Design Document

Viktor Skachkov

22-03-2023

Tutors: Márcio Paixão Dantas and Erik van der Schriek

Table of contents

Съдържание

Table of contents 2
Versioning Table 3
Introduction 4
System Context 6
Containers and tech choices 7
Components 8

Versioning Table

Version	Changes
1.1	I created the document

Introduction

Company Red Scar wants a website where they can sell their videogames because they believe this would boost their sales. It has tasked us to create the website which allows users to buy videogames and additions, give reviews to them and read news articles. It should also allow employees to add, edit and remove games, additions and news articles.

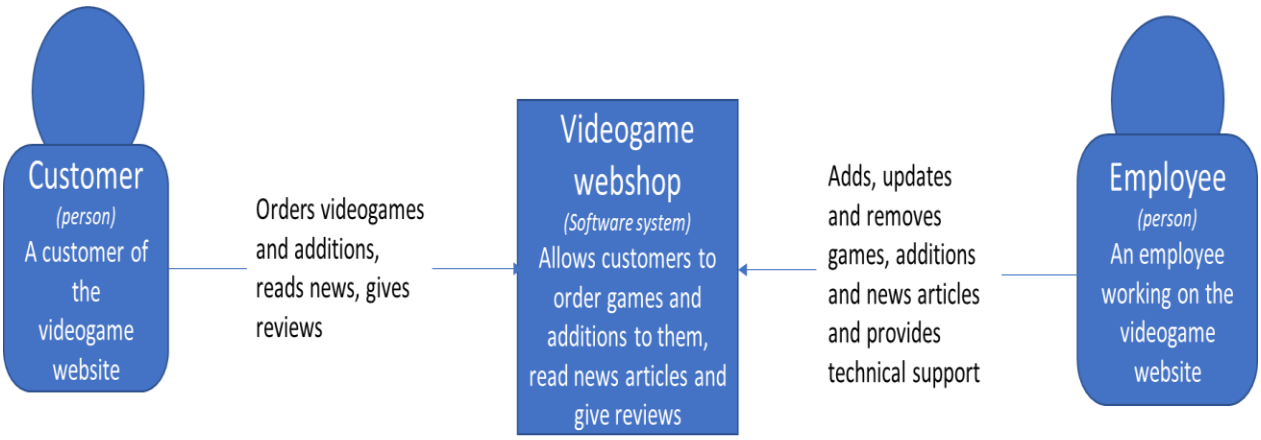
Justification of Decisions

I use React for the frontend because compared to other frontend frameworks, the React code is easier to maintain and is flexible which saves huge amounts of time and costs. It builds interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

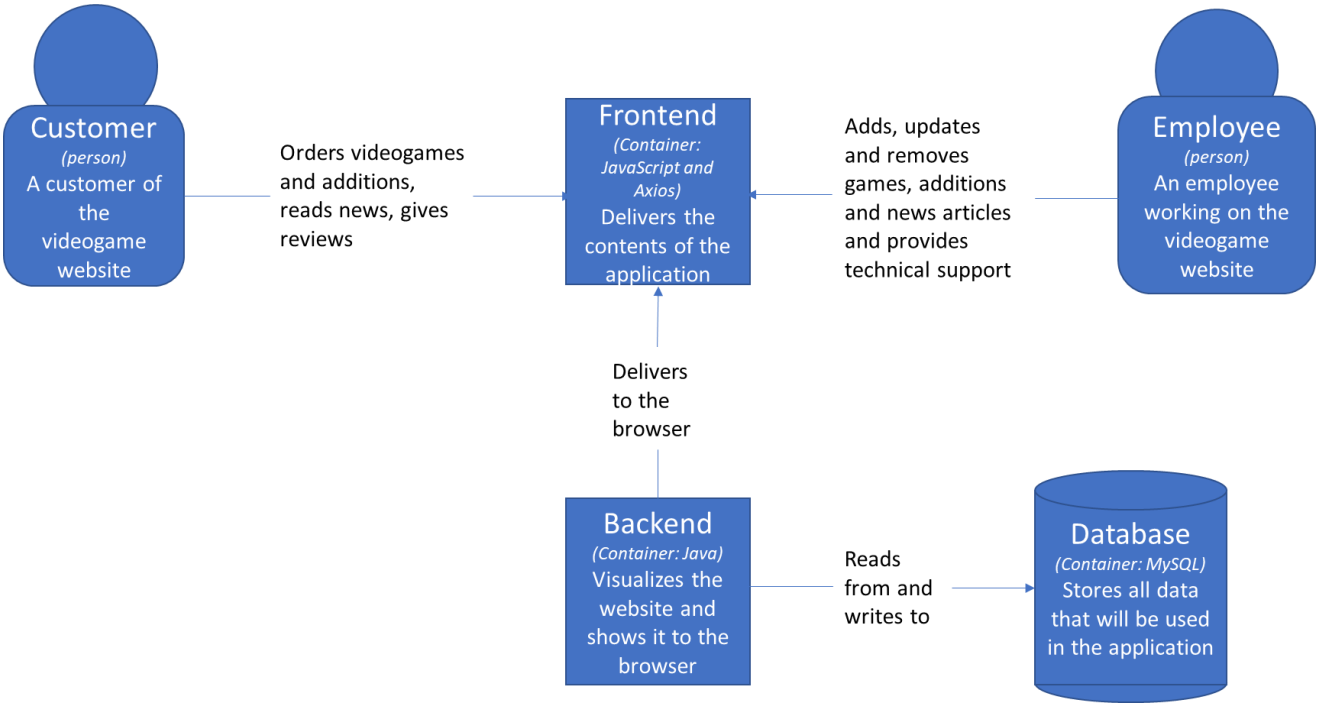
I use Java for the backend because it's a platform-independent language which means it can run on different operating systems, has efficient memory management, and it supports high scalability. It was designed to be easy to use and easy to write, compile, debug, and learn.

I use MySQL because of its high-performance query engine, tremendously fast data insert capability, and strong support for specialized web functions like fast full text searches.

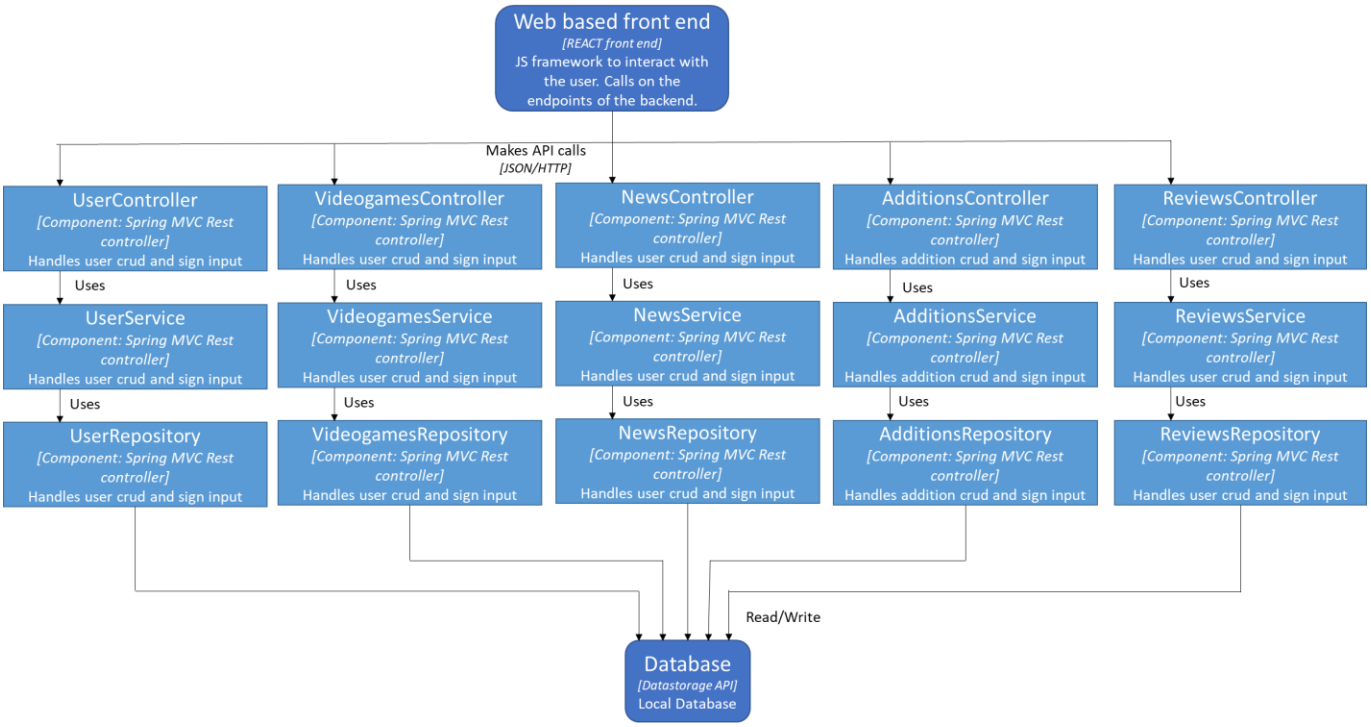
System Context



Containers and tech choices



Components


Web based front end
[REACT front end]
JS framework to interact with the user. Calls on the endpoints of the backend.
Makes API calls
[JSON/HTTP]
UserController
[Component: Spring MVC Rest controller]
Handles user crud and sign input
VideogamesController
[Component: Spring MVC Rest controller]
Handles user crud and sign input
NewsController
[Component: Spring MVC Rest controller]
Handles user crud and sign input
AdditionsController
[Component: Spring MVC Rest controller]
Handles addition crud and sign input
ReviewsController
[Component: Spring MVC Rest controller]
Handles user crud and sign input
Uses
UserService
[Component: Spring MVC Rest controller]
Handles user crud and sign input
VideogamesService
[Component: Spring MVC Rest controller]
Handles user crud and sign input
NewsService
[Component: Spring MVC Rest controller]
Handles user crud and sign input
AdditionsService
[Component: Spring MVC Rest controller]
Handles addition crud and sign input
ReviewsService
[Component: Spring MVC Rest controller]
Handles user crud and sign input
UserRepository
[Component: Spring MVC Rest controller]
Handles user crud and sign input
VideogamesRepository
[Component: Spring MVC Rest controller]
Handles user crud and sign input
NewsRepository
[Component: Spring MVC Rest controller]
Handles user crud and sign input
AdditionsRepository
[Component: Spring MVC Rest controller]
Handles addition crud and sign input
ReviewsRepository
[Component: Spring MVC Rest controller]
Handles user crud and sign input
Read/Write
Database
[Datastorage API]
Local Database