

Design Changes – milestone 1

Service Layer:

- We added User&Board Facade as fields in both BoardService and TaskService, and in UserService only UserFacade – so UserService only handles users registration, users log in&out, and check if the given parameters are valid, and BoardService&TaskService take care to all of the rest of the operations.
- We added a SystemService Class, which has as fields all the other services and facades, so they are all initiated only once in this class code and it organized all of the data of our system and made an easier access to it.
- We created a special KanException class (which extends the default exception) which supposed to specify the exception we expect to have in our program.
- We added a GetLimitOnColumn() method in BoardFacade, which gets a column as int and returns its limit (if exists) and -1 if there is no limit, and also a setLimitOnColumn() method.

Business Layer:

- We added fields named with uppercase letters only to avoid magic numbers.
- We added a <string, user> Dictionary as a field to UserFacade, so we can map an e-mail address (which is unique) to a user's object.
- We added a <string, Dictionary<string, Board>> Dictionary as a fields ro BoardFacade, so we can map users (by their unique e-mail address) to their List of Boards (implemented as a dictionary to ease the search for a specific board).
- We added a CheckForExceptions() method in BoardFacade, that checks validity before making some data operations.
- We added a CanAddTo() method in Board, that checks before adding a task that after adding it we will not pass its limit.

- We changed all of the check validity methods (check email, password, login etc.) and the changePassword methods from boolean to void, because it was unnecessary.
- We added int fields to represent the 3 kinds of columns (backlog, in progress, done).
- We added a Board.CanAddTo(int column) method to check if it's possible to add a new task to a board (or advance a task).
- We added to Board a field of array of <int, Task> dictionaries, so each Board can manage its tasks comfortably.