DESIGN DOCUMENT

"Cinema_Now"

Andreea Șindrilaru
Fontys University of Applied Sciences
S3-CB-01

Table of Contents

1. Introduction

1.1. Purpose of the document

The software design document is meant to provide a clear description of the design of the web application for the individual project of the 3rd semester. The document's purpose is to provide a clear understanding of what needs to be built and how it is expected to look like at the end. The design document will cover the frontend as well as the backend of the software, thus giving a clear insight of every aspect of the design of the application.

1.2. Document Overview

This document serves as a description of the functionality and software design for Cinema_Now application.

1.3. Definitions, Acronyms and Abbreviations

API : is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

Interface: A software interface may refer to a wide range of different types of interfaces at different "levels": an operating system may interface with pieces of hardware. Applications or programs running on the operating system may need to interact via data streams, filters, and pipelines and in object-oriented programs, objects within an application may need to interact via methods.

Axios: is a promise-based HTTP Client for node.js and the browser. It is isomorphic (= it can run in the browser and node.js with the same codebase). On the server-side it uses the native node.js http module, while on the client (browser) it uses XMLHttpRequests.

OOP (Object-oriented programming) is a programming paradigm based on the concept of „objects", which can contain data and code: data in the form of fields (often known as attributes or *properties*), and code, in the form of procedures.

REST/CRUD means using the HTTP, GET, POST, PUT and DELETE operations to implement CRUD operations: - use of client-server architecture, stateless communication, etc.

C4 Architecture is an easy to learn, developer friendly approach to software architecture diagramming, assisting with communication inside/outside of software development/product teams.

Class is an extensible program-code-template for creating objects, providing initial values for state (member variable) and implementations of behavior (member functions or methods).

React (also known as React.js or ReactJS) is a free and open-source frontend JavaScript library for building user interfaces or UI components.

1.4. Reference Material

(Class (computer programming), n.d.)
(Spring Boot - Introduction, n.d.) (Pagination vs Infinite Scroll, 2020)

(Object-oriented programming, n.d.)

(C4 Model for visualising software arhitecture, n.d.)

(Why are Developers Planning to Drift Towards React Instead of Angular, 2021)

(Programming Principles, n.d.)

(What is an API? (Application Programming Interface), n.d.)

Getting Started. (n.d.). Retrieved from Axios-HTTP: https://axios-http.com/docs/intro

MULDERS, M. (2019, 09 16). *What Is Spring Boot?* Retrieved from What Is Spring Boot?:
https://stackify.com/what-is-spring-boot/

Software Interfaces. (n.d.). Retrieved from Wikipedia:
https://en.wikipedia.org/wiki/Interface_(computing)#Software_interfaces

2.System Overview

For the 3rd semester of ICT & Software Engineering, students were asked to make a full-stack web application. The project must contain certain guidelines and requirement by using the agile software development methodology. More information regarding the documentation can be found in the "Documentation" folder. The web application will work as a cinema app where clients can buy tickets for movies, where based on how often they visit the cinema, they will be granted the "isLoyal" title which will be taken into account by the admin, send complaints in case something has occurred during their visit, and where admins can update the movie projections, add new movies, view complaints, and view all clients. The main users of this application are client and admin. When buying a ticket, the clients will not be able to choose a seat at the moment.

3.System Architecture

3.1. Architectural Design

The application is developed keeping in mind the SOLID principles and making sure that security is provided to the users. Interfaces are used to ensure the connection to the database in order to avoid repetition in the implementation. The backend of the application is connected with the front-end by using HTTP requests with AXIOS in order to connect to the REST endpoints. The C4 diagram below shows how the system works.

C4 diagram consists of 4 levels: System Context(C1), Container (C2), Components (C3) and Code (C4).

LEVEL 1: SYSTEM CONTEXT
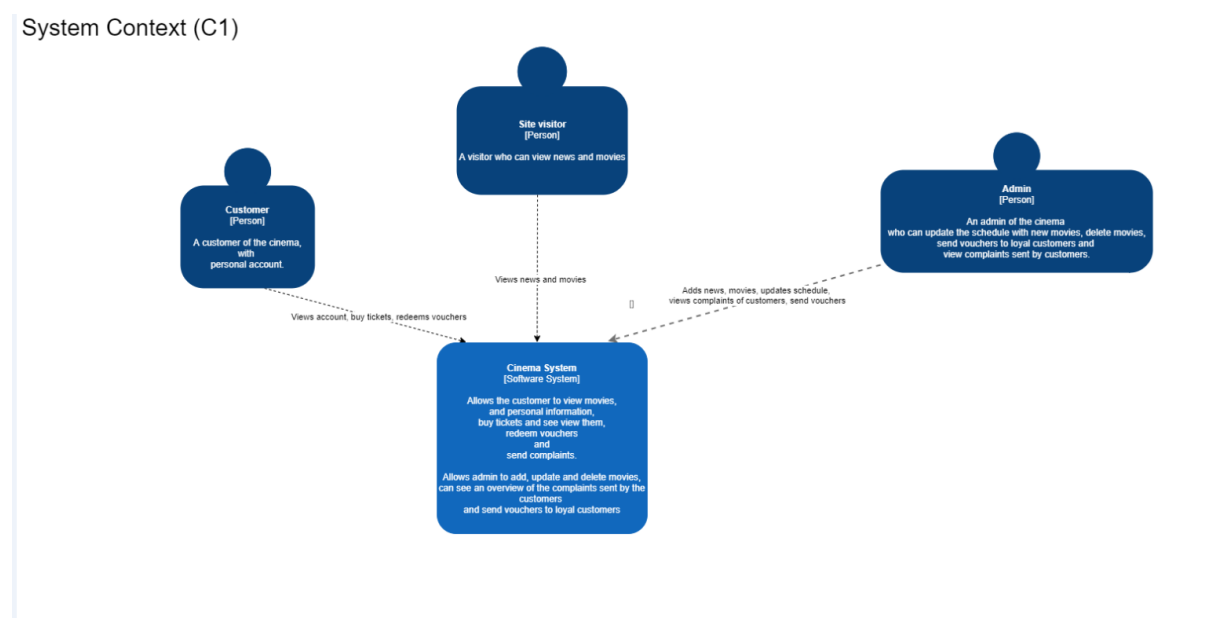
The above diagram shows the different users of the application: Site Visitor, Client and Administrator, each user having different type of accessing the website.

Cinema System is the main software part which allows the client to view movies and news, send complaints and buy tickets while the site visitor can only view news and movies. The administrator is the one responsible of managing the clients, movies, complaints, vouchers etc.

The system uses an email system so that the client can receive offers, vouchers and reset password.

LEVEL 2: CONTAINERS
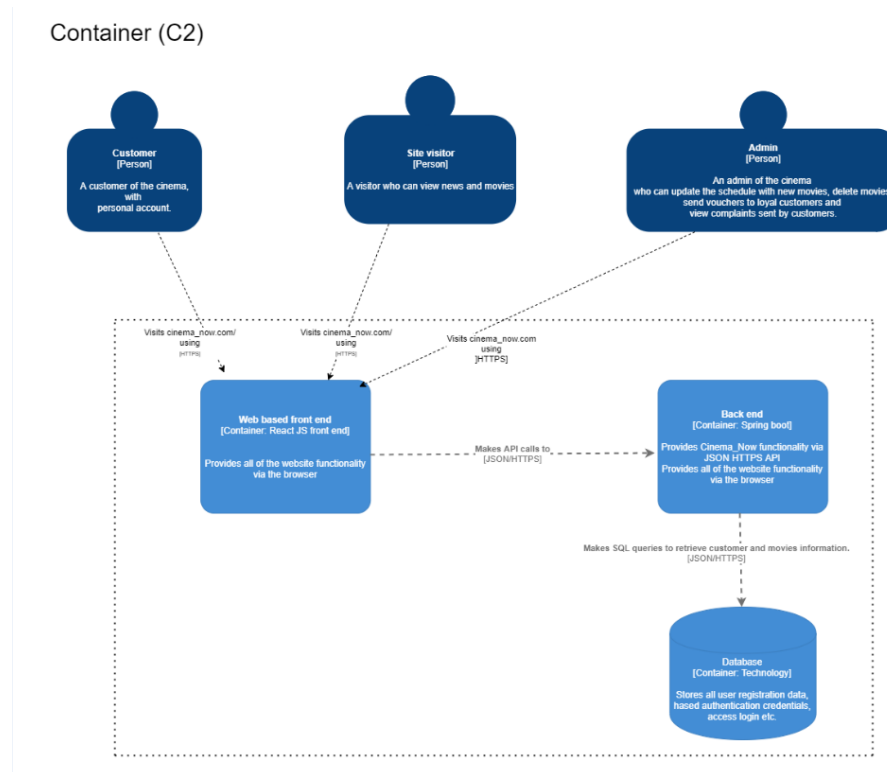
This diagram describes the connection between users and systems, and between back-end and front-end with the database.

Front-end provides the UI and all website functionality within the browser. Users and admins user the UI to interact with the backend.

Back-end provides all the data and logic for the website by using Sprint Boot and JSON, therefore making the front-end functionalities possible.

Database is used to store all the data of all users/content and is directly interacting with the back end.

LEVEL 3: COMPONENTS

Components (C3)
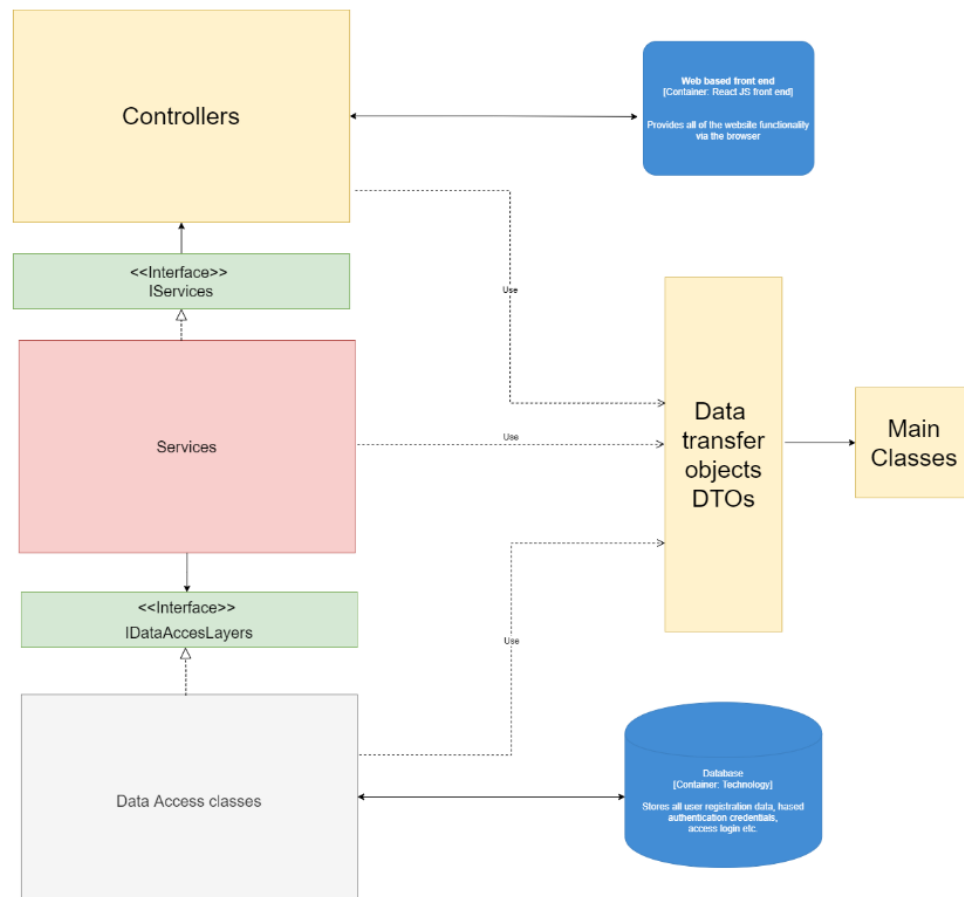
Level 3 represents all components which are used in the functionality of the application.

Controllers are used in order to connect the back end with front-end. With the help of interfaces, the connection with the service components is flexible.

Service components are the logic which the controllers basically use. They are directly connected to data access layers and can CRUD data.

DTOs are used for transferring the relevant data to the front-end. We can't transfer the whole object; so, "subclasses" (DTO's) of the main ones are used to send only the relevant information.

Database is used to store all the data of all users/content and is directly interacting with the backend.

3.2. Software choices

In order to back up the software choices for this project, the DOT framework methodology has been used to see which technologies would fit best for this project.

3.2.1 Spring Boot

According to (MULDERS, 2019) Spring Boot is an open-source micro framework, maintained by a company called Pivotal which provides Java developers a platform to get started with an auto configurable Spring application. Therefore, Spring Boot is a good framework to get started with the back-end part for full-stack web development, especially for beginners thanks to the fact that it reduces development process and increases efficiency by having a default setup for unit and integration tests. Therefore, thanks to its features, we are obliged to use Spring Boot this semester for this individual project.

Spring Boot offers some advantages such as productivity, reduction of development time and it's easy to understand. The framework's goals are to avoid complex XML configurations (this is a big plus especially for people who are beginning full stack web development) and developing Spring applications in an easier way.

The most important thing about Spring Boot would be that everything is auto configured, so there is no need for manual configurations. It is worth mentioning that the Java version used for this application is Java 11.

3.2.2 Separating Concerns

Separating concerns stands for splitting a computer program into distinct sections. That would mean that each component of the app has to follow a layered design (e. g. 3 presentation layer – logic layer – data access layer). By using this design principle, the code is easier to understand and can be extended with new features.

3.2.2 React JS

Nowadays, there are a lot of frameworks to choose from, ranging from React JS, Angular, Vue JS, Django etc. each one having its unique functionalities depending on the complexity of the project. Since, Cinema_Now is not a very complex project, it has been decided to use React JS. This framework works with components, each function being considered as a component. Therefore, separating concerns is crucial. For beginners, React is much more easier to get started with comparing to the other mentioned frameworks, this being the reason why we have been asked to use it for this semester by our teachers.

Also, React is quick, efficient, works with an MVC template and makes creating front-end more easy. Thanks to the fact it has a large community, you can easily find anything about different problems you might encounter while creating your application.

3.3. CI/CD Pipeline Diagram
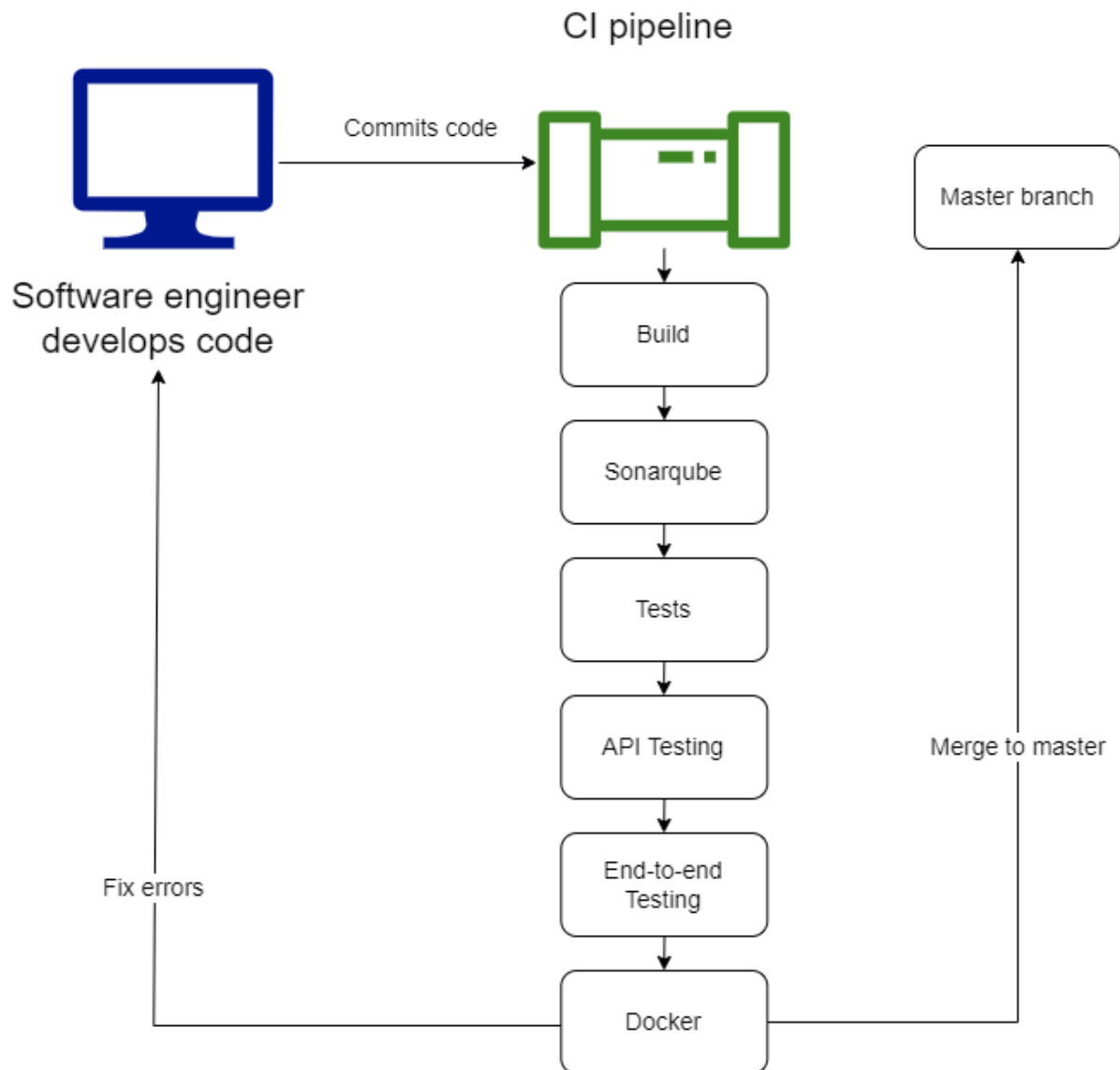
The application needs to be checked each time a new feature is added, regarding functionality, so that in case there are errors, they can be fixed in time.

Sonarqube is used to check the quality of the code, possible bugs or vulnerabilities within the backend. After this stage, all unit tests, API tests and end-to-end tests are run to make sure everything works accordingly.

Lastly, the Docker stage comes up. In this stage, the images for the frontend and backend need to be updated with to the latest version of the application.

After the pipeline passes, everything is pushed to the master branch.

3.4. Design choices

Usage of JPA

As mentioned in the software choices, Spring Boot will be used for the backend of this application. In order to retrieve the data from the database, it has been decided to use JPA instead of the classical JDBC so that the needed effort for the data access layers would be reduced significantly. Main benefits of using JPA are the input sanitization which increases the security aspect of the application and provides simple queries (for example, if I want to retrieve a user by id, I will just make a method called findUserById and JPA will run the query automatically). It also allows the possibility to create custom queries to retrieve data.

Usage of UUIDs

In order to improve the security of the application, it has been decided to use UUIDs for the entities instead of Long. A UUID is more difficult to remember than a simple number, making hacking into the system to retrieve information way harder, especially if the attacker doesn't know what he is looking for.

Storing pictures for movies

In general, there are multiple ways to store files. The main way would be to store the file locally which would have no impact on the project, but it would have on the device's memory as if it were to store thousands of files, it would run out of memory. A much better approach would be to store them in the database by using LOB, as it allows file storing into the database. The counter side of it would be that if the files grew in size, it would have a negative on impact on the performance of the database. For now, I will store the pictures in the database as storing them in my laptop would impact the performance. In the future, I will research and come up with a better solution.

Pagination

According to (Pagination vs Infinite Scroll, 2020) pagination is a method to separate digital content into different pages on a website. Users can navigate between pages by clicking the links, which are normally found in the bottom of the page. Nowadays, it is used on every web application.

Pagination should be used for the admin side, as the data is displayed in tables to make visualization easier. However, due to the implementation time, it will not be added, keeping in mind that if there are thousands of records and there is no pagination, the performance of the app would go down significantly. In the future, I will add pagination in the early stages of the project.

WebSockets

In order to make the application more interactive, it has been decided to use WebSockets to create a chat system so that users can socialize and discuss the latest news about movies. Each user's username will be displayed underneath the message and depending on who is the receiver/sender, the popup will display in the right/left side of the chat.

Spring Security

One of the core features of the application is security. Since only as a user, you can buy tickets, it is crucial to make sure that the authentication process takes place.

Also, it is important to make sure what each user can do. For example, if am a user, then I cannot access the users page which can be accessed only by the admin. Therefore, JWT will be used to check what claims does each user have.

For now, the token is stored in the local storage of the browser, even though it is unsafe. A better solution for this would be to store the token in a HTTPOnly Cookie, because the localStorage/ sessionStorage are vulnerable to attacks.

Seat Selection

Since Cinema_Now is a web application to buy tickets, the user should be able to select seats. Due to implementation time in the frontend, I have decided not to implement it for now. The backend part is quite easy to implement, we have a Seat object which has a relation of 1 to many with Room class. The frontend implementation takes more time than the backend one, since checks are needed to see if a seat has been already taken and also the CSS part would take considerably a lot of time.