

---

---

# Rapport de progression d'une application de streaming audio

Cailliau Ethann / Handouche Quentin / Lim Julie /Ablers Hugo /  
Gigot Nolan

---

# Présentation du projet

Le but du projet était de réaliser une application de streaming audio. Si son but est la lecture de sons, elle doit contenir des fonctions supplémentaires comme la lecture offline, le système d'abonnement ou un système de notation.

---

# Présentation de l'équipe

- Cailliau Ethann
- Handouche Quentin
- Lim Julie
- Ablers Hugo
- Gigot Nolan

# Product Backlog de début de projet

The Product Backlog is organized into the following columns and tasks:

- BDD**
  - Stocker musiques
  - Moteur de recherche
  - Stocker utilisateurs
  - Chiffrer les données utilisateur
  - API
  - Stocker les playlists
  - Stocker liste d'amis
  - Automatiser le rangement des données
  - + Ajouter une carte
- FRONTEND**
  - Afficher menu accueil
  - Visualisation des résultats de recherche
  - Page inscription
  - Page login
  - Affichage de la bibliothèque (playlist/artistes liés...)
  - + Ajouter une carte
- USERS**
  - Système d'upload musique
  - Gérer amis
  - Compte familial : création de profils
  - Compte familial : partage d'abonnement
  - Application native PC/Mac/Linux
  - Application mobile
  - Application console
  - + Ajouter une carte
- NAV**
  - Activer la lecture aléatoire intelligente
  - Pouvoir accéder à son compte
  - Gérer les paramètres du compte
  - Pouvoir voir l'activité de nos amis
  - Système de notation
  - Afficher les paroles de la musique que on écoute
  - Ecouter hors ligne
  - Téléchargement de musiques
  - Afficher des infos supplémentaires en cliquant sur le titre
  - Recommandation des artistes qu'on suit
  - + Ajouter une carte
- PAY SYSTEM**
  - Système de base (abonnement)
  - Incorporation de Paypal
  - Incorporation de VISA
  - Incorporation de CB
  - Incorporation de Mastercard
  - Ajout d'un système de pourboire
  - Support Apple Pay
  - Support Google Pay
  - + Ajouter une carte
- PLAYLIST**
  - Gestion playlist
  - Partage de playlists publiques via un lien
  - Playlists collaboratives modifiables par plusieurs utilisateurs
  - Réorganiser des chansons dans une playlist
  - Ajout automatique de chansons recommandées dans les playlists
  - + Ajouter une carte
- PLAYBACK SYSTEM**
  - Lecteur musique
  - Gestion volume audio
  - Gestion de file d'attente des musiques
  - Lecture aléatoire
  - + Ajouter une carte

## sprint #1



Arrêt maladie :  
- 20pts de capacité

Total de points  
dépensés :  
100/100

Ce premier sprint a été placé sous l'œil de l'infrastructure et de la formation. En choisissant des composants robustes, nous voulions réduire l'impact éventuel d'incidents futurs. Cela ne nous dérangeait pas de perdre du temps maintenant, si c'est pour en gagner plus tard.

L'arrêt maladie d'un des membres de notre équipe ne nous a pas grandement impactés. Nous avons à peine commencé l'application.

## sprint #2

Total de points  
dépensés :  
80/80

Développement

FRONT-END -  
US7  
Affichage menu  
accueil  
-10 points

PLAYBACK  
SYSTEM - US11  
Gestion volume  
audio  
-10 points

NAVIGATION -  
US8  
Moteur de  
recherche  
-10 points

PLAYBACK  
SYSTEM - US12  
Gestion file  
d'attente  
-10 points

FRONT-END -  
US9  
Affichage résultat  
de recherches  
-10 points

FRONT-END -  
US13  
Page login  
10 points

PLAYBACK  
SYSTEM - US10  
Lecteur musique  
-10 points

FRONT-END - US  
Page inscription  
10 points

Les fêtes de fin d'année nous a un peu impacté, mais le développement de l'application a vraiment pu commencer. Notre équipe a commencé à réaliser les parties essentielles de l'application. Le but est d'au moins avoir un lecteur audio avec un système d'utilisateur.

## sprint #3

Total de points  
dépensés :  
100/100

Développement

USERS - U15  
Upload musique  
-10 points

BDD - US16  
Chiffrement  
-10 points

BACK - US17  
Création API  
-10 points

BDD - US18  
Stocker users  
-10 points

BDD - US19  
Gestion Utilisateur  
-10 points

BDD - US20  
Gestion Artistes  
-10 points

Maintenance

Formation de  
l'équipe à  
PostgreSQL  
-10 points

Installation de  
PostgreSQL V1.1  
-30 points

Le contrat passé avec un réseau social a ralenti le développement de notre application. En effet, nous avons dû changer l'infrastructure de notre base de données, ce qui nous a obligés à tout réinstaller et à refaire une formation.

Nous avons dû aussi inclure une API pour créer une connexion avec le réseau social.

Nous peaufinons aussi les derniers détails avant de lancer une release lors du prochain sprint.

## sprint #4

### Developpement

BDD - US21  
Stockage playlists  
-10 points

FRONT - US22  
Gestion playlists  
-10 points

FRONT - US23  
Affichage  
bibliothèque  
-10 points

BDD - US24  
Stocker liste  
d'amis  
-10 points

USER - US25  
Ajouter un  
utilisateur dans la  
liste d'amies  
-10 points

FRONT - US26  
Supprimer un  
utilisateur de la  
liste d'amie  
-10 points

PLAYBACK  
SYSTEM - US27  
Lecture aléatoire  
-10 points

NAV - US28  
Lecture aléatoire  
intelligente  
-10 points

Front-End - US29  
Afficher liste  
d'amie  
-10 points

Total de points  
dépensés :  
/100

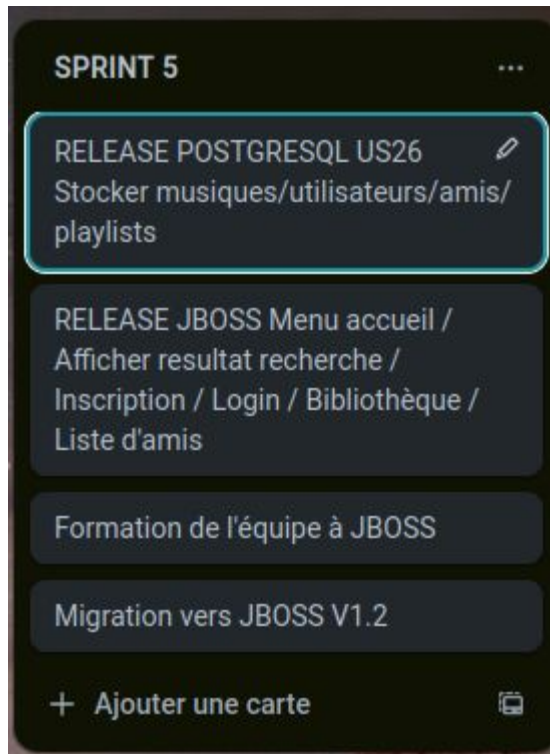
### Contraintes

Réparation de  
l'incident  
-10 points

Ce qui nous a retardé n'est pas l'incident, mais plutôt la panne sur AWS qui nous a empêché de réaliser notre première release.

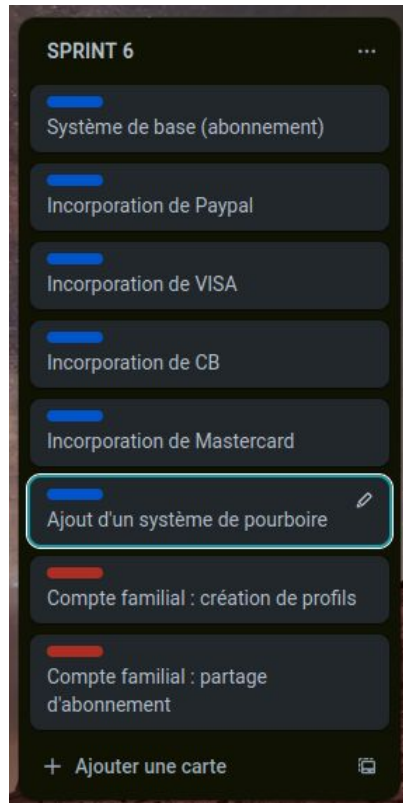
Nous nous sommes quand même servi de ce temps perdu pour intégrer les playlists, la bibliothèque ou encore la dimension sociale de l'application.



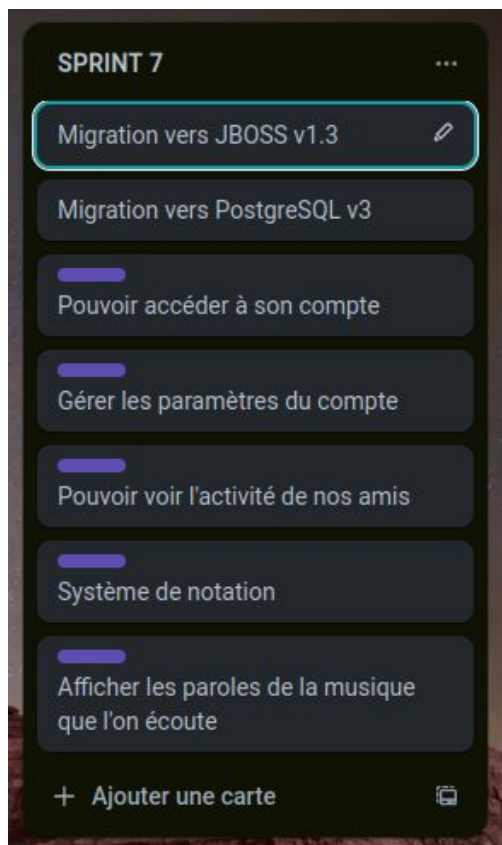


L'incompatibilité avec PostgreSQL et WebSphere nous a poussés à devoir changer pour JBoss en catastrophe. Nous avons dû aussi nous former à cette technologie, nous ralentissant de plus en plus.

Nous avons quand même utilisé le temps restant pour réaliser la release et mettre à jour Postgres.



L'incident ne nous a que peu impactés. Nous continuons le développement de l'application dont le sprint est porté sous l'oeil du système de paiement et du compte familial.



Rien à signaler. Nous procédons aux mise à jour et à l'ajout de fonctionnalités pour une release au prochain sprint.

## sprint #8



L'arrêt maladie de notre collaborateur ainsi que l'évolution du cout du cloud public ont grandement ralenti le développement.

Alors que nous étions partis finaliser pour faire une nouvelle release au prochain sprint, nous avons dû reporter ça au suivant.

## sprint #9

Developpement

PAY SYSTEM -  
US48

Support Google  
Pay

-10 points

Maintenance

Mise à niveau vers  
PostgreSQL v3.1

-10 points

Contrainte

Correction urgence  
PSQL

-30 points

Total de points  
dépensés :  
70/70

Là encore, le développement fut grandement ralenti par la période estivale et la correction d'urgence de Postgres.

C'est un sprint de perdu. Nous rattrapons notre retard et procédons à la mise à jour.



Pour le sprint final, nous avons finalisé les fonctionnalités que nous avons jugées obligatoires avant la fin. Tout n'a pas été inclus, mais les fonctionnalités les plus importantes sont quand même intégrées.

---

# Conclusion

Malgré les précautions prises dès le premier sprint pour réduire les risques, différents événements indépendants de notre volonté ont quand même ralenti le projet. Cependant, le projet contient tout de même les fonctionnalités essentielles que nous devons inclure avant son déploiement.

Nous avons su prioriser ce qui nous a paru essentiel et nous avons réagi face aux obstacles.

---