Task Documentation

Marina Gluzberg
26.04.2018

SQL Structure

The given structure was modified as follows:

- The *pool* table was excluded and alternatively marking of the active/online is done directly on the *article* table by extending it with a BIT field that indicates if the article is not available for locking.
- An additional table was introduced: *client_user*. This table contains the users' credentials required for login to the virtual pool client.

Solution Structure

- **VirtualPool.Client**
  - This project contains the console application that manages the dialog with user and triggers the relevant actions of the *IPoolManager* from VirtualPool.Manager project.
- **VirtualPool.Data**
  - The DB context and models.
- **VirtualPool.Manager**
  - The *IPoolManager* interface that declares the following actions:
    - Login
    - Logout
    - BlockArticle
    - ReleaseArticle
- **VirtualPool.Manager.Data**
  - Contains the *PoolManager* class whoich implements the *IPoolManager* using the DB context from **VirtualPool.Data**.

Implementation Details

- At article locking – since all related articles (articles of the same product) must be locked as well – the locking is done on the product level directly.

Usage

- The program is run as a console application with the following parameters:
  - User ID
  - User Password
- If the provided credentials are missing / invalid, then the application shuts down.
- Otherwise, the user is prompted to enter and desired action (block or release) and a corresponding article ID.
- In case of invalid input or in case the specified article cannot be blocked or released by the current user – a corresponding error notice is displayed.
- In case of successful lock or release of an article – a corresponding message is displayed.
- To end the application – the user must enter the keywork *quit*.

Potential Improvements

Among others, the following can be seen of further steps of improvement if the current implementation:

- The password field in the *client_user* table should be encrypted.
- The initialization of the *PoolManager* in the main Program constructor can be done alternatively using dependency injection.