

Deliverable 1

~CINEMA PROJECT~

STUDENT NAME:BIRISAN DENISA OPHELIA

GROUP:302236

# CUPRINS

[1. Project specification 2](file:///C:\Users\Asus\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297885)

[2. Functional requirements 2](file:///C:\Users\Asus\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297886)

[3. Use Case Model 3](file:///C:\Users\Asus\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297887)

[4. Suplementary specification 5](file:///C:\Users\Asus\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297888)

[5. Glossary 6](file:///C:\Users\Asus\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297889)

# 1.Project specification

For my semester project, since I have quite a passion for watching movies, I’ve chosen to create my own personal cinema so that I could always choose the movies and provide the best quality and variety for my future clients.

The backend will be implemented in Java, via usage of the well known framework Spring Boot, whilst for the front-end I will learn to code it in React.

My take on the cinema would be: enhancing the customer experience by implementing a user-friendly ticket booking system, showcasing the vision by organizing Q&A sessions with directors and actors, introducing a loyalty program to reward repeat customers and incentivize them to visit more frequently, collaborating with independent filmmakers to exhibit their work and offer a platform for emerging talent.

Main programmatic objectives are: better understanding of the Spring framework, tying the site to MySQL, back-end to front-end integration, knowledge in React.

# 2.Functional requirements

* **Login/Register:**

A new client has to either create a new account or to login to his existing one. In the first case, the back-end will take care of the validation of data so as not to sign-up with invalid credentials (e.g. invalid email address). As for the admin part, the could register, but he would have to wait for approval from all the others admins (this part could be implemented via email), to ensure security. Depending on which kind of user is logged in, he/she would be redirected to the according web page, with different functionalities. The login for client would allow to account for the number of orders made so as to reward with bonus points and encourage loyalty.

* **Functionalities for Client:**

Up to date, I’ve implemented the following CRUD operatons:

* **Movie Table:** find a movie by name, update the movie by changing the watchlist in which it is placed,find all movies in a watchlist
* **Review Table:** save a review, find all reviews with a certain rating of stars, delete his/her personal review,update his/her review by changing the text
* **User Table:** update my age as a client
* **Ticket Table:** find ticket by name, price and seat
* **Cart Table: view all tickets under a certain sum in the current cart, delete all from the cart**

A much needed functionality would be to connect the site to a pay interface, where you could introduce you personal data safely and buy the ticket you wish for. The bill would be sent via email, in case the client wants to check that everything is alright.

Another idea would be to implement a Chat Bot which would give the client faster information regarding the shipping payment, details etc.

* **Funcționalități pe Admin**:

Up to date, I’ve implemented the following CRUD operatons:

* **Movie Table:** add a new movie**,** delete a movie from the site, modify the producer in case the previous admin got it wrong the first time around, delete a specific ticket for a movie or update the price and place for it
* **Review Table:** delete a review (no matter whose it is)
* **User Table:** delete a client if he/she has violated some rule
* **WatchList Table:** find a watchlist belonging to a client, find a watchlist containing a movie, replacing a movie with another movie in a watchlist, delete a watchlist
* **Ticket Table: update price for a ticket, update seat for a ticket**

The admin is the one responsabile for sending emails with the bill, as well as generating reports with the activity on the site that day: best seller movie (blockbuster), all movies sold in a weeks’ time etc.

In the future weeks, I might have to add other tables for the Admin, providing I would have to extend his/her functionalities to administrate the physical location and the personnel.

**3.Use Case Model 1**

### Use Cases Identification

*Use-Case1:  Register*

*Level:  User-Goal*

*Primary Actor: User*

*Main success scenario:  Successful register*

*Extensions: Format ilegal de date(email fără @, parola fara 1 uppercase letter, 1 symbol, 3 numbers )*

*Use-Case2:  LogIn*

*Level:  Subfunction*

*Primary Actor: User*

*Main success scenario:  Successful login*

*Extensions: -*

*Use-Case3:  LogOut*

*Level:  Subfunction*

*Primary Actor: User*

*Main success scenario:  Successful logout*

*Extensions: -*

*Use-Case4:  ModifyAge*

*Level:  Subfunction*

*Primary Actor: Client*

*Main success scenario:  the client changes his/her age*

*Extensions: He/She enters an invalid age number(negative or very high number e.g. 999)*

*Use-Case5: BuyATicket*

*Level:  Client-Goal*

*Primary Actor: Client*

*Main success scenario: The client searches for the movie he wants. He will be redirected to a page where he could see the available seats and prices for each of them. He selects the wanted ticket and then presses on the “Buy” button, afterwards being redirected to the paypal page.*

*Extensions: The client doesn’t have enough money to buy=>generate error message.*

*Use-Case6: ChatWithChatBot*

*Level:  Client-Goal*

*Primary Actor: Client*

*Main success scenario: The client asks the bot some questions and it delivers the basic, predefined answers*

*Extensions:The chatBot is not well enough equipped to answer.*

*Use-Case7: WriteAReview*

*Level:  Subfunction*

*Primary Actor: Client*

*Main success scenario:The client writes a review for a movie.*

*Extensions: -*

*Use-Case8: ModifyAReview*

*Level:  Subfunction*

*Primary Actor: Client*

*Main success scenario:The client modifies the comment for a review.*

*Extensions: -*

*Use-Case9: Delete/Insert/ModifyAMovie*

*Level:  Admin-Goal*

*Primary Actor: Admin*

*Main success scenario: The admin performs the specified CRUD operations.*

*Extensions:-*

*Use-Case10: GenerateABill*

*Level:  Subfunction*

*Primary Actor: Admin*

*Main success scenario: The admin send the bill on the email to verify*

*Extensions: The report contains wrong data.*

*Use-Case11: DeletAllItemsFromCart*

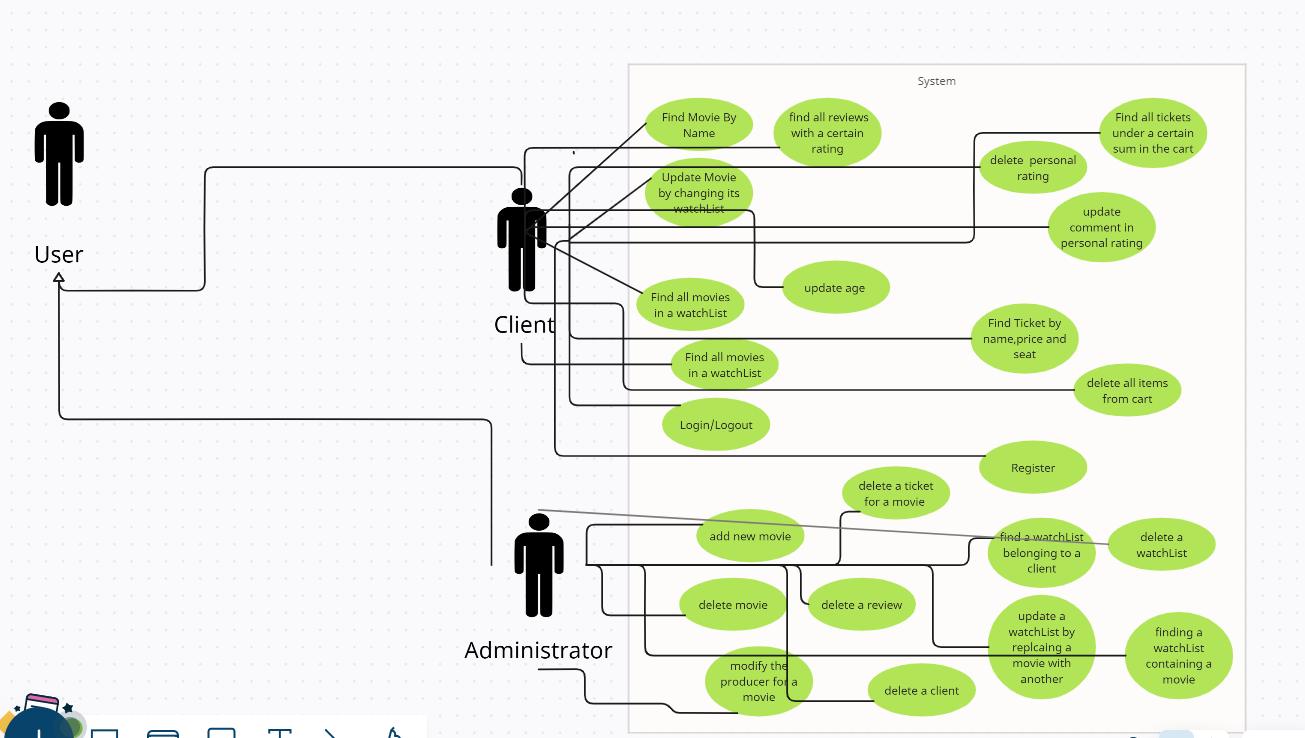
*Level:  Client-Goal*

*Primary Actor: Client*

*Main success scenario: The client chooses to rethink his/her options and not buy this time.*

*Extensions:-*

### UML Use Case Diagram



**4.** **Supplementary Specification**

**4.A. Non-functional requirements**

* *Usability*: how user-friendly the interface is, how easy it is to find what you are looking for quickly and how eye-pleasing the site looks like.
* *Accessibility*: this answers a simple question “How much harder is it to acces this website if I have a visual/hearing or motor impairment?”. To make life as comfortable as possible for there people, I would include a bot that would read the text as soon as you land on a certain page and to whom you can communicate and help you navigate smoothly.
* *Security*: I will implement an encryption of the password to make it difficult for a third party to steal personal data from clients, a security protocol for the paypal part.
* *Internationalization and localization*: The website would be able to support multiple languages and currencies.

**4.B. Design Constraints**

The site will be built in Java with Spring. I have also used: Mockito to test the CRUD operations.

The project is designed using the MVC design pattern. The model side is satisfied by all the entities used, the repos used to persist the data, as well as the services where I implemented the methods. The Controller and View part will be developed for the next assignments. For the testing of the controllers I will use Postman, while the front-end will be manufactured in React.

**5. Glossary**

* ***React*** is an open-source **JavaScript library** used for building user interfaces,created by Facebook and is widely used in web development. React uses a declarative approach.
* ***ORM (Object-Relational Mapping)*** is a programming technique that enables developers to interact with a database using high-level object-oriented APIs instead of writing raw SQL queries. This approach helps developers work with data more efficiently and easily, and can save time and effort in developing applications that require database access.
* **Hibernate** is often used as the default ORM framework for data persistence in Spring Boot applications. Spring Boot provides an environment that is easy to configure and manage, while Hibernate takes care of the underlying persistence and database management with minimal configuration.
* ***CRUD*** refers to the four basic functions that can be performed on data in a database or application:Create (inserting a new record), Read (querying the database for specific data),Update (modifying existing data in the database), Delete.