

# Projet SI

---

## Portail TV

DELUCE Stéphane

COMBES Simon

GAUCHER François

KAKE Abdoulaye

LE THAI BINH Dorian

# Table des matières

|   |           |
|---|-----------|
| <b>1) Contexte de la réalisation</b>          | <b>2</b>  |
| 1.1) Contexte                                 | 2         |
| 1.2) Objectifs                                | 2         |
| 1.3) Hypothèses                               | 2         |
| 1.4) Bases Méthodologiques                    | 3         |
| <b>2) Description générale</b>                | <b>4</b>  |
| 2.1) Environnement du projet                  | 4         |
| 2.2) Caractéristiques des utilisateurs        | 4         |
| 2.3) Fonctionnalités du système               | 5         |
| 2.4) Structure générale du système            | 6         |
| <b>3) Fonctionnalités réalisées</b>           | <b>9</b>  |
| 3.1) Lecture du fichier XMLTV :               | 9         |
| 3.2) Gestion des images                       | 10        |
| 3.3) Générateur de contenus :                 | 10        |
| 3.4) Mise en place de l'architecture back end | 11        |
| 3.5) Conception des vues                      | 12        |
| <b>4) Tests de performances</b>               | <b>14</b> |
| <b>5) Conclusion</b>                          | <b>15</b> |
| 5.1) Conclusion générale                      | 15        |
| 5.2) Améliorations possibles                  | 16        |
| <b>6) Annexes</b>                             | <b>17</b> |
| 6.1) Planification                            | 17        |
| 6.1.1) Gantt prévisionnel                     | 17        |
| 6.1.2) Gestion des tâches                     | 17        |

---

## 1) Contexte de la réalisation

### 1.1) Contexte

Les guides media TV et vidéo sont des composants standards sur Internet. Ils cataloguent les contenus média disponibles sur la TV, les plateformes de replay, de streaming ou encore de vidéo à la demande. On peut trouver une multitude de guides, que ce soit sur les sites d'acteurs de la presse télé comme "Télé Loisir", "Allo Ciné", etc... ou de plateformes proposant des services de streaming ou VOD comme "My Canal +", "Netflix", etc... Ces guides permettent aux utilisateurs d'avoir accès aux informations principales liées au média qu'ils souhaitent consulter.

La réalisation d'un portail web proposant un guide média TV/Vidéo se trouve confrontée à différentes problématiques.

En effet, comme tout portail internet, ce dernier doit être accessible via différentes plateformes (ordinateur, tablette, smartphone) et garder un niveau de performance élevé peu importe le type de terminal utilisé.

Par ailleurs, ce type de portail propose un contenu majoritairement visuel et régulièrement mis à jour. Ces caractéristiques soulèvent des problématiques autour de la gestion des données et de l'architecture du système d'information afin d'optimiser les performances de ce dernier.

### 1.2) Objectifs

L'objectif principal de ce projet est donc de créer un service web proposant un guide des programmes TV à venir sur les différentes chaînes de télévision. Comme il a été précisé précédemment, ce service web doit se présenter sous la forme d'un portail média et doit respecter un certain niveau de performance afin que la qualité d'expérience de ses utilisateurs ne soit pas réduite.

Dans un second temps, ce projet a pour vocation de collecter les statistiques de notre plateforme afin de réaliser des analyses sur des données telles que les visites, les goûts, les tendances ou encore les recherches des utilisateurs.

### 1.3) Hypothèses

Afin de respecter les conditions de performance évoquées dans la présentation de ce projet, il est nécessaire de s'interroger sur les technologies que nous utiliserons afin d'améliorer la fluidité de notre application web et d'optimiser l'accès aux données.



Dans le but de réaliser un système évolutif et, bien que cette fonctionnalité ne soit pas prioritaire, pour prévoir d'éventuelles requêtes de la part des utilisateurs, nous avons décidé de créer une base de données servant à retranscrire les données du fichier XMLTV regroupant l'ensemble des informations du programme TV.

#### 1.4) Bases Méthodologiques

Dans le cadre du développement de ce projet, nous utiliserons le standard XMLTV, qui est une représentation des programmes TV au format XML. Cette norme permet de récolter toutes les informations liées à des programmes TV et rend indépendant la façon de décrire et d'afficher un programme TV. L'utilisation de ce standard est indispensable car la réalisation d'un modèle de données indépendant prendrait trop de temps.

Nous avons décidé de gérer l'évolution de ce projet en suivant une méthode agile ce qui nous permet d'échanger régulièrement avec notre MOA, Mr. DELALANDRE, afin de répondre au mieux à sa demande et orienter nos choix en fonction de nos échanges.

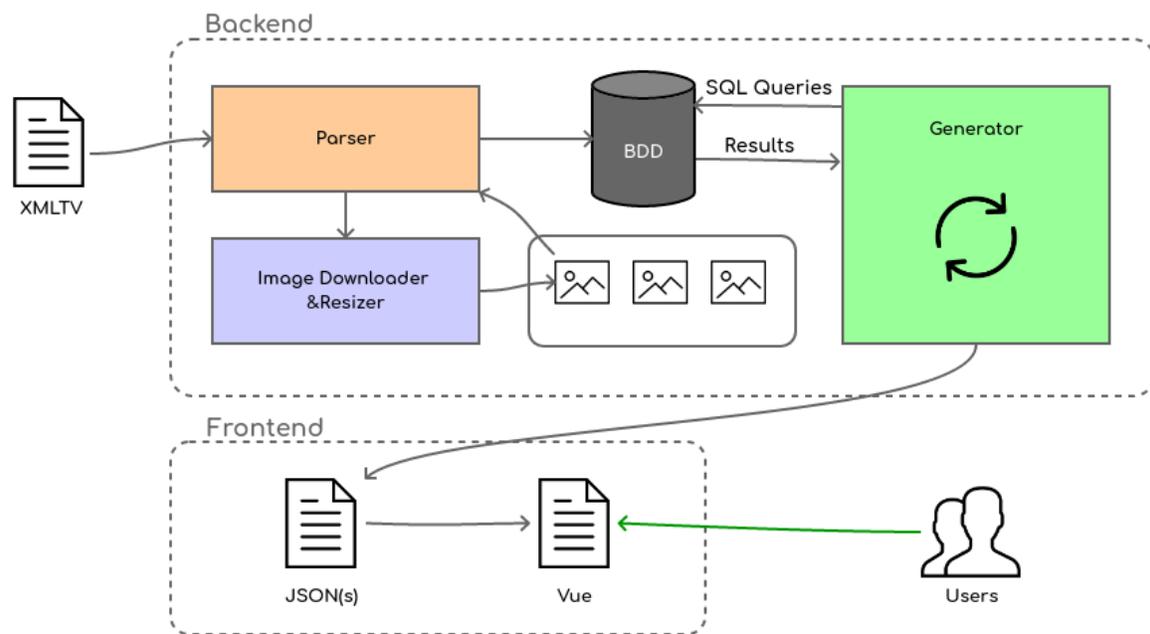
Au sujet des règles de programmation, nous devons respecter les contraintes énoncées dans les parties précédentes. Afin de produire une application accessible et performante sur tous les types de plateforme (ordinateur, tablette, smartphone), le choix d'un Framework FrontEnd efficace est primordial. Par ailleurs, la conception d'une base de données permettra d'améliorer les performances d'accès aux données par requêtes (recherche selon un type, un acteur, un titre, etc...). Toutes ces contraintes peuvent être réalisées avec n'importe quel langage de programmation, ce qui nous offre l'opportunité de choisir ce dernier.

## 2) Description générale

### 2.1) Environnement du projet

Comme il a été précisé précédemment, il est nécessaire d'assurer la compatibilité de notre service web avec n'importe quel type de terminal (ordinateur, tablette, smartphone) tout en conservant de bonnes performances afin d'augmenter la qualité d'expérience de nos utilisateurs.

Puisque la solution proposée est un site internet, sa compatibilité est assurée sur tous les systèmes d'exploitation (Mac OS, Linux, Windows). En revanche, il est nécessaire de fournir un service adapté aux différents navigateurs internet existants.



La figure ci-dessus montre l'environnement global de la solution proposée. Avec cette architecture, notre application sera capable de parser un fichier XML formaté en respectant les critères XMLTV, de récupérer les informations et les images de ce fichier et de générer une vue dynamique à partir de toutes ces données qui seront également stockées dans une base de données.

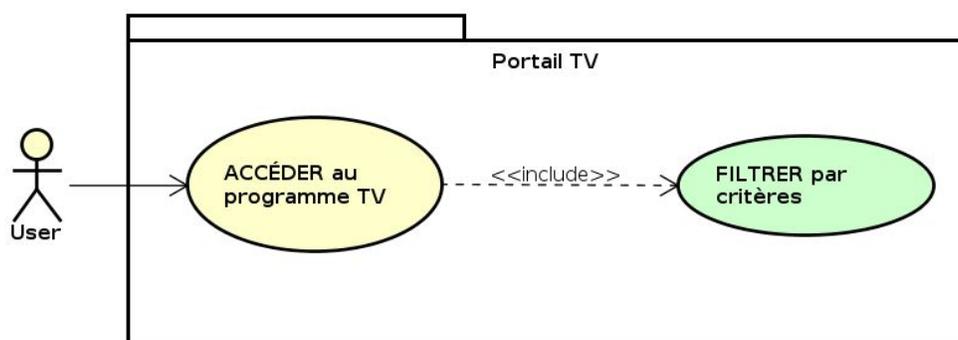
### 2.2) Caractéristiques des utilisateurs

**L'utilisateur quelconque** : ce profil est destiné aux visiteurs de notre service web. Son interface utilisateur doit être fluide, ergonomique et intuitive pour lui offrir la meilleure qualité d'expérience

possible. Aucune authentification n'est requise pour ce type d'utilisateur qui peut accéder à l'ensemble des ressources en lecture uniquement. Son champ d'action se limite à la consultation et à la recherche de programmes TV. Il est défini par les quelques règles suivantes :

- Doit posséder un navigateur (quelconque) et une connexion internet;
- Ne nécessite pas de connaissances particulières en informatique;
- Ne doit pas posséder une bonne expérience de l'application;
- Peut être un utilisateur occasionnel ou régulier;
- Possède des droits de consultation, de visualisation et de recherche sur les différents programmes TV.

### 2.3) Fonctionnalités du système

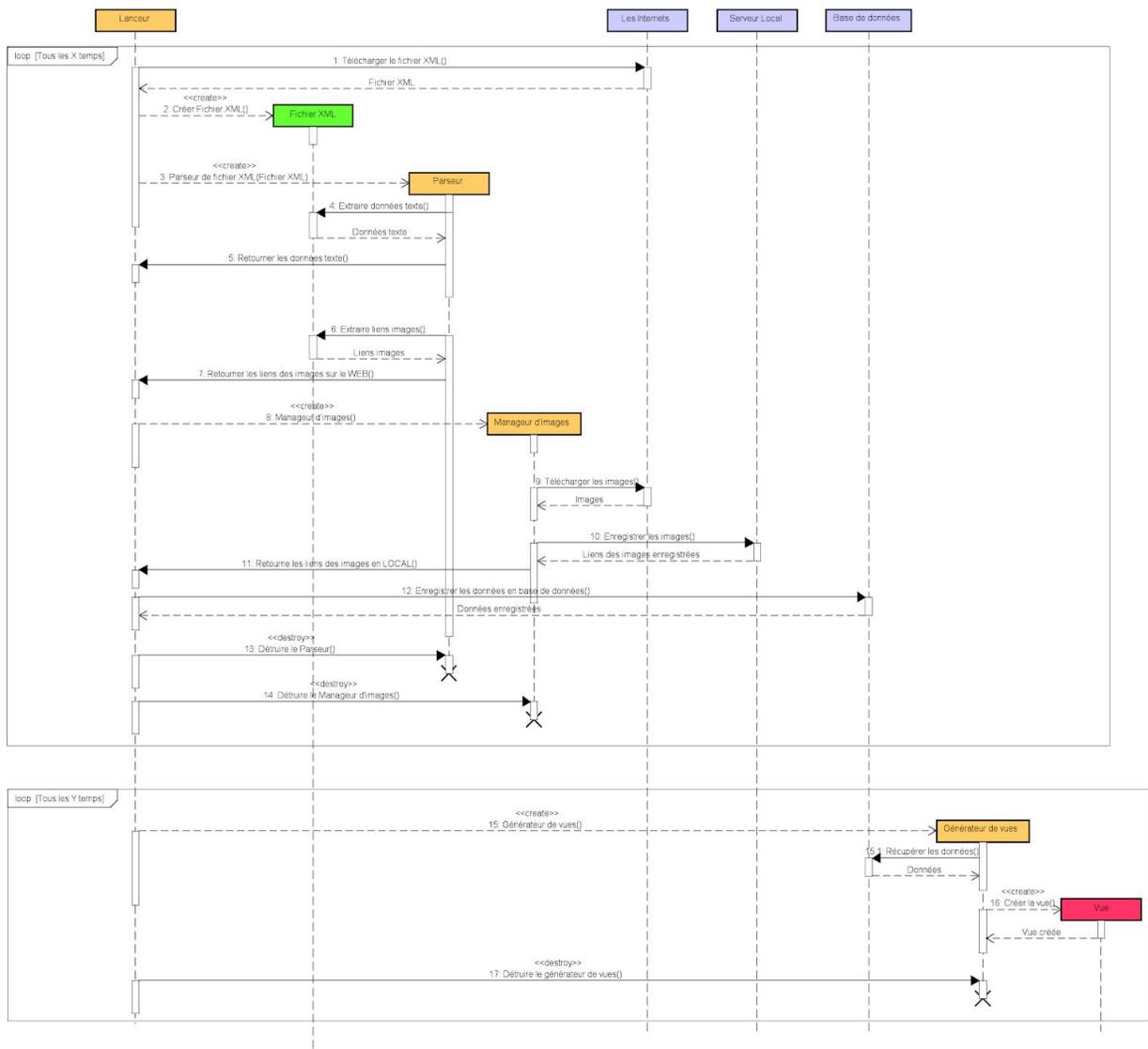


Le système développé doit fournir un ensemble de fonctionnalités. Parmi elles, les fonctions principales sont :

- **La traduction du fichier XML en objets** : un parseur devra fournir des méthodes permettant d'extraire, à partir d'un fichier XML formaté, les informations nécessaires à l'affichage des programmes TV sur notre page internet. Cette classe sera donc utilisée à chaque actualisation de la base de données.
- **Le téléchargement des vignettes des programmes TV** : afin de gagner en efficacité et en performance, le système se doit de récupérer les vignettes des différents programmes et de les stocker localement sur le serveur. Une classe spécifique offrira donc un ensemble de méthodes permettant de télécharger, redimensionner et modifier la qualité de ces images afin d'optimiser le temps de chargement sur les différents terminaux.
- **La création de vues** : notre service étant dynamique, un générateur de vues sera chargé de créer la vue du portail de manière périodique selon une variable de temps.

La génération de la page portail devra être automatique, l'administrateur n'aura qu'à mettre à jour le fichier xml.tv pour alimenter la base de données. Cette étape ne sera pas traitée dans le cadre de ce projet.

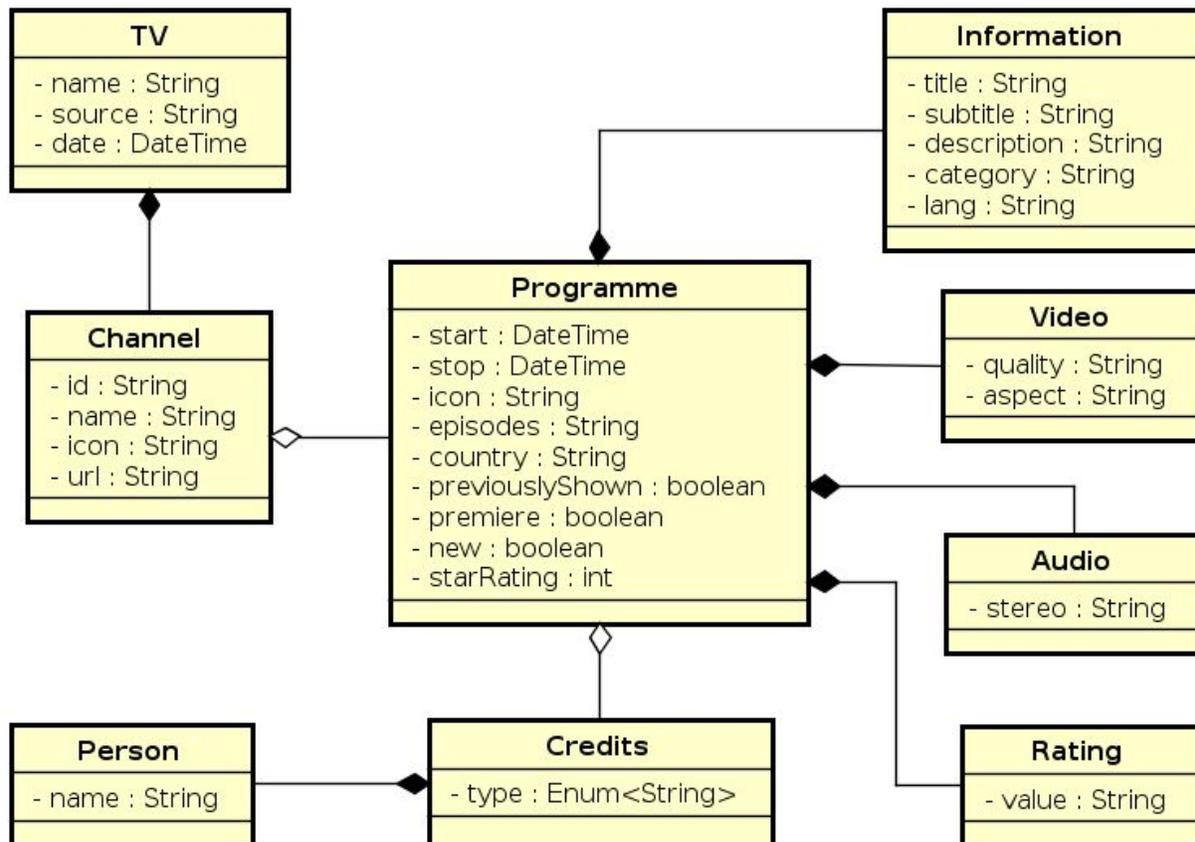
Nous avons déduit du cas d'utilisation précédent un enchaînement de séquences décrites ci après :



## 2.4) Structure générale du système

Le portail web que nous allons développer sera installé sur un serveur et les clients qui souhaitent accéder à notre contenu devront s'assurer d'avoir un accès à internet.

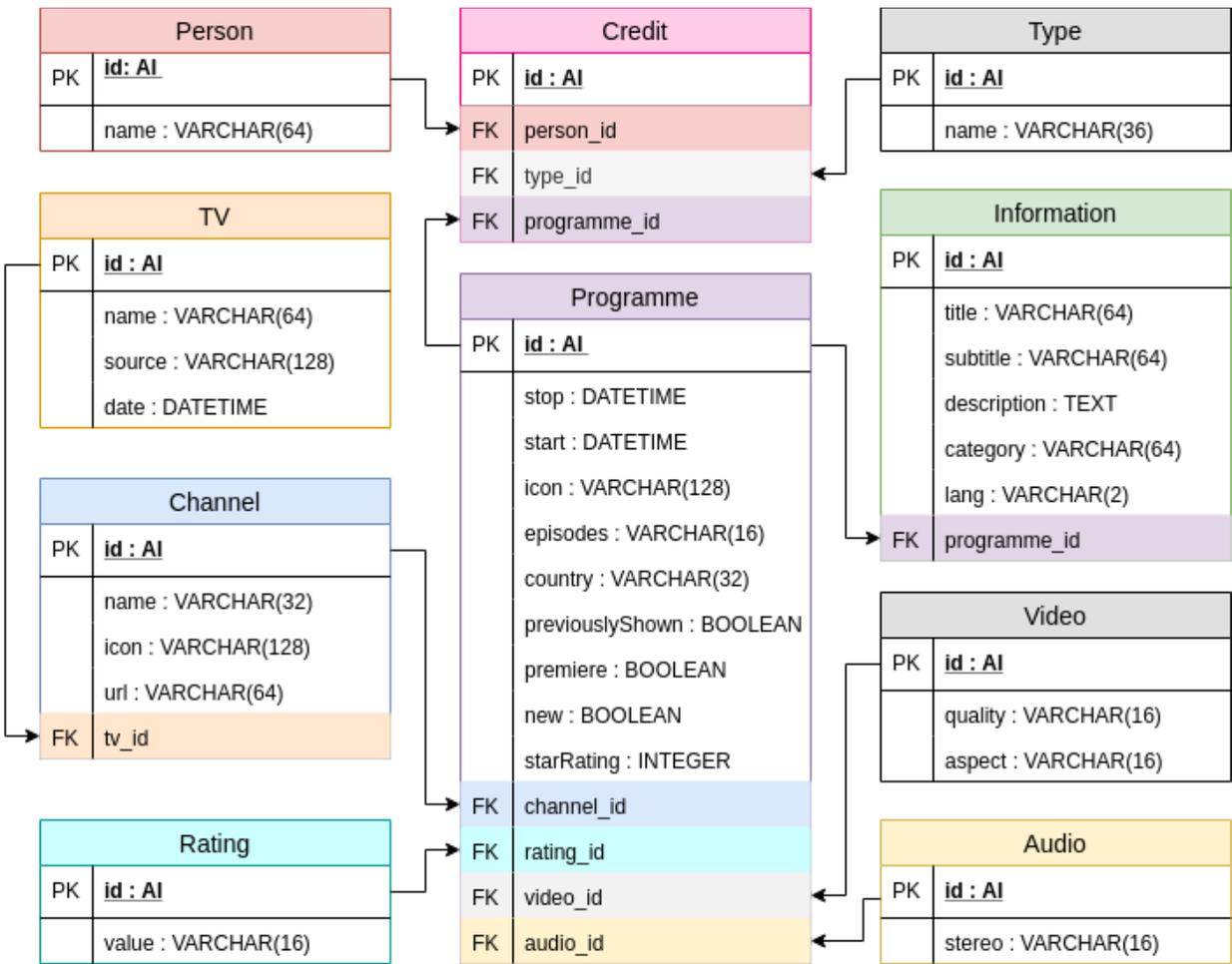
Concernant la structure générale du système, nous avons précédemment décrit dans un schéma sa structure globale, mais nous pouvons d'ores et déjà identifier des classes qui seront indispensables à la réalisation de ce projet et qui sont présentées dans le diagramme de classes suivant :



En entrée, nous disposons d'un fichier XML qui contient des chaînes et leurs programmes TV associés sur un temps donné. Les programmes peuvent être illustrés par une vignette, qui est définie dans le fichier XML par une URL.

Ce fichier devra être parsé afin d'être enregistré en base de données. Les vignettes renseignées dans le fichier source devront être téléchargées sur le serveur et être converties en différentes tailles pour optimiser le temps de chargement sur les différents terminaux.

Un générateur de vue récurrent va interroger la base de données régulièrement afin de générer la page du portail. Le temps entre chaque génération sera un paramètre qui pourra être personnalisé.



Le diagramme ci-dessus présente l'architecture de la base de données qui complète le diagramme de classe précédent. Cette simple structure permettra de stocker toutes les informations pratiques afin d'être efficace lors de l'exécution de requêtes pour faire des recherches en base de données.

## 3) Fonctionnalités réalisées

### 3.1) Lecture du fichier XMLTV :

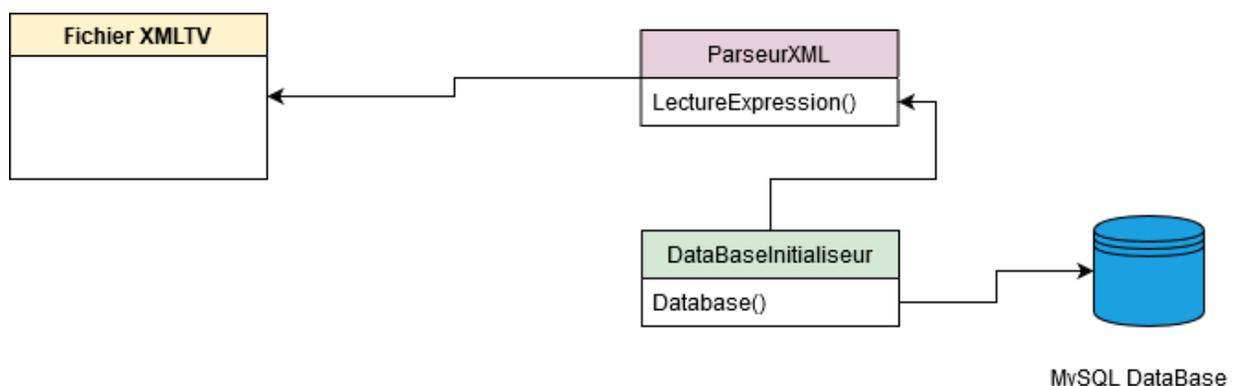
La récupération des données dans un premier temps vient de la lecture d'un fichier XML télécharger sur le site XMLTV. Ce site fournit un fichier XML contenant toutes les informations sur les chaînes et les programmes de la semaine en cours.

C'est ce qui nous pousse à mettre un parseur en place, pour lire ce fichier XML afin d'insérer dans la base de données, les informations récupérés.

Le but est d'utiliser la bibliothèque XPATH, utile pour parcourir facilement un document XML qui nous permet d'évaluer des expressions et récupérer le résultat. Avec cette bibliothèque, nous avons mis une méthode qui prend des expressions correspondant au format du fichier XMLTV et nous fournit un résultat .

Par exemple, si nous mettons `/tv/programmes[1]/title/text()` cela donne le titre du 1er programme du fichier XMLTV.

Avec cette méthode, une autre classe spécialisée dans l'insertion des éléments dans la base de données ou l'initialisation de la base de données fait appelle au parseur pour récupérer les informations dont elle a besoin puis l'insère dans notre base de données MySQL.



## 3.2) Gestion des images

Dans les fichier XMLTV, les programmes peuvent être illustrés par des images. Les chaînes possèdent aussi une image qui sont les logos des chaînes. Ces images sont hébergés sur des serveurs externes et dans des résolution bien trop grandes pour notre utilisation.

Le gestion des images était donc un enjeu très important pour les performances de notre application puisque le chargement d'une image en grand format (1280x720) pour être affichée en petite taille et ce depuis un site distant réduit énormément les performances.

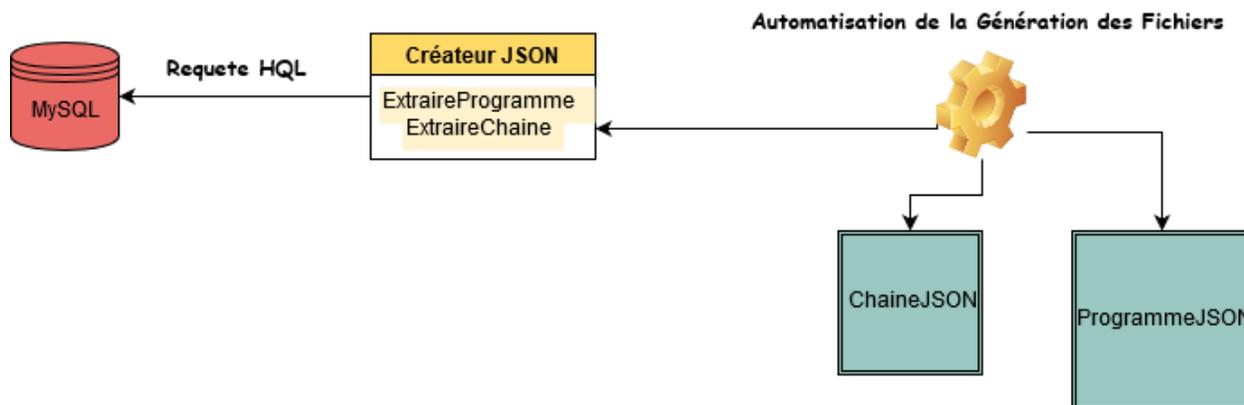
Pour pallier à cette situation, nous avons fait le choix de télécharger les images pour les stocker sur notre serveur et de créer des miniatures qui seront chargées en fonction de la taille de l'écran du client.

Il est donc nécessaire pour chaque image de la télécharger, puis d'en créer des copies (autant que de dimensions disponibles) rangées dans des répertoire distincts. Le traitement des images est réalisé à partir de la librairie "Thumbnailator" qui fournit un bon ratio qualité d'image / poids de la librairie.

Nous avons donc développé deux méthodes de redimensionnement, la première réalise un rognage de l'image réelle pour que l'image attendue ne soit pas déformée lors du redimensionnement. La seconde réduit l'image aux dimensions souhaitées.

## 3.3) Générateur de contenus :

Pour cette partie, le but est de générer du contenu, les informations et la description des données (chaîne, programme, ...) dans un fichier au format JSON. Ces données sont récupérés à partir de la base de données. Puis nous avons mis en place une génération automatique qui au cours de la journée génère des fichiers JSON.





La création des fichiers JSON est fait par l'appel d'un programme qui extrait des données dans la base de données par une requête HQL, qui est une version SQL de la bibliothèque Hibernate, puis écrit les résultats obtenu, convertit au format JSON dans un fichier.

Et le générateur, divise le temps en matin, midi et soir puis à ces moments de la journée, un fichier JSON est généré contenant les programmes de cette tranche d'heure. Pour le fichier JSON concernant les chaînes, une génération est faite tous les 7 jours au moment dont un nouveau fichier XMLTV est téléchargé.

La vue fera une récupération de ces fichiers puis générera une vue dynamique à partir de toutes ces données lu dans le fichiers JSON choisi.

Technologies utilisés :

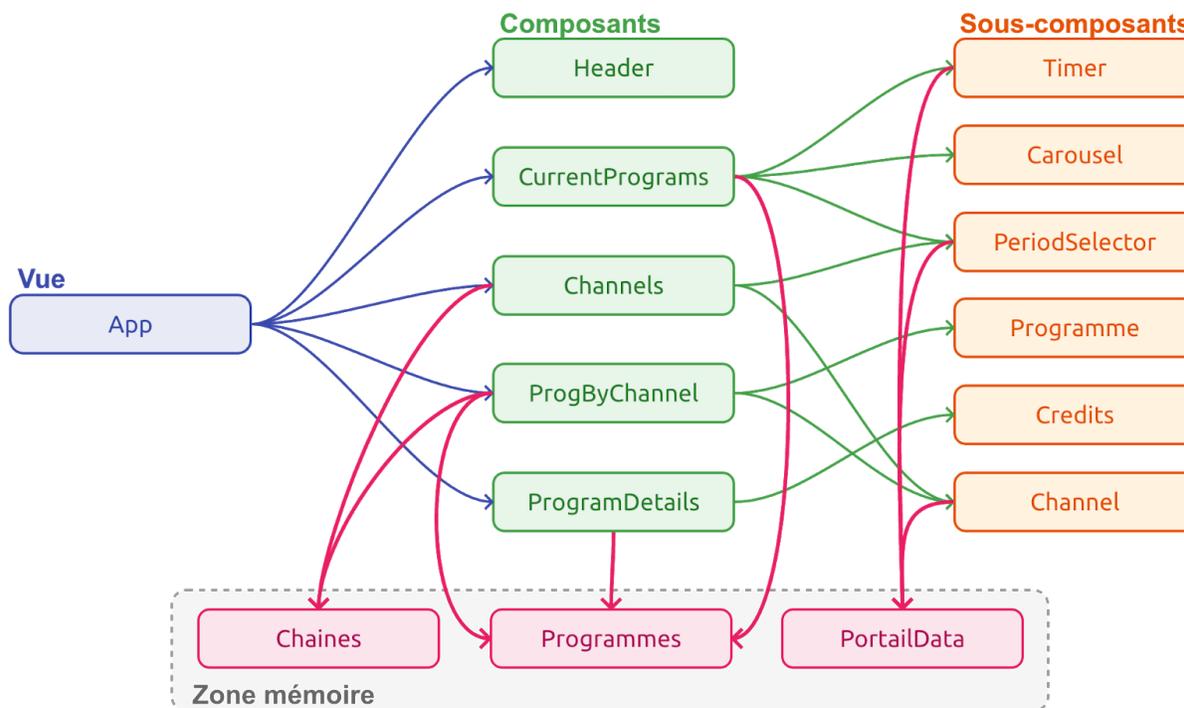
- GSON: un framework google pour la conversion d'objet au format JSON ,
- HQL: pour faire des requêtes sur la base de données et obtenir des Objets conforme à notre Model.

### 3.4) Mise en place de l'architecture back end

Nous avons décidé d'utiliser un environnement JEE pour la partie back end. Nous avons configuré notre projet pour que ce dernier puisse tourner sur un serveur JEE accessible 24h sur 24h. Le serveur JEE a été configuré afin de lancer automatiquement des tâches récurrentes pour générer les fichiers que le client va exploiter.

### 3.5) Conception des vues

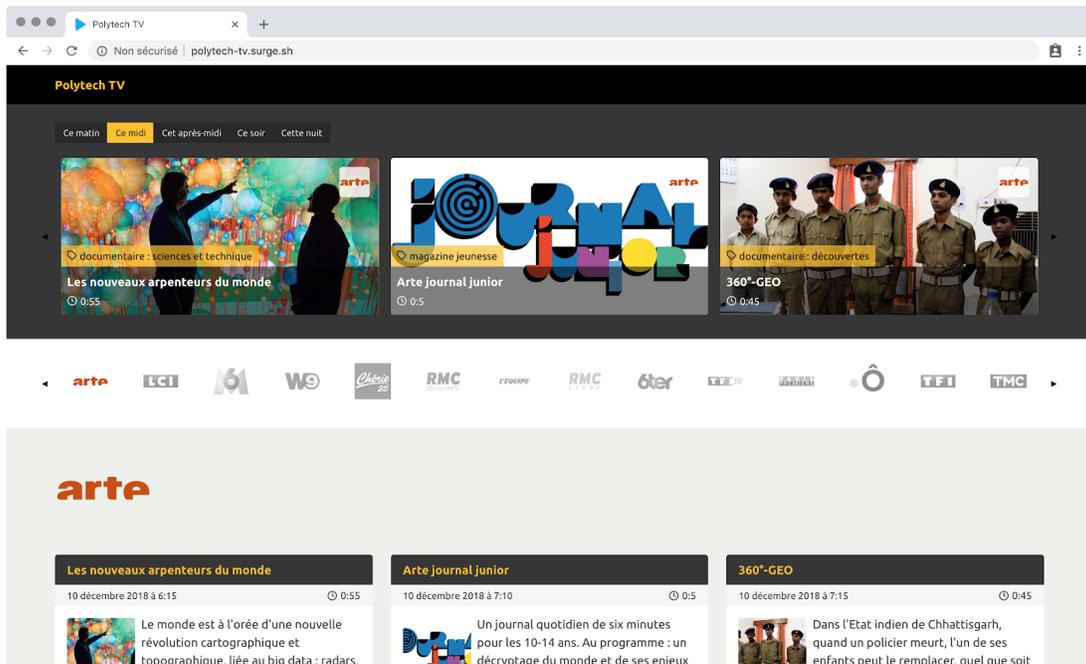
Nous avons fait le choix de faire une application web monopage (single page application) à l'aide du framework javascript **VueJs**. L'un des intérêts de ce framework est qu'il permet de décomposer l'application en composants web. Ce système permet de décomposer les vues en plusieurs éléments simples.



La structure générale peut être décrite avec le schéma ci-dessus :

- Des composants sont créés afin de structurer la page à afficher;
- Ces composants peuvent être composés de sous-composants. Ces derniers sont créés afin d'être réutilisables par plusieurs composants;
- Les composants et sous-composants utilisent de plus une "zone mémoire" afin de récupérer les informations de programme;
- Une classe "vue" reprend l'ensemble des composants et sous-composants qui seront déployés sur le portail TV.

L'ensemble des programmes à afficher sont répertoriés dans un fichier JSON généré par le "serveur" de notre projet.



Le site a été conçu pour être responsive : il s'adapte ainsi à tout type de plateforme (ordinateur, tablette et smartphone).

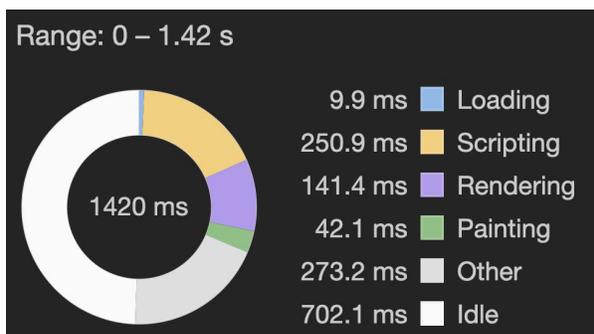
## 4) Tests de performances

Pour réaliser les tests de performances, nous avons utilisé les outils intégrés du navigateur "Google Chrome". Il permet d'émuler des connexions dégradées, ce qui va nous être utile pour faire les tests sur des connexions mobiles.

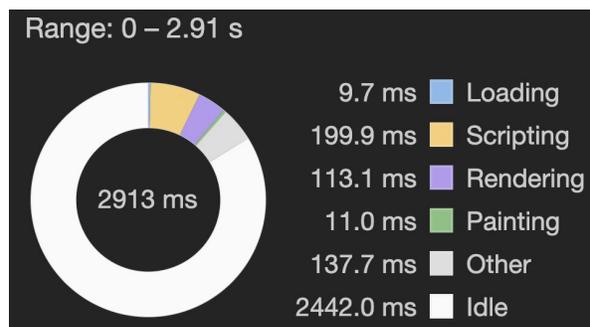
Pour optimiser les temps de chargement, nous avons porté une attention particulière aux images. Les images chargées ont deux tailles, une petite de 64 pixels et une plus grande de 128 pixels.

De plus, les éléments ne sont chargés uniquement lorsqu'ils sont affichés, puis stockés dans le cache du navigateur pour ne pas avoir à les charger plusieurs fois.

Nous avons effectué deux tests de performances correspondant à des cas courants d'utilisation : une connexion 3G dégradée sur mobile et une connexion ADSL sur ordinateur. Ces deux tests donnent ainsi un ordre d'idée du temps de chargement avec respectivement le "pire des cas" et le cas le plus répandu.



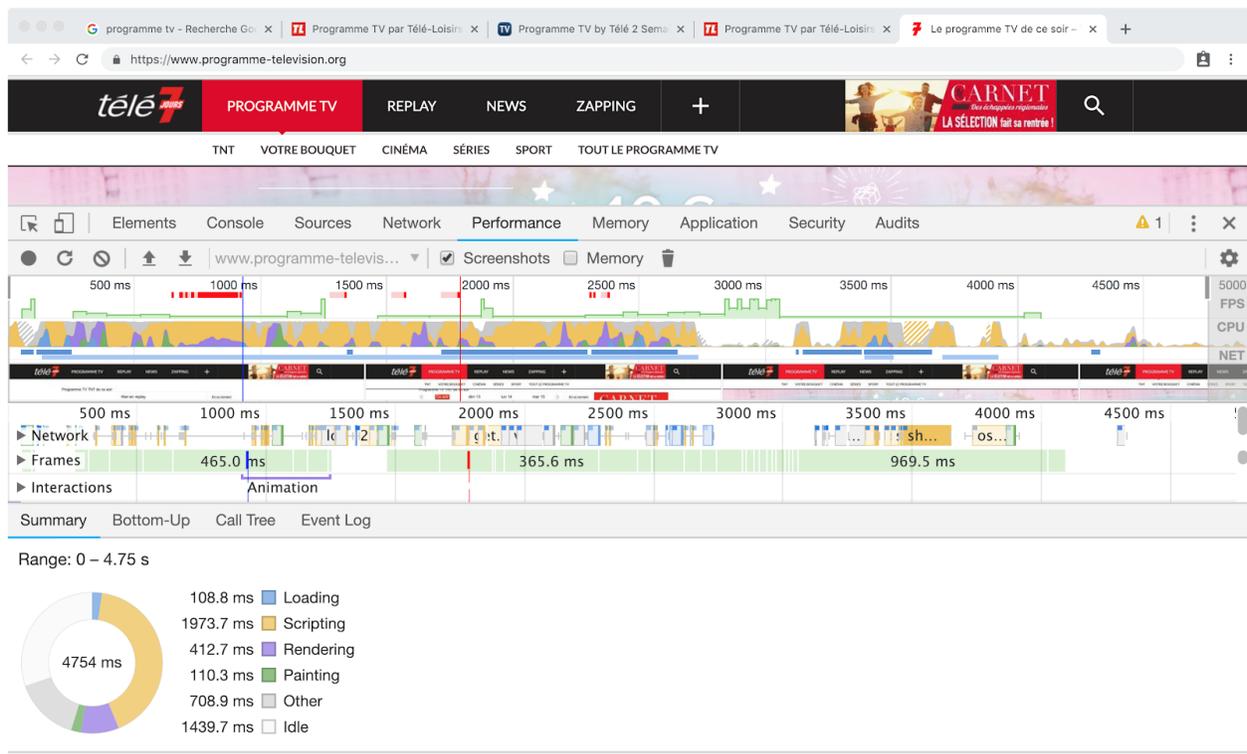
**ADSL sur ordinateur**



**3G dégradée sur mobile**

Nous avons également effectué des tests sur des plateformes similaires afin de situer nos résultats.

| Télé-Loisirs | Télé 2 semaines | Télé 7 jours |
|--------------|-----------------|--------------|
| <b>6.10s</b> | <b>4.32s</b>    | <b>4.75s</b> |



On remarque ainsi, par comparaison, que le temps de chargement de notre site ne nécessite pas une attente trop longue.

## 5) Conclusion

### 5.1) Conclusion générale

Ce projet nous aura permis de voir différents aspects de la mise en place d'un portail TV.

La lecture du fichier XMLTV regroupant les informations du programme a pu être faite de manière efficace.

L'ensemble de ces informations peuvent être consultées de manière claire et simple à travers notre site et sur tout type de plateformes.

L'optimisation du chargement des données sur notre site faisait partie des objectifs importants du projet. En effet, l'utilisateur doit pouvoir consulter les programmes de manière fluide. Les tests de performance effectués montrent que le temps de chargement est raisonnable.



La création du serveur est utile pour conserver et mettre à jour les données de programme. Cependant, bien qu'il a été créé et configuré correctement, il n'a pas pu être intégré totalement à temps dans notre projet.

## 5.2) Améliorations possibles

Certaines fonctionnalités permettant d'optimiser notre projet peuvent être incluses :

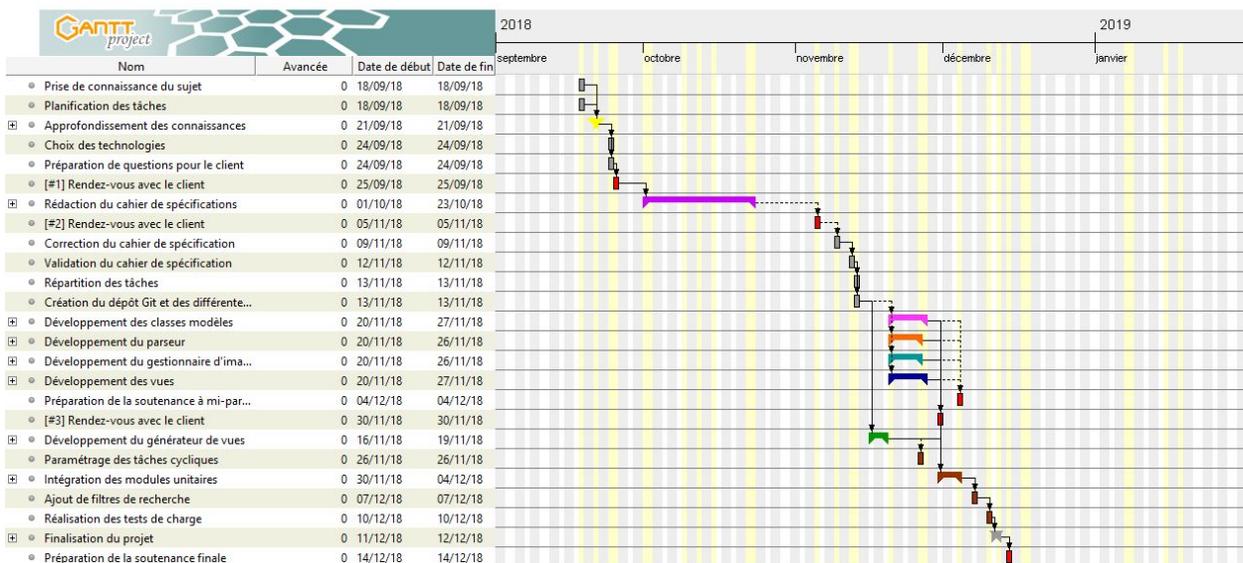
- Conception de filtres de recherche : l'utilisateur pourrait rechercher un programme, un présentateur... dans une barre de recherche incluse dans le site. Cette fonctionnalité peut se réaliser en mettant en place des requêtes SQL sur le serveur.
- Conception de filtres de période : le site possède actuellement un bandeau représentant les moments de la journée (matin ; midi ; après-midi ; soir ; nuit ). L'objectif serait d'afficher les programmes correspondant à la période en cours ou cliquée par l'utilisateur.
- Collection de statistiques sur le site : comme il a été énoncé dans les objectifs du projet, le but est d'analyser les visites du site afin de connaître les goûts des clients.

## 6) Annexes

### 6.1) Planification

#### 6.1.1) Gantt prévisionnel

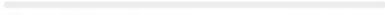
Nous avons réalisé un Gantt prévisionnel ci-dessous



Certaines tâches n'ont pas pu être traités ou réalisés dans les temps, notamment à cause du retard pris dans le développement des classes modèles et du générateur de vues.

#### 6.1.2) Gestion des tâches

Au niveau de la gestion des tâches, Nous les avons définis à l'aide de l'outil GitLab

|  |   |               |                   |
|--|---|---------------|-------------------|
| <b>Gestion de projet</b><br>Sep 18, 2018–Jan 18, 2019<br>Deluce / Portail-TV | <br>9 Issues · 0 Merge Requests     | 77% complete  | Project Milestone |
| <b>Cahier de spécifications</b><br>Deluce / Portail-TV                       | <br>18 Issues · 0 Merge Requests    | 94% complete  | Project Milestone |
| <b>Comments</b><br>Deluce / QtProject  | <br>12 Issues · 0 Merge Requests    | 100% complete | Project Milestone |
| <b>Conception des vues</b><br>started on Oct 22, 2018<br>Deluce / Portail-TV | <br>8 Issues · 0 Merge Requests     | 75% complete  | Project Milestone |
| <b>Création/Alimentation de la base de données</b><br>Deluce / Portail-TV    | <br>7 Issues · 0 Merge Requests     | 100% complete | Project Milestone |
| <b>Gestion des classes modèles</b><br>Deluce / Portail-TV                    | <br>7 Issues · 0 Merge Requests     | 100% complete | Project Milestone |
| <b>Gestion des images</b><br>Deluce / Portail-TV                             | <br>3 Issues · 0 Merge Requests     | 100% complete | Project Milestone |
| <b>Génération des fichiers</b><br>Deluce / Portail-TV                        | <br>1 Issue · 0 Merge Requests      | 0% complete   | Project Milestone |
| <b>Générateur de contenu</b><br>Deluce / Portail-TV                          | <br>4 Issues · 0 Merge Requests   | 50% complete  | Project Milestone |
| <b>Intégration</b><br>Deluce / Portail-TV                                    | <br>11 Issues · 30 Merge Requests | 95% complete  | Project Milestone |
| <b>Lecture du fichier XMLTV</b><br>Deluce / Portail-TV                       | <br>3 Issues · 0 Merge Requests   | 100% complete | Project Milestone |

L'outil permet de définir pour chacun des tâches à réaliser. Nous pouvons voir de plus l'avancée des tâches.

Le tableau ci-dessous résume quel(s) personne(s) ont été affectée(s) ou ont contribué aux différentes parties mentionnées dans notre projet GitLab :

| Tâche                                       | Personne(s) affectée(s) | Personne(s) ayant réalisée(s) |
|---|-------------------------|-------------------------------|
| Gestion de Projet                           | Stéphane                | François / Simon / Stéphane   |
| Création/Alimentation de la base de données | Simon                   | Simon                         |
| Intégration des composants                  | Groupe entier           | -                             |
| Générateur contenu (json)                   | Dorian                  | Dorian / Abdoulaye            |
| Gestion des classes modèles                 | Dorian                  | Dorian / Abdoulaye, Simon     |
| Lecture du fichier XMLTV                    | Abdoulaye               | Abdoulaye / Simon             |
| Gestion des images                          | Simon                   | Simon                         |
| Développement / Intégration des vues        | Stéphane, François      | Stéphane, François            |

