Projet tuteurés

## REMERCIEMENT

PATRICK SABI

BRUNO BELIERE

Table des matières

[REMERCIEMENT 1](#_Toc286049288)

[Java 1](#_Toc286049289)

[Base de données 1](#_Toc286049290)

[Restitution des données 2](#_Toc286049291)

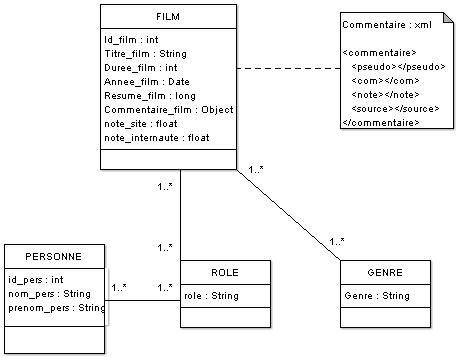
[Problèmes 3](#_Toc286049292)

# Java

# Base de données

Au début de notre projet s’est posé le problème du choix du type de base de données que nous allions utiliser. En effet, il nous est mis à disposition deux bases de données Oracle XMLP1 et XMLP2. Il fallait donc choisir entre une base de données relationnelle qui prévoit l’utilisation de SQL2 ou une base de données objet utilisant du SQL3.

Pour une utilisation de ce genre il s’est révélé plus simple d’utiliser une base de données relationnelle.



# Restitution des données

La restitution s’effectue principalement sous le langage de programmation PHP. Bien sûr on retrouve aussi Du HTML et du CSS permettant la mise en forme des pages de l’interface.

L’interface doit être ergonomique et homogène afin de faciliter la navigation mais aussi sobre tout en respectant un choix de couleur épuré. Pour ce faire, Nous avons établis des règles permettant d’atteindre facilement l’interface recherchée. Nous avons aussi établis des aspects fonctionnels et techniques permettant une navigation améliorée sur l’interface.

POLICE DE CARACTERE : Nous avons choisis la police de caractères « verdana » sans serif où les empattements des caractères sont inexistants et donnent une impression de modernité. Le nombre de police est limité à une seule car trop de diversité donne un aspect lourd à l’interface qui devient désagréable à regarder. Cela porterait atteinte au confort de navigation de l’utilisateur. Aussi l’utilisation d’un nombre de polices de caractères trop élevé sera limitée et nous n’utilisons pas de types de police gras afin de remédier à cet effet.

COULEURS : Une interface optimale ne contient pas plus de trois couleurs. Ainsi, nous nous basons sur deux couleurs principales qui sont le Orange et le Kaki.

LIENS : Afin d’optimiser la souplesse de l’interface nous utilisons au maximum des liens notamment dans la barre de menu qui est le moyen le plus efficace de recherche sur les interfaces.

TAILLE DES ELEMENTS : Les pages seront assez courtes et le nombre d’éléments affichés sont limités. Cela nécessite le développement d’un système de pagination en PHP permettant de répartir les films sur un nombre de pages nécessaires. Aussi, toute liste de film n’affiche que les informations capitales, utile pour sa recherche. En cliquant sur ce dernier, la fiche entière du film sera affichée, permettant ainsi de voir tous les détails relatifs au film, tels que nom d’auteur, année de production, note attribuée, commentaires, etc.…

### NAVIGATION

Page principale (affiche tous les films par défaut).

Menu

Ajouter un film (permet la recherche d’un film grâce à un formulaire via l’application java)

Page recherche simple (fonction Ajax qui fait une recherche approximative d’un film par nom)

Page genres (affichage des genres disponibles dans la base de données Oracle)

Page films (propose un tri par alphabet)

Affichage des films disponibles sur les sites. Permet de choisir le bon résultat.

Page liste film (liste les films disponibles selon les options sélectionnées. C’est un affichage simple)

Affichage détaillé d’un film

MENU : Le menu comporte les principaux liens nécessaires à une recherche rapide de données. En effet, ici on retrouve des liens vers différentes pages tels que la recherche par alphabet ou par genres. Aussi on y retrouve le formulaire de recherche simplifié.

Ici nous avons à faire à un menu de type classique tel qu’on les retrouves souvent. L’avantage est la pertinence et l’efficacité de recherche car l’utilisation d’un menu est universelle pour toute application. L’utilisateur ne sera donc pas perturbé et trouvera rapidement les données de film recherchés.

CHAMP DE RECHERCHE : Il permet une recherche rapide d’un fim donné sous condition de connaitre un minimum d’information sur ce dernier (nom…). C’est une recherche approximative et plusieurs résultats peuvent être trouvés et affichés.

Le formulaire sera accompagné d’un script d’**auto-complétion** en ajax. Démuni d’un bouton de validation de la recherche, pour chaque caractère entré dans le champs, les résultats affichés seront réactualisés. Cette méthode apporte un confort conséquent à l’utilisateur mais peut cependant ralentir le système étant donné que la requête est ré-executée pour chaque nouveau caractère ajouté. Dans notre cas, les requêtes utilisées pour la recherche de données dans Oracle sont simples, n’utilisant que peu de jointure empéchant un ralentissement trop important de la recherche. Aussi, le nombre de film ne dépassant pas les quelques milliers, le parcours de la base Oracle se fera assez rapidement.

PAGINATION :

L’interface de restitution des données doit afficher un nombre de données assez important en l’occurrence d’une liste de films assez nombreux. En effet, ils existent trop de films pour pouvoir les afficher dans une seule page. De ce fait, nous utilisons la solution de pagination pour les films et commentaires dès que le nombre d’éléments donnés est d’ordre trop important. Cette solution consiste en la répartition du nombre de film sur un nombre de pages donnés.

Exemple : (nombre total d’élément à afficher / nombre de films par pages)

=> Résultat arrondi à l’unité supérieure

En php :

$elements\_totaux = 9 ;

$nombre\_par\_page = 2 ;

//ceil() permet d’arrondir à l’unité supérieur ;

//ne pas utiliser round() ;

$nb\_pages = ceil($elements\_totaux/$nombre\_par\_page) ;

//affichage des pages

if ($nbpages > 1 ){ for ($i=1; $i<=$nb\_pages; $i++){ echo $i ;

} }

# Problèmes

Base de données :

IMDB : Le site web IMDB refuse strictement le parcours des pages par un robot grâce à la protection par proxy. Cela pose problème pour notre projet car ce site contient l’une des plus grandes bases de données de films, très complet, fiable mais aussi détenteur de nombreux commentaires et critiques de film qui nous auraient été très utiles. Par conséquent, nous avons décidé de prendre contact avec cette société en expliquant les faits, afin de demander une autorisation de parcours des données. Malheureusement, nous avons reçus une réponse négative. En effet, la charte stipule l’interdiction de parsing / scraping de données. Cependant, une sélection de données peut être téléchargée gratuitement où encore une licence permettant l’accès à la totalité des données peut être acheté. Dans les deux cas, cette solution va à l’encontre de notre projet qui consiste à parcourir des pages spécifiques dans le but d’en extraire les données. Voici les messages envoyés à IMDB et leur réponse vis-à-vis de ce dernier :

