

CAHIER DES CHARGES

SAE 5.01



LOIS GENAY | PLATEFORME MULTIMEDIA | 28 SEPTEMBRE 2025

Sommaire

TABLE DES MATIERES

1. Contexte et enjeux	I
2. Objectifs du projet	I
3. Description des tâches	2
1) Phase de conception	2
2) Phase de développement	2
3) Phase d'intégration et déploiement	2
1) Phase de validation et documentation	2
4. Schéma global de la solution	3
5. Planning prévisionnel	4
Répartition générale des rôles	4
Tableau planning détaillé	4
Répartition de la charge horaire	5
6. Ressources mobilisées	6

I. CONTEXTE ET ENJEUX

Dans le cadre de la SAÉ 5.01, il est demandé de concevoir et réaliser un projet mettant en avant au moins deux compétences majeures du BUT Réseaux & Télécommunications.

Le projet que je propose s'inscrit dans le domaine des **réseaux** et de l'**informatique**, avec une touche en **télécommunications** pour la gestion des flux multimédia.

L'idée est de développer une plateforme multimédia auto-hébergée, accessible via une interface web sécurisée, permettant :

- La lecture de **films et séries** avec récupération automatique des métadonnées (affiches, synopsis, acteurs) ;
- L'écoute de **musiques** indexées via une base de données musicale ;
- La consultation d'une **galerie photos** intelligente, exploitant les métadonnées EXIF pour classer les clichés par date, lieu ou appareil photo.

L'ensemble sera conteneurisé avec Docker pour garantir portabilité, isolation et facilité de déploiement. Ce projet se distingue de solutions existantes (Plex, Jellyfin, Emby) par sa dimension **pédagogique**, car il sera construit de zéro, en maîtrisant toute la chaîne technique (backend, frontend, conteneurisation, intégration des API).

2. OBJECTIFS DU PROJET

Développer une **interface web** (responsive et sécurisée) permettant à un utilisateur authentifié de naviguer dans les différents médias (vidéos, musiques, photos).

Mettre en place une **API backend** pour gérer l'authentification, les métadonnées et le streaming des fichiers.

Intégrer des **indexeurs** externes (ex. The Movie Database pour les films, MusicBrainz pour la musique, lecture EXIF pour les photos).

Assurer la **lecture fluide des contenus multimédia** directement dans le navigateur via des players HTML5.

Conteneuriser la solution via **Docker** afin de faciliter le déploiement sur un NAS ou un serveur local.

Documenter la solution (guide utilisateur + guide technique).

3. DESCRIPTION DES TACHES

1) Phase de conception

- Étude de faisabilité et analyse de projets similaires (Plex, Jellyfin, etc.).
- Conception de l'architecture technique (schémas).
- Définition des technologies utilisées (Docker, Node.js/Python, React/Vue.js, API REST).

2) Phase de développement

- Développement du **backend** (API REST + gestion utilisateurs).
- Développement du **frontend web** (interface avec authentification et navigation).
- Intégration du **player multimédia** (HTML5 pour audio/vidéo, visionneuse pour photos).
- Mise en place des **indexeurs de métadonnées** (appel API externes).

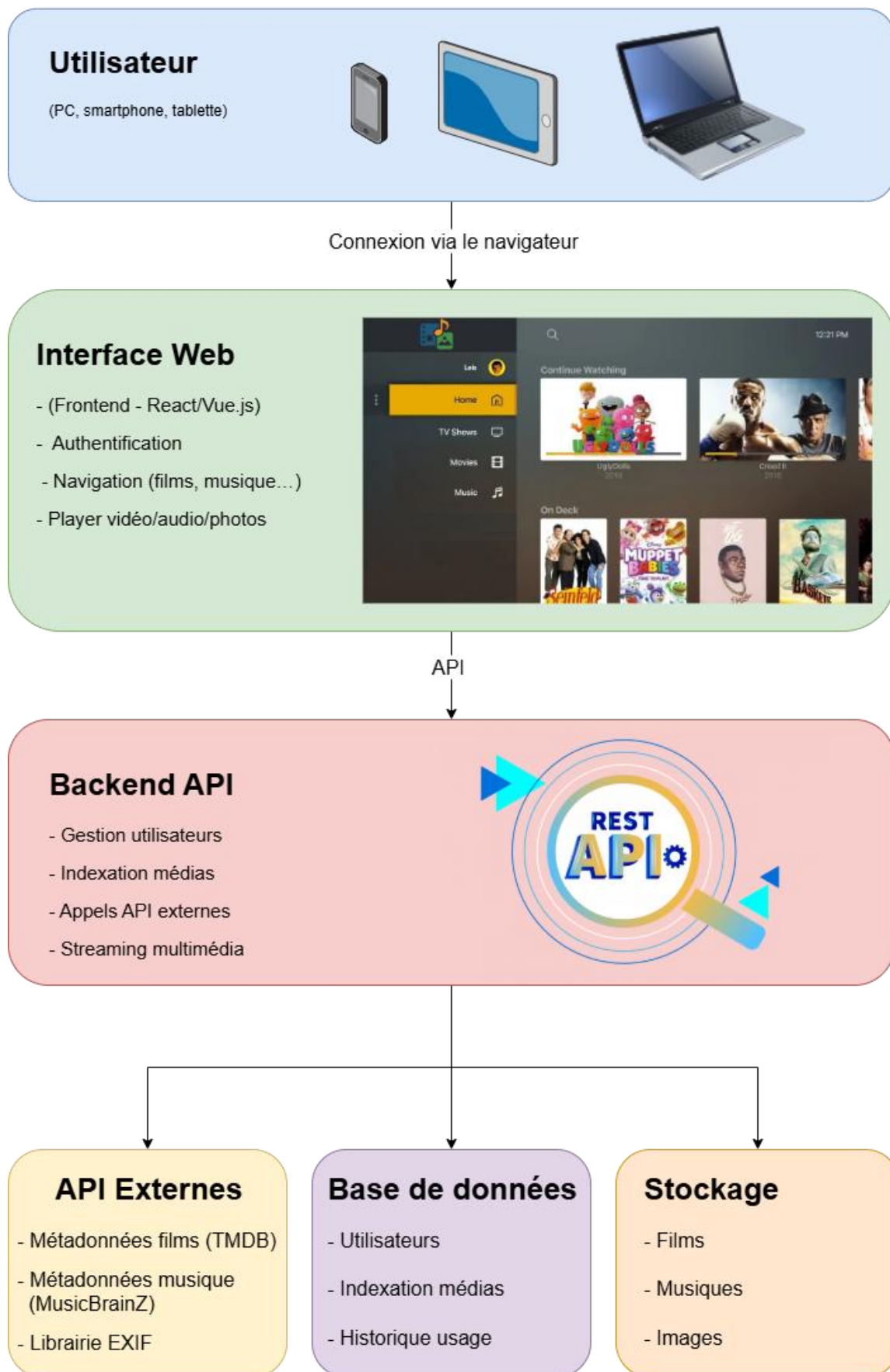
3) Phase d'intégration et déploiement

- Création du container Docker et fichier docker-compose.yml.
- Connexion au stockage (NAS ou répertoire local).
- Tests d'accès multi-utilisateurs et streaming.

4) Phase de validation et documentation

- Tests fonctionnels et de performance.
- Rédaction du rapport final (8–12 pages).
- Rédaction de la documentation utilisateur et administrateur.

4. SCHEMA GLOBAL DE LA SOLUTION



5. PLANNING PREVISIONNEL

Répartition générale des rôles

- **A** : Développement **Backend + API** (authentification, gestion des médias, streaming).
- **B** : Développement **Frontend Web** (interface utilisateur, navigation, player).
- **C** : **Intégration & DevOps** (Docker, gestion stockage NAS, documentation, tests).

Tableau planning détaillé

Semaine	Étape / Objectif	Tâches prévues	Responsable(s)	Livrable
S36–S37	Idée et cadrage	<ul style="list-style-type: none"> - Validation de l'idée - Projets similaires (Plex, Jellyfin, etc.) - Étude de faisabilité rapide 	Moi	Compte rendu idée
S38–S39	Conception & CDC	<ul style="list-style-type: none"> - Rédaction cahier des charges (individuel) - Schéma architecture du projet - Choix techno (Node.js, React, etc.) 	Moi	Cahier des charges (28/09)
S40	Mise en place environnement	<ul style="list-style-type: none"> - Création repo GitHub- Setup projet Docker (base container) - Préparation NAS/dossier test 	C	Repo + Docker de base
S41–S42	Développement Backend (MVP)	<ul style="list-style-type: none"> - API utilisateurs (auth, rôles) - Scan dossier multimédia (films/séries/musiques/photos) - Gestion BDD (SQLite/MySQL) 	A	API backend simple
S42–S43	Développement Frontend (MVP)	<ul style="list-style-type: none"> - Maquettage interface web - Authentification + navigation simple - Première intégration avec API 	B	Interface web prototype
S44	Intégration Docker + tests initiaux	<ul style="list-style-type: none"> - Intégrer API + Front dans container - Connexion au stockage (dossier NAS simulé) 	C + A	Docker + test

Semaine	Étape / Objectif	Tâches prévues	Responsable(s)	Livrable
S46	Rapport intermédiaire	- Présentation 3 slides (travail réalisé, difficultés, suite) - Démo API + interface basique	Tous	Présentation 3 slides
S47-S48	Fonctionnalités avancées Backend	- Intégration API TMDB (films/séries) - Intégration API MusicBrainz (musiques) - Lecture EXIF photos	A	Backend enrichi
S48-S49	Fonctionnalités Frontend	- Player vidéo/audio HTML5 intégré - Galerie photo (classement par date/lieu) - Affichage métadonnées (affiches, résumés, etc.)	B	Front enrichi
S49-S50	Intégration & Démo intermédiaire	- Démo streaming fluide- Galerie photos opérationnelle	Tous	Présentation 3 slides
S51-S52	Finalisation projet	- Optimisation performance (caching, requêtes API) - Documentation technique + utilisateur - Tests sur plusieurs navigateurs/appareils	Tous	Rapport final (15/01)
S3-S4	Présentation & JPO	- Préparer PPT + démo - Présentation finale + JPO	Tous	PPT + démo finale (7/02)

Répartition de la charge horaire (environ 100 h au total)

- Étudiant A (Backend/API) → 35 h
- Étudiant B (Frontend/UI) → 35 h
- Étudiant C (Docker, intégration, doc/tests) → 30 h

6. RESSOURCES MOBILISEES

- **Cours et acquis des années précédentes :**
 - Virtualisation et conteneurisation (Docker).
 - Développement web (HTML, CSS, JavaScript).
 - Protocoles réseaux et services (HTTP, streaming).
 - Bases de données et API.
- **Ressources externes :**
 - Documentation Docker.
 - API The Movie Database (films/séries).
 - API MusicBrainz (musique).
 - Librairies EXIF (Node.js).
- **Intervenants / encadrants :**
 - Enseignant référent projet : **M.BEYSSON**
 - Enseignant spécialisé en réseaux : (M.KROMENACKER)