Design document

Date	**:**	**15/06/2022**
Version	**:**	**0.3**
State	**:**	**Finished**
Author	**:**	**Nikola Stankov**

Version history

Version	Date	Author(s)	Changes	State
0.1	10/3/2022	Nikola Stankov	Initial version	Finish
0.2	28/03/2022	Nikola Stankov	Improved version	Unfinished
0.3	15/06/2022	Nikola Stankov	Final version	Finished

Distribution

Version	Date	Receivers

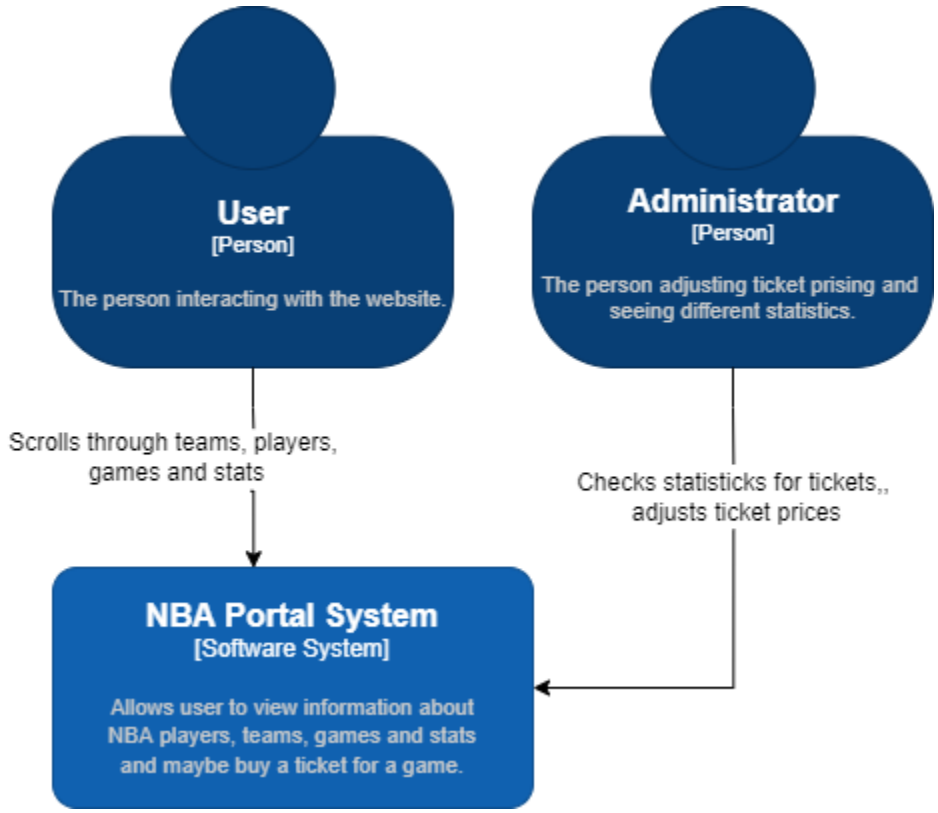
Table of contents

1. Introduction

The project which architecture is depicted in this document is going to be a NBA portal where users can view information about NBA games, teams and players and purchase tickets for future games.

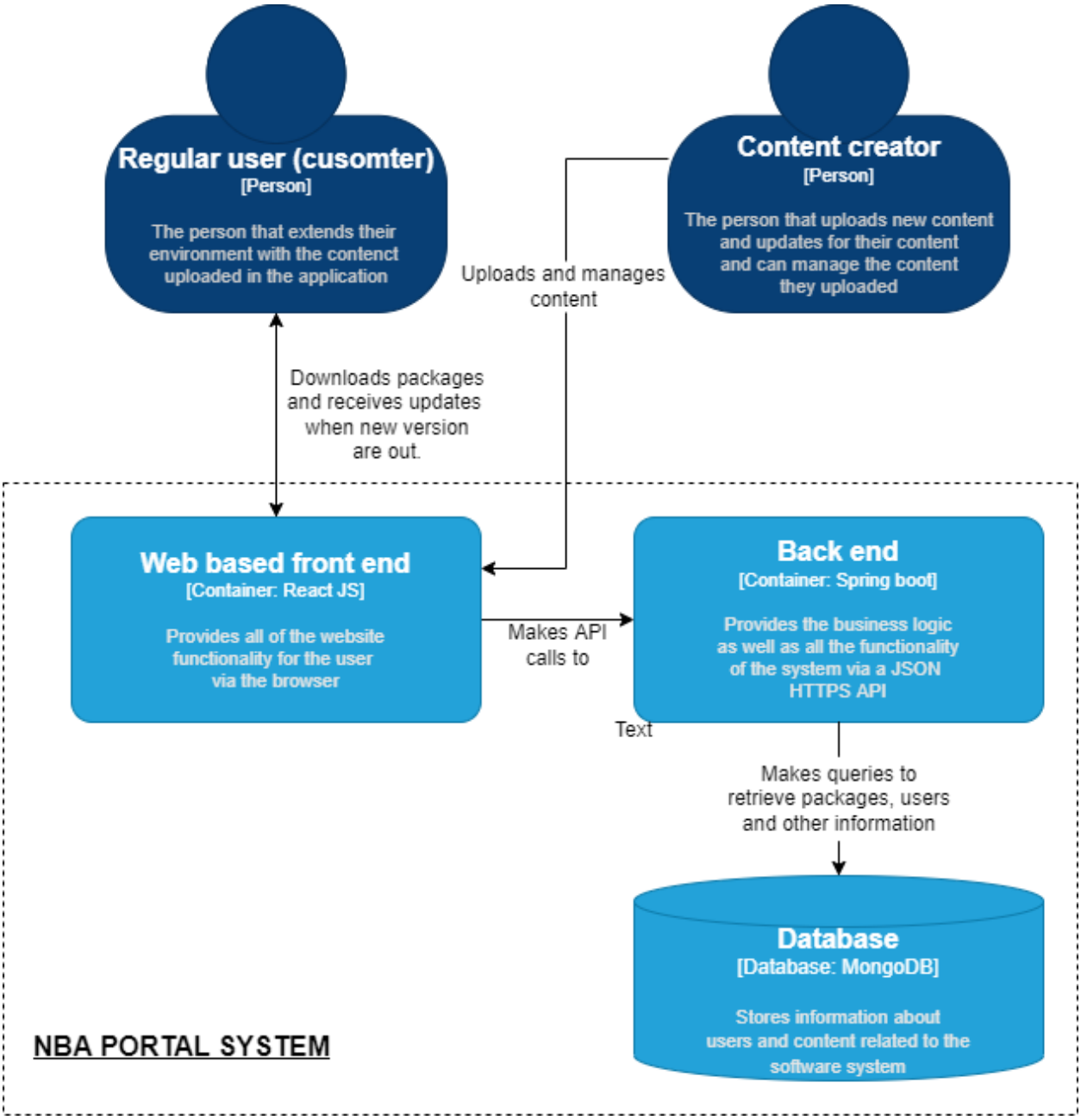
2. Architecture

2.1. System Context Diagram



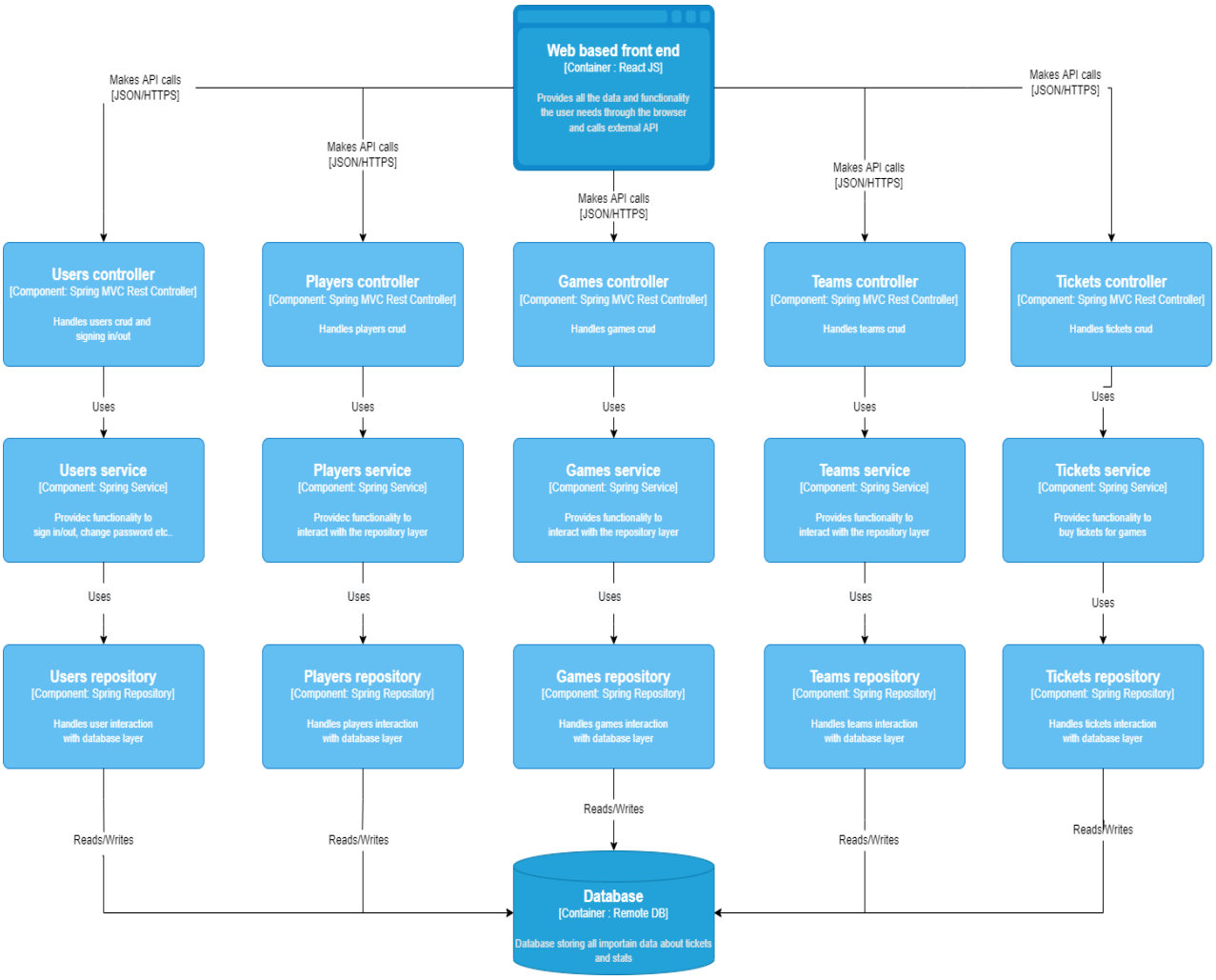
The system will be used by two types of users: a normal user and an administrator. They will interact with the system which will allow them to see information about teams, games, players and tickets, as well as make a purchase for a ticket for a specific game.

2.2. Containers Diagram

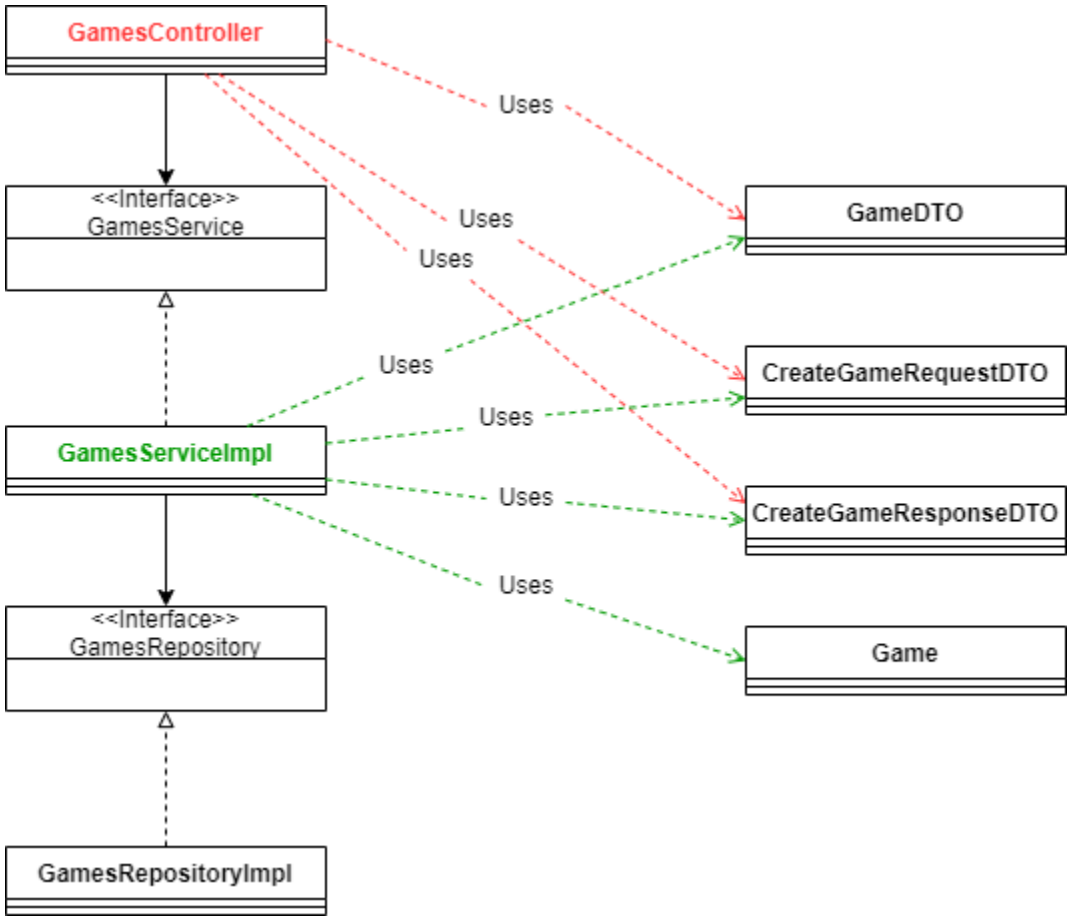


The system consists of a Web based front end which is build on ReactJS, a backend which holds all the logic and API endpoints which is build on Spring boot. The backend interacts with an internal database which allows the persistence of data for indefinite period of time.

2.3. Components Diagram

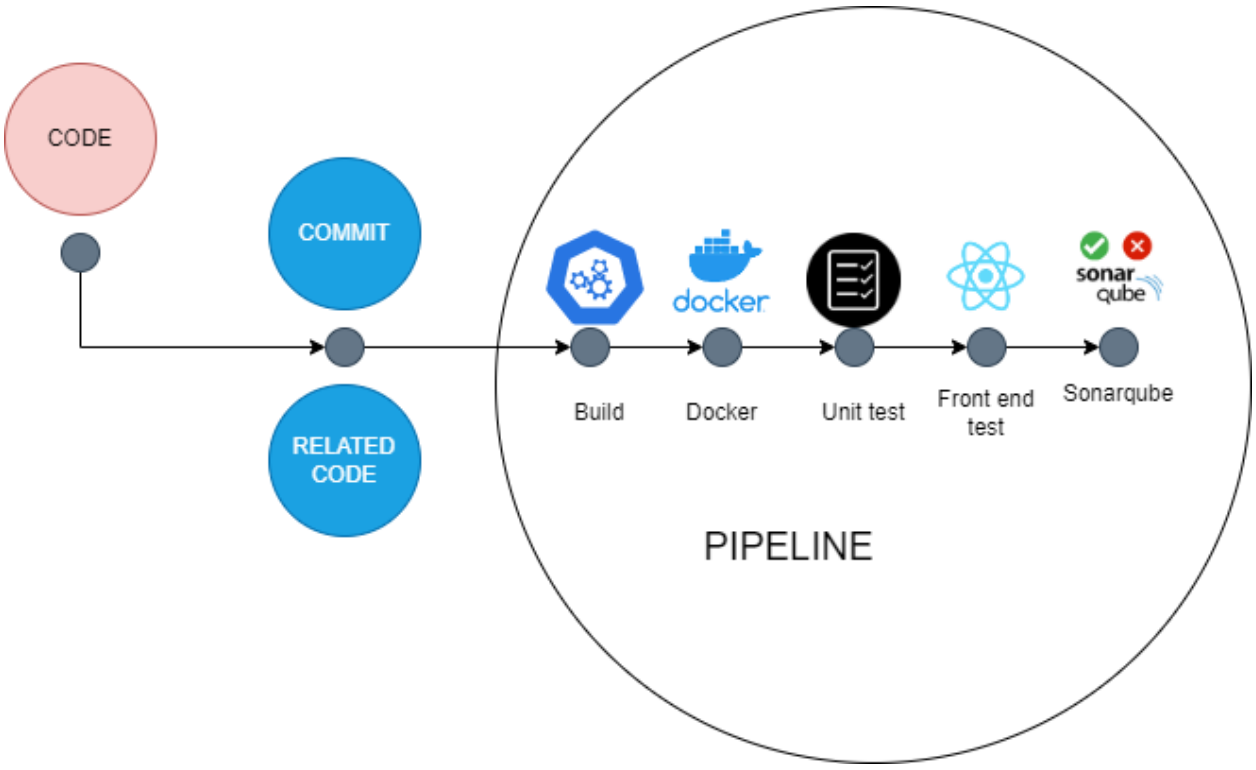


2.4. C4 Diagram



This is how every component in the application is set up. There is a three-layered design in place (Controller, Service and Repository) for every endpoint as well as dependency inversion with interfaces between the layers. There are models which are used by the Repository layer mainly while the other two layers make use of DTOs.

3. CI/CD Pipeline



This is how my CICD pipeline is set up in the moment. After I push to my repository the pipeline goes through 4 stages:

1. Build – in this stage the backend is being built and if it has been built with no errors the stage passes.
2. Docker – in this stage the frontend and backend images are being rebuild and the docker-compose start all 3 containers (backend, frontend and db).
3. Unit test – in this stage all unit tests for the backend are being executed. If all of them pass the stage passes.
4. Front end test – in this stage the front end is being tested. If all tests pass the stage passes.
5. SonarQube – in this stage the code quality is being assessed by SonarQube. If the code passes in SonarQube the stage passes.

4. Sequence diagram

4.1. Buying a ticket