**Research document**

Authentication/Authorization in Spring boot

1. **Problem statement**

The purpose of this document is to showcase the process involved with researching authentication in spring boot.

The context of the document is my project, a video game store. Its idea is to have users register and purchase games from a website. Additional features such as shopping cart and chat room are included. As there are 2 types of users – customers and admins, the purpose of authorization in this context is to restrict the customers from accessing routes and resources that they should not be able to.

1. **Main question**

How to implement authentication and authorization in a spring boot backend application?

1. **Sub Questions**
   1. How to implement a secure login functionality for my app?
   2. How to distribute permissions to different users based on their roles?
2. **DOT Framework research**

For answering my questions, I have chosen the *Choose fitting technology* pattern, because I am trying to find a technology that can be used to solve my problem. The methods I have chosen are the once I consider most relevant to the respective question.

* 1. How to implement a secure login functionality for my app

# 1. Available product analysis

# Login functionality has been implemented before and there are available example projects that showcase and explain how implementation is done.

# 2. Problem analysis

# *Why?* Secure authentication is required in any application where the user account stores valuable information regarding the user, or any assets such as balance in the form of a real currency and products that are purchased with it. In order for the authentication to be secure one must avoid sending user credentials in raw format (e. g. without any sort of encryption).

# 3. Requirement prioritization

# The requirements for authentication are an app that allow users to register securely using their email. The users should also be able to log in safely and not worry about their account credentials being hijacked.

# 4. Security test

# Security is the most important topic for this research document. A security test will be performed according to the OWASP criteria in a separate document.

* 1. How to distribute permissions to different users based on their roles

# 1. Best good and bad practices

# Since this is a part of a Java Spring Boot application, adhering to the SOLID principles is a priority. Regarding practices that are specific to authorization, I am going to follow the ones used in the tutorial I have chosen to learn from.

# 2. Problem analysis

# *Why?* Authorization prevents users from accessing resources they are not supposed to, such as seeing or editing someone else’s personal information. Authorization is definitely required in an application where multiple users can upload, edit and delete different pieces of content.

# 3. Requirement Prioritization

# Priority would be that uploading a game is restricted to registered users only so that admins have control over what is uploaded. It is important to also restrict the users from editing games that are not uploaded by them. Same goes for the profiles of other users.

# 4. Non-functional test

# Authorization allows the system to function the way it is intended to. By appropriately limiting the authorities of the users, usability and reliability inside the application are guaranteed.

1. **Answering the sub questions**

After the *Available product analysis* step, I have decided to follow [one of the available tutorials](https://www.bezkoder.com/spring-boot-react-jwt-auth/?fbclid=IwAR2iHheIAQuHeRH_1WiWcZvM58jrXhHs8NjYaeZGzBwC8K3Fb2qhS5bcfPY#Spring_Boot_React_Authentication_example) on how to use Spring security to add both authorization and authentication to my application.

1. **Results**

After conducting some research regarding authorization and authentication I have found that both of the problems above can be solved using JWT. It is a standard for securely transmitting information between parties as a JSON object. It can be used to digitally sign a request in order to verify who the signer is and give them the required permissions in the system. The tutorial I have chose to follow implements it suing Spring Security.