

# 182 – Réaliser la sécurité informatique d'un système

## Rapport personnel

Romain Benedetti

Module du  
01.02.2024 au 07.03.2024

Date de création : 01.02.2024

Version 1 du 01.02.2024



# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                    | <b>3</b>  |
| 1.1      | Objectifs du module                    | 3         |
| <b>2</b> | <b>Analyse</b>                         | <b>4</b>  |
| 2.1      | Présentation du projet "TourisTunes"   | 4         |
| 2.2      | Uses Case                              | 5         |
| 2.3      | Maquettes                              | 6         |
| 2.3.1    | Non connecté                           | 6         |
| 2.3.2    | Non connecté – Listening music         | 6         |
| 2.3.3    | Connecté                               | 7         |
| 2.3.4    | Connecté – Listening music             | 7         |
| 2.3.5    | Upload songs                           | 8         |
| 2.3.6    | My profile                             | 8         |
| 2.4      | Diagramme activité                     | 9         |
| 2.5      | Diagramme de séquences systèmes        | 10        |
| 2.6      | Schéma ER                              | 12        |
| 2.6.1    | Identification des entités principales | 12        |
| 2.6.2    | Relations entre les entités            | 13        |
| <b>3</b> | <b>Conception</b>                      | <b>14</b> |
| 3.1      | Diagrammes de classe                   | 14        |
| 3.1.1    | Client                                 | 14        |
| 3.1.2    | Serveur                                | 15        |
| 3.2      | Schéma relationnel                     | 15        |
| 3.3      | Diagramme séquence interactions        | 16        |
| 3.4      | Conception des tests                   | 16        |
| <b>4</b> | <b>Implémentation</b>                  | <b>17</b> |

# Table des matières

|     |                      |    |
|-----|----------------------|----|
| 4.1 | Descente de code     | 17 |
| 4.2 | Problèmes rencontrés | 17 |
| 4.3 | Tests fonctionnels   | 17 |
| 4.4 | Hébergement          | 17 |
| 5   | Synthèse             | 18 |
| 6   | Conclusion           | 19 |

---

## 1 Introduction

---

### 1.1 Objectifs du module

- Principes de conception d'une application cliente Web.
- Principes de conception d'une application serveur Web.
- Utilisation de la norme UML pour un projet internet Client-Serveur
- Connexion entre des applications clientes et serveur Web.
- Connexion d'application serveur sur une base de données Web.
- Mise en service des principes HTML/CSS/JS (Ajax, JQuery) - PHP0.
- Utilisation des sessions dans l'application serveur Web.
- Utilisation des Cookies dans l'application cliente.
- Utilisation du cryptage des données sensibles (mot de passe)
- Protection contre les injections SQL
- Utilisation des transactions SQL
- Consommation de Webservices dans une application Web.
- Découverte et mise en pratique de standards pour les documentations de code (PHPDoc et JSDoc)
- Protocole et requêtes http de base.
- Hébergement et test final en production de l'application cliente-serveur



## 2 Analyse

---

### 2.1 Présentation du projet "TourisTunes"

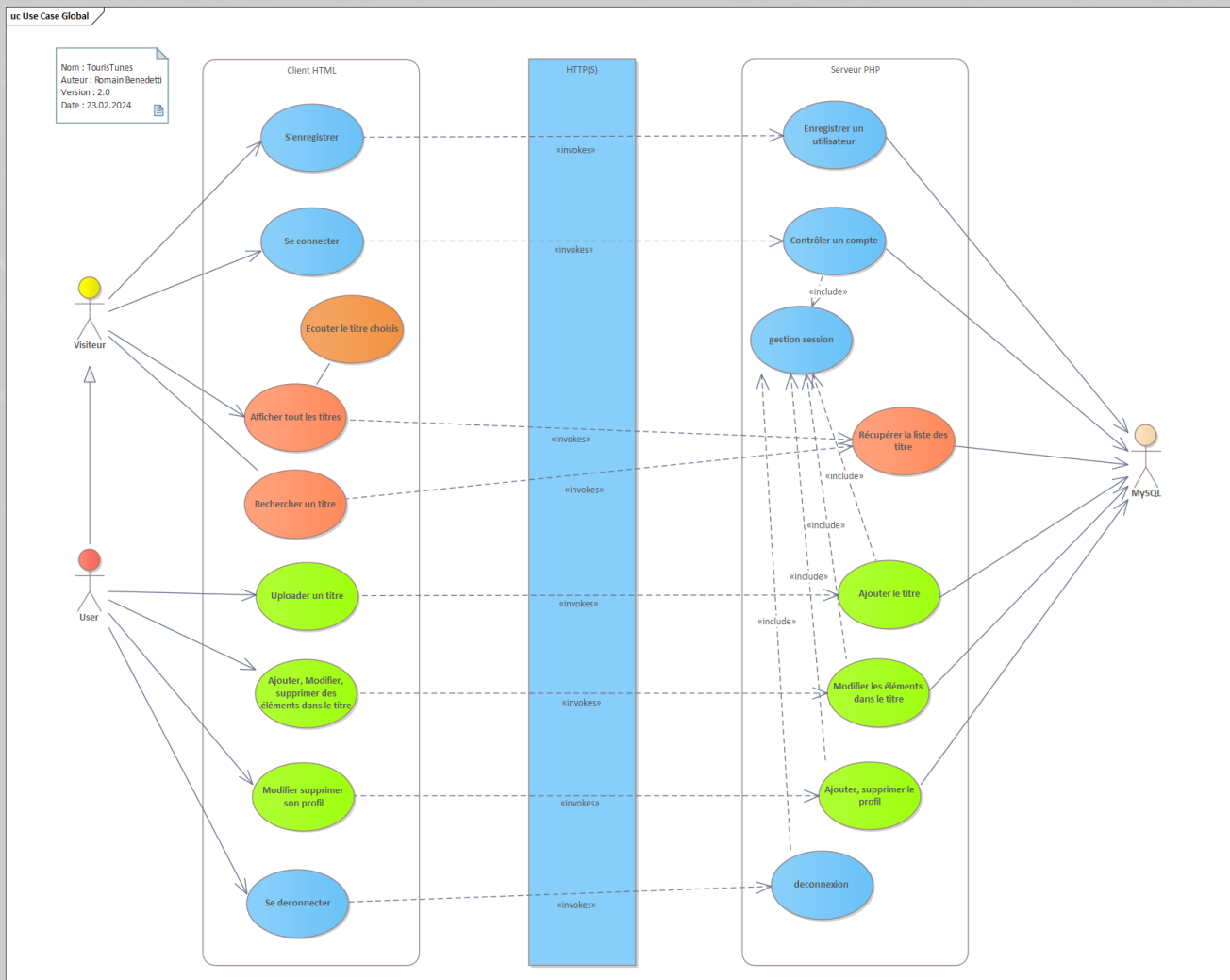
Mon projet de plateforme en ligne tire son inspiration de SoundCloud, mais avec une intention particulière : offrir aux amateurs de musique un espace dédié à la découverte et au partage de créations de sons. Les utilisateurs pourront créer un compte sur cette plateforme afin d'uploader leurs propres chansons et de les partager avec la communauté.

L'objectif principal est de créer une interface facile à utiliser et conviviale où les artistes émergents et les amateurs de musique pourront se connecter, explorer et écouter un large éventail de genres musicaux. Les visiteurs auront la possibilité d'explorer la bibliothèque de titres disponibles et d'apprécier les talents des artistes présents sur la plateforme, même sans être connectés.

Ceux qui s'inscrivent auront accès à des fonctionnalités supplémentaires telles que la possibilité de communiquer avec eux en laissant des commentaires et des likes sur leurs chansons.

En tant que producteur de musique, cette plateforme me permettra de partager mes propres créations et de recevoir des commentaires positifs d'autres artistes et auditeurs.

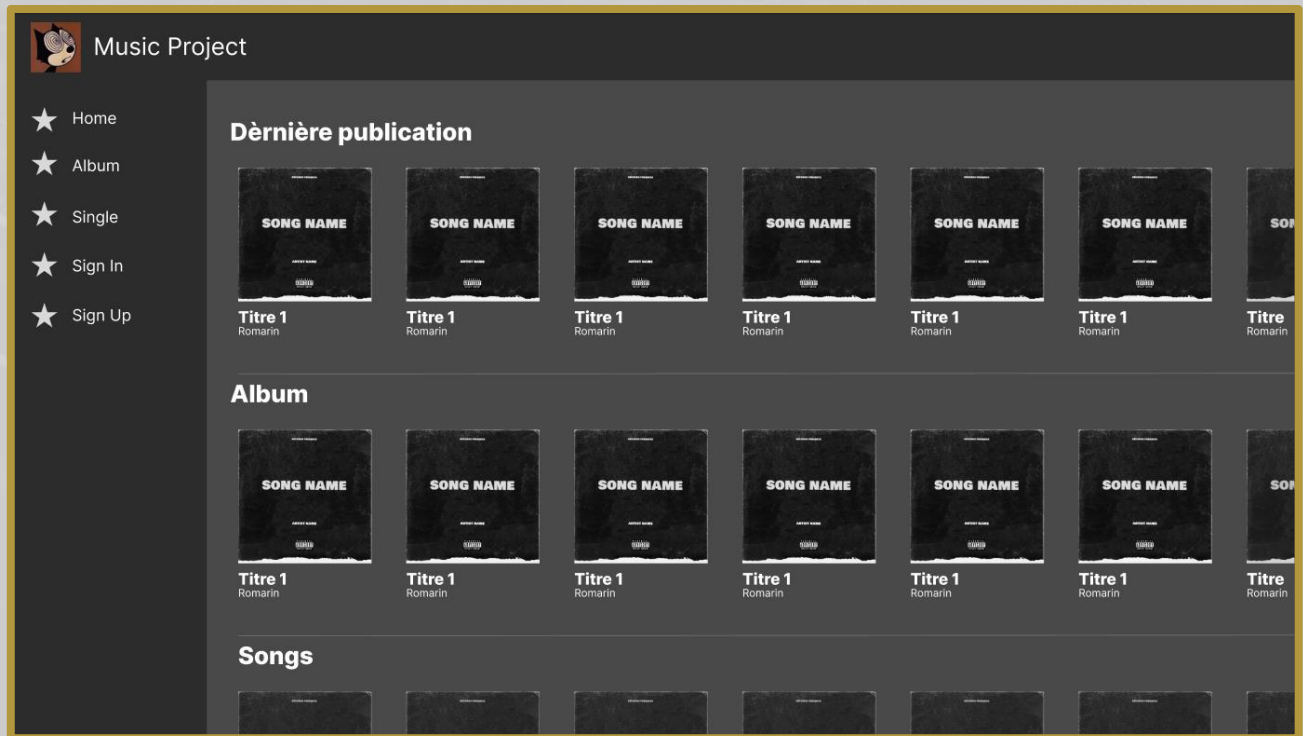
## 2.2 Uses Case



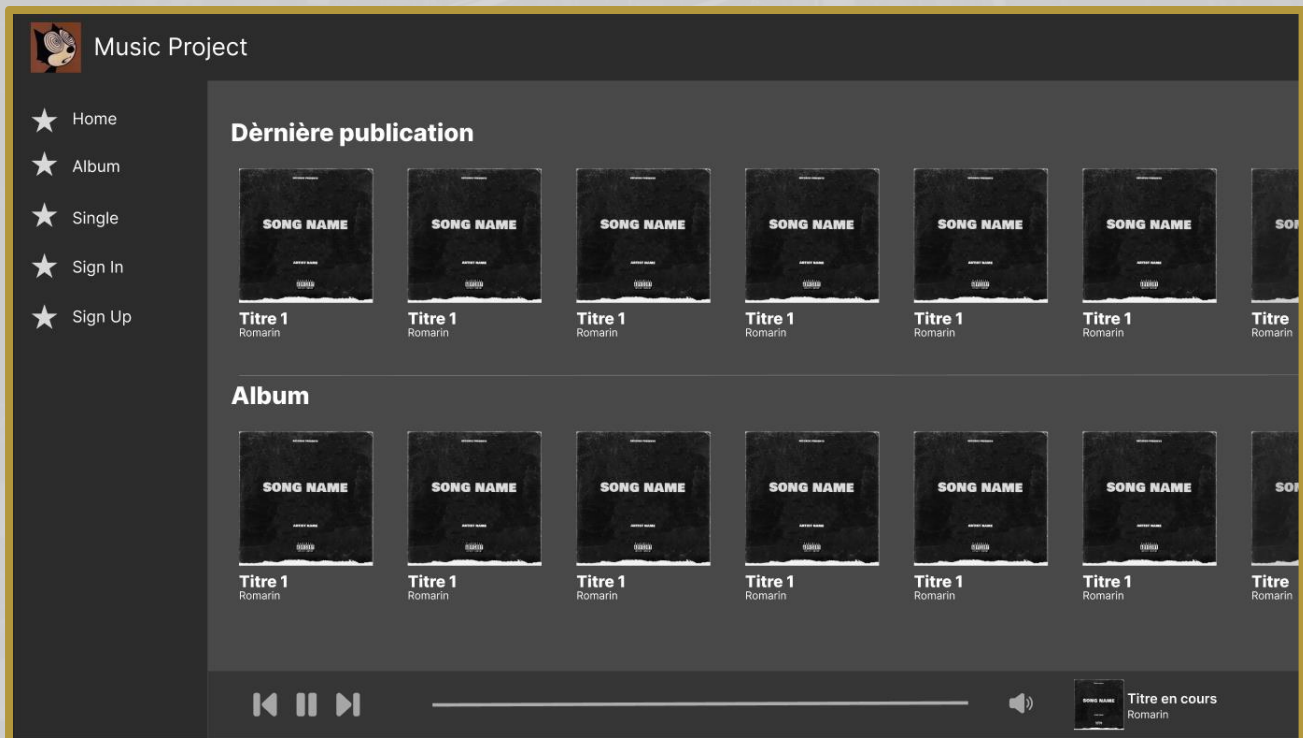
Dans ce use case on voit toutes les actions principales du côté client et côté serveur.

## 2.3 Maquettes

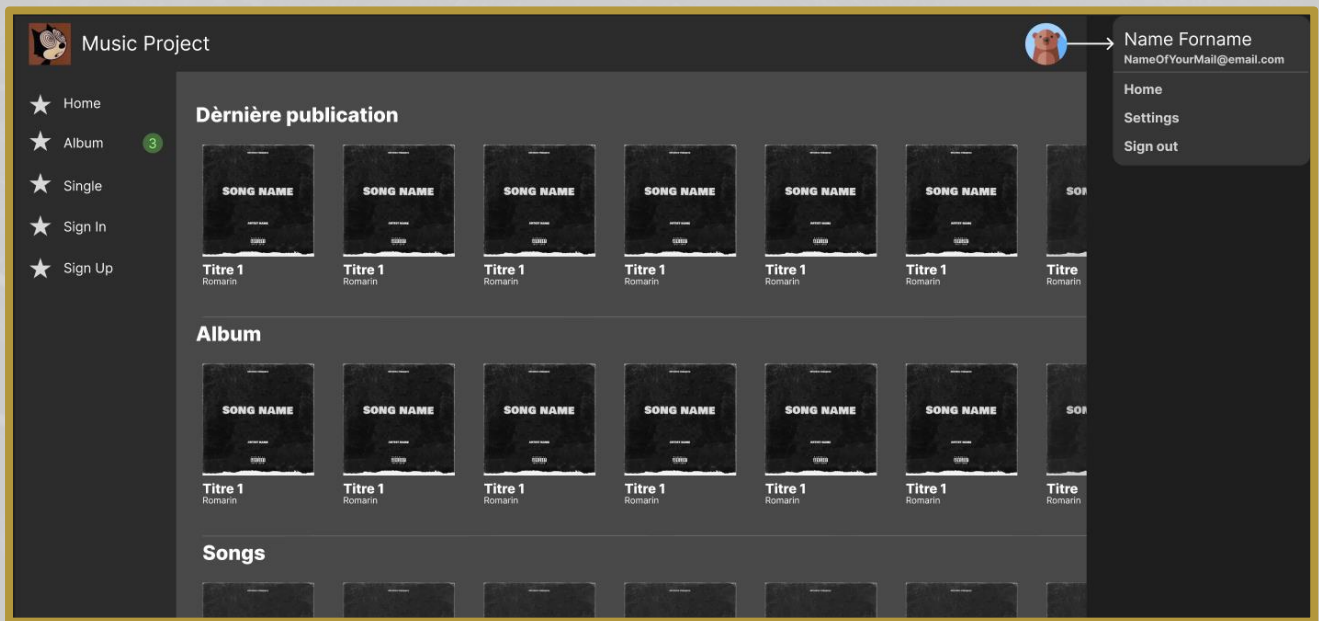
### 2.3.1 Non connecté



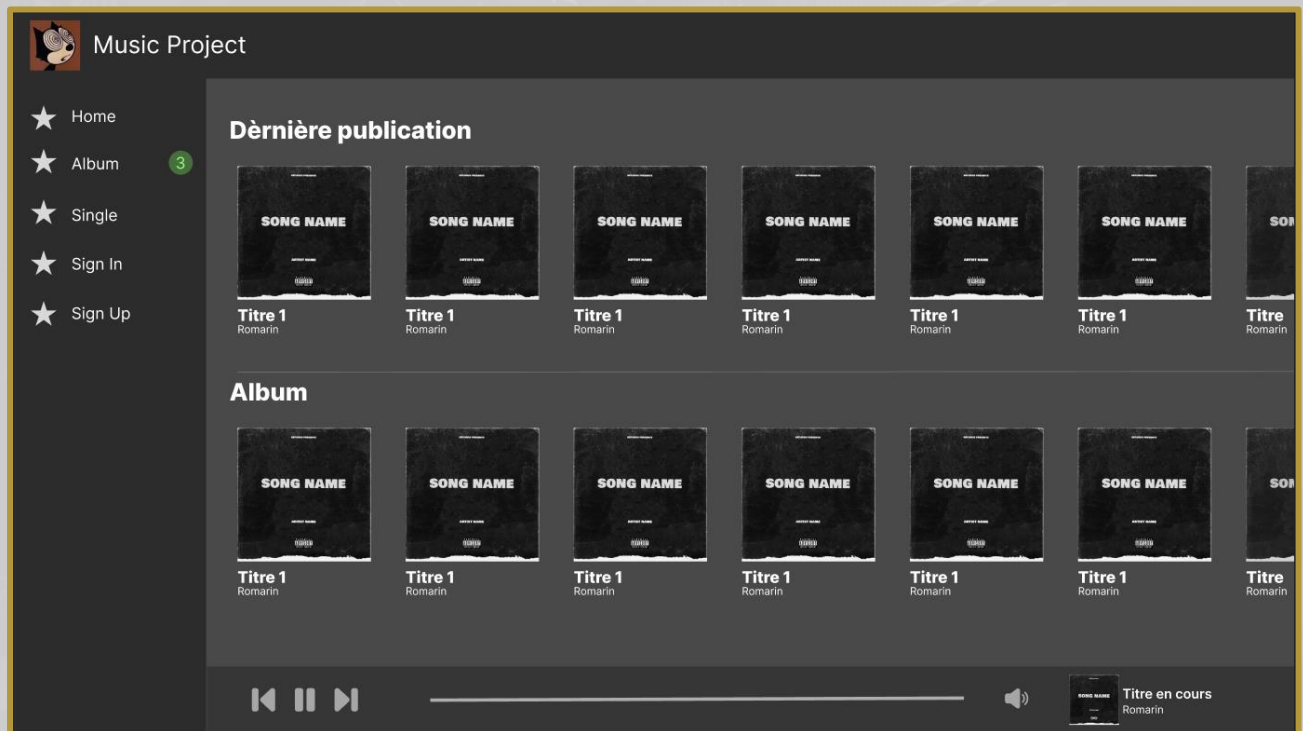
### 2.3.2 Non connecté – Listening music



## 2.3.3 Connecté

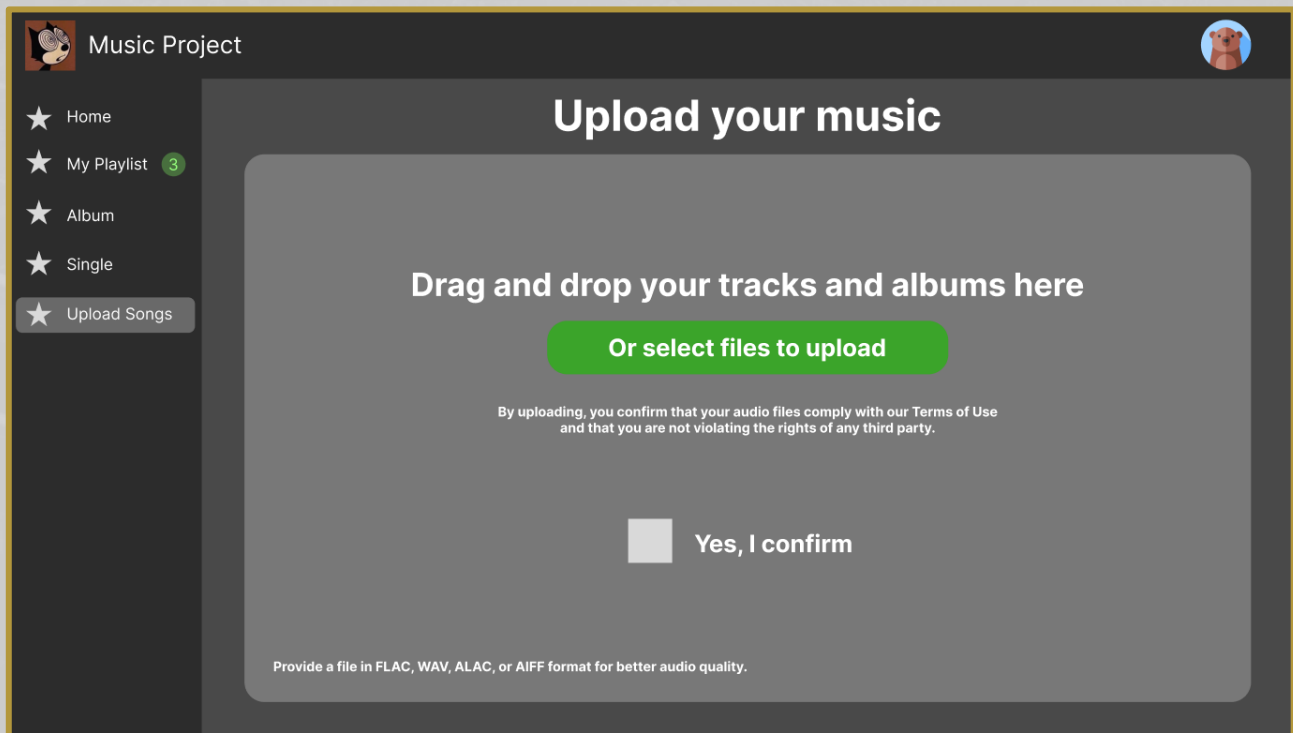


## 2.3.4 Connecté – Listening music

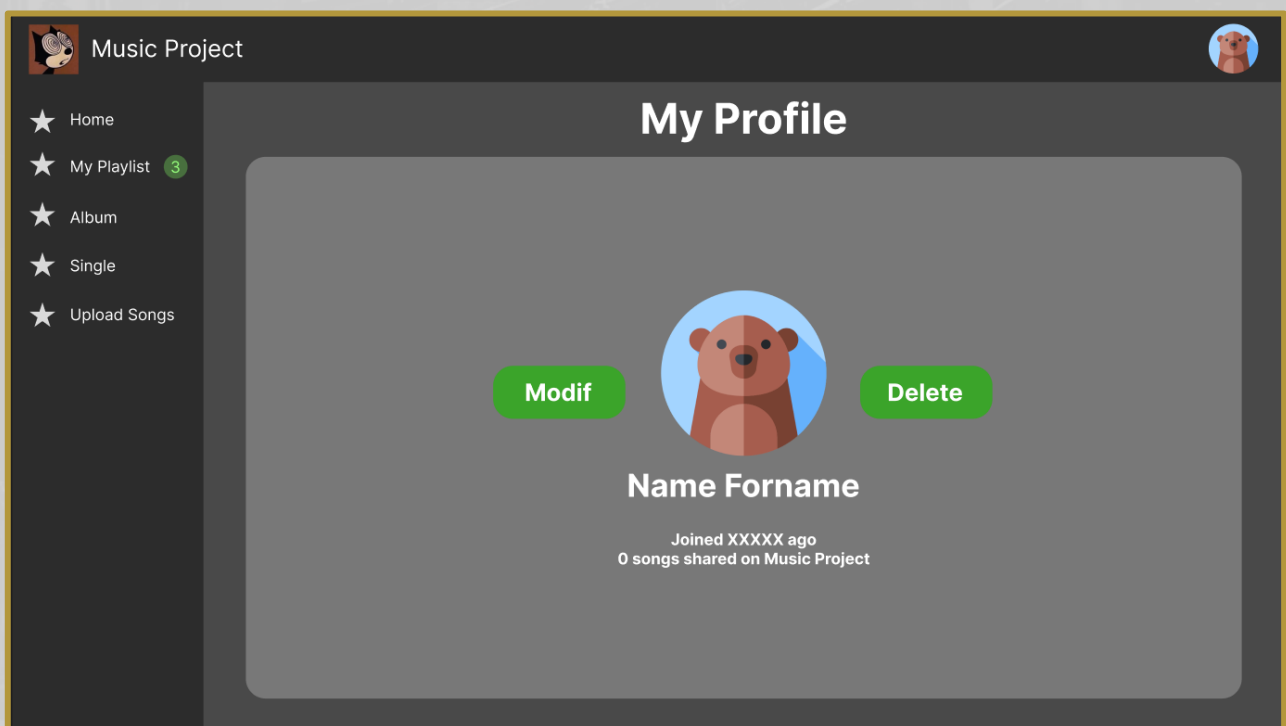




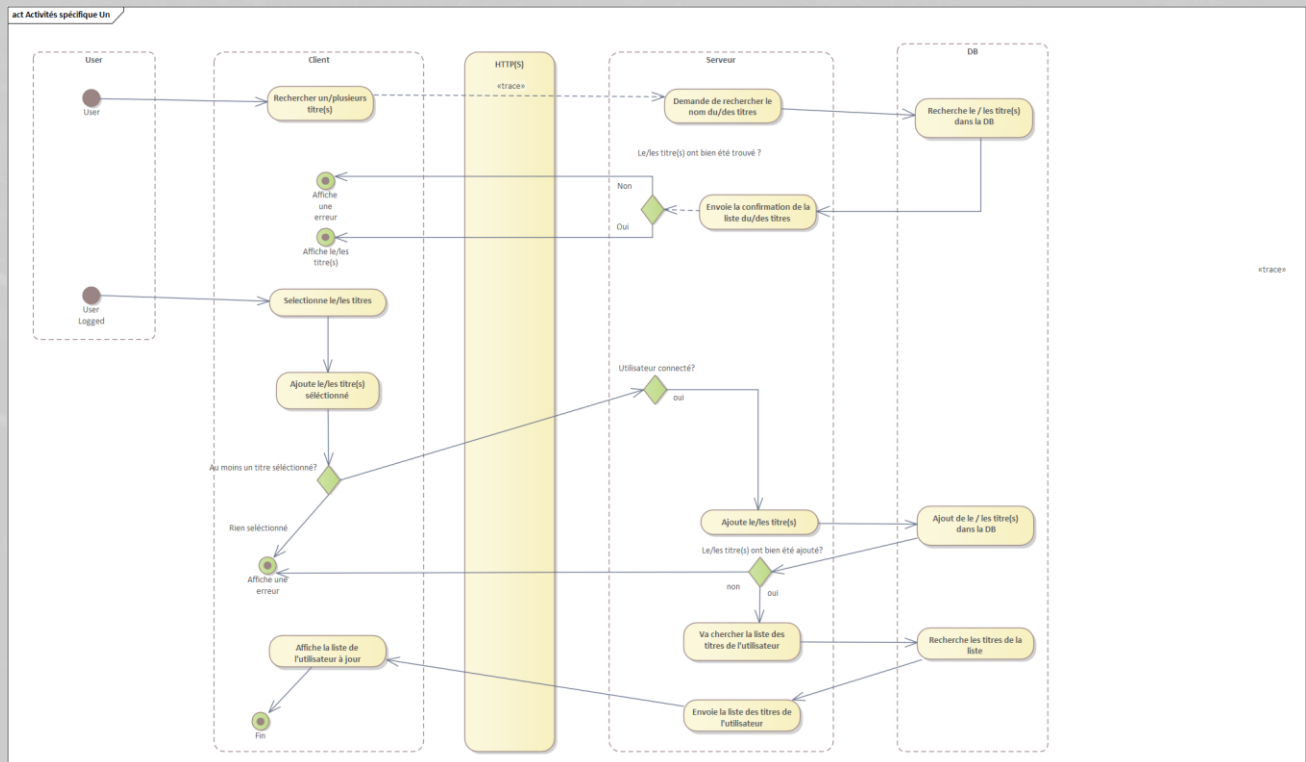
## 2.3.5 Upload songs



## 2.3.6 My profile

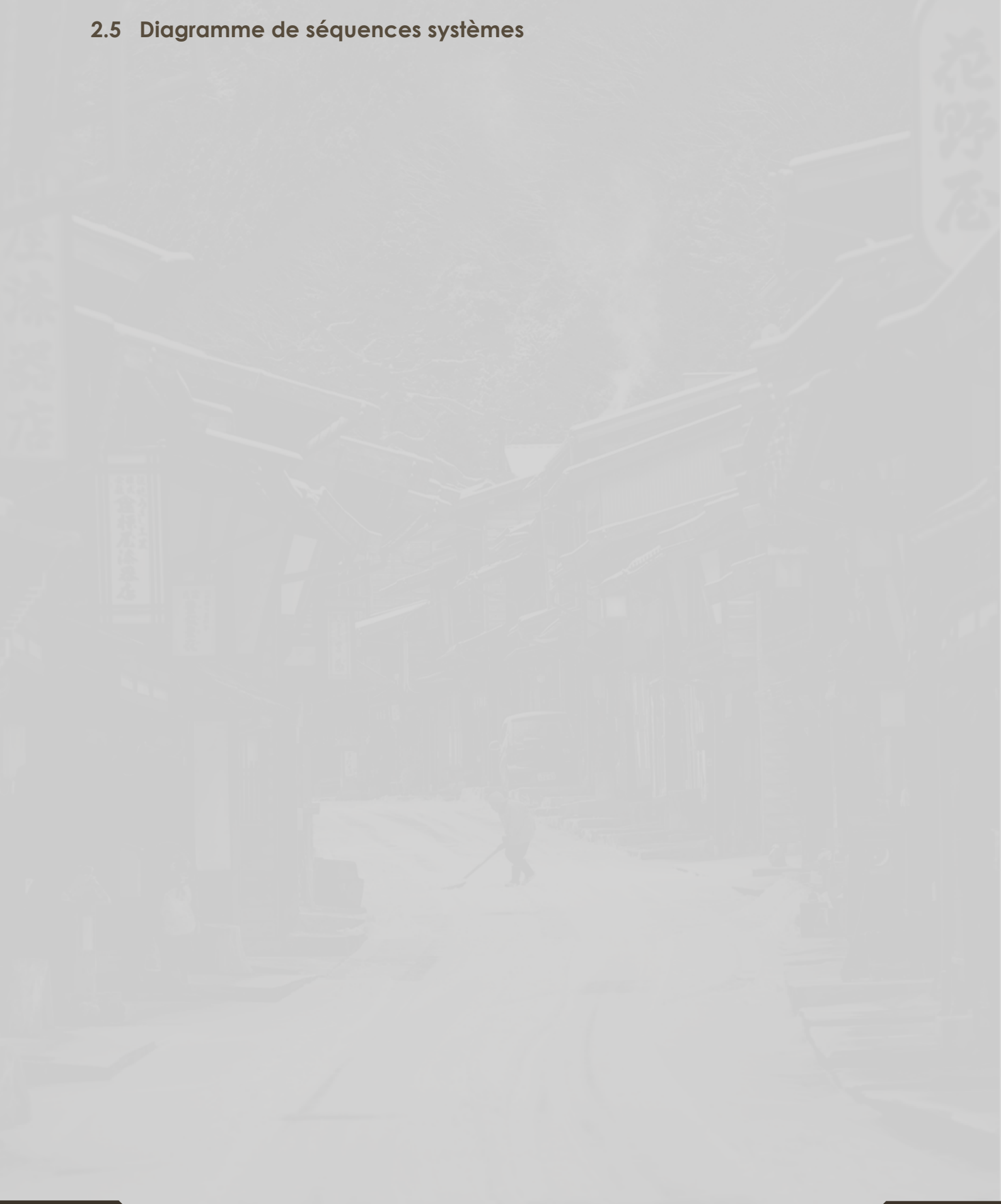


## 2.4 Diagramme activité



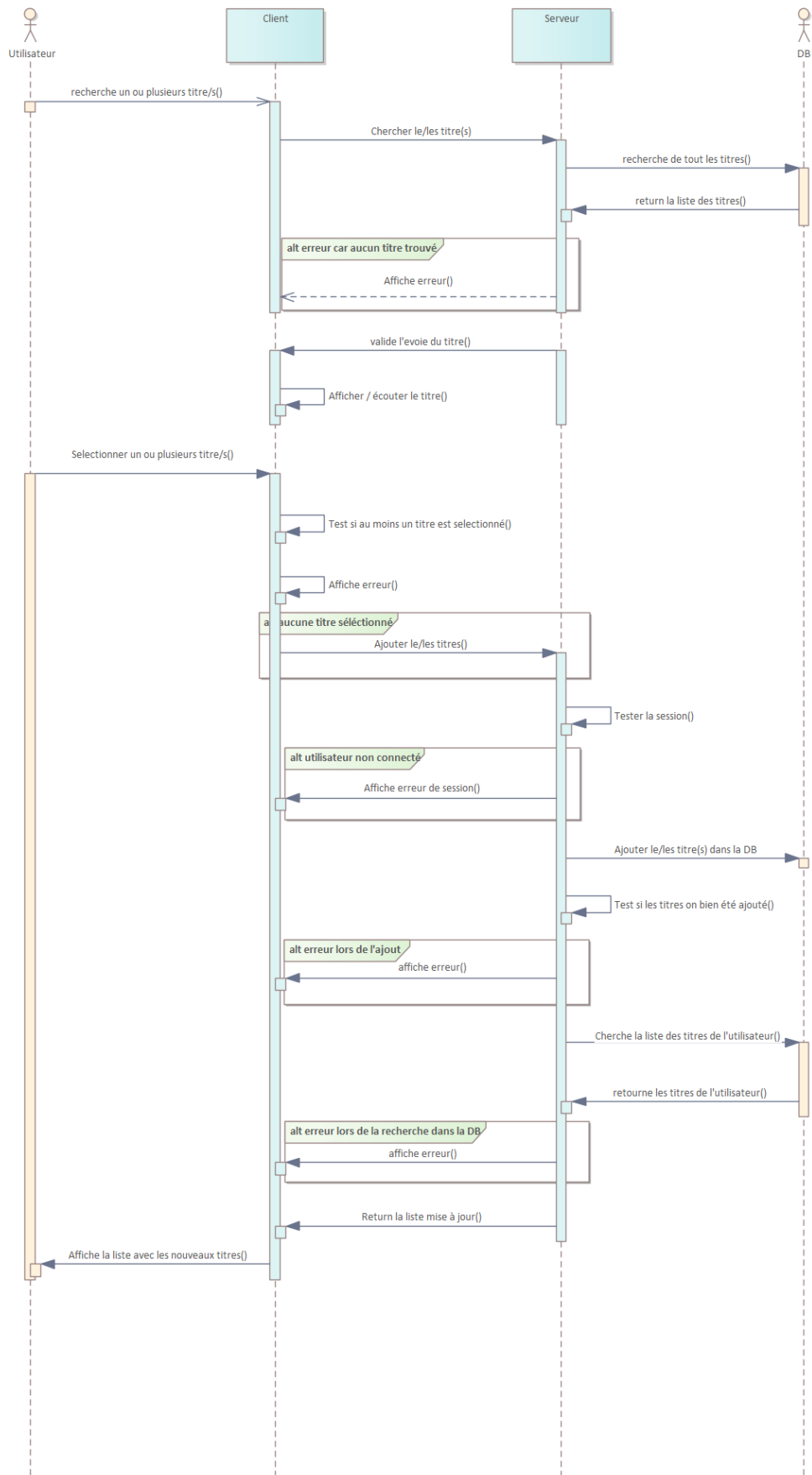
Ce diagramme d'activité montre la même chose que le diagramme de séquence mais d'une autre façon qui nous permet de mieux voir quelle partie (client ou serveur) fait quoi.

## 2.5 Diagramme de séquences systèmes



## sd Séquences spécifique Un

Nom : TourisTunes  
Auteur : Romain Benedetti  
Version : 2.0  
Date : 23.02.2024



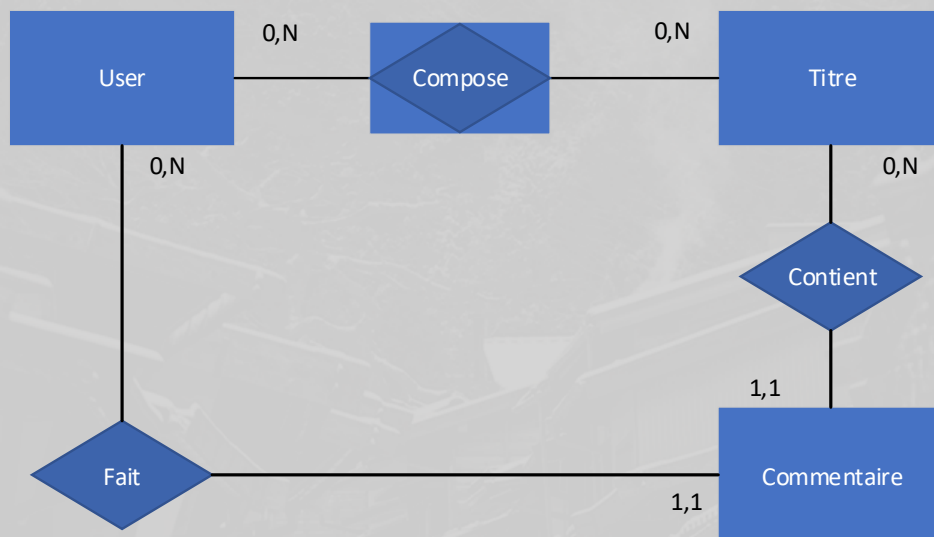


Dans ce diagramme de séquence system on voit deux actions :

Une première action va aller chercher tous les titres qui se trouve dans la db.

La seconde montre le fait d'ajouter un / des titres dans la liste de l'utilisateur. Ce dernier peut sélectionner un ou plusieurs titres puis cliquer sur ajouter pour les ajouter dans sa liste de sons. Pour finir on affiche à nouveau la liste avec les nouveaux titres.

## 2.6 Schéma ER



### 2.6.1 Identification des entités principales

⇒ Utilisateur :

- ↗ ID Utilisateur
- ↗ Nom d'utilisateur
- ↗ Mot de passe
- ↗ Adresse e-mail
- ↗ Autres informations utilisateur (facultatives)

⇒ Titre :

- ↗ ID Titre
- ↗ Nom du titre
- ↗ Durée
- ↗ Format
- ↗ Lien vers le fichier audio
- ↗ Lien vers l'image
- ↗ Date de publication

- ↗ Featurings (si un titre a des collaborations)
- ↗ Autres métadonnées du titre
- ⇒ Fonctionnalités supplémentaires :
  - ↗ Commentaires
  - ↗ Likes
  - ↗ Données de lecture (combien de fois le titre a été écouté)

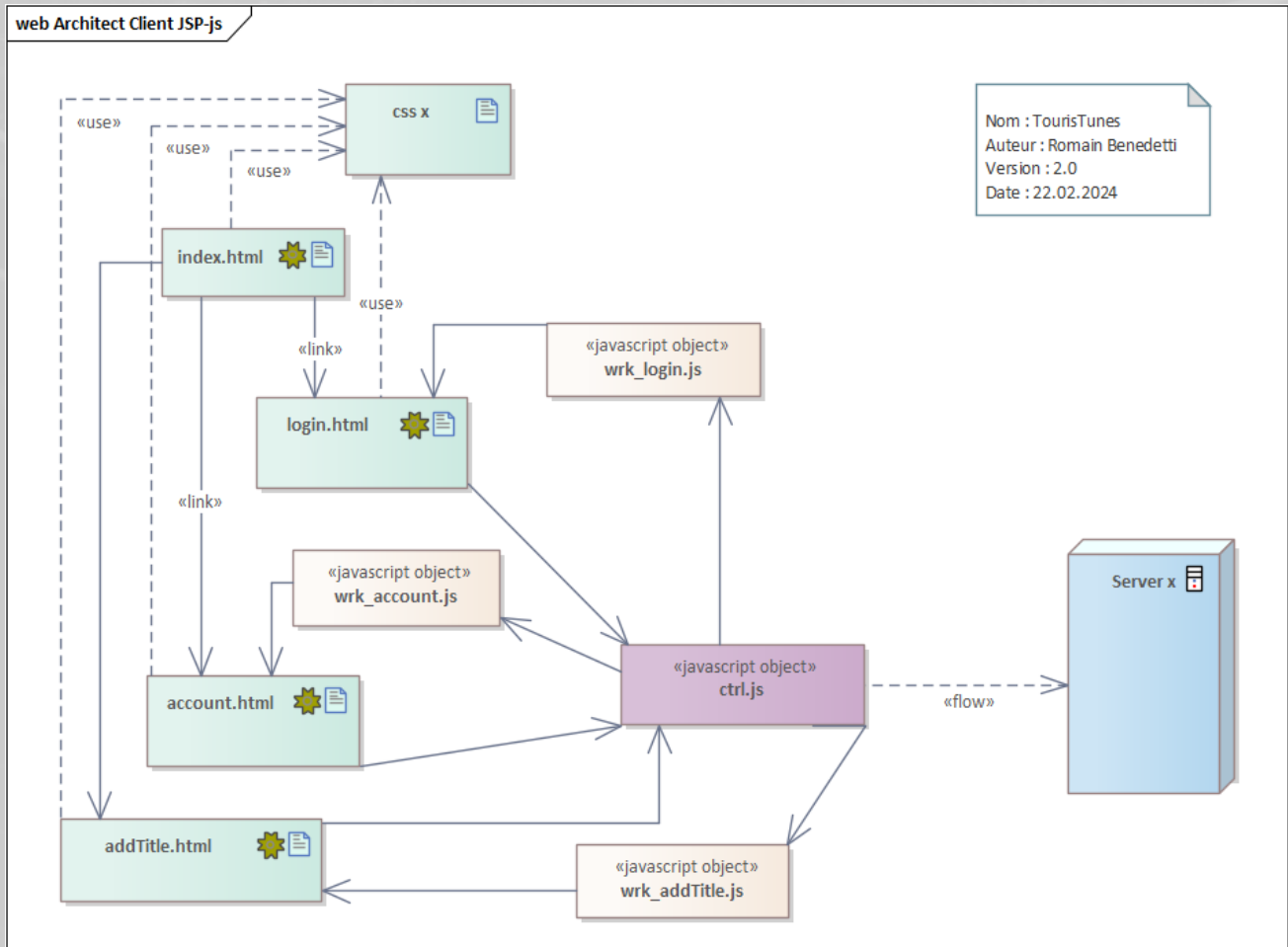
### 2.6.2 Relations entre les entités

| Coté 1  | Coté 2   |
|---|--|
| Un utilisateur peut composer plusieurs titres                 | Un titre est composé par plusieurs utilisateurs              |
| Un titre peut contenir plusieurs commentaires, likes, etc,    | Un commentaire, like, etc., est contenu par un seul titre    |
| Un utilisateur peut avoir plusieurs commentaires, likes, etc, | Un commentaire, like, etc., est fait par un seul utilisateur |

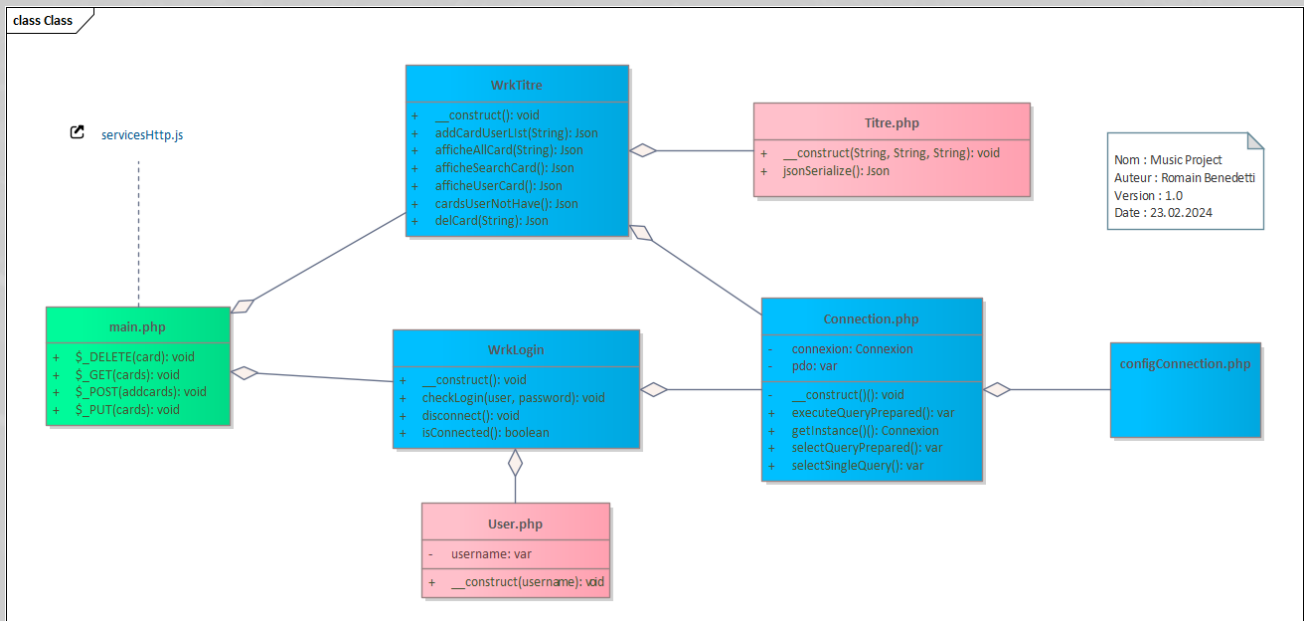
## 3 Conception

### 3.1 Diagrammes de classe

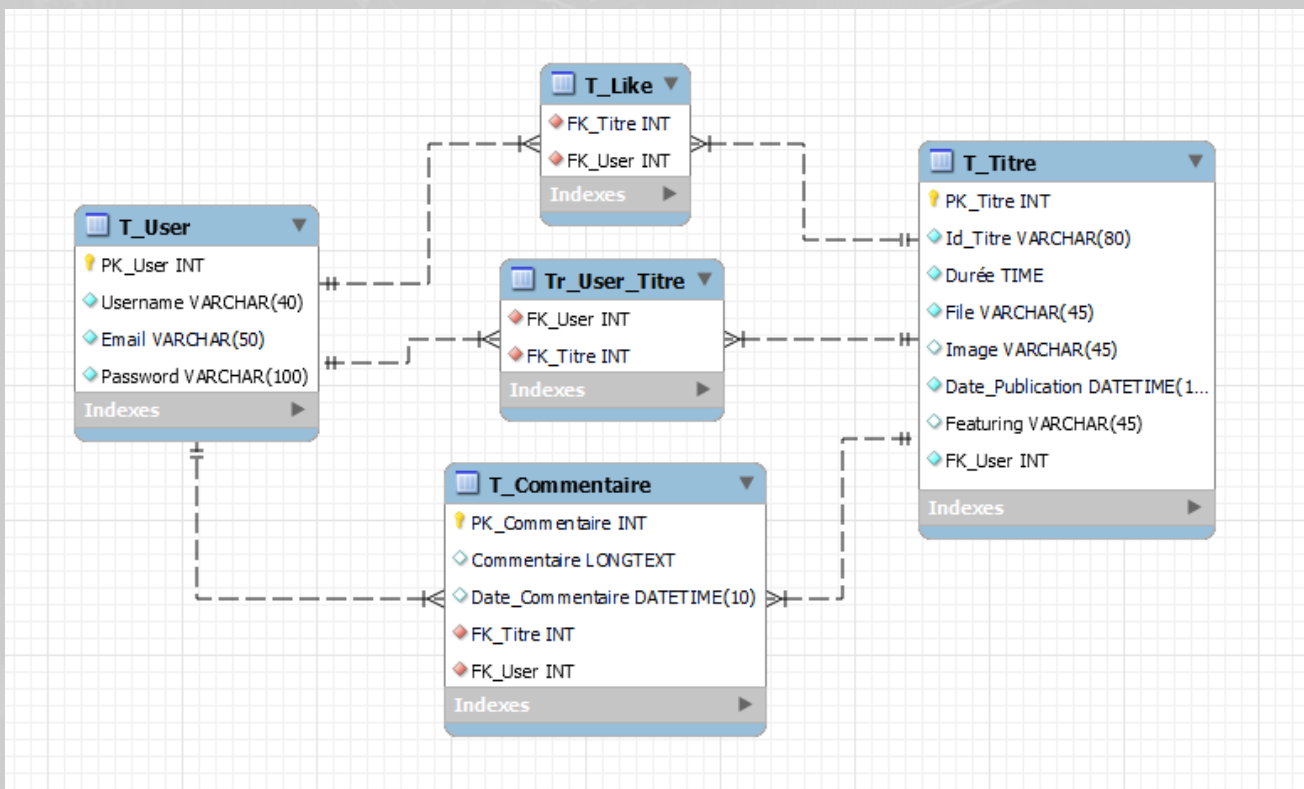
#### 3.1.1 Client



## 3.1.2 Serveur

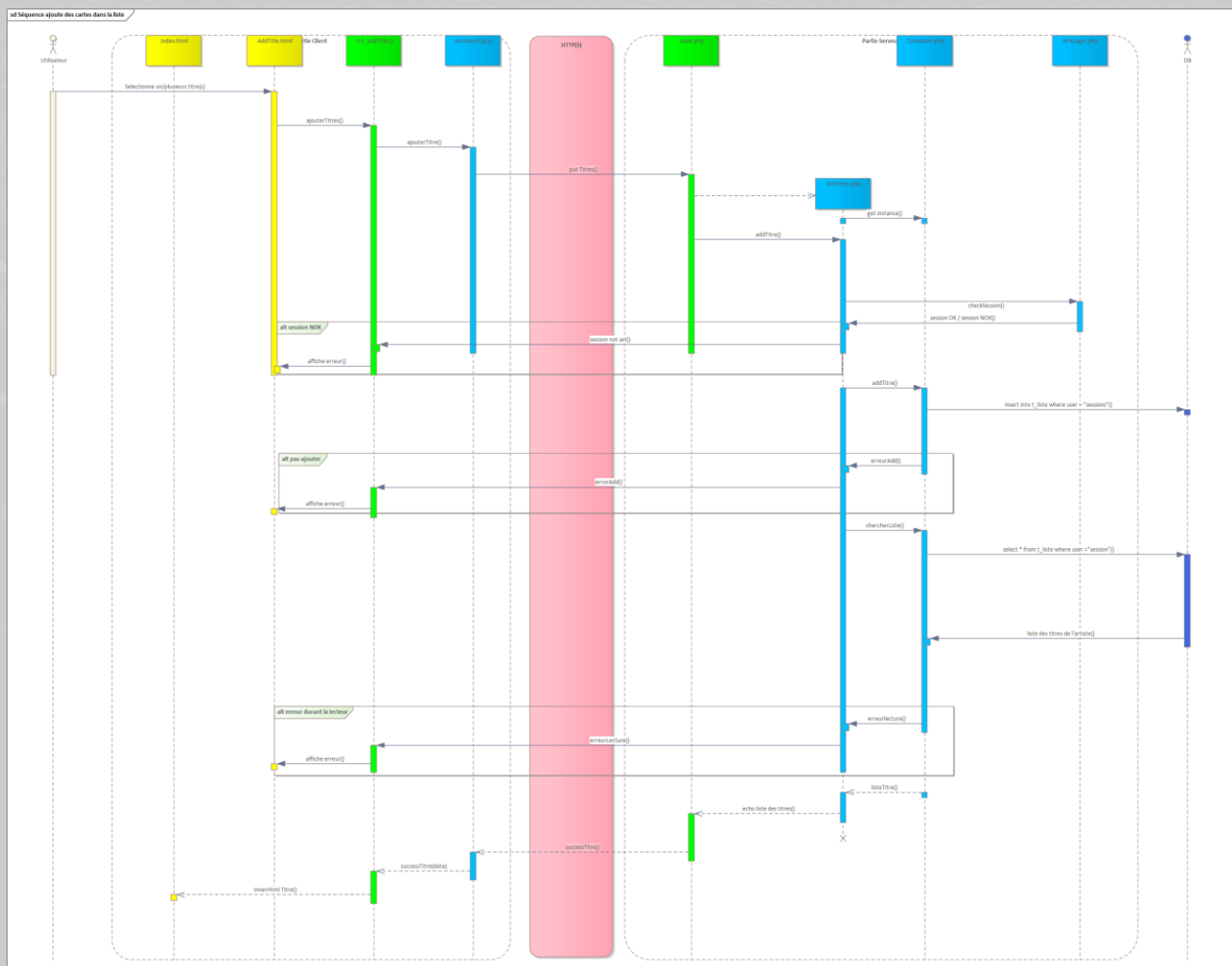


## 3.2 Schéma relationnel





## 3.3 Diagramme séquence interactions



## 3.4 Conception des tests

| Utilisateur    | Cas       | Tests   | Attendu   | Obtenu |
|----------------|-----------|---|---|--------|
| Visiteur       | Connexion | Avec login existant et mot de passe correct.  | On est connecté en tant qu'artiste                          |        |
|                | Connexion | Avec un login qui contient une injection SQL. | L'injection ne fonctionne pas, le login n'est pas possible. |        |
|                | Recherche | Rechercher une musique                        | Recherche possible, le/les titre(s) s'affiche               |        |
|                | Afficher  | Afficher les musiques de l'utilisateur        | Affichage possible utilisateur correct                      |        |
| User (Artiste) | Afficher  | Afficher les musiques de l'utilisateur        | Affichage possible utilisateur correct                      |        |
|                | Ajouter   | Ajouter une musique dans la liste             | Ajout possible utilisateur connecté                         |        |
|                | Supprimer | Supprimer une musique de la liste             | Suppression possible utilisateur connecté                   |        |

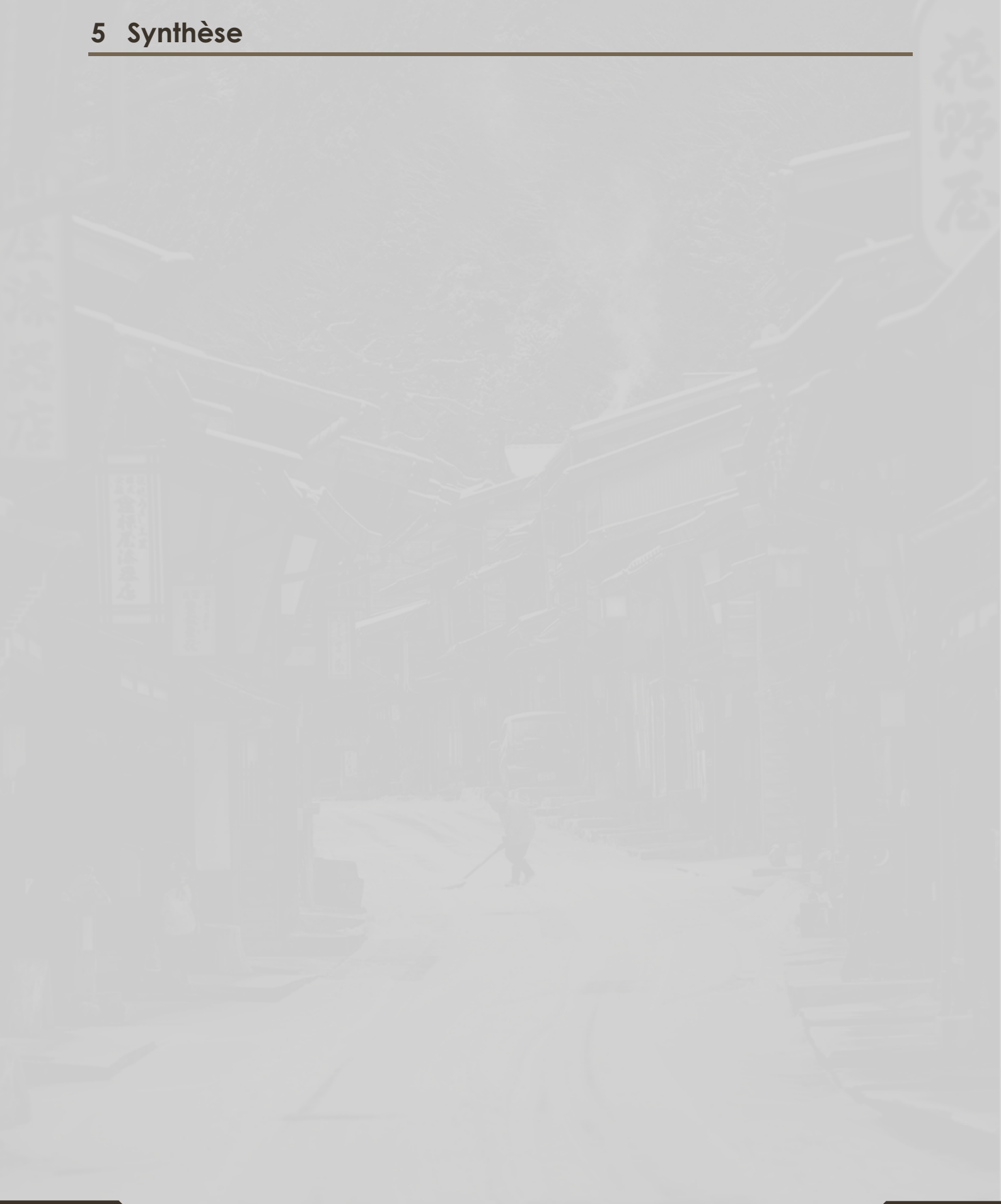
## 4 Implémentation

---

- 4.1 Descente de code
- 4.2 Problèmes rencontrés
- 4.3 Tests fonctionnels
- 4.4 Hébergement

## 5 Synthèse

---



## 6 Conclusion

---