Assignment 3

Analysis and Design Document

Student: Margin Razvan Cristian

**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 4

3. System Architectural Design 5

4. UML Sequence Diagrams 9

5. Class Design 10

6. Data Model 11

7. System Testing 11

8. Bibliography 11

1. Requirements Analysis

# Assignment Specification

The assignment states that we need to implement an application for managing News Agency. The application should have two types of users (the regular user or the reader and the administrator or the writer). Only the Writer has to login with a username and a password in order to access the functionalities of the application.

The reader should be able to view his articles. Furthermore, he is able to see in real time if any changes to the articles is being done by a writer.

The writer should be able to create/update articles after he logs into his account.

The application must support multiple users at the same time.

# Functional Requirements

* Reader can see a list of articles without logging into the system.
* Reader can see a particular article.
* Teacher can log in in order to manipulate the articles.
* Teacher can perform CRUD operations on the data
* Data is stored in a relational database
* Application is client-server and can support multiple clients at the same time.

# Non-functional Requirements

* Security
  + Writer data should be protected against attackers so that no personal information is leaked and nobody else besides him can manipulate the data
* Adaptability
  + This application could be used for managing other News agencies as well, being able to be used as a template for other agencies interested in developing a similar application

2. Use-Case Model



Use-Case description format:

**Use Case:** Writer creates Article

**Level:** user-goal level

**Primary actor:** Writer

**Main success scenario:**  Login => Insert Data (for new article) => Add Article

**Extensions:** Failure if invalid data inserted for Article or Writer’s login data are invalid

3. System Architectural Design

**3.1 Architectural Pattern Description**

* Layering

We use Layering architectural pattern in order to break up our software components into groups. Each of the layers are separated based on their primarily functional responsibilities. In our application we will have 3 layers. The Presentation layer will be responsible for displaying the data to the user, containing the user interface components. The Business layer will handle the logic behind the app. The Data layer will perform the communication between our application and the relational database used.

* Model-View-Presenter

We use the MVP architectural pattern in order to divide our application into 3 connected components. The Model will contain the data representation of our problem domain. The View is the presentation, i.e the User Interface of the application. The Presenter will be the core of our application, which will handle inputs and convert it into commands for the view.

* Client-Server

We will use a Client-Server architectural pattern in order to separate the backend of the application and the frontend. Furthermore, we need to provide this kind of pattern because we have a Client-Server application, which needs to support multiple concurrent clients. The communication is done through the help of sockets.

**3.2 Diagrams**



The layer diagram shows the structure of our application through dependencies. One can observe that we use a 3 layered architectural pattern, each layer with its own responsibility. The responsibilities of the layers have been explained in the paragraph above. Also, in order to create a communication between the client and the server, we add a communication layer which connects the application through certain handlers. The MVP pattern is present in the Client part of the app.



The deployment diagram is a structure diagram through which we see the architecture of our application distributed through artifacts. The artifacts represent physical elements from the real world that are a result of a development process. We have the Client App artifact which works as a GUI, the Server App artifact which controls the whole system, processing the data, and the Database Server, in which we store the data.

4. UML Sequence Diagrams

UML Sequence Diagram for Writer trying to login



5. Class Design

**5.1 Design Patterns Description**

Observer Pattern is going to be used in this assignment in order to update in real time the list of articles that the reader is able to see if any changes are made by one of the writers logged in.

**5.2 UML Class Diagram**

**Server**



**Client**



6. Data Model

The data model of this application consists in the main entities that we have also represented in the Class Diagram. We have the Writers and the Articles, because the user does not need to login, so we don’t need to store info about him. Between these models there are relationships that define the behavior of the application

* A writer can be the author of several articles.

Therefore, we can see the data model and the relationships between them in the table below.



Finally, the data is stored in a relational database: PostgreSQL.

7. System Testing

In our application we will use Junit Tests in order to test the functionalities of our system.

8. Bibliography

<https://www.techrepublic.com/article/commonly-used-architectural-patterns-in-java-applications/>

<https://www.thoughts-on-java.org/ultimate-guide-association-mappings-jpa-hibernate/>

http://download.oracle.com/javase/tutorial/networking/sockets/index.htm