

Service Layer

TaskService

- _boardFacade: BoardFacade

+ AddTask(email, bn, title, desc, due): Response<>
+ AdvanceTask(email, bn, column, ti): Response<>
+ UpdateTask(email, bn, column, ti, due, title, desc): Response<>
+ AssignTask(email, bn, column, ti, emailAssignee): Response<>

TaskSL

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

BoardService

- _boardFacade: BoardFacade

+ CreateBoard(email, bn): Response<>
+ DeleteBoard(email, bn): Response<>
+ LimitColumn(email, bn, column, limit): Response<>
+ GetColumnTasks(email, bn, column): Response<List<TaskSL>>
+ GetColumnLimit(email, bn, column): Response<int>
+ GetColumnName(email, bn, column): Response<string>
+ InProgressTasks(email): Response<List<TaskSL>>
+ JoinBoard(email, bi): Response<>
+ LeaveBoard(email, bi): Response<>
+ TransferOwnership(owner, newOwner, bn): Response<>
+ GetUserBoardsAsId(email): Response<List<int>>
+ GetBoardName(bi): Response<string>
+ LoadData(): Response<>
+ DeleteData(): Response<>

BoardSL

+ Name: string

UserSL

+ Password: string
+ Email: string

UserService

- _userFacade: UserFacade

+ Login(email, pass): Response<string>
+ Register(email, pass): Response<>
+ Logout(email): Response<>

ServiceFactory

- _taskService: TaskService
- _boardFacade: BoardFacade
- _userFacade: UserFacade
- _boardService: BoardService
- _userService: UserService

+ GetUserService(): UserService
+ GetBoardService(): BoardService
+ GetTaskService(): TaskService

Response

+ ErrorMessage: string
+ ReturnValue: object

שינויים מאז העיצוב הקודם:

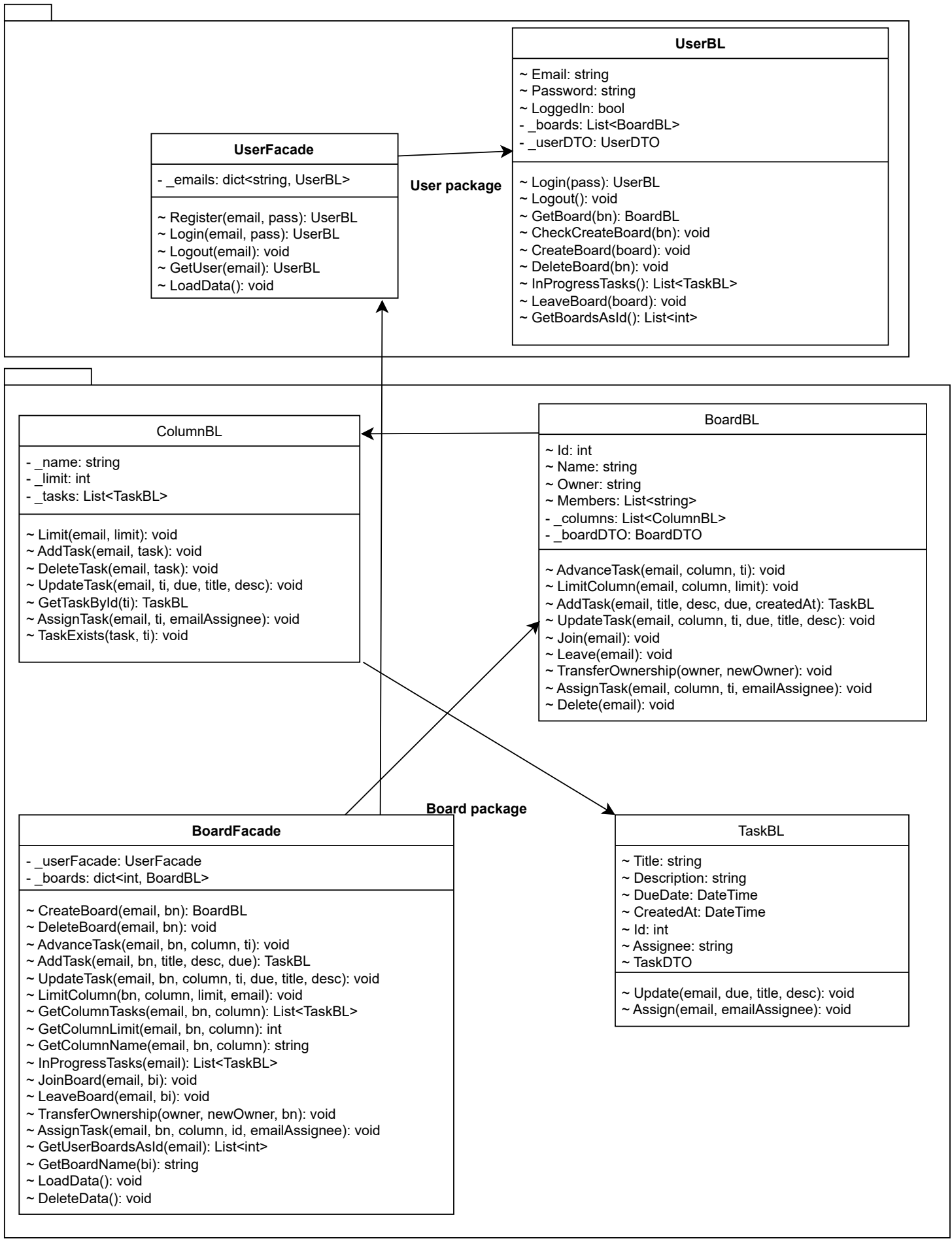
Access modifiers were changed based on the structure of the project and connection between classes, functions and fields.

Changed structure of ServiceLayer, BuisnessLayer and DataAccessLayer.

Enhanced code using OOP principles.

Changed tables in SQL to match project requirements.

Business Layer



Data Access Layer

BoardUserDTO : IDTO

~ Id: int
~ Email: string
- _controller: BoardUserController

~ Insert(): void
~ Delete(): void
~ GetParticipants(): List<string>
~ GetBoards(): List<int>

BaseController<TDTO> where TDTO : IDTO

_connectionString: string
_tableName

~ Insert(TDTO dto): void
~ SelectAll(): List<TDTO>
~ DeleteAll(): void
~ DeleteAllAndResetAutoIncrement(): void
ExecuteQuery(command, parameterAction, operationName): void
ConvertReaderToDTO(reader): TDTO {abstract}

BoardController : SingleKeyController<BoardDTO>

~ GetNextId(): int

BoardUserController : CompositeKeyController<BoardUserDTO>

~ GetParticipants(): List<string>
~ GetBoards(): List<int>

CompositeKeyController<TDTO> : BaseController<TDTO> where TDTO : IDTO

~ Delete(keyColumn1, key1, keyColumn2, key2): void
~ Update(keyColumn1, key1, keyColumn2, key2, column, newValue): void

ColumnDTO

~ Limit: int
~ Tasks: List<TaskDTO>
- _controller: TaskController

~ AddTask(task): void
~ RemoveTask(task): void

UserDTO : IDTO

~ Email: string
~ Password: string
~ LoggedIn: bool
- _controller: UserController

~ Insert(): void
~ SelectAll(): void

IDTO Interface

+ Update(column, newValue): void
+ GetColumnNames(): string[]
+ GetColumnValues(): object[]

BoardDTO : IDTO

~ Id: int
~ Owner: string
~ Name: string
~ Columns: List<ColumnDTO>
- _controller: BoardController

~ AddTask(task, email): void
~ AdvanceTask(task, email, column): void
~ LimitColumn(limit, column): void
~ Delete(): void
~ SelectAll(): List<BoardDTO>

TaskDTO : IDTO

~ Id: int
~ BoardId: int
~ Assignee: string
~ Title: string
~ Description: string
~ DueDate: DateTime
~ CreatedAt: DateTime
~ Column: int
- _controller: TaskController

~ Insert(): void
~ Delete(): void

SingleKeyController<TDTO> : BaseController<TDTO> where TDTO : IDTO

~ Delete(keyColumn, key): void
~ Update(keyColumn, key, column, newValue): void

TaskController : CompositeKeyController<TaskDTO>

~ GetNextId(): int

UserController : SingleKeyController<UserDTO>

