

## **Sujet n°3 : « GESTION D'UNE VIDEO ET DVD THEQUE »**

### **PopCornTheque**

#### ***Groupe n°1 :***

- ◆ STAINE Florian
- ◆ SENE Victor
- ◆ TREHIOU Paul
- ◆ OBRY William

***UTBM Tronc Commun  
SI20 SEMESTRE A14***



## Table des matières

1. Historique.....	3
2. Présentation Projet.....	3
3. Cadre.....	3
4. Extension possible.....	3
5. Conception du Projet.....	4
5.1.Modèle de dictionnaire.....	4
5.2.Le modèle entité association.....	1
5.2.1.Sous-Modèle commentaires.....	1
5.2.2.Sous-Modèle Film.....	0
5.2.3.Sous-Modèle Emprunt.....	1
5.2.4.Sous-Modèle Social.....	0
5.3.La modélisation logique.....	1
5.3.1.Sous-Modèle commentaire.....	1
5.3.2.Sous-Modèle film.....	1
5.3.3.Sous-Modèle emprunts.....	0
5.3.4.Sous-Modèle social.....	0
6. La modélisation de l'IHM.....	1
6.1.Imaginons la page Principale & la présentation d'un film :.....	1
6.2.Page description d'un film :.....	2
6.3.Page d'accueil du site :.....	3
6.4.Page d'inscription au service :.....	3
6.5.Page principale après connexion.....	4
6.6.Page d'ajout d'un ami.....	4
6.7.Page mon espace :.....	5
7. Requête SQL.....	6
7.1.Ajouter un nouveau film.....	6
7.2.Ajouter un nouvel utilisateur.....	6
7.3.Ajouter un support possédé.....	6
7.4.Demande d'emprunt d'un support.....	6
7.5.Emprunt d'un support.....	6
7.6.Ajout d'un commentaire sur un film.....	6
7.7.Ajout d'un commentaire sur un utilisateur.....	6
7.8.Ajouter un utilisateur en amis.....	6
7.9.Afficher la liste de film disponible.....	6
7.10.Liste de film d'un utilisateur.....	7
7.11.Ajouter un support.....	7
7.12.Connexion au site.....	7
7.13.Afficher les détails d'un film.....	7
7.14.Filtrage des films disponibles.....	7
7.15.Afficher informations utilisateur.....	7
7.16.Affichage de tous les emprunts en cours.....	8
7.17.Affichage de tous les prêts en cours.....	8
7.18.Affichage de toute les demandes d'emprunt envoyées.....	8
7.19.Affichage de toutes les demandes d'emprunt reçues pouvant être validées.....	8
7.20.Associer l'ID du support dans demande_emprunt à un nom de film.....	8
7.21.Marquer un support emprunté comme rendu.....	8
8. Description API.....	9
9. Explication de certaines sources PHP.....	9
9.1.Les sessions - la connexion.....	9
10. Les sessions - l'identification sur les autres pages.....	10
11. Les sessions - la déconnexion.....	10

## 1. Historique

Date	Auteur	Description
15 Octobre	Victor SENE	Création du document
12 Novembre 2014	Victor SENE	Cadrage & Description Projet
19 Décembre 2014	William OBRY	Partage & Mise en forme
29 Décembre 2014	William OBRY	Ajout commande SQL & Screenshots
4 Janvier 2015	Victor SENE	Finalisation
7 Janvier 2015	Groupe entier	Ajout dernières requêtes et maquettes

## 2. Présentation Projet

A quoi sert notre application ?

Nous avons pour but de créer une application qui va permettre à chacun de gérer sa vidéothèque personnelle et permettre l'échange de différent support entre utilisateurs. Une fonction social est intégré à l'application avec une liste d'amis pour chaque utilisateurs est la possibilité de créer une liste de film que l'on veut emprunter.

Le projet se présentera sous la forme d'un site web permettant à l'utilisateur d'utiliser l'application sur n'importe quel support disposant d'un accès internet et sans installation préalable.

## 3. Cadre

- Description des films ou vidéos (acteurs, genre du film, ...)
- Description du support physique permettant de stocker un film
- Demandes de location des supports à des amis ou des personnes souhaitant louer le support avec les films
- Gestion des retours des locations
- L'application va permettre de faire une recherche de film
- Ajout des supports que l'on possède
- Ajouter des ami(e)s

## 4. Extension possible

- Il sera possible d'utiliser l'application comme un moyen de faire une watchlist de films
- L'application pourrait signaler à l'utilisateur la sortie du film au cinéma
- Afficher les séances en rapport avec les films qui intéresse l'utilisateur.
- Nous voudrions rendre l'application disponible sur un mobile pour plus de facilité lors de l'utilisation sur de petit écran.

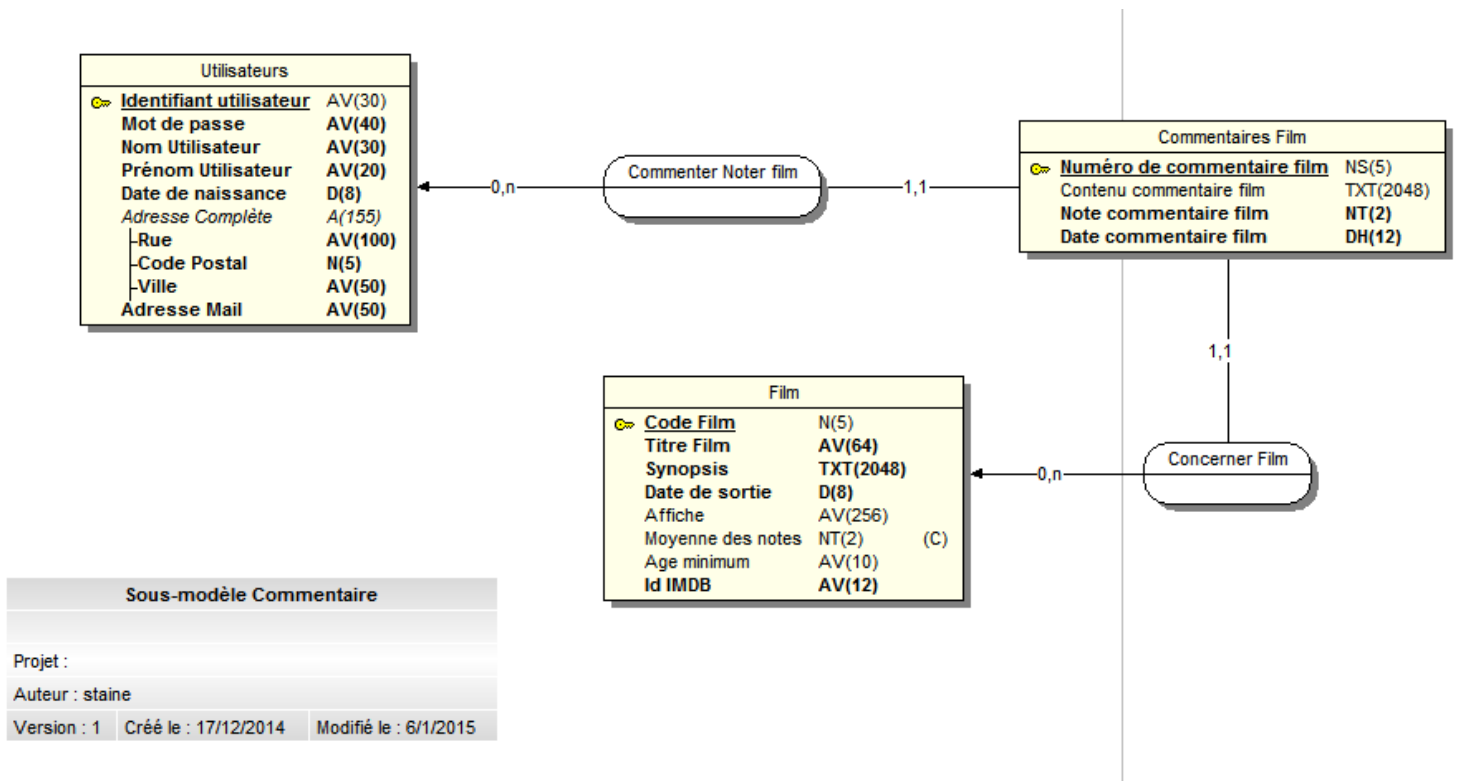
## 5. Conception du Projet

### 5.1. Modèle de dictionnaire

Nom Conceptuel	Nom Logique	Type	Nature	Longueur	Obligatoire	Identifiant	Type Win'Design	Remarque
Adresse complète	UTI_ADRESSE_COMPLETE	CA	A	155	O		A(155)	
Adresse mail	UTI_MAIL	E	AV	50	O		AV(50)	
Affiche	FILM_AFFICHE	E	AV	256			AV(256)	Lien vers une image
Age minimum	FILM_AGE_MIN	E	AV	10			AV(10)	
Code artiste	ART_ID	E	N	5	O	O	N(5)	Générée automatiquement
Code film	FILM_ID	E	N	5	O	O	N(5)	Générée automatiquement
Code postal	UTI_CODE_POSTAL	E	N	5	O		N(5)	
Contenu commentaire film	COMF_CONTENU	E	TXT	2048			TXT(2048)	
Contenu commentaire utilisateur	COMU_CONTENU	E	TXT	2048			TXT(2048)	
Date commentaire	COMF_DATE	E	DH	12	O		DH(12)	
Date commentaire utilisateur	COMU_DATE	E	DH	12	O		DH(12)	
Date de naissance	UTI_DATE_NAISSANCE	E	D	8	O		D(8)	
Date de sortie	FILM_DATE_SORTIE	E	D	8	O		D(8)	
Date emprunt	EMPR_DATE	E	D	8	O		D(8)	
Date retour réel	EMPR_RETOUTR_REEL	E	DH	12			DH(12)	
Date retour théorique	EMPR_RETOUTR_THEORIQUE	E	D	8	O		D(8)	
Durée emprunt	EMPR_DUREE	CA	NS	2			NS(2)	
Est rendu	EMPR_RENDU	E	BOO	1	O		BOO(1)	
Id IMDB	FILM_IMDB_ID	E	AV	12	O		AV(12)	
Identifiant emprunt	EMPR_ID	E	NS	5	O	O	NS(5)	Générée automatiquement
Identifiant support	SUP_ID	E	NS	10	O	O	NS(10)	Chaque support est unique, il définit un support sur lequel est contenu un film
Identifiant utilisateur	UTI_ID	E	AV	30	O	O	AV(30)	Générée automatiquement
Libre	SUP_LIBRE	E	BOO	1	O		BOO(1)	
Mot de passe	UTI_MOT_DE_PASSE	E	AV	40	O		AV(40)	
Moyenne des notes	FILM_NOTE	CA	NT	2			NT(2)	Calculée à partir des notes commentaires
Nom artiste	ART_NOM	E	A	30	O		A(30)	
Nom support	SUP_NOM	E	AV	50	O		AV(50)	
Nom utilisateur	UTI_NOM	E	AV	30	O		AV(30)	
Note commentaire film	COMF_NOTE	E	NT	2	O		NT(2)	
Note commentaire utilisateur	COMU_NOTE	E	NT	2	O		NT(2)	
Numéro de commentaire film	COMF_ID	E	NS	5	O	O	NS(5)	Générée automatiquement
Numéro de commentaire utilisateur	COMU_ID	E	NS	5	O	O	NS(5)	Générée automatiquement
Prénom artiste	ART_PRENOM	E	A	20	O		A(20)	
Prénom utilisateur	UTI_PRENOM	E	AV	20	O		AV(20)	
Rue	UTI_RUE	E	AV	100	O		AV(100)	
Synopsis	FILM_SYNOPSIS	E	TXT	2048	O		TXT(2048)	
Titre film	FILM_TITRE	E	AV	64	O		AV(64)	
Ville	UTI_VILLE	E	AV	50	O		AV(50)	

## 5.2. Le modèle entité association

### 5.2.1. Sous-Modèle commentaires



#### **Commenter et noter film**

Un utilisateur effectue zéro, un ou plusieurs commentaires sur un film.

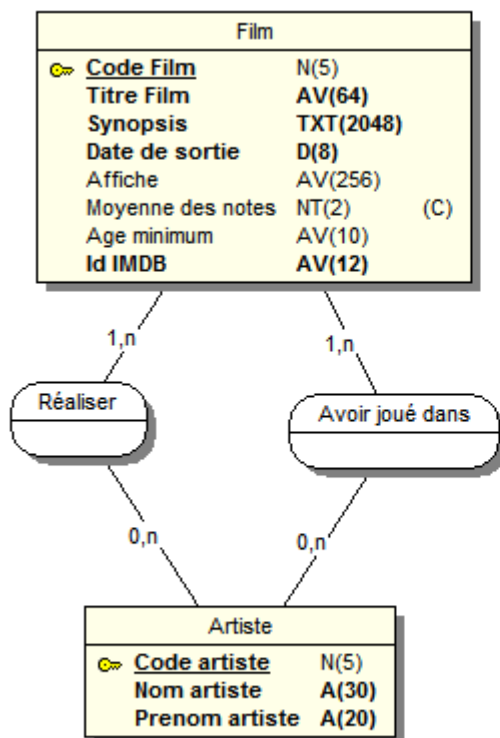
Un commentaire de film est effectué par un et un seul utilisateur.

#### **Concerner film**

Un film est commenté par zéro, un ou plusieurs commentaires de film.

Un commentaire de film concerne un et un seul film.

### 5.2.2. Sous-Modèle Film



Sous modèle Film		
Projet :		
Auteur : UTBrOM		
Version : 1	Créé le : 17/12/2014	Modifié le : 6/1/2015

#### **Réaliser**

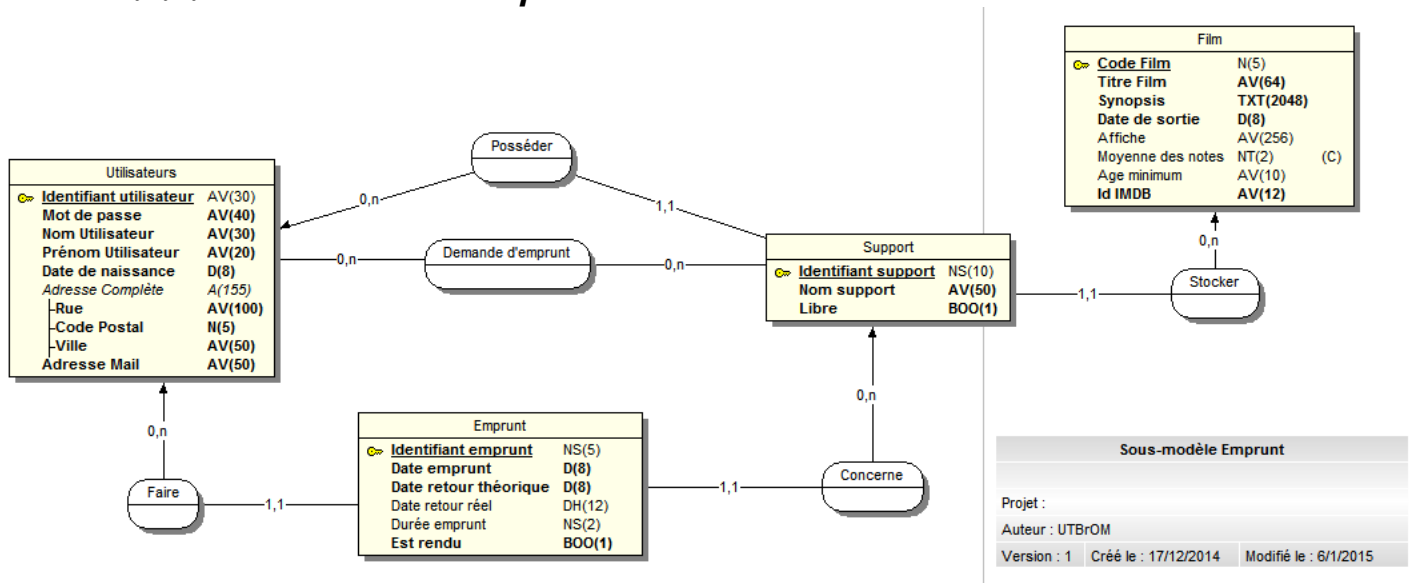
Un artiste peut avoir réalisé zéro, un ou plusieurs films. Un film est réalisé par au moins un réalisateur mais il peut y en avoir plusieurs.

#### **Avoir joué dans**

Un artiste a joué dans zéro, un ou plusieurs films.

Un film possède un ou plusieurs acteurs.

### 5.2.3. Sous-Modèle Emprunt



#### Faire

Un utilisateur fait zéro, un ou plusieurs emprunts.

Un emprunt est réalisé par un et un seul utilisateur.

#### Posséder

Un utilisateur possède zéro, un ou plusieurs supports.

Un support est possédé par un et un seul support.

#### Demande d'emprunt

Un utilisateur effectue zéro, une ou plusieurs demandes d'emprunt.

Un support est concerné par zéro, une ou plusieurs demandes d'emprunt.

#### Concerner

Un emprunt concerne un et un seul support

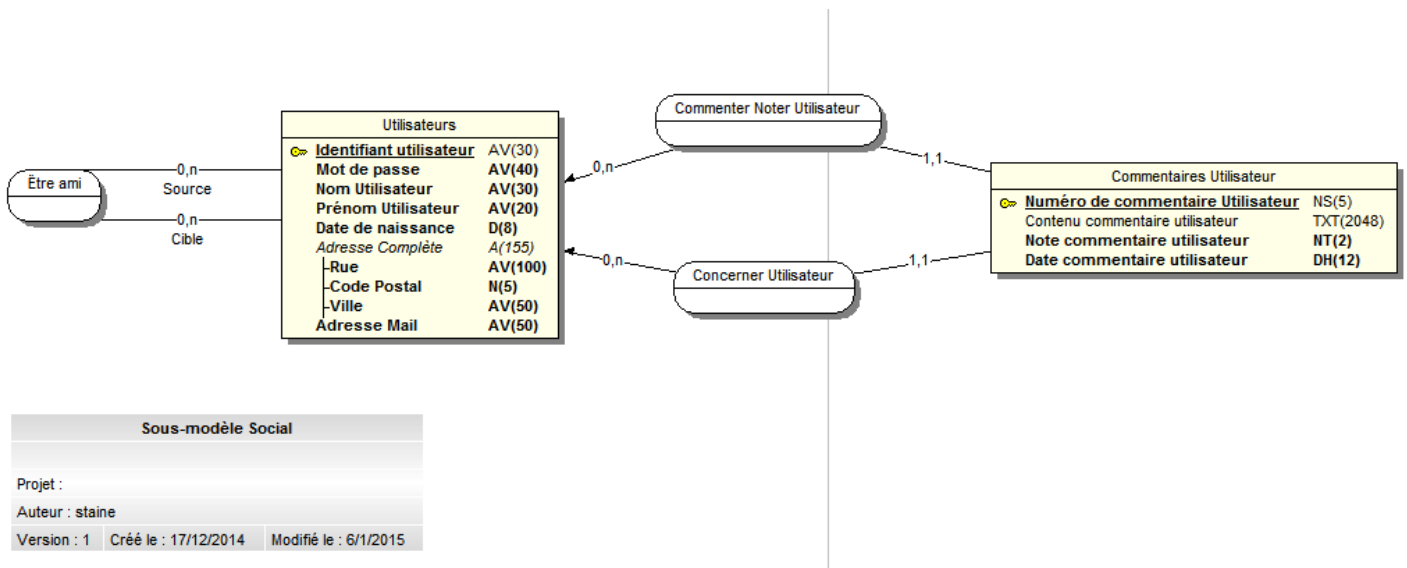
Un support est concerné par zéro, un ou plusieurs emprunts.

#### Stocker

Un support stocke un et un seul film

Un film est stocké sur zéro, un ou plusieurs supports.

## 5.2.4. Sous-Modèle Social



### ***Être ami***

Un utilisateur est ami avec zéro, un ou plusieurs utilisateurs.

### ***Commenter Noter Utilisateur***

Un utilisateur rédige zéro, un ou plusieurs commentaires utilisateur.

Un commentaire utilisateur est rédigé par un et un seul utilisateur.

### ***Concerner Utilisateur***

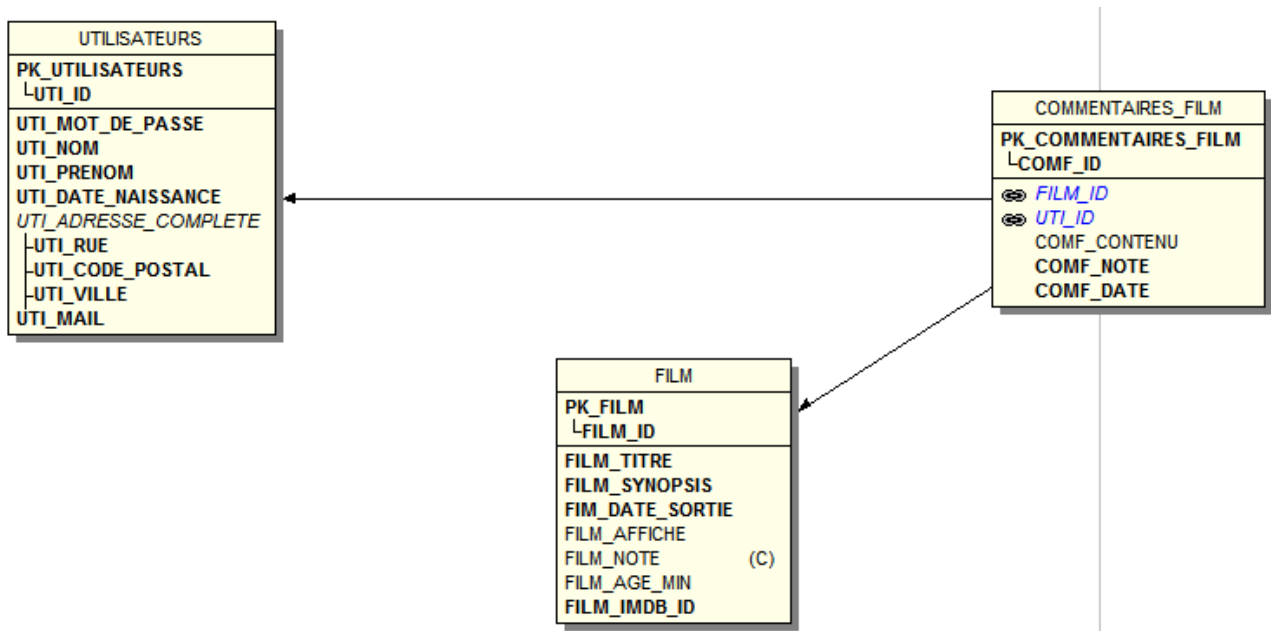
Un utilisateur est concerné par zéro, un ou plusieurs commentaire utilisateur.

Un commentaire utilisateur concerne un et un seul utilisateur.

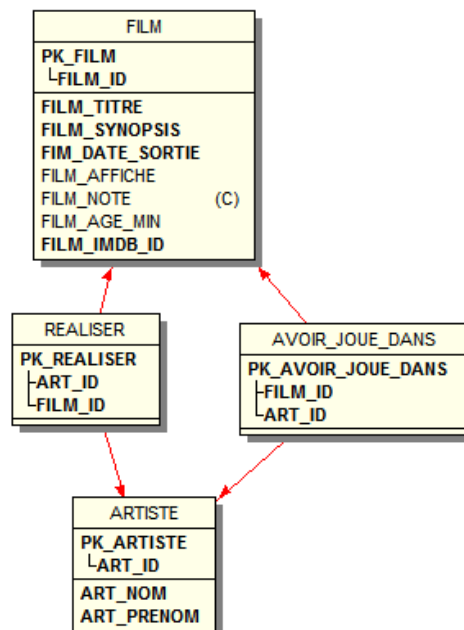


### 5.3. La modélisation logique

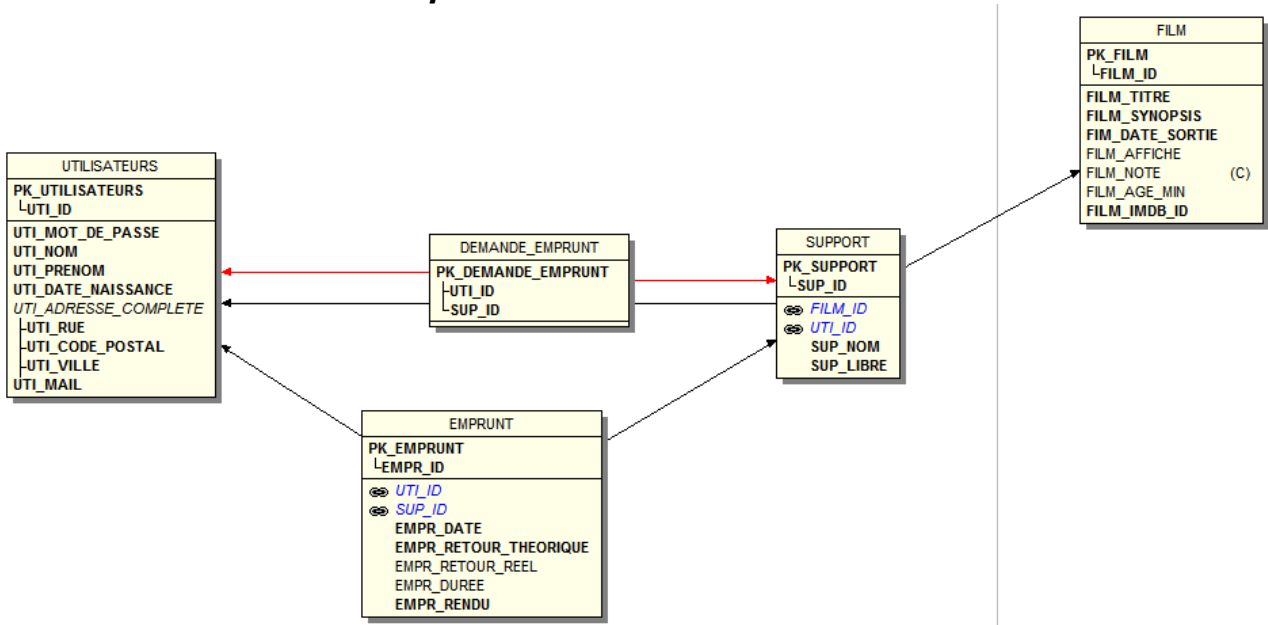
#### 5.3.1. Sous-Modèle commentaire



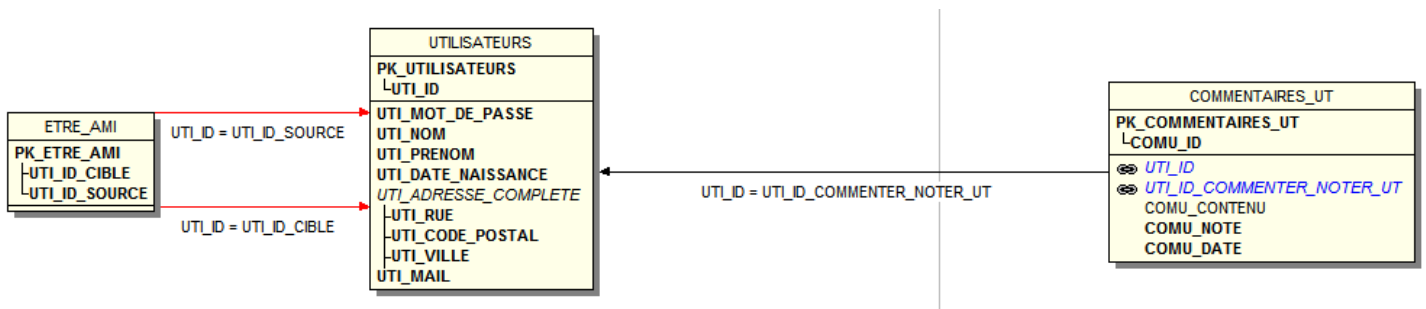
#### 5.3.2. Sous-Modèle film



### 5.3.3. Sous-Modèle emprunts



### 5.3.4. Sous-Modèle social

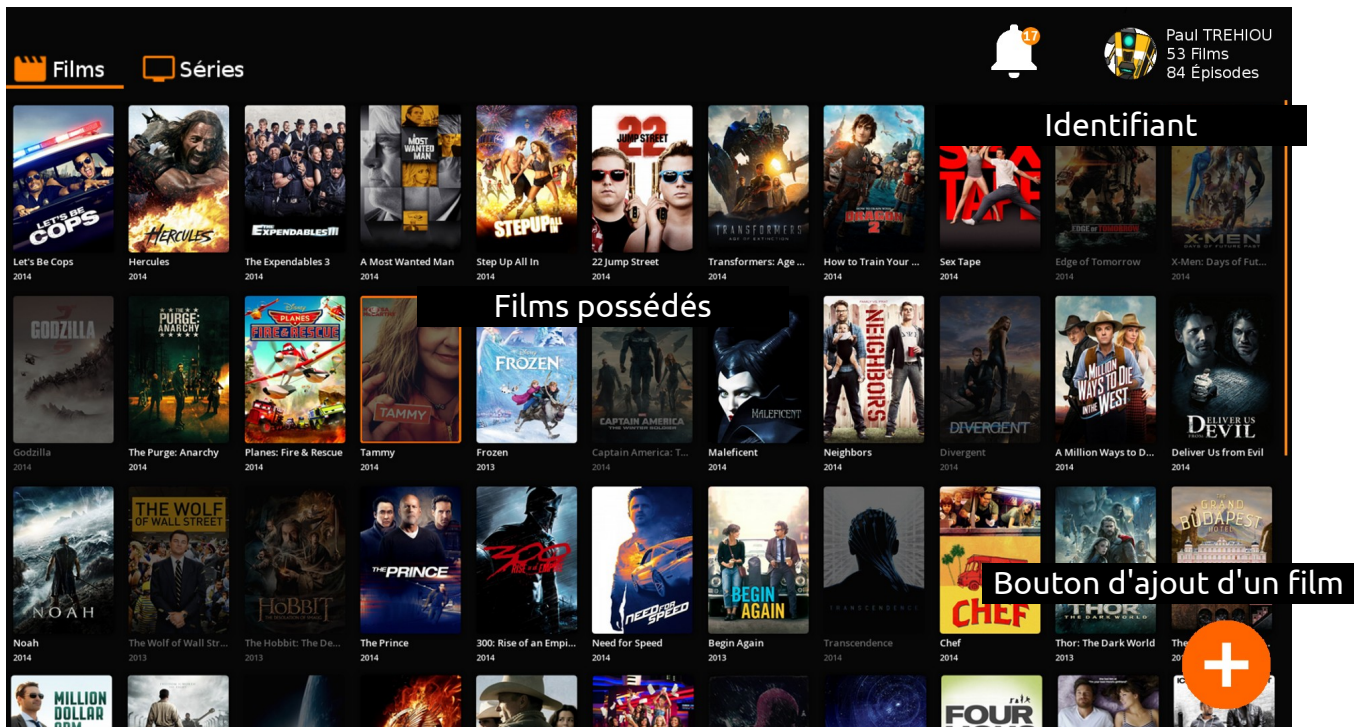


## 6. La modélisation de l'IHM

Vous pouvez visiter directement le site [ici](#).

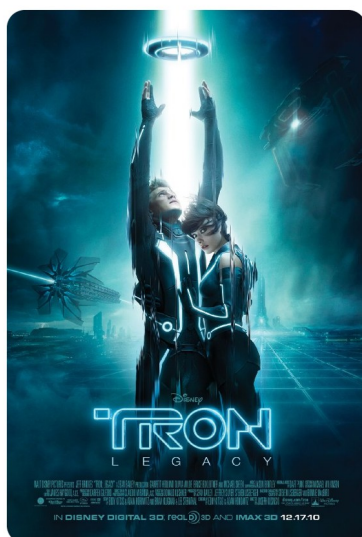
Voici quelques screenshots de la maquette html ainsi que des fichiers images avec l'allure imaginée pour la vidéothèque.

### 6.1. Imaginons la page Principale & la présentation d'un film :



Cette maquette fût imaginée pour donner une idée du travail à faire en CSS. Cependant notre site reste encore aujourd'hui assez austère mais fonctionnel. Ce sont des améliorations à poursuivre.

Ci-dessous la trame pour la description d'un film :



## Tron : L'héritage

Propriétaire	Victor SENE
À rendre dans	5 jours
Support	DVD
Genre	Action, Aventure, Science-Fiction
Disponible à l'origine	2010-12-16
Durée	2:05:08
Acteurs	Garrett Hedlund, Jeff Bridges, Olivia Wilde, Bruce Boxleitner, Yaya DaCosta, Serinda Swan, Beau Garrett, Elizabeth Mathis, James Frain, Amy Esterle, Brandon Jay McLaren, Michael Sheen, Owen Best, Michael Teigen, Daft Punk, Ron Selmour, Conrad Coates, Kis Yurij
Scénariste	Adam Horowitz, Richard Jefferies, Edward Kitsis, Steven Lisberger
Réalisateur	Joseph Kosinski
Référence	<a href="#">IMDb</a>

Sam Flynn, 27 ans, est le fils expert en technologie de Kevin Flynn. Cherchant à percer le mystère de la disparition de son père, il se retrouve aspiré dans ce même monde de programmes redoutables et de jeux mortels où vit son père depuis 25 ans. Avec la fidèle confidente de Kevin, père et fils s'engagent dans un voyage où la mort guette, à travers un cyber univers époustouffant visuellement, devenu plus avancé technologiquement et plus dangereux que jamais...



## 6.2. Page description d'un film :



[Mon Espace](#) [Ajouter un\(e\) ami\(e\)](#) [Liste des Films par user](#) [Deconnexion](#)

# The Hobbit: The Battle of the Five Armies



### Synopsis :

Bilbo and Company are forced to engage in a war against an array of combatants and keep the terrifying Smaug from acquiring a kingdom of treasure and obliterating all of Middle-Earth.

**Utilisateurs qui disposent de ce film (cliquez pour faire une demande d'emprunt):**

[nyanloutre \(DVD\)](#)

### Commentaires :

**nyanloutre**

**2015-01-06 14:17:32**

**Note : 8/10**

Très bon film

ID du Film :

Note : ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☒ 10

Commentaire :

Valider

On y accède depuis la PAGE PRINCIPALE DU SITE, on peut y trouver la description du film, les utilisateurs possédant ce film (avec le support), ainsi que des notes et commentaires. En cliquant sur le nom d'un utilisateur, on envoie une demande d'emprunt.

ID du Film :

Note : ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☒ 10

Commentaire :

Valider

Ici on peut inscrire les commentaires et notes données au films afin de soutenir ou non le film et le conseillé à d'autres utilisateurs.

### 6.3. Page d'accueil du site :

On y trouve le logo de la PopCornTheque ainsi que deux boutons. On peut créer un compte ou se connecter au service avec un compte existant.



[Connexion](#) [Inscription](#)

### 6.4. Page d'inscription au service :

On doit remplir tous les champs afin de créer un compte utilisateur du service PopCornTheque. On accède à cette page en cliquant sur le bouton "Inscription" de la PAGE D'ACCUEIL DU SITE.

Pseudo :

Nom :

Prenom :

Date de naissance :

Rue :

Code postal :

Ville :

Email :

Mot de passe :

Valider



## 6.5. Page principale après connexion



[Mon Espace](#) [Ajouter un\(e\) ami\(e\)](#) [Liste des Films par user](#) [Deconnexion](#)

### Bienvenue sur PopcornTeque

 Films

[The Hobbit: The Battle of the Five Armies](#) [Despicable Me](#) [Transformers: Age of Extinction](#) [The Expendables](#) [The Lego Movie](#)



On y accède après s'être connecté au service après la PAGE DE CONNEXION AU SERVICE. On trouve les liens vers différentes pages permettant d'ajouter un ami, d'emprunter un film, de consulter la liste des films possédés par un user, d'accéder à son espace personnel ainsi que de se déconnecter. On peut rechercher un film dans la barre de recherche ou en choisir un dans la liste (cette liste affiche tous les films disponibles) se trouvant en dessous.

## 6.6. Page d'ajout d'un ami

On y accède depuis la PAGE PRINCIPALE DU SERVICE, on peut y entrer le pseudo d'un ami, ce qui nous permettra de lui envoyer une demande d'ami qu'il pourra accepter ou non. Cela permettra d'accéder rapidement à la liste des films de cet utilisateur.

ID de votre nouvelle ami(e) :

## 6.7. Page mon espace :



[Mon Espace](#) [Ajouter un\(e\) ami\(e\)](#) [Liste des Films par user](#) [Deconnexion](#)

Profil de Tung

### Mes Films

Id du film	Titre	Nom du support
12	Lucy	DVD
13	The Hunger Games	DVD
24	Dora the Explorer: It's a Party	DVD
29	#Yolo	DVD
32	William McKinley at Canton, Ohio	DVD
43	About Conception and Contraception Bobine	

### Mes infos

Nom d'utilisateur : Tung  
 Nom : Florian  
 Prénom STAINE  
 Date de naissance : 06/07/1995  
 Rue : 28 rue du Malsaucy  
 Code postal : 90350  
 Ville : Evette  
 eMail : florian.staine@utbm.fr

### Mes amis

Nyanloutre  
 Zilliq

### Followers

nyanloutre  
 Zilliq

### Mes emprunts

7 à rendre le 2015-02-03 [Rendre](#)  
 72 à rendre le 2015-02-14 [Rendre](#)

### Mes prêts

Lucy a été emprunté par Zilliq et devra etre rendu le 0000-00-00  
 Dora the Explorer: It's a Party a été emprunté par nyanloutre et devra etre rendu le 2015-01-12

### Vous voulez emprunter :

Utilisateur	Titre du film	Annuler la demande
Zilliq	<a href="#">Star Wars: Episode V - The Empire Strikes Back</a>	<a href="#">Supprimer</a>
Zilliq	<a href="#">Pi</a>	<a href="#">Supprimer</a>
Zilliq	<a href="#">Florian - Liebe aus ganzem Herzen</a>	<a href="#">Supprimer</a>
Zilliq	<a href="#">Les visiteurs</a>	<a href="#">Supprimer</a>
Zilliq	<a href="#">Rambo: First Blood Part II</a>	<a href="#">Supprimer</a>
vsilet	<a href="#">The Bourne Legacy</a>	<a href="#">Supprimer</a>
vsilet	<a href="#">Frozen</a>	<a href="#">Supprimer</a>
vsilet	<a href="#">Frozen</a>	<a href="#">Supprimer</a>

### On veut vous emprunter :

Utilisateur	Titre du film	Créer l'emprunt	Refuser la demande
-------------	---------------	-----------------	--------------------

On y accède depuis la PAGE PRINCIPALE DU SERVICE, on peut y trouver la liste de nos films, nos informations personnelles (Pseudo, nom, prénom, date de naissance, adresse, et adresse électronique), nos demandes d'amis (envoyées et reçues) ainsi que nos demandes d'emprunts (envoyées et reçues). C'est l'une des pages les plus importantes.

## 7. Requête SQL

Les variables seront renseignées auparavant soit par des inputs utilisateur ou une requête sur une API

### 7.1. *Ajouter un nouveau film*

```
INSERT INTO FILM VALUES(NULL, :titre, :synopsis, :sortie, :affiche, NULL, :age, :imdb)
```

On utilise la fonction PHP lastInsertId pour récupérer l'ID du film qui vient d'être ajouté

```
INSERT INTO ARTISTE VALUES(NULL, :nom, :prenom)
```

La requête suivante sert à trouver l'ID d'un artiste même si celui ci existe déjà (auquel cas la requête précédente n'aura rien fait)

```
SELECT ART_ID FROM ARTISTE
```

```
WHERE ART_NOM = :nom AND ART_PRENOM = :prenom
```

```
INSERT INTO AVOIR_JOUE_DANS VALUES(:idfilm, :idart)
```

On refait les mêmes requêtes pour le réalisateur

### 7.2. *Ajouter un nouvel utilisateur*

```
INSERT INTO UTILISATEURS VALUES(:user, :password, :nom, :prenom, :date_naissance, :rue, :code_postal, :ville, :mail)
```

### 7.3. *Ajouter un support possédé*

```
INSERT INTO SUPPORT VALUES(NULL, :film_id, :utilisateur_id, :support_nom, 1)
```

### 7.4. *Demande d'emprunt d'un support*

```
INSERT INTO DEMANDE_EMPRUNT VALUES(:utilisateur_id, :support_id)
```

### 7.5. *Emprunt d'un support*

```
INSERT INTO EMPRUNT VALUES(NULL, :current_id_user, :current_id_support, :currt_date, :retour_emprunt_date, NULL, :libre)
```

```
DELETE FROM DEMANDE_EMPRUNT
```

```
WHERE SUP_ID = :current_id_support AND UTI_ID = :current_id_user
```

```
UPDATE SUPPORT SET SUP_LIBRE = 0 WHERE SUP_ID = :current_id_support
```

### 7.6. *Ajout d'un commentaire sur un film*

```
INSERT INTO COMMENTAIRES_FILM VALUES(NULL, :film_id, :user_id, :commentaire, :note, :date_actuelle)
```

### 7.7. *Ajout d'un commentaire sur un utilisateur*

```
INSERT INTO COMMENTAIRES_UT VALUES(NULL, :current_user_id, :target_user_id, :commentaire, :note, :date_actuelle)
```

### 7.8. *Ajouter un utilisateur en amis*

```
INSERT INTO ETRE_AMI VALUES(:target_user_id, :current_user_id)
```

### 7.9. *Afficher la liste de film disponible*

```
SELECT * FROM FILM
```



### **7.10. Liste de film d'un utilisateur**

```
SELECT F.FILM_ID, SUP_NOM, FILM_AFFICHE, FILM_TITRE FROM SUPPORT AS S
INNER JOIN FILM AS F
    ON F.FILM_ID = S.FILM_ID
WHERE UTI_ID = ?
```

Le Inner Join est ici utilisé pour avoir les affiches et les noms des films que possède un utilisateur.

### **7.11. Ajouter un support**

```
INSERT INTO SUPPORT VALUES(NULL, :film_id, :utilisateur_id, :support_nom, 1)
```

### **7.12. Connexion au site**

On récupère le mot de passe enregistré pour cet utilisateur puis on le compare à celui envoyé

```
SELECT UTI_MOT_DE_PASSE FROM UTILISATEURS WHERE UTI_ID = :nom
```

### **7.13. Afficher les détails d'un film**

```
SELECT * FROM FILM WHERE FILM_ID = ?
```

On cherche tous les supports libres

```
SELECT * FROM SUPPORT WHERE FILM_ID = ? WHERE SUP_LIBRE = 1
```

On cherche tous les commentaires

```
SELECT * FROM COMMENTAIRES_FILM WHERE FILM_ID = ?
```

### **7.14. Filtrage des films disponibles**

Le CONCAT sert à contourner un problème lié au php qui ne considère pas les % comme partie de la requête

```
SELECT * FROM FILM WHERE FILM_TITRE LIKE CONCAT('%', ?, '%')
```

### **7.15. Afficher informations utilisateur**

```
SELECT * FROM UTILISATEURS WHERE UTI_ID = ?
```

On affiche tous les supports que possède l'utilisateur

```
SELECT F.FILM_ID, SUP_NOM, FILM_AFFICHE, FILM_TITRE FROM SUPPORT AS S
INNER JOIN FILM AS F
    ON F.FILM_ID = S.FILM_ID WHERE UTI_ID = ?
```

On affiche les personnes ajoutées en ami

```
SELECT * FROM ETRE_AMI WHERE UTI_ID_SOURCE = ?
```

Et les personnes qui nous ont rajouté un ami

```
SELECT * FROM ETRE_AMI WHERE UTI_ID_CIBLE = ?
```

**7.16. Affichage de tous les emprunts en cours**

```
SELECT SUP_ID, EMPR_RETOUTR_THEORIQUE
FROM EMPRUNT
WHERE UTI_ID = ? AND EMPR_RENDU = FALSE
```

**7.17. Affichage de tous les prêts en cours**

```
SELECT S.UTI_ID, E.EMPR_RETOUTR_THEORIQUE, F.FILM_TITRE, E.UTI_ID, E.EMPR_ID
FROM EMPRUNT AS E INNER JOIN SUPPORT AS S ON E.SUP_ID = S.SUP_ID INNER JOIN
FILM AS F ON S.FILM_ID = F.FILM_ID WHERE S.UTI_ID = ? AND EMPR_RENDU = FALSE
```

**7.18. Affichage de toute les demandes d'emprunt envoyées**

```
SELECT S.SUP_ID, S.UTI_ID, F.FILM_ID, F.FILM_TITRE FROM DEMANDE_EMPRUNT AS DE
INNER JOIN SUPPORT AS S
    ON DE.SUP_ID = S.SUP_ID
INNER JOIN FILM AS F
    ON S.FILM_ID = F.FILM_ID
WHERE DE.UTI_ID = ?
```

**7.19. Affichage de toutes les demandes d'emprunt reçues pouvant être validées**

```
SELECT DE.UTI_ID, S.SUP_ID, S.SUP_LIBRE, F.FILM_ID, F.FILM_TITRE
FROM DEMANDE_EMPRUNT AS DE

INNER JOIN SUPPORT AS S
    ON DE.SUP_ID = S.SUP_ID
INNER JOIN FILM AS F
    ON S.FILM_ID = F.FILM_ID

WHERE S.UTI_ID = ? AND S.SUP_LIBRE = 1
```

**7.20. Associer l'ID du support dans demande\_emprunt à un nom de film**

```
SELECT F.FILM_ID, F.FILM_TITRE FROM DEMANDE_EMPRUNT AS DE
INNER JOIN SUPPORT AS S
    ON DE.SUP_ID = S.SUP_ID
INNER JOIN FILM AS F
    ON S.FILM_ID = F.FILM_ID

WHERE DE.UTI_ID = ?
```

Ici la requête permet de savoir le nom du film que l'on veut emprunter.

**7.21. Marquer un support emprunté comme rendu**

```
UPDATE EMPRUNT SET EMPR_RETOUTR_REEL = :retour_emprunt_date, EMPR_RENDU
= :libre WHERE EMPR_ID = :emprunt_id
```

```
SELECT SUP_ID FROM EMPRUNT WHERE EMPR_ID = :emprunt_id
```

```
UPDATE SUPPORT SET SUP_LIBRE = 1 WHERE SUP_ID = :id_support
```

## 8. Description API

Afin de récupérer facilement des informations sur les films nous utilisons l'API de <http://www.omdbapi.com/>. Le fonctionnement de cet API est simple :

1. Quand l'utilisateur veut ajouter un nouveau film, le serveur envoie une requête de ce type <http://www.omdbapi.com/?s=tron>
2. Le serveur récupère un tableau comportant une liste de tous les films correspondant à la recherche sous forme d'un tableau (il contient le nom du film, les acteurs, réalisateurs, affiches, ...) et le présente à l'écran
3. Une fois que l'utilisateur a choisi son film, il est sauvegardé dans la base de donnée et toutes les relations liées au film sont créées

## 9. Explication de certaines sources PHP

Pour que les utilisateurs restent connecté tout au long de leur navigation sur le site nous avons décidé d'utiliser les sessions qui permettent d'identifier précisément un ordinateur.

### 9.1. Les sessions - la connexion

```
<?php
```

```
ini_set('display_errors', 'On'); //Affichage des erreurs
```

```
session_start(); //Ici on démarre une nouvelle session qui se compose d'un cookie stockée sur l'ordinateur de l'utilisateur et qui permet au serveur d'identifier l'ordinateur
```

```
$_SESSION = array();
```

```
$user = $_POST['user'];
```

```
$password = sha1($_POST['password']); //La fonction sha1 calcule l'emprunte sha1 du mot de passe ce qui permet de ne jamais avoir le mot de passe en clair qui est stocké
```

```
try
```

```
{
```

```
    $bdd = new PDO('mysql:host=localhost;dbname=PopCornTheque', 'poppoppop', 'password');
```

```
}
```

```
catch (Exception $e)
```

```
{
```

```
    die('Erreur : ' . $e->getMessage());
```

```
}
```

```
$req = $bdd->prepare('SELECT UTI_MOT_DE_PASSE FROM UTILISATEURS WHERE UTI_ID = :nom');
```

```
$req->execute(array(
```

```
    'nom' => $user
```

```
));
```

```
$realpassword = $req->fetch()[0];
```

```
if ($realpassword == $password){ //On vérifie que le mot de passe est le bon
```

*//Ces deux variables session sont des variables qui sont associées à la session en cours. Même si l'utilisateur change de page elles seront conservées ce qui permet de l'identifier sur tout le site*

```
$_SESSION['login'] = $user;
$_SESSION['password'] = $password;
header('Location: index.php');

}
else{

    echo '<p>Mauvais mot de passe</p>';
    include 'connexion.html';
}

$req->closeCursor();
?>
```

## 10. Les sessions - l'identification sur les autres pages

Ce code est lancé au début de chaque pages nécessitant une identification

*session\_start(); //On relance la session, si l'utilisateur c'était précédemment connecté, cela récupère les variables de la session sinon aucune variable n'est retournée*

```
if ((!isset($_SESSION['login'])) || (empty($_SESSION['login'])))
{
```

*// la variable 'login' de session est non déclaré ou vide*

```
echo '<li><a href="connexion.html" title="Connexion">Connexion</a></li>';
echo '<li><a href="formulaireinscription.html">Inscription</a></li></nav>';
exit();
}
```

On peut ensuite utiliser `$_SESSION['login']` pour identifier l'utilisateur sur cette page

## 11. Les sessions - la déconnexion

*session\_start(); //On relance la session*

*\$\_SESSION = array(); //On vide toutes les variables de session*

*session\_destroy(); //On supprime la session*

```
header('Location: index.php'); //On revient vers la page d'accueil
```