Solution finale projet Java
Projet TIA

Ronan Abhamon, Florian Bigard, Kevin Hivert et Nicolas Reynaud

2014

Sommaire

1 Introduction

Nous avions comme idée première de faire un site à la YouTube. Après réflexion et suite aux 6 projets à réaliser en moins d'un mois grâce à une administration particulièrement performante, nous nous sommes mis d'accord de ne mettre que des liens de vidéos (permettant de les héberger sous DailyMotion, Youtube ou autre...).

Dans ce rapport, nous étudierons la conception du projet (client lourd, client léger). Cette partie comprendra différents schémas et explications. Nous ferons ensuite une description des fonctionnalités que nous avons intégrées à notre projet. Pour réaliser ces fonctionnalités, nous avons dû utiliser différentes technologies dont nous expliquerons l'intégration dans la partie suivante. Nous parlerons ensuite des différents codes inspirés d'Internet qui nous ont aidés sur certaines parties. Nous finirons par une estimation du temps de chaque partie, puis à une conclusion qui comprendra un bilan critique et autres conneries demandées dans le sujet.

2 Description générale de l'application

Le but de notre projet est de proposer une interface Web permettant l'échange de vidéos au sein d'une communauté. Le système part d'un concept simple : l'Internet regorge de vidéos en tout genre, pourquoi ne pas centraliser celles dignes d'intérêt ?

Cette communauté étant un ensemble de membres, nous avons donc créé un espace pour chaque membre. Qui dit espace dit aussi :

- compte
- inscription
- connexion

Ces membres peuvent partager des vidéos de plateformes de médias comme par exemple YouTube (et non pas les télécharger de leurs ordinateurs) sur notre application. Le fait de partager n'est pas suffisant, ils peuvent aussi émettre une critique sur ces médias : un système de votes est donc important. Pour renforcer ce système, la mise en place de commentaires nous a semblé aussi très utile.

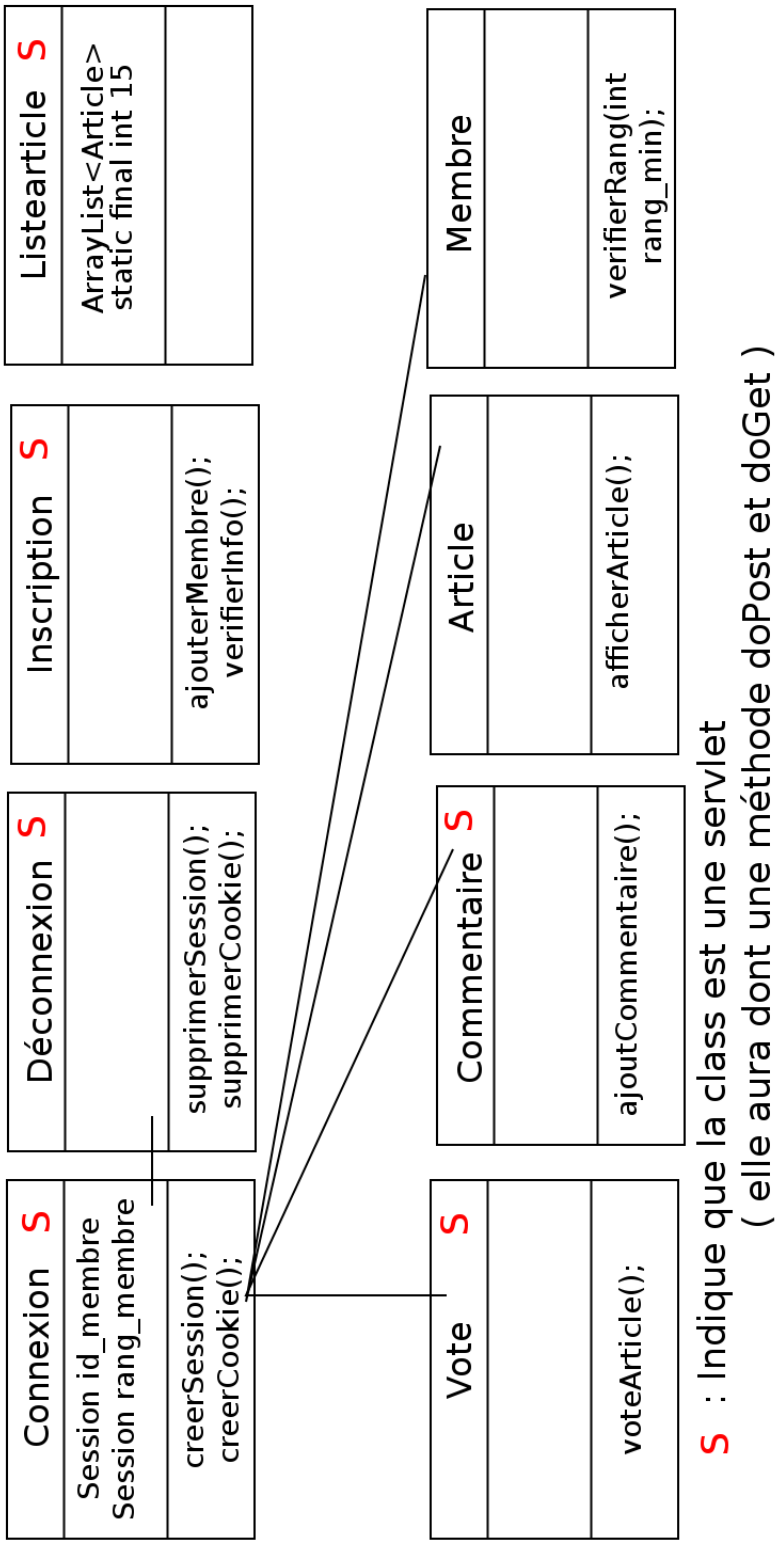
Ces membres doivent donc interagir sur notre application, celle-ci est constituée de 2 environnements :

- Un site internet léger pour accéder aux vidéos, commentaires et autres informations.
- Un client lourd, soit un programme permettant d'accéder aux mêmes données que le site internet lui-même.

Concernant les vidéos, elles peuvent être intégrées dans des playlists. Elles peuvent donc être échangées entre les membres de cette communauté. Les vidéos n'étant rien d'autre que des urls dans notre base de données, une simple recherche de mots clefs peut mener à un ensemble de vidéos. Une fonction recherche est donc indispensable. Enfin les utilisateurs ont la possibilité de trier ces vidéos par date, nombre de votes ou nom.

Enfin, un tchat permet aux membres de communiquer entre eux. S'ils sont tous deux connectés, ils communiquent directement à l'aide de sockets. Sinon ils communiquent grâce au serveur qui stocke les messages en attendant que l'autre personne se connecte et lise ses propres messages.

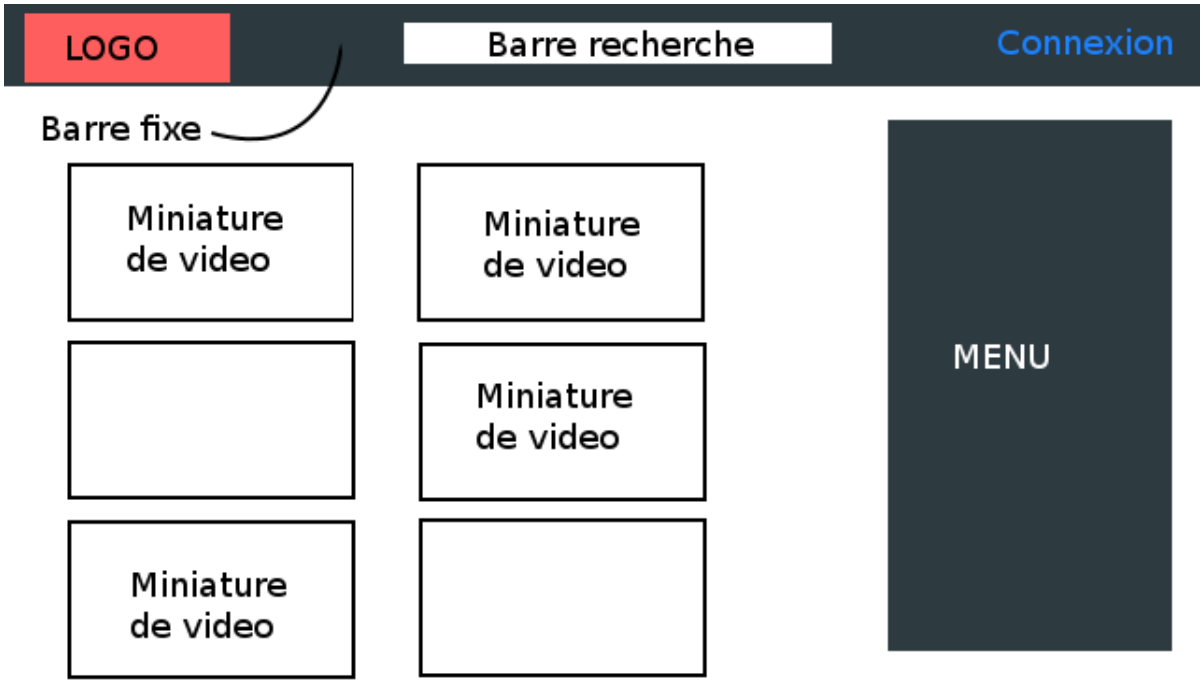
3 Conception



4 Plan du site

4.1 Pages principales

Design des pages principales



4.1.1 Accueil

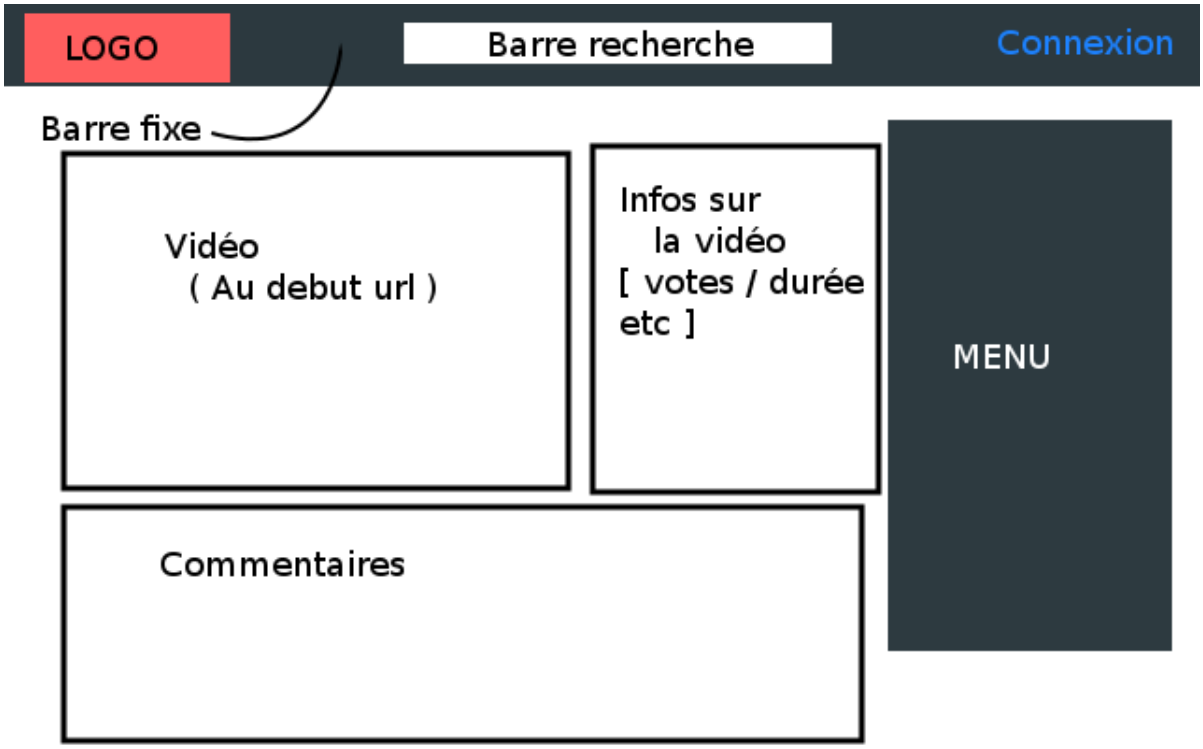
La page d'accueil contient les miniatures des vidéos qui ont déjà été postées. Celles ci sont triées par date de publication de la plus récente à la plus ancienne. Il s'agit de la première page visible sur le site à l'arrivée d'un visiteur.

4.1.2 Top - Flop

Les pages Top et Flop ont pratiquement la même fonction que la page d'accueil, cependant la page Top trie les vidéos par nombre de votes décroissants [des plus appréciées aux moins appréciées]. La page Flop quant à elle trie dans le nombre croissant de votes [Des moins appréciées aux plus appréciées].

4.2 Pages associées à une vidéo

Design des pages d'affichage de vidéos :



Sur cette page, un membre peut, mettre en favoris la vidéo, voter pour cette vidéo et la commenter.

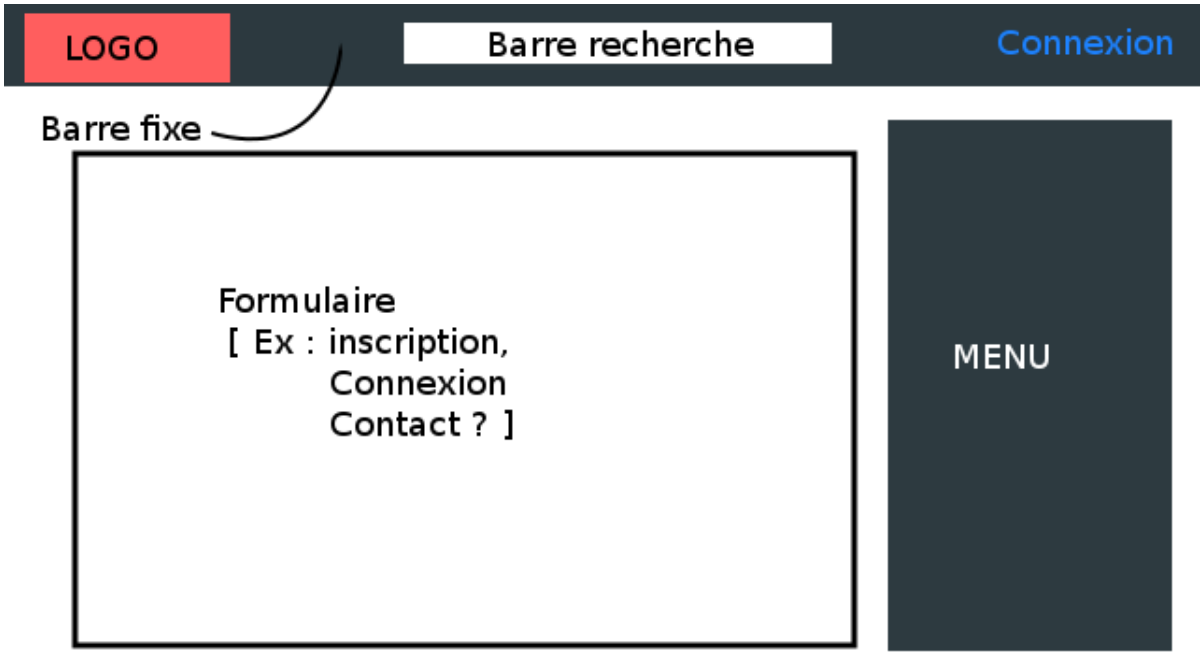
4.2.1 Visualisation d'une vidéo

4.2.2 Random

La page random donne une vidéo de façon aléatoire parmi toutes celles disponibles sur le site.

4.3 Pages secondaires - Formulaires

Design des pages secondaires :



4.3.1 Inscription

Cette page permet à un visiteur de s'inscrire pour qu'il devienne membre de la communauté.

4.3.2 Connexion

La page connexion permet aux membres du site de se connecter en passant leur statut de visiteur à membre.

4.3.3 Déconnexion

La page déconnexion a pour but de déconnecter une personne si elle est connectée.

4.3.4 Poster article

Cette page permet simplement aux membres de poster un article.

5 Description des fonctionnalités

Notre projet propose donc un certain nombre de fonctionnalités, ces dernières étant identiques sur chaque support comme il l'était stipulé dans l'énoncé du projet.

Les utilisateurs étant enregistrés dans une base de données la première fonctionnalité est bien entendu l'inscription, la connexion et la déconnexion d'un membre. Ainsi les vidéos, commentaires etc... peuvent être attribués à un membre précis de la communauté.

Les membres ainsi que les visiteurs peuvent chercher et visionner par la suite des vidéos, les recherches se font par rapport aux titres des vidéos mais 4 types de tri sont possibles :

- Popularité décroissante : Les vidéos ayant le plus fort indice de popularité sont donc affichées en haut de la liste.
- Popularité croissante : Ici ce sont donc les vidéos ayant le plus faible indice de popularité qui sont affichées en premier.
- Date décroissante : Les vidéos sont donc affichées de la plus récente à la plus ancienne, c'est un bon moyen pour voir les nouveautés.

– Date croissante : Les vidéos sont triées de la plus ancienne à la plus récente, un tri pour les plus nostalgiques donc.

Le visionnage d'une vidéo affiche le titre de la vidéo, la popularité, la description ainsi que la liste des commentaires. Le client léger affiche la vidéo si le lien mène bien vers une vidéo valide, le client lourd lui, permet d'accéder à la vidéo sur internet grâce à son lien.

Les membres connectés ont, comme nous l'avons vu, la possibilité de regarder les vidéos, tout comme les visiteurs non enregistrés mais contrairement à ces derniers les membres ont la possibilité de pouvoir voter, commenter et ajouter une vidéo à leurs favoris.

Les votes nous permettent par la suite de créer un classement par popularité, un vote pouvant être positif ou négatif. Chaque vote ajoute ou enlève 1 point à la popularité de la vidéo.

Les commentaires permettent aux membres d'exprimer leurs avis, de discuter et d'échanger autour d'une vidéo.

Les membres peuvent également ajouter une vidéo à leur playlist personnel, nous décrirons ce point juste après.

Les membres possèdent donc chacun une playlist, à laquelle ils peuvent ajouter des vidéos afin de les conserver et les retrouver plus facilement.

Cette playlist peut être exportée sous format XML afin d'être partagée avec les autres membres de la communauté, au tout autre but que pourrait trouver l'utilisateur.

Les membres ont évidemment la possibilité d'ajouter des vidéos et ainsi les partager avec le reste de la communauté (pas celle de l'anneau). Lorsqu'un membre ajoute une vidéo on lui demande le lien vers une vidéo (Youtube ou autre). Il lui est également demandé un lien vers une miniature à afficher, ainsi qu'un titre et une description de la vidéo.

Nous voulions ajouter la possibilité de modifier les vidéos, cependant le manque de temps aura eu raison de cette fonctionnalité. Une autre fonction qui n'a pas pu voir le jour est un système de rang par lequel des administrateurs et des modérateurs auraient pu modifier voir supprimer certaines vidéos si le contenu n'était pas convenable.

6 Technologies mises en oeuvre

6.1 Client lourd

6.1.1 Tchat

Pour la réalisation du Tchat, nous avons dû nous servir des sockets couplés à des threads, comme lors du TP2 (serveur convertisseur).

6.1.2 Swing

Swing a permis de construire une interface fonctionnelle. Cependant Swing contraint à utiliser des Components pouvant vite devenir incontrôlables d'où l'utilisation d'une classe http ://WrapLayout pour fixer facilement certains éléments sensibles. Nous pensons notamment aux miniatures des vidéos.

6.1.3 Singleton

Nous avons aussi créé une classe singleton nous permettant de stocker les informations du membre qui utilise le client lourd. Nous pouvons donc ensuite récupérer simplement les différentes informations (id, ip, pseudo...).

6.1.4 Template

La partie Swing du projet dispose de plusieurs classes comme les aperçus des vidéos découlant de templates (Patron de méthode). Cela permet de définir un comportement global de classe mais permet aussi de définir des interactions différentes pour les classes héritantes.

6.2 Client léger

6.2.1 Jsp

Les jsp font partie intégrante de cette partie, j'ai également pu également utiliser les inclusions que celle ci fournissait afin de bien séparer chaque partie du projet.

6.2.2 Servlet

J'ai pu utiliser les Servlets, lequel m'on permis de générer les XML afin de passer les données au client Lourd, en fonction des données qui m'était envoyé par le client lourd.

6.2.3 SQL

Bien entendu, j'ai du utiliser le SQL afin de faire mes requetes sur la base de données. Ce qui implique également d'utiliser les connexion depuis Java, notamment a l'aide de la librairie externe : "jdbc".

7 Codes inspirés d'Internet

7.1 Partie réseau

Pour la réalisation de l'API, nous avions besoin de communiquer avec le serveur via des requêtes HTTP (POST et GET). Cette partie fut un peu plus délicate, il nous a fallu aller chercher sur StackOverflow une aide pour la réalisation de ce type de requête. Nous n'irons pas vous cacher notre étonnement lorsque nous nous sommes rendus compte que chacune des méthodes faisait dans les 100 lignes.

8 Estimation du temps

8.1 Client léger

Le temps passé sur le client léger est très conséquent pour ma part, (Reynaud Nicolas) , ayant travaillé seul sur ce dernier , j'y ai passé mes nuits et mes week-end, de même que pour le design et la gestion de la Bdd. Pour le tout, je pense être à plus de 60 heures de boulot. [Par exemple travail du week end de 3 jours que nous avons eu : 19 au 21 ; le 19 : 18h00 - 4h30 ;le 20 : 10h00 - 22h00 ; le 21 : 10h00 : 0h00].
Ainsi le temps consacré est vraiment très conséquent.
Cependant j'ai pu apprécier la modularité, si un changement m'était demandé, par exemple sur une requête SQL, je n'avais qu'une petite modification à faire.

8.2 Client lourd

Le temps consacré au client lourd est conséquent, le temps de mise en place de l'interface graphique ainsi que des parsers XML doit bien atteindre les 40h. C'est assez important mais il existe une explication : Faire une interface accessible sans être repoussante, ainsi que positionner correctement les éléments, ce n'est pas simple. D'autant plus de le faire proprement.

8.3 Tchat

Nous avons dû compter la bagatelle d'environ 30 heures pour réaliser la partie conception et mécanique du Tchat. Nous comptons dans ce nombre d'heures l'intégration à l'interface du client lourd. La conception fut assez difficile à mettre en oeuvre, car il nous a fallu manipuler des Thread couplés avec des Sockets ce qui n'était pas évident. De plus la création de tests fut encore une fois difficile dans le sens où nous ne pouvions lancer deux fois le logiciel sur un seul ordinateur étant donné que celui-ci écoutait sur un port précis.

9 Ressources utilisées

Comme indiqué dans la partie Swing, une unique ressource a été utilisée telle quelle pour l'interface : http ://WrapLayout

La majorité du temps StackOverflow a sauvé le projet.

10 Conclusion

Les interfaces et menus prévus ont été effectués dans les temps. Cependant Swing est assez limité pour concevoir rapidement une interface complexe. (A côté il existe d'un côté GTK mais le temps de dev explose littéralement avec cette lib, et d'un autre Qt où il est possible de faire des fenêtres en utilisant du CSS.)

Pour conclure, nous avons du simplifier certaines idées que nous avions afin de pouvoir tenir les délais de plus nous pensons sincèrement que la JAVA n'est pas un langage adapté pour un projet WEB. Il existe des technos plus récentes et plus puissantes qui nous auraient permis de faire plus dans le même temps imparti, le projet en aurait été que plus intéressant et plus agréable, mais puisqu'il faut bien se plier aux règles nous avons fait de notre mieux pour fournir une applications décente et pas trop désagréable à l'oeil.