

Cahier de charges - Projet Tuteuré

M. CARDINALE, S. LEFEVRE, Z. MELLAH, A. MIRAS

Table des matières

Introduction	2
Objectif du projet	2
Méthode de travail	2
Synopsis (user stories)	2
Le cas de l'internaute (pas connecté)...	2
qui cherche un film	2
qui cherche une personne	2
qui cherche une recommandation de film	2
qui veut s'inscrire	2
qui veut se connecter	2
Le cas de l'utilisateur (connecté)...	3
qui se connecte	3
qui veut ajouter un film vu/préfére	3
qui se déconnecte	3
Solution technique	3
Développement	3
Déploiement	3
Application technique	3
Algorithme de recommandation	4
Recommandation en fonction du contenu	4
Recommandation par rapprochement d'utilisateur	4
Notre choix	4
Technologies	4
Données	4
Diagrammes	5
Gantt	5
Cas d'utilisation	5
Classes	6
Base de données	7
Wireflows	8

Introduction

Ce document à pour objectif de présenter la structure de notre projet et de détailler certains points énoncés dans le cahier des charges fonctionnels de manière plus précise.

Objectif du projet

Notre projet aura pour objectif la création d'une application web de recommandations de films.

L'application sera écrite intégralement en Python en utilisant le framework Django, déployée sur un serveur Heroku / AWS et sera accessible via une interface web. Elle comprendra un système d'authentification et de création de profil, les utilisateurs auront la possibilité d'enregistrer les films qu'ils ont vus et ceux qu'ils souhaitent visionner et de créer des listes de films (favoris par exemple). Les utilisateurs pourront se suivre entre eux et noter les films. Chaque film, réalisateur et acteur aura sa propre page détaillée contenant ses informations. Les pages de films seront agrémentées de bandes annonces.

Le noyau du projet sera l'algorithme de recommandation des films qui sera fonction des oeuvres vus, des personnes similaires, des notes et des genres.

Les fonctionnalités sociales ne font pas parties de l'implémentation minimale.

Méthode de travail

Ce projet se déroulant sur une longue période nous auront besoin d'une grande flexibilité afin d'aboutir à un rendu final optimal. Nous opterons donc pour l'utilisation de la méthode agile SCRUM, de plus cette méthode étant couramment utilisée dans le monde du travail cela nous permettra d'avoir une expérience initiatique avant de rentrer dans le monde du travail.

Synopsis (user stories)

Le cas de l'internaute (pas connecté)...

qui cherche un film

L'internaute recherche un film dans la barre de recherche. Il arrive sur la page du film et peut connaître son auteur, ses acteurs, son genre, son année de sortie, ses notes, un synopsis.

qui cherche une personne

L'internaute recherche la personne dans la barre de recherche. Il arrive sur la page de la personne et peut lire une biographie, son âge, le nombre de films associés,...

qui cherche une recommandation de film

L'internaute clique sur le bouton "Recommandations" sur la barre de navigation. Il choisit trois films qu'il aime et qui vont permettre de lui en recommander d'autres. Un carrousel s'affiche sur le côté droit du site où défile les films recommandés. Par défaut, les films recommandés seront des films populaires actuellement.

qui veut s'inscrire

L'internaute clique sur le bouton "Créer un compte", un pop-up s'affiche et il donne les informations nécessaires. Il remplit le captcha et un message s'affichera pour lui confirmer que son inscription est validée.

qui veut se connecter

L'internaute clique sur le bouton "Se connecter" et un pop-up s'affiche où il donne les informations nécessaires. Si l'authentification réussit, il est basculé sur la page d'accueil des utilisateurs du site.

Le cas de l'utilisateur (connecté)...

qui se connecte

L'utilisateur s'est connecté. La page d'accueil lui propose un film recommandé directement visible.

qui veut ajouter un film vu/préféré

L'utilisateur recherche un film dans la barre de recherche. Une liste de films apparaît dans la page Rechercher. Il passe sa souris sur l'affiche du film souhaité et il peut directement l'ajouter en tant que film vu, à voir ou préféré en cliquant sur les boutons correspondants.

qui se déconnecte

L'utilisateur se déconnecte en appuyant sur le bouton correspondant sur la barre de navigation.

Solution technique

Développement

Pour développer notre application Web nous utiliserons le framework Django écrit en Python. Ce choix nous semble idéal car Django permet d'uniformiser le développement d'une application Web sous un même langage. De plus, son implémentation du modèle MVT(Model-View-Template) nous permettra de structurer correctement notre projet. Pour la base de données nous avons plusieurs choix: SQLite, MySQL, PostgreSQL. Nous avons décidé d'utiliser MySQL pour plusieurs raisons: - Nous sommes déjà tous et toutes familiers avec ce logiciel - Un serveur MySQL et une interface web sous PHPMyAdmin nous est fournis Toutefois, pour les premiers tests et par souci de simplicité, nous utiliserons SQLite, car il est très simple d'utilisation et très léger. Malgré ses qualités nous l'avons pas retenu car il présente des lacunes sécuritaires certaines(impossible de chiffrer les bases de données). Nous l'utiliserons donc pour les premiers prototypes puis nous basculerons vers MySQL(procédure très simple).

Pourquoi Python ? Certains d'entre nous sont déjà familiers avec Python car nous l'utilisons dans entreprises respectives. Il est aussi relativement aisé à prendre en main donc aucun de nous ne sera particulièrement désavantagé en l'utilisant.

Déploiement

Pour le déploiement de notre application nous avons décidé d'héberger nous-mêmes notre projet en utilisant le *home-serveur* de Zakaria. Pour cela nous allons utiliser un RaspberryPi Zero W, très léger et simple à prendre en main. Le choix du serveur web est conditionné par l'utilisation de Django qui offre dizaine d'interfaces compatibles. Nous utiliserons probablement Gunicorn car son utilisation est très simple mais cela reste à confirmer. Par mesure de précaution nous allons aussi déployer notre application sur Heroku pour avoir une solution de secours.

Application technique

L'application aura une interface web. L'utilisateur pourra naviguer sur les différentes pages : la page d'accueil/home, des pages profils d'utilisateurs, des pages films, son compte personnel, un profil restreint des autres utilisateurs. Lorsque l'utilisateur clique sur un film, il aura accès à plusieurs types d'informations (synopsis, genre, durée, acteurs ...) et ainsi voir les bandes annonces des films. L'utilisateur pourra également faire des recherches plus poussées. Il aura à disposition une barre de recherche et pourra affiner ses recherches avec l'aide de filtre. Notre application fera appel à une base de données. Cette dernière contiendra toutes les informations relatives aux films et aux utilisateurs de notre interface web.

Les utilisateurs pourront aussi noter les films et ainsi affiner leurs recommandations, ce qui signifie que l'application, grâce à un algorithme de recommandation de film, propose des films qui pourraient plaire à

l'utilisateur. Cet algorithme permettra de recommander des films aux utilisateurs basé sur deux critères : les films qu'ils ont appréciés et les genres/thèmes qu'ils ont bien notés ou le plus regardés.

Un utilisateur peut suivre un autre utilisateur pour voir ses informations et avoir les recommandations qu'il a eu.

Algorithme de recommandation

L'algorithme de recommandation est la fonctionnalité la plus importante de l'application web. Elle doit permettre à n'importe quel internaute de se voir recommander des films en fonction des films populaires actuellement(internaute) ou des films qu'il a vu et préférés(utilisateur connecté).

Schématiquement, il y a deux principaux algorithmes de recommandation: - Algorithme de recommandation en fonction du **contenu** - Algorithme de recommandation par **rapprochement d'utilisateur**

Recommandation en fonction du contenu

On extrait de chaque film aimé/regardé par un utilisateur les genres auxquels ils appartiennent, les acteurs qui y ont joués, le réalisateur qui a dirigé le film. À partir de ces données, on sélectionne les films ressemblants et on propose à l'utilisateur le film ayant la note la plus élevée.

Recommandation par rapprochement d'utilisateur

Le principe est le même sauf qu'au lieu de trouver des films similaires on trouve des **utilisateurs** similaires. Disons que l'utilisateur A peut être rapprochés aux utilisateurs B,C et D car ils partagent le même genre, la même tranche d'âge ou les mêmes goûts. Dans notre cas, l'algorithme va tenter de trouver les films les plus populaires parmi ces 3 utilisateurs et va les proposer à l'utilisateur A.

Notre choix

Dans l'idéal, il est possible de lier les deux algorithmes (c'est le cas sur les plateformes comme Netflix) mais pour commencer nous allons implémenter la recommandation en fonction du **contenu**. Ce sera l'implémentation minimale. Par la suite, nous implémenterons le 2ème algorithme et lierons les deux.

Technologies

Nous utiliserons la librairie **surprise, numpy et pandas**. [Surprise](#) est spécifiquement conçue pour les systèmes de recommandation de contenu. La plupart des algorithmes les plus courants(à base de factorisation de matrices Utilisateurs * Films, méthodes des voisins) sont déjà implémentés. Nous utiliserons la méthode de factorisation des matrices Utilisateurs * Films car la documentation à son sujet nous paraît plus dense et riche.

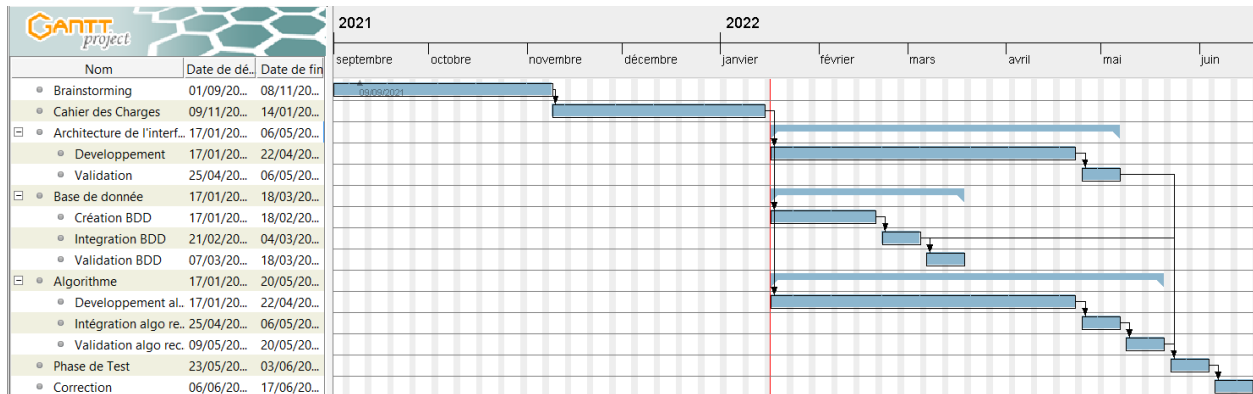
Données

Les données que nous utiliseront proviennent du [laboratoire MovieLens](#) de l'Université du Minnesota. En particulier, celles qui regroupent les avis, notes et films préférés de leurs utilisateurs. Ce qui nous permettra de remplir facilement notre base de données. Du côté films, ils fournissent également une [base de données](#) de plusieurs dizaines de milliers de films avec leurs genres.

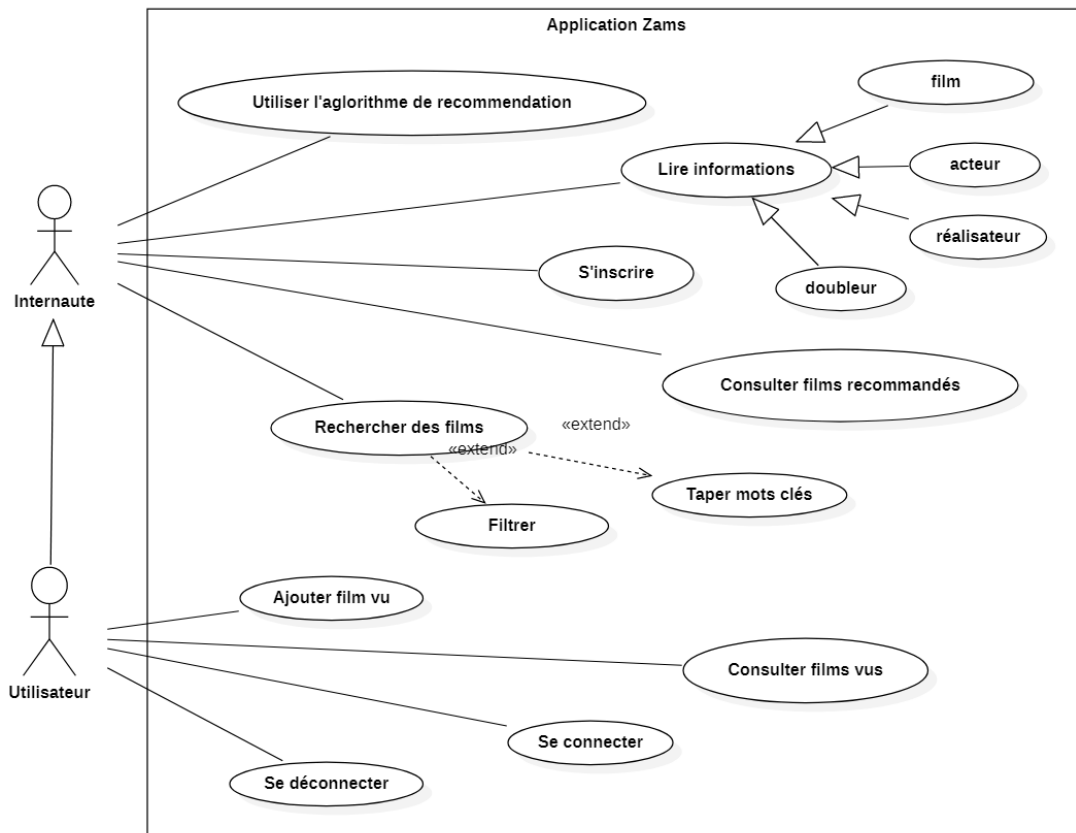
Pour le deuxième algorithme, nous utiliserons le [dataset d'IMDB](#) qui permet d'obtenir les acteurs et les réalisateurs d'un film.

Diagrammes

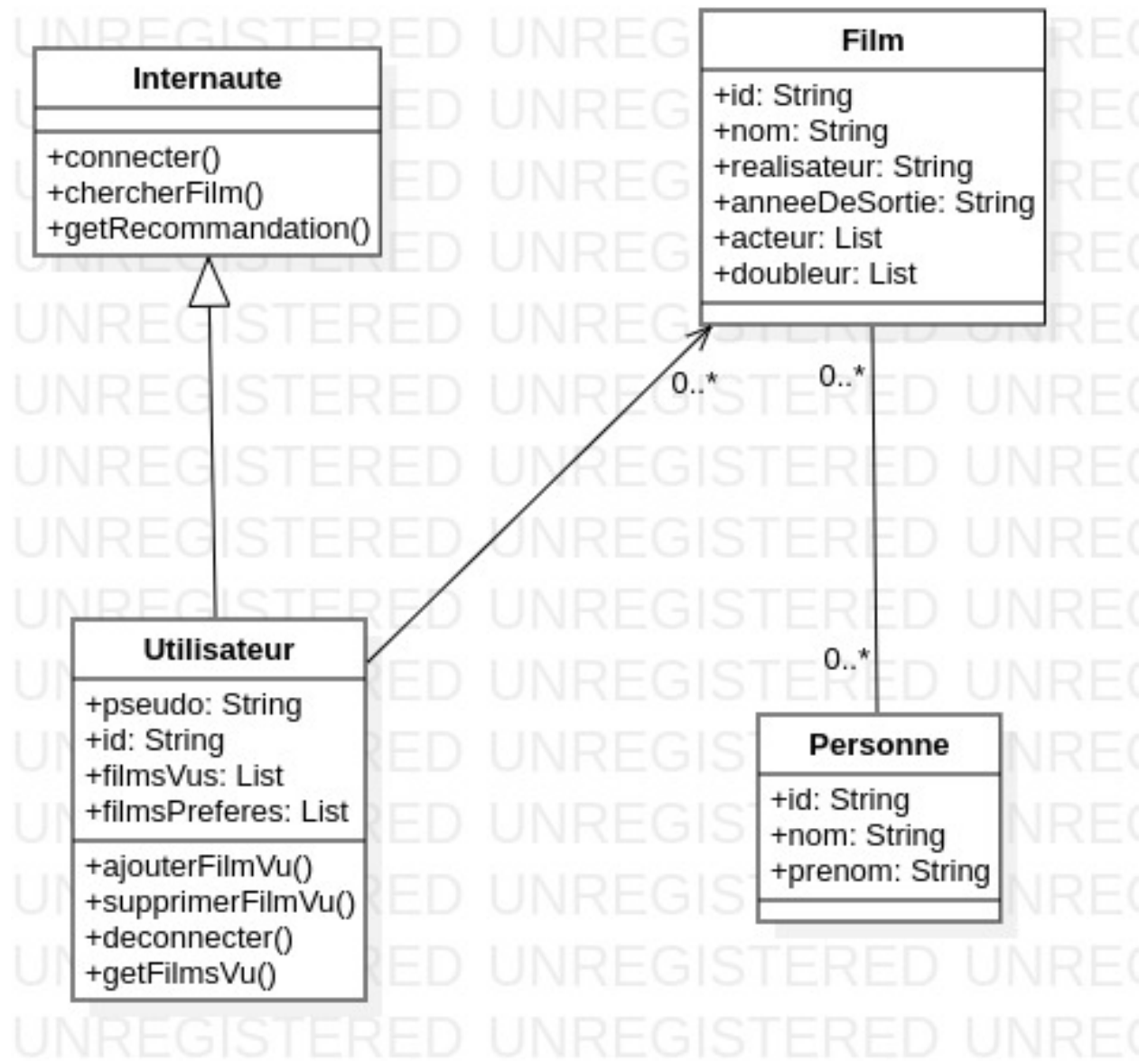
Gantt



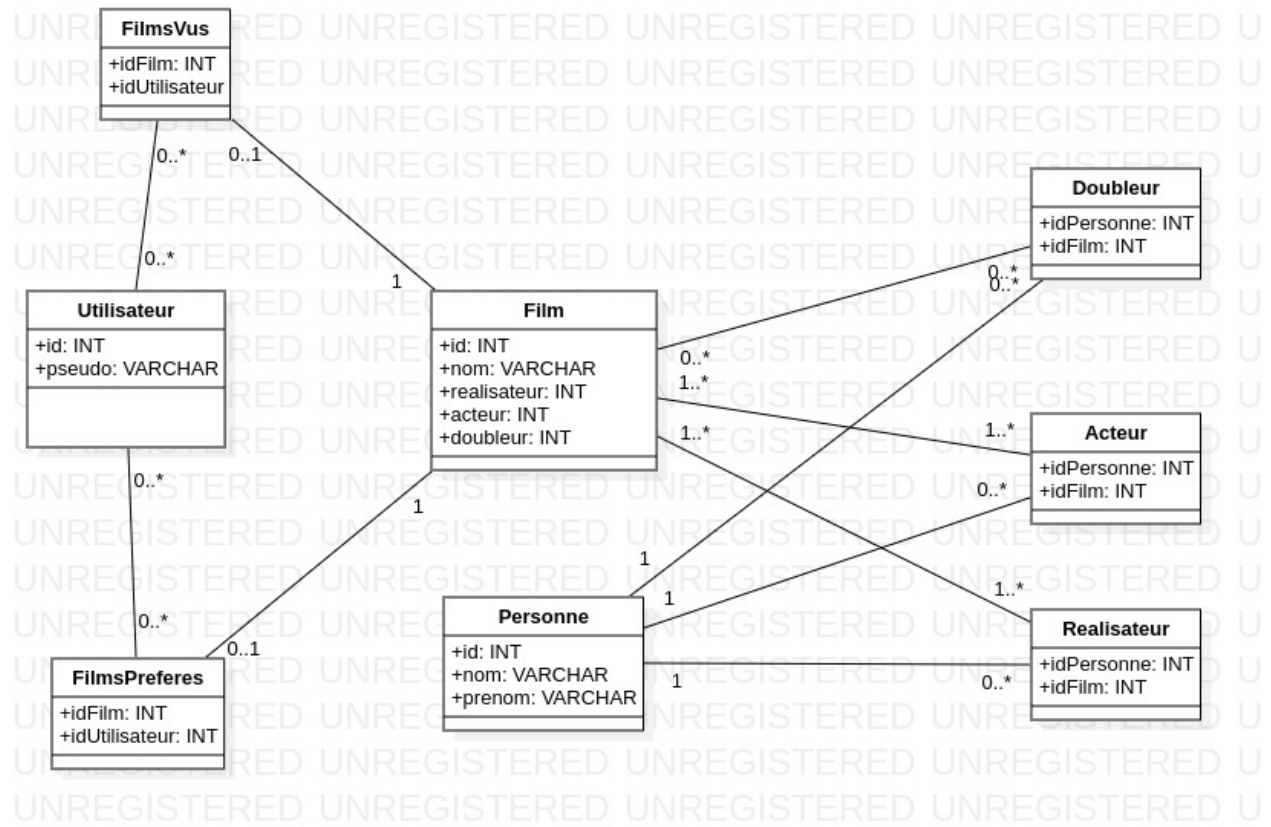
Cas d'utilisation



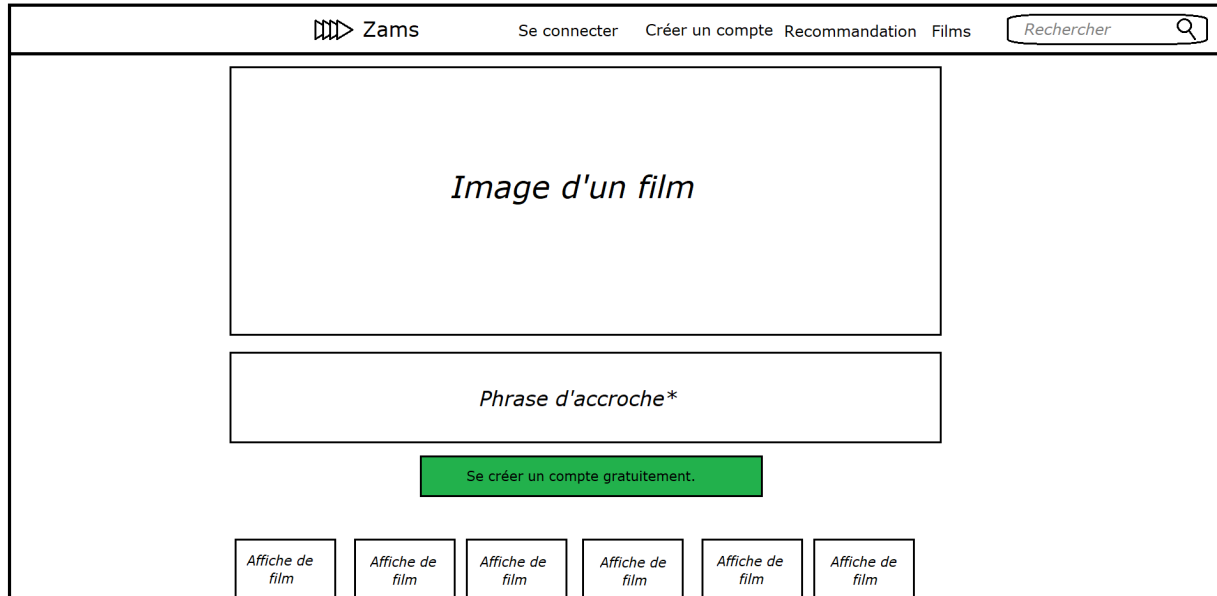
Classes



Base de données



Wireflows



* Description de ce que fait le site genre : "Dites les films que vous aimez et nous allons vous en conseillez d'autres."
"Et si vous regardiez ce film ce soir ?"
"Renseignez les films que vous avez vu."
Laissez nous faire le reste."

FIGURE 1 – Accueil: Internaute (pas connecté)

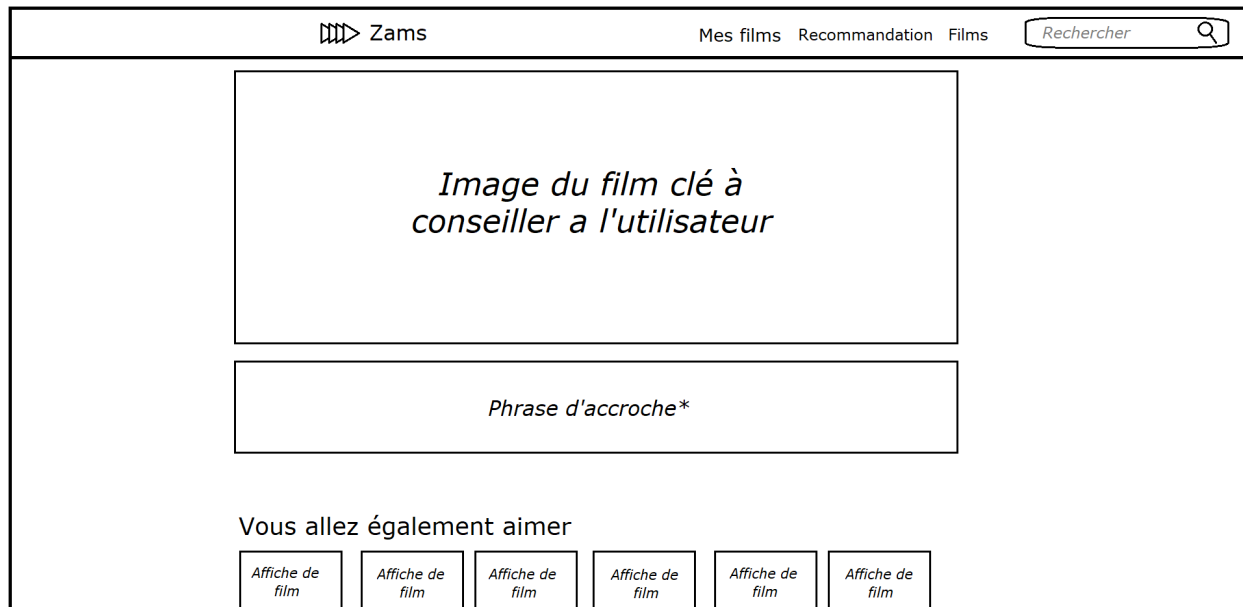


FIGURE 2 – Accueil: Utilisateur (connecté)

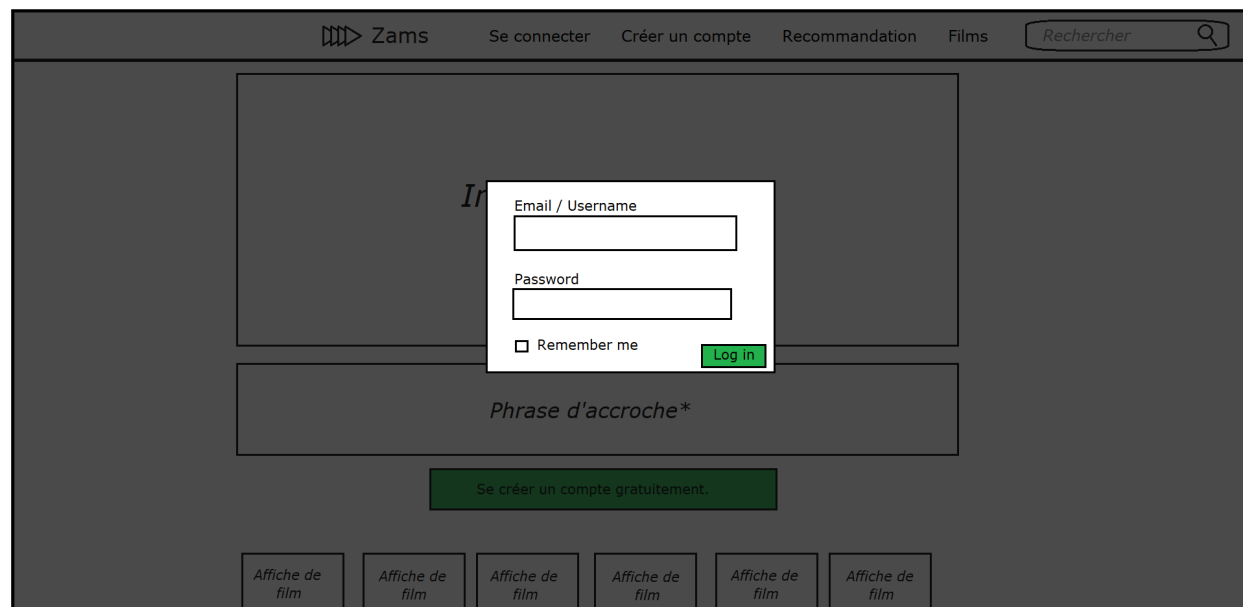


FIGURE 3 – Login

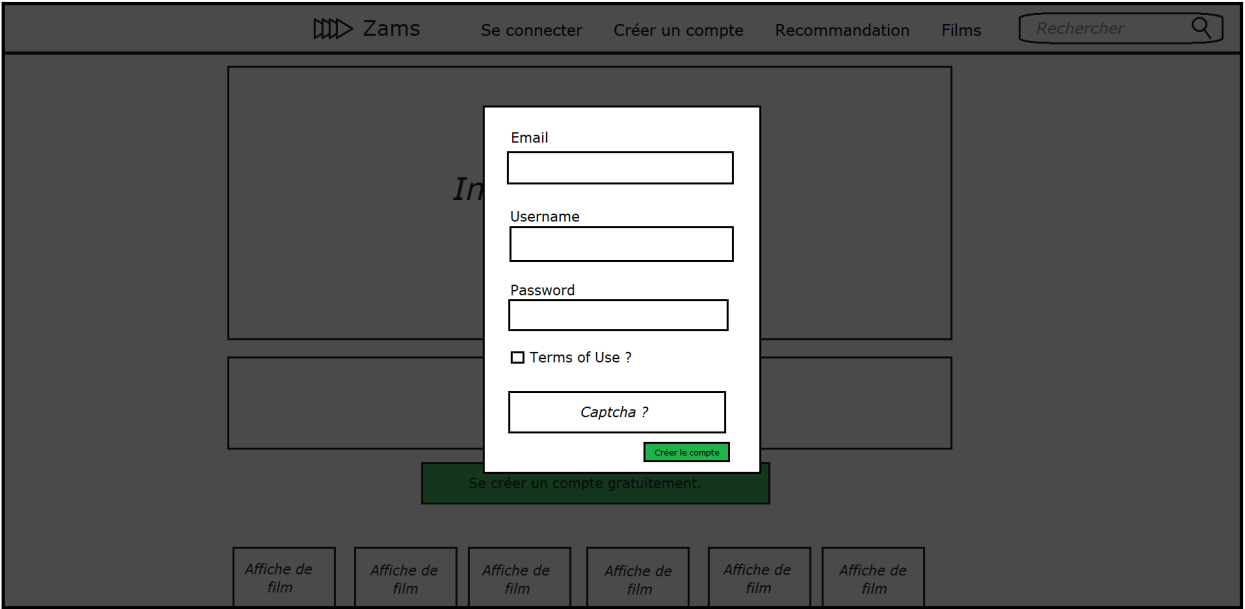


FIGURE 4 – Register

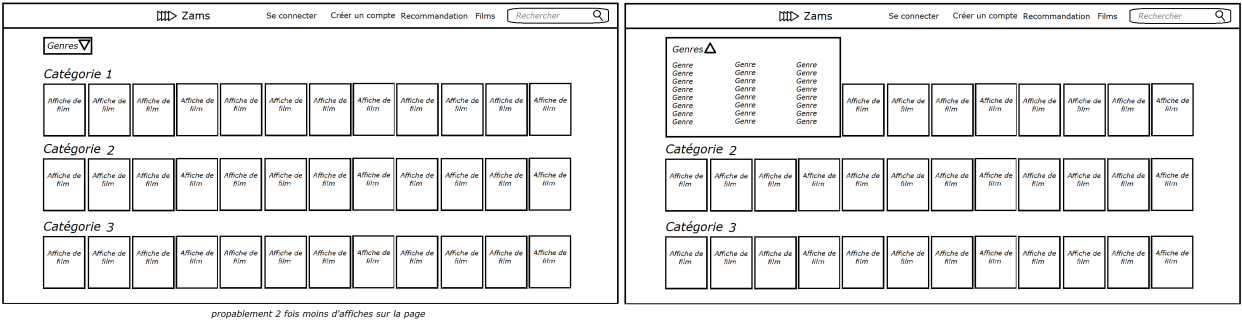


FIGURE 5 – Page Films

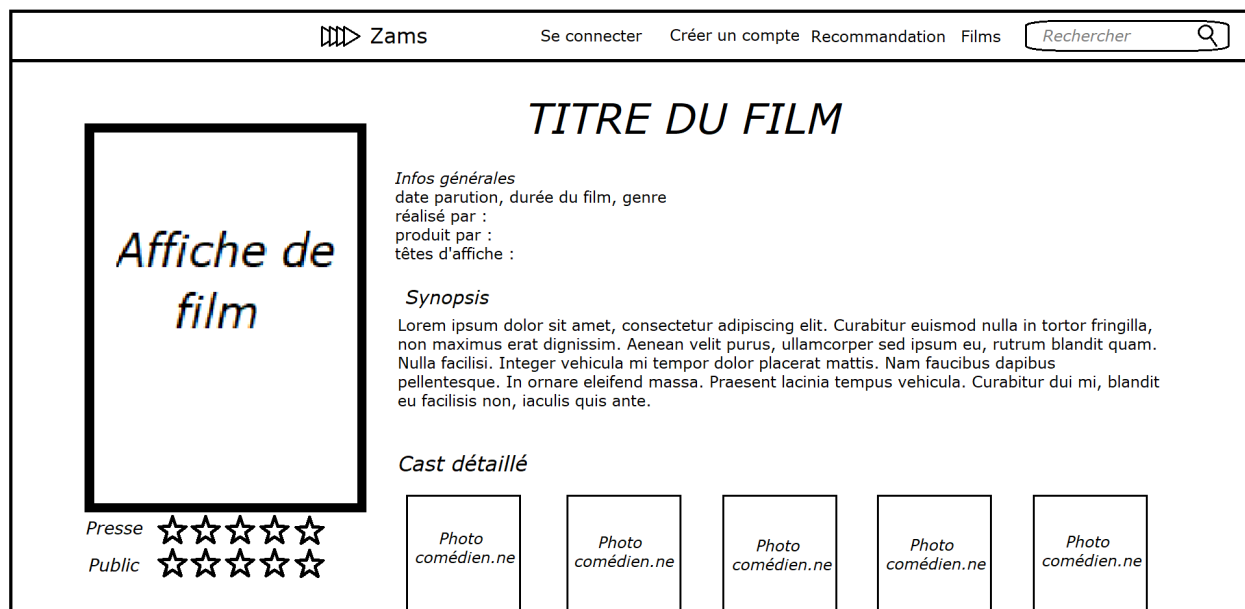


FIGURE 6 – Page d'un film

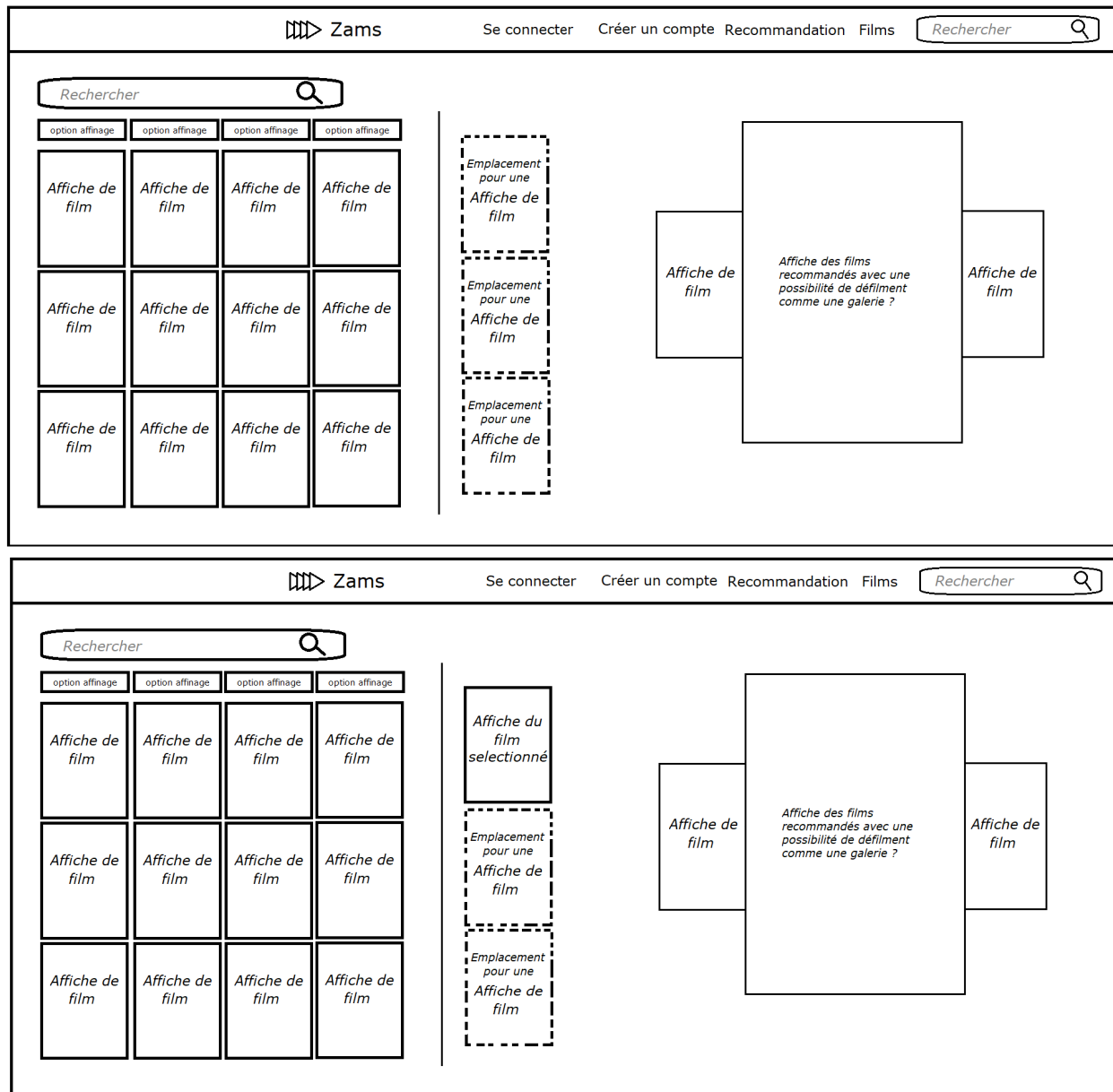


FIGURE 7 – Recommandation



FIGURE 8 – Survoler une affiche

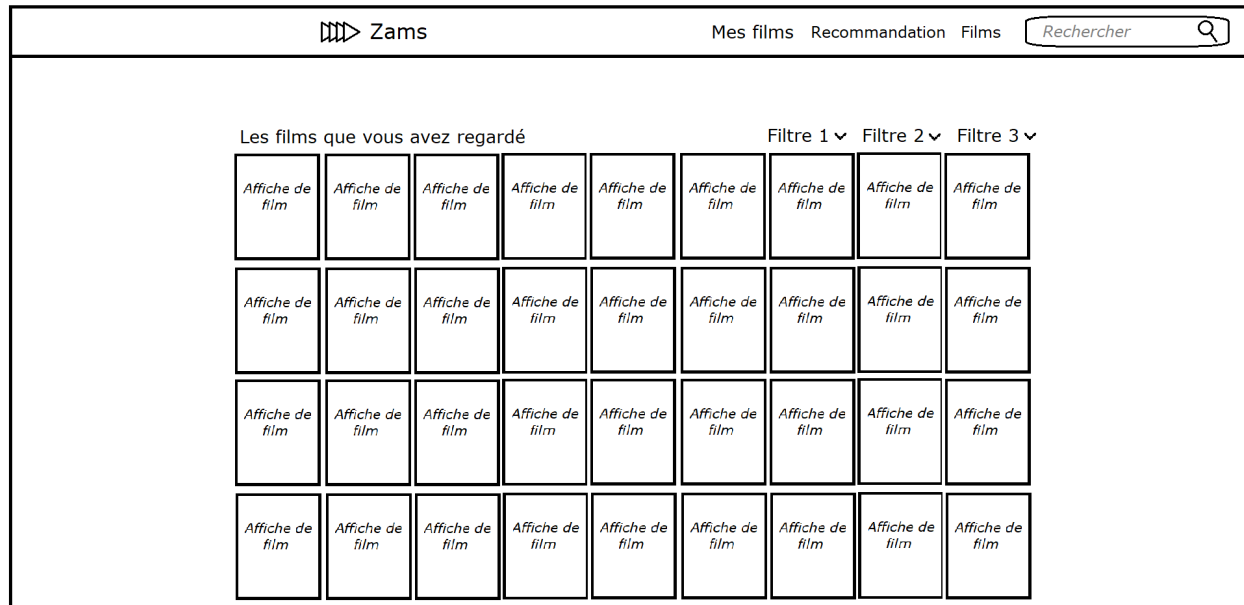


FIGURE 9 – Mes Films (connecté)

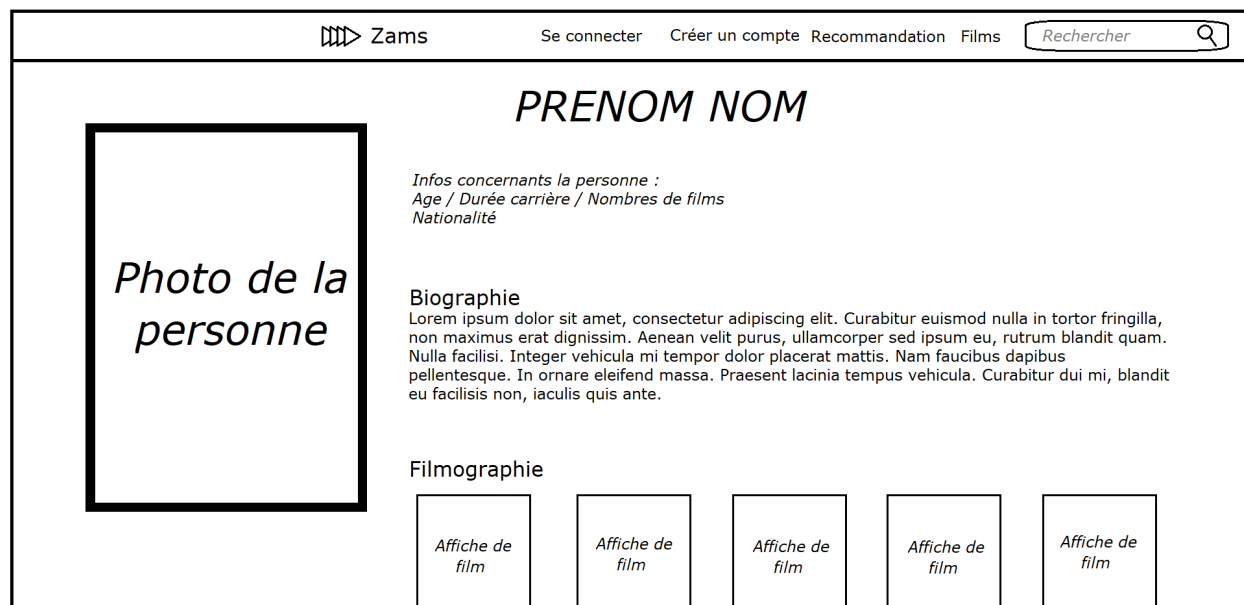


FIGURE 10 – Page d'une personne