

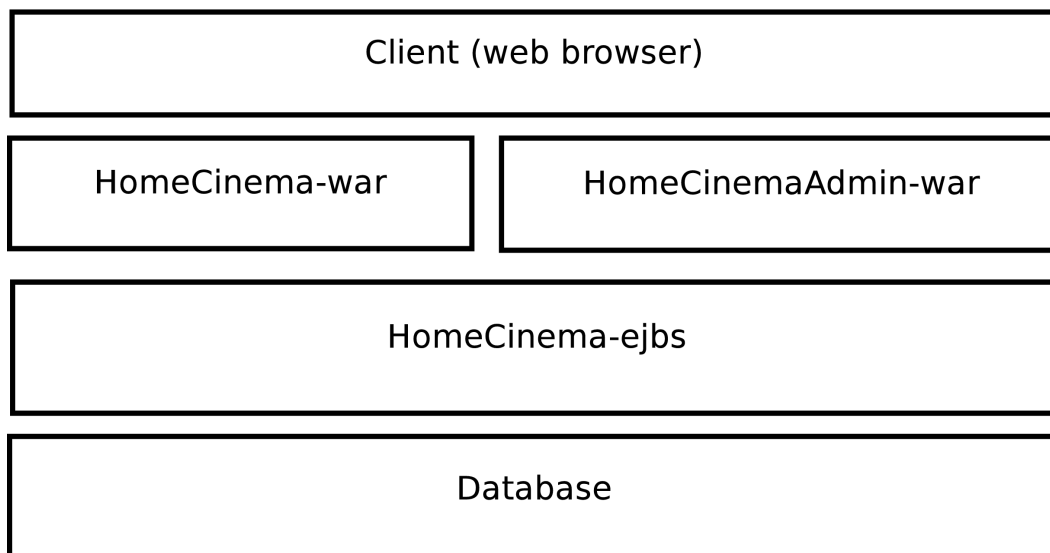
# Rapport de conception système

Dans le cadre du Projet ECOM de M2PGI, nous avons développé une application de e-commerce nommé HomeCinéma permettant d'acheter des films et de les télécharger ou de les visionner en streaming. Le document suivant présente l'architecture de notre application et détaille les différents composants pour permettre de prendre rapidement connaissance des principaux composants JEE de votre application ECOM. Dans un premier temps, nous donnerons une description globale de notre conception, puis nous décrirons notre schéma d'architecture système (SAS) en détaillant les composants et les liens utilisés pour chaque fonctionnalité.

## 1. Description globale

### 1.1. Architecture

Notre application de e-commerce basée sur la technologie J2EE suit une architecture multi-tier. D'une manière tout à fait classique, nous retrouvons la Base de Données tout en bas de la « pile ». Elle est utilisée par la couche modèle, module que nous avons appelé HomeCinema-ejb. Au-dessus, on trouve la couche logique, cœur de notre application : les Managed Bean. Nous avons fait le choix d'avoir deux WARs différents, un pour les clients, l'autre pour l'administration. Enfin, tout en haut de cette pile, on trouve les clients, ici des navigateurs web.



*Figure 1 : Architecture mise en place sous forme de « Pile »*

Nous avons fait le choix de mettre les parties communes entre les WARs et les EJBs dans une bibliothèque. Cette dernière comprend des Data Transfer Object (DTO), et les interfaces des EJBs. De cette manière, les WAR et les EJBs peuvent tourner sur des serveurs J2EE différents. Il nous est également possible d'avoir une application "lourde" qui utilise directement les EJBs, afin de par exemple remplir notre Base de Données. Cette dernière est utilisée par les EJBs grâce à un Object-relational mapping (ORM), ici EclipseLink. Enfin, toutes nos pages web sont des pages JSF.

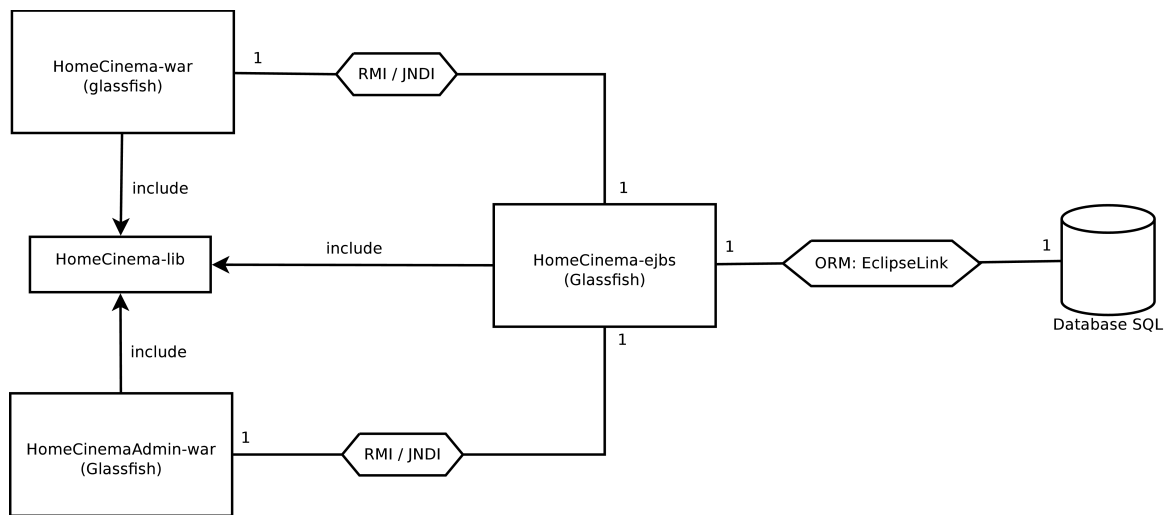


Figure 2 : Diagramme C&C

## 1.2. Entités

Nous avons des produits (*Product*), qui contiennent au moins un film (*Film*). Ces derniers ont un produit principal, mais ils peuvent aussi appartenir à d'autres produits, comme des packs. Les films sont composés de fichiers vidéos (*Video*): une bande annonce et au moins un fichier vidéo du film. Ils ont des acteurs et des réalisateurs que nous identifions comme des personnes (*Person*) qui peuvent avoir participé à un ou plusieurs films. Les films ont également un ou plusieurs genres (*Genre*) et un pays d'origine (*Country*).

Les utilisateurs du site sont enregistrés dans *User*. La création et l'activation d'un utilisateur est faite à l'aide d'un code généré (*UserActivation*). Les utilisateurs activés ont un caddie qui est une liste de produits. Une fois le paiement validé, le caddie est transformé en transaction terminée. Le caddie et les transactions sont stockés dans *Transaction*. L'utilisateur possède une liste de films (*UserFilms*) comprenant des données spécifiques à chaque utilisateur, comme l'état du visionnage d'un film.

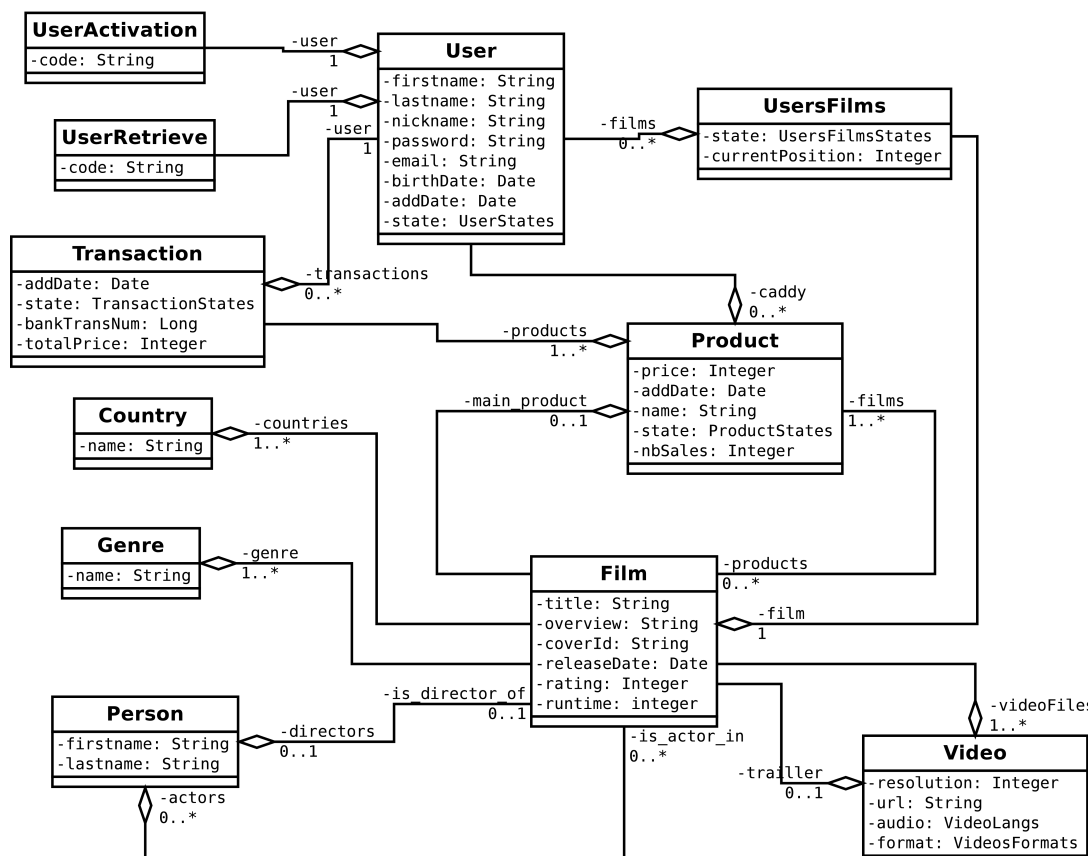


Figure 3 : Diagramme de classe

### 1.3. EJBs

Les fonctions ont été regroupées dans les EJBs selon l'entité qu'elle modifie. Tous nos EJBs sont *Stateless*. De plus, toutes les fonctions manipulant des DTOs de films, produits ou genres doivent savoir dans quel langage est le DTO. Le détail de nos EJBs est le suivant :

ManageProduct	<ul style="list-style-type: none"><li>✓ création et modification d'un produit</li><li>✓ modification des films d'un produit</li><li>✓ ajout d'un film à un produit (en le créant si nécessaire)</li><li>✓ consultation de la liste des produits et des informations d'un produit</li><li>✓ recherche de produits avec des filtres (acteur, réalisateur, etc.)</li></ul>
ManageFilm	<ul style="list-style-type: none"><li>✓ création et modification d'un film</li><li>✓ ajout de genres, d'acteurs et de réalisateurs à un film</li><li>✓ ajout de fichier vidéo et trailer à un film</li><li>✓ choix du produit principal</li><li>✓ récupération d'une liste de tous les films</li><li>✓ récupération des informations sur un film</li></ul>
ManagePerson	<ul style="list-style-type: none"><li>✓ récupération d'une personne</li></ul>
ManageTransaction (fonctions associées au paiement et au caddie)	<ul style="list-style-type: none"><li>✓ ajout et suppression des produits du caddie</li><li>✓ réalisation du paiement</li><li>✓ consultation du caddie</li></ul>
ManageUser	<ul style="list-style-type: none"><li>✓ inscription d'un utilisateur</li><li>✓ modification des informations d'un utilisateur</li><li>✓ récupération d'un mot de passe</li><li>✓ activation/désactivation d'un utilisateur</li></ul>
ManageUserFilm	<ul style="list-style-type: none"><li>✓ mise à jour de la position de lecture d'un film d'un utilisateur</li><li>✓ récupération de la position de lecture d'un film d'un utilisateur</li></ul>
ManageVideo	<ul style="list-style-type: none"><li>✓ consultation de la liste des vidéos</li></ul>

Figure 4 : Tableau détaillant les EJBs

## 2. Description des composants

### 2.1. Composants Utilisateur

→ *Fonctionnalités Utilisateur* → *Inscription* :

Un visiteur peut s'inscrire sur le site via un formulaire créé à cet effet qui se trouve dans la page *signup.xhtml* et qui est accessible depuis un lien hypertexte se trouvant sur toutes les pages du site. Les informations du formulaire sont transmises à un Managed Bean nommé *SignUpManagedBean*, que nous avons mis en *RequestScoped* et qui se charge de vérifier les informations envoyées par l'utilisateur. Celles-ci sont ensuite envoyées à l'EJB *ManageUser* pour l'insertion dans notre système de persistance. Le transfert de données est simplifié par l'usage du DTO *UserDto*.

De plus, un mail est envoyé au nouvel utilisateur afin qu'il active son compte en confirmant son inscription. Cette activation utilise *ActivateUserManagedBean* et *ManageUser*.

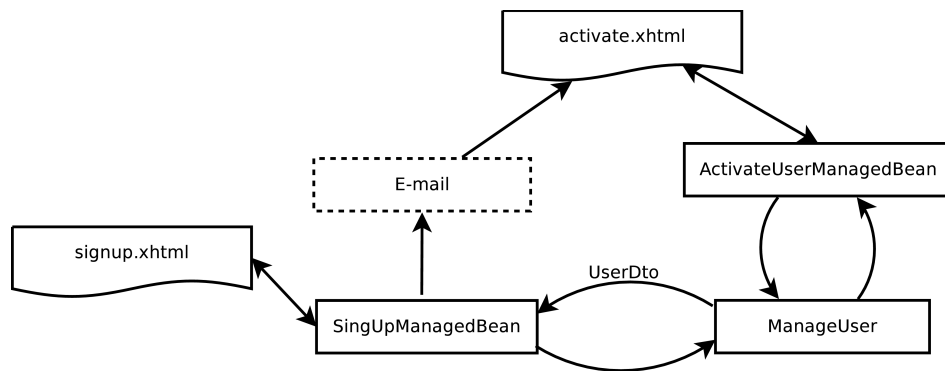


Figure 5 : SAS du composant lié à la fonctionnalité d'inscription

→ *Fonctionnalités Utilisateur* → *Connexion/Déconnexion* :

Pour avoir accès à toutes les fonctionnalités de l'application, l'utilisateur doit s'authentifier en renseignant son adresse e-mail (ou son nom d'utilisateur) et son mot de passe (donné lors du processus d'inscription). Les données sont transmises au Managed Bean SessionManagedBean qui est en SessionScoped. L'EJB ManageUser est là encore utilisé. Cette fois-ci, il vérifie si le couple (adresse e-mail ou nom d'utilisateur / mot de passe) est présent dans la Base de Données. Dans le cas où le couple est valide, une variable de session sera stockée durant toute la session de l'utilisateur. La déconnexion consiste à réinitialiser la variable de session créée lors de la connexion.

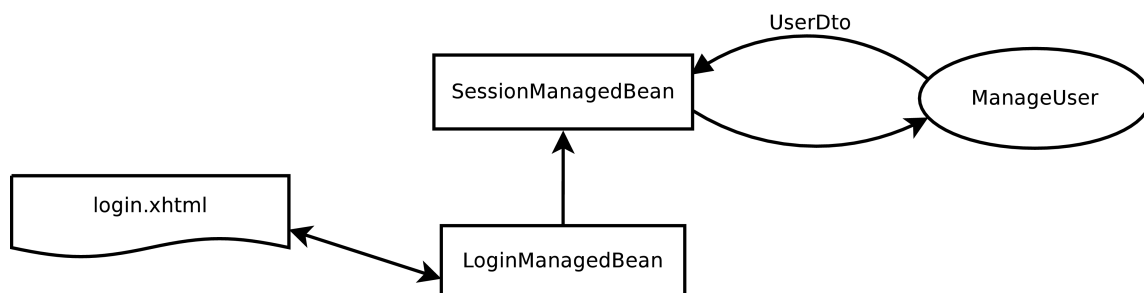


Figure 6 : SAS du composant lié aux fonctionnalités de connexion/déconnexion

→ *Fonctionnalités Utilisateur* → *Retrouver mot de passe* :

Pour retrouver un mot de passe oublié, l'utilisateur donne son adresse e-mail et il reçoit ensuite un message avec un lien contenant un code aléatoire. Ce lien dirige l'utilisateur vers un formulaire sur lequel il peut renseigner un nouveau mot de passe qui est envoyé à l'EJB ManageUser via le Managed Bean en RequestScope intitulé ForgotPassManagedBean.

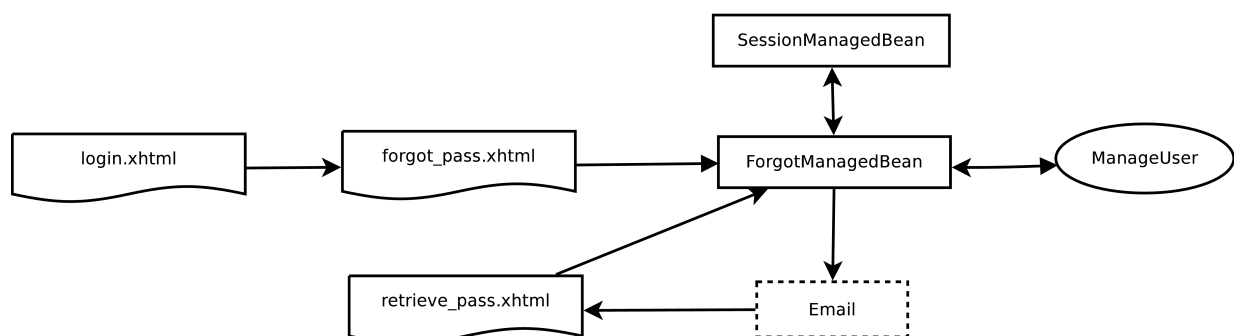


Figure 7 : SAS du composant lié à la fonctionnalité de récupération d'un mot de passe

→ *Fonctionnalités Utilisateur* → *Mise à jour des informations du compte* :

Le changement d'adresse e-mail se fait depuis l'espace membre, l'opération est très basique et consiste à renseigner la nouvelle adresse e-mail que l'on souhaite assigner à son compte et le mot de passe pour confirmer. Si le mot de passe est correct, le changement se fait au niveau du sous-système EJB via le Managed Bean nommé `InfoUserManagedBean` (RequestScope). Sinon, un message d'erreur est affiché. Le changement de mot passe est similaire et consiste à renseigner le nouveau mot de passe, sa confirmation et l'ancien mot de passe.

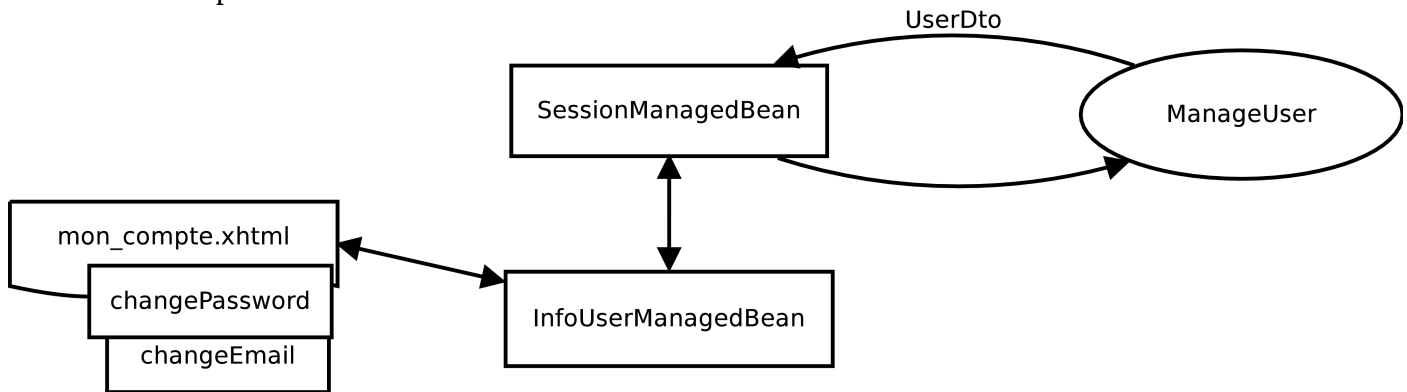


Figure 8 : SAS du composant lié à la fonctionnalité de mise à jour des informations d'un compte

→ *Fonctionnalités Utilisateur* → *Gestion du langage* :

Un utilisateur peut également changer la langue d'affichage du site depuis n'importe quelle page depuis l'entête du site. Le Managed Bean nommé `LanguageManagedBean` est utilisé pour gérer le paramétrage du langage. À l'heure actuelle, les langues *française* et *anglaise* sont supportées. Un cookie est également utilisé pour garder en mémoire le paramétrage des utilisateurs.

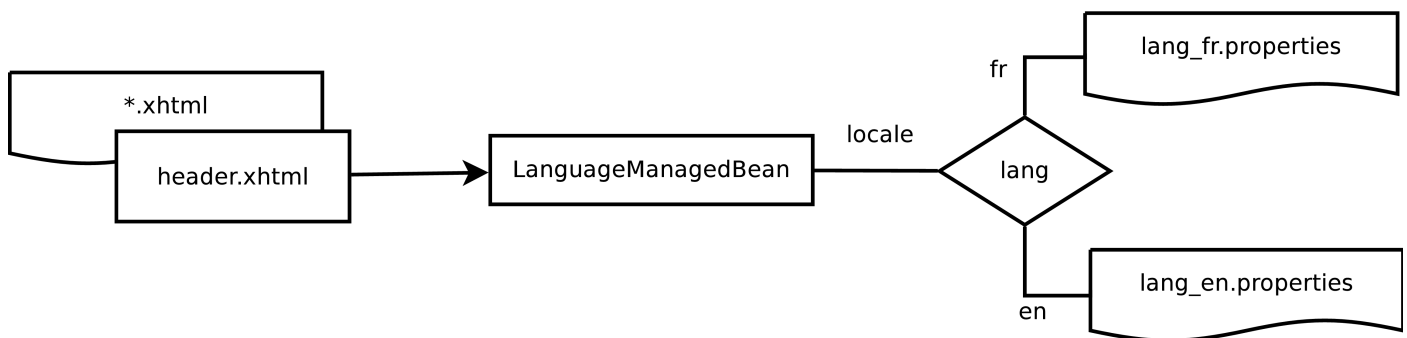


Figure 9 : SAS du composant lié à la fonctionnalité de gestion de la langue d'affichage du site

→ *Fonctionnalités Films/Produits* → *Liste de produits et recherche de produits* :

La consultation de la liste des produits, c'est-à-dire de la liste des films et des packs de films disponibles, est l'une des fonctionnalités centrales de notre application. Le principe est le suivant : l'utilisateur sélectionne d'abord un critère général (sélection des packs, des meilleurs films ou des nouveautés) ou des critères spécifiques à un film (titre, date de sortie, un réalisateur, un acteur et un ou plusieurs genres). Ensuite, une liste de produits validant le ou les critères est affichée. La page du site réalisant cette fonctionnalité est *films.xhtml*. Cette page est divisée en deux parties : la première permet de paramétrer la recherche et la seconde permet d'afficher la liste des résultats. Cette liste est mise à jour sans rechargement de toute la page en Ajax à chaque envoi d'une nouvelle recherche.

Cette fonctionnalité utilise deux principaux Managed Bean nommés SearchProductManagedBean et ListProductsManagedBean dont les détails sont les suivants :

- SearchProductManagedBean est utilisé pour stocker tous les critères de recherche dans le cas d'une recherche spécifique de films selon des critères précis (titre, date, etc.). Il est défini en SessionScoped pour permettre de garder les critères de recherche tout au long de la session.

- ListProductsManagedBean est utilisé pour gérer l'affichage de la recherche en fournissant une liste de ProductDto à la page. Pour cela, il gère des données liées à l'affichage de *films.xhtml* (type de recherche courant et pagination de la liste des résultats) et l'exécution de la recherche en faisant appel à l'EJB ManageProduct. Enfin, ce Managed Bean est également utilisé pour produire une liste statique, récupéré une seule fois, de films intéressants qui sont affichés en pied de page.

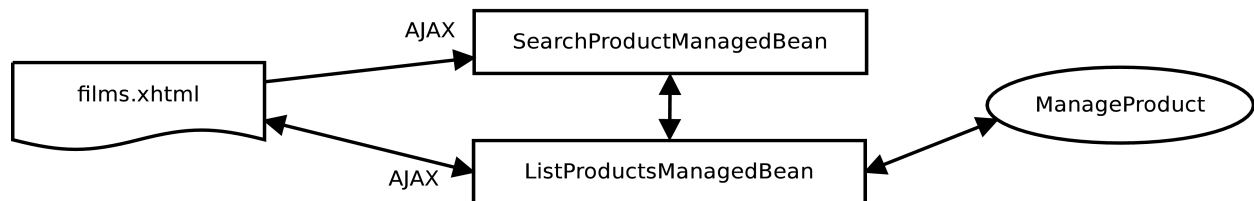


Figure 10 : SAS du composant lié à la fonctionnalité d'affichage et de recherche de produits

→ Fonctionnalités Films/Produits → Fiche d'un film et d'un pack de films :

Pour chaque film disponible, l'utilisateur peut consulter une fiche donnant plusieurs informations sur celui-ci. Cette fiche permet également de voir le prix et le nombre de ventes du film. Cette fonctionnalité est mise en place sur la page *fiche\_film.xhtml*.

Cette page utilise LoginManagedBean (affichage paramétré si l'utilisateur est connecté) et principalement le Managed Bean nommé FilmManagedBean. Ce Managed Bean est utilisé pour récupérer le FilmFicheDto contenant les informations du film à afficher dans la fiche. Il est défini en ViewScope. Le ManagedBean récupère tout d'abord un identifiant de film. Il fait ensuite appel à l'EJB ManageFilm pour récupérer le FilmFicheDto correspondant. Enfin la page *fiche\_film.xhtml* récupère ces informations via les méthodes de FilmManagedBean. De plus, FilmManagedBean utilise LanguageManagedBean pour transmettre la langue utilisée par l'utilisateur à l'EJB et récupérer les informations dans cette langue.

Il est également possible de consulter une fiche pour un pack de films. Cette fonctionnalité est accessible sur la page *fiche\_product.xhtml*. Elle utilise LoginManagedBean et principalement le Managed Bean intitulé ProductManagedBean. Ce Managed Bean est semblable à FilmManagedBean. Il est lui aussi défini en ViewScoped. Il récupère à partir d'un identifiant le ProductDto correspondant via l'EJB ManageProduct. S'il ne s'agit pas d'un pack (c'est-à-dire si le produit contient un seul film), le ManagedBean fait une redirection vers la page *fiche\_film.xhtml*. Il fournit alors toutes les méthodes nécessaires à la page *fiche\_product.xhtml* pour afficher les informations du pack. Là encore, LanguageManagedBean est utilisé pour permettre un affichage des informations dans la langue sélectionnée par l'utilisateur. Enfin, un plugin de slider JQuery est utilisé pour gérer l'affichage des affiches des films du pack.

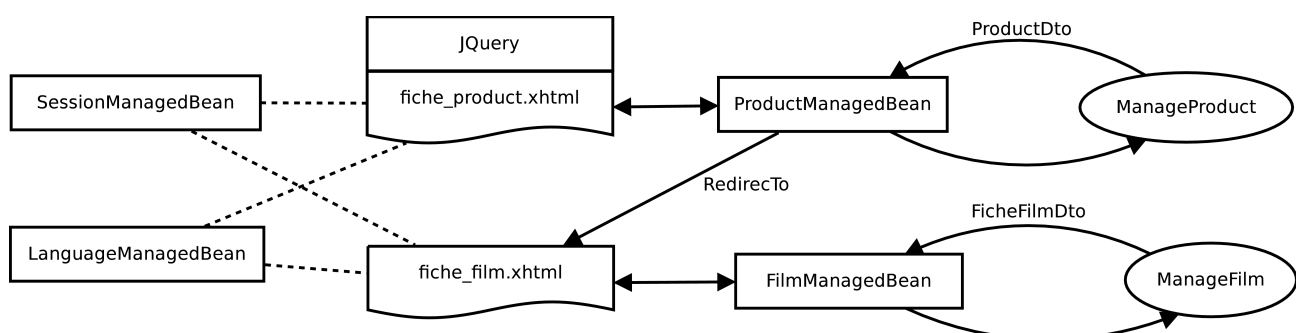


Figure 11 : SAS du composant lié aux fonctionnalités d'affichage de fiche film et fiche pack de films

→ *Fonctionnalités Films/Produits* → *Visionner un film* :

Les utilisateurs peuvent visionner en streaming ou télécharger les films qu'ils ont achetés. La page de visionnage est *visionneuse.xhtml*. Elle utilise la gestion vidéo de HTML5 et plusieurs Managed Bean :

- SessionManagedBean est utilisé pour vérifier que l'utilisateur est connecté.
- AccountManagedBean est utilisé pour vérifier si le film a bien été acheté par l'utilisateur qui accède à la page en interrogeant l'EJB ManageUser et sa fonction de récupération des films d'un utilisateur.
- FilmManagedBean est utilisé pour récupérer la vidéo via l'EJB ManageFilm. Plusieurs paramètres peuvent être sélectionnés (qualité, format et langue).
- UsersFilmsManagedBean gère la progression de la lecture de la vidéo via l'EJB ManageUsersFilm.

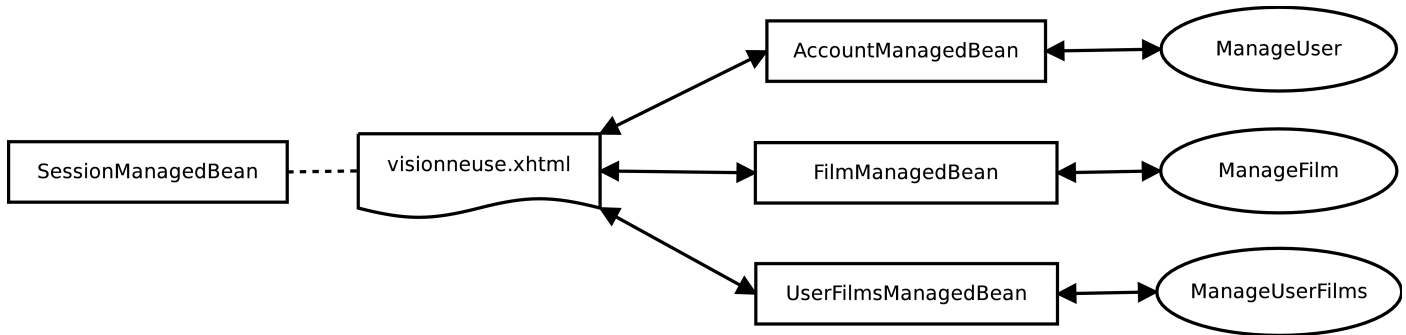


Figure 12 : SAS du composant lié à la fonctionnalité de visionnage d'un film

→ *Fonctionnalités Caddie* → *Ajouter/Supprimer du caddie* :

L'ajout d'un film au caddie se fait par la page de sa fiche lorsque l'utilisateur est connecté. La modification du caddie est aussi possible à partir de la page mon caddie où il est possible d'y supprimer un produit et de valider le caddie pour passer au paiement. Cette fonctionnalité utilise deux Managed Bean :

- CaddieManagedBean, en RequestScoped, contient les opérations à effectuer sur le caddie, telle que lister les produits, ajouter un produit et supprimer un produit. Il utilise l'EJB ManageTransaction pour modifier le caddie et l'EJB ManageProduct afin de récupérer la liste des produits du caddie. Les entités sont converties en CaddieDto qui contient des ProductDto.
- SessionManagedBean, en SessionScoped, maintient un état du caddie de l'utilisateur afin de faciliter son accès. Ce dernier utilise l'EJB ManageTransaction.

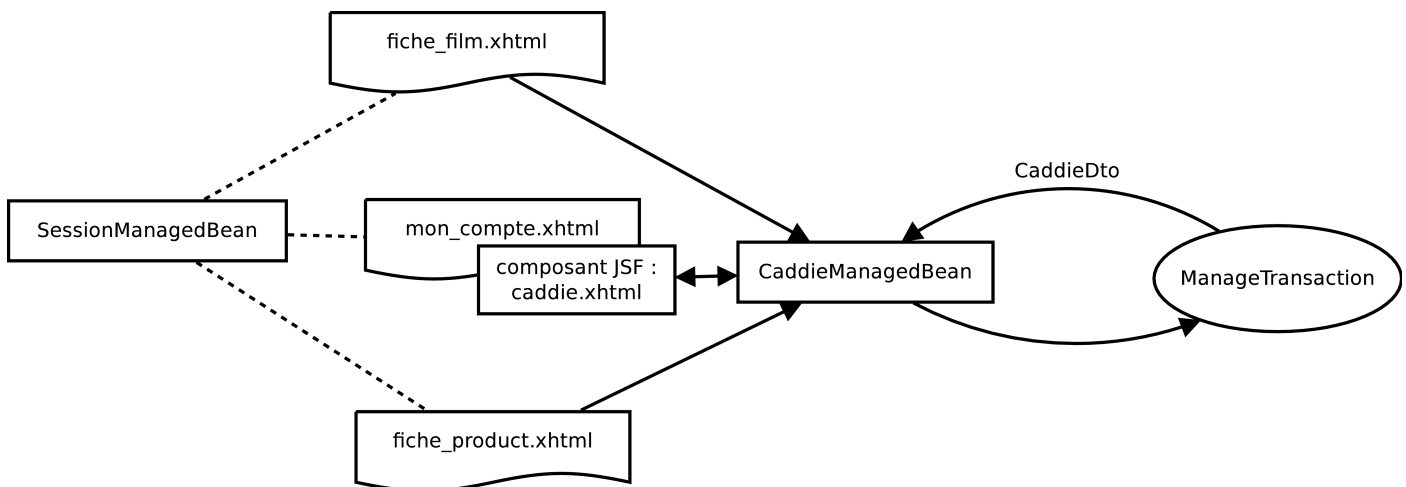


Figure 13 : SAS du composant lié aux fonctionnalités de gestion du caddie

### → Fonctionnalités Caddie → Paiement :

Une fois le caddie complété, sur l'onglet « mon caddie » de la page *moncompte.xhtml*, l'utilisateur peut le valider. Il arrive alors sur la page de paiement. Une fois le paiement effectué, les films du caddie sont ajoutés aux films de l'utilisateur et la transaction est enregistrée dans son historique. Lors de la demande de la validation du caddie, l'utilisateur passe dans un état spéciale : paiement. Cet état empêche la navigation sur le site, si une page est accédée, alors le paiement est automatiquement annulé. Cette fonctionnalité utilise un Managed Bean : *PaiementManagedBean* qui est en *RequestScoped*. Ce Managed Bean utilise l'EJB *ManageTransaction*. Une fois le paiement réalisé, l'utilisateur est redirigé vers la page "mon compte".

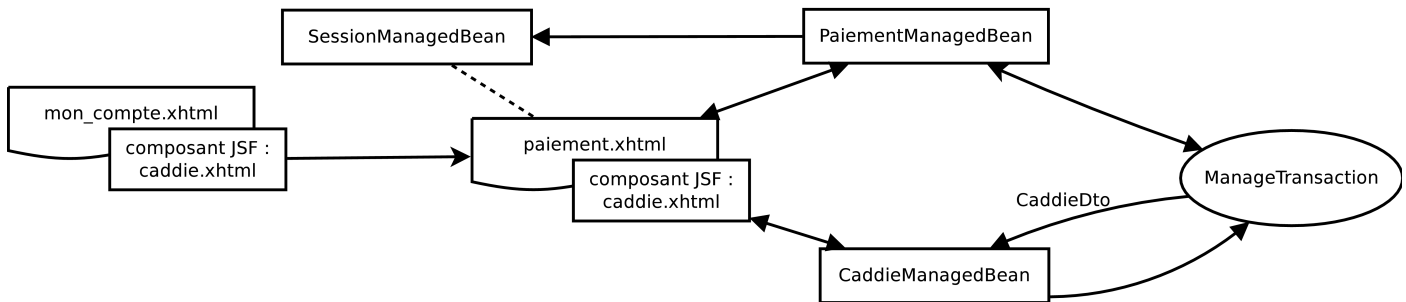


Figure 14 : SAS du composant lié à la fonctionnalité paiement

### → Fonctionnalités Caddie → Historique :

L'utilisateur peut voir son historique des transactions sur la page *moncompte.xhtml*. Cette fonctionnalité utilise *HistoryManagedBean* (*RequestScoped*) qui lui-même utilise l'EJB *ManagedTransaction*.

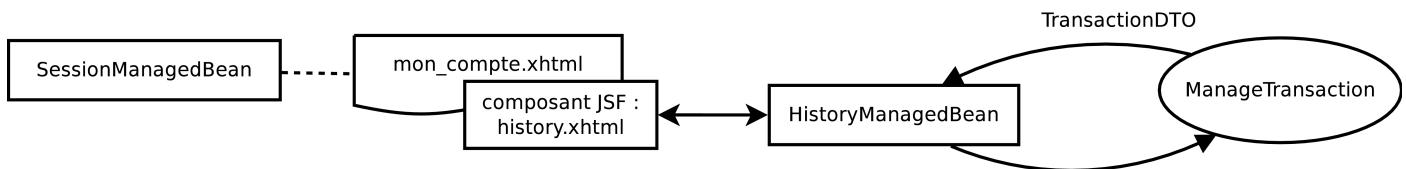


Figure 15 : SAS du composant lié à la fonctionnalité de visualisation d'un historique des transactions

### → Fonctionnalités complémentaires du site :

La page *sitemap.xhtml* est une page statique qui présente le plan du site. La page *contact.xhtml* est une page proposant un formulaire de contact utilisant le Managed Bean en *RequestScope* nommé *ContactManagedBean*. Ce dernier est uniquement utilisé pour récupérer les valeurs du formulaire et retourner un statut (envoi réussi ou non). À l'heure actuelle, la fonctionnalité de contact n'est pas utilisée car seule la structure (page JSF et Managed Bean) a été développée. Aucun traitement du message est donc disponible et le statut renvoyé est toujours « envoi réussi ».

## 2.2. Composants Administrateur

Une version administrateur non complète existe. Elle permet de consulter, d'ajouter et de modifier les films, les produits et les utilisateurs. Pour chaque entité, deux pages existent : une pour obtenir la liste des éléments et une pour créer/afficher/modifier un élément. Un Managed Bean est associé à chaque page. Chaque page fait appel à l'EJB de l'entité à laquelle il est associé.



Composant <i>page JSF</i>	Managed Bean → <i>EJB</i>	Description
Products <i>products.xhtml</i>	ProductManagedBean → <i>ManageProduct</i>	✓ affichage de la liste des produits
Product <i>product.xhtml</i>	ProductManagedBean → <i>ManageProduct</i>	<ul style="list-style-type: none"> <li>✓ création d'un produit</li> <li>✓ affichage d'un produit</li> <li>✓ modification d'un produit</li> <li>✓ activation/désactivation d'un produit</li> <li>✓ ajout/suppression d'un film à un produit</li> </ul>
Users <i>users.xhtml</i>	UsersManagedBean → <i>ManageUser</i>	<ul style="list-style-type: none"> <li>✓ affichage de la liste des utilisateurs</li> <li>✓ suppression d'un utilisateur</li> </ul>
User <i>user.xhtml</i>	UserManagedBean → <i>ManageUser</i>	<ul style="list-style-type: none"> <li>✓ affichage des informations d'un utilisateur</li> <li>✓ édition des informations d'un utilisateur</li> <li>✓ activation/désactivation d'un utilisateur</li> </ul>
Films <i>films.xhtml</i>	FilmsManagedBean → <i>ManageFilm</i>	✓ affichage de la liste des films
Film <i>film.xhtml</i>	FilmManagedBean → <i>ManageFilm</i>	<ul style="list-style-type: none"> <li>✓ affichage des informations d'un film</li> <li>✓ affichage des acteurs, réalisateurs et genre d'un film</li> <li>✓ modification/création d'un film</li> <li>✓ chargement des informations d'un film depuis TMDB</li> <li>✓ consultation des produits auquel le produit appartient</li> <li>✓ sélection du produit principal</li> </ul>
Videos <i>videos.xhtml</i>	VideoManagedBean → <i>ManageVideo</i>	<ul style="list-style-type: none"> <li>✓ affichage de toutes les vidéos d'un film</li> <li>✓ ajout d'une vidéo ou d'un trailer à un film</li> </ul>

Figure 16 : Tableau détaillant la partie administration