26/12
2022

LE CINÉ DU COIN

By Alexandre NGUYEN
Jason GOUVET
William TRAVERS
Valentin MAIA

SOMMAIRE

Le Ciné du COIN

1-PROJECT DESCRIPTION

For our final project, our team wants to create a cinema booking website. On this site, customers should be able to create an account and book their seats for their favorite movie.

When they make their reservation, they can select which time they want to book. Then, they are going to receive a reservation confirmation by mail

Le Ciné du COIN
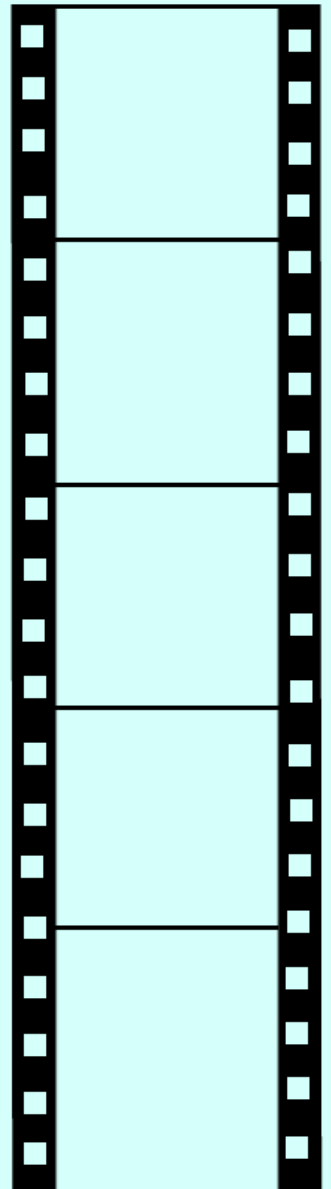
2-FEATURES

JAVA CLASS

```
package ucy.ece318.assignment3;

import java.sql.*;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.view.RedirectView;

@RestController
public class CustomerController {
    @Autowired
    final static String DB_URL="jdbc:mysql://sql7.freesqldatabase.com/sql7530714";
    final static String USER = "sql7530714";
    final static String PASS = "GWMQEgUDTC";
    static Statement stmt;
    static Connection connection;

    public static void connectToDatabase(){
        try {
            Class.forName(className: "com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(DB_URL, USER, PASS);
            stmt=connection.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

Customer Controller class that handles DB and customers

```
package ucy.ece318.assignment3;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Customer {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String name;
    private String surname;
    private String password;


    public Customer(String n, String sn, String pswrd){
        name = n;
        surname = sn;
        password = pswrd;


    }
```
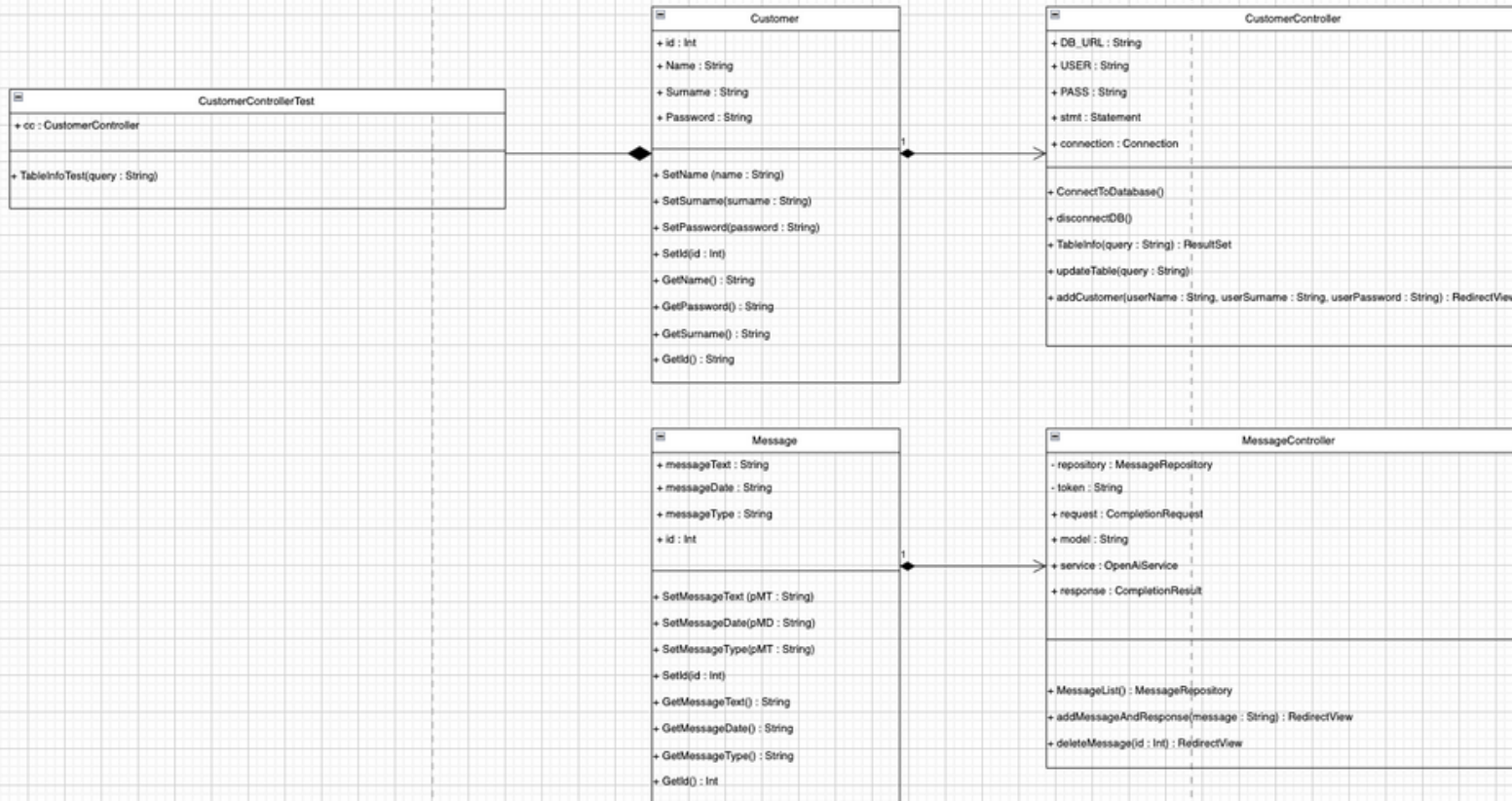
Customer class for the characteristics of a Customer

2-FEATURES
CLASS MAPPING

CustomerControllerTest

+ cc : CustomerController

+ TableInfoTest(query : String)

Customer

+ id : Int
+ Name : String
+ Surname : String
+ Password : String

+ SetName (name : String)
+ SetSurname(surname : String)
+ SetPassword(password : String)
+ SetId(id : Int)
+ GetName() : String
+ GetPassword() : String
+ GetSurname() : String
+ GetId() : String

CustomerController

+ DB_URL : String
+ USER : String
+ PASS : String
+ stmt : Statement
+ connection : Connection

+ ConnectToDatabase()
+ disconnectDB()
+ TableInfo(query : String) : ResultSet
+ updateTable(query : String)
+ addCustomer(userName : String, userSurname : String, userPassword : String) : RedirectView

Message

+ messageText : String
+ messageDate : String
+ messageType : String
+ id : Int

+ SetMessageText (pMT : String)
+ SetMessageDate(pMD : String)
+ SetMessageType(pMT : String)
+ SetId(id : Int)
+ GetMessageText() : String
+ GetMessageDate() : String
+ GetMessageType() : String
+ GetId() : Int

MessageController

- repository : MessageRepository
- token : String
+ request : CompletionRequest
+ model : String
+ service : OpenAiService
+ response : CompletionResult

+ MessageList() : MessageRepository
+ addMessageAndResponse(message : String) : RedirectView
+ deleteMessage(id : Int) : RedirectView

2-FEATURES

REST MAPPING

```java
@GetMapping("/addCustomer")
public RedirectView addCustomer(@RequestParam final String userName, @RequestPar
    connectToDatabase();
    ResultSet rs = tableInfo(Query: "SELECT COUNT(Customer_id) FROM Customer");
    int custID=rs.getInt(columnIndex: 1)+1;
    updateTable("INSERT INTO Customer values("+custID+",\'"+userPassword+"\',\'"
    disconnectDB();

    return new RedirectView(url: "");
}
```

This mapping uses the database, and insert a new customer into it

2-FEATURES

GRAPHICAL USER INTERFACES

```
<script type="text/javascript">
    $(document).ready(function(){
        var width = 1000;
        height = 570;
        var numberOfBlinds = 1;
        var margin = 0;
        var color = '#000';
        gapHeight = 0;

        var container = $('#container');

        container.width(width).height(height);
        var blindWidth = width / numberOfBlinds;

        images = new Array();
        $('ul li', container).each(function() {
            images.push($(this));
        });

        images[0].addClass('active');
        activeImage = 0;

        for (var i = 0; i < numberOfBlinds; i++) {
            var tempEl = $(document.createElement('span'));
            var borders = calculateBorders();

            tempEl.css({
                'left': i * blindWidth,
                border: margin + 'px solid ' + color,
                borderTop: borders.borderWidthTop + 'px solid ' + color,
                borderBottom: borders.borderWidthBottom + 'px solid ' + color,
                'height': height,
                'width': blindWidth
            });

            container.prepend(tempEl);
```
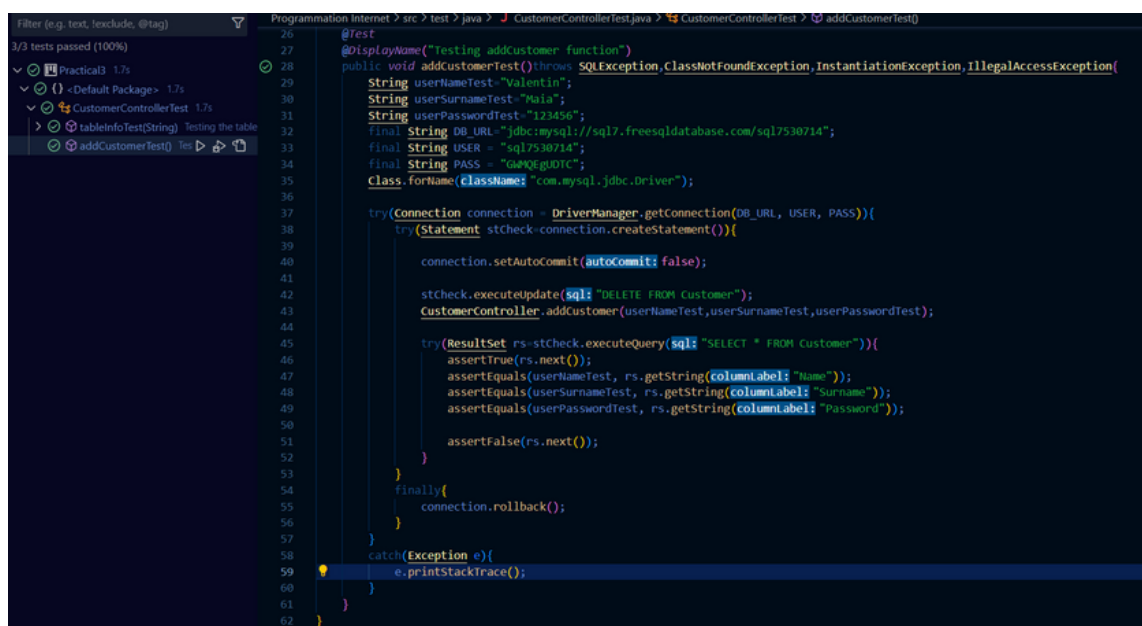
In this script, the movie posters defile one after the other with a fluid animated transition.

Le Ciné du LOIN

2-FEATURES

J-UNIT TEST

In the function shown above, we test the addCustomer function. In order to do so, we add a customer in our database, and then check if the data in the database are the same has the ones we used at the beginning.

3-PROJECT WEBSITE OVERVIEW

Header And Footer

Customers can access the menu by clicking the buttons listed here: Home, Movies, and Login to access different links. (Currently, we don't have enough servers to upload new pages)

Clicking the title or logo will take you to a link.

These are the buttons that customers can interact with. They glow white when you pass the cursor over them.
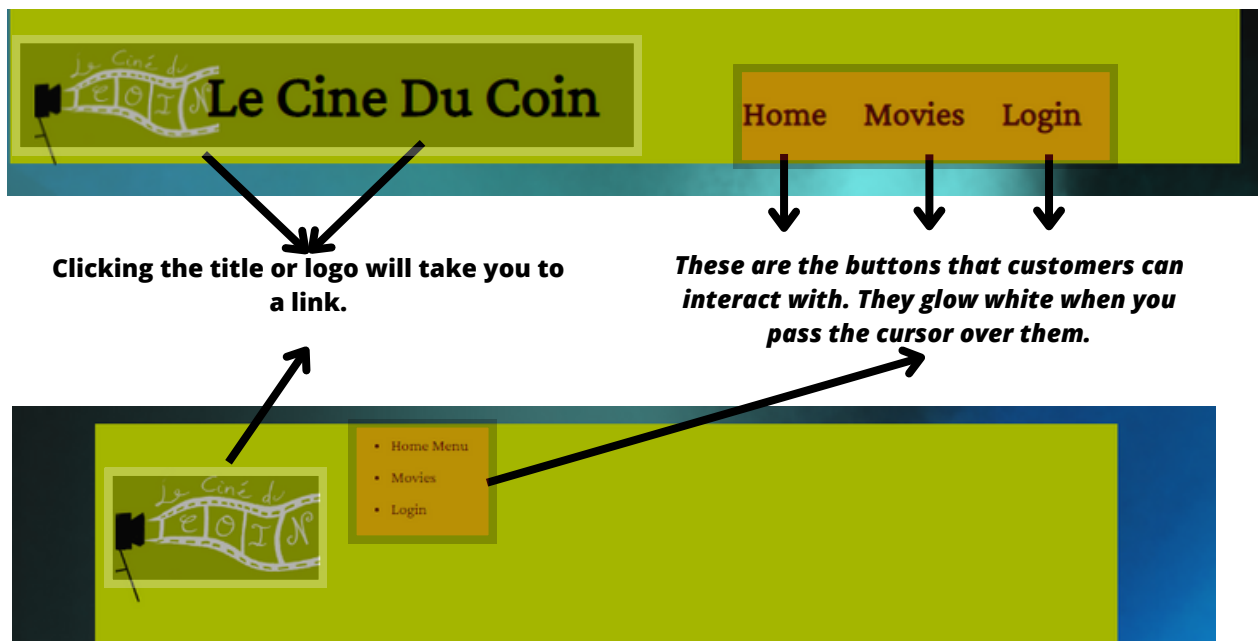
3-PROJECT WEBSITE OVERVIEW

Highlight

**With the help of JavaScript, pictures of movies are displayed with a timer transition animation.
The title and the board are clickable.**

They are interactive
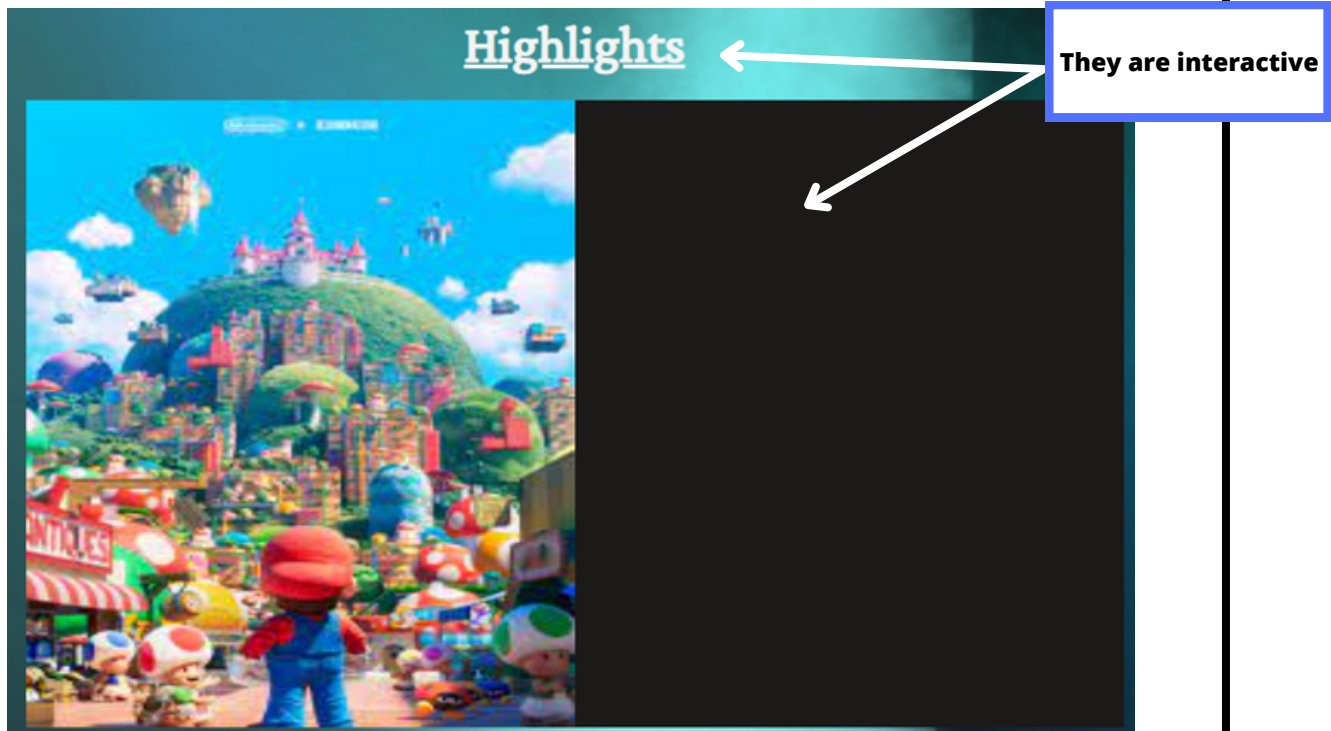
3-PROJECT WEBSITE OVERVIEW

Movies

Here you will find movie posters, titles, genres, synopses, and schedules for each screening.
The entire block is interactive. You can book a seat for the movie by clicking the different schedule buttons.
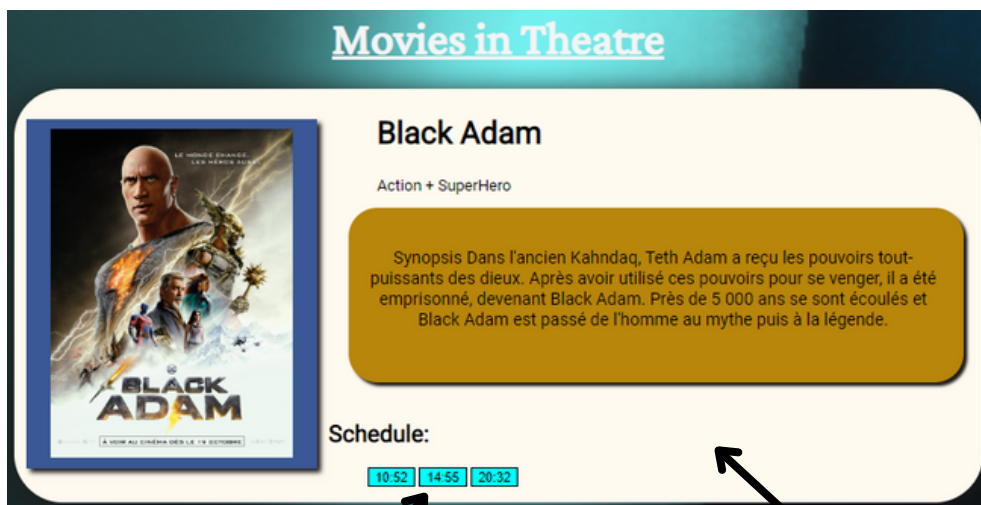
Clicking on it updates the database and books a place for the customer (see in the next page).

When you pass the cursor over the block, it changes color.

3-PROJECT WEBSITE OVERVIEW

Login

The customer can register on our site and create an account here.

There are three blocks associated with Name, Surname, and Password. In the corresponding section, customers can enter different information. By clicking on the submit button, the account is now created with the current information.

Insert the information in the following segment

Click on this button to create an account with the current informations

3-PROJECT WEBSITE OVERVIEW

We can see by adding a customer on the website that the customer is correctly added in the database.

3-PROJECT WEBSITE OVERVIEW

Chat bot

By inserting messages in the message bar and clicking the send button, you can chat with a bot. Then the bot is going to answer you like a "normal" person
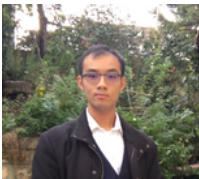
Conversation box

Type a message

Insert your message here

Click on this button to send your message

4-OUR TEAM

Alexandre NGUYEN

Role: Web Designer, HTML, CSS, Website animation with Javascript, Report manager, GitHub manager

Description: My dream is to become a neural IT engineer, someone who wants to fuse our brains with machines.

Valentin MAIA

Role: SQLDatabase, UML Diagram

Description: Since i'm a kid I always wanted to be an engineer. It was always interesting to know more about this incredible field.

Jason GOUVET

Role: Rest/React/Java (mapping)

Description: As long as I can remember, I always loved to play with electronic systems. My goal is to be an embedded systems engineer.

William Travers

Role: Java/JUnit Test/SQL/React

Description: I've always wanted to be an engineer. Since my childhood I'm following the steps of my father.

Le Ciné du COIN