

# Rapport de conception intermédiaire

## Projet TIA

Ronan ABHAMON, Florian BIGARD, Kevin HIVERT et Nicolas REYNAUD

2014

# Sommaire

<b>1</b>	<b>Description générale de l'application</b>	<b>3</b>
<b>2</b>	<b>Modélisation</b>	<b>4</b>
<b>3</b>	<b>Plan du site</b>	<b>5</b>
3.1	Pages principales . . . . .	5
3.1.1	Accueil . . . . .	5
3.1.2	Top - Flop . . . . .	5
3.2	Pages associées a une vidéo . . . . .	6
3.2.1	Visualisation d'une vidéo . . . . .	6
3.2.2	Random . . . . .	6
3.3	Pages secondaires - Formulaire . . . . .	7
3.3.1	Inscription . . . . .	7
3.3.2	Connexion . . . . .	7
3.3.3	Déconnexion . . . . .	7
3.3.4	Poster article . . . . .	7
<b>4</b>	<b>Utilisation des technologies</b>	<b>7</b>
4.1	HTML5/CSS3 . . . . .	7
4.2	JavaScript/JQuery . . . . .	8
4.3	Java/Servlet . . . . .	8
4.4	SQL . . . . .	8
4.5	XML . . . . .	8
4.6	Java/Swing . . . . .	8
4.7	Sockets TCP . . . . .	8
<b>5</b>	<b>Répartition du travail et désignation d'un chef</b>	<b>9</b>
5.1	Chef d'équipe . . . . .	9
5.2	Répartition du travail . . . . .	9

# 1 Description générale de l'application

Le but de notre projet est de réaliser 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 digne d'intérêt ?

Il y va donc de plusieurs principes :

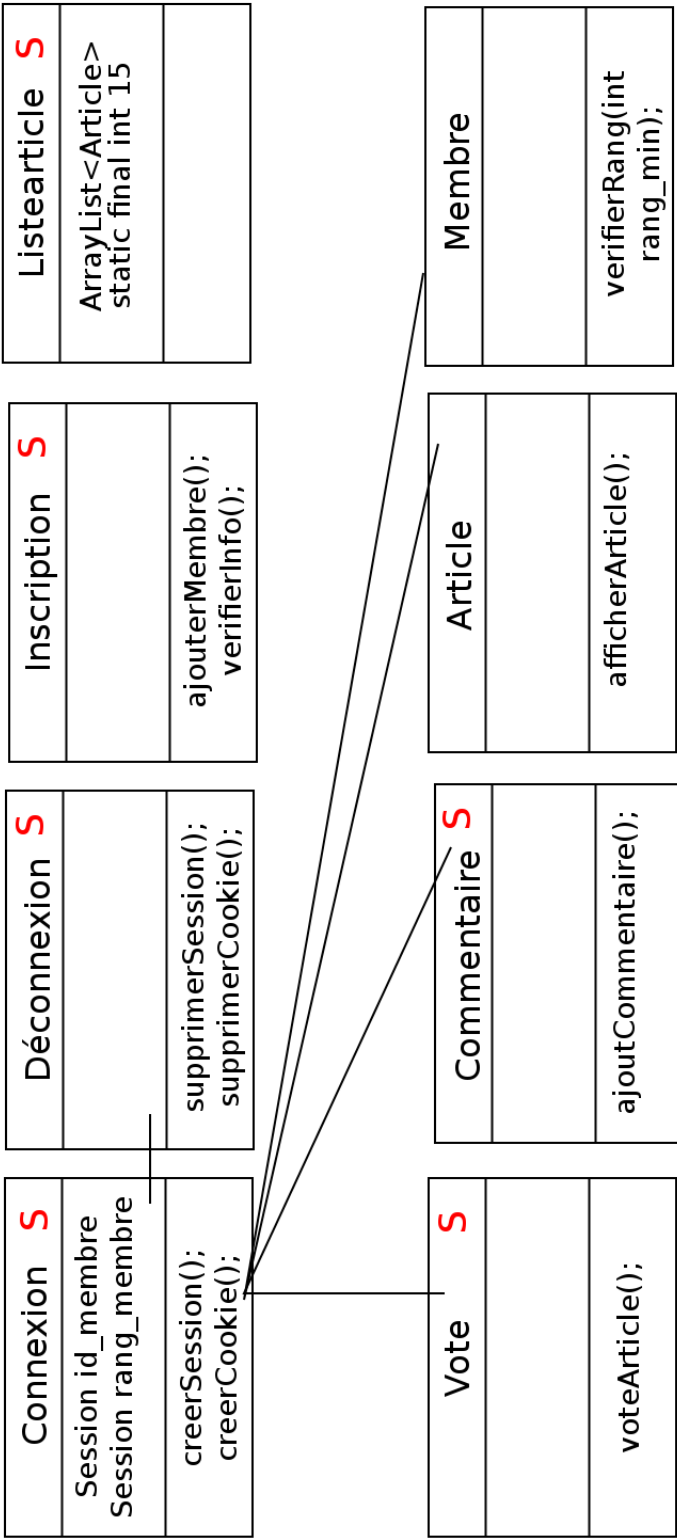
Cette communauté étant un ensemble de membres, nous prévoyons de créer un espace pour chaque membre. Qui dit espace dit aussi : compte, inscription et connexion évidemment. Ceux-ci doivent pouvoir partager des vidéos de plateformes de médias comme par exemple Youtube (et non pas les télécharger de son ordinateur) sur notre application. Le fait de partager n'est pas suffisant, il faut pouvoir aussi émettre une critique sur ces médias, un système de votes est donc plus qu'envisageable. Pour renforcer ce système, la mise en place de commentaires nous semble 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 pourront être intégrées au sein de playlists. Elles devront par ailleurs pouvoir échanger 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 doit pouvoir mener à un ensemble de vidéos, une fonction recherche est donc indispensable. Enfin les utilisateurs auront la possibilité de trier ces vidéos par date, nombre de votes ou tout autre système que nous jugerons utile.

## 2 Modélisation

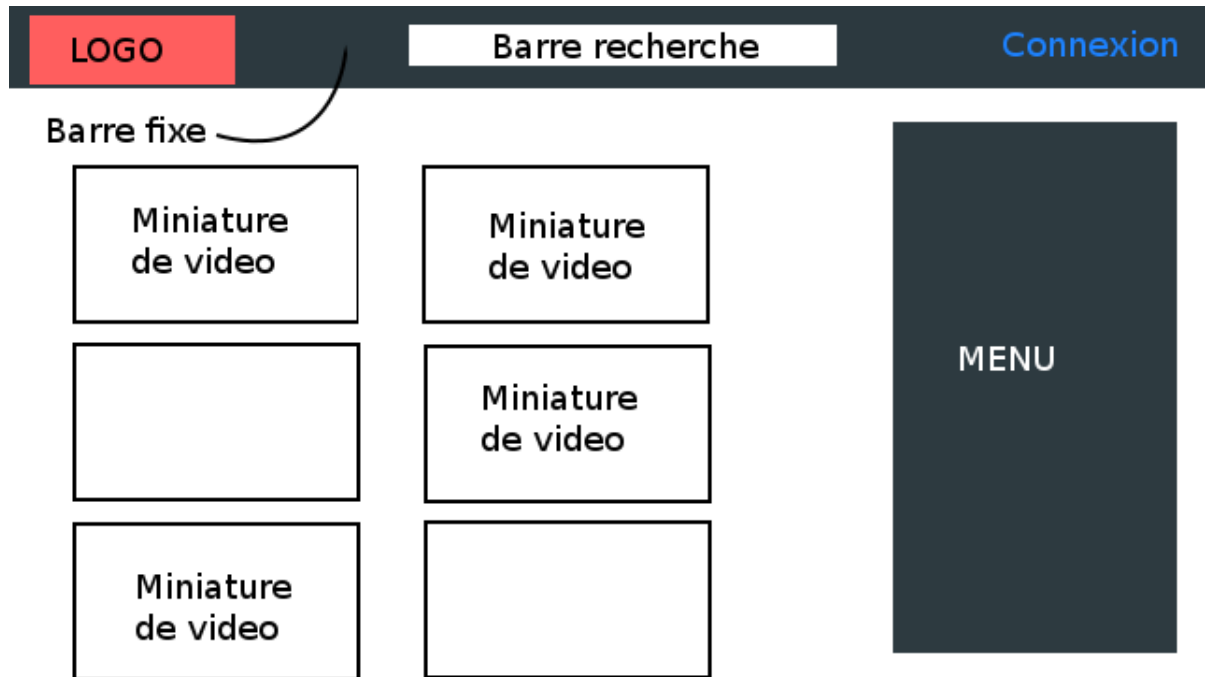


**S** : Indique que la class est une servlet  
( elle aura donc une méthode doPost et doGet )

### 3 Plan du site

#### 3.1 Pages principales

Design des pages principales



##### 3.1.1 Accueil

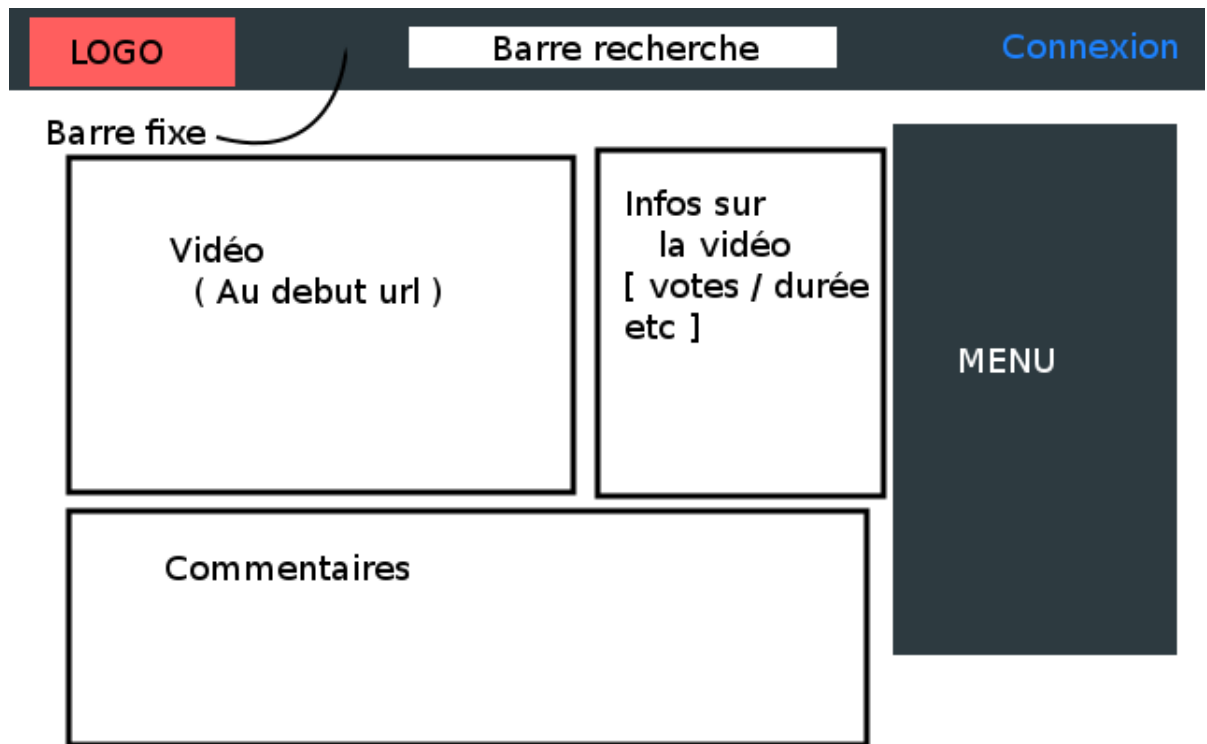
La page accueil contiendra les miniatures des vidéos qui ont déjà été postée. Celle ci seront triée par date de publication, de la plus récente a la plus vieille. Il s'agit de la première page visible sur le site à l'arrivée d'un visiteur.

##### 3.1.2 Top - Flop

Les pages Top et Flop auront pratiquement la même fonction que la page accueil, cependant la page Top triera les vidéos par nombre de vote croissant [ des plus appréciées aux moins appréciées ]. Et la page Flop dans le nombre décroissant de votes [ Des moins appréciées aux plus appréciées ].

## 3.2 Pages associées a une vidéo

Design des pages d'affichage de vidéos



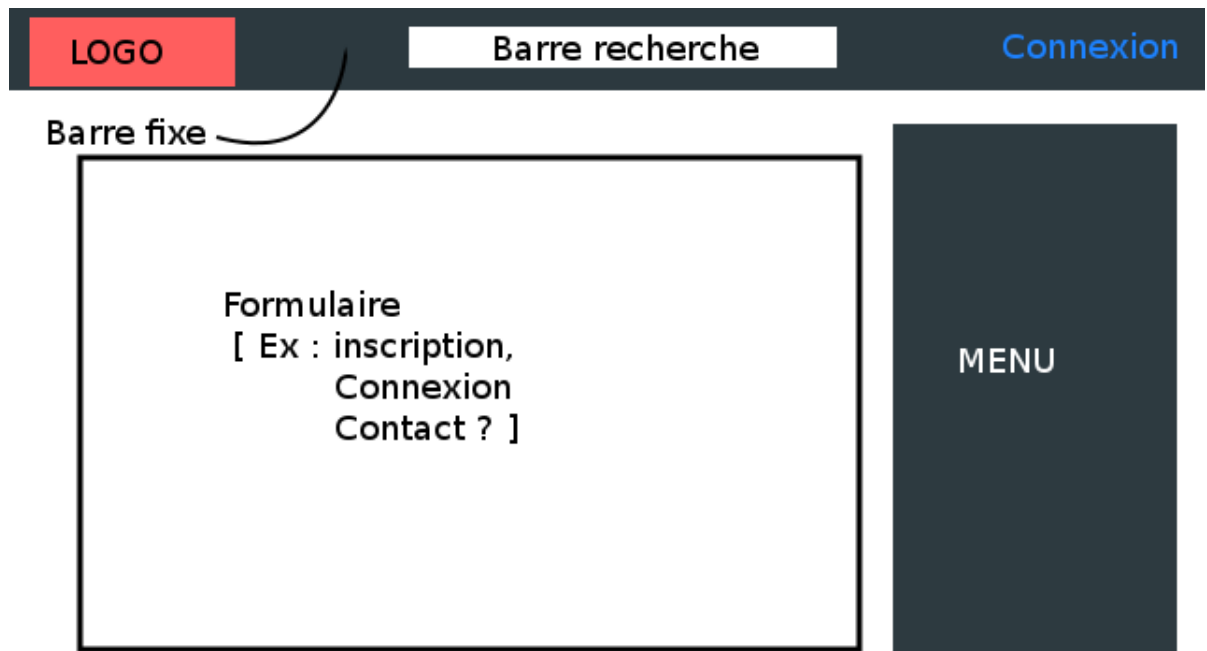
### 3.2.1 Visualisation d'une vidéo

### 3.2.2 Random

La page random donnera une vidéo de façon aléatoire parmi toutes celles disponible sur le site.

### 3.3 Pages secondaires - Formulaires

Design des pages secondaires



En adaptant la taille du cadre formulaire bien entendu

#### 3.3.1 Inscription

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

#### 3.3.2 Connexion

La page connexion permettra aux membres du site de se connecter passant leurs status de visiteur a membres.

#### 3.3.3 Déconnexion

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

#### 3.3.4 Poster article

Cette page permettra simplement aux membres de poster un article.

## 4 Utilisation des technologies

### 4.1 HTML5/CSS3

Le client léger étant un navigateur Web, il faut que notre serveur affiche des pages Web, et donc HTML. Nous avons choisi de combiner HTML5 et CSS3 afin de profiter des dernières avancées dans le secteur du Web et ainsi empêcher les utilisateurs d'Internet Explorer d'accéder à notre site. De plus, à terme nous prévoyons d'afficher directement les vidéos (au lieu de juste un lien YouTube), il sera donc intéressant d'exploiter la balise *video* d'HTML5.

### 4.2 JavaScript/JQuery

Nous pensons aussi exploiter le JavaScript et notamment le JQuery afin de rendre le site plus dynamique (votes en AJAX par exemple), avec pourquoi pas quelques animations. Encore une fois, si nous voulons afficher nous-même les vidéos il sera intéressant d'exploiter le JavaScript. Quant au JQuery, il nous permettrait un développement plus rapide (cette partie n'étant pas essentielle, il ne serait pas judicieux d'y passer trop de temps).

### 4.3 Java/Servlet

Bien évidemment, nous devons utiliser les Servlets et Java afin de générer nos pages Web. Tout le moteur de notre serveur Web sera donc réalisé en Java.

### 4.4 SQL

Toutes les données concernant les membres, les vidéos, les votes, commentaires... seront stockés dans une base de données SQL. Java communiquera donc avec la base de données avec les pilotes adaptés (JDBC). Il créera les objets grâce aux données récupérées de la base SQL, qu'il pourra ainsi afficher, modifier ou supprimer. Bien évidemment, il pourra aussi insérer de nouvelles données.

### 4.5 XML

L'utilisateur peut exporter une liste de vidéos au format XML (une playlist par exemple) et ainsi la sauvegarder pour la réimporter plus tard. Un membre pourra aussi facilement envoyer une playlist à autre membre. En effet, la playlist étant du XML (et donc au format texte), elle pourra facilement s'envoyer (que ce soit via une base de données, ou tout simplement via des sockets).

### 4.6 Java/Swing

Le client lourd sera réalisé en Java à l'aide de la bibliothèque Swing. Cette application reprendra grosso-modo le même design que le site web.

### 4.7 Sockets TCP

En ce qui concerne le tchat P2P entre deux clients lourds, nous utiliserons simplement les sockets TCP.



## 5 Répartition du travail et désignation d'un chef

### 5.1 Chef d'équipe

Notre chef d'équipe sera : Nicolas REYNAUD

### 5.2 Répartition du travail

**Swing Ronan** s'occupera de l'aspect visuel du client lourd à l'aide de swing, et également du design global du site.

**Servlet Nicolas & Kevin** s'occuperont de la partie Web avec notamment les servlets. Nicolas s'occupera en plus de la Base de donnée.

**Tchat Florian** s'occupera de la partie du tchat en Peer to Peer ( P2P ), ainsi que du client lourd avec Ronan.