

Created:

- IsLegalPassword- in order to validate the password
- IsValidEmail- To validate the email address at registration
- IsHebrew - to validate email
- HasAnyBoard- Check if a BoardController is empty - for exception handling purposes
- HasThisBoard- Checks if a BoardController has a board - for exception handling purposes
- Added getters and setters in Board
- GetCollist- in order to get all tasks in a given column in a board
- GetNameOrdinal - Returns the name of a column given its id in order to fulfill GradingService requirements
- GetAllInProgress - See design changes for explanation

A User object no longer holds a collection of their boards and Tasks.

The class user is therefore connected only to the UserController, which limits our duality.

The checks in order to get a user's board and tasks are preformed through calling an instance of

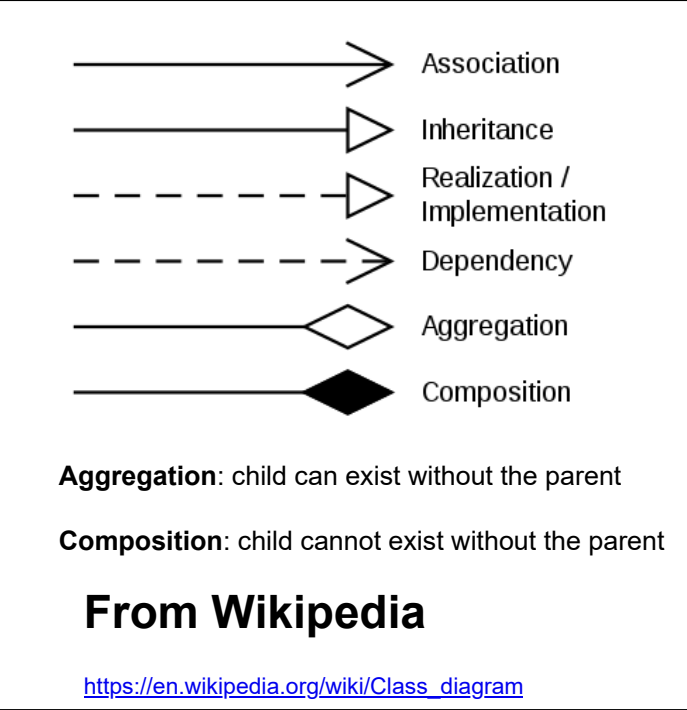
`UserController.IsLoggedIn()`. This allows a better (lower) duality between the classes.

Setting the max number of tasks in now done from the Board (rather than Task). This is done because it is an attribute of a board rather than a task. It can be said that initially we made a design mistake.

AddTask - same as max tasks.

new methods for all new functional requirements. Data Access layer DTO's and Mappers.

one

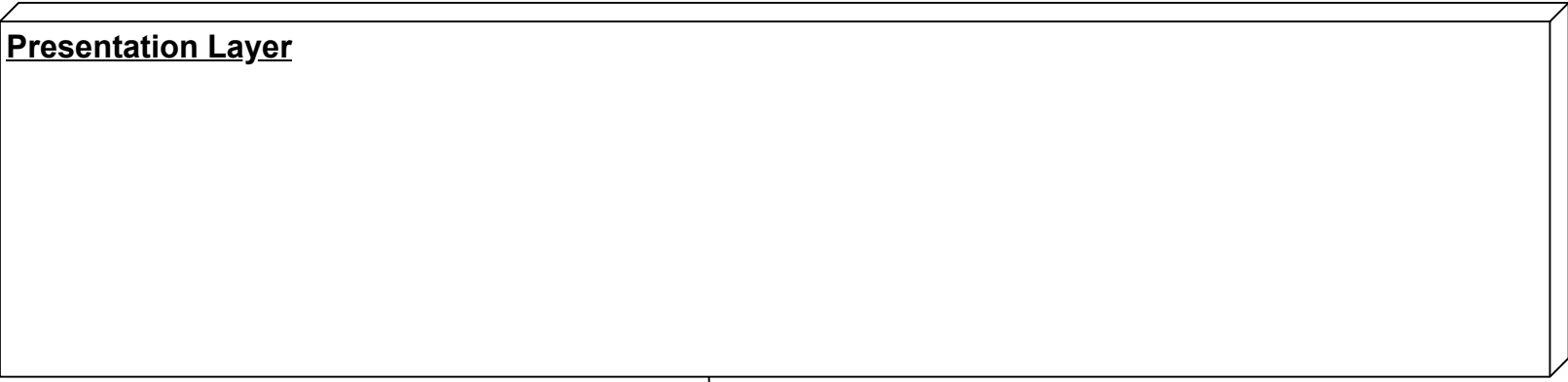


+ public

+ public

- private

protected



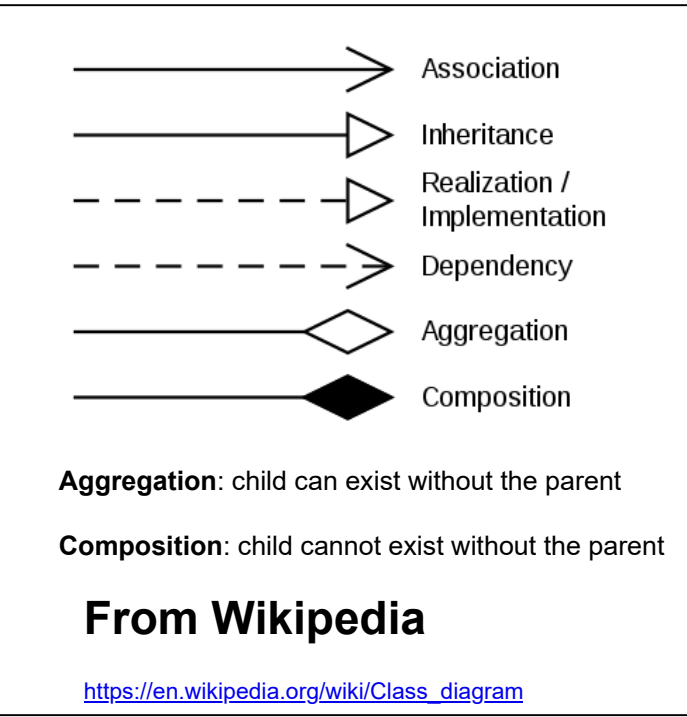
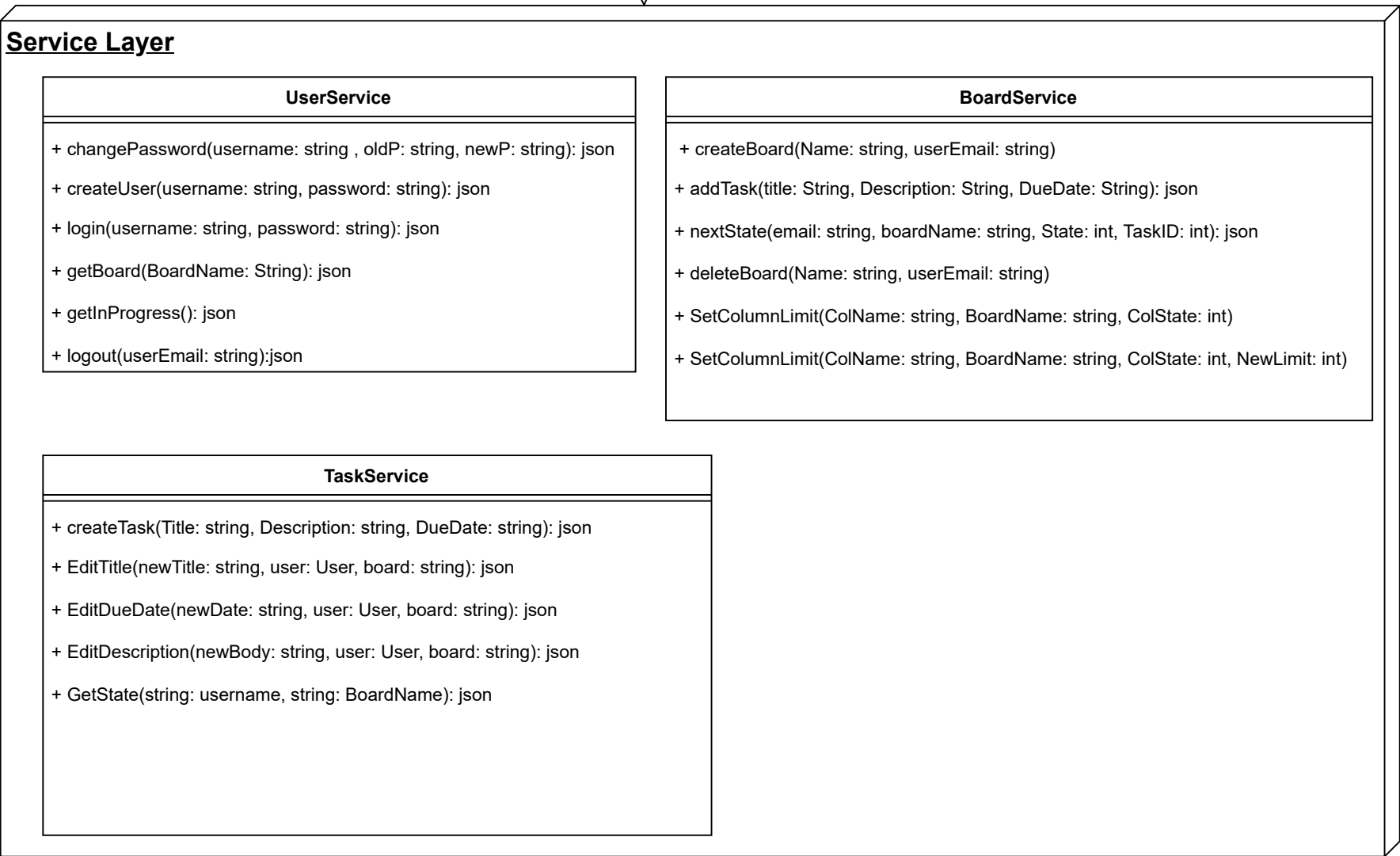
Changes Made after submitting Milestone 1:

Created:
IsLegalPassword- in order to validate the password
IsValidEmail- To validate the email address at registration
IsHebrew - to validate email
HasAnyBoard- Check if a BoardController is empty - for exception handling purposes
HasThisBoard- Checks if a BoardController has a board - for exception handling purposes
Added getters and setters in Board
GetColList - in order to get all tasks in a given column in a board
GetNameOrdinal - Returns the name of a column given its id in order to fulfill GradingService requirements
GetAllInProgress - See design changes for explanation

Design changes:

A User object no longer holds a collection of their boards and Tasks.
The class user is therefore connected only to the UserConroller, which limits our duality.
The checks in order to get a user's board and tasks are preformed through calling an instance of
UserController.IsLoggedIn(). This allows a better (lower) duality between the classes.

Setting the max number of tasks in now done from the Board (rather than Task).This is done because
it is an attribute of a board rather than a task. It can be said that initially we made a design mistake.
AddTask - same as max tasks.



Visibility

