

SOUNDROID

Rapport de développement



Table des matières:

Table des matières:	2
Introduction	3
Fonctions par Niveau:	4
I - Mise en place du projet	4
Méthode de travail:	4
Conception du projet:	5
II - Choix de développement	6
Lecteur musical:	6
Choix de Classes:	7
Bibliothèque et base de données:	7
Conclusion	8

Introduction

Dans le cadre du cursus d'ingénieur informatique à l'ESIPE (Ecole supérieure d'ingénieurs de Paris-Est), les étudiants de la filière suivent en deuxième année un cours d'interface graphique. Cette matière enseignée par M.Michel CHILOWICZ, chercheur au Laboratoire d'Informatique Gaspard-Monge de Marne-la-Vallée présente aux étudiants les bases du développement d'applications sous android.

L'apprentissage de la matière se fait par des cours dont les notions sont ensuite appréhendées par les étudiants lors de travaux pratiques. Ces TP ont permis aux élèves d'améliorer leurs qualités de développement avec des sujets à la difficulté croissante. Le but de cet enchaînement, était d'amener chaque apprenti ingénieur à réaliser un projet de groupe sur plusieurs semaines dans la matière. Le sujet de projet proposé par M.CHILOWICZ consistait à réaliser un lecteur musical fonctionnant sous android, agrémenté de différentes fonctionnalités. Le présent rapport a pour but d'explicitier les choix fait lors du développement fait par messieurs Julien BESSODES, Guillaume MATHECOWITSCH et Nils PERNET. Il décrira également les raisons de ces choix et les différents problèmes rencontré par le trinôme lors de la création de cette application.

Le barème de ce projet et la liste des fonctionnalités ont été découpés en 3 niveaux. L'application est considérée comme terminée si elle propose les fonctionnalités du niveau 1, les niveaux 2 et 3 étant alors des fonctionnalités bonus.

Fonctions par Niveau:

Les informations suivantes ne présentent que les fonctionnalités disponibles dans le projet du groupe BESSODES-MATHECOWITSCH-PERNET (groupe 4).

Niveau 1:

- Lecteur musical
- Recherche de pistes
- Listes de lecture
- Base de données des pistes musicales
- Export de la base
- Partage musical vers un URL

Niveau 2:

- Historique de lecture
- Import d'une base
- Pause automatique

Niveau 3:

- Base de donnée des pistes musicales ORM
- Listes de lecture enfant d'autres listes.

I - Mise en place du projet

Méthode de travail:

Le choix de ce trinôme était évident pour chaque membre. Ayant l'habitude de travailler ensemble et s'appréciant les uns les autres nous avons de suite décidé de collaborer sur ce projet. Cependant, en période de pandémie mondiale, le travail en groupe s'est montré compliqué, notamment à cause des règles de confinement. Nous avons donc décidé de créer un serveur de travail consacré à ce projet sur le logiciel de

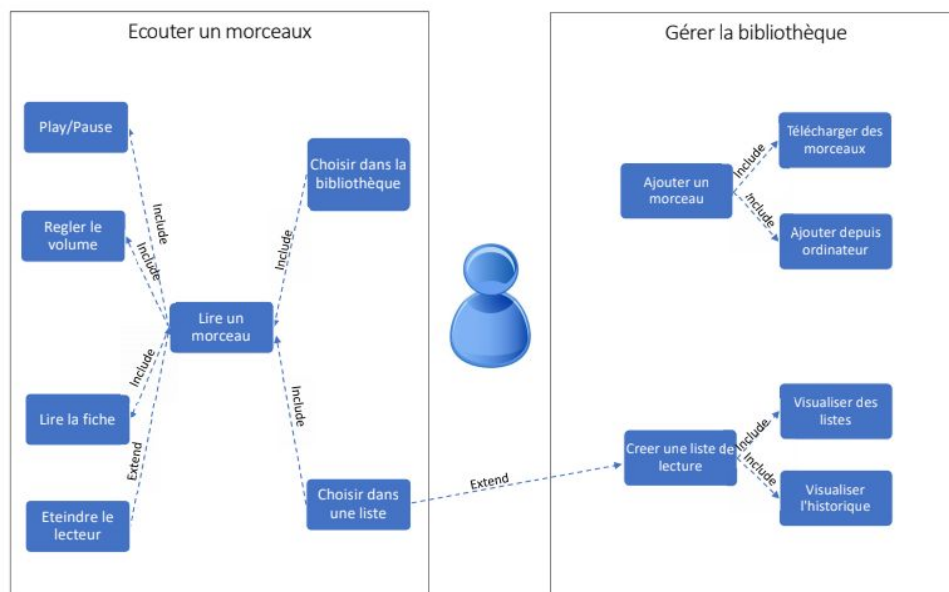
Voice over IP, Discord. Tout au long du projet, nous avons pu discuter, échanger, et imaginer l'application grâce à ce serveur.

Le développement a eu lieu sur l'IDE Android studio, que nous avons appris à utiliser lors des TP, et le versionning sur la plateforme Git, dans un répertoire dédié. Cependant, des problèmes récurrents avec Android studio, et ce, malgré de nombreuses réinstallations, ont forcé Guillaume MATHECOWITSCH et Nils PERNET à travailler sur la machine de ce dernier en utilisant TeamViewer pour utiliser android studio et git. De même, les essais de l'application sur machine n'ont pu être réalisés qu'avec des machines virtuelles, l'équipe n'ayant pas de machine physique à disposition et le confinement empêchant de s'en procurer.

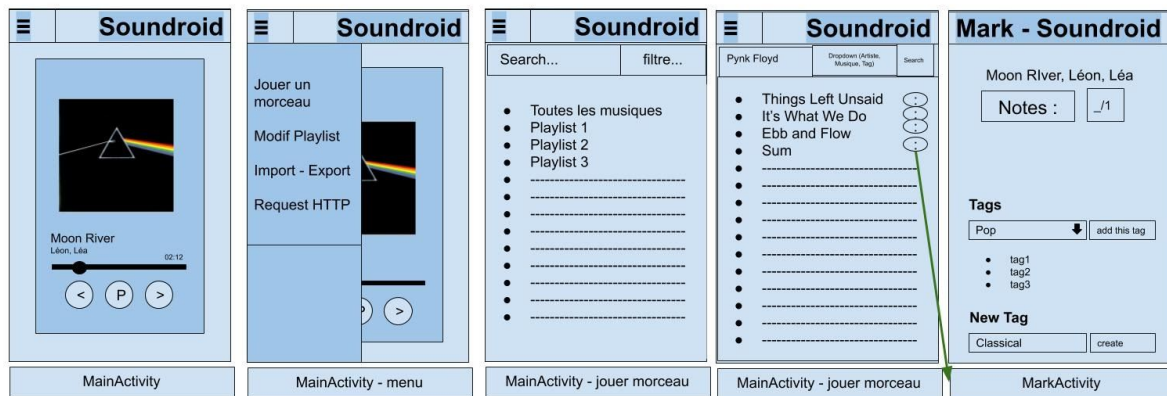
Ces différentes parades ont permis au projet de se concrétiser malgré tout et d'aboutir à un lecteur musical opérationnel proposant de nombreuses fonctionnalités.

Conception du projet:

Après avoir pris connaissance du sujet, et avoir découvert les différents niveaux de fonctionnalités attendues, nous avons pu brainstormer sur la conception du projet sur notre serveur discord. Afin de savoir quoi et comment développer, nous avons commencé par lister les différentes fonctionnalités à implémenter et réfléchir aux cas d'utilisation. En y réfléchissant, nous sommes parvenus à réaliser une ébauche de diagramme nous aidant à comprendre le fonctionnement de l'application de façon synthétique.



Dans un second temps, nous avons commencé à réfléchir aux différentes vues que nous souhaitions implémenter. Nous avons imaginé plusieurs vues, une activité principale correspondant au lecteur en lui même, une vue pour la recherche, une vue pour les playlists et une pour les fiches d'informations (non implémentée). A ces vues s'ajoute évidemment un menu assez classique sur la gauche de l'écran, comme *sur le modèle de ceux que nous en avons utilisé en TP*. Les vues précédemment imaginées ont par la suite permis de réaliser un wireframe, permettant par la même occasion d'imaginer le travail à réaliser et débiter le développement.



II - Choix de développement

Lecteur musical:

Lors du développement, nous avons d'abord voulu réussir à créer un lecteur de musique fonctionnel, sans trop de fonctionnalités. Pour réaliser cela, nous avons fait le premier choix fort de ne pas créer de service pour le lecteur musical. En effet, en ayant connaissance des nombreuses fonctionnalités à implémenter, nous n'avons pas trouvé de plus value à implémenter un service. N'ayant pas de connaissances dans le domaine et n'ayant que peu de temps, il nous a semblé plus judicieux de créer le lecteur le plus simplement possible puis de nous concentrer sur les autres fonctionnalités. Concernant l'activité, nous avons choisi de travailler avec une navigation drawer activity, toujours dans l'optique de gagner du temps, mais aussi pour améliorer le confort de l'utilisateur. Concernant les différentes interfaces portant sur les listes de titres et de playlists, nous avons utilisé des recyclerview avec viewholder. De

plus, nous avons décidé d'utiliser un maximum de boîtes de dialogue afin d'augmenter l'interactivité avec l'utilisateur. Ces différents choix nous ont permis de produire un player efficace et proposant de nombreuses options dont nous pouvons être satisfaits. Cependant, nous regrettons de ne pas avoir rendu possible l'affichage d'une image en fonction du titre à l'écoute. Cette option était envisagée, mais par manque de temps, nous avons préféré privilégier l'implémentation de fonctionnalités plus conséquentes.

Choix de Classes:

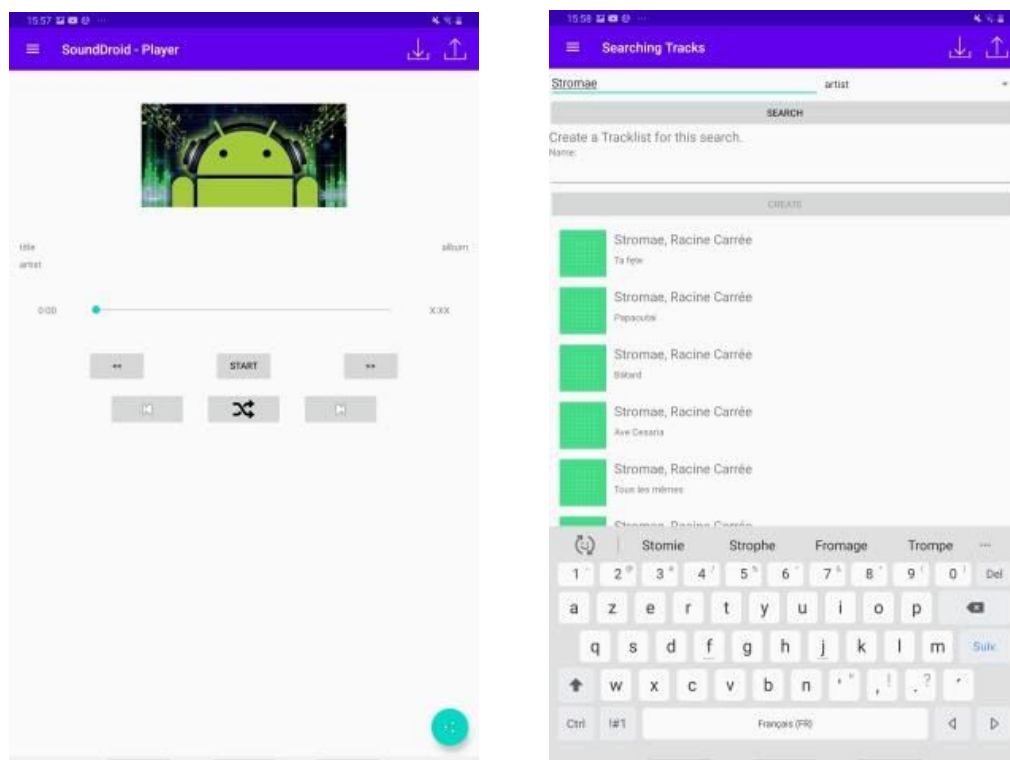
Concernant les différentes classes créées pour le projet, nous avons fait plusieurs choix qui nous ont semblés judicieux. Nous avons décidé de découper les types de contenu. On aura donc par exemple une classe pour les pistes, la classe Track. Chaque classe de type de contenu est accompagnée par une classe Singleton "manager". Ainsi, la classe Track est accompagnée d'une classe TrackManager permettant de manipuler l'objet Track. De plus, les objets track et tracklists ont tout deux un identifiant de type HASH. Nous avons décidé d'utiliser ce type car il nous permet de manipuler les musiques et les listes de lectures comme des tracklistables. Cela permet notamment d'inclure des listes de lectures dans d'autres listes. Sachant que nous devions mettre en place une base de donnée pour représenter la bibliothèque, nous avons fait en sorte que chaque classe représente une table dans cette dernière.

Bibliothèque et base de donnée:

Bien que le niveau 1 de fonctionnalité demandait d'utiliser l'API des collections pour représenter notre base de donnée, nous avons décidé de gagner du temps en implémentant directement la base de donnée de niveau 3, avec une bibliothèque ORM. Comme annoncé précédemment, chaque classe implémentée représente une table de la base de donnée. Cela correspond aux DAO, objets d'accès aux données de notre bibliothèque. Nous avons décidé de ne pas utiliser de librairie pour représenter la base de données afin de coder plus bas niveau. Nous avons fait ce choix car coder bas niveau nous permet de manipuler les données présentes dans la bibliothèque plus facilement. De plus, une fonctionnalité de niveau 1 et une de niveau 2 concernaient respectivement l'export et l'import de la base de donnée. Pour réaliser ces fonctions, nous avons mis en place une classe PortableDatabase permettant de représenter la base de donnée. Cette classe utilise le format JSON pour importer et exporter une bibliothèque.

Conclusion

Finalement, nous avons, malgré avoir rencontré de nombreux problèmes liés au covid19 et à la mise en place de nos espaces de travail, réussi à réaliser un travail conséquent et à mener à bien notre projet. Nous avons été en mesure de réaliser une application de lecture de musique fonctionnelle et plus. Celle-ci s'avère en effet également particulièrement ergonomique et propose de nombreuses fonctionnalités.



Ce projet a donc renforcé nos capacités de développement d'applications android, mais il nous a également appris à concevoir un projet complet selon un cahier des charges, et à travailler en équipe sur un développement. Nous gardons donc énormément de choses de ce travail, la fierté d'avoir été en capacité de le réaliser, mais surtout une expertise et une expérience enrichissante proche de celles que nous rencontrerons dans le monde du travail.