**Design Document**

Video game store

|  |  |
| --- | --- |
| Author | Ivan Koitchev |
| Last update | 26.09.2021 |
| Version | 1.0 (Sprint 2) |

Table of Contents

**Introduction2**

**Project Description & scope**

**Design Choices**3

Frontend

Backend

Database

**C4 Architecture4**

1. **Introduction**

This document contains all the pieces of documentation that are relevant to my project. Its purpose is to guide the reader through the various aspects of the project, such as architecture and use cases, as well as explain my design decisions where relevant.

1. **Project description & scope**

A website where a user can create an account and purchase video games online. Purchases happen through via shopping cart and each user can add funds to their balance (real payment implementation is not in the scope of the project, it is just a placeholder). Each user can upload a new game, but it must be approved by the website admin.

People who have purchased a game can give feedback regarding the product in the form of a review. Reviews are displayed on the page of the related product. There is a search a game functionality, which includes an option to filter search by number of purchases, genre etc.

Users that have their password forgotten can request a password recovery by receiving an email, containing possibly a new randomly generated password. Users can edit their profile, change their username, email, and password as well as non-essential information such as bio, age, address, payment info.

Each user can delete or edit the information about a product in case the same user is the one that had it uploaded. If more than one property is changed (genre, picture, description) an admin approval might be required for the change to be made. An admin can only delete or approve a product, but not edit. Users will be able to see statistics about how much they have spent and on what products.

1. **Design Choices** 
   1. Frontend

ReactJs is a library used mostly for the purpose of creating single-page web applications. One of its biggest strengths is that it has a strong community behind it. This is important for me because I can easily find required information via a google search, which greatly speeds up the development process. Therefore, it suits the requirements for a frontend framework for this project.

* 1. Backend

I chose to use Spring boot for the development of my API because it is a powerful framework which is excellent for creating a backend service. It also has a strong community behind it which means it is easy to find help if needed during the process of implementation.

Spring boot is used to build stand-alone and production ready Spring applications. Some of its main strengths are the shorter development time and efficiency increase, which are achieved through a default setup of unit and integration tests. It also provides flexible XML configurations, robust batch processing, database transactions, easy workflow, along with a wide variety of tools for development.

* 1. Database

My choice for a database is MySQL. It offers a high performance and on-demand scalability, which means it able to facilitate the management of deeply embedded apps using a smaller footprint, even in massive warehouses that stack terabytes of data. It is also easy to get started because the setup does not take too much time. Finally, being open source offers more flexibility and support.

1. **C4 Architecture**
   1. **Diagram

      Description automatically generated**C1 – System Context diagram
   2. Diagram

      Description automatically generatedC2 – Container Diagram
   3. Diagram

      Description automatically generatedC3 – Component Diagram
   4. C4 - Code

The code diagram will be included later in the development process of the application