Manev,Georgi G.S.

426015@student.fontys.nl

Abstract

Documentation about the design decisions made   
while building this full-stack web application.

Software Design Documentation

vidPit – a video sharing platform

**Project Name: vidPit**

**Date:** 03-January-2021

**Written By:** Georgi Manev

**Version:** 4

Contents

[Project Description 1](#_Toc60598013)

[Front-end section 2](#_Toc60598014)

[Description of criteria 2](#_Toc60598015)

[Comparison of the front-end frameworks 3](#_Toc60598016)

[My choice of frontend framework 4](#_Toc60598017)

[Wireframes 5](#_Toc60598018)

[Back-end section 8](#_Toc60598019)

[User Stories 8](#_Toc60598020)

[Planning 12](#_Toc60598021)

[API Endpoints 13](#_Toc60598022)

[ERD Design 15](#_Toc60598023)

[Justification of back-end 15](#_Toc60598024)

[Results of quality assurance metrics tool 16](#_Toc60598025)

[Security-related design decisions 18](#_Toc60598026)

## Project Description

VidPit is a web application for video sharing. Main purpose of the application is to allow users of the platform to upload their videos on it so other people can watch them, give feedback on the videos and follow other content creators. The application consists of a home page where the newest videos are shown. People can search for a videos by title, user or category. People can also create profile on the website so they can upload their videos and build their audience. Users can give feedback on other users’ videos by commenting on them or rating them. Users can follow other users so they receive notifications when they upload a video. The website is regulated by moderators which have the permission to delete videos and ban users from the website.

For building this full-stack web application I am using the following tools:

* Front-end / Client Side tools:

- **HTML5**

- **CSS3**

**- JavaScript**

- **Bootstrap** - This is HTML and CSS based framework and supports JavaScript.

- **Vue.js -** progressive JavaScript framework for building user interfaces.

* Back-end / Server Side tools:

- **Java**

**- MySQL**

**- Spring –** Java framework for building web applicationswith REST APIs

**- Hibernate –** ORM framework for Java

**- GitLab** for version control

**- GitLab CI/CD** for continuous integration and deployment

**- SonarQube** for software quality metrics

# Front-end section

## Description of criteria

The criteria that are important for me selecting a good front-end framework are the following:

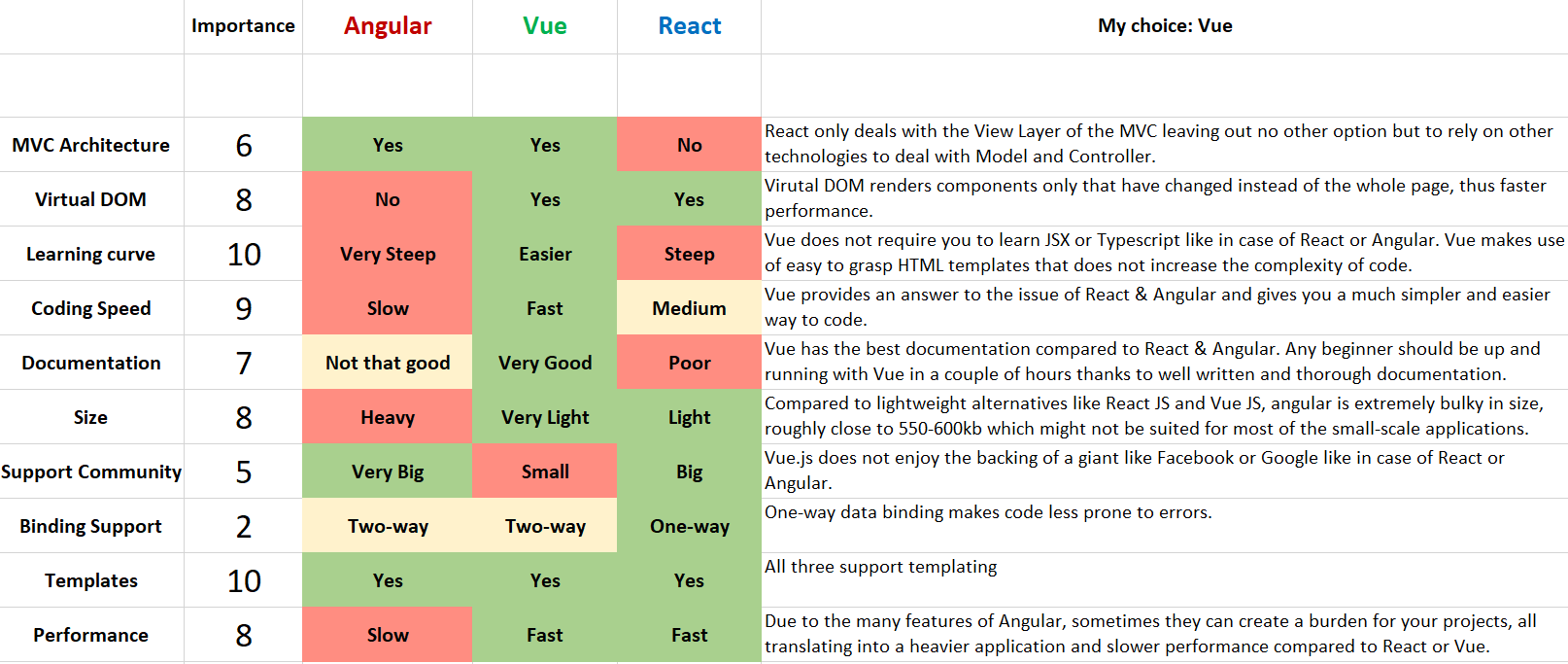
* MVC Architecture
* Virtual DOM
* Learning Curve
* Coding Speed
* Documentation
* Size
* Support Community
* Binding Support
* Templates
* Performance

Here is an explanation of the above criteria:

* MVC Architecture – Does the framework use MVC Architecture ? MVC is a software design pattern consisting of a Model, a View, and a Controller.
* Virtual DOM – Does the framework utilize a Virtual DOM ? The virtual DOM is a programming concept where an ideal, or “virtual”, representation of a UI is kept in memory and synced with the “real” DOM.
* Learning Curve – How difficult is for beginners to gain experience in this framework.
* Coding Speed – The average time that will take to implement the same feature using the three different frameworks.
* Documentation – How well written and detailed is the documentation of the framework.
* Size – Transfer size from the Chrome network tab. GZIPed response headers plus the response body, as delivered by the server. The smaller the file, the faster the download, and less to parse.
* Support Community – Support by community of active developers and large companies.
* Binding Support – Support for Data binding . When properties in the model get updated, so does the UI.
* Templates – Does the Front-end framework support templates ?
* Performance – Performance score from Lighthouse Audit that ships with Chrome. Lighthouse is an open-source, automated tool for improving the quality of web pages.

## Comparison of the front-end frameworks

For the comparison I selected Angular, Vue and React because they are currently the most popular JavaScript front-end frameworks according to data by GitHub and Google trends.

In the table (matrix) below, I compare the 3 frameworks using the criteria from section 1.

The scale of **Importance** that I use is in the range from 1 to 10. The higher the number the more important is the criteria on the left for me choosing the front-end framework.

I used system of colours to better visualize the comparison between the frameworks.

* **Green** – means that fully meets the criteria
* **Yellow** – means that partially meets the criteria
* **Red** – means that doesn’t meet the criteria

## My choice of frontend framework

From the three frameworks that I compared my final choice will be Vue. For me as a new developer it is very important to start with a framework that doesn’t have such steep learning curve. Also with the projects that we are currently building Vue is much better suited for making small to mid-size applications. Vue also has the speed and size advantage to the other two frameworks. Despite the support from Facebook for React and from Google for Angular, Vue’s community is constantly growing which makes it one of the fastest growing frameworks.

## Wireframes

In this section I will present my current ideas for the UI of the website illustrated through some wireframes.

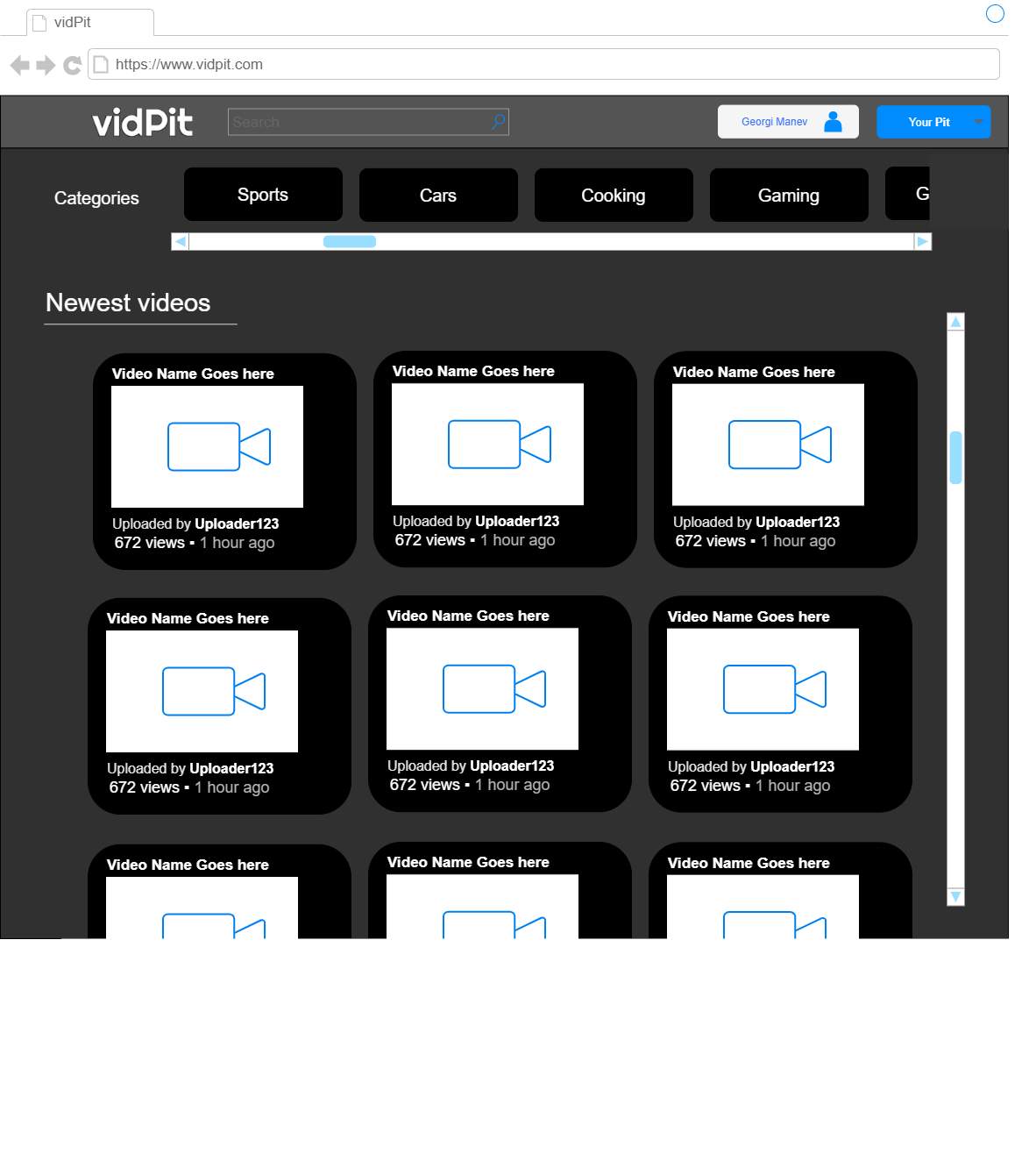
* From the main page the user can navigate through the newest videos, search for a user, video or go to his profile.

Figure 1: Main Page of the website

* The video page appears when user has clicked on a video, user can see the views, comment on the video or follow the content creator.

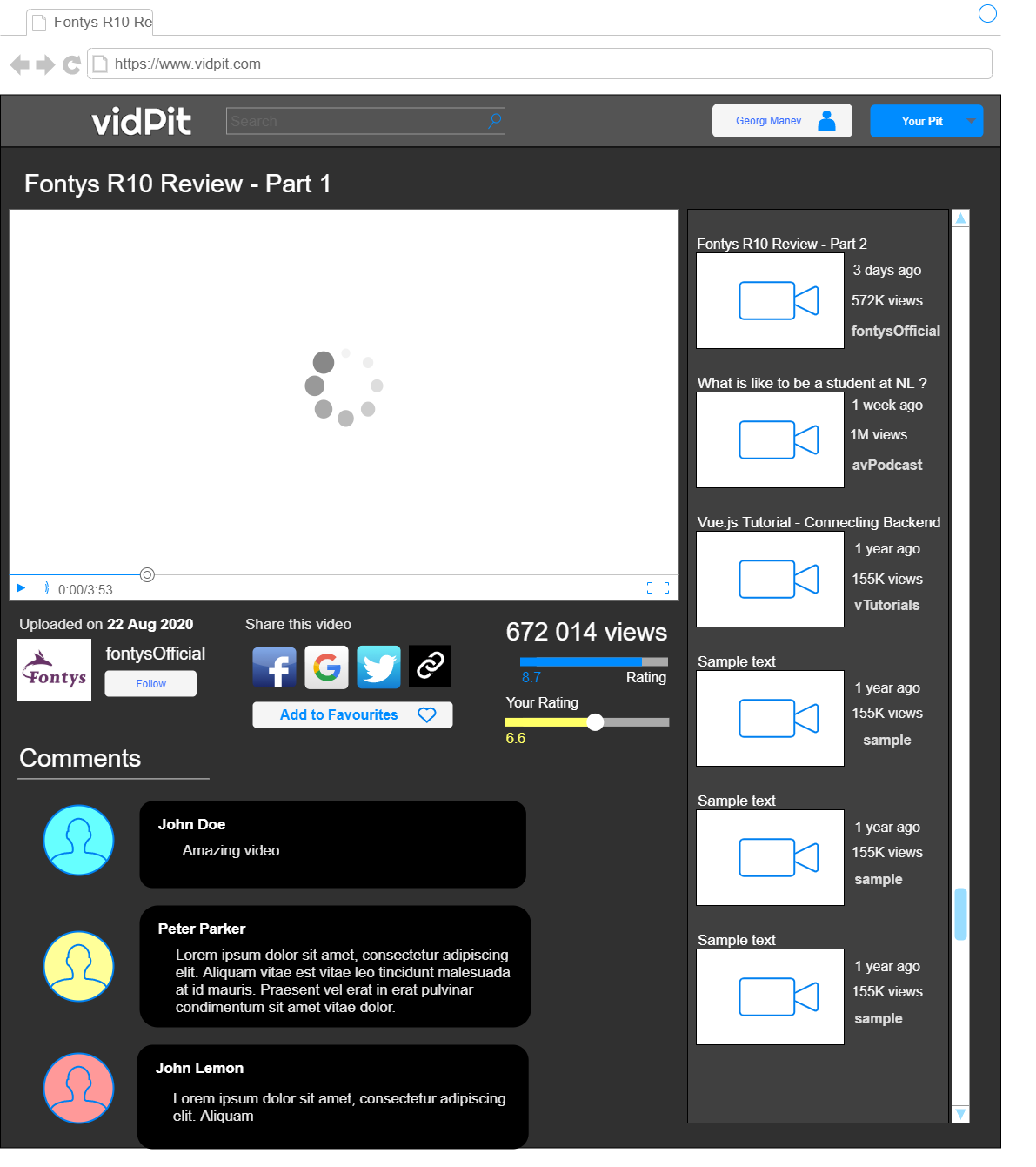


Figure 2: The video page

* From the profile page user can change his profile picture, view statistics, see his favourite and uploaded videos and edit his account personal information like password, email, username.

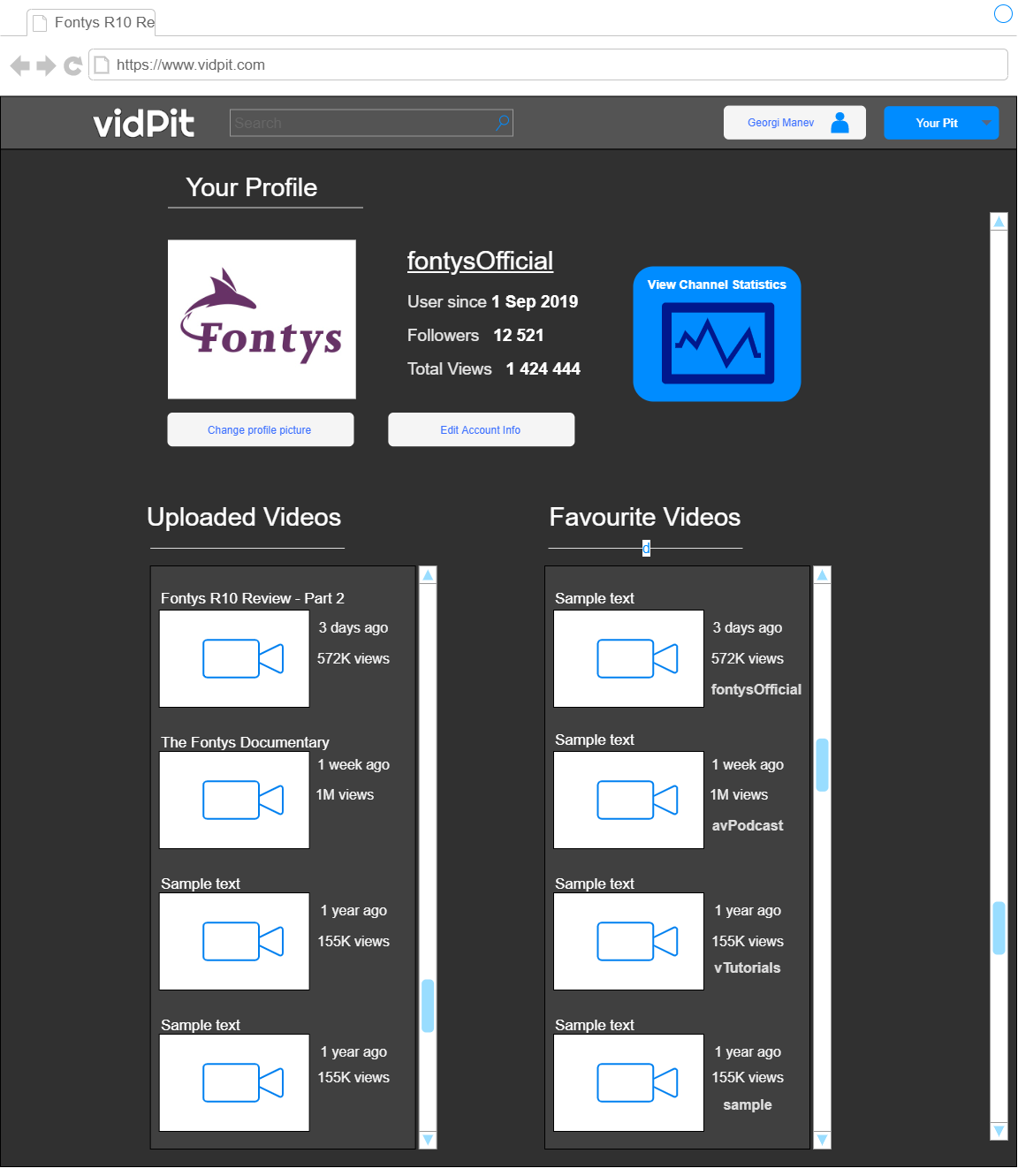


Figure 3: Profile page

# Back-end section

## User Stories

Legend:

M – Must Have

S – Should Have

C – Could Have

**User story:** As a new user I want to sign up to be able to use the site:

**Priority**: M

**Actors:** New user

**Acceptance criteria examples:**

User enters email and password and after email verification the customer is presented with welcome screen

**Requirements:** email: must be in valid format: [abc@xyz.com](mailto:abc@xyz.com); password: minimum 10 characters; email verification timeout 3 days.

**User story:** As a registered user I want to login into my account

**Priority**: M

**Actors:** Registered user

**Acceptance criteria:** User logs in successfully into his account and gets redirected to the home page

**Requirements:** email and password should match to an existing user

**User story:** As a registered user I want to change my profile picture

**Priority**: C

**Actors:** Registered user

**Acceptance criteria:** User uploads new profile picture and the current profile picture is changed

**Requirements:** picture must be in JPEG or PNG file format; picture format must be 1:1 and maximum size must be 1024x1024

**User story:** As a user I want to search for a specific video

**Priority**: M

**Actors:** User (registered or non-registered)

**Acceptance criteria :** The website list the video that the user searched for and also other videos related to it

**Failure:** Can’t find any videos matching the user’s criteria

**Requirements:** the video should or other similar videos should exist in the website’s database

**User story:** As a user I want to watch a video

**Priority**: M

**Actors:** Any user

**Acceptance criteria:** User clicks on a video or enters video’s link and a new page with the video loading opens

**Requirements:** the video should exist in the website’s database

**User story:** As a user I want to see the home page of the website

**Priority**: M

**Actors:** Any user

**Acceptance criteria:** The home page is shown successfully

**User story:** As a registered user I want to edit my profile info

**Priority**: M

**Actors:** Registered user

**Acceptance criteria:** User changes his password, username, bio, etc.

**Requirements:** username: must be in valid format: johnDoe97 (no special characters); password: minimum 10 characters;

**User story:** As a registered user I want upload videos to the platform so that other people can watch them

**Priority**: M

**Actors:** Registered user

**Acceptance criteria:** User uploads a video and gets published successfully

**Requirements:** video format should be .mp4, .mov, .flv, .wmv; videos should be maximum 1GB, title must not contain any special symbols or characters

**User story:** As a registered user I want to see statistics about my uploaded videos and subscribers

**Priority**: M

**Actors:** Registered user

**Acceptance criteria:** User views various statistics about his videos, viewers and subscribers

**Failure:** User doesn’t have any uploaded videos and statistics don’t appear

**Requirements:** user should have at least 1 uploaded video to see statistics about his channel

**User story:** As a registered user I want to comment on other people’s videos

**Priority**: C

**Actors:** Registered user

**Acceptance criteria:** User posts comment under another user’s video

**Requirements:** comment should contain maximum 2500 characters

**User story:** As a user I want to see recommendations of other videos related to the one I am watching currently

**Priority**: C

**Actors:** User

**Acceptance criteria:** User sees recommended videos below the one he is watching at the moment

**Requirements:** at least two videos that are similar should exist on the platform

**User story:** As a registered user I want to rate other content creators’ videos

**Priority**: S

**Actors:** Registered user

**Acceptance criteria:** User rates videos from 1 to 10 and successfully submits his vote

**Requirements:** rating bar should be clicked; user has to be registered

**User story:** As a registered user I want to follow other content creators

**Priority**: S

**Actors:** Registered user

**Acceptance criteria:** User clicks Follow button and successfully starts following the desired content creator

**Requirements:** Follow button should be clicked; user has to be registered

**User story:** As an administrator I want to ban users for not complying with the rules of the platform

**Priority**: M

**Actors:** Admin

**Acceptance criteria:** The administrator bans successfully the violator from the platform

**Requirements:** user has to have administrator permissions to ban other people

**User story:** As an administrator I should be able to delete videos for content that is against the website’s rules

**Priority**: M

**Actors:** Admin

**Acceptance criteria:** The admin deletes successfully the violating video

**Requirements:** user has to have administrator permissions to delete other user’s videos

**User story:** As an administrator I don’t want other users to access my admin page

**Priority**: M

**Actors:** Admin

**Acceptance criteria:** User gets an unauthorized message when accessing admin page.

**Requirements:** user shouldn’t have administrator permissions.

## Planning

* Sprint 3
  + **User story:** As a new user I want to sign up to be able to use the site
  + **User story:** As an administrator I want to ban users for not complying with the rules of the platform
* Sprint 4
  + **User story:** As a new user I want to sign up to be able to use the site
  + **User story:** As a registered user I want to login into my account
  + **User story:** As an administrator I don’t want other users to access my admin page
  + **User story:** As an administrator I should be able to delete videos for content that is against the website’s rules (Authorization)
* Sprint 5
  + **User story:** As a user I want to search for a specific video
  + **User story:** As a user I want to watch a video
  + **User story:** As a registered user I want to edit my profile info
  + **User story:** As a registered user I want upload videos to the platform so that other people can watch them
  + **User story:** As an administrator I should be able to delete videos for content that is against the website’s rules (Implementation)
* Sprint 6
  + **User story:** As a registered user I want to change my profile picture
  + **User story:** As a registered user I want to see statistics about my uploaded videos and subscribers
  + **User story:** As a registered user I want to comment on other people’s videos
  + **User story:** As a user I want to see recommendations of other videos related to the one I am watching currently
  + **User story:** As a registered user I want to rate other content creators’ videos
  + **User story:** As a registered user I want to follow other content creators

## API Endpoints

Video Object -

* Title : String
* ID : Int
* Description: String
* Views : Int
* Rating: Int
* Category: Enum<Category>
* Comments: List<Comment>
* Date of upload: DateTime
* Content: .mp4/.wmv/etc.

**GET** /videos *– returns all videos*

**GET** /videos/video {id} – *returns a video with a specific ID*

**GET** /videos?<Category>&.... - *Search for videos*

possible parameters:

Category=?

Title=?

**POST** /videos/video – *uploads a video with predefined parameters – Expects: Title, Description, Category, Content*

**PUT** /videos/video – *edits a video information*

**DELETE** /videos/video/{id} – *deletes a video with a specific ID*

Users:

User Object -

* Username : String
* Email : String
* Password: String
* ID – Int
* Videos: List<Video>
* Followers : Int
* Following: List<User>

**GET** /users – returns all users – (admin command)

**GET** /users/profile {id} – returns a user profile with a specific ID

**GET** /users/profile {id}/myvideos – returns the user’s uploaded videos

**GET** /users/profile {id}/analytics – returns various statistics about users’s channel

**GET** /admin – returns the admin page in which the admin can delete(ban) and edit users

**POST** /users/profile – creates a user with predefined parameters – Expects: Username, Password, Email

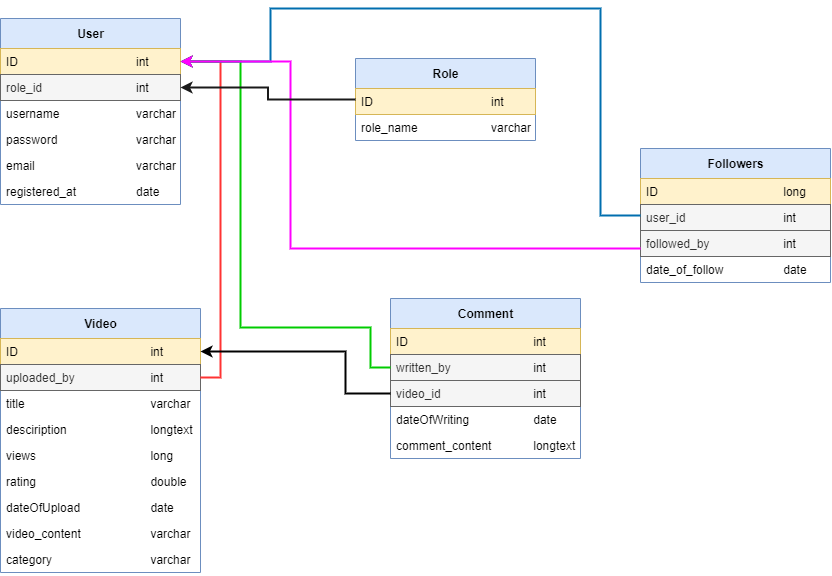
**PUT** /users/profile – edits a user profile information

**DELETE** /videos/video/{id} – deletes a user with a specific ID

Show front page:

**GET** /home – returns the front (home) page of the website (“home” will be replaced with the name of the website)

## ERD Design

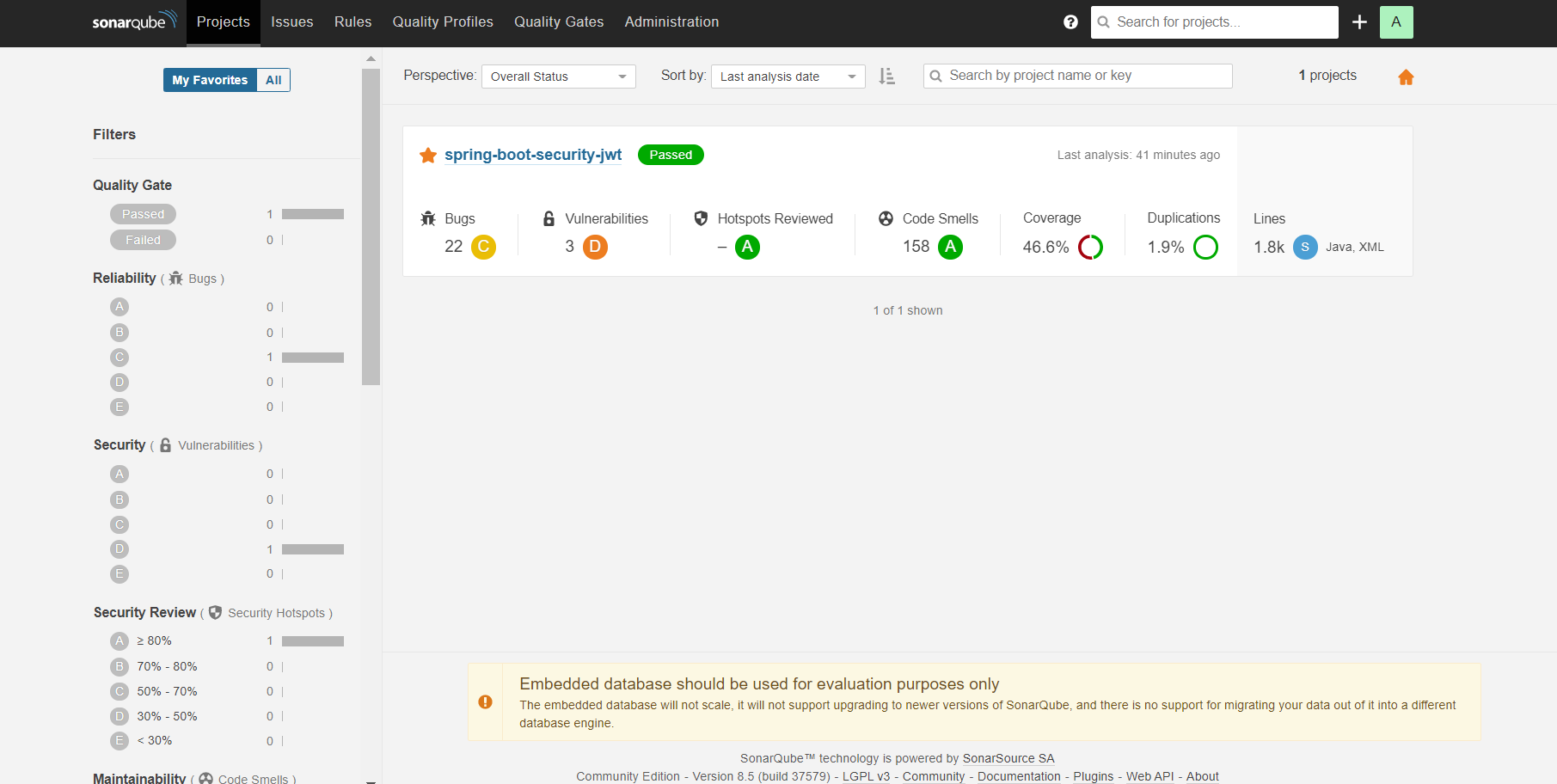


This is my design for the tables in the database and the relations between them. The primary key is highlighted in yellow, the foreign keys are highlighted in grey.

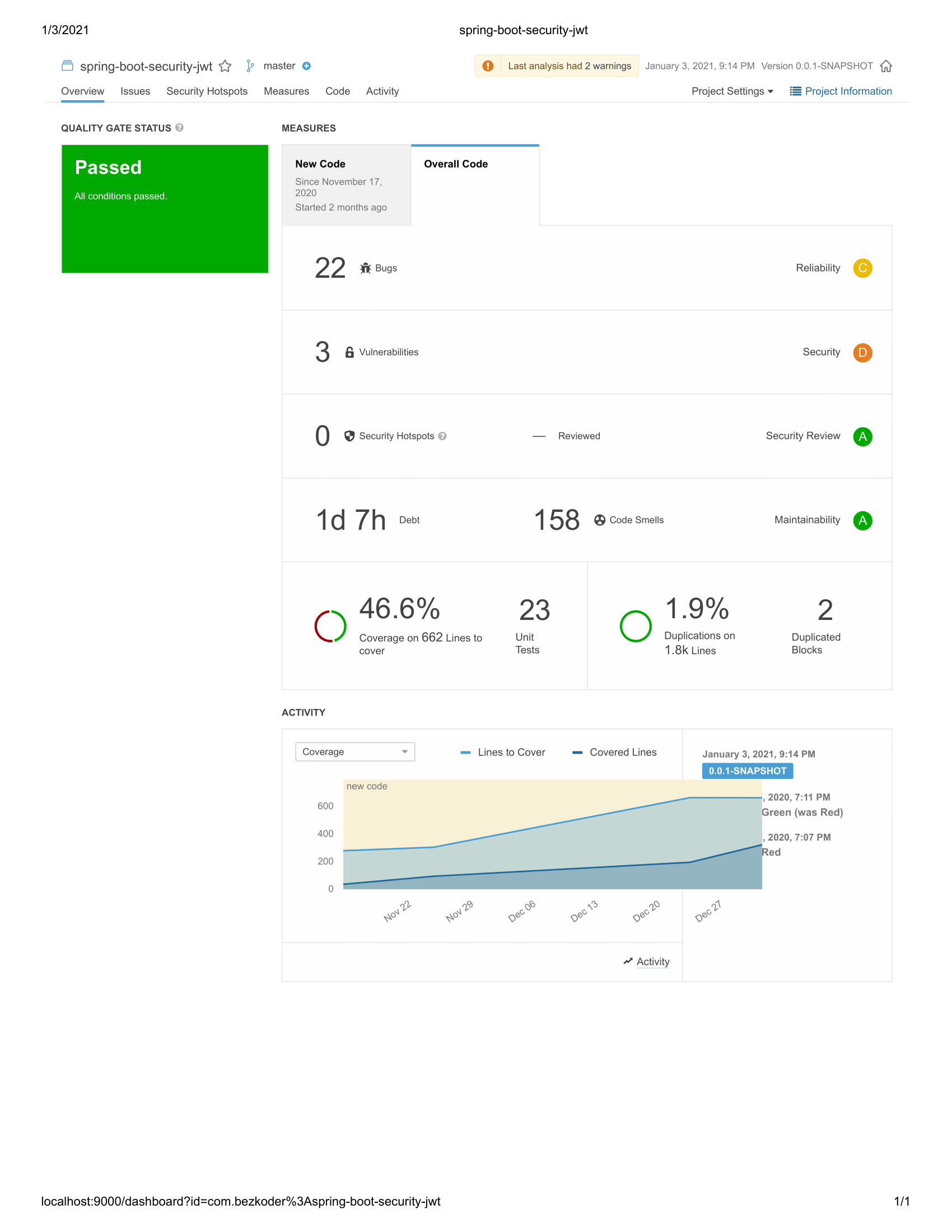
## Justification of back-end

I am using Spring because is the most popular Java Framework and has the biggest community support. The Spring Framework is a full framework that allows a developer to create Java enterprise applications. Spring simplifies integration with other Java frameworks like JPA/Hibernate ORM /etc. web frameworks. Along with the Spring framework, there are many other Spring sister projects that help to build applications addressing modern business needs. Spring includes Spring Security - a Robust security framework to secure applications. It focuses on providing both authentication and authorization to Java applications. By using Hibernate We can easily migrate from one database software to another Database software. Because of writing HQL queries these are database software independent but JDBC queries are database specific and they will be changed from one database to another database.

## Results of quality assurance metrics tool



The screenshot above is of the project page of SonarQube with all the conditions of the quality gate passing. The screenshot below shows an overview of the newly added code to the project since the last commit.



The screenshot above is of the project page of SonarQube for the submission for Sprint 5. Now showing increased coverage from 26.5% to 46.6%.

## Security-related design decisions

For authentication and authorization vidPit uses Spring Security. Spring Security is a framework that provides authentication, authorization, and protection against common attacks. With first class support for both imperative and reactive applications, it is standard for securing Spring-based applications. Before users can make a request with the API they will need to authenticate the request by providing a JWT token in the request’s header. VidPit uses Bearer JWT token authentication instead of Basic. The name “Bearer authentication” can be understood as “give access to the bearer of this token.” The bearer token is a cryptic string, usually generated by the server in response to a login request, unlike the Basic authentication which is the user ID and the password being passed over the network as a base64 encoded text. Base64 is a reversible encoding, therefore the basic authentication scheme is not secure.

## System design following C4-model

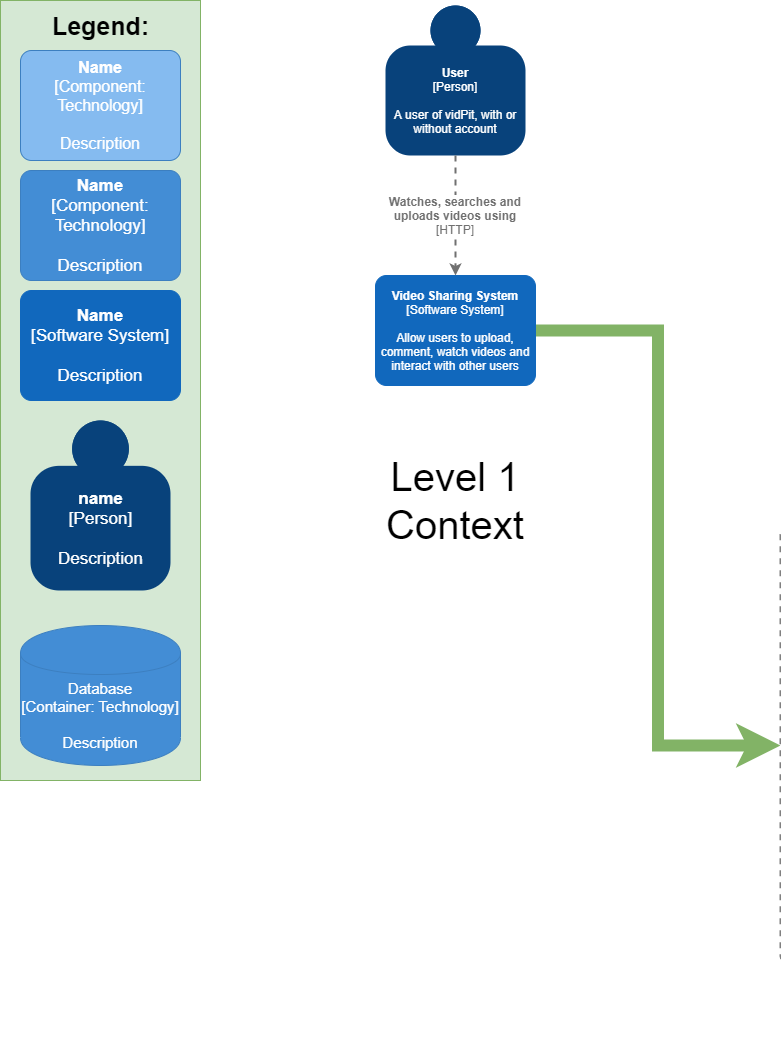


Figure : Level 1

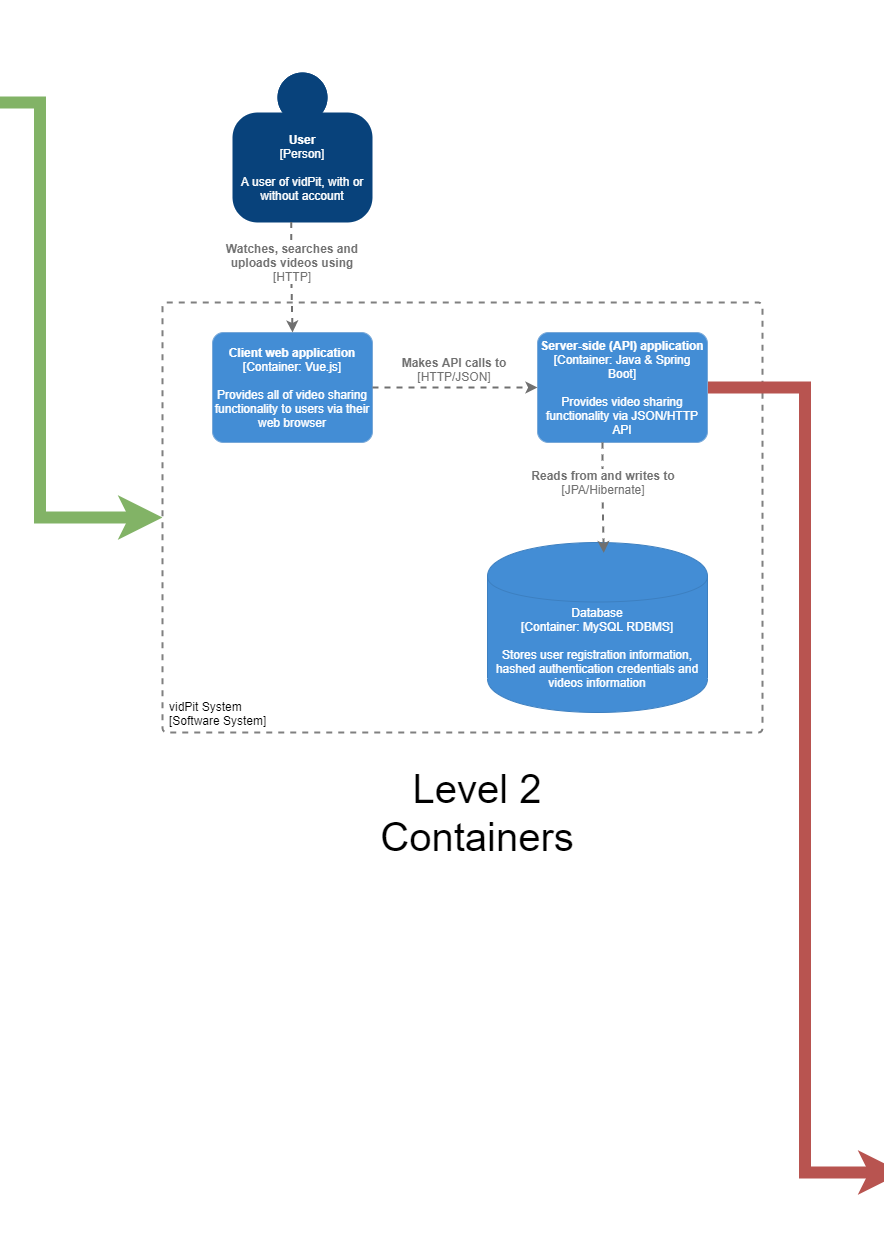


Figure : Level 2

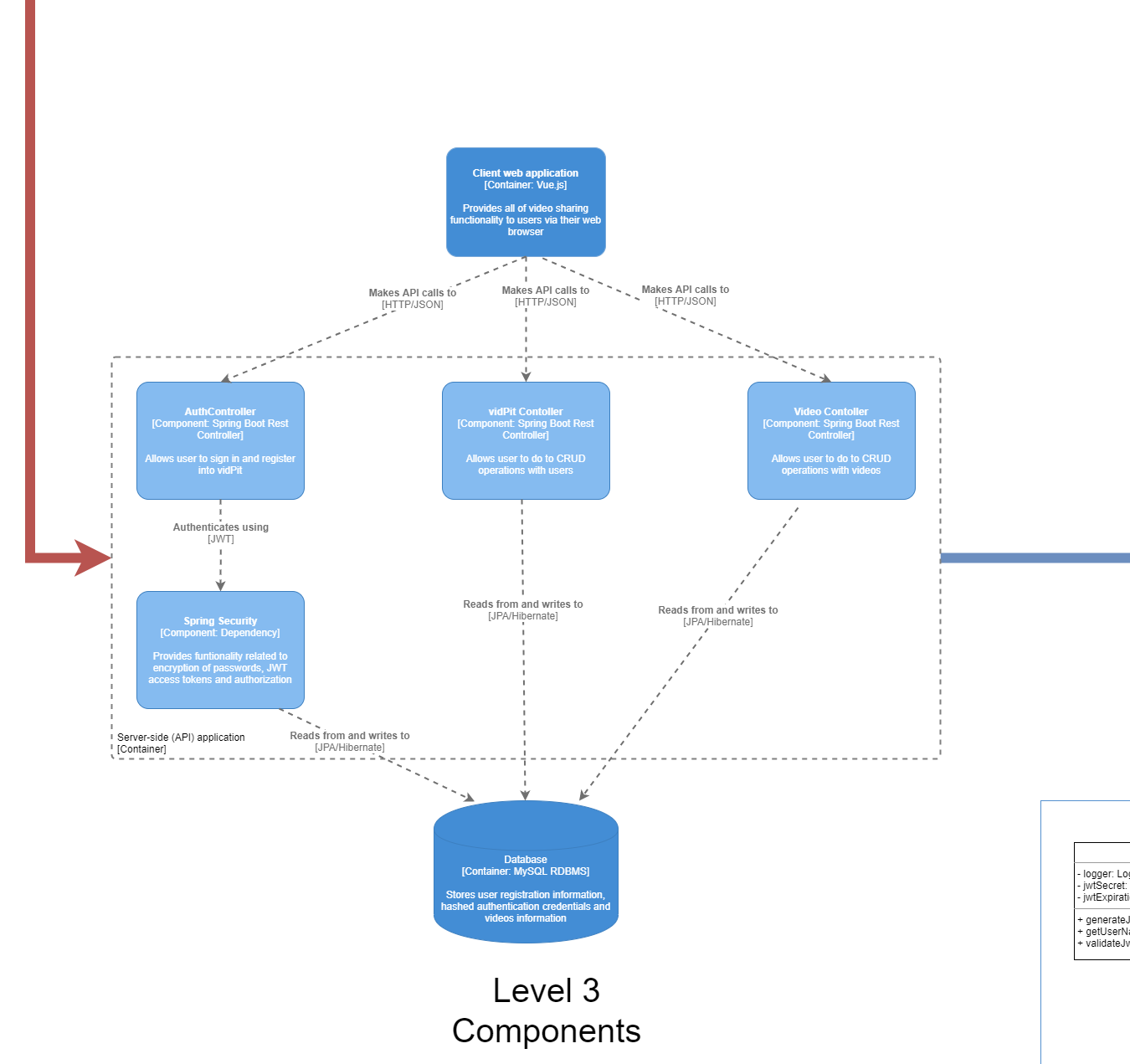


Figure : Level 3

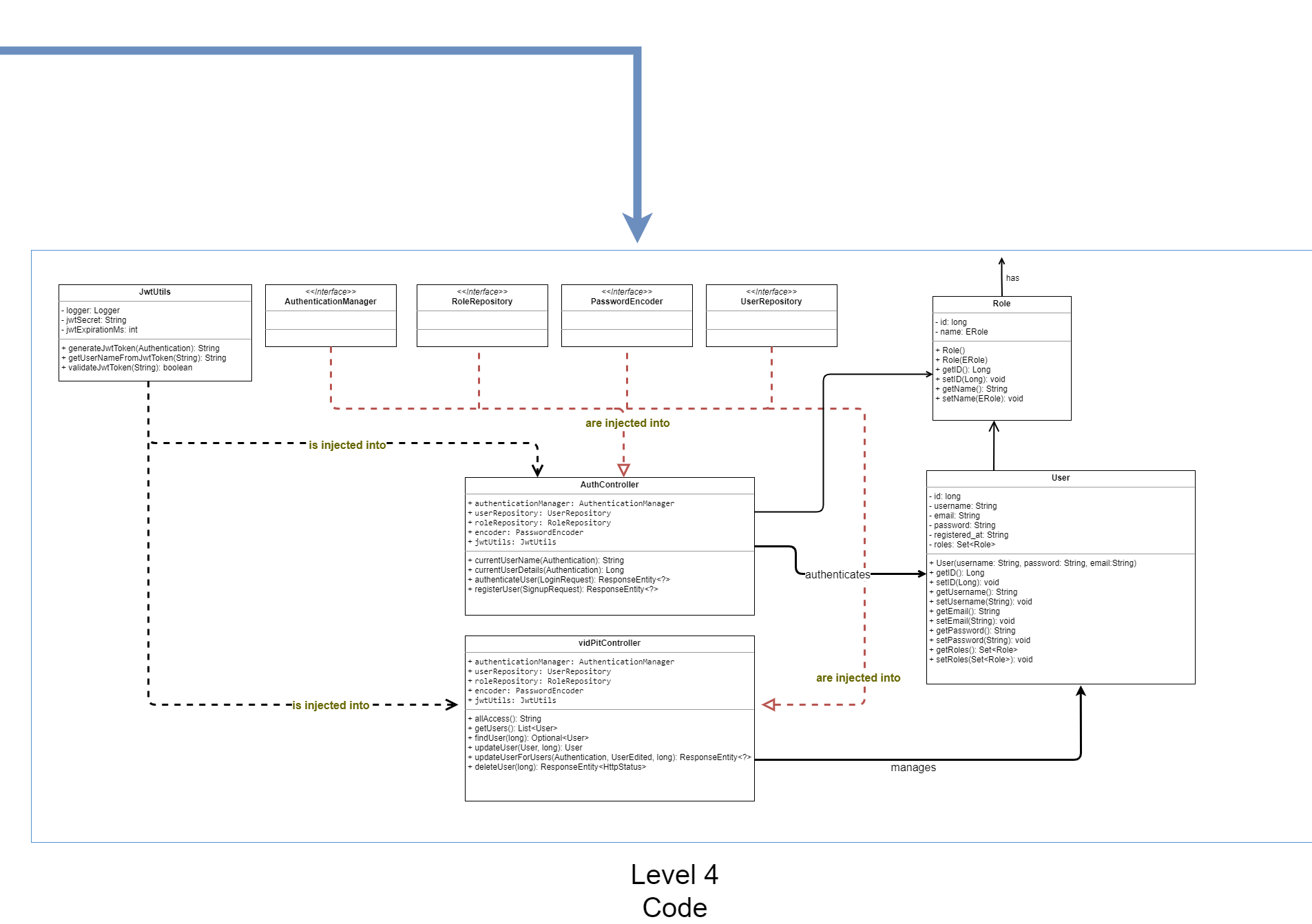


Figure : Level 4 – UML

## The OWASP Criteria relating to the application

Legend:

* **Green** – means that meets the criteria
* **Yellow** – means that partially meets the criteria
* **Red** – means that doesn’t meet the criteria
* **Grey** – means that is not applicable to the application

