



INF3710 –Fichiers et Bases de données

Hiver 2020

TP No. 4

Groupe 2

1954157 – Ismail Bakkouri

1947497 – Yuhan Li

Soumis à : Amal Zouaq

Le 19 avril 2020

INTRODUCTION

Les bases de données se retrouvent partout de nos jours, que ce soit dans les sites web, le cloud ou les logiciels et applications de notre quotidien. Le travail pratique qui nous a été attribué vise exactement ce sujet. Son objectif est de concrétiser nos connaissances acquises en BD relationnelle dans un contexte de développement Web.

Ainsi, nous avons dû concevoir une application Web permettant d'acheter et télécharger des vidéos en ligne. Notre plateforme, *Netflix_Poly*, offre ses services aux membres (mensuels ou par visionnement), et ces derniers ont la chance de commander des copies physiques de film ou d'en visionner directement en ligne.

Toutes les fonctionnalités et les détails d'utilisation ont dû être considéré dans le développement de *Netflix_Poly*, ce qui veut dire qu'une analyse d'étude de cas s'est fait pointilleusement. Cette dernière s'est faite à travers une modélisation conceptuelle et relationnelle qui nous ont permis de construire la base de données. Nos connaissances en BD relationnelles ont également été mis à l'épreuve dans l'implantation d'une liste de requêtes fournit. Du côté Web, la conception, autant cliente que serveur, s'est effectuée avec Node et Angular. Nous devons faire en sorte d'utiliser les informations de notre base de données pour le côté client. Bien entendu, il a fallu s'assurer de la qualité du code, la modularité du code, et la gestion des exceptions afin d'offrir la meilleur expérience utilisateur possible.

MODÈLE CONCEPTUEL

Hypothèses

1. Il y aura une entité Membre liée à une entité Film. Ils auront une relation many to many (* : *), car nous assumons qu'un même film peut être visionné par plusieurs et qu'un membre peut être en train de visionner plusieurs films. Ainsi, une entité de relation se forme et on peut y stocker la date et la durée de visionnement.
2. Étant donné qu'on a besoin de calculer le prix de livraison d'un DVD physique (une entité), nous pensons le faire avec le code postal du membre. Ainsi, nous avons pensé séparer l'adresse du membre en deux attributs : l'adresse de la rue (numéro et rue) et le code postal.
3. Puisqu'un DVD physique est représenté par un seul film mais qu'un film peut posséder plusieurs DVD physiques, nous pensons qu'il faudrait faire une relation 1 : *. De plus, les données d'un DVD physique sont rentrées seulement si ce dernier représente un film. Cette entité dépend donc de l'entité Film : le id d'un DVD est une clé partielle.
4. Nous assumons qu'une carte de crédit (une entité) n'est posséder que par une seule personne (Membre). Bien entendu, nous pouvons tous posséder plusieurs cartes de crédit. La relation serait donc * : 1.
5. Concernant l'information des participant d'un film, nous pensons qu'il est bien d'avoir deux entités qui le représente : Personne et Rôle. La Personne participe dans un Film et la Personne possède un Rôle, dont les attributs sont salaire et nom du rôle (clé partielle). Rôle est une entité partielle, car il ne peut pas exister s'il n'y a pas une Personne qui l'obtient.
**COMMENTAIRE : Cette hypothèse n'a pas été retenue. Nous avons décidé de ne pas mettre d'entité partielle Rôle, et de mettre ses attributs plutôt dans l'entité de relation former entre Personne et Film. Elle se nomme Participation. Cela est beaucoup plus simple et propre pour l'obtention des informations d'un film.
6. Nous pensons que le lien entre les films nominés/gagnants et les Oscars peut se faire avec deux relations entre l'entité Film et CatégorieOscar. Puisqu'en temps normal plusieurs films sont nominés et qu'un film peut être nominé dans plusieurs catégories, CatégorieOscar nomine Film avec une cardinalité many to many. Pour les films gagnants, un Film peut gagner une ou plusieurs CatégorieOscar, ce qui donnerait une relation 1 : *.
7. Il serait bien de spécifier que CatégorieOscar devrait être une entité de partielle de l'entité CeremonieOscar, puisqu'une catégorie n'existe pas sans cérémonie.

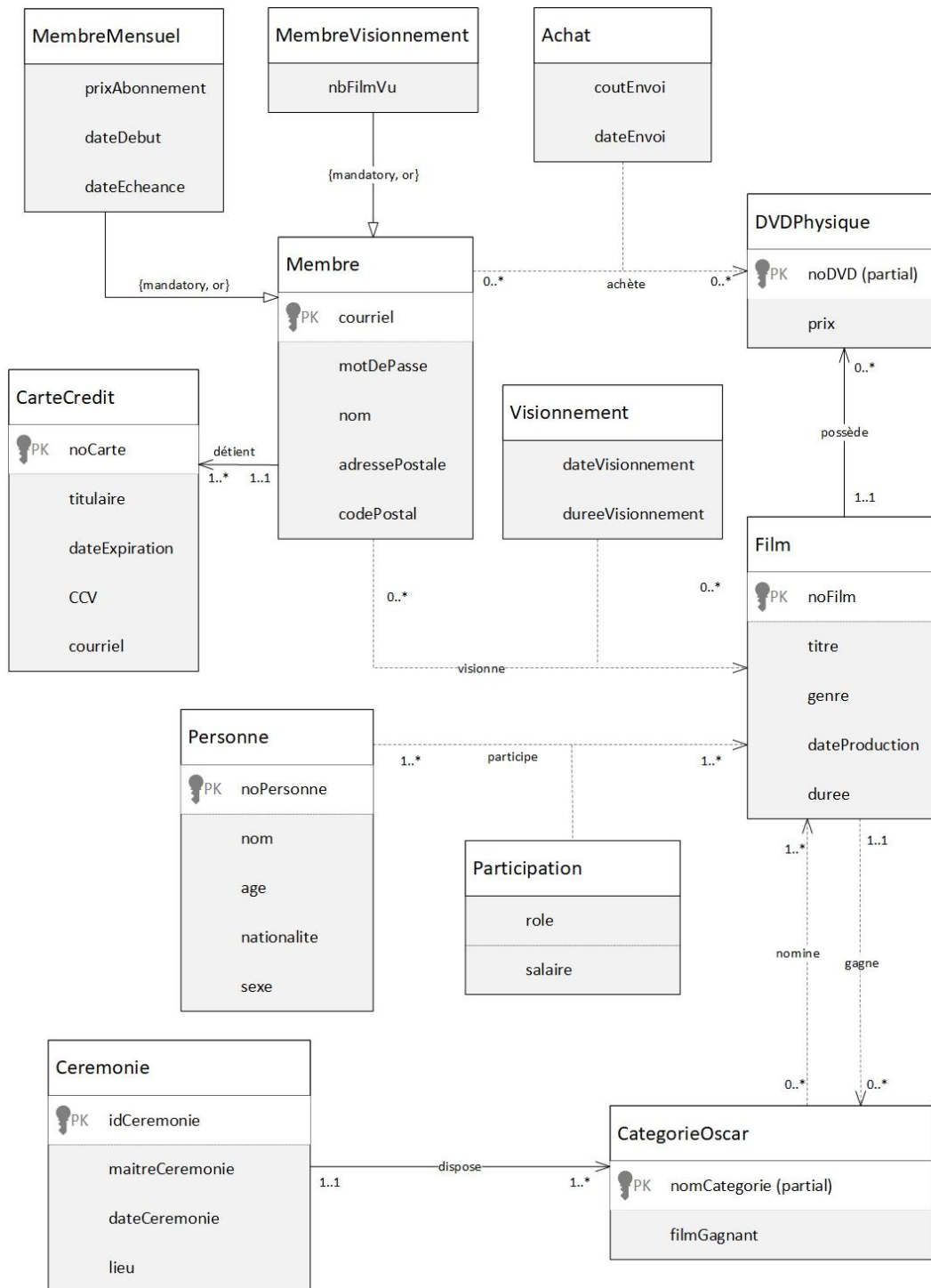


Figure 1. Modèle conceptuel de la plateforme Netflix_Poly

MODÈLE RELATIONNEL

membre(courriel, motDePasse, nom, adressePostale, codePostal)

membreMensuel(courriel, motDePasse, nom, adressePostale, codePostal, prixAbonnement, dateDebut, dateEchéance)

FK courriel REFERENCES Membre(courriel)

membreVisionnement(courriel, motDePasse, nom, adressePostale, codePostal, nbFilms)

FK courriel REFERENCES Membre(courriel)

carteCredit(noCarte, titulaire, dateExpiration, CCV, courriel)

FK courriel REFERENCES Membre(courriel)

film(noFilm, titre, genre, dateProduction, durée)

visionnement(courriel, noFilm, dateVisionnement, duréeVisionnement)

FK noMembre REFERENCES Membre(noMembre)

FK noFilm REFERENCES Film(noFilm)

DVDPhysique(noDVD, noFilm, prix)

FK noFilm REFERENCES Film(noFilm)

achat(courriel, noFilm, noDVD, coutEnvoi, dateEnvoi)

FK courriel REFERENCES Membre(courriel)

FK noFilm REFERENCES Film(noFilm)

FK noDVD REFERENCES DVDPhysique(noDVD)

cérémonie(idCérémonie, maitreCérémonie, dateCeremonie, lieu)

catégorieOscar(nomCatégorie, idCérémonie, filmGagnant)

FK idCérémonie REFERENCES Cérémonie(idCérémonie)

FK filmGagnant REFERENCES Film(noFilm)

nomination(idCérémonie, catégorie, noFilm)

FK idCérémonie REFERENCES Cérémonie(idCérémonie)

FK catégorie REFERENCES CatégorieOscar(nomCatégorie)

FK noFilm REFERENCES Film(noFilm)

personne(noPersonne, nom, dateNaissance, nationalite, sexe)

participation(noPersonne, noFilm, nomRole, salaire)

FK noPersonne REFERENCES Personne(noPersonne)

FK noFilm REFERENCES Film(noFilm)

ALGÈBRE RELATIONNEL

Requête 1

$\text{info_HarryPotter} \leftarrow \sigma_{\text{titre} = \text{'Harry Potter'}}(\text{participation} \bowtie \text{personne} \bowtie \text{film})$

$\Pi_{\text{titre, genre, dateProduction, duree, nomRole, nom}}(\text{info_HarryPotter})$

Requête 2

$\text{filmVisionné} \leftarrow \Pi_{\text{noFilm, dateVisionnement}}(\text{visionnement})$

$\text{filmAchat} \leftarrow \Pi_{\text{noFilm, dateEnvoi}}(\text{achat})$

$\text{film_achatVisio} \leftarrow (p_v(\text{filmVisionné}) \bowtie_{v.\text{noFilm}=a.\text{noFilm}} p_a(\text{filmAchat}))$

$\text{film_achatOuVisionné} \leftarrow (p_f(\text{Film}) \bowtie_{(f.\text{noFilm}=va.v.\text{noFilm}) \vee (f.\text{noFilm}=va.a.\text{noFilm})} p_{va}(\text{film_achatVisio}))$

$\Pi_{\text{titre, genre, derniereDate}}(\text{film_achatOuVisionné})$

Requête 5

$\text{achat_membre} \leftarrow \pi_{m.\text{nom}}(p_m(\text{membre}) \bowtie p_a(\text{achat}) \bowtie p_d(\text{dvdPhysique}))$

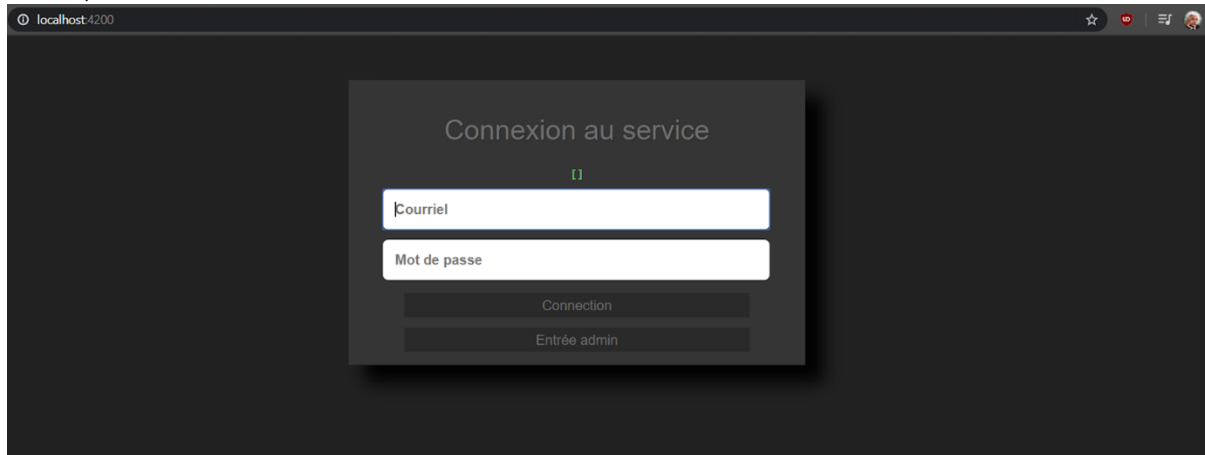
$\text{cout_total} \leftarrow \pi_{m.\text{nom}} \sigma_{\text{SUM}(a.\text{prix})}(\text{achat_membre})$

$\text{moyenne} \leftarrow \pi_{m.\text{nom}} \sigma_{\text{AVG}(\text{cout_total})}(\text{achat_membre})$

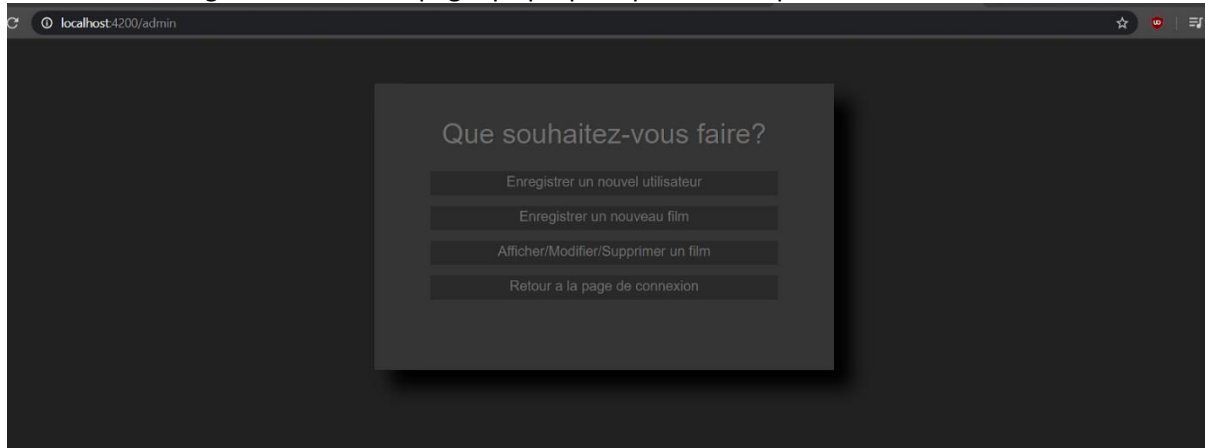
$\Pi_{m.\text{nom}} (\sigma_{\text{cout_total} > \text{moyenne}}(\text{achat_membre}))$

Présentation de l'application développée:

Quand on lance la partie client, la page d'accueil qui correspond à une demande d'authentification est lancée, elle ressemble à ceci:



Pour pouvoir accéder à l'interface de visionnement des films, il faudra se connecter. Bien sûr, un message d'erreur surviendra dans le champ en vert si la connexion échoue ou bien si l'utilisateur n'a pas de compte. Si vous voulez créer un compte, il est possible de le faire en cliquant sur le bouton "entrée admin". Vous serez alors redirigé vers une autre page qui propose plusieurs options d'admin:



Pour créer un compte il faut choisir la première option qui redirige vers la page suivante:

Enregistrer nouvel utilisateur

Nom

Prénom

Courriel

Mot de passe

Adresse

Code postal (A1B 2C4)

Membre Mensuel

Membre par vue

Prix abonnement

Date d'échéance (YYYY-MM-DD)

Enregistrer l'utilisateur

Retour au choix des options

Page de connexion

Il faut ensuite remplir les informations demandées, qui sont indiquées clairement dans les boîtes. Si l'utilisateur ne remplit pas un champ ou se trompe dans le format des données, un message d'erreur lui est affiché. Par exemple si on omet le courriel ou si son format est invalide:

Enregistrer nouvel utilisateur

[Veuillez Entrer une adresse mail valide]

nom du membre

Prénom du membre

@courrielInvalide

1500 chemin de la tour montréal

A2B 3F5

Enregistrer nouvel utilisateur

[Veuillez Entrer une adresse mail valide]

nom du membre

Prénom du membre

Courriel

1500 chemin de la tour montréal

Quand le serveur réussit l'enregistrement, un message de confirmation est affiché (image de gauche ci-dessous). Par contre, si on réessaie d'insérer un membre avec le même courriel, le serveur renvoie une erreur à l'écran (image de droite), car deux membres ne peuvent pas avoir la même adresse courriel qui correspond à la PK des membres (loi d'unicité d'une clé primaire) :

Enregistrer nouvel utilisateur

[Enregistrement Réussi.]

Enregistrer nouvel utilisateur

[Enregistrement impossible. Un utilisateur avec ce courriel exist déjà.]

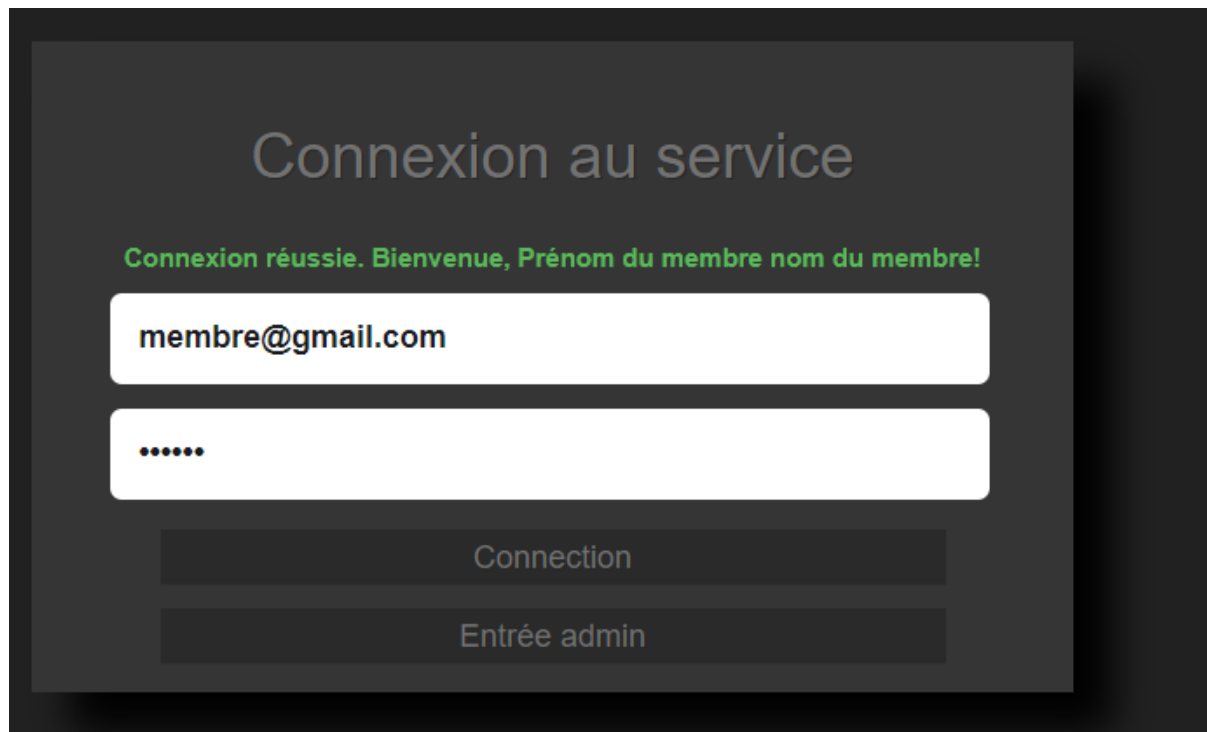
Une fois l'ajout validé par le serveur, il sera possible d'afficher le membre directement sur PGAdmin:

Data Output							Explain	Messages	Notifications
	<div><div><div></div></div><div><div>courriel</div><div>[PK] character varying (30)</div></div></div>	<div><div><div></div></div><div><div>motdepasse</div><div>character varying (200)</div></div></div>	<div><div><div></div></div><div><div>nom</div><div>character varying (30)</div></div></div>	<div><div><div></div></div><div><div>adressepostale</div><div>character varying (40)</div></div></div>	<div><div><div></div></div><div><div>codepostal</div><div>character varying (7)</div></div></div>	<div><div><div></div></div><div><div>nbfilms</div><div>integer</div></div></div>			
1	ismail@hotmail.com	mdp12345	Bakkouri	6590 34e Avenue	H1T 3B4	10			
2	123god@hotmail.com	12345mdp	ashashahan	216 Kindersley Ave	H3R 1R6	1			
3	ming@hotmail.com	1234567mdp	xiao Yuhan	216 Kindersley Ave	H3R 1R6	5			
4	andie@hotmail.com	12345678mdp	pogdorica	216 Kindersley Ave	H3R 1R6	9			
5	victor@hotmail.com	123456789mdp	kim	216 Kindersley Ave	H3R 1R6	2			
6	membre@gmail.com	d380168c4904d4fd73ba5579...	Prénom du membre nom du membre	1500 chemin de la tour mont...	A2B 3F5	0			

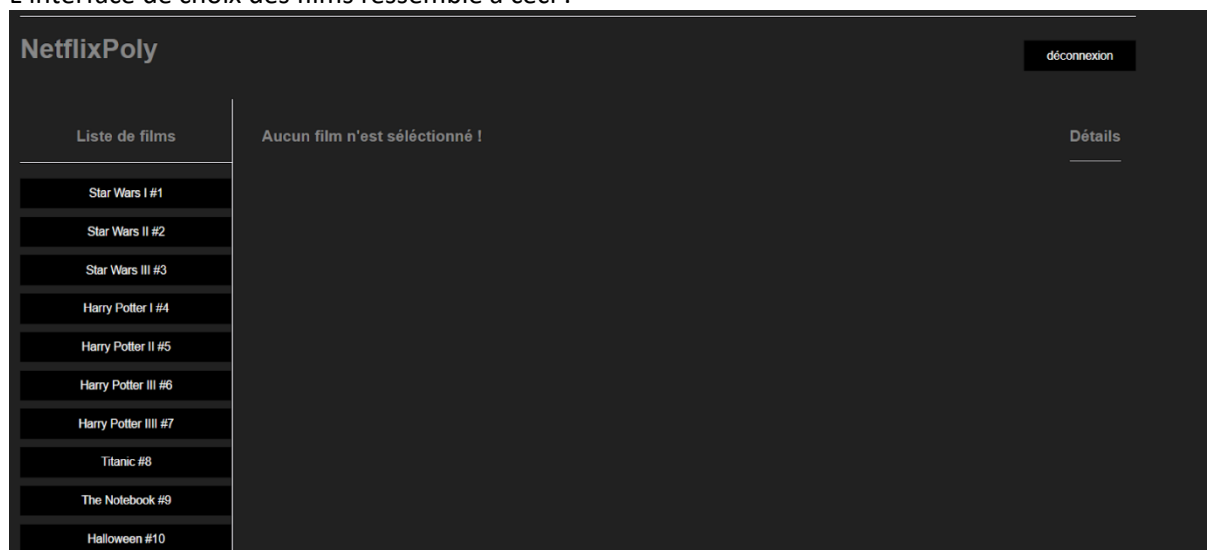
Les mots de passe des 5 premiers membres ne sont pas hachés puisque nous les avons ajoutés manuellement pour les requêtes du TP. Le membre ajouté à partir de l'application a son mot de passe haché.

Une fois l'enregistrement terminé, nous pouvons retourner à la page de connexion et nous connecter à la plateforme Poly_Netflix. Si la connexion réussit, un message de bienvenue est affiché à l'utilisateur avant d'être redirigé vers la page de visionnement des films :

Si la connexion réussit un message de bienvenue est affichée à l'utilisateur avant d'être redirigé vers la page de visionnement de films:



L'interface de choix des films ressemble à ceci :



Nous avons tenté de rendre cette page le plus simple et structuré possible, afin de faciliter l'expérience de l'utilisateur. Par exemple, l'interface n'est pas surchargée de texte et chaque mot va directement sur le point. De plus, nous avons séparé la page en trois "colonnes", soit respectivement de gauche à droite: la liste des films, le film sélectionné, et ses détails. Cette organisation suit une logique que nous sommes tous habitués depuis notre enfance, soit celle de lire de gauche à droite. Selon nous, tous ces points forment un excellent guidage (critère d'ergonomie de Bastien et Scapin) qui oriente bien l'utilisateur et lui donne une utilisation agréable.

Une fois qu'un membre clique sur un des films de la liste, les détails de ce dernier sont affichés à droite et un lecteur vidéo s'affiche au milieu, comme on peut le voir sur l'image ci-dessous :

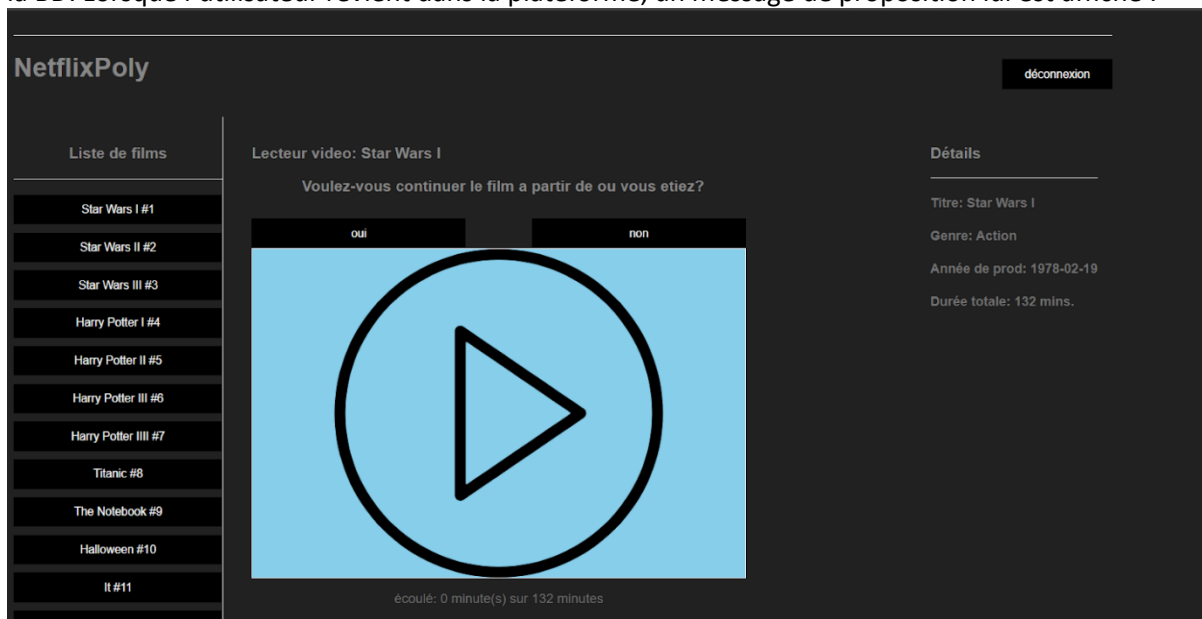


L'utilisateur peut maintenant commencer à visionner le film qu'il a choisi. Un timer de la durée démarre dès qu'il clique sur le bouton play.

Note: pour faciliter les tests de l'application, les minutes se déroulent comme des secondes c'est à dire que si l'utilisateur allume le lecteur pendant 10 secondes, 10 minutes de lecture seront enregistrées. Le temps de lecture est juste en bas du lecteur vidéo, comme l'indique la flèche bleue sur l'image ci-dessous:



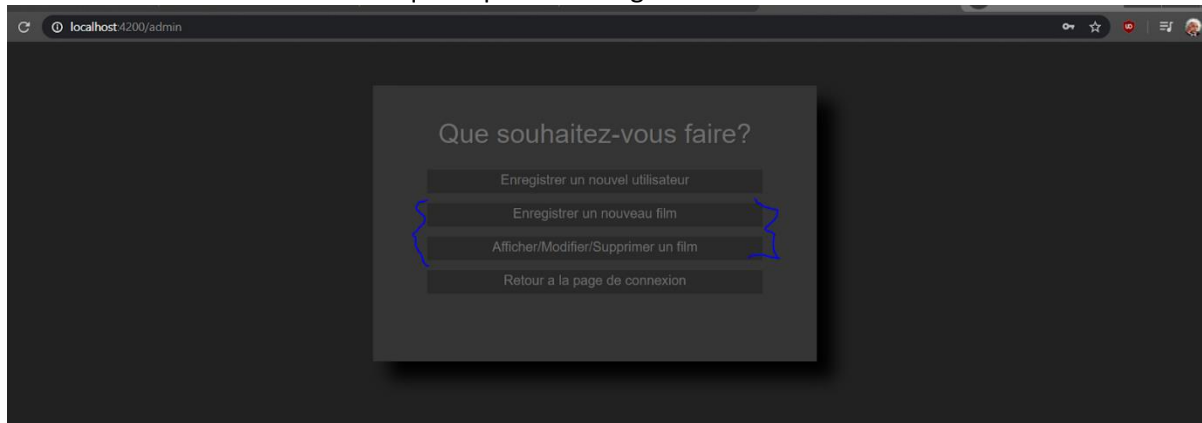
Si l'utilisateur change de film ou se déconnecte, le serveur sauvegarde automatiquement le temps dans la BD. Lorsque l'utilisateur revient dans la plateforme, un message de proposition lui est affiché :



S'il clique sur oui, le temps repart de là où il était avant de changer de film ou de se déconnecter, sinon le temps repart de 0. L'utilisateur peut sauvegarder autant de temps qu'il le souhaite en BD. Une fois la lecture terminée, il peut cliquer sur le bouton déconnexion en haut à droite pour quitter l'interface de visionnement et terminer sa session.

Concernant les autres options admin, nous avons:

Enregistrer un nouveau film et afficher/modifier/supprimer un film. les options sont accessibles dans l'entrée admin au même endroit que l'option d'enregistrement d'un membre:

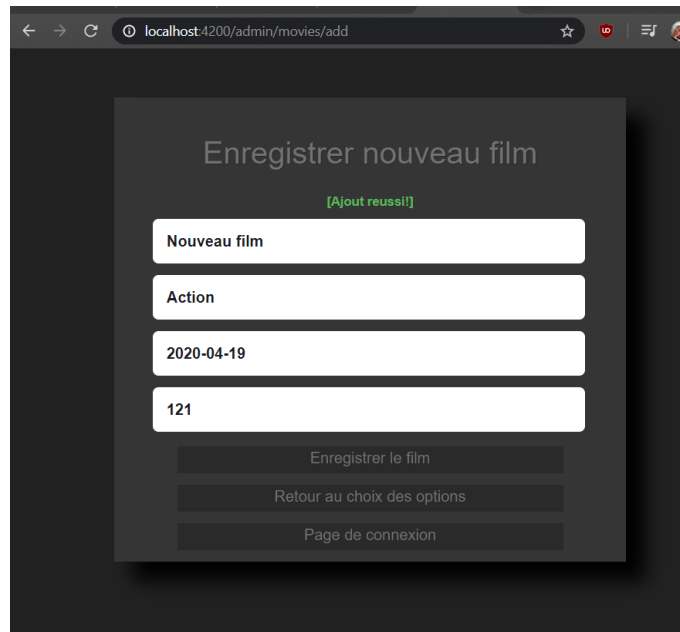


Si le membre souhaite enregistrer un film il clique sur l'option correspondante, il est ensuite dirigé vers la page suivante:



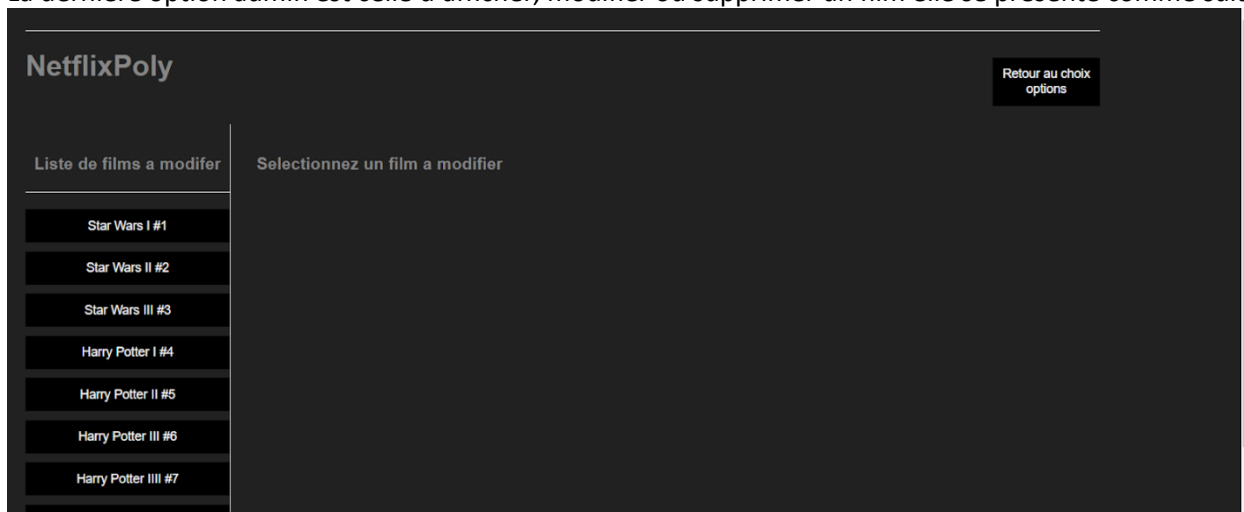
The screenshot shows a web browser window with the address bar displaying 'localhost:4200/admin/movies/add'. The main content area features a dark gray modal box titled 'Enregistrer nouveau film'. Inside the modal, there is a green status indicator '[]'. Below this, there are four white input fields with the following labels: 'Titre', 'Genre', 'Date de production (aaaa-mm-jj)', and 'Duree (min)'. At the bottom of the modal, there are three buttons: 'Enregistrer le film', 'Retour au choix des options', and 'Page de connexion'.

Tout comme avec l'enregistrement d'un membre, il doit entrer des données valides. Par exemple une date de abc123 ne passera pas. Une fois l'ajout du film réussit, le serveur envoie un message de confirmation: Une fois l'ajout fini l'admin peut retourner au choix d'options admin ou à la page de connexion.



The screenshot shows the same web browser window, but the modal box now displays a green status indicator '[Ajout réussi!]'. The input fields are filled with the following values: 'Nouveau film' for 'Titre', 'Action' for 'Genre', '2020-04-19' for 'Date de production', and '121' for 'Duree'. The buttons at the bottom remain the same: 'Enregistrer le film', 'Retour au choix des options', and 'Page de connexion'.

La dernière option admin est celle d'afficher, modifier ou supprimer un film elle se présente comme suit:



Nous retrouvons notamment le film que nous venons d'ajouter à la toute fin de la liste:



Quand l'admin clique sur un film une fenêtre contenant des champs éditables et des boutons s'affichent, il peut alors modifier le titre, le genre, la date de production ou encore la durée du film en minutes ou tout simplement supprimer le film de la base de données:



En modifiant le titre: le serveur retourne une confirmation affichée à l'écran:

NetflixBoly

Retour au choix options

Liste de films a modifier

Star Wars I #1

Star Wars II #2

Star Wars III #3

Harry Potter I #4

Harry Potter II #5

Harry Potter III #6

Harry Potter IIII #7

Titanic #8

The Notebook #9

Halloween #10

Modifier les données de Nouveau titre de nouveau film

[Mise a jour du film réussie]

Nouveau titre de nouveau film

Action

2020-04-19

121

Modifier

Supprimer

De même pour la suppression:

Liste de films a modifier

Star Wars I #1

Star Wars II #2

Star Wars III #3

Harry Potter I #4

Harry Potter II #5

Harry Potter III #6

Harry Potter IIII #7

Titanic #8

The Notebook #9

Halloween #10

Modifier les données de Nouveau titre de nouveau film

[Suppression reussie!]

Nouveau titre de nouveau film

Action

2020-04-19

121

Modifier

Supprimer

Le film disparaît ensuite de la liste de films à gauche et la fenêtre d'édition se ferme après 500 millisecondes.

Les nouvelles données stockées en BD peuvent ensuite être affichées par les membres qui vont se logger au système, ou directement sur PGadmin.

Guide d'installation et de configuration :

1. Premièrement, ouvrez le terminal, allez dans le dossier serveur, faites un `npm ci` (fortement recommandé) ou `npm install` (pas recommandé).
2. Ensuite, allez sur le fichier `DBservice.ts` présent dans le répertoire `webApp/server/app/services`.
3. Vous allez voir un commentaire `// À configurer`. Il s'agit de vos propres options de connexion à `pgAdmin`:

```
export class DatabaseService {  
  
  //A configurer  
  public connectionConfig: pg.ConnectionConfig = {  
    user: 'postgres',  
    database: 'netflix',  
    password: '123456',  
    port: 5432,  
    host: '127.0.0.1',  
    keepAlive: true,  
  };  
}
```

4. Renseignez votre nom d'utilisateur dans le champ `user`, le nom de votre BD dans le champ `database`, ainsi que votre mot de passe pour `pgadmin` dans le champ `password`. Sauvegardez vos modifications.
5. Faites ensuite un `npm start` dans le terminal sur le répertoire du serveur.
6. Si tout s'est bien passé, il devrait vous écrire "listening on port 3000". Sinon, vous indiquerez une erreur de `username/mot de passe incorrect` ou de `BD inexistante`.

```
> server@1.1.0 start C:\Users\Ismail\Desktop\TP4_INF3710\webApp\server  
> nodemon  
  
[nodemon] 2.0.2  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching dir(s): app\**\*  
[nodemon] watching extensions: ts  
[nodemon] starting `ts-node ./app/www.ts`  
Listening on port 3000  
□
```

7. Allez ensuite sur le répertoire du client et exécutez encore, de même que le serveur, un `npm ci` (fortement recommandé) ou `npm install` (pas recommandé).
8. Quand les modules ont fini de se télécharger, exécutez la commande `npm start` qui devrait compiler les fichiers et ouvrir une fenêtre avec l'application. S'il n'y a pas d'erreur, la console devrait afficher quelque chose de similaire :

```
chunk {main} main.js, main.js.map (main) 147 kB [initial] [rendered]  
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 264 kB [initial] [rendered]  
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]  
chunk {styles} styles.js, styles.js.map (styles) 10.5 kB [initial] [rendered]  
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.15 MB [initial] [rendered]  
Date: 2020-04-19T16:40:41.145Z - Hash: fc8c539cbb5cfbbc5442 - Time: 12628ms  
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **  
i [wdm]: Compiled successfully.
```

9. Vous pouvez commencer à utiliser l'application.