



UE

Applications Web

---

## Projet Web : Media Player

---

*Auteurs :*

Samer KHEDHRI

Ines MESSOUSSI

*Encadrants :*

Daniel MULLER

René CHALON

Version du  
29 mars 2022

## Table des matières

<b>1</b>	<b>Analyse et spécification des besoins</b>	<b>2</b>
1.1	Les besoins fonctionnels . . . . .	2
1.2	Les besoins non fonctionnels . . . . .	3
<b>2</b>	<b>Conception</b>	<b>5</b>
2.1	Conception logique : . . . . .	5
2.2	Conception physique . . . . .	6
<b>3</b>	<b>Implémentations</b>	<b>7</b>
3.1	Aperçu du travail réalisé . . . . .	7

## Introduction

Afin de mettre en oeuvre et en application les différents aspects de la programmation avec le langage Javascript ainsi que le plateforme Node.js bien sûr couplé avec le langage HTML et CSS , nous nous sommes lancés dans la conception et l'implémentation d'une application Web proposant un "Media Player" permettant la visualisation et la manipulation des différents types de fichiers ( vidéo , audio , image).

Dans ce media player , on propose un système de playlist ou l'utilisateur peut manipuler les fichiers à entendre ou à visualiser en ajoutant et en supprimant des éléments à la et de la playlist.

Dès la début du projet , nous avons pris un grand intérêt dans l'ergonomie des interfaces et l'évitement de l'utilisation des outils pré préparé tel que le Bootstrap afin de vraiment réaliser notre application

## 1 Analyse et spécification des besoins

Avant la réalisation , on a fixé des besoins à satisfaire . Ces besoins se présentent dans :

### 1.1 Les besoins fonctionnels

#### Pour le lecteur des vidéos :

- ✓ Visualiser les différentes vidéos .
- ✓ Ajouter et supprimer des fichiers de la playlist.
- ✓ Lecture consécutifs des vidéos et des audios : dès l'un est fini , ce lui après commence directement.
- ✓ Capacité de manipuler la lecture des vidéos et audios avec une interface de commande simplifié sous forme des boutons.

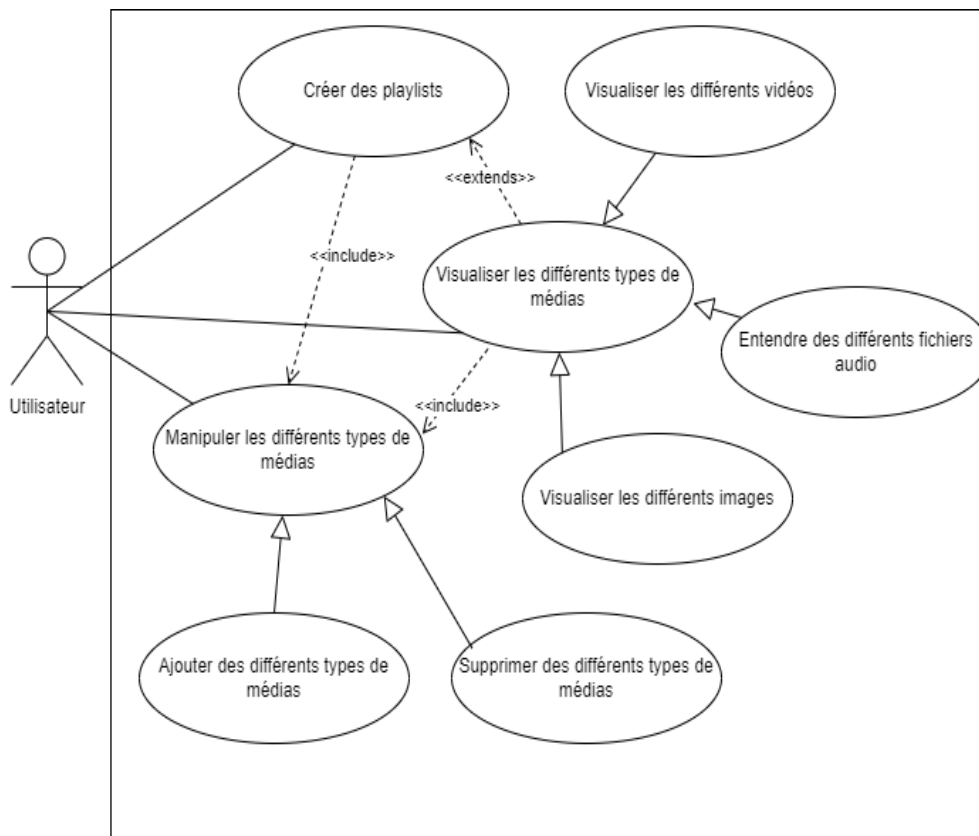
#### Pour le lecteur des images :

- ✓ Visualiser les différentes images .
- ✓ Ajouter et supprimer des images.
- ✓ Capacité de manipuler les images en appliquant des filtres bien déterminés.
- ✓ Sauvegarder l'image après l'application des filtres

## 1.2 Les besoins non fonctionnels

- 1)**Rapidité** : le temps de réponse de l'application ne doit pas dépasser 4 secondes.
- 2)**Compatibilité** : L'application web doit fonctionner sur n'importe quelle navigateur Web.
- 3)**Ergonomie** : L'interface doit être esthétiquement plaisante et facile à comprendre et à utiliser par l'utilisateur.

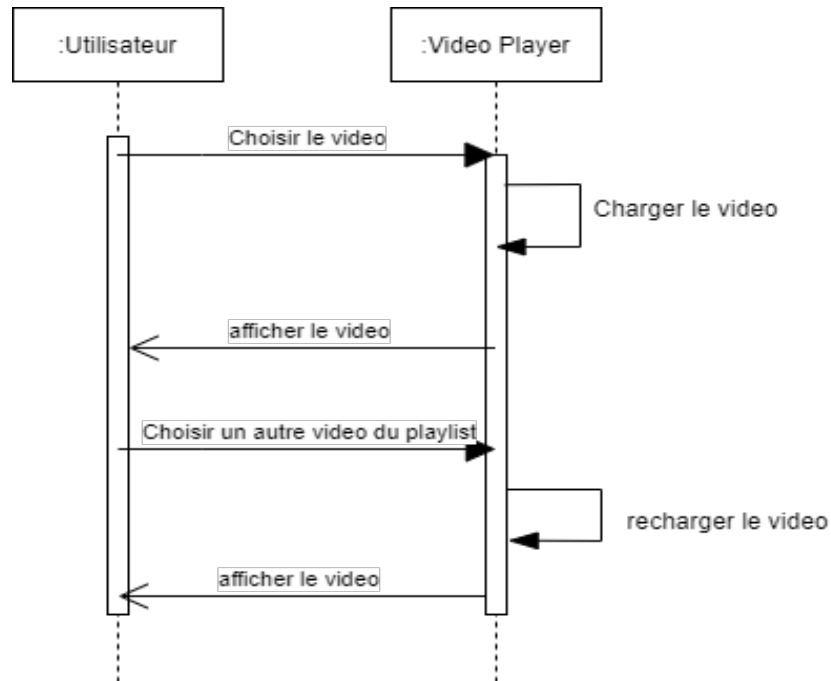
On peut résumer les objectifs de notre application Web avec un diagramme de cas d'utilisation présenté ci-dessous :



**FIGURE 1** – Diagramme de cas d'utilisation

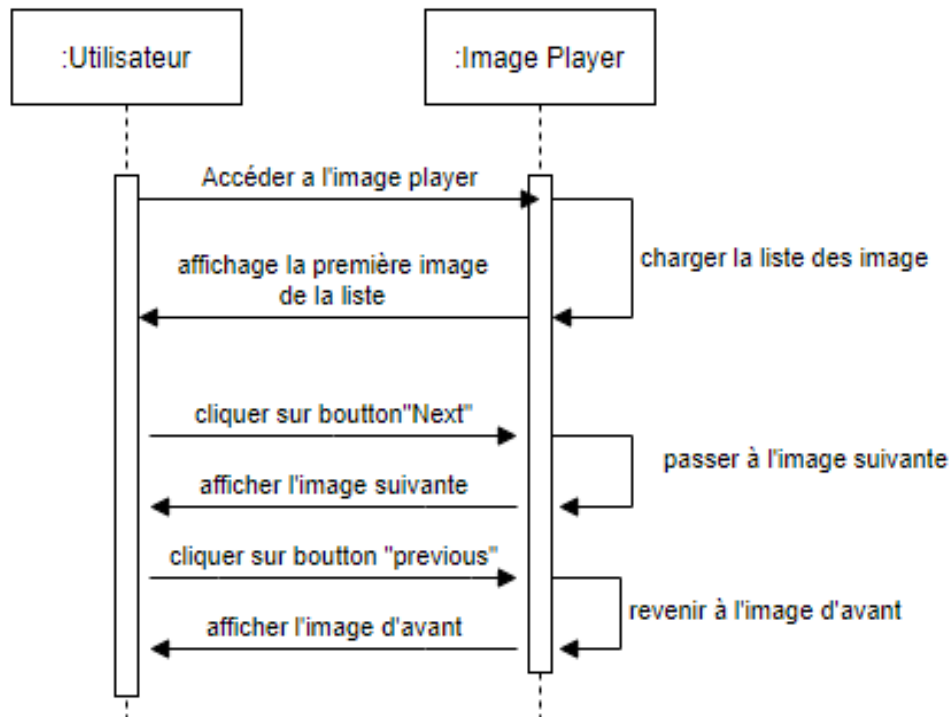
L'un des use-case qu'on peut la traiter est la visualisation d'un vidéo . il s'agit d'un use-case qui hérite de celui de la visualisation des différentes types des

médias . l'étude du scénario nominal est garantit par le diagramme de séquence affiché ci-dessous :



**FIGURE 2** – Diagramme de séquence en boîtes noires

Passons maintenant à notre deuxième média player "image player". Le scénario nominal pour la visualisation des images est représenté par le diagramme de séquence ci-dessous :



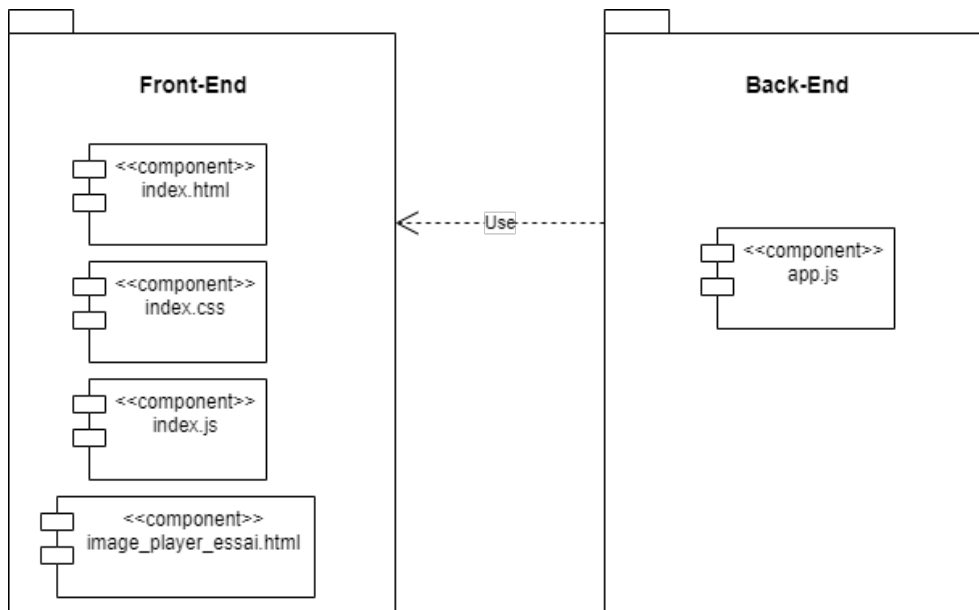
**FIGURE 3** – diagramme de séquence de visualisation d’images

## 2 Conception

Cet application est basé sur une conception simpliste permettant de réaliser le fonctionnement désiré sans compliquer trop les choses.

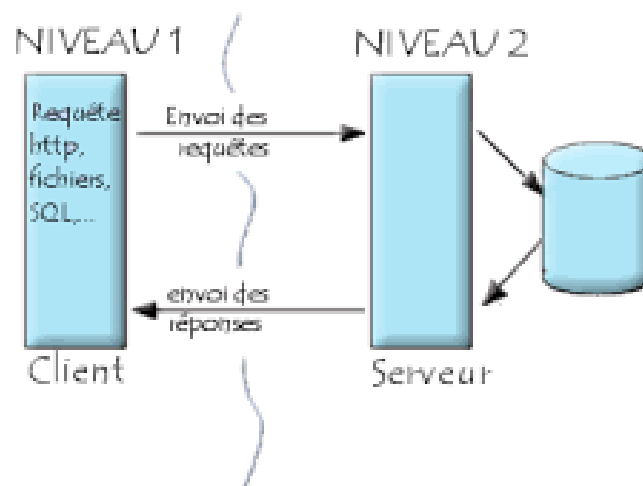
### 2.1 Conception logique :

L’application est divisé essentiellement en deux paquets : le premier présentant la partie back-end implémenté sous le cadriciel Node Js . Ce paquetage est utilisé par le deuxième paquetage qui présente la partie front-end permettant l’affichage des informations récupérés et l’interaction avec l’utilisateur. Ceci est mieux présenté dans le diagramme de paquetage ci-dessous :

**FIGURE 4** – Diagramme de paquetage

## 2.2 Conception physique

Puisqu'il s'agit d'une application simpliste, la meilleure structure physique qui sera compatible est une architecture 2-tiers ou on dispose d'un serveur et d'un client qui échange des requêtes pour le transfert des informations .

**FIGURE 5** – Architecture 2-tiers

## 3 Implémentations

### 3.1 Aperçu du travail réalisé

L'application consiste principalement de deux parties implémenté : Video Player et Image Player .chaque partie est implémenté individuellement puis les deux sont liée .

pour le Video Player , suite à l'exécution de l'application , on aura l'interface suivante :

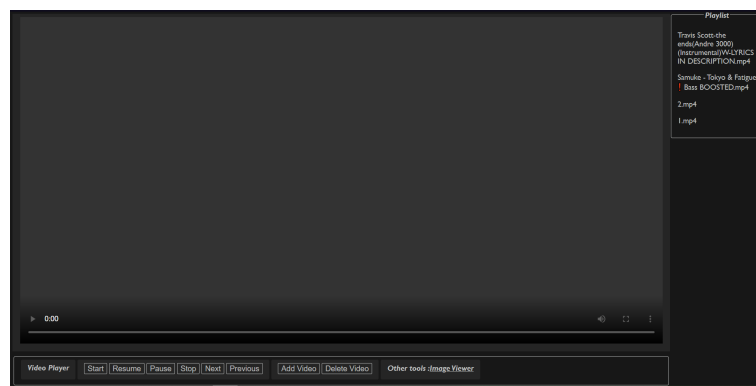


FIGURE 6 – Interface du Video Player

Ensuite si nous voulons consulter l'image player nous pouvons tous simplement cliquer sur le bouton en bas à droite de cette interface et donc nous obtenons l'interface suivante :

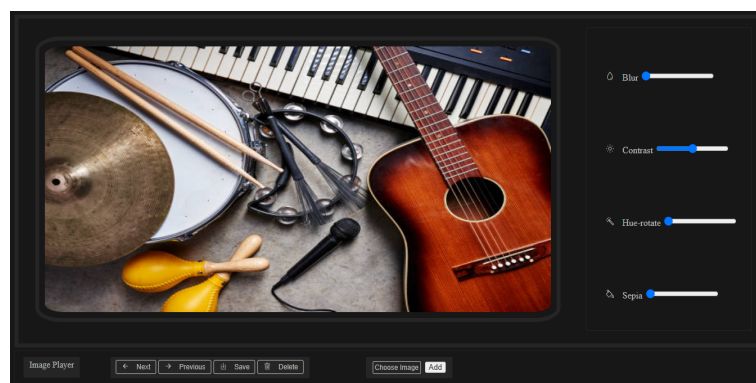
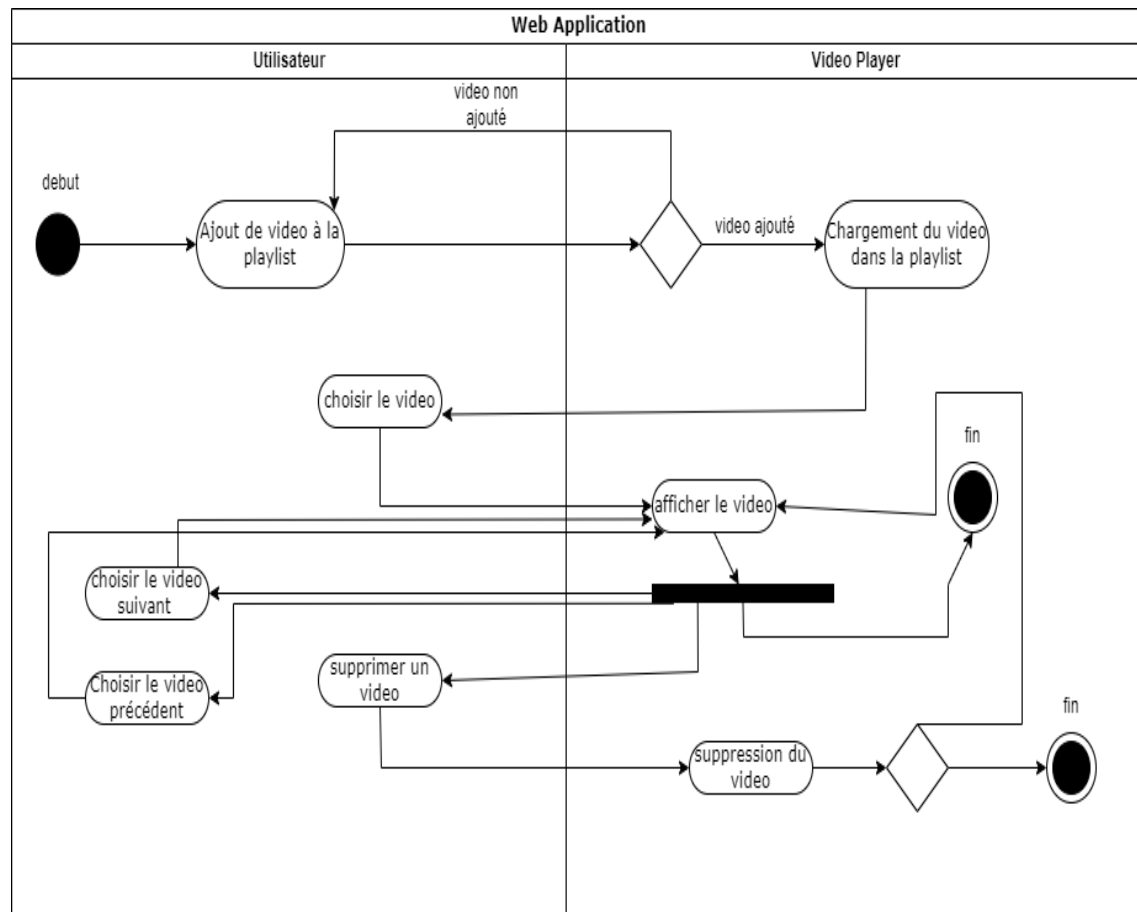


FIGURE 7 – Interface de l'image Player

La navigation et l'utilisation de ces différentes interfaces constituent l'objectif de



l'application web développée. On peut comprendre exactement la façon d'utiliser chaque interface à l'aide des diagrammes d'activité présentés ci-dessous :



**FIGURE 8** – Diagramme d'activité du Video Player

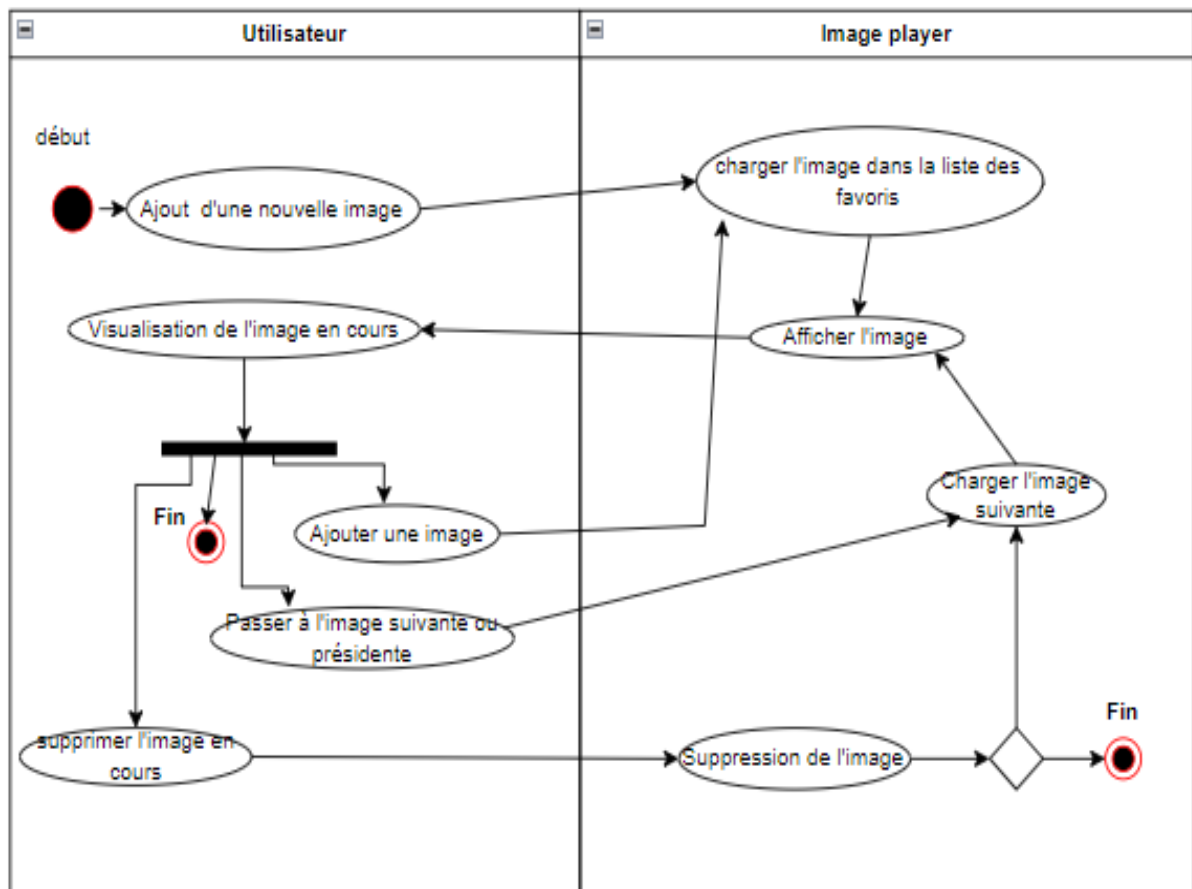


FIGURE 9 – Diagramme d'activité de l'image Player

## Conclusion

Tout au long de la réalisation de notre mini application "Media Player" nous avons sans doute acquérir des compétences en JavaScript et NodeJS. Nous avons pu ainsi nous familiariser avec ces derniers et donc appris à bien gérer les interactions client-serveur.

Pour le moment notre "Media Player" permet d'exploiter trois types de médias des vidéos, des audio et des images. Ainsi, il est possible de travailler encore pour ajouter d'autres fonctionnalités afin de l'améliorer.