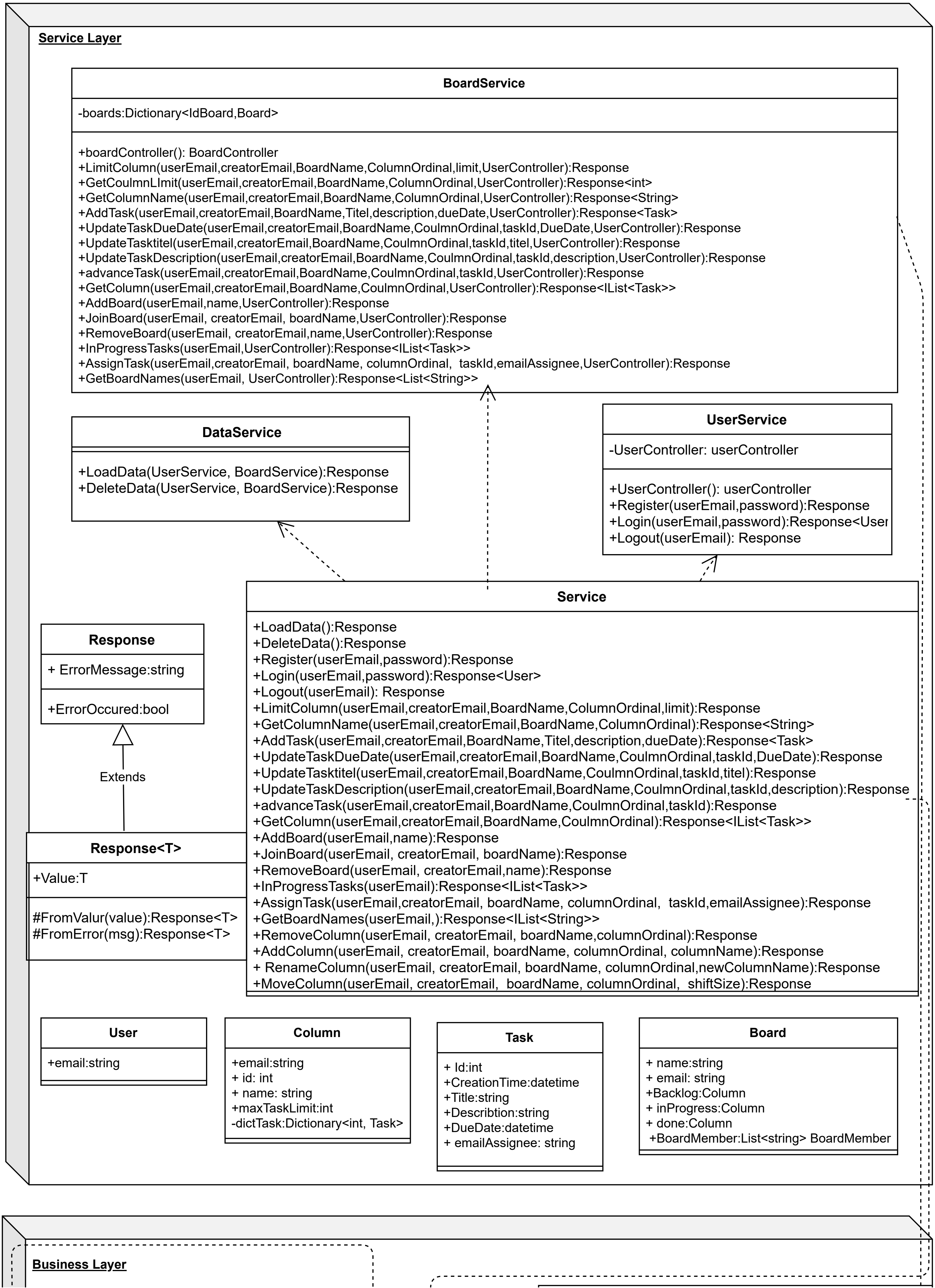


Service Layer

BoardService

-boards:Dictionary<IdBoard,Board>

+boardController(): BoardController
+LimitColumn(userEmail,creatorEmail,BoardName,ColumnOrdinal,limit,UserController):Response
+GetCoulmnLImit(userEmail,creatorEmail,BoardName,ColumnOrdinal,UserController):Response<int>
+GetColumnName(userEmail,creatorEmail,BoardName,ColumnOrdinal,UserController):Response<String>
+AddTask(userEmail,creatorEmail,BoardName,Titel,description,dueDate,UserController):Response<Task>
+UpdateTaskDueDate(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,DueDate,UserController):Response
+UpdateTasktitel(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,titel,UserController):Response
+UpdateTaskDescription(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,description,UserController):Response
+advanceTask(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,UserController):Response
+GetColumn(userEmail,creatorEmail,BoardName,CoulmnOrdinal,UserController):Response<IList<Task>>
+AddBoard(userEmail,name,UserController):Response
+JoinBoard(userEmail, creatorEmail, boardName,UserController):Response
+RemoveBoard(userEmail, creatorEmail,name,UserController):Response
+InProgressTasks(userEmail,UserController):Response<IList<Task>>
+AssignTask(userEmail,creatorEmail, boardName, columnOrdinal, taskId,emailAssignee,UserController):Response
+GetBoardNames(userEmail, UserController):Response<List<String>>

DataService

+LoadData(UserService, BoardService):Response
+DeleteData(UserService, BoardService):Response

UserService

-UserController: userController

+UserController(): userController
+Register(userEmail,password):Response
+Login(userEmail,password):Response<User
+Logout(userEmail): Response

Response

+ ErrorMessage:string

+ErrorOccured:bool

Extends

Response<T>

+Value:T

#FromValur(value):Response<T>
#FromError(msg):Response<T>

Service

+LoadData():Response
+DeleteData():Response
+Register(userEmail,password):Response
+Login(userEmail,password):Response<User>
+Logout(userEmail): Response
+LimitColumn(userEmail,creatorEmail,BoardName,ColumnOrdinal,limit):Response
+GetColumnName(userEmail,creatorEmail,BoardName,ColumnOrdinal):Response<String>
+AddTask(userEmail,creatorEmail,BoardName,Titel,description,dueDate):Response<Task>
+UpdateTaskDueDate(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,DueDate):Response
+UpdateTasktitel(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,titel):Response
+UpdateTaskDescription(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId,description):Response
+advanceTask(userEmail,creatorEmail,BoardName,CoulmnOrdinal,taskId):Response
+GetColumn(userEmail,creatorEmail,BoardName,CoulmnOrdinal):Response<IList<Task>>
+AddBoard(userEmail,name):Response
+JoinBoard(userEmail, creatorEmail, boardName):Response
+RemoveBoard(userEmail, creatorEmail,name):Response
+InProgressTasks(userEmail):Response<IList<Task>>
+AssignTask(userEmail,creatorEmail, boardName, columnOrdinal, taskId,emailAssignee):Response
+GetBoardNames(userEmail,):Response<IList<String>>
+RemoveColumn(userEmail, creatorEmail, boardName,columnOrdinal):Response
+AddColumn(userEmail, creatorEmail, boardName, columnOrdinal, columnName):Response
+ RenameColumn(userEmail, creatorEmail, boardName, columnOrdinal,newColumnName):Response
+MoveColumn(userEmail, creatorEmail, boardName, columnOrdinal, shiftSize):Response

User

+email:string

Column

+email:string
+ id: int
+ name: string
+maxTaskLimit:int
-dictTask:Dictionary<int, Task>

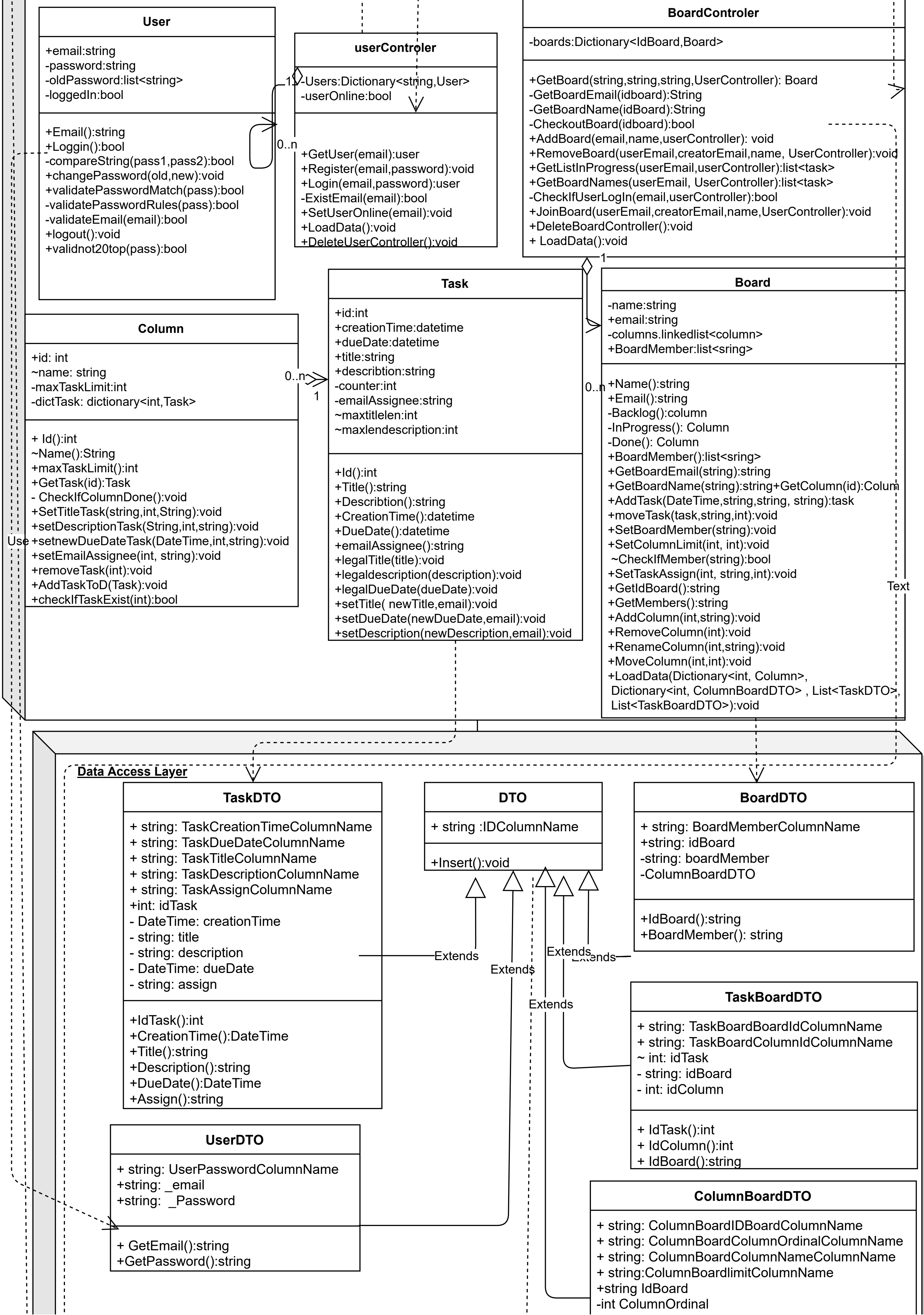
Task

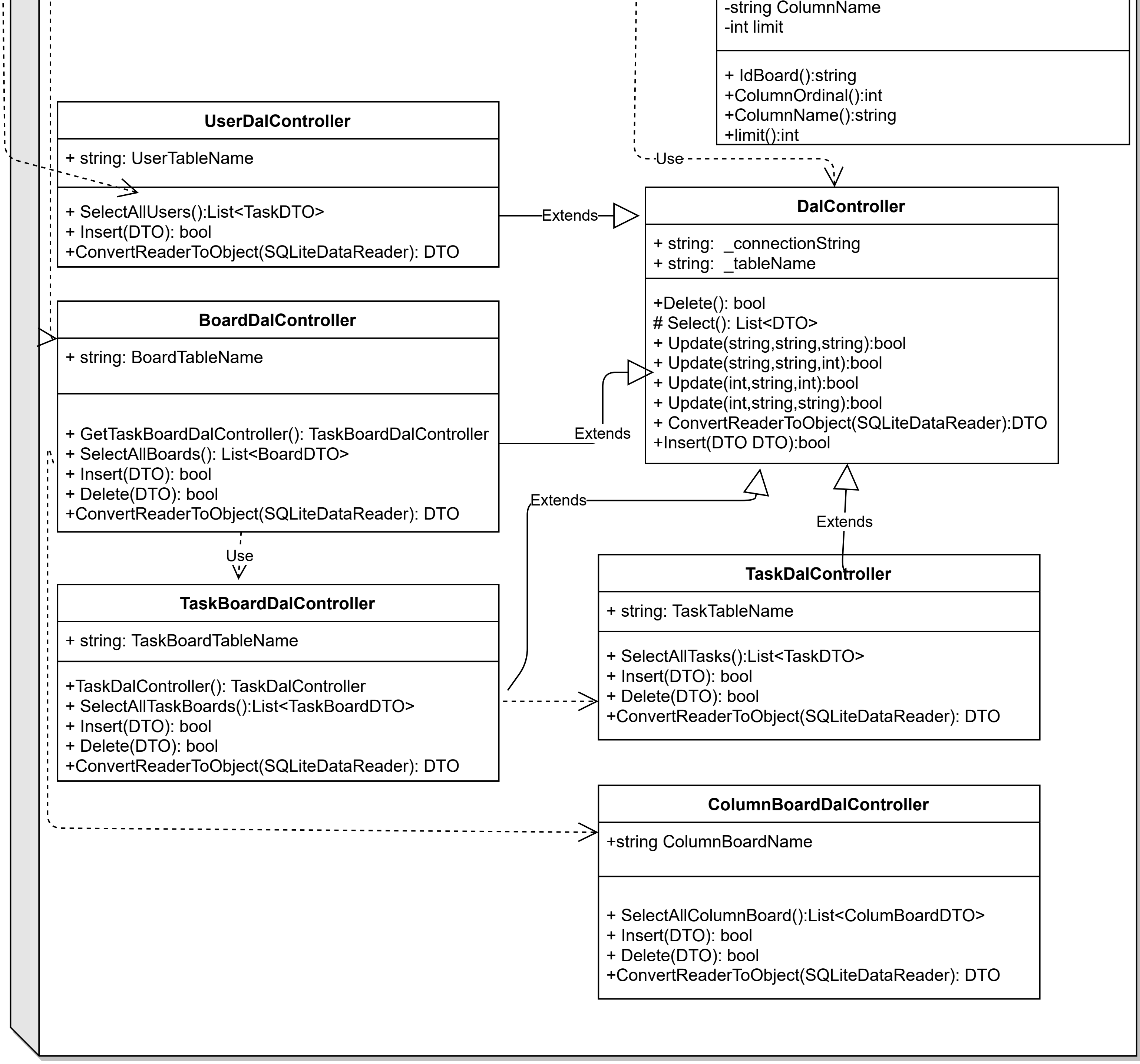
+ Id:int
+CreationTime:datetime
+Title:string
+Describtion:string
+DueDate:datetime
+ emailAssignee: string

Board

+ name:string
+ email: string
+Backlog:Column
+ inProgress:Column
+ done:Column
+BoardMember:List<string> BoardMember

Business Layer


User
+email:string
-password:string
-oldPassword:list<string>
-loggedIn:bool
+Email():string
+Loggin():bool
-compareString(pass1,pass2):bool
+changePassword(old,new):void
+validatePasswordMatch(pass):bool
-validatePasswordRules(pass):bool
-validateEmail(email):bool
+logout():void
+validnot20top(pass):bool
1
0..n
userControler
-Users:Dictionary<string,User>
-userOnline:bool
+GetUser(email):user
+Register(email,password):void
+Login(email,password):user
-ExistEmail(email):bool
+SetUserOnline(email):void
+LoadData():void
+DeleteUserController():void
BoardControler
-boards:Dictionary<IdBoard,Board>
+GetBoard(string,string,string,UserController): Board
-GetBoardEmail(idboard):String
-GetBoardName(idBoard):String
-CheckoutBoard(idboard):bool
+AddBoard(email,name,userController): void
+RemoveBoard(userEmail,creatorEmail,name, UserController):void
+GetListInProgress(userEmail,userController):list<task>
+GetBoardNames(userEmail, UserController):list<task>
-CheckIfUserLogIn(email,userController):bool
+JoinBoard(userEmail,creatorEmail,name,UserController):void
+DeleteBoardController():void
+ LoadData():void
1
Task
+id:int
+creationTime:datetime
+dueDate:datetime
+title:string
+describtion:string
-counter:int
-emailAssignee:string
~maxtitlelen:int
~maxlendescription:int
+Id():int
+Title():string
+Describtion():string
+CreationTime():datetime
+DueDate():datetime
+emailAssignee():string
+legalTitle(title):void
+legaldescription(description):void
+legalDueDate(dueDate):void
+setTitle(newTitle,email):void
+setDueDate(newDueDate,email):void
+setDescription(newDescription,email):void
Board
-name:string
+email:string
-columns.linkedlist<column>
+BoardMember:list<sring>
0..n
+Name():string
+Email():string
-Backlog():column
-InProgress(): Column
-Done(): Column
+BoardMember():list<sring>
+GetBoardEmail(string):string
+GetBoardName(string):string+GetColumn(id):Colum
+AddTask(DateTime,string,string, string):task
+moveTask(task,string,int):void
+SetBoardMember(string):void
+SetColumnLimit(int, int):void
~CheckIfMember(string):bool
+SetTaskAssign(int, string,int):void
+GetIdBoard():string
+GetMembers():string
+AddColumn(int,string):void
+RemoveColumn(int):void
+RenameColumn(int,string):void
+MoveColumn(int,int):void
+LoadData(Dictionary<int, Column>, Dictionary<int, ColumnBoardDTO> , List<TaskDTO>, List<TaskBoardDTO>):void
Text
Column
+id: int
~name: string
-maxTaskLimit:int
-dictTask: dictionary<int,Task>
0..n
1
+ Id():int
~Name():String
+maxTaskLimit():int
+GetTask(id):Task
- CheckIfColumnDone():void
+SetTitleTask(string,int,String):void
+setDescriptionTask(String,int,string):void
Use
+setnewDueDateTask(DateTime,int,string):void
+setEmailAssignee(int, string):void
+removeTask(int):void
+AddTaskToD(Task):void
+checkIfTaskExist(int):bool
Data Access Layer
TaskDTO
+ string: TaskCreationTimeColumnName
+ string: TaskDueDateColumnName
+ string: TaskTitleColumnName
+ string: TaskDescriptionColumnName
+ string: TaskAssignColumnName
+int: idTask
- DateTime: creationTime
- string: title
- string: description
- DateTime: dueDate
- string: assign
+IdTask():int
+CreationTime():DateTime
+Title():string
+Description():string
+DueDate():DateTime
+Assign():string
DTO
+ string :IDColumnName
+Insert():void
Extends
Extends
Extends
Extends
Extends
BoardDTO
+ string: BoardMemberColumnName
+string: idBoard
-string: boardMember
-ColumnBoardDTO
+IdBoard():string
+BoardMember(): string
TaskBoardDTO
+ string: TaskBoardBoardIdColumnName
+ string: TaskBoardColumnIdColumnName
~ int: idTask
- string: idBoard
- int: idColumn
+ IdTask():int
+ IdColumn():int
+ IdBoard():string
UserDTO
+ string: UserPasswordColumnName
+string: _email
+string: _Password
+ GetEmail():string
+GetPassword():string
ColumnBoardDTO
+ string: ColumnBoardIDBoardColumnName
+ string: ColumnBoardColumnOrdinalColumnName
+ string: ColumnBoardColumnNameColumnName
+ string:ColumnBoardlimitColumnName
+string IdBoard
-int ColumnOrdinal



-string ColumnName
-int limit
+ IdBoard():string
+ColumnOrdinal():int
+ColumnName():string
+limit():int
Use
UserDalController
+ string: UserTableName
+ SelectAllUsers():List<TaskDTO>
+ Insert(DTO): bool
+ConvertReaderToObject(SQLiteDataReader): DTO
Extends
DalController
+ string: _connectionString
+ string: _tableName
+Delete(): bool
# Select(): List<DTO>
+ Update(string,string,string):bool
+ Update(string,string,int):bool
+ Update(int,string,int):bool
+ Update(int,string,string):bool
+ ConvertReaderToObject(SQLiteDataReader):DTO
+Insert(DTO DTO):bool
BoardDalController
+ string: BoardTableName
+ GetTaskBoardDalController(): TaskBoardDalController
+ SelectAllBoards(): List<BoardDTO>
+ Insert(DTO): bool
+ Delete(DTO): bool
+ConvertReaderToObject(SQLiteDataReader): DTO
Extends
Extends
Extends
Use
TaskBoardDalController
+ string: TaskBoardTableName
+TaskDalController(): TaskDalController
+ SelectAllTaskBoards():List<TaskBoardDTO>
+ Insert(DTO): bool
+ Delete(DTO): bool
+ConvertReaderToObject(SQLiteDataReader): DTO
TaskDalController
+ string: TaskTableName
+ SelectAllTasks():List<TaskDTO>
+ Insert(DTO): bool
+ Delete(DTO): bool
+ConvertReaderToObject(SQLiteDataReader): DTO
ColumnBoardDalController
+string ColumnBoardName
+ SelectAllColumnBoard():List<ColumBoardDTO>
+ Insert(DTO): bool
+ Delete(DTO): bool
+ConvertReaderToObject(SQLiteDataReader): DTO