

Assignment A3

Analysis and Design Document

Student: Nicolae-Florian Onica
Group: 30432

Table of Contents

1. Requirements Analysis	3
1.1 Assignment Specification	3
1.2 Functional Requirements	3
1.3 Non-functional Requirements	3
2. Use-Case Model	3
3. System Architectural Design	4
4. UML Sequence Diagrams	6
5. Class Design	6
6. Data Model	7
7. System Testing	7
8. Bibliography	8

1. Requirements Analysis

1.1 Assignment Specification

Use Java/C# API to design and implement a client-server application for a news agency. The application has 2 types of users: the readers and the writers. The readers can view a list of articles, read an article and do not need to login in order to use the application. The writers need to authenticate in order to create, update or delete articles. So, the writer accounts are preset by the application developer and cannot be altered. An article has the following components:

- Title
- Abstract
- Author
- Body

The application must support multiple concurrent users. If a writer posts a new article, the readers must see it in the list of articles in real time, without performing any refresh operation.

1.2 Functional Requirements

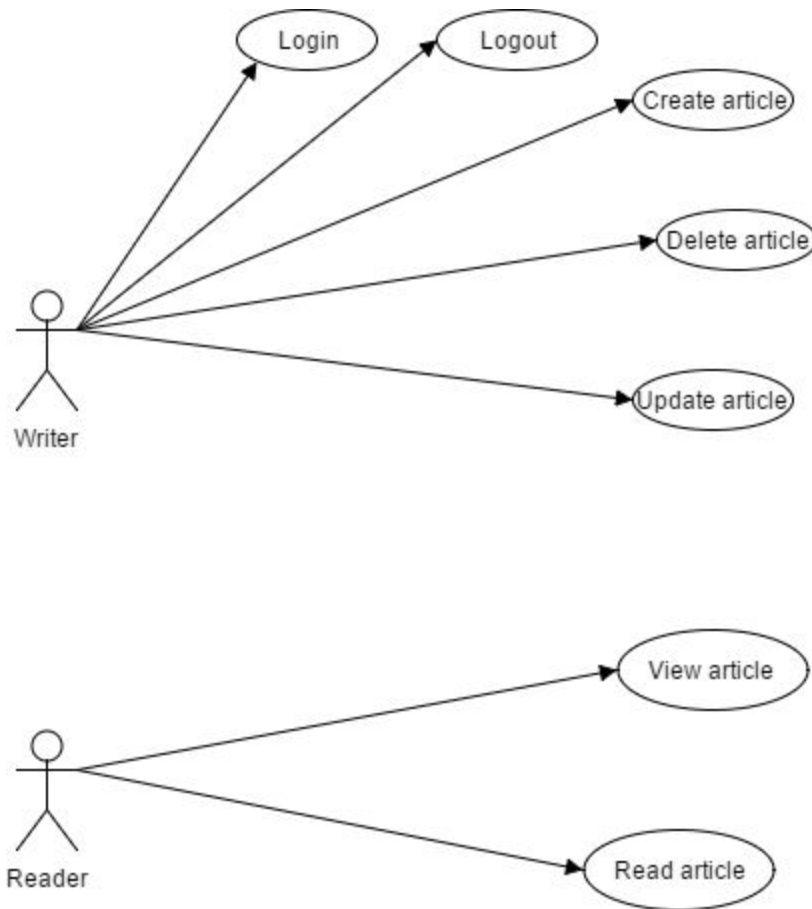
- The readers can view a list of articles, read an article and do not need to login in order to use the application.
- The writers need to authenticate in order to create, update or delete articles. So, the writer accounts are preset by the application developer and cannot be altered.
- The application must be client-server.
- Use the Observer design pattern for updating the list of articles in real time
- For sending data from the client to the server use JSON serialization.
- When writing an article, show a list that supports multi-select for choosing the related articles.

1.3 Non-functional Requirements

2. Use-Case Model

There are two actors in the system: the Reader, who is able to view and read a selected article and the Writer, who can to login/logout, and perform CRUD operations on articles.

Use case diagram:



Use case: update article

Level: user-goal level

Primary actor: writer

Main success scenario: The article is updated

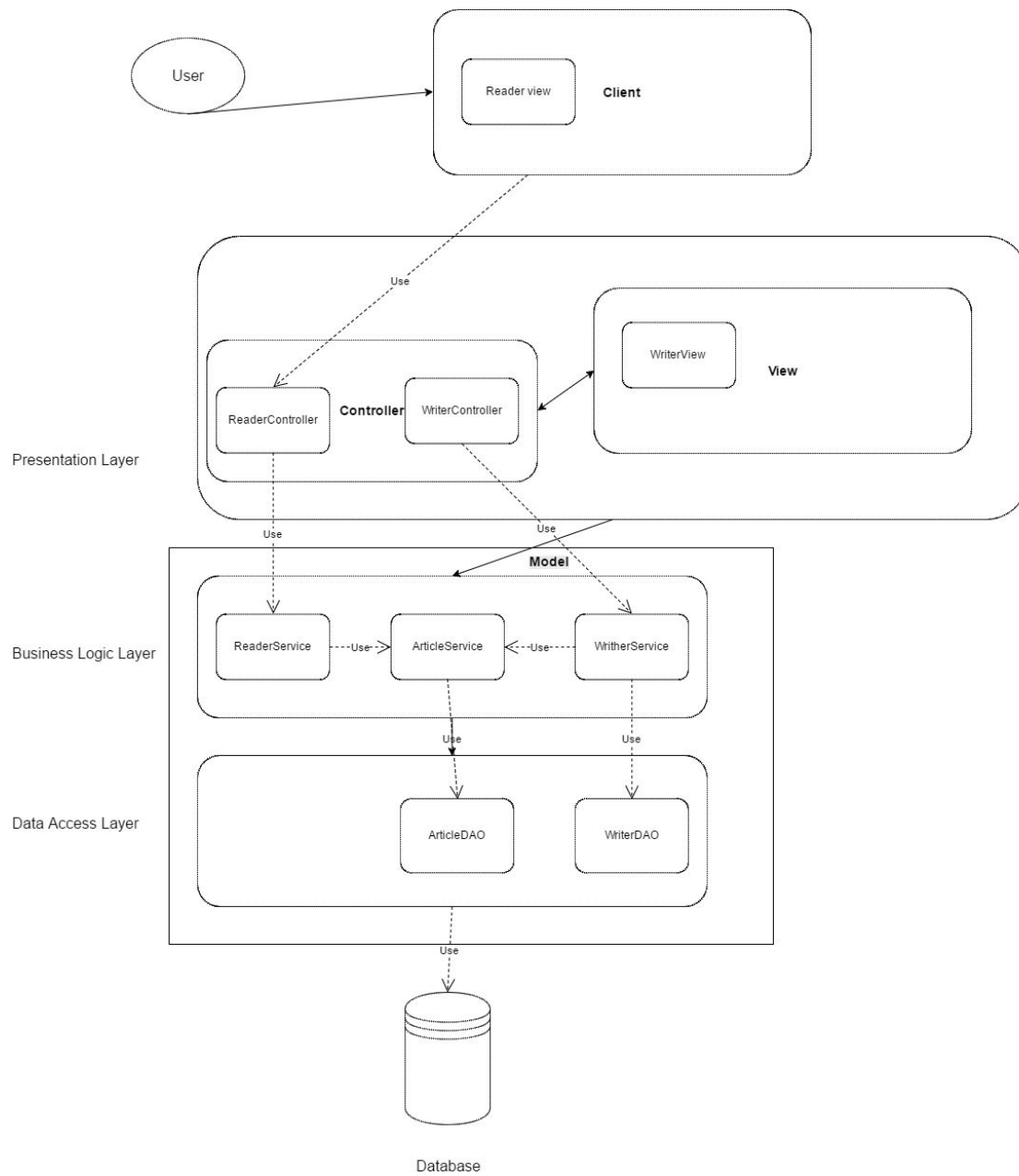
Extensions: Failure case: The server fails and the article will not be updated

3. System Architectural Design

3.1 Architectural Pattern Description

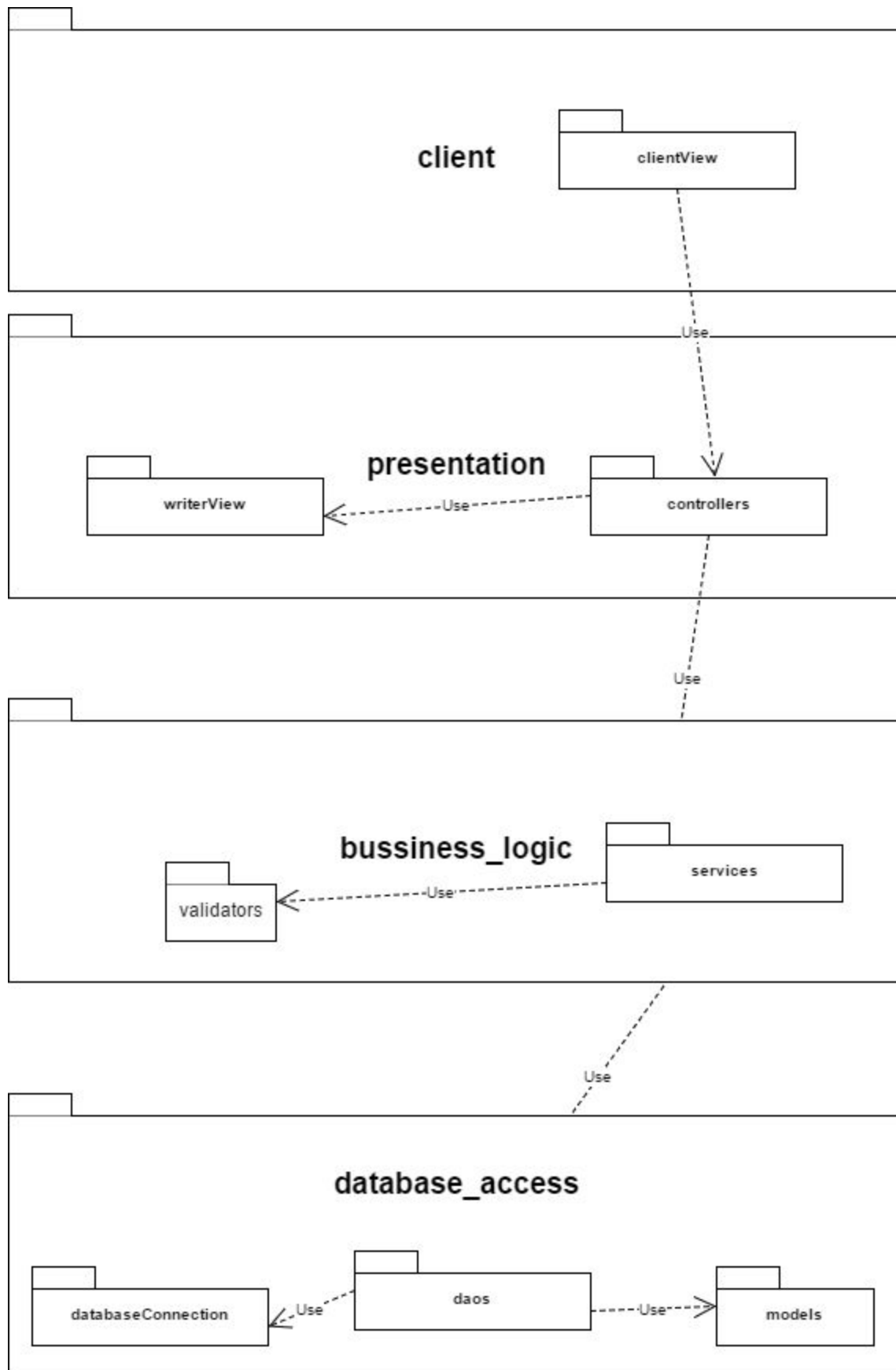
The main architectural pattern used in this system is the client-server architecture. It divides the system into two entities, the Client and the Server. The Client is used for user interaction and sends requests to the the server, which is processes the information and sends a response back to the Client, which is displayed to the user.

The other architectural pattern used for this application is the layered architecture pattern. The components within this pattern are organized into horizontal layers, each layer performing a specific role within the application (e.g., presentation logic or business logic). In this case three layers will be used: presentation, business and database layer.

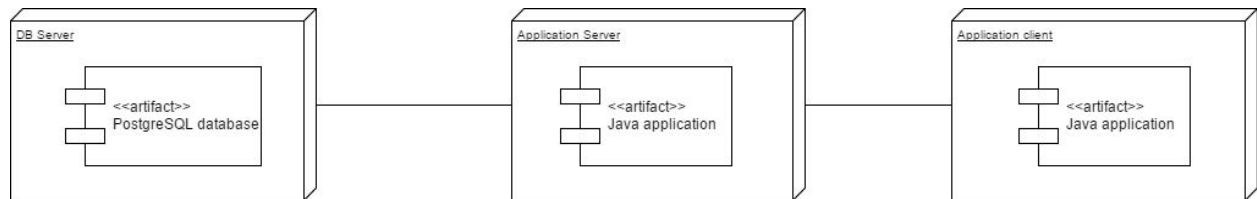


3.2 Diagrams

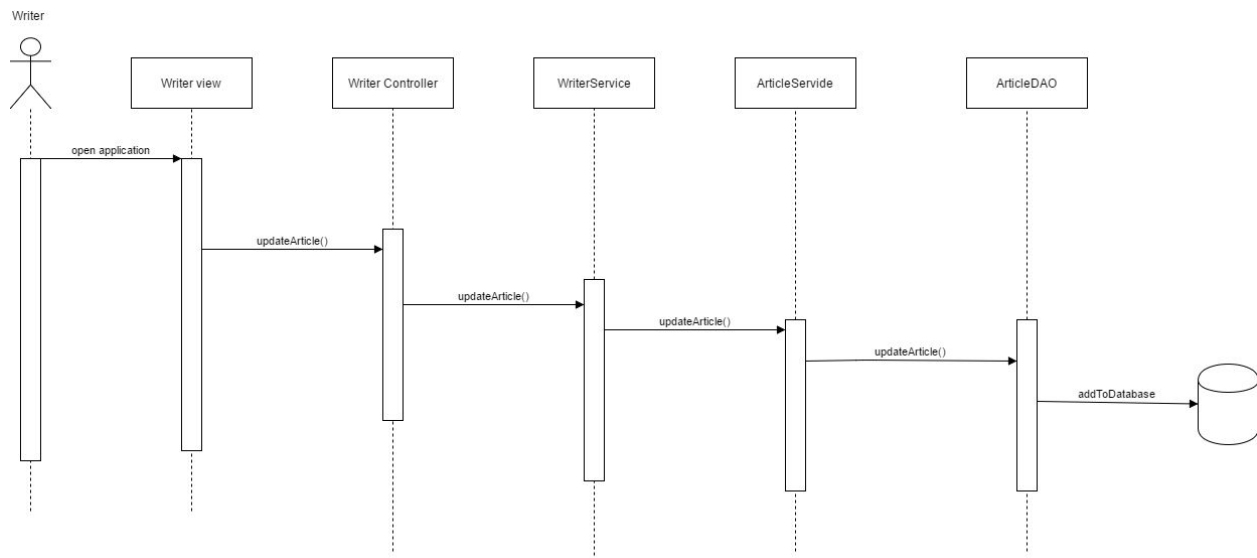
Package diagram



Deployment diagram



4. UML Sequence Diagrams

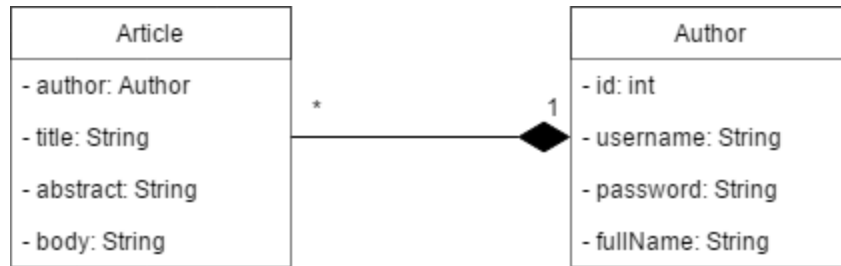


5. Class Design

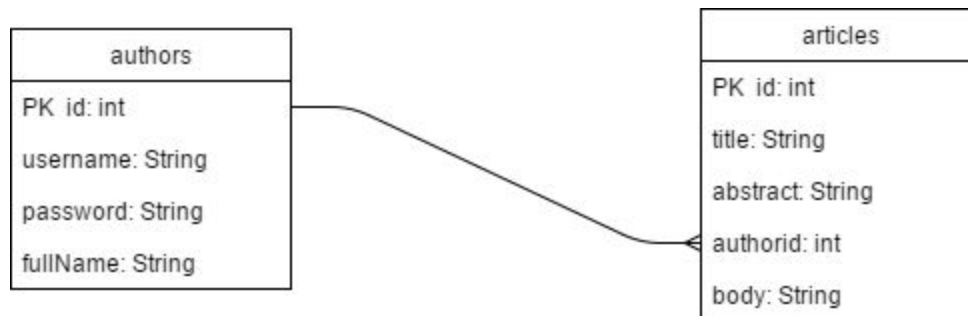
5.1 Design Patterns Description

The design pattern used in this application is the Observer pattern. It has two main components, an Observable class (in our case the article) which will be the subject of observation for the Observer interface (in this case the reader which should view updates in real time), which has an update method which is being called anytime the Observable updates.

5.2 UML Class Diagram



6. Data Model



7. System Testing

The testing strategy used is unit testing which implies separating the code into unit and testing each unit to see if they meet the required functionality.

8. Bibliography