

Student: Rogoz Bogdan
Group: 30433

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	7
6. Data Model	7
7. System Testing	8
8. Bibliography	8

1. Requirements Analysis

1.1 Assignment Specification

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

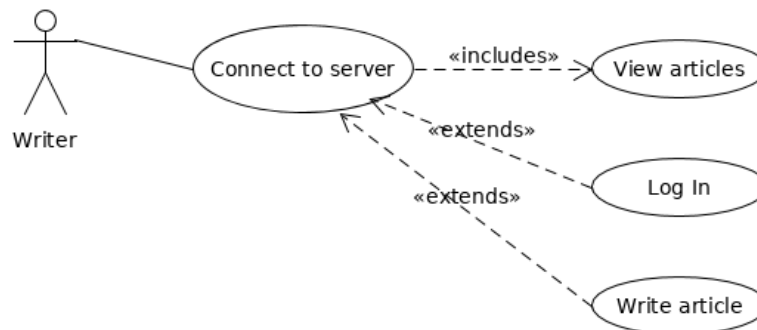
1.2 Functional Requirements

- 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
- The admin is the only one who can create writer accounts, but cannot create new admin accounts
- When reading an article the user should be able to see the title and the abstract of the related articles
- The application must support multiple concurrent users

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
- When writing an article, show a list that supports multi-select for choosing the related articles

2. Use-Case Model



Use case : Write article

Level : user goal

Primary actor : writer

Main success scenario :

1. The user fills the IP and Port fields
2. The user clicks the "Connect" button
3. The user fills the email and password fields
4. The user clicks the "Log In" button
5. The user clicks the write article button

Extensions :

1. The user fills the IP and Port fields

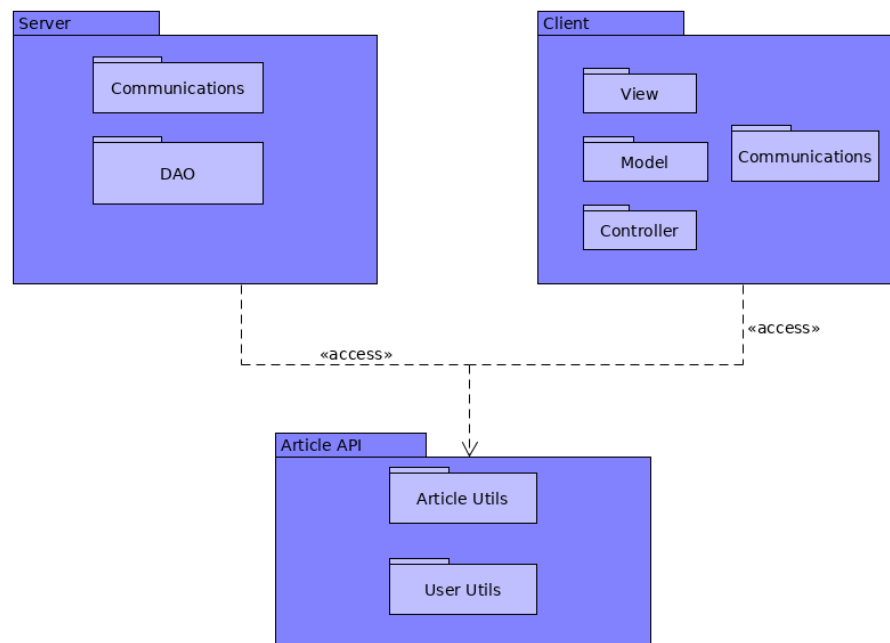
2. The user clicks the “Connect” button
 3. An error message is displayed
-
1. The user fills the IP and Port fields
 2. The user clicks the “Connect” button
 3. The user fills the email and password fields
 4. The user clicks the “Log In” button
 5. An error message is displayed

3. System Architectural Design

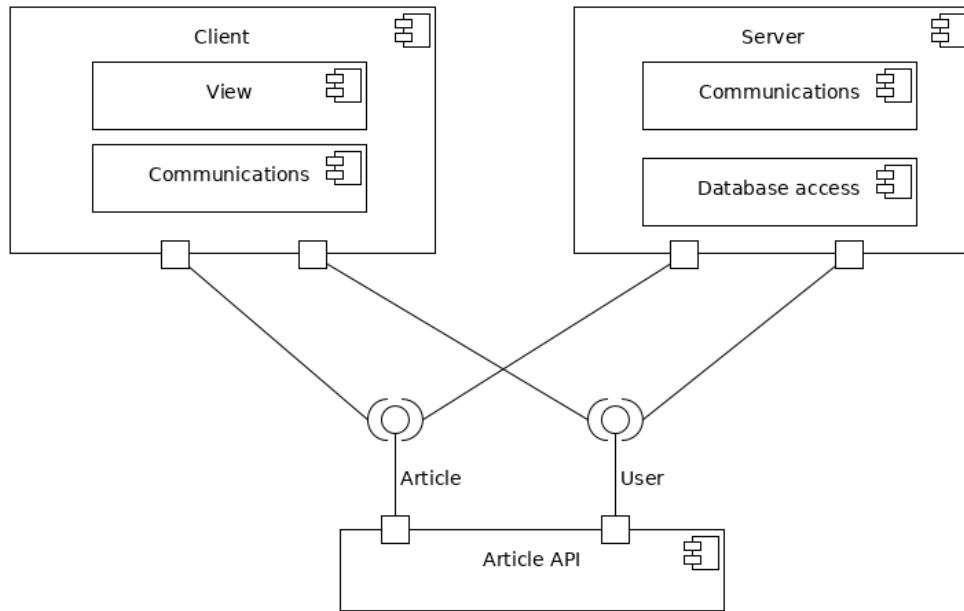
3.1 Architectural Pattern Description

- Model – View – Controller : it is made of 3 main components:
 - Model : observable objects that hold the data represented by the View component. If the model changes, the View is updated, and vice-versa.
 - View : the visual part of the application. It contains stages, scenes, text fields etc.
 - Controller : handles user input and sets the properties of the visual components.
- Client – Server : a centralized architecture, consisting of:
 - Server : the central core, listens for incoming client connections and performs communication through messages
 - Client : a connection to the server, can connect and disconnect at any time

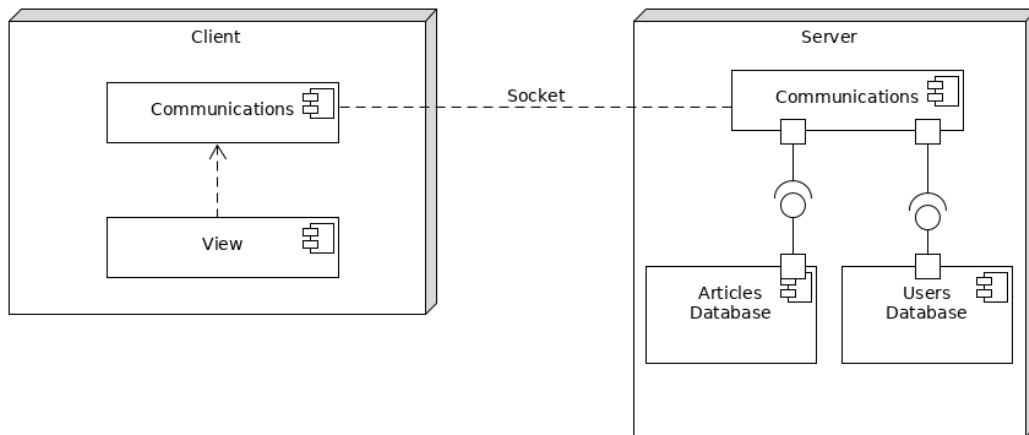
3.2 Diagrams



Package diagram

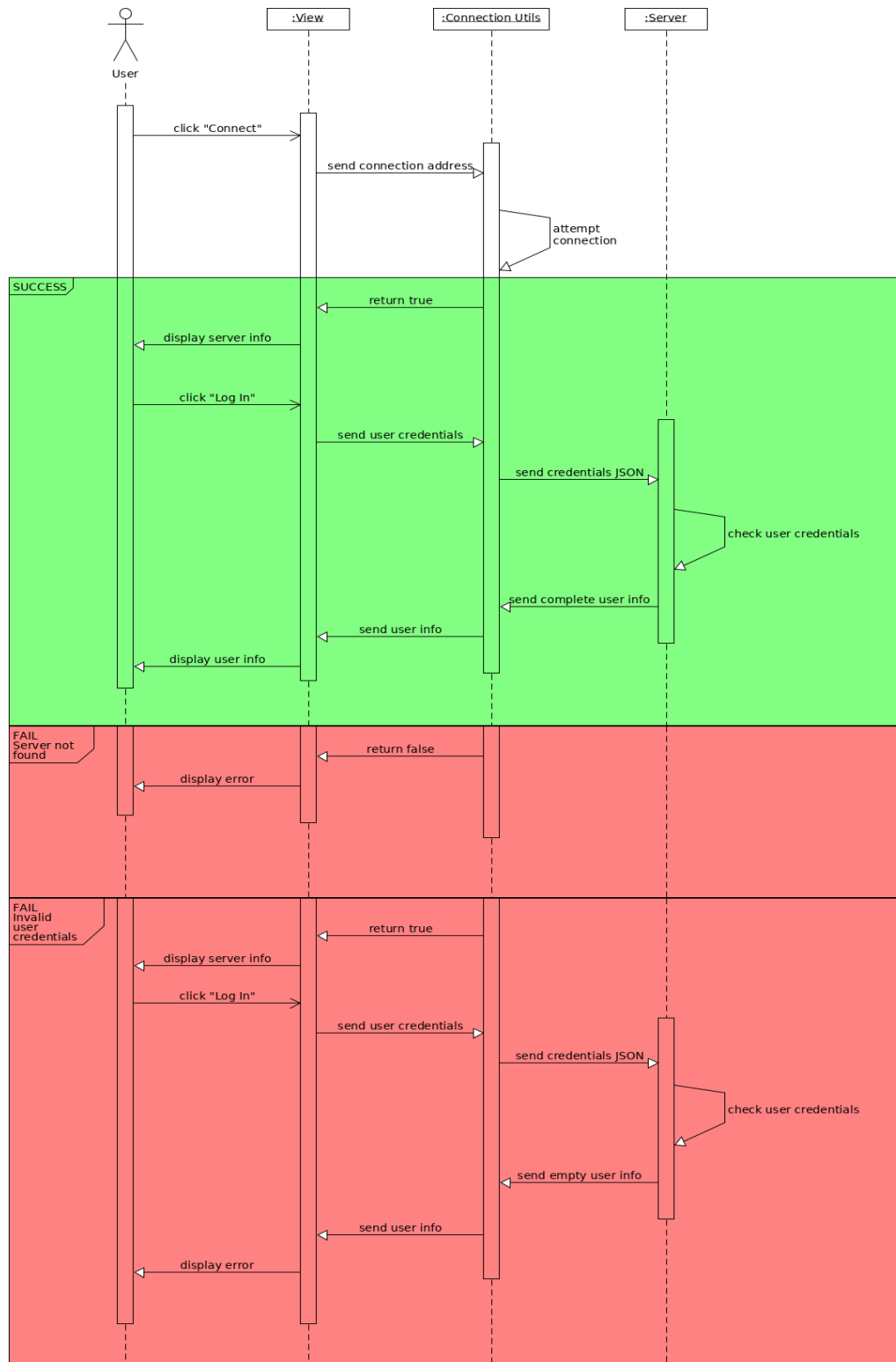


Component diagram



Deployment diagram

4. UML Sequence Diagrams

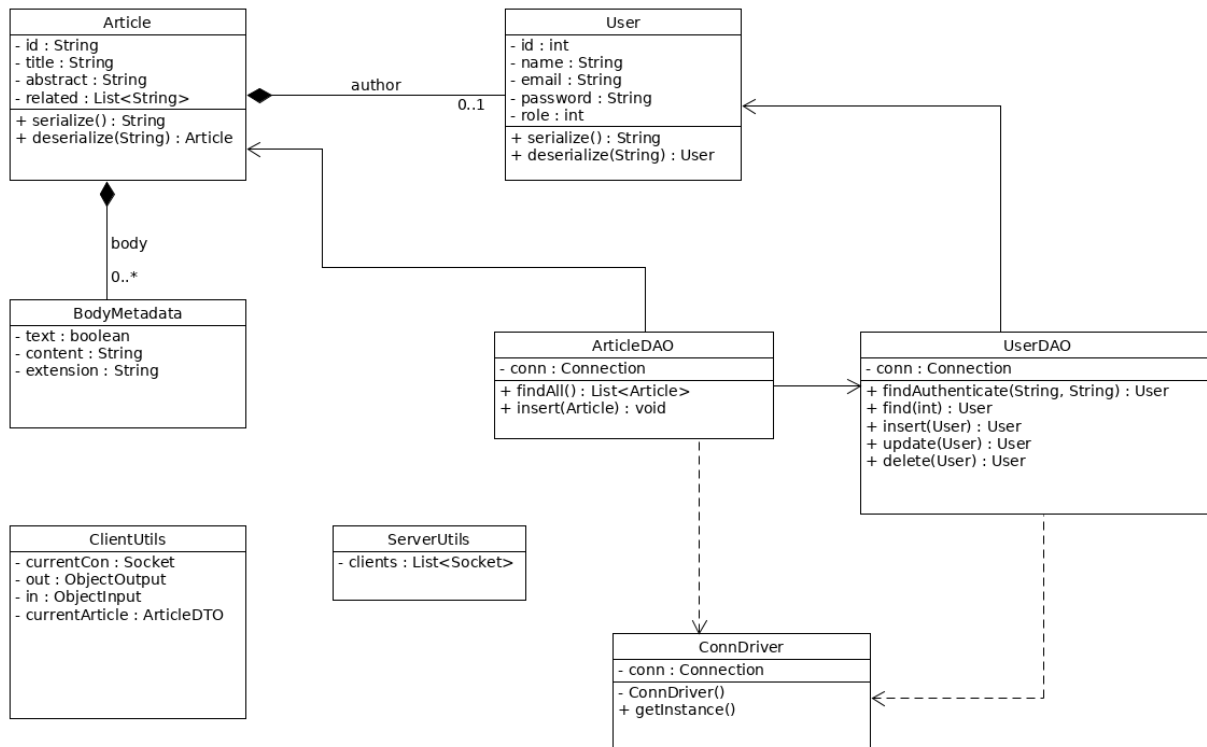


5. Class Design

5.1 Design Patterns Description

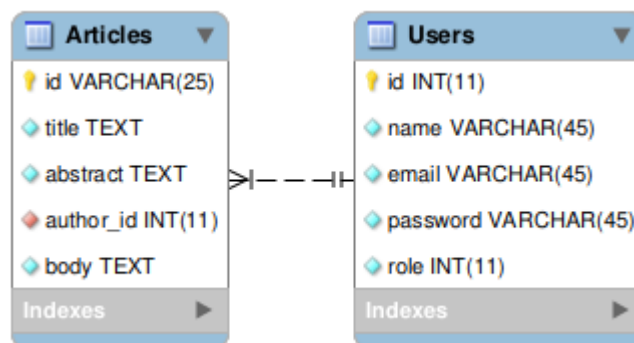
- Singleton – Used for ensuring correct functionality of the JDBC Driver
- Observer – Used for real time update of the articles list

5.2 UML Class Diagram



6. Data Model

- User / UserDTO – contains the information about user accounts
- Article / ArticleDTO – contains the content and information about articles



7. System Testing

Unit testing should be employed for this software:

- check email regex
- check connection status
- check log in credentials
- check conflicts when performing database operations

8. Bibliography

<https://docs.oracle.com/javase/tutorial/networking/sockets/index.html>

https://www.tutorialspoint.com/jackson/jackson_first_application.htm

https://www.tutorialspoint.com/design_pattern/observer_pattern.htm

<https://www.journaldev.com/1535/composite-design-pattern-in-java>