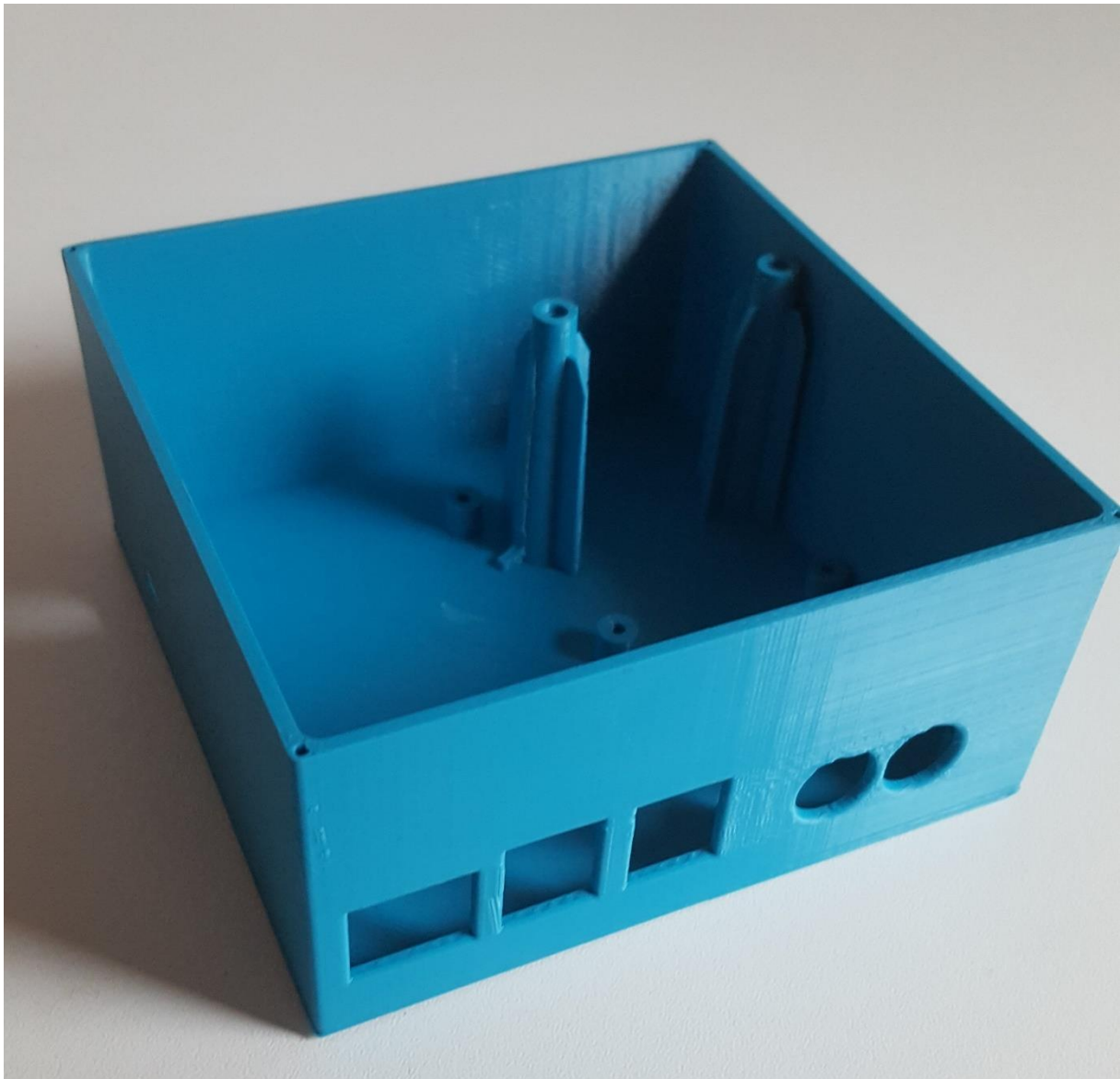


Dubourdieu

Aurélien

Synthèse personnelle

Jukebox Connecté



PRÉSENTATION DU BESOIN

1) Mise en situation

Généralement, dans une soirée entre amis, une personne s'occupe de la musique, mais celle-ci n'est pas forcément au goût de chacun. Le jukebox connecté est un objet de divertissement qui lit la musique à la manière d'un jukebox « nextgen » permettant ainsi de satisfaire un groupe d'amateur de musique.

2) Le besoin et son marché

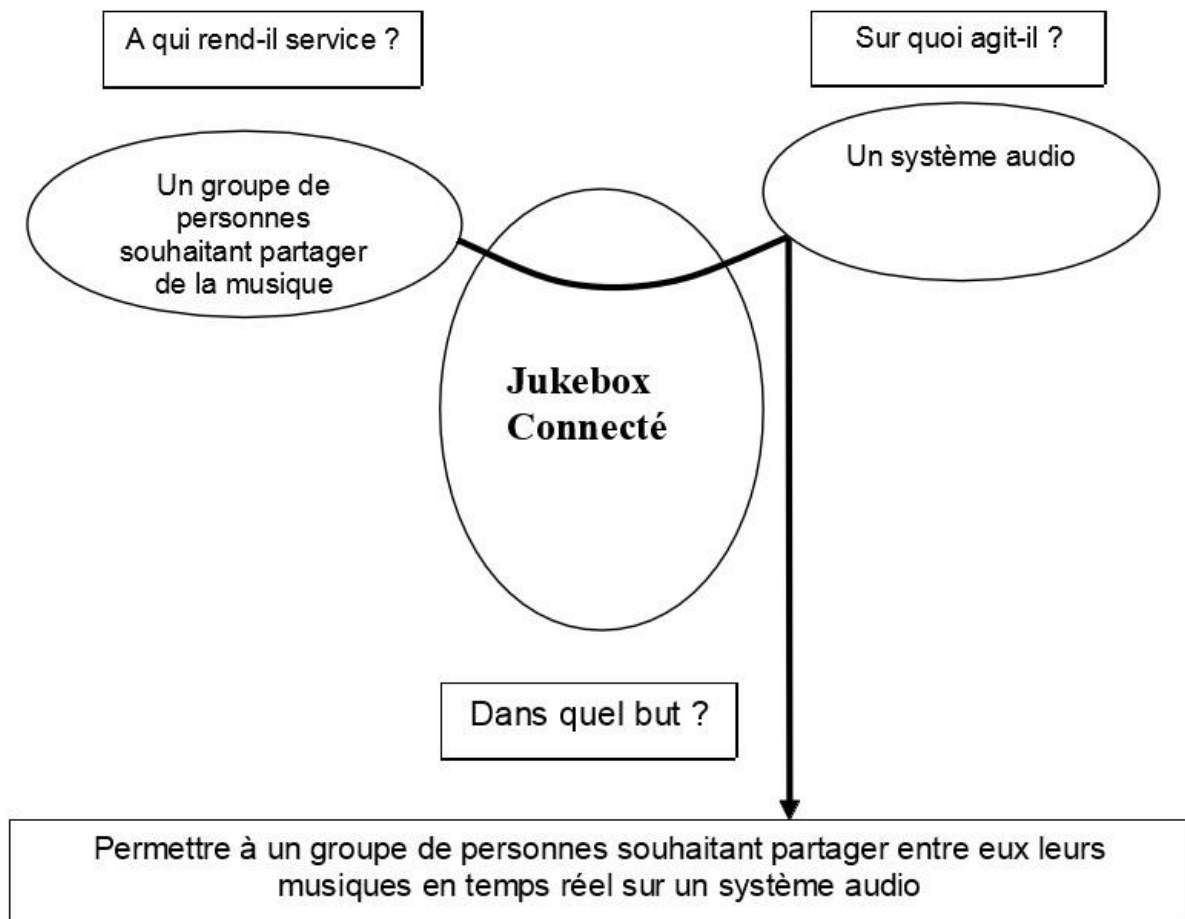
Permettre à un groupe de personnes souhaitant partager entre eux leurs musiques en temps réel sur un système audio. L'accélération de la mutation des usages liés à la consommation de musique a marqué l'année 2015: avec désormais 3 millions d'abonnés, soit 5% de la population française, le streaming franchit un cap majeur et s'affirme comme le moteur de croissance de la musique enregistrée.

3) Le contexte du projet et ses objectifs

Pour remédier à cela, nous nous proposons de développer un système de partage de musiques entre les différentes personnes invitées à la soirée.

Expression fonctionnelle du besoin

4) Énoncé du besoin



5) Validation du besoin

Pourquoi ce besoin existe-il ?

- Pour satisfaire un groupe d'amateur de musiques
- Pour permettre à n'importe qui d'ajouter de la musique en soirée
- Parce que la musique ne peut convenir à tout le monde
- Pour adapter la musique aux goûts de chacun

Qu'est ce qui pourrait le faire disparaître ?

- Plus de soirées entre amis (improbable)
- Pas d'accès internet (improbable)

Qu'est ce qui pourrait le faire évoluer ?

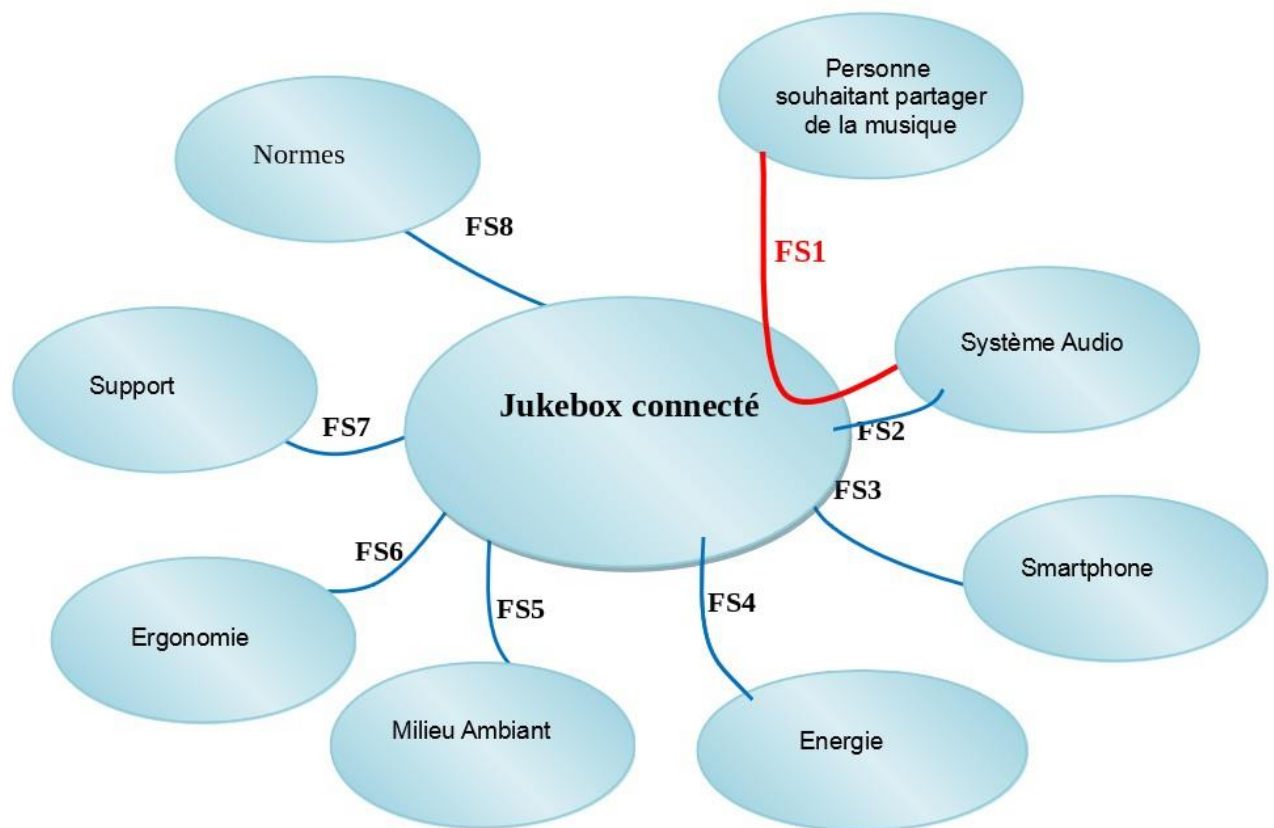
- Evolution des modes de communications

Conclusion

En conclusion, le besoin est validé à court terme.

Analyse fonctionnelle du besoin

6) Relations avec le milieu extérieur



7) Listes des fonctions de service

FS1 : Permettre à toute personne de partager sa musique sur un système audio.

FS2 : Être connecté au système audio

FS3 : Être contrôlé à distance depuis un smartphone.

FS4 : Être alimenté en énergie électrique

FS5 : Résister aux agressions du milieu ambiant.

FS6 : Être simple d'utilisation.

FS7 : Être stable sur son support.

FS8 : Respecter les normes en vigueur

8) Caractéristiques des fonctions

FS1	Permettre à toute personne de partager sa musique sur un système audio.	<ul style="list-style-type: none"> - Mise en marche - Sélection - Vote - Choix - Temps de requête 	<u>Programme adapté</u> <u>Algorithme</u> 1s	/ / Maximum	Dès la mise sous tension. Gestion de la priorité de lecture avec un algorithme favorisant la répartition des musiques jouées par les utilisateurs.
FS2	Être connecté au système audio.	<ul style="list-style-type: none"> - Type de connexion 	RCA Bluetooth	/ /	Les enceintes pourront être connectées, au choix, en RCA, jack 3.5 ou en Bluetooth.
FS3	Être contrôlé à distance depuis un smartphone.	<ul style="list-style-type: none"> - Dialogue avec l'interface - Mode de connexion à l'interface - Connectivité 	Via la page Web <u>QrCode</u> ; <u>NFC</u> ; <u>ip</u> <u>Wifi (portée: 8 mètre)</u> ; Ethernet	/ / Minimum	Connexion aisée à l'aide du <u>NFC</u> et du <u>QrCode</u>
FS4	Être alimenté en énergie électrique.	<ul style="list-style-type: none"> - Alimentation 	230V 50 Hz	/	/
FS5	Résister aux agressions du milieu ambiant.	<ul style="list-style-type: none"> - Aux chocs 	Chute de 80cm	± 20cm	/
FS6	Être simple d'utilisation.	<ul style="list-style-type: none"> - Interface - Réactivité 	Simple 1s	/ Maximum	L'utilisation doit pouvoir se faire sans explication de fonctionnement et avec une rapidité convenable.
FS7	Être stable sur son support.	<ul style="list-style-type: none"> - Stabilité 	Support horizontal	± 10°	Il devra être stable sur une surface horizontale.
FS8	Respecter les normes en vigueur.	<ul style="list-style-type: none"> - Normes françaises et européennes 	/	/	/

9) Résumé des fonctions techniques

FS1 : Permettre à l'utilisateur de partager la musique sur un système audio

FT1 : Gérer des requêtes de musique sur le serveur

FT2 : Diffuser la musique demandée

FS2 : Permettre à l'utilisateur de contrôler le jukebox depuis son smartphone

FT3 : Permettre à l'utilisateur d'avoir accès aux fonctionnalités du serveur

FT4 : Connecter l'utilisateur à l'interface

FT5 : Contrôler les fonctionnalités du jukebox

FT6 : Gérer des musiques par vote

FS3 : Alimenter le système

FT7 : Alimenter le système en énergie électrique (secteur)

FS4 : Résister aux contraintes extérieures

FT8 : Choisir des matériaux adaptés et résistants aux chocs pour le boîtier

FS5 : Être simple d'utilisation et intuitif

FT9 : Développer une interface web simple (graphiquement)

FT10 : Assurer une utilisation rapide

FS6 : Être visuellement agréable

FT11 : Faire en sorte que le boîtier attire le regard (couleurs et design)

FT12 : Permettre à l'application d'avoir une interface graphique propre et agréable au niveau des différents menus

FT13 : Dimensionner le boîtier pour qu'il soit transportable

FS7 : Être stable sur un support

FT14 : Faire qu'il tienne sur tout support horizontal

FT15 : Assembler les différents composants électroniques du jukebox

FS8 : Respecter les normes en vigueur

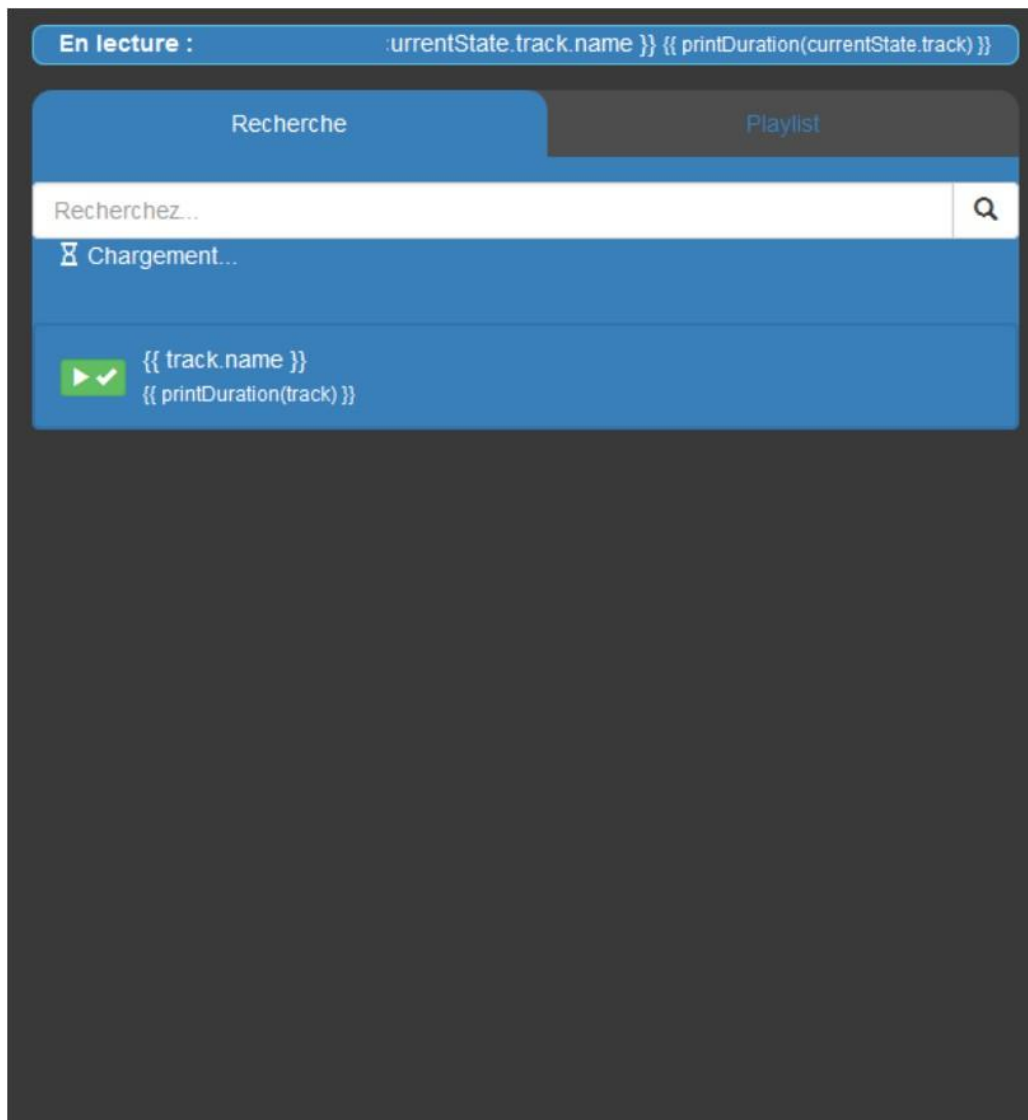
FT16 : Respecter les normes en vigueur (dans ce domaine, pour la France)

10) Ma partie au sein du projet

Dans notre projet nous avons choisi de concevoir un boîtier contenant une carte son (pour avoir une meilleure qualité sonore et permettre aux utilisateurs de connecter des enceintes), un Raspberry pi 3 (qui va avoir pour fonction d'être le serveur, il va permettre aux utilisateurs de se connecter et de pouvoir rechercher leurs musiques) et une carte NFC (pour faciliter la connexion au boîtier)

Dans notre projet je me suis occupé de tester et de choisir la meilleure extension de Mopidy.

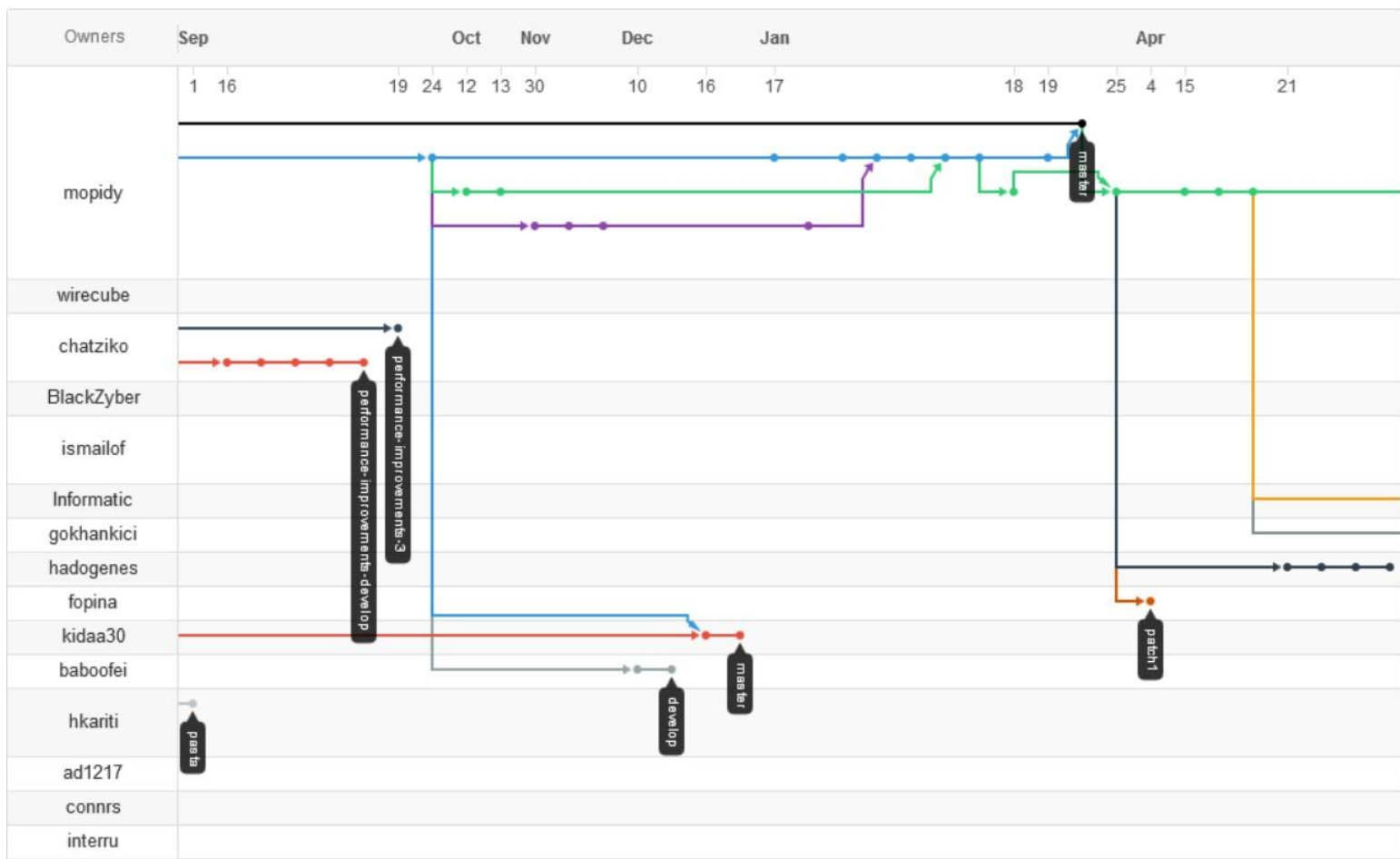
Mopidy est le serveur qui est installé sur le Raspberry pi 3, lui permettant ainsi de rechercher les musiques que lui demande l'utilisateur. Nous avons choisi de prendre Mopidy pour gérer les différentes requêtes de l'utilisateur car il est en « open source » (il n'a aucun droit d'auteur) et possède de nombreuses options. L'utilisateur effectue les recherches à partir d'une interface web que nous avons créé spécialement pour lui et qui est divisée en deux parties qui sont « recherche » et « playlist », ainsi l'utilisateur peut chercher la musique de son choix et ensuite l'ajouter à une playlist que les différents utilisateurs connectés au Raspberry pi 3 pourront voir et ainsi voter pour les musiques qu'ils souhaiteront passer.



Mais Mopidy possède beaucoup d'extensions mises à jour par les différents utilisateurs et j'ai dû toutes les tester pour voir celle qui correspondait le mieux au rendu que l'on voulait.

Comme on peut le voir ci-dessous il y a l'extension de base Mopidy et ensuite les différentes extensions que les utilisateurs lui ont rajoutées. Extensions toutes indépendantes les unes des autres. J'ai dû tester ces extensions en prenant plusieurs critères en compte qui sont les suivants :

- La rapidité de recherche que nous avons définie à 1s dans le cahier des charges
- Si la musique souhaitée est trouvée.
- Quand la recherche de la musique est terminée l'utilisateur doit pouvoir l'ajouter à la playlist lui permettant ainsi de l'écouter et de la partager avec les autres utilisateurs connectés.



Ainsi grâce à l'invite de commande CMD connectée au Raspberry pi 3 j'ai pu installer les différentes extensions de Mopidy et tester le temps de requête pour chaque extension.



La recherche de musique se fait sur youtube, donc il suffit d'installer l'extension voulue, de lancer plusieurs recherches et de vérifier le temps moyen de réponse pour chaque extension et si chaque musique peut s'ajouter à la playlist.

J'ai ainsi pu réaliser en même temps mon AMS (Attendu Mesurer Simuler)

J'ai choisi de tester chaque extension sur 10 recherches avec pour critères ceux que j'ai cité précédemment. Cela m'a permis d'en déduire que la meilleure extension est celle nommée « informatic » car c'est celle qui possède le plus bas temps de recherche et le moins d'erreurs lors de l'ajout de musique à la playlist comme le montre les exemples ci-dessous.

Mopidy-youtube -base										
temps de recherche (s)	5,8	8,1	4,5	5,6	5,1	4,8	4,8	5,3	5,2	6
moyenne temps de recherche	5,52									

Ici on peut voir que l'extension de base de Mopidy est à 5.52s de temps de recherche moyen, mais nous cherchons à avoir 1s de temps de recherche moyen.

Mopidy-youtube -ismailof										
temps de recherche	6,4	5,9	5,2	4,5	5,2	5,74	5,8	5,5	5,8	5,6
moyenne temps de recherche	5,564									

Mopidy-youtube -informatic										
temps de recherche	0,7	1,3	0,8	0,7	0,9	0,8	0,9	0,6	0,8	1
moyenne temps de recherche	0,85									

On peut ainsi clairement voir que la meilleure extension parmi toutes celles qui sont mises à disposition est l'extension « informatic ». C'est ainsi que nous avons décidé de prendre cette extension pour faire gérer les requêtes des différentes musiques.

11) Conclusion

Au cours de cette année ce projet m'a permis de découvrir de nouvelles notions me permettant ainsi de faire avancer le projet.

J'ai pu découvrir la gestion d'un serveur et des termes que je ne connaissais pas. Ce projet m'a aussi permis de découvrir le travail de groupe, la répartition des tâches et l'entraide en cas de difficulté.

Enfin je dirais que cela m'a aussi permis d'approfondir mes connaissances dans différents domaines et de changer ma vision de ce qu'est la gestion de serveurs en informatique.

