Individual Track Project

Web Application - Writeo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Date : 17/01/2021** | | | |
|  | **Version : 3.0** | | | |
|  | **Author : Kristian Hadzhikolev** | | | |
|  | Date | Revision History | Revision class | Comments | |
| Sprint 1 | 18.09 | 1.1 | - | - | |
| Sprint 2 | 09.10 | 1.2 | Major | Initial version(ER Diagram, Architecture, Argumentation about frontend and backend) | |
| Sprint 3 | 06.11 | 1.3 | Minor | Remaking design of diagrams and adding research methods | |
| Sprint 4 | 27.11 | 2.0 | Major | Adding CI setup, description for architecture, pagination and sources | |
| Sprint 5 | 16.12 | 2.1 | Minor | Changed some descriptions according to feedback | |
| Sprint 6 | 22.01.2021 | 3.0 | Major | Added all missing stuff and updated diagrams | |

# Entity Relation Diagram

# 

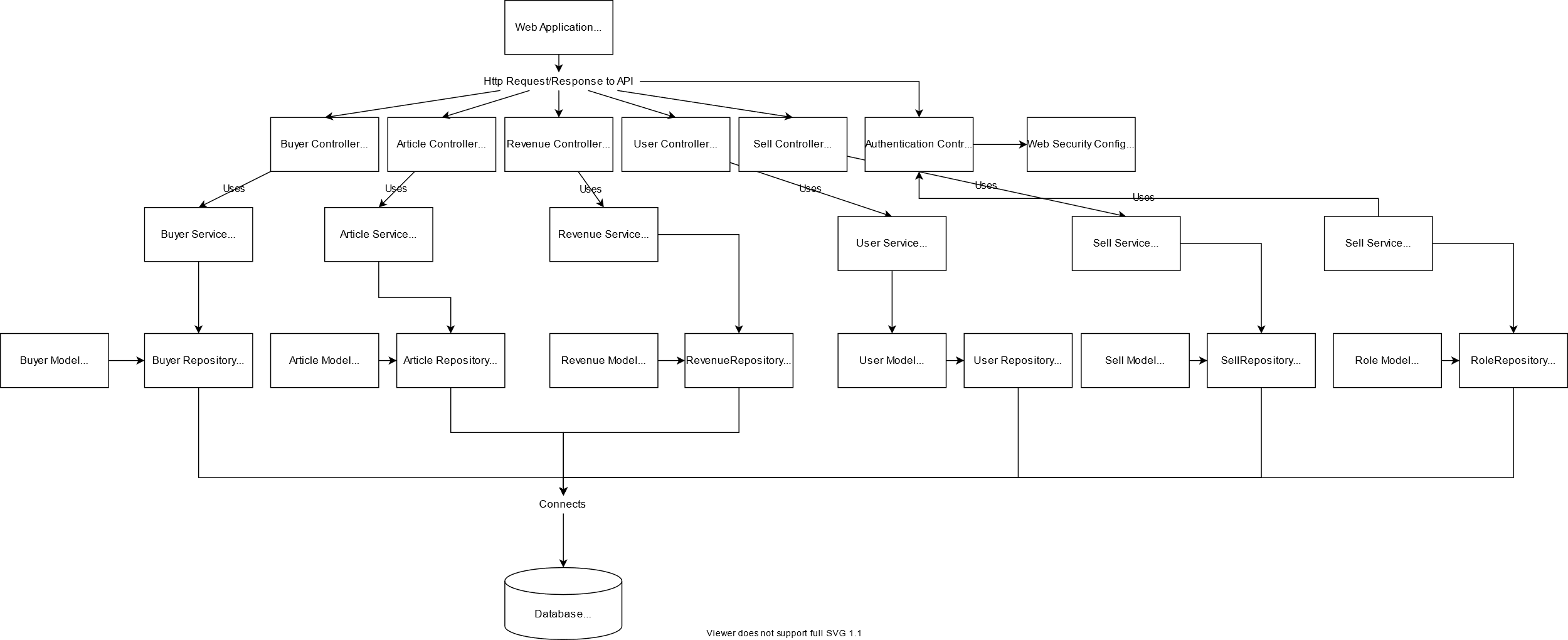
This diagram shows how the database and it’s tables and fields. User is for the profiles, article is for saving articles and their author ids, sell is for the sold articles and their buyer ids, buyer is for buyers information and finally revenue is for counting sold articles revenue by month.

# Continuous Integration Setup

Diagram

Description automatically generated

# High-Level Architecture



Bigger Picture is included in the Documents folder.

The React web application connects to the spring boot API(sending requests and receiving response). The API has 3 layers – Models(Objects with all their properties, for example Article has id, title and etc.), Controllers(which handle requests and return response) and Services connecting the Repository Interfaces to the controllers(The services check the data passed on from the controllers and use the repository interfaces to connect to the database).

Not every part is in the diagram, because it will not be readable.

# Design Decisions

The following ICT research methods were used to decide:

Benchmark creation(to list the possibilities), literature study(to read online about each one of the possibilities), trend analysis(to see, which one possibility has the best community is suitable for modern solution), context mapping(to decide, which possibility is most suitable for the project);

## Frontend

React framework will be used for the front-end of the web application. It was chosen, because it is one of the best frameworks on the market along with Vue and Angular. Other Pros are the flexibility, the developer community and the detailed documentation. It is the best in the market right now (Daityari, 2020). Testing is made with cypress.

## Backend

Spring Boot framework will be used for the backend of the web application. It was chosen, because as react it offers good documentation and it has many functions that make the process of developing faster and much more efficient.

Sonarqube is elite tool for code analysis and helps finding and fixing issues fast. (Author, 2016)

Testing is made is JUnit, Spring Runners and Mock Beans for integration Tests and JUnit for Unit Tests

# Burn-Down Chart

# Sources

# Bibliography

Author, V. (2016). Benefits of Using SonarQube For Code Reviews.

Daityari, S. (2020). Angular vs React vs Vue: Which Framework to Choose in 2020.

Schwarzmüller, M. (n.d.). Angular vs React vs Vue - My Thoughts. 2020.

Tutorialspoint. (n.d.). Spring Boot - Introduction.