Overall Objective

Create the architecture and design of a medical journals publication and subscription system. Implement the system's services and applications.

Prerequisites

The following prerequisites should be respected.

1. Use the .NET technology stack for development along with any compatible open technologies.
2. Use any database and development environment tools according to the technology stack chosen.
3. Do not use any proprietary technologies or tools that are not available for evaluation.

Functional Requirements

The system allows publishing and subscribing to medical journals in a secure way. The system implements the following specifications.

1. For publishers
    1. A web portal to upload and manage a list of medical journals
    2. To upload journals in PDF format
2. For public users
    1. A web portal to find and subscribe to journals of their interest
    2. A desktop client
        1. To list and read the subscribed journals

2. That does not allow copying the journal's content in any common way
    1. Not from web requests/responses
    2. Not from their file system
    3. Not from the client user interface (copy to clipboard, screenshot, etc)

Assume any functional details required to achieve the above requirements based on logic and your experience, but follow the KISS principle.

Other Technical and Non-functional Requirements

The following list of technical specifications should be adhered to

1. Assume missing/unclear requirements to fill in the gaps in the specifications.
2. Choose a mix of server and client side technologies for an efficient design.
3. Secure the applications and services.
4. Use only software enforced protection that does not require any special hardware.

What we will evaluate

1. Efficacy of your submission: fundamentally how well your solution is able to achieve the assignment objective and solve the stated problem.
2. Code quality
    1. Code modularity
    2. Application organization across files and within each file - please ensure you follow the framework standards

        3. Code documentation - balancing between self documenting code and comments
        4. Unit and integration testing
        5. Exception handling where available and expected in the frameworks you're using
        6. For any technology used, the correct usage of that technology based on consecrated best practices
    3. Design
        1. Clarity and completeness of the readme and design documents
        2. Fitness of solution to problem
        3. Efficiency of communication flows between front-end and back-end, if applicable
    4. Functional completeness
    5. Scoring ratio matrix (out of 10), all of these are individually mandatory so don't skip any:
        1. Design quality = 2
        2. Code quality = 3
        3. Docs and demo quality = 2
        4. Specifications compliance = 3

Design Document

Create a design document containing the following

1. High level requirement analysis
2. High level presentation of the data model
3. Architecture diagrams describing the composition and working of the system, explaining the component interaction and process, control and data flows.

4. Explain the breakdown of the system into components with technical implementation details of each component along with the design patterns involved and with reasons that justify your choices.
5. Use both visual elements (diagrams) and text descriptions to maximize the amount of information conveyed while keeping the document as compact as possible

Source Code

You should deliver all the implemented source code including any dependencies. For the dependencies that could not be included due to size, the readme file should have proper instructions on how to download and install them.

ATTENTION! YOUR APPLICATION WILL BE REJECTED IF IT:

- does not compile;

- does not contain unit tests;

- contains failing unit tests.