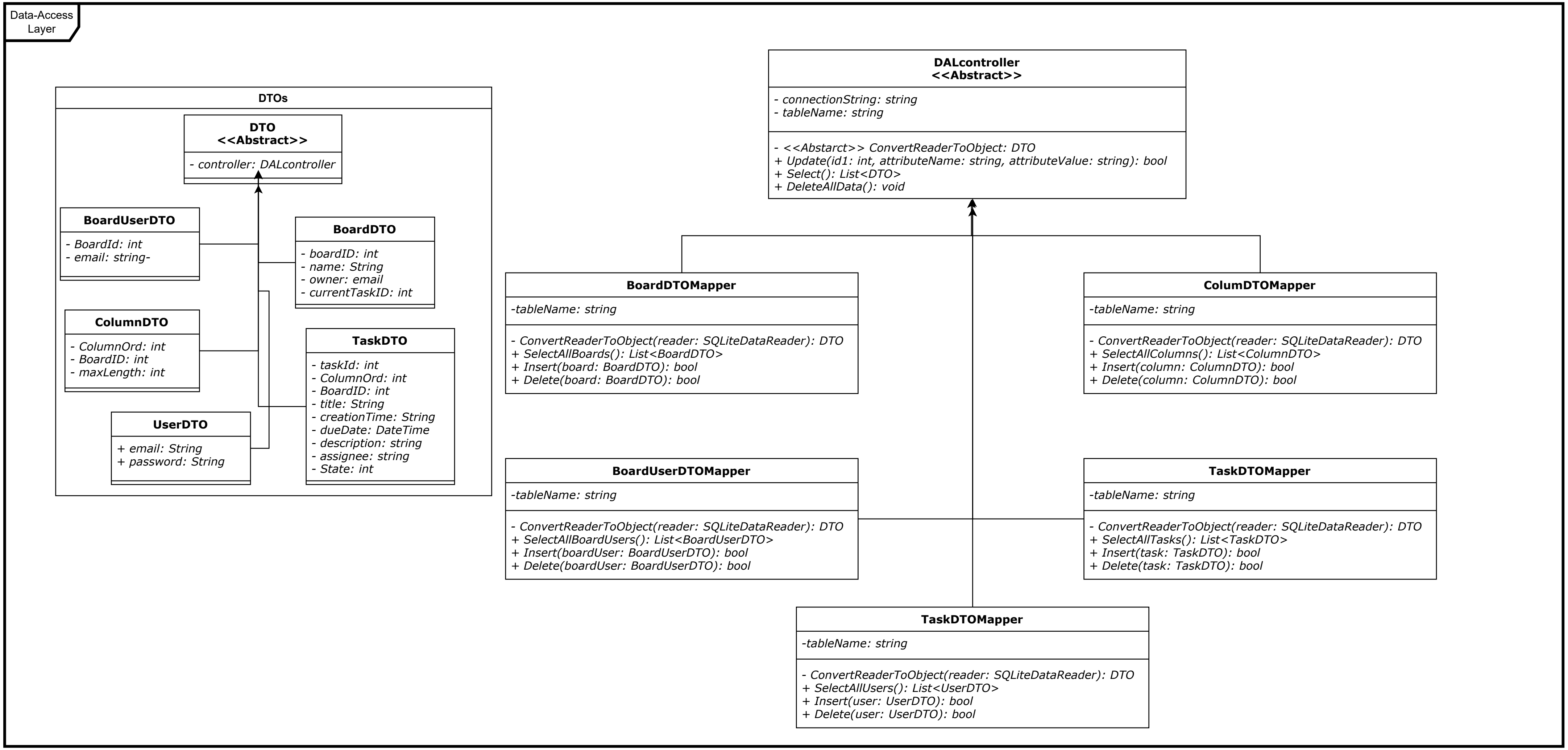
+ ErrorMessage: string
+ ReturnValue : object

- Response(string msg):
- RespnseObj(Object obj) : void

**UserController**

- users: Dictionary<String , String>
- mapper: UserDTOMapper
- minPasswordLength: int
- maxPasswordLength: int

- Register(email: string, password: string): void
- logIn(email: string, password: string): string
- logOut(email: string): void
- isValidPassword(password: string): bool
- isValidEmail(email: string): bool
- changePassword(email: string, password: string): void
- LoadAllUsers(): void
- DeleteAllUsers(): void
+ RemoveBoard(email: String, boardID: String): void
+ GetUser(email: string): User
+ PerformOperation(email: string): void


Data-Access Layer
DTOs
DTO
<<Abstract>>
- controller: DALcontroller
BoardUserDTO
- BoardId: int
- email: string-
BoardDTO
- boardID: int
- name: String
- owner: email
- currentTaskID: int
ColumnDTO
- ColumnOrd: int
- BoardID: int
- maxLength: int
TaskDTO
- taskId: int
- ColumnOrd: int
- BoardID: int
- title: String
- creationTime: String
- dueDate: DateTime
- description: string
- assignee: string
- State: int
UserDTO
+ email: String
+ password: String
DALcontroller
<<Abstract>>
- connectionString: string
- tableName: string
- <<Abstarct>> ConvertReaderToObject: DTO
+ Update(id1: int, attributeName: string, attributeValue: string): bool
+ Select(): List<DTO>
+ DeleteAllData(): void
BoardDTOMapper
-tableName: string
- ConvertReaderToObject(reader: SQLiteDataReader): DTO
+ SelectAllBoards(): List<BoardDTO>
+ Insert(board: BoardDTO): bool
+ Delete(board: BoardDTO): bool
ColumDTOMapper
-tableName: string
- ConvertReaderToObject(reader: SQLiteDataReader): DTO
+ SelectAllColumns(): List<ColumnDTO>
+ Insert(column: ColumnDTO): bool
+ Delete(column: ColumnDTO): bool
BoardUserDTOMapper
-tableName: string
- ConvertReaderToObject(reader: SQLiteDataReader): DTO
+ SelectAllBoardUsers(): List<BoardUserDTO>
+ Insert(boardUser: BoardUserDTO): bool
+ Delete(boardUser: BoardUserDTO): bool
TaskDTOMapper
-tableName: string
- ConvertReaderToObject(reader: SQLiteDataReader): DTO
+ SelectAllTasks(): List<TaskDTO>
+ Insert(task: TaskDTO): bool
+ Delete(task: TaskDTO): bool
TaskDTOMapper
-tableName: string
- ConvertReaderToObject(reader: SQLiteDataReader): DTO
+ SelectAllUsers(): List<UserDTO>
+ Insert(user: UserDTO): bool
+ Delete(user: UserDTO): bool