

P_BitRuisseau



Berchel Joachim – MID3
Vennes
08.11.2024 - 12.01.2025
Jonathan Melly

Table des matières

| | | |
|----------|---|----------|
| 1 | INTRODUCTION | 3 |
| 2 | PLANIFICATION INITIALE..... | 4 |
| 3 | JOURNAL DE TRAVAIL..... | 6 |
| 4 | RAPPORT DE TEST..... | 7 |
| 5 | ÉTAT DES LIEUX ET CONCLUSION | 8 |
| 5.1 | ÉTAT DU PROJET | 8 |
| 5.2 | PISTE D'AMÉLIORATION..... | 8 |
| 5.3 | CONCLUSION PERSONNELLE..... | 8 |
| 6 | UTILISATION DE L'IA..... | 9 |

1 INTRODUCTION

Ce projet est effectué dans le cadre du module « 321 Programmation distribué ». L'objectif de ce projet est de vérifier les acquis théoriques du module par un projet pratique.

L'objectif de ce projet est de réaliser une médiathèque de musique pouvant être recherché et téléchargé depuis une autre médiathèque en utilisant le protocole MQTT.

Ce projet est accessible depuis GitHub au lien :

« <https://github.com/joaberch/joachim-MonProjetGenial> »

2 PLANIFICATION INITIALE

La planification a été faite sur GitHub Projects, elle est accessible à ce lien :

« <https://github.com/users/joaberch/projects/3/views/1> »

Les maquettes montrent une approximation de l'application et montrent la logique technique de MQTT dans l'application.

Ma médiathèque

Chercher une musique

Bibliothèque locale - pas de MQTT

| | | |
|-----------|--|--|
| Musique A | | Auteur Taille Status Progrès Nombre de grain |
| Musique B | | Auteur 20MB Grainé 100% 0/10 |
| Musique C | | Auteur 825MB Téléchargement 84% 3/8 |
| Musique D | | Auteur 5MB Fini 100% 0/14 |
| Musique E | | Auteur Taille Status Progrès Nombre de grain |

Ajouter une musique

Sélectionner le réseau

1 - Menu d'accueil - Affichage du catalogue local

Ma médiathèque

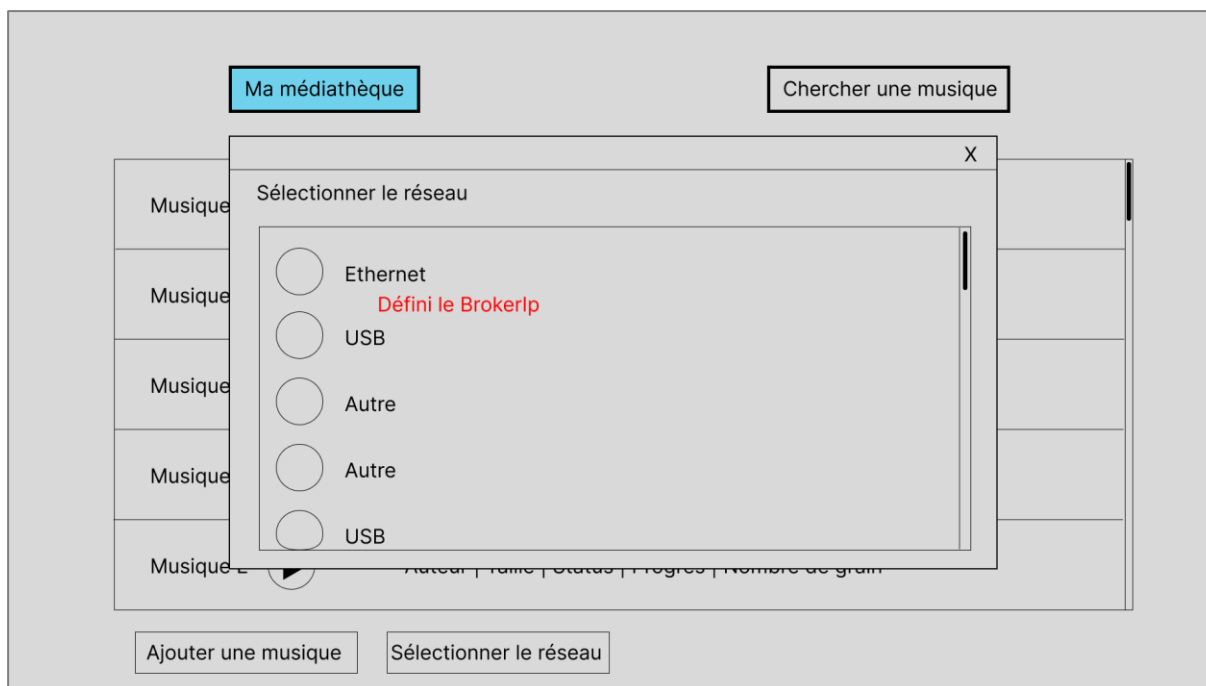
Chercher une musique

Demande de catalogue MQTT

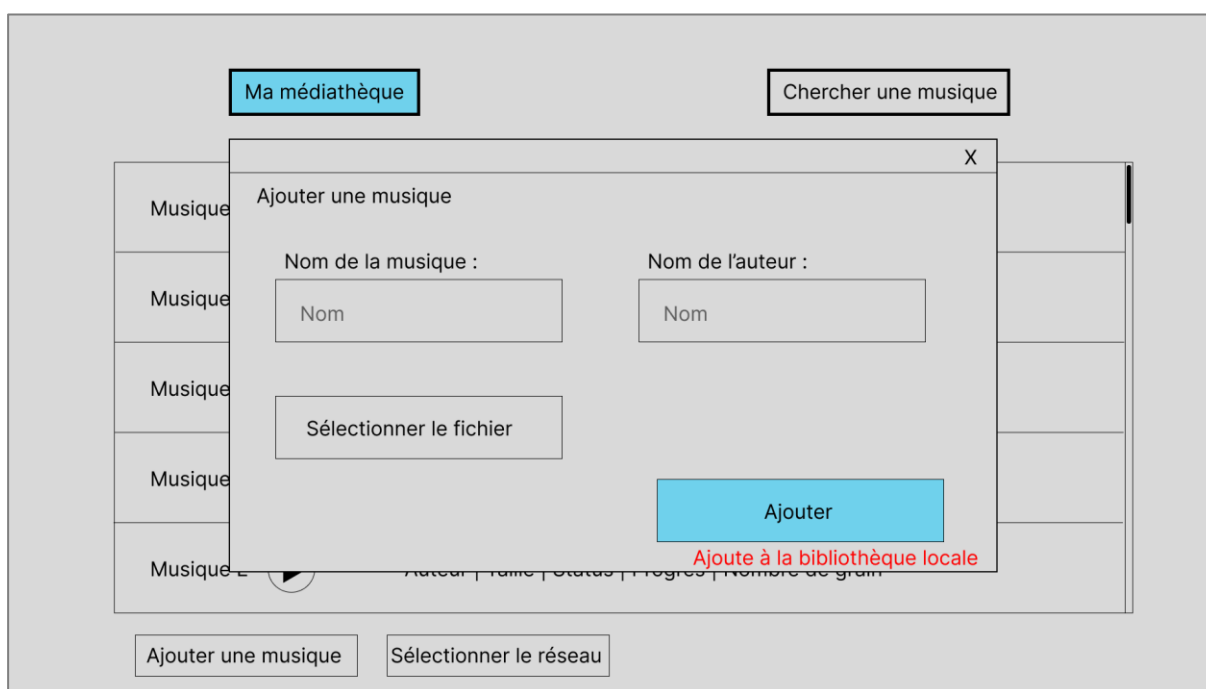
Rechercher

| Nom | Auteur | |
|----------------|--------------|--|
| Soldier's Joy | Robert Burns | <div>Télécharger</div> |
| 24K Magic | Bruno Mars | <div>Télécharger</div> |
| Cheek to Cheek | Fred Astaire | <div>Demande de fichier MQTT Télécharger</div> |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

2 - Recherche de musique, affichage des catalogues sur le réseau



3 - Sélection du réseau



4 - Ajout de musique dans la bibliothèque locale

3 JOURNAL DE TRAVAIL

Pour réaliser mon journal de travail j'ai utilisé un classeur Excel :

[Journal-de-Travail_Berchel-Joachim.xlsx](#)

J'ai aussi formaté certains commits GitHub afin de pouvoir générer un journal de travail depuis ces logs. Tous mes commits ne sont pas formatés et si un journal de travail est généré ainsi il manquera certaines parties.

4 RAPPORT DE TEST

Par manque de temps aucun test unitaire n'a été créé.

Afin de tester mon application j'ai malgré tous effectué des tests manuels :

Connexion au broker MQTT :

J'ai vérifié la connexion au broker MQTT en utilisant l'outil MQTTX.

Après la connexion au broker j'envoyais un message de test que je pouvais voir dans MQTTX.

Mon application n'affichant pas les messages sur le broker j'utilise MQTTX pour déboguer les messages envoyés.

Réception et analyse des message :

J'ai testé les requêtes de mon application avec Ethan et avec une duplication de mon programme pour vérifier que mon programme puisse recevoir les messages du même type.

Téléchargement et écoute des musiques :

Pour vérifier que mes fichiers envoyés conservent leurs intégrités j'ai :

1. Écouter le fichier.
2. Ajouter le fichier dans le catalogue de mon application.
3. Lancer une copie de mon application.
4. Demander les catalogues sur la copie et télécharger le fichier.
5. Écouter le fichier récemment téléchargé.

En faisant cela je m'assure que le transfert de fichier est fonctionnel.

5 ÉTAT DES LIEUX ET CONCLUSION

5.1 État du projet

Toutes les requêtes MQTT sont fonctionnels avec les autres programmes utilisant les enveloppes comme défini sur cette page GitHub :

« https://github.com/philippe-hjik/MID3-CID3_DistriAudio/blob/main/README.md »

L'application est donc fonctionnel pour afficher les médias locaux et télécharger les médias distants, elle ne permet pas d'écouter la musique mais les musiques téléchargé gardent toutes leurs intégrité.

5.2 Piste d'amélioration

La première piste d'amélioration concerne l'optimisation du code, le code pourrait être plus optimisé afin de rendre le programme plus fluide pour l'utilisateur.

Une autre piste d'amélioration est l'écoute des musiques depuis l'application, actuellement l'application ne permet pas d'écouter des fichiers audios tel que mp3.

Une autre piste d'amélioration serait de permettre de séquencer les téléchargements, ainsi lors du téléchargement d'une musique plusieurs programmes se chargent d'envoyer des bouts de la musique.

5.3 Conclusion personnelle

J'ai trouvé ce projet très intéressant dans la technique, j'ai l'impression d'avoir amélioré mes compétences en développement C#.

Néanmoins j'ai eu beaucoup de frustration concernant l'aspect « travail en équipe ». La structure des enveloppes devant être la même pour nous tous il fallait donc se mettre d'accord sur la logique à employer.

6 UTILISATION DE L'IA

Actuellement aucune intelligence n'a été utilisé dans le cadre de ce projet.