

# **Assignment 3**

## **Analysis and Design Document**

**Student: Ciucescu Vlad Andrei**  
**Group: 30432**

# Table of Contents

1. Requirements Analysis	3
1.1 Assignment Specification	3
1.2 Functional Requirements	3
2. Use-Case Model	4
3. System Architectural Design	5
4. UML Sequence Diagrams	7
5. Class Design	8
6. Data Model	9
7. System Testing	9
8. Bibliography	9

# 1. Requirements Analysis

## 1.1 Assignment Specification

Design and implement a Java client-server application for a news agency.

## 1.2 Functional Requirements

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 to create, update or delete articles. 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.

When writing an article, a list must be shown that supports multi-select for choosing the related articles.

## 1.3 Non-Functional Requirements

- 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.

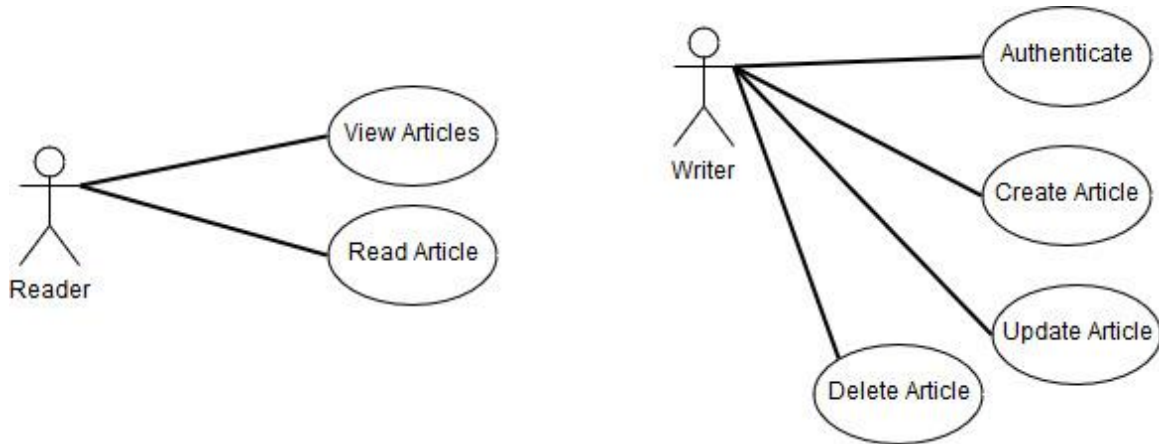
## 2. Use-Case Model

Use case: Read article

Level: user-goal level

Primary actor: Reader

Main success scenario: The reader enters the application and receives a list of available articles. He can then choose one of the articles, which will then be opened to be read.



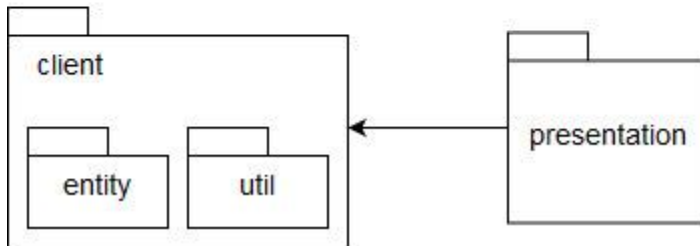
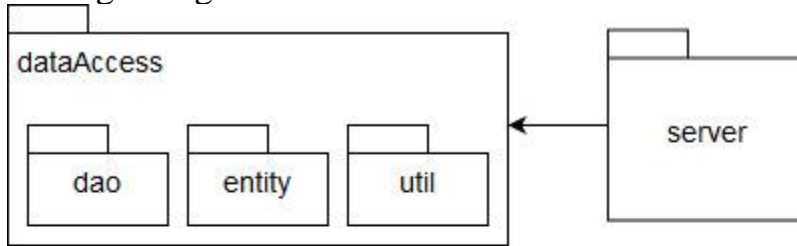
## 3. System Architectural Design

### 3.1 Architectural Patterns Description

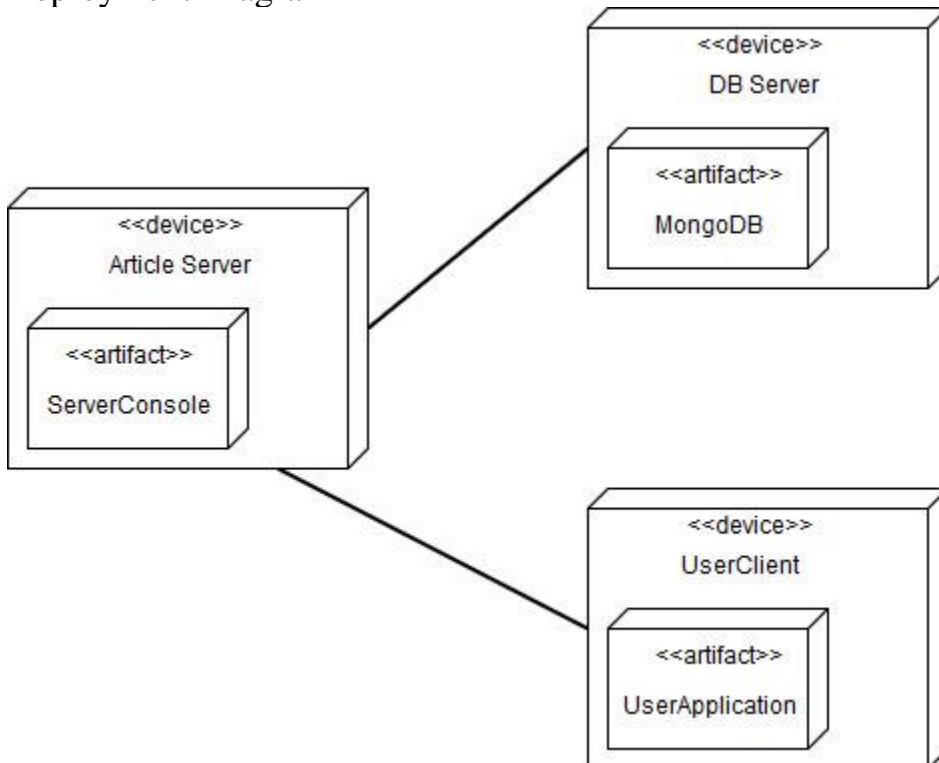
The application will be built using the client-server architecture. A client and a server establish a connection using the HTTP protocol. Once the connection is established, the client sends requests to the server in the form of JSON, which both entities understand. Afterwards, the server responds with appropriate data.

## 3.2 Diagrams

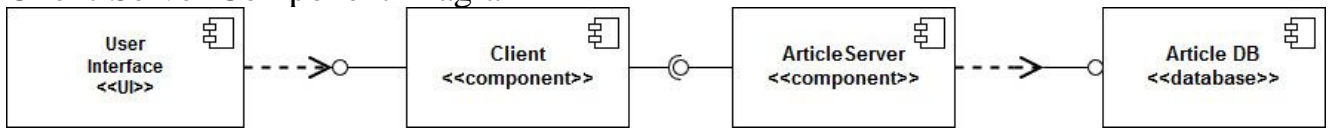
### Package Diagram



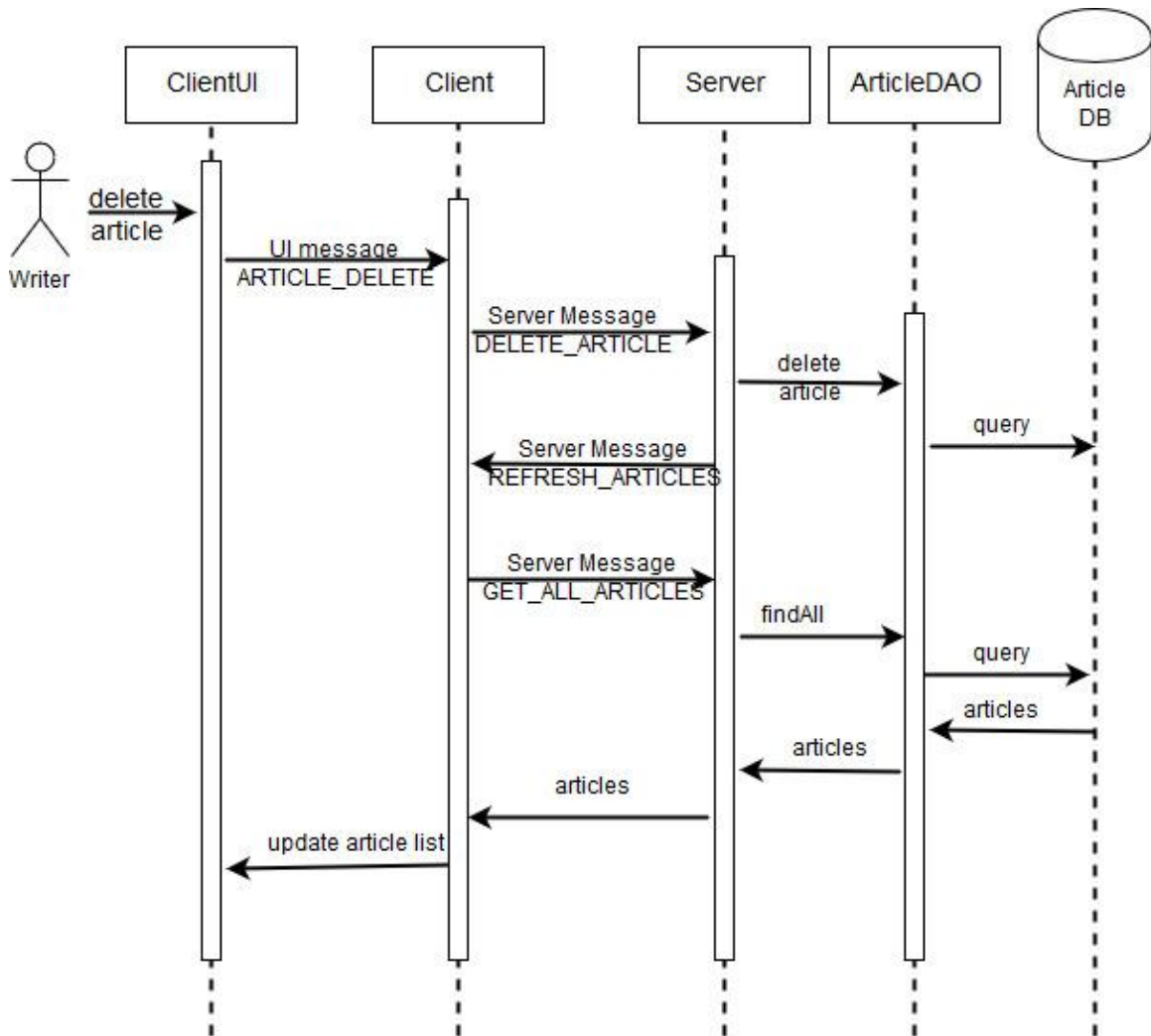
### Deployment Diagram



## Client-Server Component Diagram



## 4. UML Sequence Diagram



## 5. Class Design

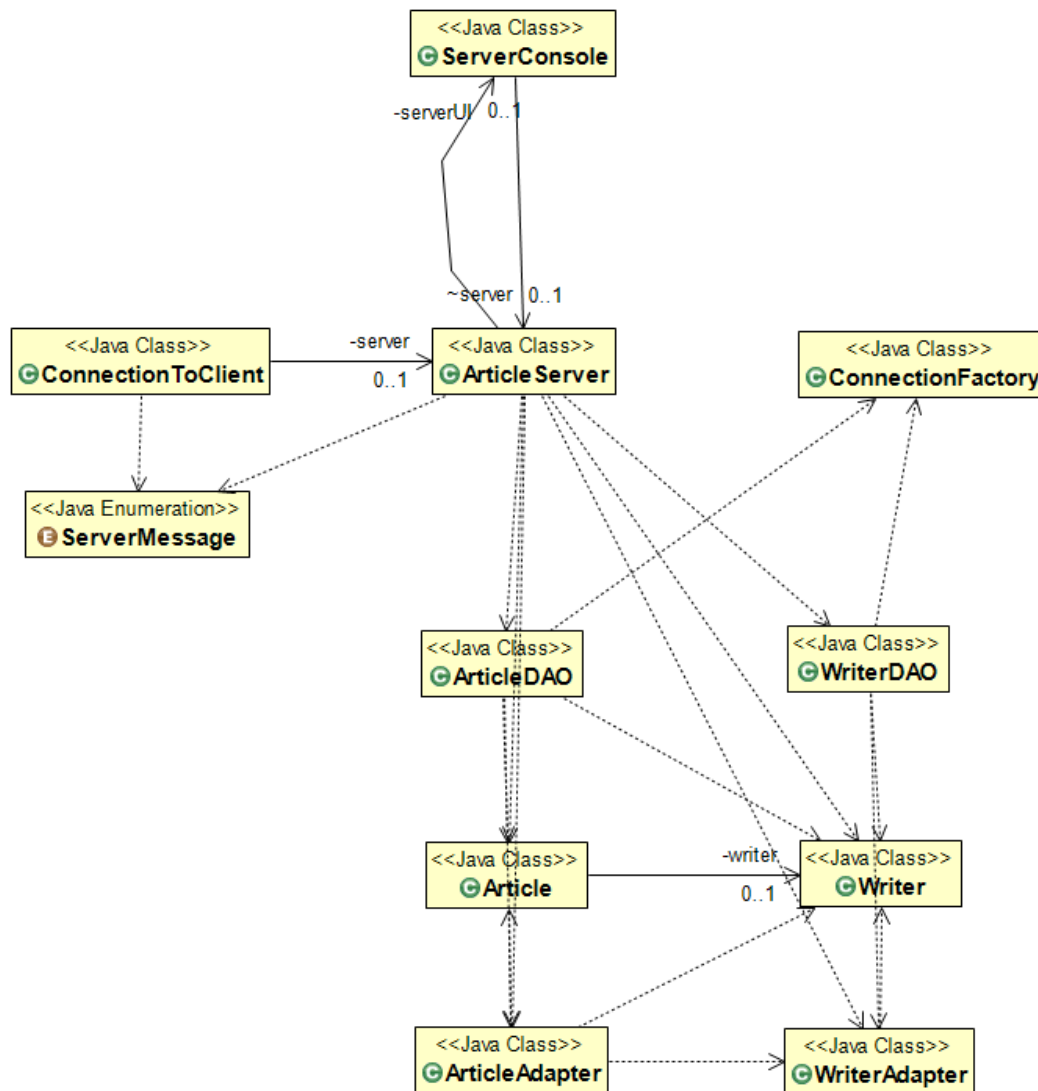
### 5.1 Design Patterns Description

Several design patterns have been used to implement the application:

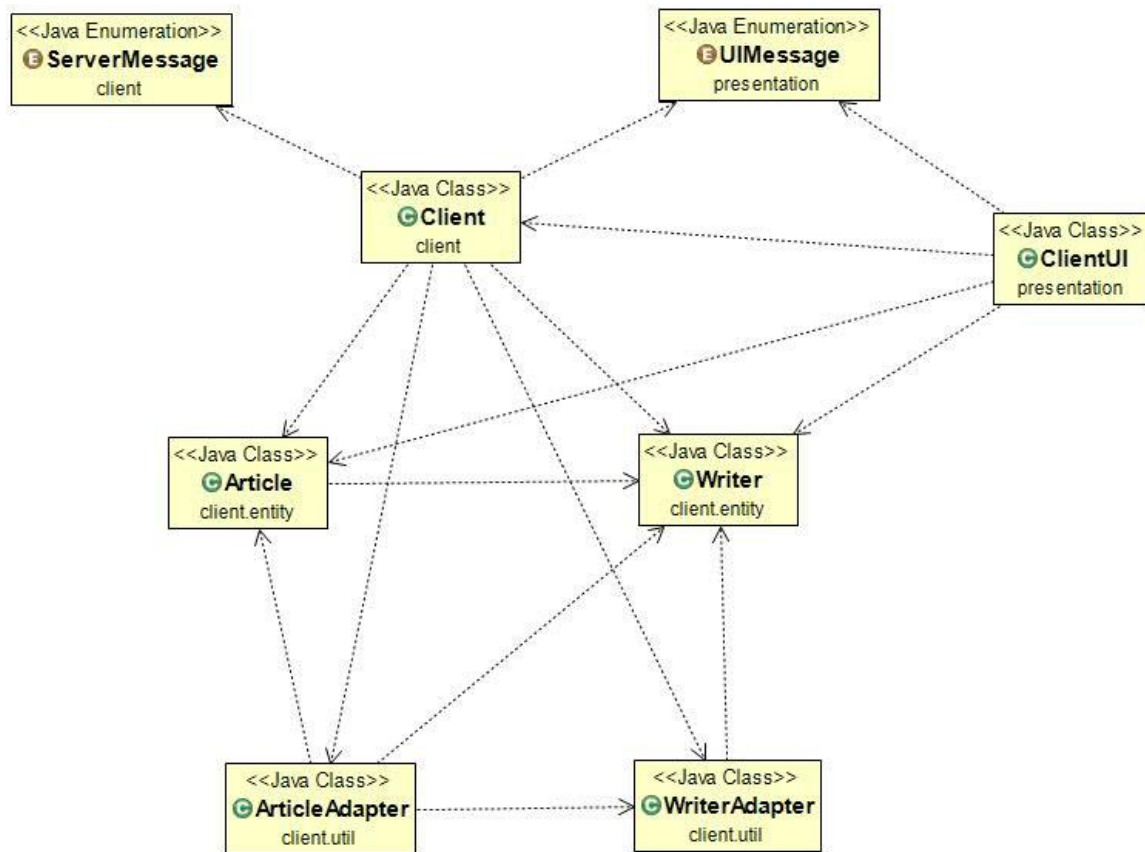
- The Adapter Pattern has been used to convert articles to and from JSON and MongoDB documents
- The Factory Pattern has been used to create connections to MongoDB
- The Observer Pattern has been used to ensure that client automatically see any update to articles. In this case, the clients have the role of the observers and the server is the observable. The server notifies the clients whenever the articles have changed.

### 5.2 UML Class Diagrams

#### Server

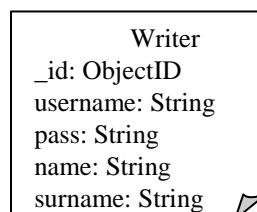
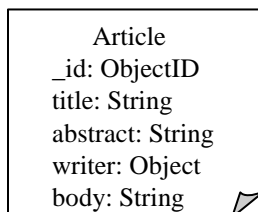


## Client



## 6. Data Model

The data is stored in a MongoDB database. As such, there exist two collections of documents: one for articles and one for writers.



## 7. System Testing

The testing strategy used is unit testing. A unit is the smallest testable part of software. As such, unit testing involves testing individual units or components (such as methods) of a software. Its purpose is to check whether each unit of the software performs as designed. For this application, unit testing will be performed with Junit.



## 8. Bibliography

<http://softwaretestingfundamentals.com/unit-testing/>

<http://www.agilemodeling.com/artifacts/>

<https://msdn.microsoft.com/en-us/library/ee658109.aspx>

[https://sourcemaking.com/design\\_patterns/factory\\_method](https://sourcemaking.com/design_patterns/factory_method)

<http://www.oodesign.com/adapter-pattern.html>

[https://en.wikipedia.org/wiki/Observer\\_pattern](https://en.wikipedia.org/wiki/Observer_pattern)

<http://www.baeldung.com/jackson>

<http://www.baeldung.com/jackson-object-mapper-tutorial>

[https://en.wikipedia.org/wiki/Client%E2%80%93server\\_model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model)

<http://www.site.uottawa.ca/school/research/lloseng/supportMaterial/source/>