

Rapport GL40

- poopify.fr -

Université Technologique de Belfort-Montbéliard

GL40 – P20

Groupe :

Antoine CHARMEAU – INFO 2

Marco LITZLER – INFO 2

Théau ZATTI – INFO 2

Responsables :

Jean-Charles Creput

Table des matières

1. Introduction	3
1.1. Choix du sujet	3
1.2. Choix techniques.....	3
1.3. Répartition du travail	3
2. Etudes	4
2.1. Keystroke	4
2.1.1. Spotify	4
2.1.2. Poopify	5
2.2. Structuring display	7
2.2.1. Logscreen.....	7
2.2.2. Structure Générale.....	7
2.2.3. Top 50	8
2.2.4. Playlist.....	8
2.2.5. Profil.....	9
2.2.6. Home	9
2.2.7. Like.....	10
2.2.8. Settings	10
2.3. Actigrammes	10
2.3.1. Ecouter une musique	10
2.3.2. Rechercher/ajouter une musique	11
2.3.3. Supprimer une musique.....	11
2.3.4. Créer un compte Poopify	11
3. Présentation de Poopify.....	12
3.1. Qu'est-ce que Poopify.....	12
3.2. Les différents composants	12
3.2.1. Home	12
3.2.2. Logscreen.....	12
3.2.3. Player-bar.....	13
3.2.4. Playlist.....	13
3.2.5. Profil.....	14
3.2.6. Search-bar	14
3.2.7. Settings	14
3.2.8. Sidenav.....	14
3.2.9. Top.....	15
3.2.10. Like.....	15

3.3. Visuels	15
4. Conclusion	18
4.1. Problèmes rencontrés.....	18
4.2. Améliorations possibles	18
4.3. Conclusion	18

1. Introduction

1.1. Choix du sujet

Pour ce projet, nous avons décidé d'étudier une des plus grandes plateformes de streaming de musique au monde : Spotify. Notre idée est de créer à notre tour une plateforme de streaming de musique mais cette fois ci à travers une WebApp qui permettrait à un grand nombre de personne d'en profiter et ceci sans téléchargement ou autres. Ce sujet est idéal pour l'UV car il mêle aspect technique du développement et étude de quelque chose d'existant.

Le site est visitable à l'adresse suivante :

<https://poopify.fr>

Il est possible d'utiliser le compte suivant :

Adresse email : test

Mot de passe : test

1.2. Choix techniques

Nous avons décidé de partir sur un langage de programmation nouveau pour notre groupe : Angular. Ce choix nous est paru évident car nous avons déjà une expérience web (HTML, CSS, JS) mais nous voulions tester l'implémentation du projet en Angular. Cela nous permet de travailler facilement en aillant réparti les composants entre nous. En plus de cet aspect visuel, nous avons poussé le projet plus loin en implémentant un Back-end grâce une base de données et des apis qui nous permettent de rechercher, stocker des musiques mais aussi de créer des comptes ou des playlists pour chaque utilisateur.

1.3. Répartition du travail

Pour ce qui est de la répartition du travail, nous nous sommes distribué plusieurs composants pour ensuite les mettre en communs.

Marco s'est occupé de la page « Home », « Top » et « Like » avec tous les liens apis pour pouvoir communiquer avec la base de données.

Antoine s'est occupé de mettre en place l'api de lecture de musique et les apis qui se connectent à la base de données. Il a aussi fait le « player » pour pouvoir gérer la musique qui est jouée.

Théau s'est occupé de la navigation avec le « sidenav », de la partie recherche de musique avec l'api Youtube de la « search-bar » et de la page avec les différentes playlists.

2. Etudes

2.1. Keystroke

2.1.1. Spotify

Tout d'abord, nous précisons que nos études se place dans le Home de l'application avec les mains situées sur le clavier.

Rechercher une musique :

H[souris] M P[souris] K[clic] H[clavier] M K[clavier]*n

$$\begin{aligned}\text{On obtient } t_R &= 2t_H + 2t_M + t_P + t_K*(n+1) \\ &= 2*0,4 + 2*1,35 + 1,1 + 0,2*(n+1) \\ &= 4,8 + 0,2*n \text{ secondes}\end{aligned}$$

Lancer une musique depuis sa playlist :

H[souris] M P[souris] K[clic] M P[souris] K[clic]

$$\begin{aligned}\text{On obtient } t_P &= t_H + 2t_M + 2t_P + 2t_K \\ &= 0,4 + 2,7 + 2,2 + 0,4 \\ &= 5,7 \text{ secondes}\end{aligned}$$

Ajouter une musique rechercher :

$t_R + H[souris] M P[souris] K[clic] M P[souris]*2 K[clic]$

$$\begin{aligned}\text{On obtient } t_A &= t_R + t_H + 2t_M + 3t_P + 2t_K \\ &= 4,8 + 0,2*n + 0,4 + 2,7 + 3,3 + 0,4 \\ &= 11,6 + 0,2*n \text{ secondes}\end{aligned}$$

Modifier les paramètres du compte :

H[souris] M P[souris] K[clic] M P[souris] K[clic] M P[souris] K[clic] M P[souris] K[clic] H[clavier] M K[clavier]*n H[souris] M P[souris] K[clic]

$$\begin{aligned}\text{On obtient } t_M &= 3t_H + 6t_M + 5t_P + t_K*(n+5) \\ &= 1,2 + 8,1 + 5,5 + 1 + 0,2*n \\ &= 15,8 + 0,2*n \text{ secondes}\end{aligned}$$

Ecouter le top 50 :

H[souris] M P[souris] K[clic] M P[souris] K[clic] M P[souris] K[clic]

$$\begin{aligned}\text{On obtient } t_E &= t_H + 3t_M + 3t_P + 3t_K \\ &= 0,4 + 4,05 + 3,3 + 0,6 \\ &= 8,35 \text{ secondes}\end{aligned}$$

Mettre une musique en file d'attente :

H[souris] M P[souris] K[clic] M P[souris] K[clic] M P[souris] K[clic]

$$\begin{aligned}\text{On obtient } t_F &= t_H + 3t_M + 3t_P + 3t_K \\ &= 0,4 + 4,05 + 3,3 + 0,6 \\ &= 8,35 \text{ secondes}\end{aligned}$$

Supprimer une musique de la playlist :

t_P + M P[souris] K[clic]

$$\begin{aligned}\text{On obtient } t_S &= t_P + t_M + t_P + t_K \\ &= 5,7 + 1,35 + 1,1 + 0,2 \\ &= 8,35 \text{ secondes}\end{aligned}$$

2.1.2. Poopify

Tout d'abord, nous précisons que nos études se place dans la même situation que pour Spotify pour avoir des résultats significatifs.

Rechercher une musique :

H[souris] M P[souris] K[clic] H[clavier] M K[clavier]*(n+1)

$$\begin{aligned}\text{On obtient } t_R &= 2t_H + 2t_M + t_P + t_K*(n+1) \\ &= 2*0,4 + 2*1,35 + 1,1 + 0,2*(n+2) \\ &= 5 + 0,2*n \text{ secondes}\end{aligned}$$

Lancer une musique depuis sa playlist :

H[souris] M P[souris] K[clic] M P[souris] K[clic]

$$\begin{aligned}\text{On obtient } t_P &= t_H + 2t_M + 2t_P + 2t_K \\ &= 0,4 + 2,7 + 2,2 + 0,4 \\ &= 5,7 \text{ secondes}\end{aligned}$$

Ajouter une musique rechercher :

$t_R + H[\text{souris}] M P[\text{souris}] K[\text{clic}]$

$$\begin{aligned}\text{On obtient } t_A &= t_R + t_H + t_M + t_P + t_K \\ &= 5 + 0,2*n + 0,4 + 1,35 + 1,1 + 0,2 \\ &= 8,05 + 0,2*n \text{ secondes}\end{aligned}$$

Modifier les paramètres du compte :

$H[\text{souris}] M P[\text{souris}] K[\text{clic}] M P[\text{souris}] K[\text{clic}] M P[\text{souris}] K[\text{clic}] H[\text{clavier}] M K[\text{clavier}] * n H[\text{souris}] M P[\text{souris}] K[\text{clic}]$

$$\begin{aligned}\text{On obtient } t_M &= 3t_H + 5t_M + 4t_P + t_K*(n+4) \\ &= 1,2 + 6,75 + 4,4 + 0,8 + 0,2*n \\ &= 13,15 + 0,2*n \text{ secondes}\end{aligned}$$

Ecouter le top 50 :

$H[\text{souris}] M P[\text{souris}] K[\text{clic}] M P[\text{souris}] K[\text{clic}]$

$$\begin{aligned}\text{On obtient } t_E &= t_H + 2t_M + 2t_P + 2t_K \\ &= 0,4 + 2,7 + 2,2 + 0,4 \\ &= 5,7 \text{ secondes}\end{aligned}$$

Mettre une musique en file d'attente :

$H[\text{souris}] M P[\text{souris}] K[\text{clic}] M P[\text{souris}] K[\text{clic}]$

$$\begin{aligned}\text{On obtient } t_F &= t_H + 2t_M + 2t_P + 2t_K \\ &= 0,4 + 2,7 + 2,2 + 0,4 \\ &= 5,7 \text{ secondes}\end{aligned}$$

Supprimer une musique de la playlist :

$H[\text{souris}] M P[\text{souris}] K[\text{clic}] M P[\text{souris}] K[\text{clic}]$

$$\begin{aligned}\text{On obtient } t_S &= t_H + 2t_M + 2t_P + 2t_K \\ &= 0,4 + 2,70 + 2,2 + 0,4 \\ &= 5,7 \text{ secondes}\end{aligned}$$

2.2. Structuring display

2.2.1. Logscreen

Le diagramme suivant décrit la structure de l'écran de connexion à l'application.

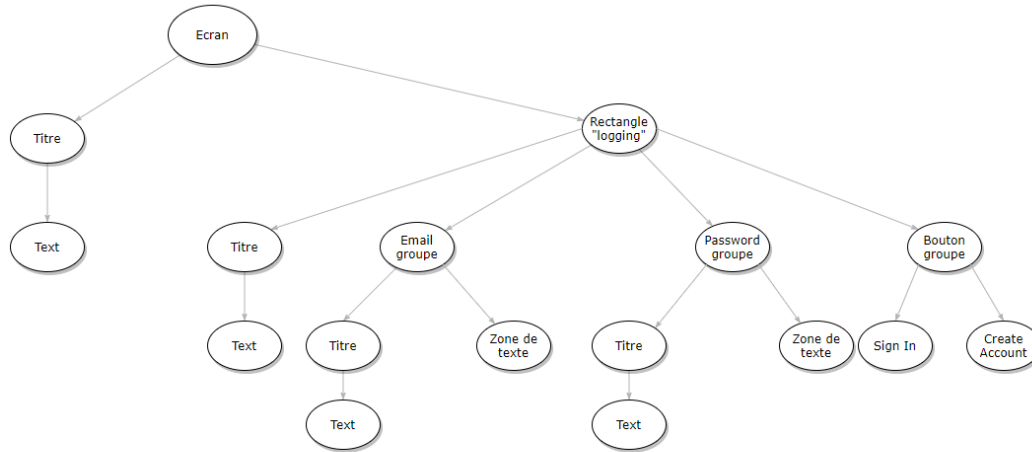


Diagramme 1 : structure de l'écran de connexion

2.2.2. Structure Générale

L'ensemble des pages constituant l'application, sont construites toutes de la même façon comme le montre le diagramme suivant :

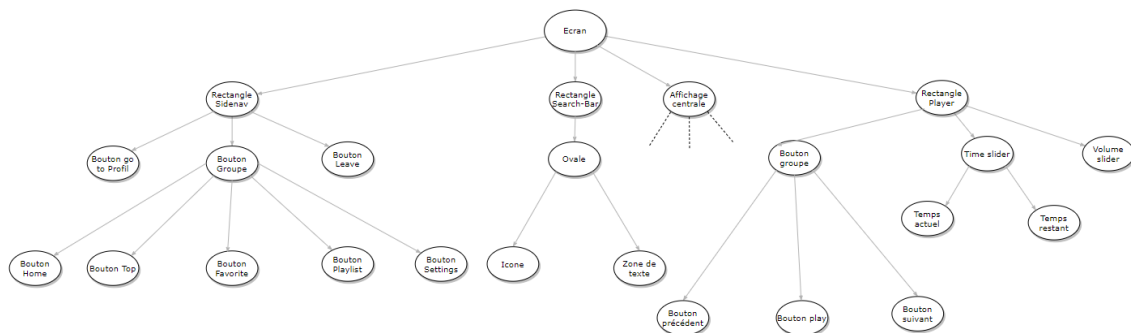


Diagramme 2 : constitution des pages de l'application

De ce fait, les diagrammes suivants ne comporteront pas explicitement cette partie par soucis de lisibilité.

2.2.3. Top 50

Cette partie correspond à l’affichage central du Top 50, toute l’interface autour est celle de la structure générale.

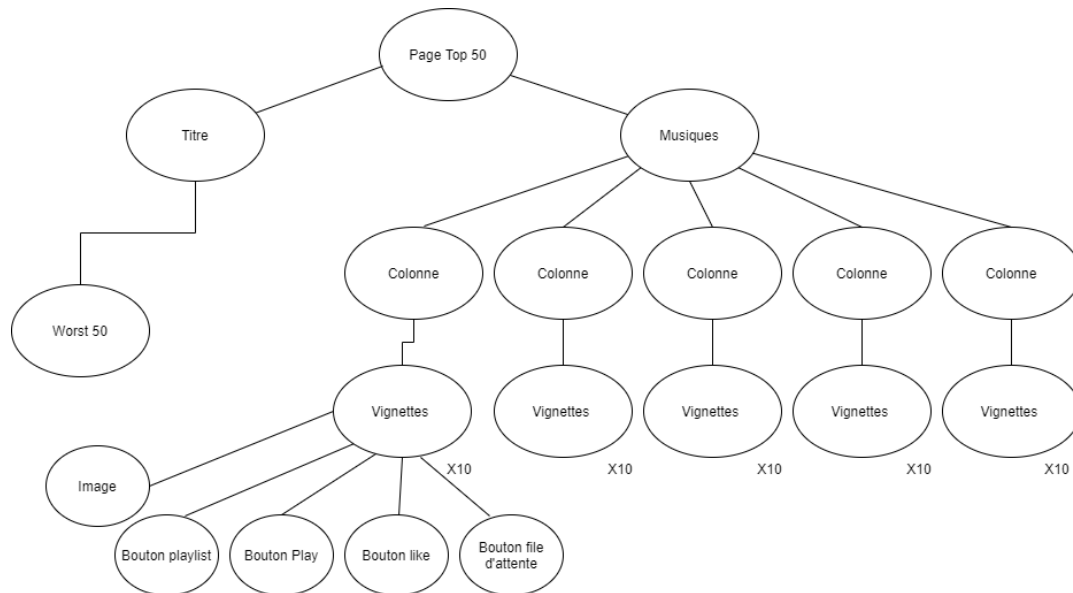


Diagramme 3 : affichage centrale du Top50

Ici, j’ai représenté uniquement une vignette car leur composition est la même pour les 50. C’est pourquoi on voit le petit « x10 » qui indique le nombre de vignettes dans chaque colonne.

2.2.4. Playlist

Cette partie correspond à l’affichage central des playlists, toute l’interface autour est celle de la structure générale.

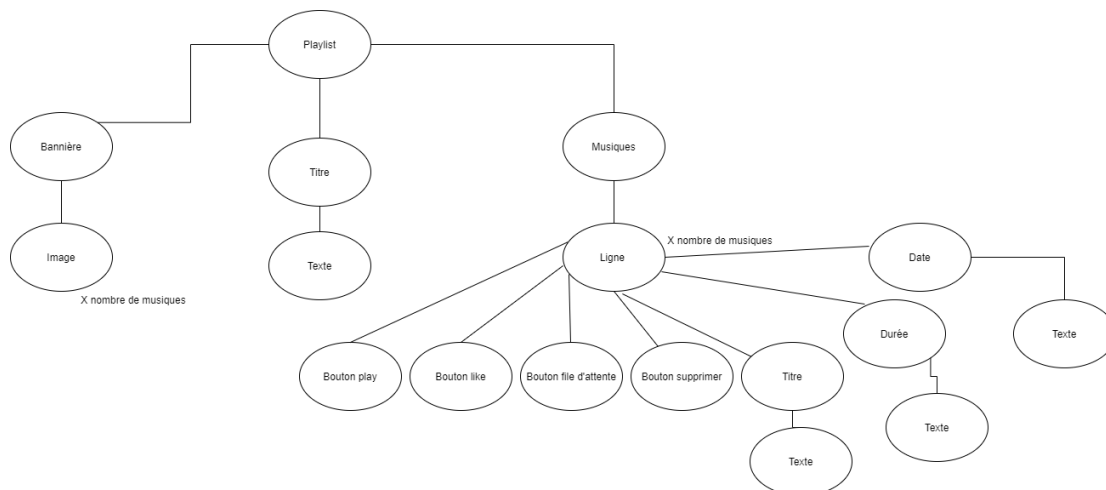


Diagramme 4 : affichage centrale des playlists

Ici, le nombre d'images dans la bannière et le nombre de lignes est variable en fonction du nombre de musique ajouté. Le « x nombre de musique » indique alors que cette élément est répété n fois.

2.2.5. Profil

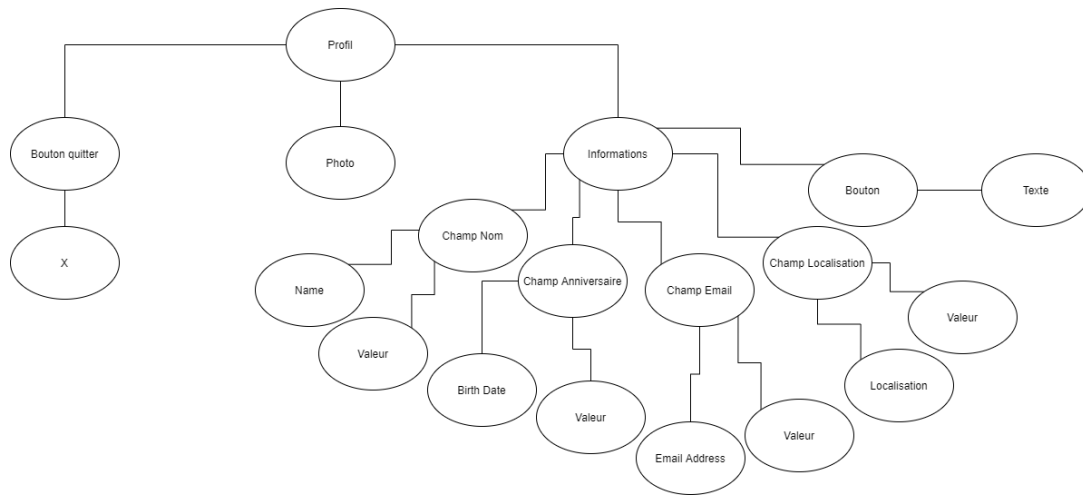


Diagramme 5 : affichage centrale du profil

2.2.6. Home

La construction de l'affichage home est une répétition du même pattern comme le montre le diagramme ci-contre :

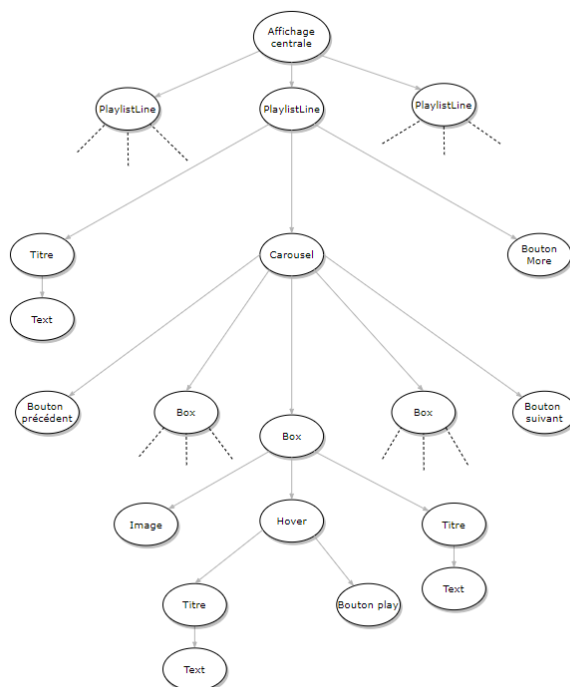


Diagramme 6 : affichage de la page home

2.2.7. Like

Pour ce qui est de la page like, elle est en tous points identique à la page playlist. Le diagramme de la playlist s'applique alors ici aussi.

2.2.8. Settings

Pour finir on a la page de settings.

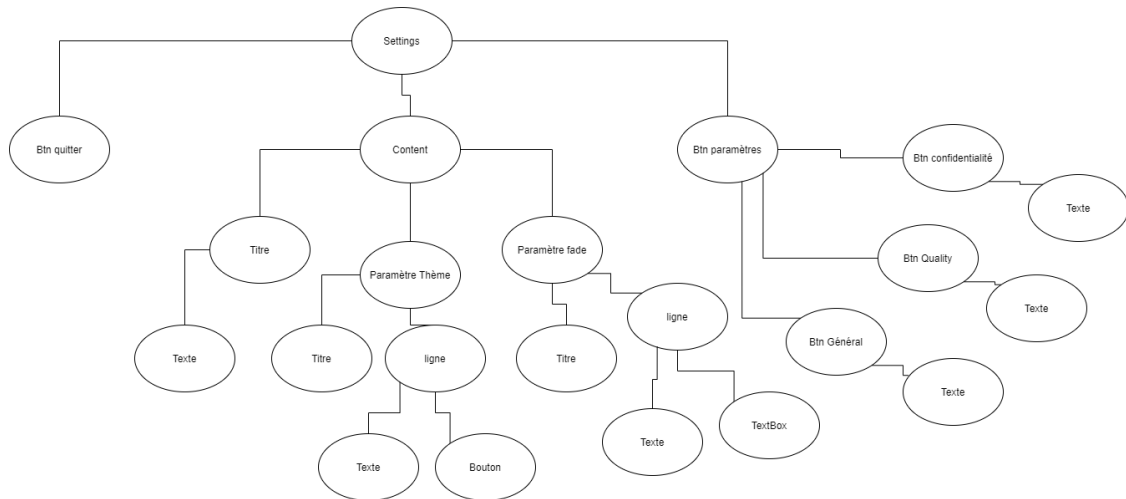
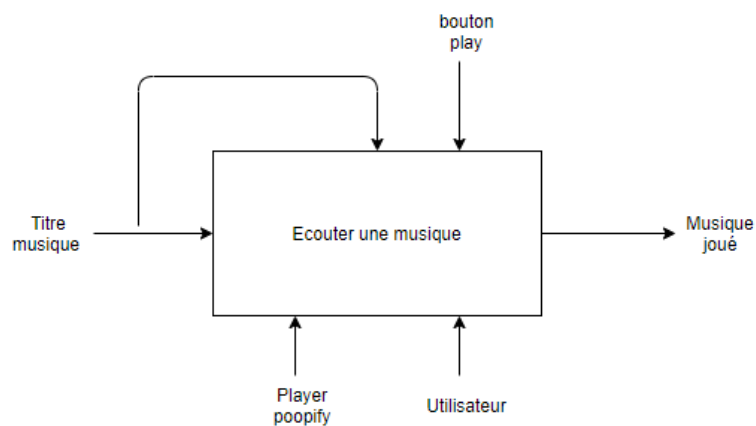


Diagramme 7 : affichage de la page de settings

2.3. Actigrammes

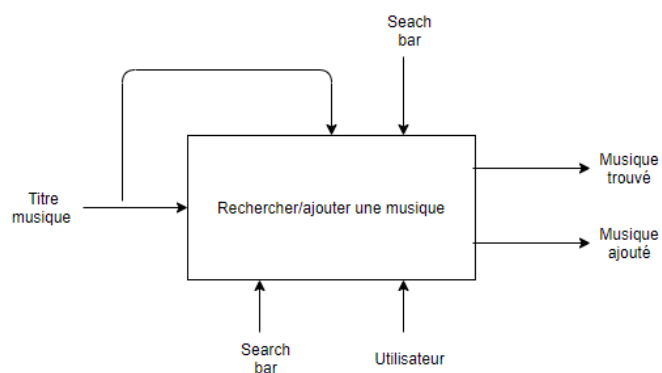
Voici une succession d'analyse de tâches des principales actions de l'application.

2.3.1. Ecouter une musique



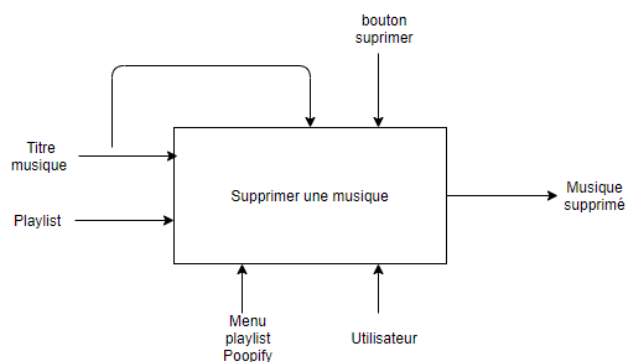
Actigramme 1 : écouter une musique

2.3.2. Rechercher/ajouter une musique



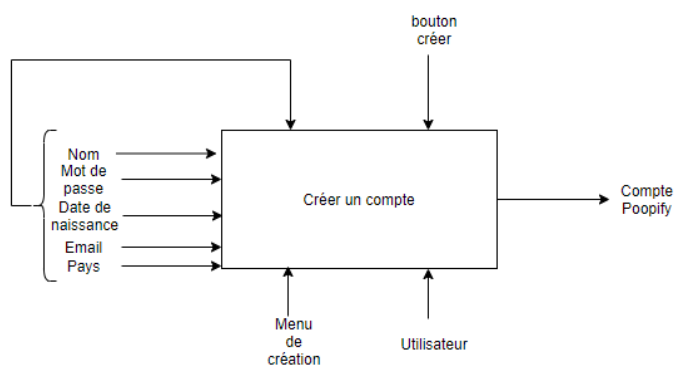
Actigramme 2 : rechercher / ajouter une musique

2.3.3. Supprimer une musique



Actigramme 3 : supprimer une musique

2.3.4. Créer un compte Poopify



Actigramme 4 : créer un compte Poopify

3. Présentation de Poopify

3.1. Qu'est-ce que Poopify

Notre projet, Poopify, comme expliqué avant est une WebApp qui permet à quiconque de créer son compte et de créer des playlists avec ses musique favorites comme Spotify. Le principe est très simple, l'utilisateur peut, grâce à l'api Youtube, rechercher des musiques et peut alors l'ajouter à sa playlist, la liker ou juste l'écouter. Toutes ces informations sont stockées dans notre base de données pour que l'utilisateur puisse par la suite écouter sa musique depuis sa playlist sans soucis. L'utilisateur a aussi la possibilité de retrouver les musiques qu'il a liker sur une page dédiée ou encore d'avoir un Top50 de musique. Tout ceci avec une interface simple et responsive pour pouvoir utiliser Poopify aussi bien sur ordinateur, tablette ou téléphone.

3.2. Les différents composants

3.2.1. Home

La page d'accueil apparaît directement après le logscreen, elle regroupe l'ensemble des éléments enregistrés sur le compte. En effet, nous pouvons retrouver les musiques écoutées récemment, l'ensemble de nos musiques favorites et également toutes nos playlists personnalisées. Pour ce faire, l'interface se doit d'être épurée avec un minimum d'information pour ne pas surcharger l'écran et ainsi ne pas déstabiliser l'utilisateur.

C'est pourquoi le contenu de chaque playlist est affiché à l'aide d'un carrousel/slider pour contrôler le nombre d'élément affiché en fonction de la largeur de la fenêtre. De plus, les musiques sont représentées à l'aide de petite « box », qui comprennent une vignette ainsi qu'une partie du titre de la musique, ce afin de garder un maximum de visibilité. Pour avoir les informations complètes de la musique il suffit de pointer la « box » pour voir apparaître la totalité des informations et même un bouton pour jouer la musique.

Il en va de même pour la playlist, un bouton « more » est présent menant à la page de la playlist, permettant d'afficher le contenu en entier de celle-ci.

3.2.2. Logscreen

Le logscreen est la première chose que voit l'utilisateur quand il se connecte sur Poopify. L'interface ce devait d'être simple et facile d'accès pour ne pas décourager l'utilisateur avant même qu'il ne se connecte. On peut donc y retrouver deux fonctions :

- Se connecter qui nécessite un email et un mot de passe. Ils vont être vérifiés par la partie backend pour voir si l'utilisateur existe.
- Créer un compte qui demande un nom, email, mot de passe, date de naissance et pays. Ces informations seront stockées dans la base de données et le mot de passe crypté pour garantir une sécurité.

Une fois connecté, l'utilisateur va alors être amené sur la page « Home » de l'application.

3.2.3. Player-bar

Le component « player-bar » correspond à la barre de lecture des musiques en bas de l'écran. Ce component est complexe puisqu'il gère 2 lecteurs de musique en parallèle. En effet, il charge la musique à jouer dans un premier lecteur et il précharge la musique suivante dans un deuxième lecteur. Cela permet de faire des transitions fondues entre les musiques. Il gère aussi la lecture de playlists mais aussi l'ajout de musique en liste d'attente. En version mobile. Le lecteur change de design pour s'adapter à tout l'écran lorsque celui-ci est actif. Sinon il se réduit en bas de l'écran avec seulement le titre de la musique et le bouton « Play/Pause » afin d'optimiser l'espace utilisé. C'est donc une partie du projet principalement back-end puisque c'est ce component qui gère toute la partie musique de Poopify.

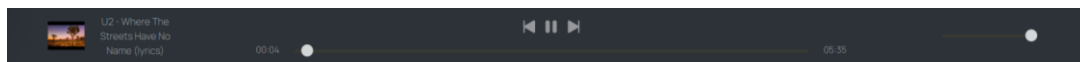


Image 1 : player version ordinateur

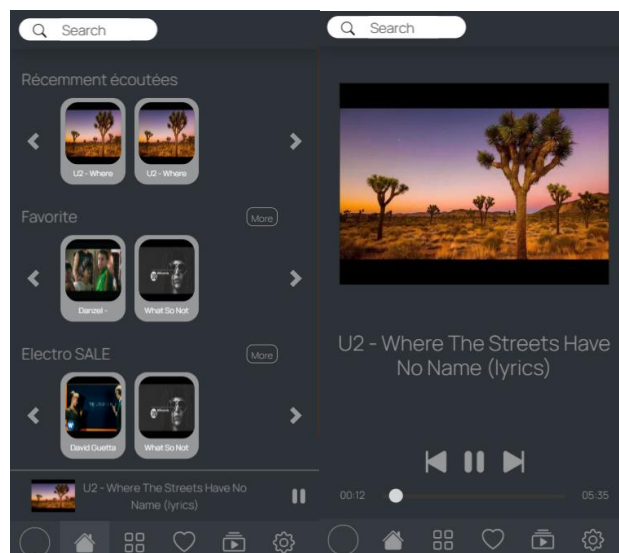


Image 2 : player version téléphone

3.2.4. Playlist

Ce component est celui qui s'occupe de l'affichage des playlists et de la gestion des musiques dans ces playlists. Ce component s'occupe alors d'afficher les musiques en fonction de l'id de playlist sélectionnée. Cela est possible grâce aux requêtes PHP de notre api. Les musiques s'affichent en ligne les unes après les autres et propose 3 options. Jouer la musique pour pouvoir l'écouter, liker la musique ou encore de supprimer la musique de la playlist. Cette dernière sera alors supprimée de la base de données.

3.2.5. Profil

Le profil est une petite fenêtre pop-up qui affiche les informations de l'utilisateur. L'utilisateur peut alors bien sûr les consulter mais aussi les modifier grâce à un bouton. Les modifications seront alors appliquées directement dans la base de données.

3.2.6. Search-bar

La search bar est un des composants les plus importants de Poopify car il permet de faire les recherches de musiques. Cette dernière fonctionne grâce à l'api Youtube. Nous faisons un lien direct entre l'input de la barre de recherche et nous l'envoyons par l'api Youtube au moteur de recherche du site qui nous affiche des résultats en rapport avec notre recherche. Ces résultats sont alors directement affichés et nous avons 3 actions possibles. Nous pouvons :

- Jouer la musique
- L'ajouter à une de nos playlists
- Liker la musique

3.2.7. Settings

Les settings utilisent une simple fenêtre pop-up qui permet à l'utilisateur de régler le thème de l'application : thème sombre ou thème clair ainsi que de pouvoir modifier la durée du fondu entre les musiques.

3.2.8. Sidenav

Ce composant s'occupe de faire la navigation entre les autres composants. Ce menu latéral possède différents onglets tous ayant un but très précis.

- Le profil pour pouvoir voir ou modifier ses informations
- Le Home pour retourner sur la page principale
- Le Top pour voir le Top50 des musiques sur Poopify
- Les Like pour voir les musiques que nous avons liker depuis le début.
- Les playlists pour pouvoir voir, ajouter ou supprimer ses playlists.
- Les paramètres pour pouvoir modifier le thème ou la durée de fondu.
- La déconnexion pour pouvoir se déconnecter de Poopify.

Tout le menu est « responsive » pour garantir une utilisation simple sur tout type de support.

3.2.9. Top

Pour ce qui est de la page qui affiche le top, dans un premier l'idée aurait été de récupérer une playlist youtube qui s'actualise mais cela aurait été trop compliqué à faire car il aurait fallu effectuer différentes recherches sur une api déjà limité. Nous avons donc opté pour une base de données que nous avons rempli avec l'aide de différentes personnes pour avoir des avis différents. Nous avons alors créé la base de données avec 50 musiques et elles sont alors simplement affichés lorsque l'utilisateur se dirige vers la rubrique correspondante. Chaque musique peut être ajouté à une playlist, liker ou juste jouée. A l'avenir on peut imaginer que cette page sera actualisée tous les X temps pour avoir quelque chose à jour.

3.2.10. Like

La page like comporte l'ensemble des musiques favorites et se comporte comme une autre playlist. Pour remplir celle-ci il suffit de cliquer sur le cœur à côté d'une musique.

3.3. Visuels

Voici quelques rendus visuels de l'application.



Image 3 : page d'accueil



Image 4 : page Top50

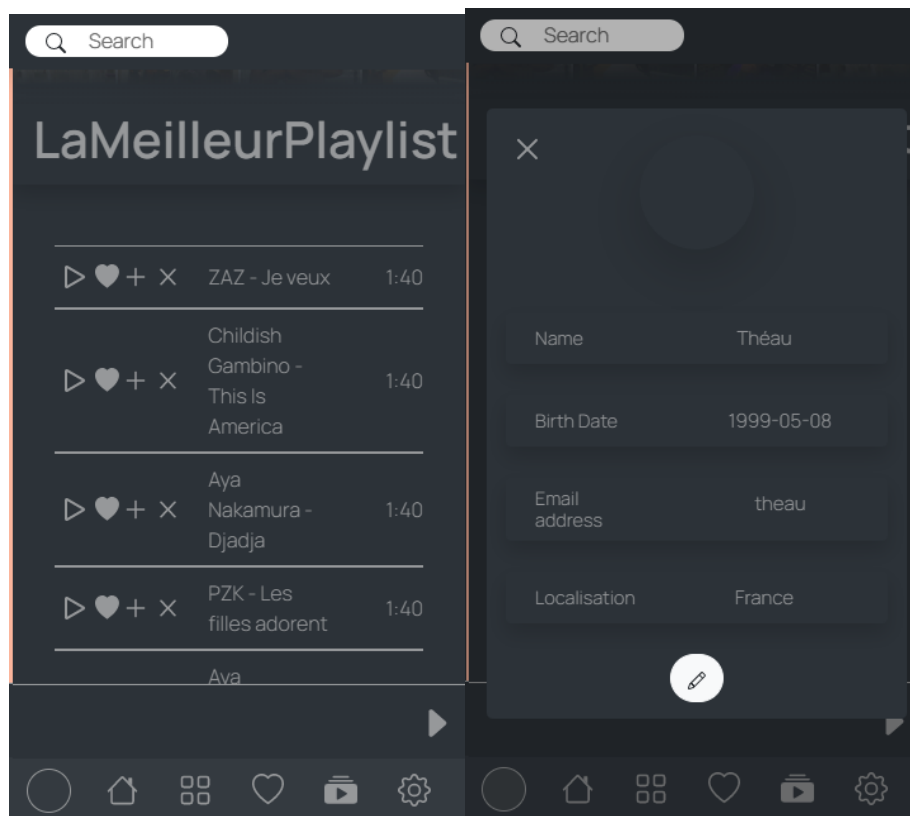


Image 5 : playlist et settings

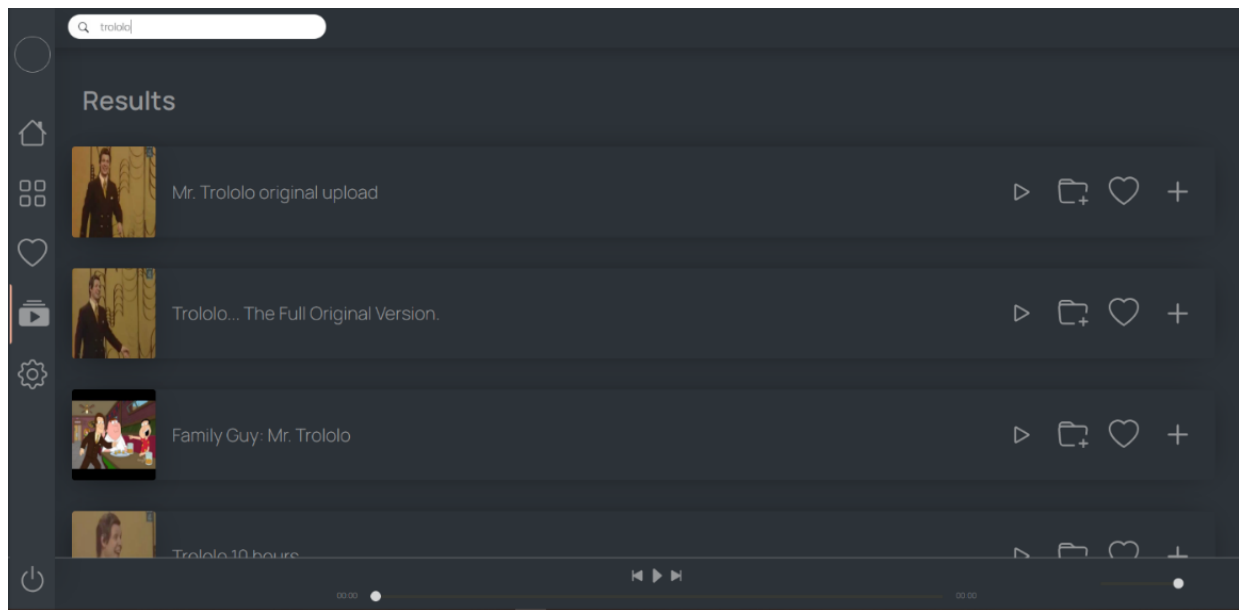


Image 6 : résultats de recherche

4. Conclusion

4.1. Problèmes rencontrés

Lors du développement de Poopify nous avons fait face à de nombreux problèmes plus ou moins compliqués. Dans un premier temps nous avons dû apprendre ce nouveau langage totalement inconnu à l'époque. Il nous a fallu comprendre comment s'imbriquent les composants et comment agencer le projet. Une fois cette première partie faite, les problèmes suivants étaient principalement la portabilité de l'application ainsi que de nombreux problèmes avec les liens back-end.

4.2. Améliorations possibles

Pour ce qui est des améliorations possibles, on peut tout d'abord imaginer un développement de plusieurs nouvelles fonctionnalités comme le partage entre compte, l'ajout d'ami ou encore des systèmes de follow. On peut aussi imaginer un système avec des top en fonctions des types de musiques ou autre pays.

4.3. Conclusion

Ainsi, ce projet nous a permis de réaliser une étude complète sur la réalisation d'une interface graphique qui se doit d'être simple, rapide, et intuitive. Pour cela nous avons choisi de faire le projet avec le framework Angular, l'occasion d'utiliser et découvrir pour une première fois pour l'ensemble du groupe ce framework. Le projet nous a permis de voir la partie réseau avec la mise en place d'un VPS avec un nom de domaine et l'utilisation d'une API en PHP pour faire le lien entre le front-end (Angular) et la base de données. Enfin, cela nous a aussi appris à mieux gérer l'organisation du travail en groupe de 3 avec la mise en place d'un repository Github basé sous Git.