Student: Turcu Lucian Andrei

**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 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

The assignment requires the implementation of a client-server application for a news agency that supports multiple concurrent users, which displays in real time a list of articles for the reader users.

# Functional Requirements

* The application must support 2 types of users: readers and writers.
* The readers can view a list of articles and read an article.
* The writers can perform CRUD operations on articles, after logging in. The writer accounts are preset by the developer.
* An article is composed of Title, Abstract, Author and Body.
* The application must run on a client-server architecture.
* The application must implement the Observer design pattern in such a way that the articles are updated in real time for the reader.

# Non-functional Requirements

* **Speed**: articles should load fast.
* **Security**: only authorized users must be able to perform CRUD operations.
* **Reliability**: the application must not crash or disrupt the service in any way.
* **Storage**: the application must hold a large number of articles.

2. Use-Case Model

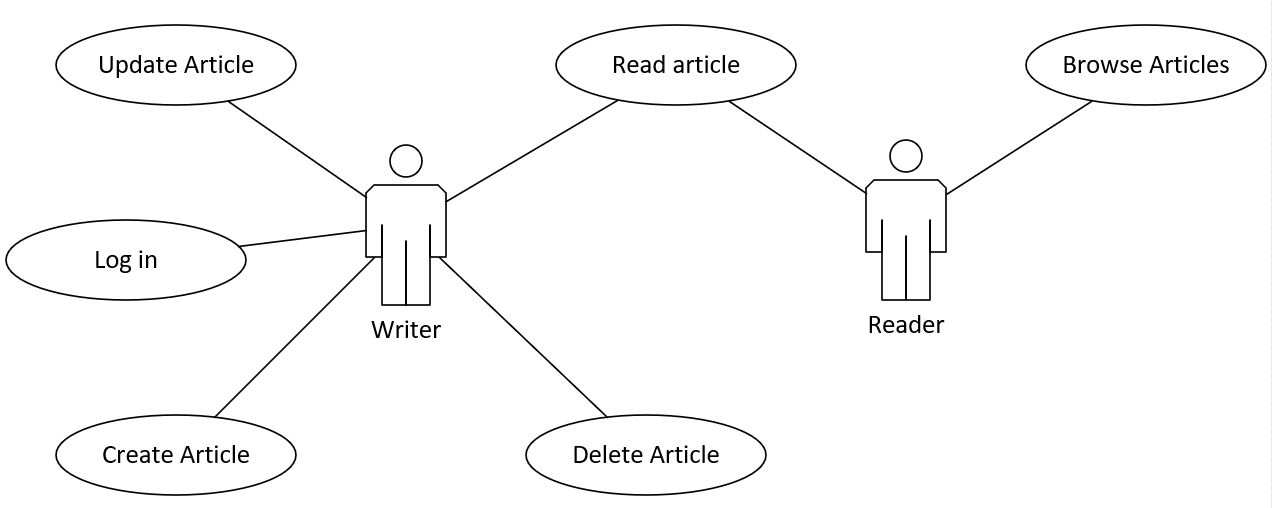
Use case: Read article

Level: User goal

Primary actor: Reader

Main success scenario: Open app, browse list, select article, read article, close article.

Extensions: Error occurred while retrieving article from database, try again or choose another.



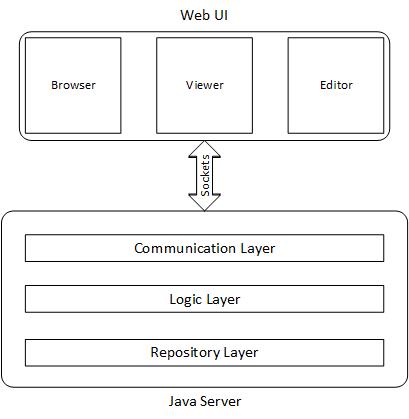
3. System Architectural Design

**3.1 Architectural Pattern Description**

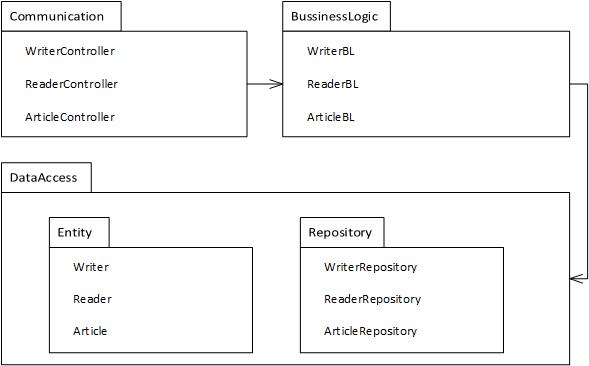
The client – server architecture distributes the tasks between the client and server components. The client performs the requests, while the server is the one that fulfills them. The server will be the Java API, and will communicate through sockets with an Angular web app. The server will also implement a layered architecture to create distinction between the different levels of functionality.

**3.2 Diagrams**

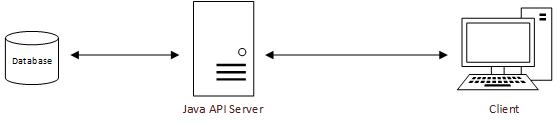
*<Architectural diagram>*



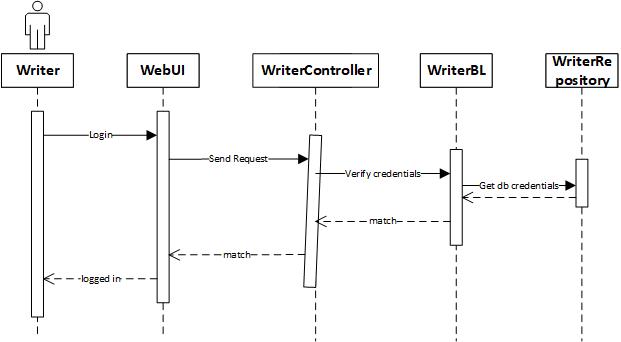
*<Package Diagram>*



*<Deployment Diagram>*

**

4. UML Sequence Diagrams

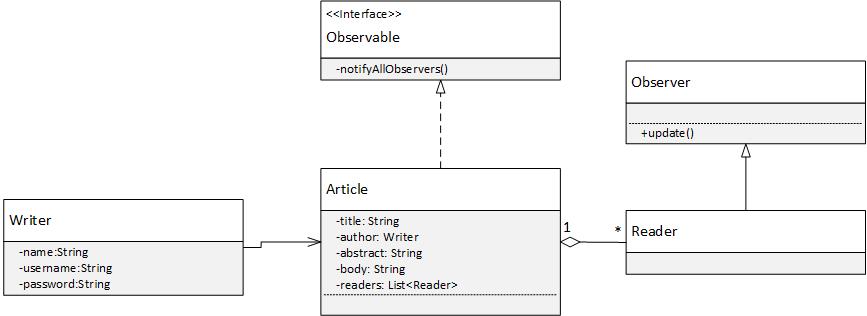


5. Class Design

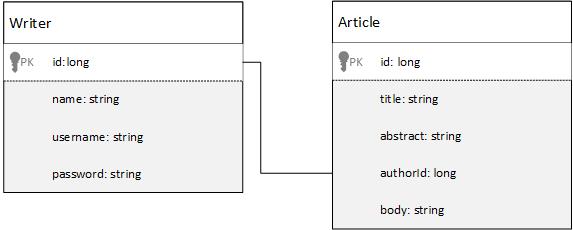
**5.1 Design Patterns Description**

The Observer design pattern falls under the category of behavioral patterns. This design pattern allows the creation of one-to-many relationships where if one object is modified, its dependent objects are automatically notified. This pattern will be used to allow the client’s views to be automatically updated when an article is modified.

**5.2 UML Class Diagram**



6. Data Model



7. System Testing

The system will be subject to a variety of use case tests to make sure all the functionality is successfully fulfilled.

8. Bibliography

https://www.tutorialspoint.com/design\_pattern/observer\_pattern.htm