Assignment 3

Student: Radu Petrisel

**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

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 the use the application. The **writers** need to authenticate in order tocreate, update or delete articles. So the writer accounts are preset by the application developer and cannot be altered.

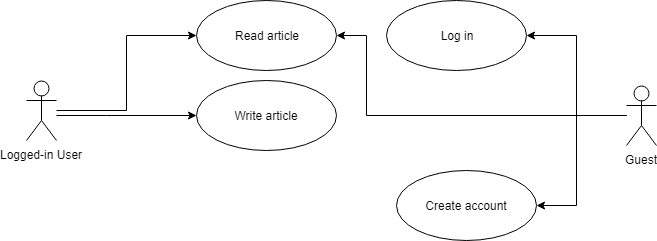
# Functional Requirements

* an article has the following components: title, abstract, body and author
* author must login in order to write an article
* any user can read articles (regardless of their login status)
* 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.

# Non-functional Requirements

* Reliability
* Scalability

2. Use-Case Model



Use case: read article

Level: user-goal level

Primary actor: user (logged in or guest)

*Main success scenario: article is read*

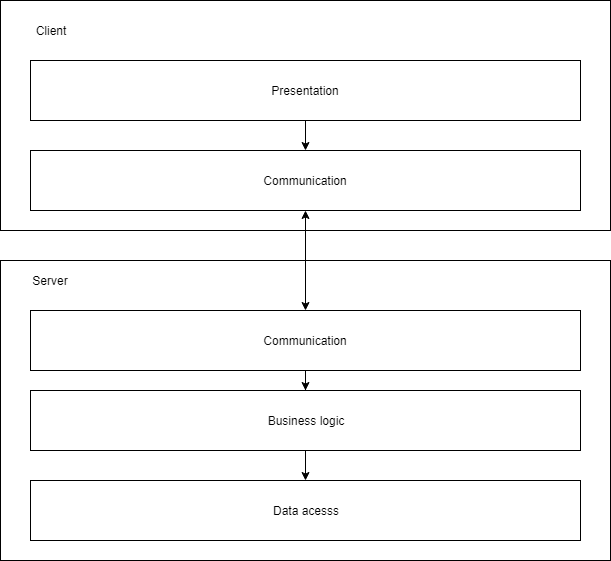
*Extensions: -*

3. System Architectural Design

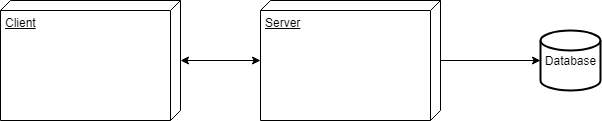
**3.1 Architectural Pattern Description**

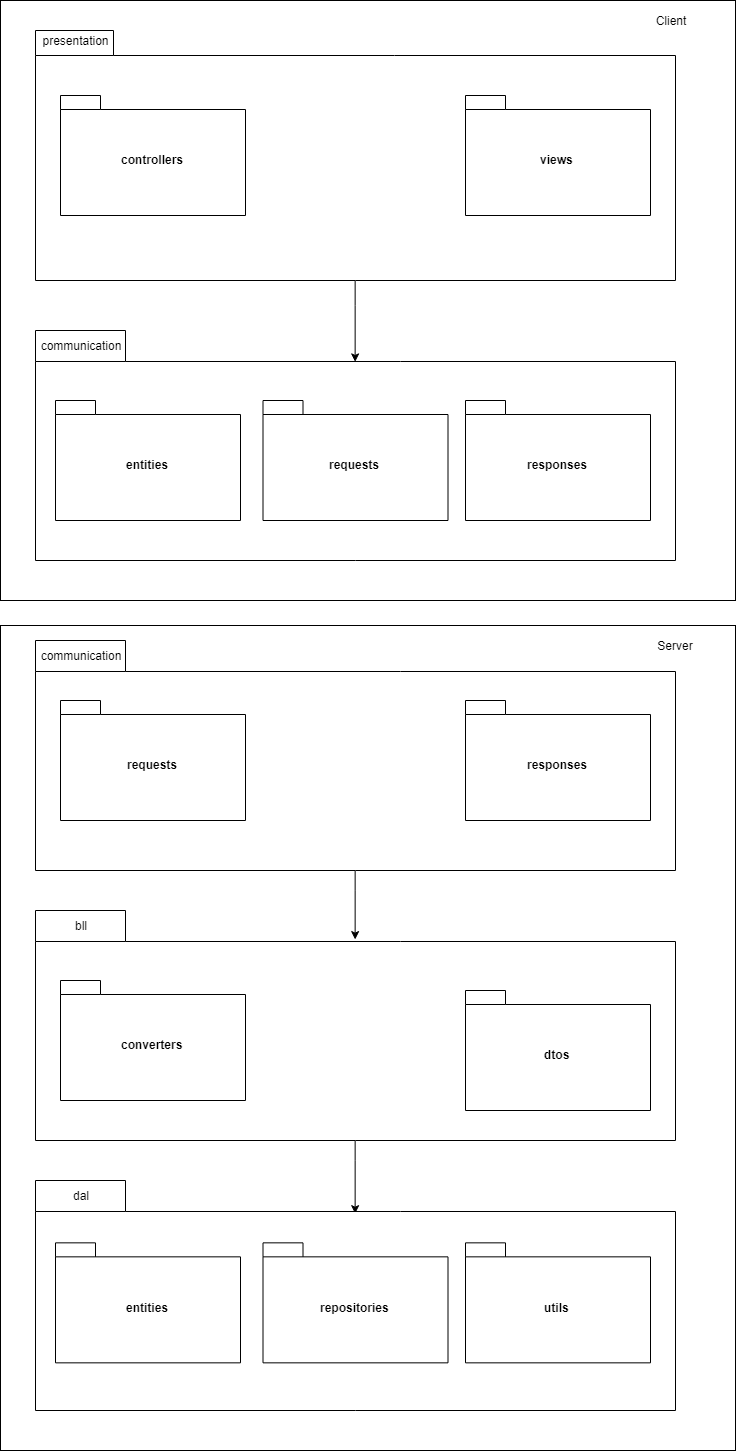
This assignment has a client-server architecture with MVC on the client side.

The client-server architecture is network architecture consisting in one server (or more identical ones) and many clients. The server gets requests from the client and processes them, sending a response to the client afterwards.



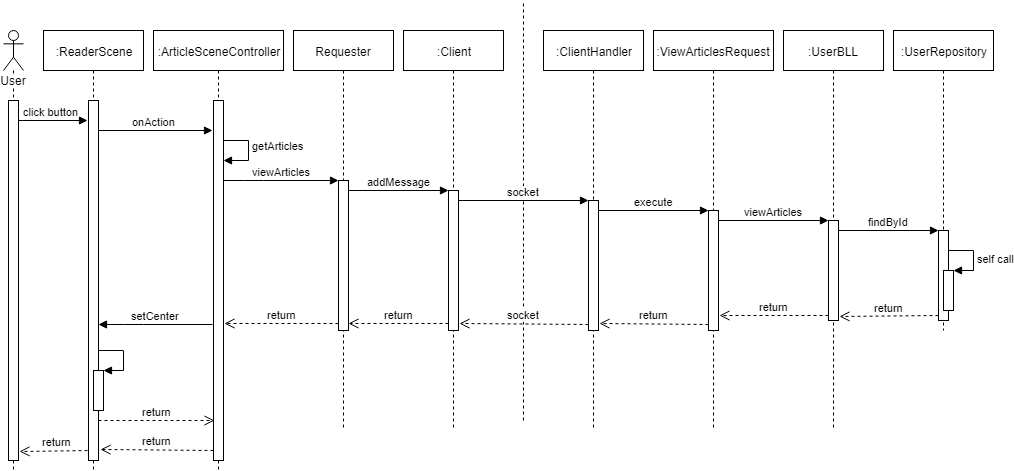
**3.2 Diagrams**

****Deployment



Package

4. UML Sequence Diagrams

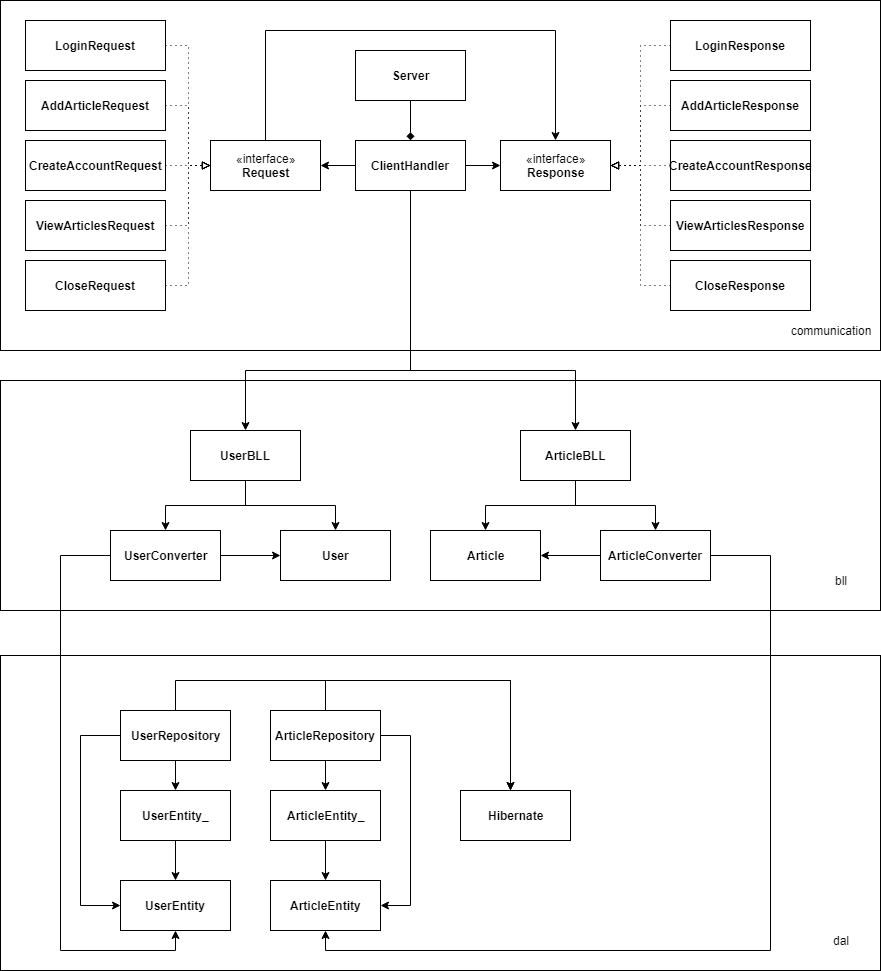


5. Class Design

**5.1 Design Patterns Description**

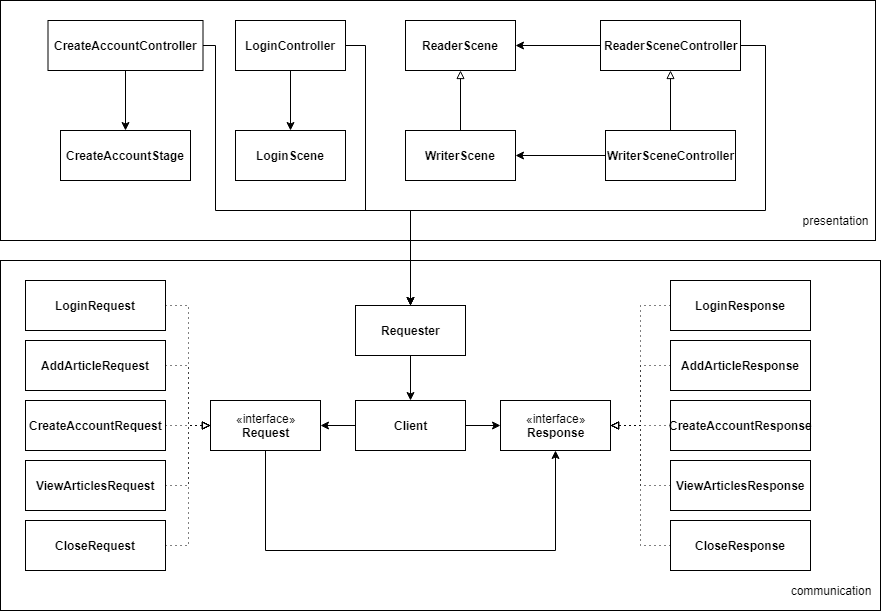
* singleton (with double checking and lock) – for the session factory
* command – for server’s request parsing
* observer – for removing a client from the server’s active connections when it closes

**5.2 UML Class Diagram**



Server

Client



6. Data Model

**

7. Bibliography

[Stackoverflow](www.stackoverflow.com)