|  |  |
| --- | --- |
|  | **2017** |
|  | I.FA-P3A  Michael Ramusi |

|  |
| --- |
| **[Audio File player]** |
| Travail pratique individuel d’informatique |

Table des matières

[1 Introduction 2](#_Toc484506840)

[1.1 Motivations 2](#_Toc484506841)

[2 Cahier des charges 2](#_Toc484506842)

[2.1 Objectif 3](#_Toc484506843)

[2.2 Fonctionnalités principales 3](#_Toc484506844)

[2.3 Modèle conceptuel 3](#_Toc484506845)

[2.4 Planning initial 4](#_Toc484506846)

[3 Analyse concurrentielle 4](#_Toc484506847)

[4 Analyse fonctionnelle 5](#_Toc484506848)

[5 Analyse organique 6](#_Toc484506849)

[6 Rapport de tests 6](#_Toc484506850)

[7 Conclusion 6](#_Toc484506851)

[8 Annexes 6](#_Toc484506852)

# Introduction

Actuellement en **dernière année** de mes 3 années de formation en **informatique** au sein du Centre de Formation Professionnel Technique (CFPT), j’ai pour tâche d’effectuer un **travail de fin de diplôme**. Ce travail, nommé **TPI**, est nécessaire pour l’obtention de mon **Certificat Fédéral de Capacité** (CFC).

## Motivations

Passionné depuis toujours par la musique, j’ai décidé de réaliser, dans le cadre de mon projet de TPI, une application liée à cette passion pour l’art musical : **AudioFile Player**. Cette application est un **lecteur de fichiers audio** qui a une particularité : les fichiers sont sur un **serveur distant**.

**Pourquoi un système de cloud ?** Tout d’abord, afin de **centraliser** en un point central ses données. En effet, aujourd’hui, beaucoup jonglent entre leur ordinateur et leur smartphone pour écouter de la musique. Avoir ses fichiers sur un seul serveur distant permet ainsi de faciliter l’écoute de sa bibliothèque musicale.

Egalement, ce système de cloud est également avantageux pour les personnes ayant une audiothèque volumineuse. En effet, sur un périphérique comme le smartphone, le stockage peut vite être saturé par les applications, vidéos etc. Accéder à son audiothèque par le biais du réseau de données peut ainsi être une solution afin **d’économiser de la place** sur son smartphone.

## Organisation

|  |  |
| --- | --- |
| *Élève :*  Michael RAMUSI  [ramusim@protonmail.ch](mailto:ramusim@protonmail.ch) | *Maître d’apprentissage :*  Jasmina TRAVNJAK  [edu-travnjakj@eduge.ch](mailto:edu-travnjakj@eduge.ch) |
| *Experts :*  Eric FEHR  [eric.fehr@skyguide.ch](mailto:eric.fehr@skyguide.ch) | Daniel VANINI  [daniel.vanini@skyguide.ch](mailto:daniel.vanini@skyguide.ch) |

Ce travail de TPI sera réalisé du 6 mai au 19 mai 2017. La réalisation du projet se déroule sur 10 jours de 8h pour un total de 80 heures de travail.

# Cahier des charges

## Objectif

L’objectif de ce projet est de créer une **application web** qui permet à un utilisateur de **déposer des fichiers musicaux** afin qu’il puisse avoir accès à ceux-ci partout, à la manière d’un **cloud**.

## Fonctionnalités principales

* Fonctions d’inscription et de connexion
* Dépôt de fichiers au format .mp3
* Vérification des métadonnées (ID3) des fichiers déposés
* Lecture des fichiers de l’audiothèque de l’utilisateur
  + Play, pause, suivant, précédent
* Affichage des musiques selon plusieurs tris
  + Par artiste
  + Par album
  + Par titre
* Création de listes de lecture (playlist)
* Fonctionnalités sociales :
  + Modification de profil
  + Ajout d’amis

## Modèle conceptuel

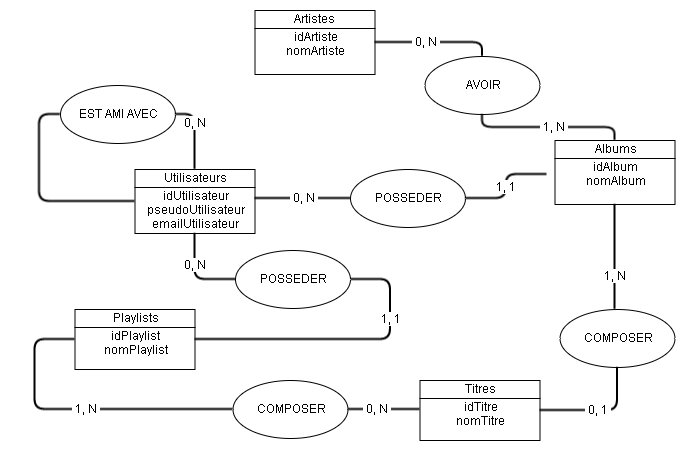


Figure : MCD AudioFile Player

Ceci est le modèle conceptuel de données défini lors de la rédaction du cahier des charges. Le travail se base sur ce modèle pour la réalisation de la base de données en MySQL.

## Planning initial

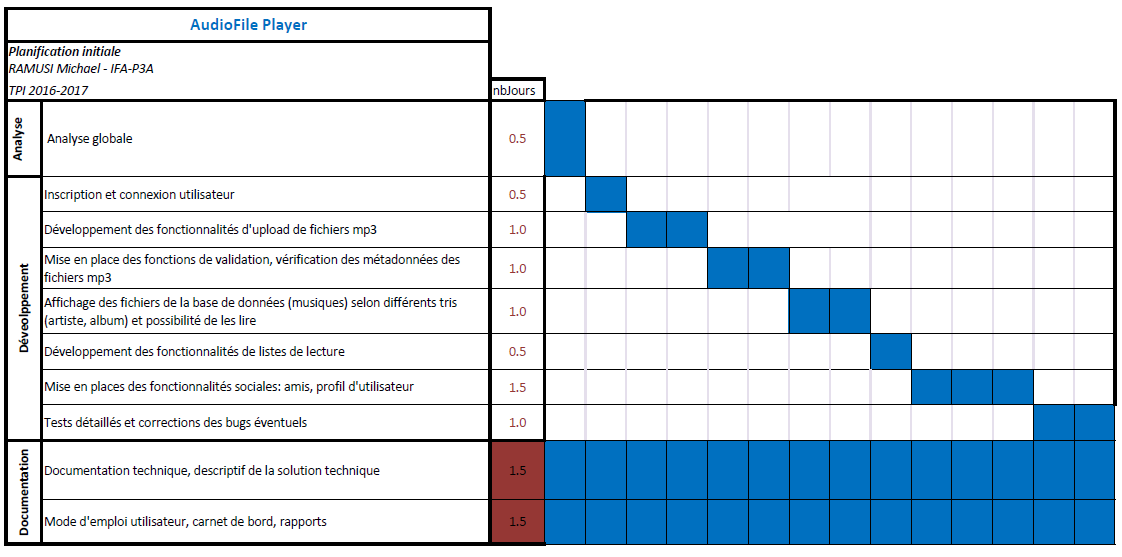


Figure : planning initial

Ceci est le déroulement des tâches nécessaires à la réalisation du projet. Comme énoncé plus haut, le travail est réalisé en 80 heures / 10 jours.

## Technologies & logiciels utilisées

Pour la réalisation de cette application j’ai décidé d’utiliser des technologies **web** par **préférence personnelle** d’abord mais également pour faciliter un éventuel passage de l’application sur une **plateforme mobile**:

* PHP
* JavaScript (jQuery)
* HTML/CSS
* MySQL

Les logiciels utilisés :

* IDE : *NetBeans IDE*
* Serveur local : *EasyPHP*
* Navigateur : *Google Chrome & Mozilla Firefox*
* Gestion de version : *GIT*
* Maquette : *Balsamiq*

# Analyse concurrentielle

## iMusic

L’application **iMusic** est un outil puissant pour gérer sa collection de musiques. En plus de pouvoir déposer ses propres titres, l’application permet d’écouter des musiques à partir d’autres logiciels comme Spotify ou YouTube. iMusic propose également sa propre librairie musicale, afin de pouvoir écouter de la musique populaire sans avoir à la télécharger. Egalement, l’application permet une **synchronisation** **locale** entre différents périphériques. Cette synchronisation locale diffère donc de la lecture en streaming de mon application.

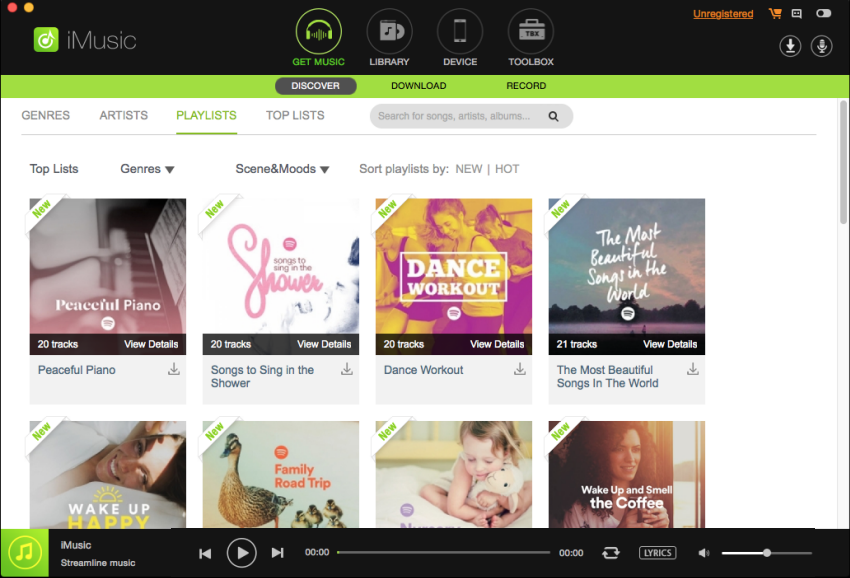


Figure : iMusic

## Google Play Music

L’application musicale signée Google. C’est une application robuste qui permet de lire ses fichiers musicaux mais également d’aller les chercher sur le magasin de titres de chez Google. La force de l’application est de pouvoir accéder à une bibliothèque **depuis plusieurs périphériques** grâce aux serveurs de Google qui offre la possibilité d’avoir **20'000 titres disponibles en streaming.** L’application ne gère cependant pas la validation des métadonnées du fichier.

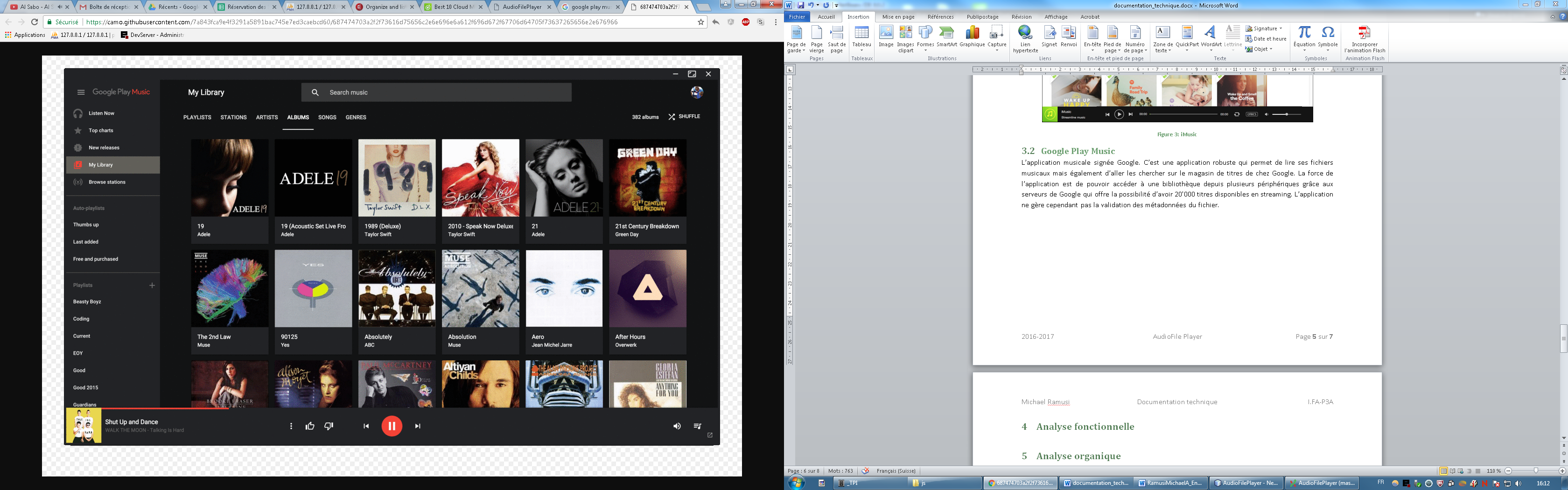


Figure : Google Play Music

# Analyse fonctionnelle

# Analyse organique

# Rapport de tests

# Conclusion

# Annexes

# Sources

Le fond d’écran utilisé est libre de droit