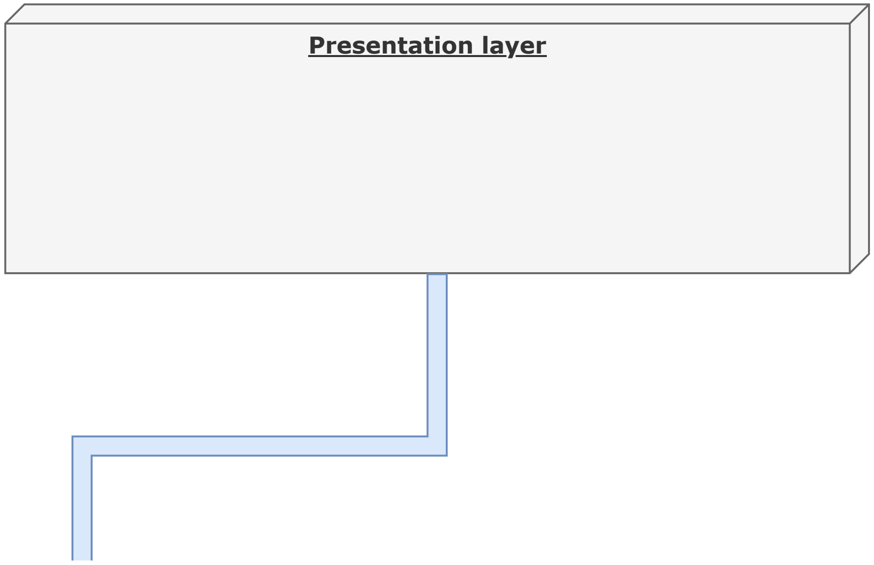
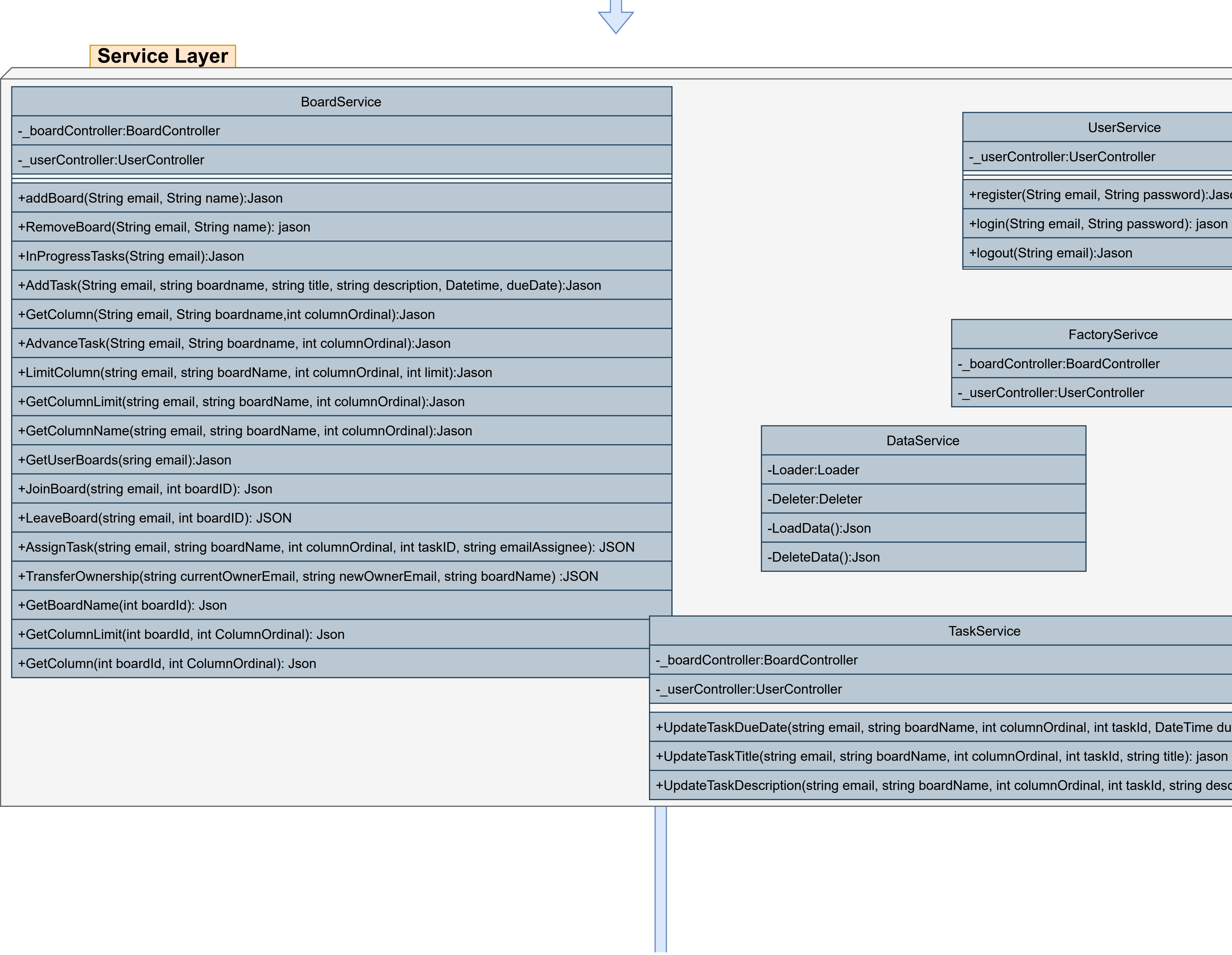

Presentation layer



Service Layer

BoardService

-_boardController:BoardController

-_userController:UserController

+addBoard(String email, String name):Jason

+RemoveBoard(String email, String name): jason

+InProgressTasks(String email):Jason

+AddTask(String email, string boardname, string title, string description, Datetime, dueDate):Jason

+GetColumn(String email, String boardname,int columnOrdinal):Jason

+AdvanceTask(String email, String boardname, int columnOrdinal):Jason

+LimitColumn(string email, string boardName, int columnOrdinal, int limit):Jason

+GetColumnLimit(string email, string boardName, int columnOrdinal):Jason

+GetColumnName(string email, string boardName, int columnOrdinal):Jason

+GetUserBoards(sring email):Jason

+JoinBoard(string email, int boardID): Json

+LeaveBoard(string email, int boardID): JSON

+AssignTask(string email, string boardName, int columnOrdinal, int taskID, string emailAssignee): JSON

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

+GetBoardName(int boardId): Json

+GetColumnLimit(int boardId, int ColumnOrdinal): Json

+GetColumn(int boardId, int ColumnOrdinal): Json

UserService

-_userController:UserController

+register(String email, String password):Jaso

+login(String email, String password): jason

+logout(String email):Jason

FactorySerivce

-_boardController:BoardController

-_userController:UserController

DataService

-Loader:Loader

-Deleter:Deleter

-LoadData():Json

-DeleteData():Json

TaskService

-_boardController:BoardController

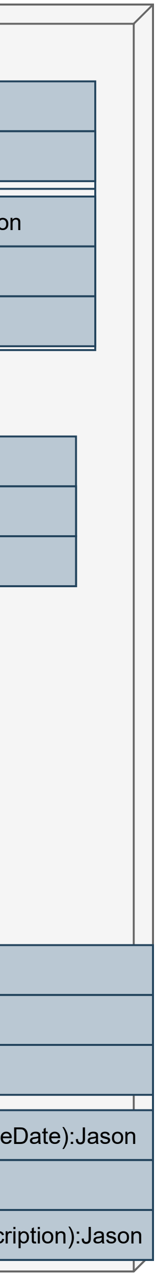
-_userController:UserController

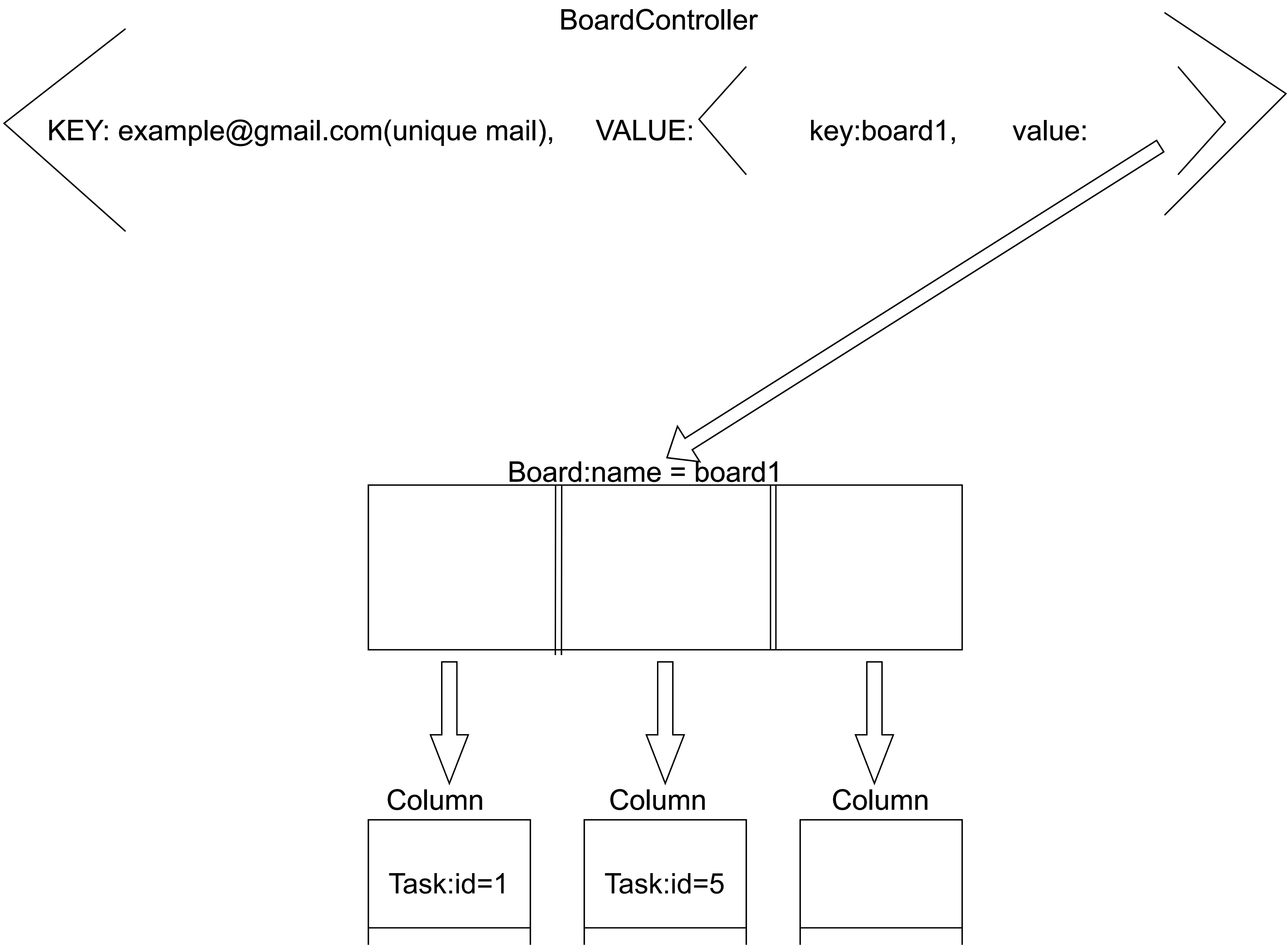
+UpdateTaskDueDate(string email, string boardName, int columnOrdinal, int taskId, DateTime du

+UpdateTaskTitle(string email, string boardName, int columnOrdinal, int taskId, string title): jason

+UpdateTaskDescription(string email, string boardName, int columnOrdinal, int taskId, string desc

Example:





BoardController

KEY: example@gmail.com(unique mail), VALUE: key:board1, value:

Board:name = board1

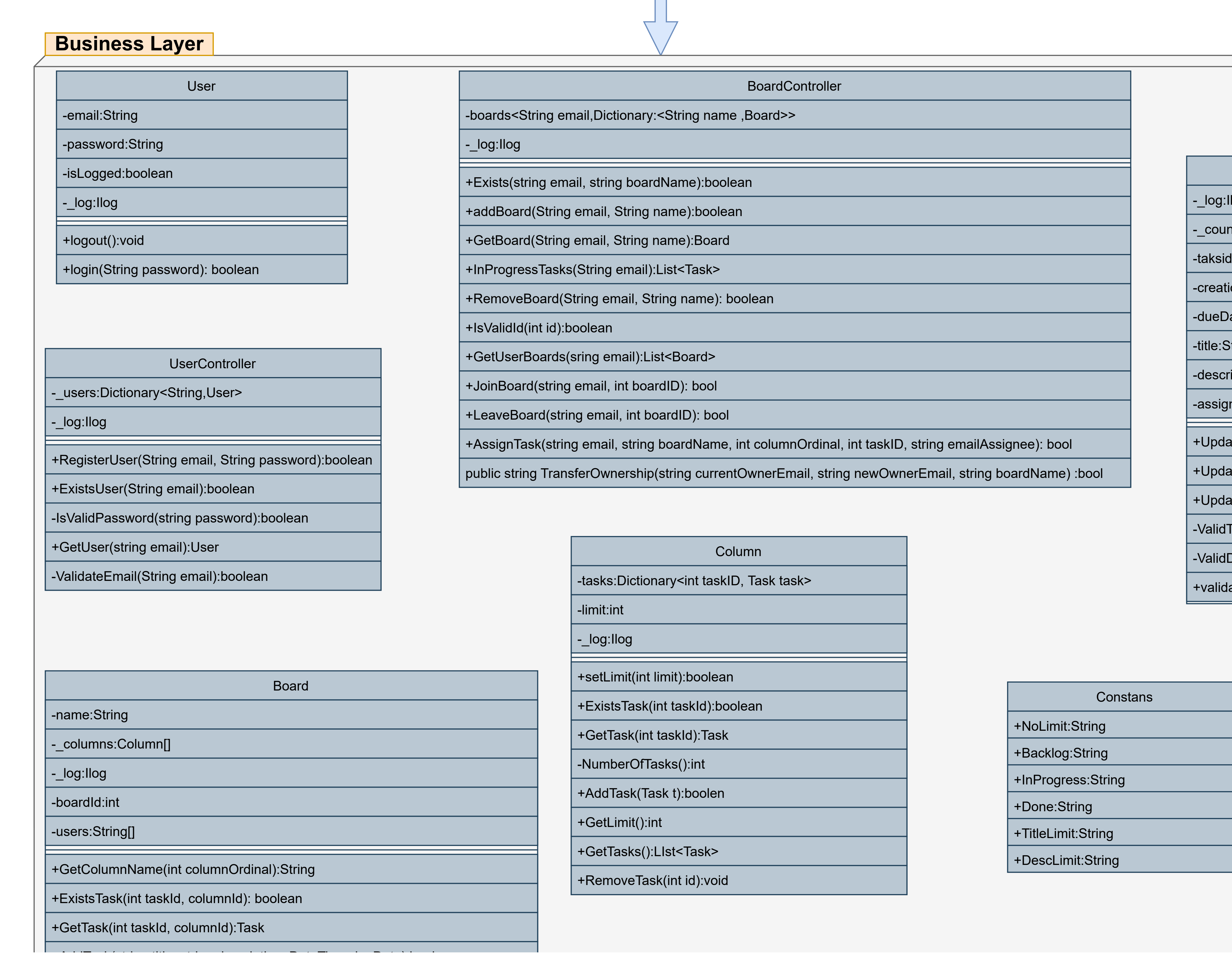
Column

Task:id=1

Column

Task:id=5

Column



Business Layer

User

-email:String

-password:String

-isLogged:boolean

-_log:Ilog

+logout():void

+login(String password): boolean

BoardController

-boards<String email,Dictionary:<String name ,Board>>

-_log:Ilog

+Exists(string email, string boardName):boolean

+addBoard(String email, String name):boolean

+GetBoard(String email, String name):Board

+InProgressTasks(String email):List<Task>

+RemoveBoard(String email, String name): boolean

+IsValidId(int id):boolean

+GetUserBoards(sring email):List<Board>

+JoinBoard(string email, int boardID): bool

+LeaveBoard(string email, int boardID): bool

+AssignTask(string email, string boardName, int columnOrdinal, int taskID, string emailAssignee): bool

public string TransferOwnership(string currentOwnerEmail, string newOwnerEmail, string boardName) :bool

-_log:Il

-_coun

-taksid

-creatio

-dueDa

-title:S

-descri

-assign

+Upda

+Upda

+Upda

-ValidT

-ValidD

+valida

UserController

-_users:Dictionary<String,User>

-_log:Ilog

+RegisterUser(String email, String password):boolean

+ExistsUser(String email):boolean

-IsValidPassword(string password):boolean

+GetUser(string email):User

-ValidateEmail(String email):boolean

Column

-tasks:Dictionary<int taskID, Task task>

-limit:int

-_log:Ilog

+setLimit(int limit):boolean

+ExistsTask(int taskId):boolean

+GetTask(int taskId):Task

-NumberOfTasks():int

+AddTask(Task t):boolen

+GetLimit():int

+GetTasks():LIst<Task>

+RemoveTask(int id):void

Board

-name:String

-_columns:Column[]

-_log:Ilog

-boardId:int

-users:String[]

+GetColumnName(int columnOrdinal):String

+ExistsTask(int taskId, columnId): boolean

+GetTask(int taskId, columnId):Task

Constans

+NoLimit:String

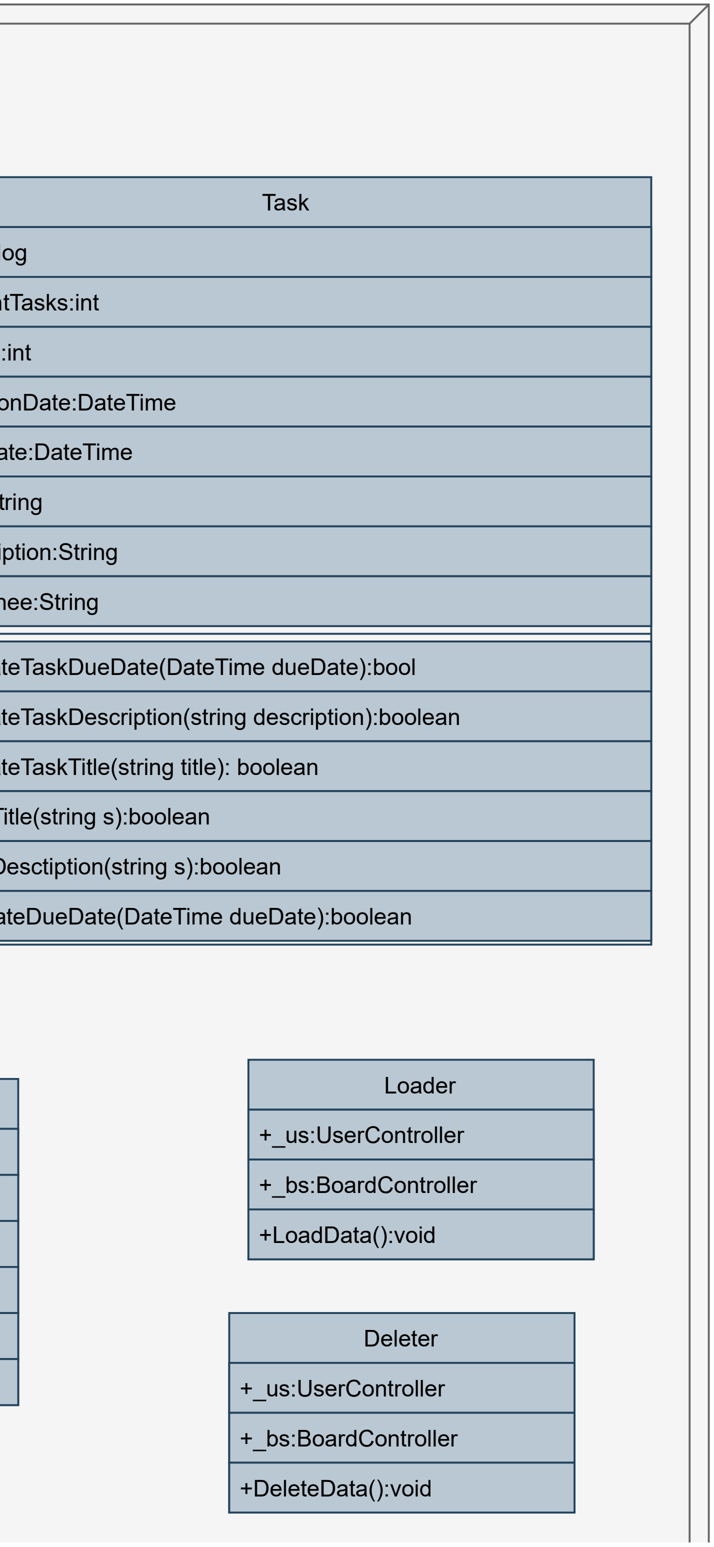
+Backlog:String

+InProgress:String

+Done:String

+TitleLimit:String

+DescLimit:String



Task

og

ntTasks:int

:int

onDate:DateTime

ate:DateTime

tring

iption:String

nee:String

teTaskDueDate(DateTime dueDate):bool

teTaskDescription(string description):boolean

teTaskTitle(string title): boolean

Title(string s):boolean

Desctiption(string s):boolean

ateDueDate(DateTime dueDate):boolean

Loader

+_us:UserController

+_bs:BoardController

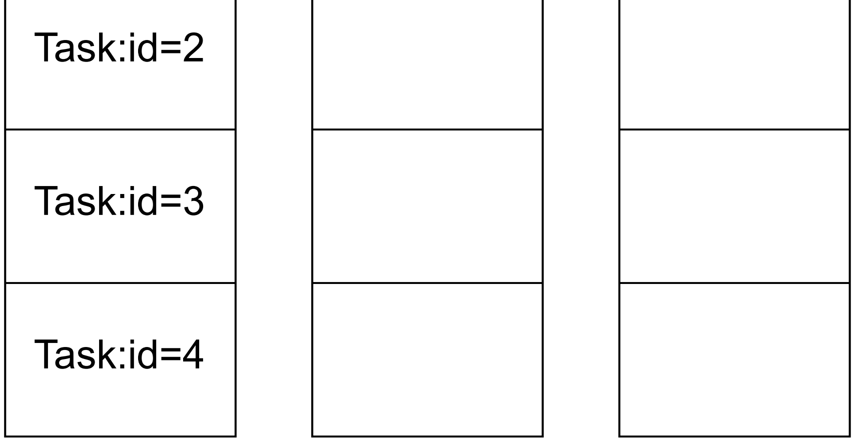
+LoadData():void

Deleter

+_us:UserController

+_bs:BoardController

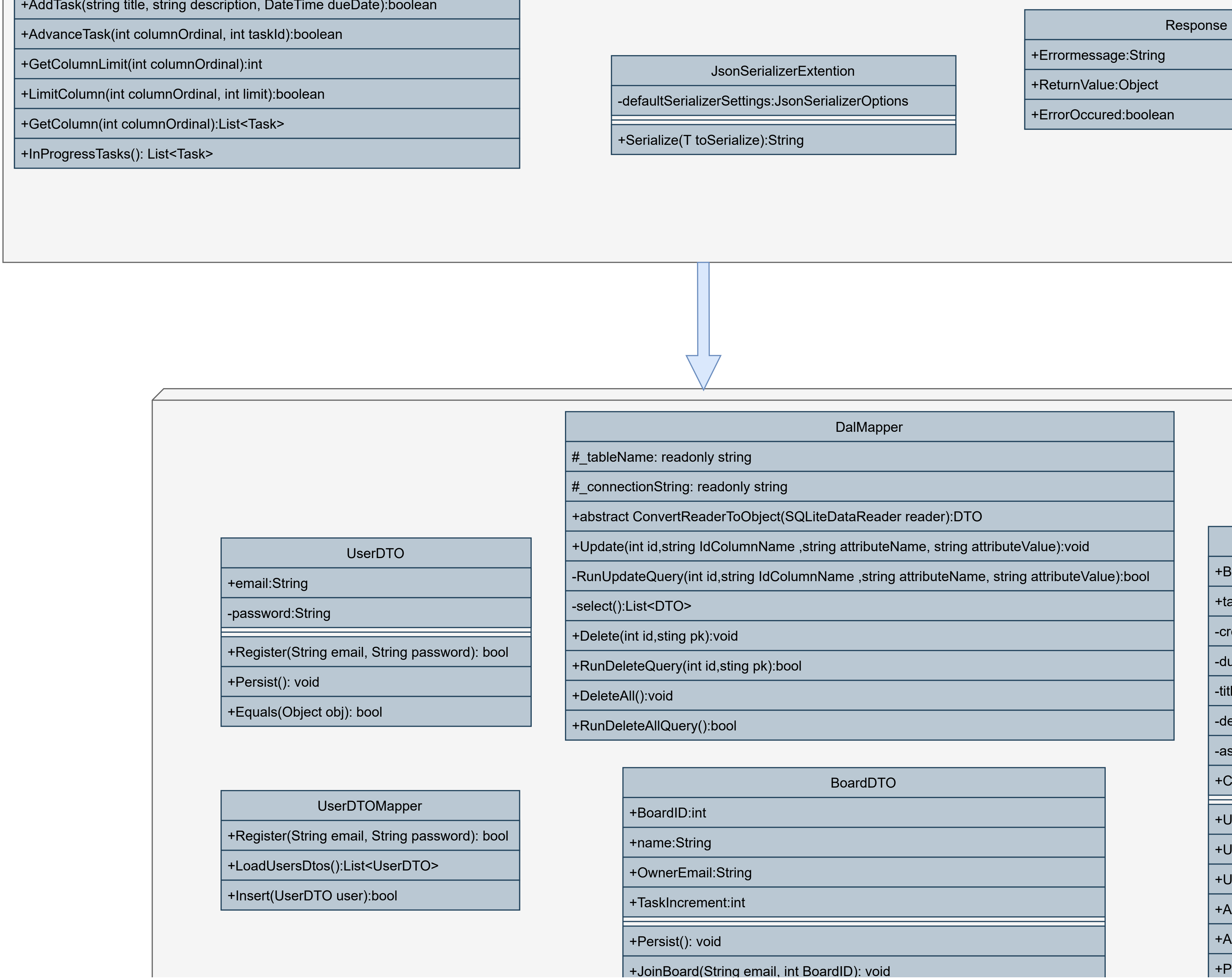
+DeleteData():void



Task:id=2

Task:id=3

Task:id=4



+AddTask(string title, string description, DateTime dueDate):boolean

+AdvanceTask(int columnOrdinal, int taskId):boolean

+GetColumnLimit(int columnOrdinal):int

+LimitColumn(int columnOrdinal, int limit):boolean

+GetColumn(int columnOrdinal):List<Task>

+InProgressTasks(): List<Task>

JsonSerializerExtention

-defaultSerializerSettings:JsonSerializerOptions

+Serialize(T toSerialize):String

Response

+Errormessage:String

+ReturnValue:Object

+ErrorOccured:boolean

DalMapper

#_tableName: readonly string

#_connectionString: readonly string

+abstract ConvertReaderToObject(SQLiteDataReader reader):DTO

+Update(int id,string IdColumnName ,string attributeName, string attributeValue):void

-RunUpdateQuery(int id,string IdColumnName ,string attributeName, string attributeValue):bool

-select():List<DTO>

+Delete(int id,sting pk):void

+RunDeleteQuery(int id,sting pk):bool

+DeleteAll():void

+RunDeleteAllQuery():bool

UserDTO

+email:String

-password:String

+Register(String email, String password): bool

+Persist(): void

+Equals(Object obj): bool

UserDTOMapper

+Register(String email, String password): bool

+LoadUsersDtos():List<UserDTO>

+Insert(UserDTO user):bool

BoardDTO

+BoardID:int

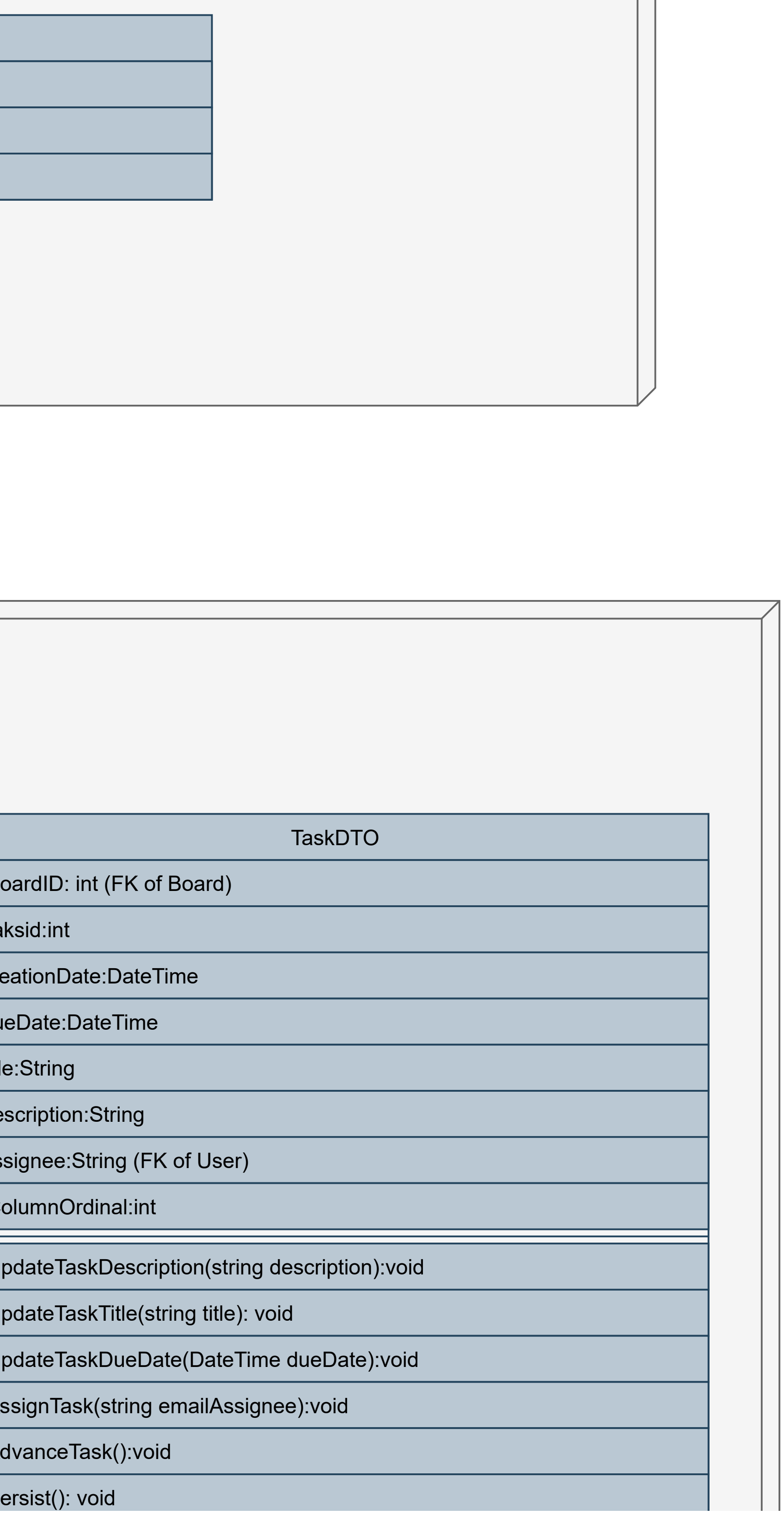
+name:String

+OwnerEmail:String

+TaskIncrement:int

+Persist(): void

+JoinBoard(String email, int BoardID): void



TaskDTO

oardID: int (FK of Board)

aksid:int

eationDate:DateTime

ueDate:DateTime

le:String

escription:String

ssignee:String (FK of User)

olumnOrdinal:int

pdateTaskDescription(string description):void

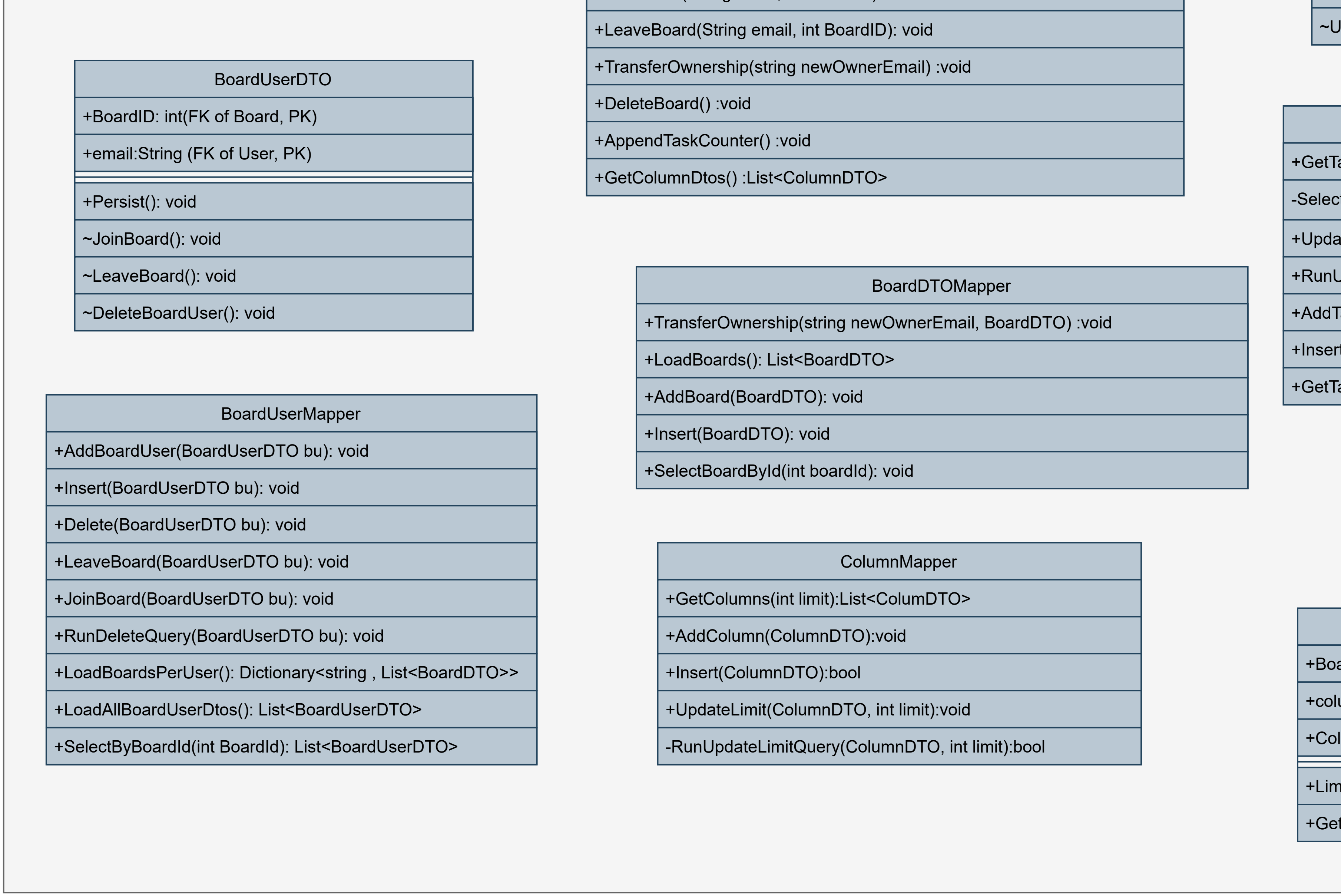
pdateTaskTitle(string title): void

pdateTaskDueDate(DateTime dueDate):void

ssignTask(string emailAssignee):void

dvanceTask():void

ersist(): void



+LeaveBoard(String email, int BoardID): void

+TransferOwnership(string newOwnerEmail) :void

+DeleteBoard() :void

+AppendTaskCounter() :void

+GetColumnDtos() :List<ColumnDTO>

~U

+GetTa

-Selec

+Upda

+RunU

+AddT

+Insert

+GetTa

BoardUserDTO

+BoardID: int(FK of Board, PK)

+email:String (FK of User, PK)

+Persist(): void

~JoinBoard(): void

~LeaveBoard(): void

~DeleteBoardUser(): void

BoardDTOMapper

+TransferOwnership(string newOwnerEmail, BoardDTO) :void

+LoadBoards(): List<BoardDTO>

+AddBoard(BoardDTO): void

+Insert(BoardDTO): void

+SelectBoardById(int boardId): void

BoardUserMapper

+AddBoardUser(BoardUserDTO bu): void

+Insert(BoardUserDTO bu): void

+Delete(BoardUserDTO bu): void

+LeaveBoard(BoardUserDTO bu): void

+JoinBoard(BoardUserDTO bu): void

+RunDeleteQuery(BoardUserDTO bu): void

+LoadBoardsPerUser(): Dictionary<string , List<BoardDTO>>

+LoadAllBoardUserDtos(): List<BoardUserDTO>

+SelectByBoardId(int BoardId): List<BoardUserDTO>

ColumnMapper

+GetColumns(int limit):List<ColumDTO>

+AddColumn(ColumnDTO):void

+Insert(ColumnDTO):bool

+UpdateLimit(ColumnDTO, int limit):void

-RunUpdateLimitQuery(ColumnDTO, int limit):bool

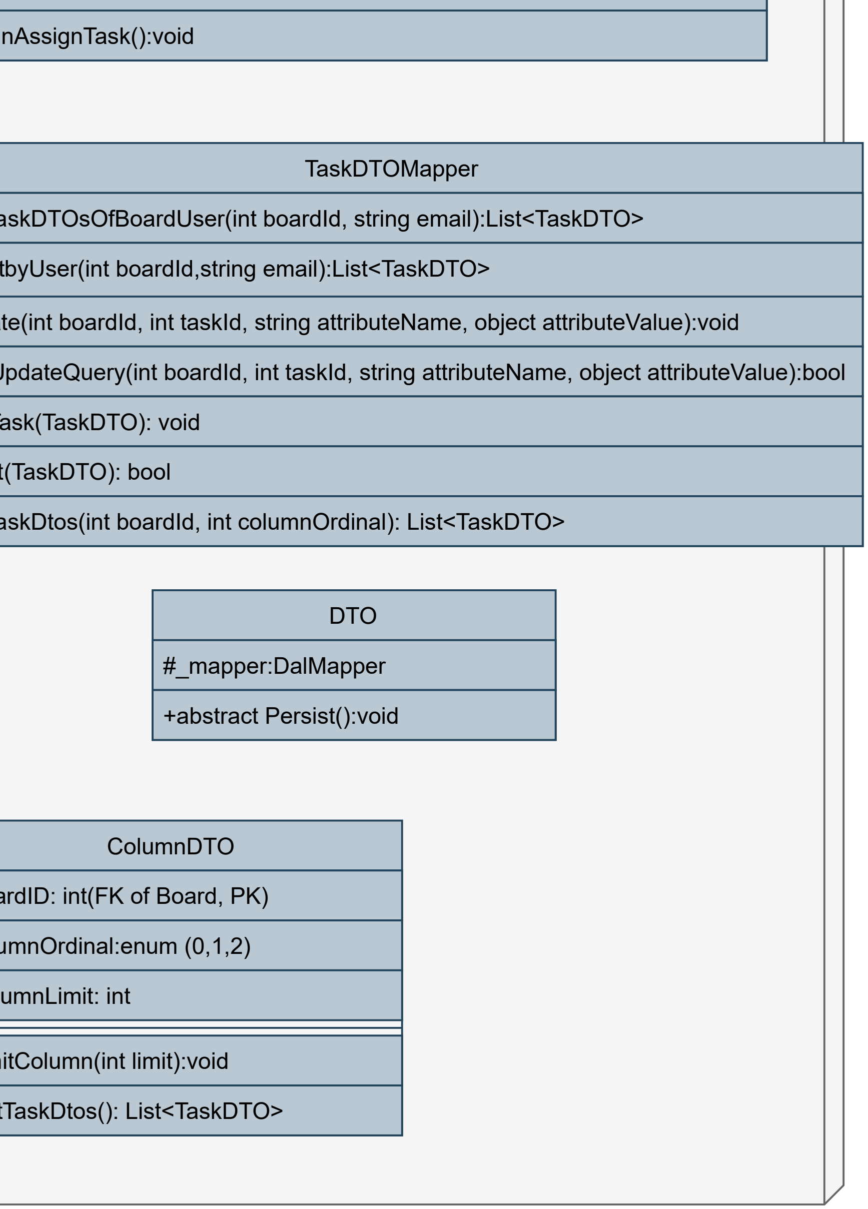
+Boa

+colu

+Col

+Lim

+Get



nAssignTask():void

TaskDTOMapper

askDTOsOfBoardUser(int boardId, string email):List<TaskDTO>

tbyUser(int boardId,string email):List<TaskDTO>

te(int boardId, int taskId, string attributeName, object attributeValue):void

UpdateQuery(int boardId, int taskId, string attributeName, object attributeValue):bool

ask(TaskDTO): void

t(TaskDTO): bool

askDtos(int boardId, int columnOrdinal): List<TaskDTO>

DTO

#_mapper:DalMapper

+abstract Persist():void

ColumnDTO

ardID: int(FK of Board, PK)

umnOrdinal:enum (0,1,2)

umnLimit: int

itColumn(int limit):void

tTaskDtos(): List<TaskDTO>