**Business Layer**

**BoardController**

- IDBoardList: Dictionary<string, Dictionary<string, int>>
- BoardList: Dictionary<int, Board>
- BoardID: int
- Tasks : Dictionary<int, Task>
- TaskID: int
- dalController: BoardDalController

+loadAllBoards(): void
+DeleteAllData(): void
+LimitColumnOfBoard(creatorEmail,boardName,columnOrdinal,limit): void
+GetColumnLimitOfBoard(creatorEmail,boardName,columnOrdinal): int
+getColumnNameOfBoard(columnOrdinal): string
+ AddBoard(email, boardname): void
+ RemoveBoard(userEmail, creatorEmail, boardName): List<int>
+AddBoardMember(userEmail, creatorEmail, boardName):void
+GetBoardMembers(creatorEmail, boardName): List<string>
+ AddTaskToBoard(creatorEmail,boardName,taskID): void
+ AdvanceTask(creatorEmail,boardName,columnOrdinal,taskID): void
+ GetListOfTaskFromColumn(creatorEmail,boardName,columnOrdinal): List<int>
+ GetBoardIdByNameAndEmail(creatorEmail,boardName): int
+ GetBoardNameById(boardID): string
+ ValidateUserIsBoardMember(userEmail, creatorEmail, boardName):void
+ ValidateUserIsNOTBoardMember(userEmail, creatorEmail, boardName):void
+ValidateBacklogNotFull(creatorEmail, boardName): void
+CreateTask(userEmail, creatorEmail, boardName, title, description, dueDate): Task
+InProgressTasks(boards, userEmail): List<Task>
+ GetTaskByID(taskId): Task
+ UpdateTaskDueDate(creatorEmail, boardName, columnOrdinal, taskId, newDueDate): void
+ UpdateTaskTitle(creatorEmail, boardName, columnOrdinal, taskId, newTitle): void
+ UpdateTaskDescription(creatorEmail, boardName, columnOrdinal, taskId, newDescription): void
+UpdateTaskAssignee(creatorEmail, boardName, columnOrdinal, taskId, emailAssignee): void
+DeleteTask(taskID):void
+ValidateUserIsAssigneeOfTask(userEmail, taskId):void
+AddColumn(creatorEmail, BoardName, columnOrdinal, ColumnName): void
+ RemoveColumn(creatorEmail, BoardName, columnOrdinal): void
+ RenameColumn(creatorEmail, BoardName, columnOrdinal, newColumnName): void
+MoveColumn(creatorEmail, BoardName, columnOrdinal, siftSize): void

**UserController**

- users: Dictionary<string, User>
- dalController: UserDalController

+loadAllUsers(): void
+DeleteAllData(): void
+ Register(email, password): void
+ Login(email, password): string
+ LogoutUser(email): void
+ RemoveBoardFromUserList(email, boardID): void
+ GetBoardsOf(email): List<int>
+ AddBoardToUserList(boarID, email): void
+ GetUser(email): User
+ ValidateUserExists(email): void
+ ValidateUserLoggedIn(email): void

1

contain

0.. n

**User**

- logged: bool
- email: string
- pass: string
-MIN_PASS_LENGTH: int
-MAX_PASS_LENGTH: int
- Boards: List<int>

+ logout(): void
+ IsMatchPassword(password): bool
+ GetBoards(): List<int>
+ AddBoardToUserList(boardID): void
+ RemoveBoardFromUserList(boardID): void

1

1

-1

creator

assinee

0.. n

1

contain

**Board**

- name: string
- creator_email: string
- column: List<Column>
- boardID: int
- boardMembers: List<string>
- MAX_COLUMN_ORDINAL: int
- MIN_COLUMN_ORDINAL: int

+ GetColumnLimit(columnOrdinal): int
+AddTask(taskID): void
+ AdvenceTask(columnOrdinal, taskid): void
+ IsColumnFull(columnOrdinal): bool
+ValidateUserIsBoardMember(userEmail):void
+ValidateUserIsNOTBoardMember(userEmail):void
+ LimitColumn(columnOrdinal, limit): void
+GetListOfTaskFromColumn(columnOrdinal):List<int>
+GetAllTasks(): List<int>
+AddBoardMember(userEmail):void
+GetBoardMembers(): List<string>
+insertTask(TaskID, ColumnOrdinal):void
+reinstateLimit(columnOrdinal, limit):void
+GetTasksOf(columnID): List<int>
+ValidateTaskExistsInColumn(columnOrdinal,taskID):bool
+AddColumn(columOrdinal, ColumnName): void
+RemoveColumn(columnOrdinal): void
+RenameColumn(columOrdinal, newColumnName): void
+moveColumn(columnOrdinal, siftSize): void

0.. n

1

contain

0.. n

**Task**

- id: int
- CreationTime: DateTime
- Title: string
-MAX_TITLE_LENGTH
- Description: string
-MAX_DESCRIPTION_LENGTH
- DueDate: DateTime
- EmailAssignee: String

+ UpdateTaskDueDate(newDueDate): void
+ UpdateTaskTitle(newTitle): void
+ UpdateTaskDescription(newDescription): void
+ UpdateTaskAssignee(emailAssignee): void
+ checkAssignee(userEmail): bool

3

*Column*

- maxTasks: int
- tasksList: List<int>
- name: string
- boardID: int
- columnOrdinal: int
- columnID: int

+ LimitColumn(limit): void
+reinstateLimit( limit):void
+insertTask(TaskID):void
+ GetColumnLimit(): int
+ AddTask(taskId): void
+ DeleteTask(taskId): void
+checkTaskExists(taskID):bool
+ GetTaskList(): List<int>
+ isColumnFull(): bool

---

**Data Acceses Layer:**

*DTO*

+ _controller

**Dtask**

- TaskID: int
- CreationTime: string
- Title: string
- Description: string
- DueDate: string
- emailAssignee: String
- boardID: int
-columnOrdinal: int

**Dboard**

-BoardID: int
-Name: string
-Creator_email: string

**Dcolumn**

- boardID: int
- columnOrdinal: int
- name: string
- limit: int
- columnID: int

**Duser**

- email: string
- pasword: string

**BoardDalController**

+SelectAllBoards(): List<Dboard>
+SelectAllColumns(): List<Dcolumn>
+SelectAllTasks(): List<Dtask>
+SelectAllBoardMembers(): Dictionary<int, List<string>>
+InsertBoard(dboard): void
+InsertNewTask(dtask): void
+deleteAllTables(): void
+deleteBoard(boardID): void
+deleteTask(taskID): void
+InsertNewColumn(Dcolumn): void
+deleteColumn(boardID, columnOrdinal): void

**UserDalController**

+SelectAllUsers(): List<Duser>
+SelectAllBoardMembers(): Dictionary<string, List<int>>
+InsertUser(duser): void
+InsertBoardToMembers(email, boardID): void
+deleteAllTables(): void

*DalController*

+Update(tableName, keyID, ID, attributeName, attributeValue:string): void
+Update(tableName, keyID, ID, attributeName, attributeValue:int): void
+select(tableName): List<DTO>
+ ConvertReaderToObject(SQLiteDataReader): DTO
+ Delete(tableName, keyID, ID): void
+DeleteAllTable(tableName): void
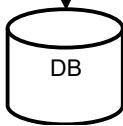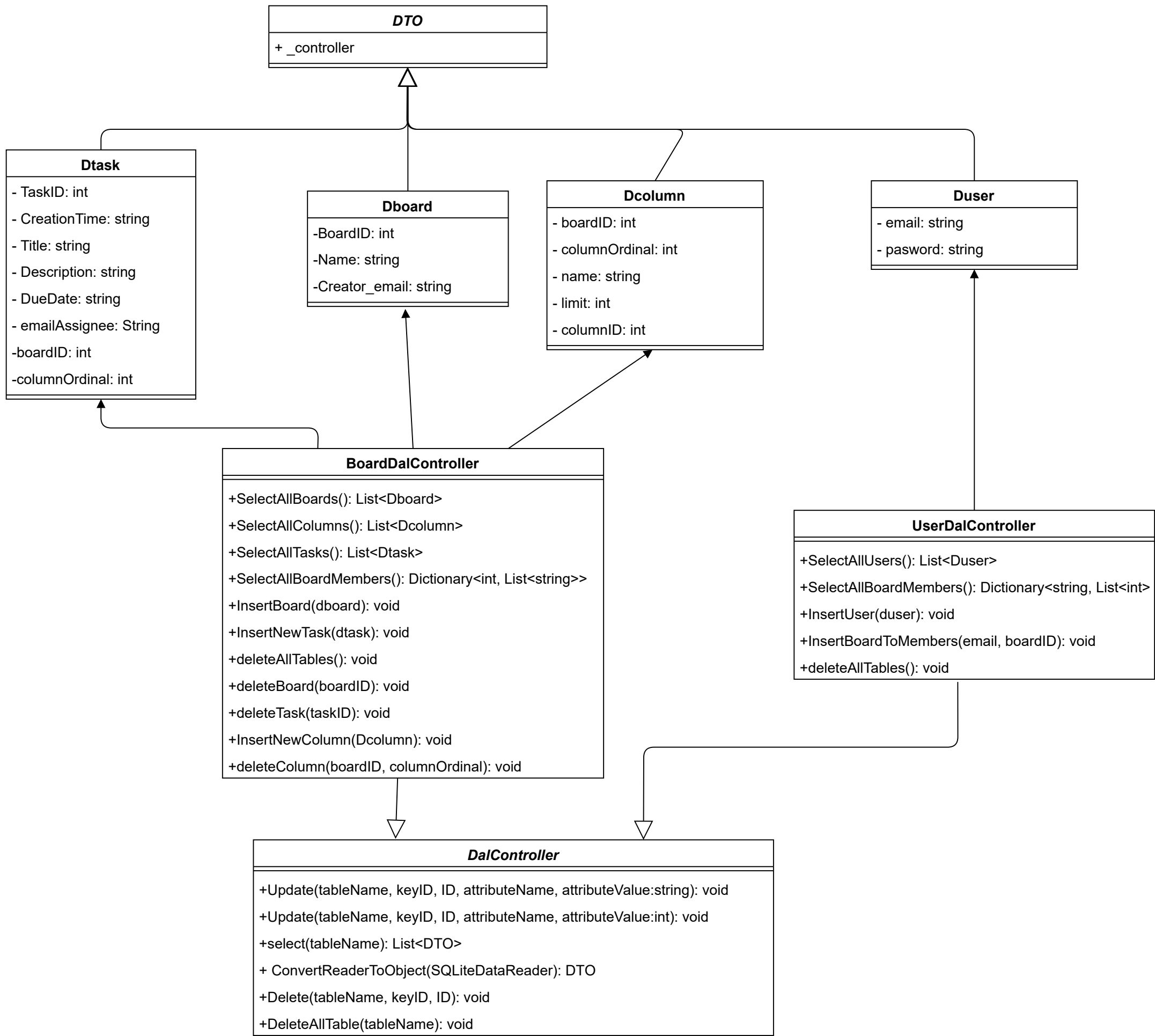
DB

# Modifications

## Board

1. Board now holds a dynamic list of columns instead of array.
2. Board now has a few functions to manage columns as required in milestone 3.

## BoardController

1. We added a few functions to add, remove and move columns as required

## Column

1. Column now has a name, and the column holds it's column ordinal.

## Dal

1. We added a DColumn object in the data layer.