

BERRUYER William
PORTE Eden
GROS Antoine
LARGUIER Lukas
MEASSON Emerik
BAPTISTA Alain

Groupe C1

CloudStudio

Dossier 1ère itération - Projet Semestre 3

Table des matières

1. Cadrage du projet.....	2
1.1 Description du projet.....	2
1.2 Objectifs du projet.....	2
1.3 Contexte du Projet.....	3
1.4 Parties prenantes.....	3
1.5 Contraintes.....	4
1.6 Risques	4
2. Expression du besoin	5
2.1 Besoins fonctionnels et non fonctionnels	5
2.2 Critères qualité	8
3. Solutions	9
3.1 Maquettes du site.....	9
3.2 Technologies envisageables	11
3.3 Organisation du travail	12
Glossaire.....	13

1. Cadrage du projet

1.1 Description du projet

Le projet consiste à aider des musiciens à créer leur musique en groupe. La création d'un morceau implique plusieurs personnes, des musiciens, des producteurs, des ingénieurs, avec des tâches différentes, mais chacune dépendant d'une ou plusieurs autres tâches. Pour travailler correctement, les créateurs travaillent ensemble, dans la même pièce. Lors du confinement, c'était complètement impossible pour eux de se voir, ce qui fait que leurs projets étaient souvent en attente, et pour le peu de projets qui ont continué, la communication entre les différents participants était compliquée. Ils devaient tous envoyer leur partie du travail aux autres avec des outils qui ne sont pas optimisés pour cela, ce qui est assez pénible et contraignant. En effet, chaque personne se retrouvait avec beaucoup de versions du projet différentes, et il leur était alors compliqué d'avancer avec des versions différentes ou à des moments différents.

Le but de notre projet est de palier aux problèmes des différentes versions, de la sauvegarde et de l'optimisation d'échange de fichier pour ce public.

Le projet se fera sous la forme d'un site web, car les musiciens travaillent sur ordinateur ce qui rend un ce choix « logique », car il correspond à l'environnement de travail des utilisateurs.

De plus cela permettra qu'il soit accessible à tous facilement, peu importe la plateforme sur laquelle l'utilisateur travaille, et peu importe le matériel des utilisateurs

1.2 Objectifs du projet

⇒ Le projet permet la coordination et le transfert de fichier audio entre tous les participants d'un projet musical, mais également la création de versions d'un même projet.

De manière plus détaillée, un musicien pourra poser un fichier sur le site et n'importe quel autre participant pourra récupérer ce fichier, pour le modifier par exemple, mais les dernières versions restent disponibles en cas d'erreur de l'utilisateur ou de mécontentement de son travail.

⇒ L'objectif principal est de faire une interface simple permettant aux utilisateurs de se transmettre des fichiers audios de manière sécurisée à travers un groupe et d'organiser l'avancée de leur projet.

⇒ Ce projet est réaliste, car nous sommes un groupe de 6 étudiants, et nous connaissons déjà suffisamment les langages de programmation nécessaire pour commencer le projet, voir même bien l'avancer sans énormément de difficultés au départ, il y aura néanmoins un certain challenge, car nous avons beaucoup de choses à découvrir, surtout pour ce qui est de l'hébergement du site web, mais aussi pour les fonctions plus poussées du site.

⇒ Au niveau du temps, nous avons une date butoir qui nous laisse 17 semaines de projet, mais le projet sera découpé en plusieurs parties pour une meilleure organisation. Chaque élève aura son travail et une durée limitée pour le faire. S'il est en difficulté, un autre l'aidera pour ne pas perdre trop de temps et avoir fini les fonctionnalités principales au minimum pour la fin du projet.

1.3 Contexte du Projet

Analyse critique de l'existant :

1. Inexistence de plateforme permettant de collaborer sur un projet musical de manière aussi poussée.
2. Les plateformes qui se rapprochent de notre idée possèdent certains points positifs par exemple :
 - La possibilité de faire des versions d'un projet afin de revenir en arrière.
 - La possibilité d'ajouter des collaborateurs à un projet.
 - La possibilité de synchroniser ses fichiers dans le cloud
3. Mais elles possèdent également de multiples défauts par exemple :
 - Insuffisance du système de collaboration qui se limite uniquement à un système de partage de fichier et qui limite beaucoup les possibilités d'échanges d'idées entre les acteurs travaillant sur le projet.
 - Surcharge d'information inutile et non intuitive pour les débutants
 - Inadapté à la production de projet musical complet, ces plateformes sont plutôt orientées vers la production de petit projet musical impliquant très peu d'acteurs et ne visant pas à obtenir un résultat professionnel.

Notre plateforme se veut plus orientée vers la collaboration ainsi que l'optimisation de l'organisation des informations relatives à un projet afin d'aboutir à une gestion de projet musicale menant à un résultat professionnel.

1.4 Parties prenantes

Le domaine d'activité ciblé par notre projet est assez vaste. Nous souhaitons que notre application web puisse être utilisée aussi bien par des professionnels du milieu de la musique (artistes, groupes, ingénieurs) que par des amateurs voulant créer leur propre projet tout seul ou avec des partenaires sans qu'ils aient de difficultés par rapport à la gestion des nouvelles versions de leur projet.

C'est pour cela que notre application doit être facile d'accès à n'importe qui, qu'il soit ou non familier avec l'outil informatique et les outils de musique numérique.

Notre application finale se doit d'être collaborative, afin de permettre à chaque utilisateur, à chaque membre d'un même projet de pouvoir communiquer ensemble durant le suivi de leur projet et ainsi permettre une meilleure organisation et un risque plus faible d'avoir des informations différentes entre chaque membre du groupe, car leurs fichiers seront centralisés sur une même plateforme, et ils pourront commenter chaque nouvelle version pour savoir laquelle est la bonne et les modifications ayant été apportées dessus. Les utilisateurs attendent de notre plateforme un site fiable et sécurisé afin d'avoir les bonnes informations au bon moment sans pour autant risquer une perte de donnée ou que d'autres utilisateurs puissent avoir accès à leurs données.

Durant ce projet nos compétences doivent englober plusieurs domaines, tout d'abord nous devons faire un site web cohérent et moderne, de plus il doit être facile à utiliser, ce qui fait qu'il doit avoir un design "responsive", mais également adaptée au public visé, nous ajoutons à cela la compétence de gestion de base de données, que nous devons créer et mettre en place sur notre plateforme, et ce de manière sécurisée.

Durant ce projet nous ne nécessitons pas l'intervention de sous-traitants, car notre équipe comporte des membres spécialisés dans chaque domaine ou langage que nous utiliserons.

1.5 Contraintes

La première contrainte qui se pose à nous est la gestion de la synchronisation des fichiers et version du projet. En effet, chaque membre dun projet doit avoir accès aux mêmes versions, nommées de la même façon, et il ne faut pas quil y ait de "décalage", entre les utilisateurs. Pour gérer cela, il faut que chaque fois quun fichier est mis à jour, la base de données actualise automatiquement la version mise précédemment sur la plateforme.

La seconde contrainte est celle de permettre à chaque utilisateur davoir, grâce à son compte, son espace personnel, où apparaissent les différents projets auxquels il participe, mais aussi son profil, avec ses informations. Pour accéder à son espace, utilisateur doit rentrer son identifiant, ainsi que son mot de passe. Pour permettre un tel service, nous devons stocker dans une base de données les informations de utilisateur, pour vérifier laauthenticité de ces dernières lorsquils se connecteront.

La troisième contrainte est celle du stockage des fichiers. Effectivement, il faut que les fichiers uploadés par les utilisateurs se stockent correctement dans la base de données, et que les versions les plus récentes neécrasent pas les anciennes. Pour les fichiers de projet globaux (à la différence des pistes par pistes), il faut que les utilisateurs puissent écouter une démo directement sur la plateforme. Une autre contrainte est celle des rôles. En effet, chaque utilisateur va, dans un projet, être attribué à un ou plusieurs rôles (Chanteur/Guitariste/Ingénieur Son/etc.), et il faut que chaque rôle soit facilement identifiable, et que les utilisateurs puissent modifier uniquement les fichiers les concernant.

Il faut de plus, que les fichiers ne puissent pas être accessibles par une tiers personnes, ne faisant pas partie du projet. Pour cela, laccès aux projets et autres fichiers est sécurisé par une connexion nécessaire pour identifier utilisateur.

Une autre contrainte est celle de linterface. Les utilisateurs étant pas forcément des usagers courants de loutil informatique, il faut que cette dernière soit la plus simple possible, afin doptimiser le temps de travail des utilisateurs.

Il faut également que le serveur sur lequel est hébergé la plateforme tourne 24h/24, pour permettre aux usagers de travailler à toute heure.

De plus, pour anticiper de potentiels dégâts physiques sur le serveur hébergeant la base de données, il faut réaliser régulièrement une sauvegarde de cette dernière.

1.6 Risques

Le premier risque, le plus important, est celui de la perte de données. Il est en effet possible que, si la plateforme ne fonctionne pas correctement, les données uploadées par utilisateur ne soient pas correctement sauvegardées.

Un autre risque majeur est celui de la mise à jour des versions, qui, si elle nest pas faite correctement, peut entraîner au minimum une confusion entre les utilisateurs, et donc du temps perdu, au pire, des pertes de données, etc.

Enfin, un risque, soit quune tierce personne ait accès au projet, par le biais des identifiants dun utilisateur présent sur le projet, ou de toute autres façons, et que cette dernière supprime, intentionnellement ou non, des fichiers du projet. Cest pour cela quune sécurité va être mise en place, nécessitant lapprobation de plusieurs utilisateurs, participant au projet, avant la suppression définitive dun fichier.

2. Expression du besoin

2.1 Besoins fonctionnels et non fonctionnels

Le site doit être entièrement rédigé en anglais, car l'anglais est la langue de référence en matière de production de musique, en particulier assistée par ordinateur. De plus l'anglais permettra de toucher un maximum de public.

Le site doit permettre à l'utilisateur de créer un compte ou de se connecter à son compte si celui-ci en possède déjà un. L'utilisateur doit également pouvoir modifier les informations de son compte telles que son nom, son nom d'artiste, sa photo de profil, son adresse mail, son mot de passe, ses préférences en termes de notification.

L'utilisateur doit pouvoir se déconnecter de son compte.

Le site doit permettre à l'utilisateur de créer des projets, un projet correspond au travail sur un morceau de musique, mais doit aussi pouvoir créer des groupes de projets (l'équivalent d'un album). L'utilisateur doit pouvoir accéder à ses projets depuis le site une fois qu'il est connecté à son compte. Dans chaque projet l'utilisateur doit pouvoir accéder à différentes sections correspondant à l'étape d'avancement de son projet, ces étapes sont :

- Raw Session qui correspond à une version brute du morceau sans effets ni post-production.
- Mix qui correspond à une version mixée du morceau (c'est-à-dire une version travaillée par l'ingénieur son).
- Master qui correspond à une version Masterisé du morceau c'est-à-dire une version travaillée par l'ingénieur Master pour correspondre aux standards et aux normes de l'industrie. À la fin de cette dernière étape, le morceau est prêt à être publié

À l'intérieur de chaque section, l'utilisateur doit pouvoir créer des versions de cette section afin d'avoir des backups du projet au cas où l'utilisateur voudrait revenir sur une version antérieure à la version sur laquelle il travaille. Dans chaque version de son projet l'utilisateur doit pouvoir importer sur le site le fichier de projet correspondant à son morceau, ainsi que chaque piste du morceau s'il le souhaite, mais aussi une version de démonstration du morceau permettant d'écouter le rendu de la version directement sur le site. Pour chaque version l'utilisateur doit pouvoir rédiger une description de la version afin de mieux l'identifier et de la distinguer parmi les autres. L'utilisateur doit également pouvoir laisser un commentaire pour chaque version s'il veut s'exprimer sur le contenu de celle-ci. L'utilisateur doit pouvoir écouter les pistes du morceau une par une s'il le souhaite, néanmoins il faut que celle-ci ait été importée à l'avance. L'utilisateur doit pouvoir télécharger chaque piste d'une version, une par une, plusieurs à la fois ou toute d'un coup ; il peut également télécharger le fichier de projet de la version. Chaque version doit afficher sa date de création.

Un utilisateur qui crée un projet en est le propriétaire est donc l'administrateur de celui-ci, cet utilisateur doit pouvoir ajouter des collaborateurs à son projet, mais aussi leur attribuer des rôles (Guitariste/Batteur/etc.).

L'utilisateur doit pouvoir, dans chacun de ses projets, accéder à un chat général accessible par tous les collaborateurs du projet. Ce chat permet de discuter à propos du projet dans sa globalité. L'utilisateur doit être informé par mail des messages échangés dans le chat.

L'utilisateur doit modifier les paramètres de ses projets, tels que l'image du projet ou son nom, il doit également pouvoir supprimer ses projets s'il le souhaite.

Lors de son inscription, l'utilisateur doit renseigner son nom, son prénom, sa date de naissance, son nom d'artiste, son adresse mail, et créer un mot de passe. Les informations servant à l'accès à son compte sont son adresse mail et son mot de passe.

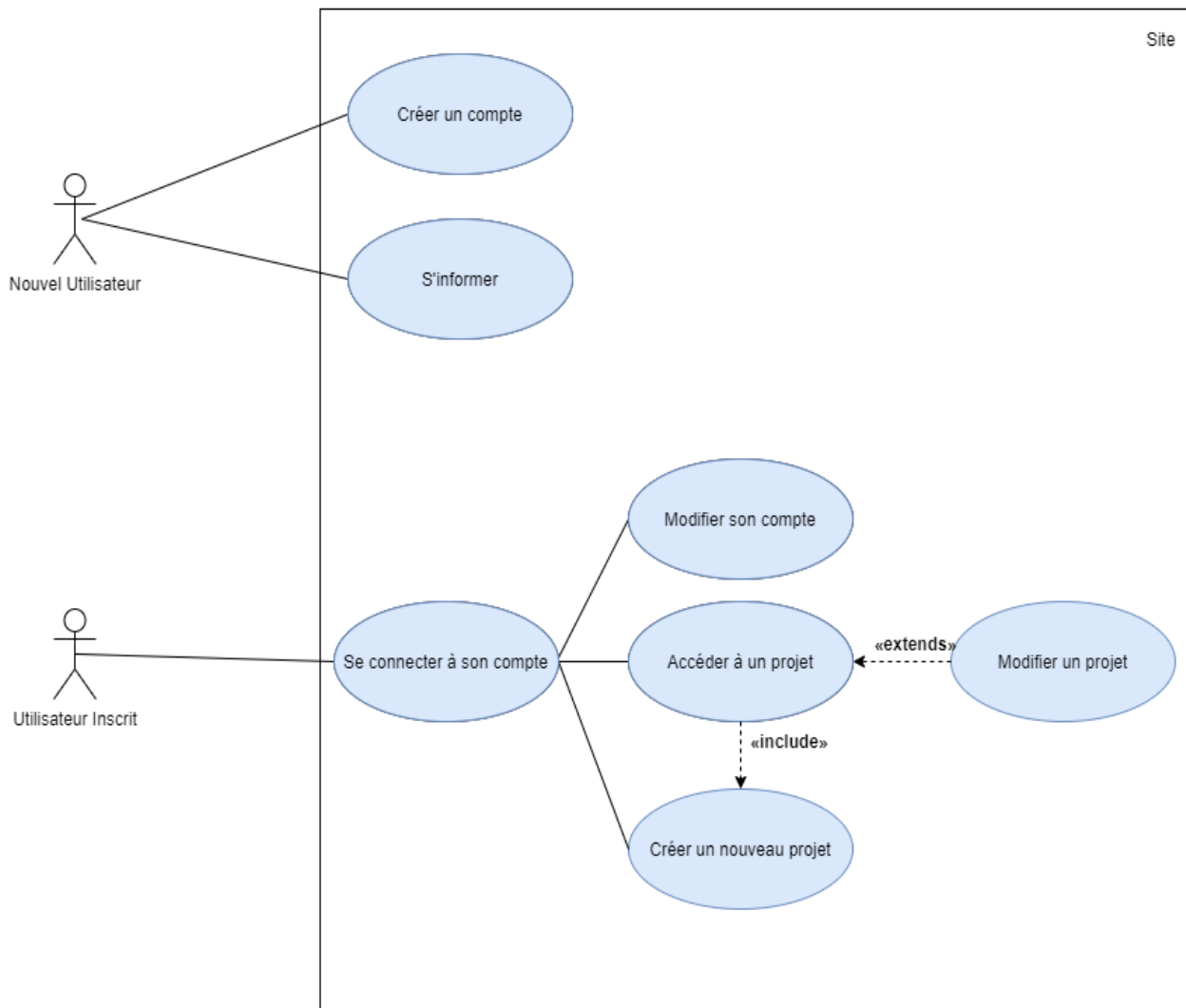


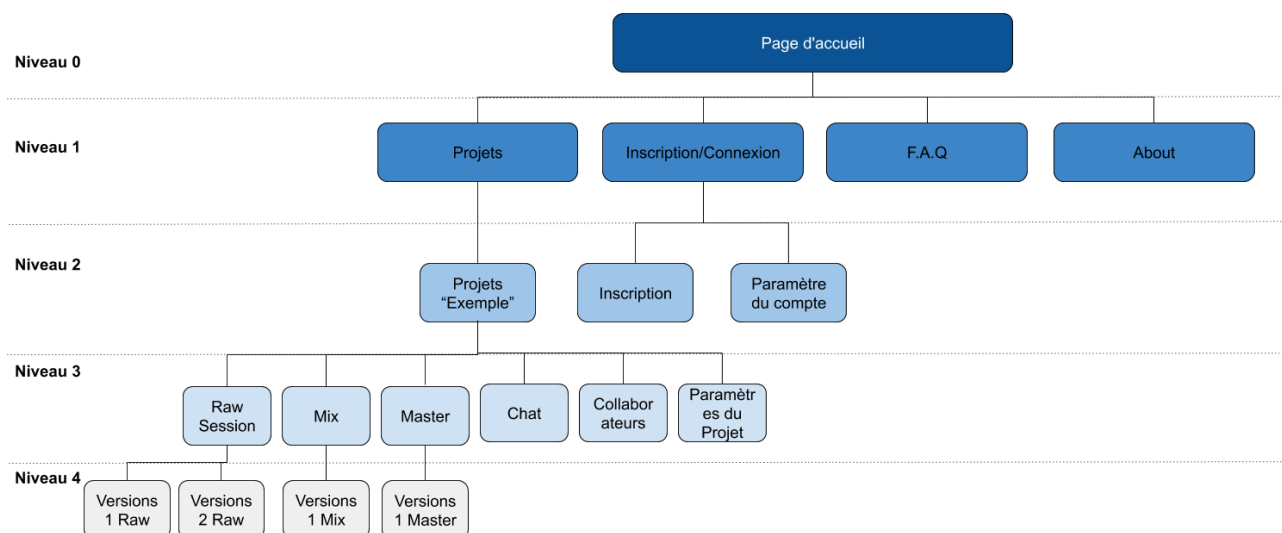
Diagramme de cas d'utilisation

Toutes ces fonctions n'ont pas la même importance dans le développement du site, certains sont essentiels à son fonctionnement alors que d'autres ne sont que des ajouts apportant de petites améliorations au site. Nous avons donc hiérarchisé ces besoins afin de visualiser efficacement lesquels sont les plus importants.

BESOINS FONCTIONNELS	
Priorité	Définition du besoin
1	Créer un compte utilisateur
1	Créer / Supprimer un projet pour un morceau
1	Modifier les informations de son compte
2	Diviser son projet en différentes étapes (Raw Session, Mix, Master)
2	Import /Export des fichiers et pistes nécessaires à la conduite du projet
2	Créer / Supprimer une version du projet
2	Ajouter/supprimer des collaborateurs à un projet (recherche par nom d'artiste)
2	Rédiger une description pour chaque version du projet créé
3	Import et possibilité d'écouter de la démo d'un morceau et des pistes dans une version du projet
3	Envoyer / Lire des messages dans un chat global à chaque projet
4	Définir des rôles pour les collaborateurs d'un projet
4	Créer/Supprimer des groupes (album) pour rassembler des projets
4	Notifications de réceptions des messages par mail
4	Tutoriel pour les nouveaux utilisateurs
4	Envoyer / Lire des commentaires pour chaque version d'un même projet

Les besoins surlignés en bleu doivent impérativement être développés pour le prototype final.

BESOINS NON FONCTIONNELS	
Priorité	Définition du besoin
1	Menu ergonomique (sur ordinateur / tablette / téléphone) et utilisation d'un lexique compréhensible pour les débutants
1	Sécurité des bases de données (utilisation de vues, gestion des permissions en lien au site, et à chaque projet)
1	Avoir accès à un fichier à jour dès qu'il est déposé
1	Site toujours en ligne et avec une sauvegarde sur un disque dur en cas de panne
2	Un compte administrateur pour avoir la liste des utilisateurs et leur(s) projet(s)
2	Code bien réparti et clair (avec des commentaires pour aider à la compréhension) afin de faciliter la testabilité et la maintenance du site
3	Service gratuit pour profiter à tous
3	Site traduit en français



Arborescence du site

2.2 Critères qualité

La facilité d'apprentissage : Notre plateforme doit permettre aux utilisateurs visés (les musiciens) de prendre rapidement en main son interface ainsi que la signification des différents menus et des différentes fonctions.

La facilité de compréhension : Notre plateforme doit être facilement compréhensibles pour tout utilisateur qui voudrait s'en servir qu'il fasse partie du public visé ou non. Nous devons limiter l'utilisation de terme trop technique et implémenter une page permettant d'aider un utilisateur à comprendre l'interface si celui-ci est en difficulté.

La sécurité : Notre plateforme doit empêcher tout accès non autorisé, avec l'utilisation d'identifiant et de mot de passe pour se connecter à son compte, mais il doit également empêcher tout accès non autorisé aux données stockées sur notre plateforme.

L'exactitude : La plateforme doit être précise du stockage des données afin d'être fonctionnel.

L'aptitude : Les fonctionnalités principales de la plateforme doivent toutes être parfaitement fonctionnelles pour que la plateforme fonctionne.

L'efficacité des temps de réalisation : L'import et l'export de fichier doivent être assez rapides pour maintenir une bonne productivité du point de vue de l'utilisateur, l'utilisation des autres fonctionnalités telle que la création ou l'ouverture d'un projet doivent également être rapides.

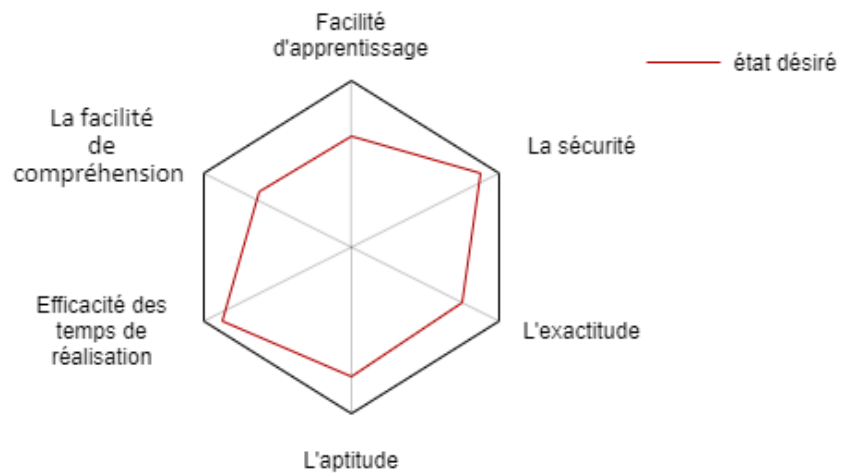
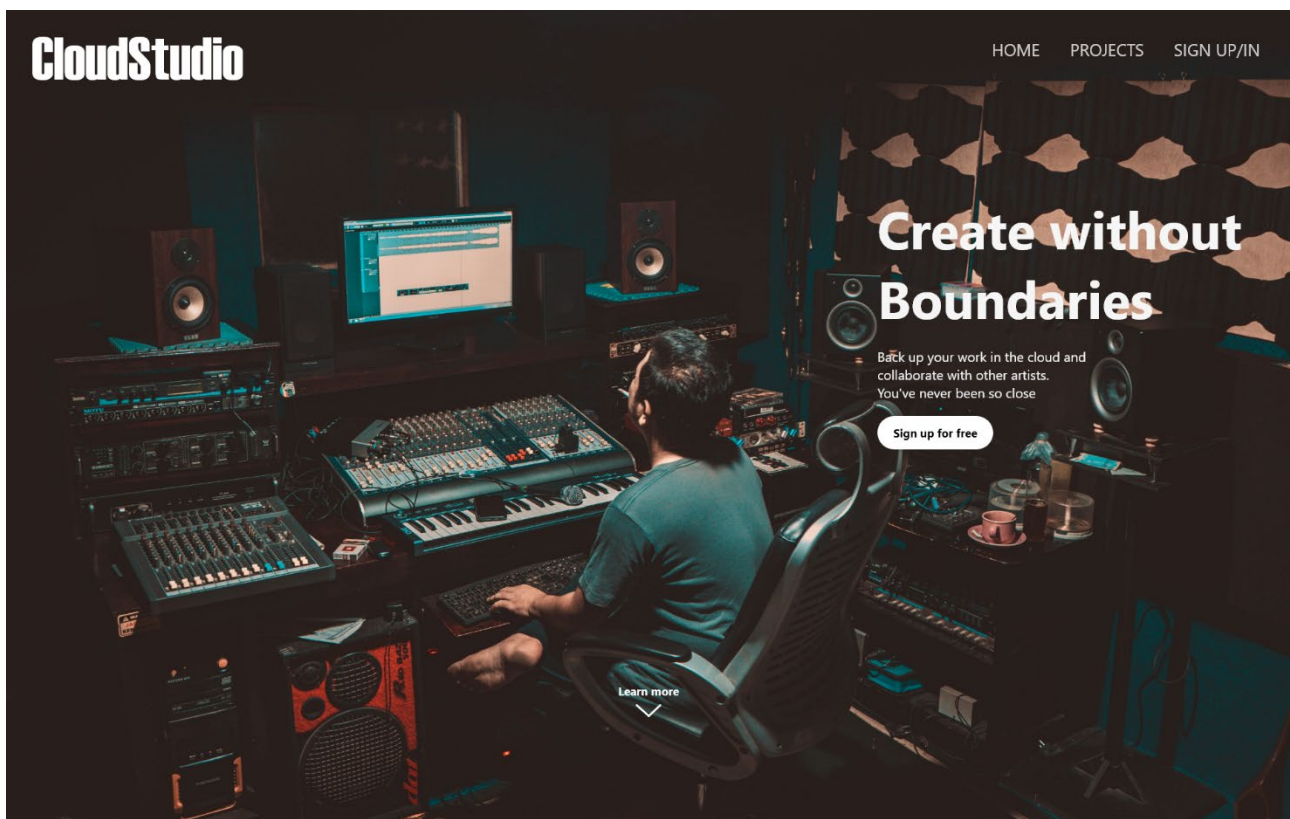


Diagramme d'indicateur de réussite

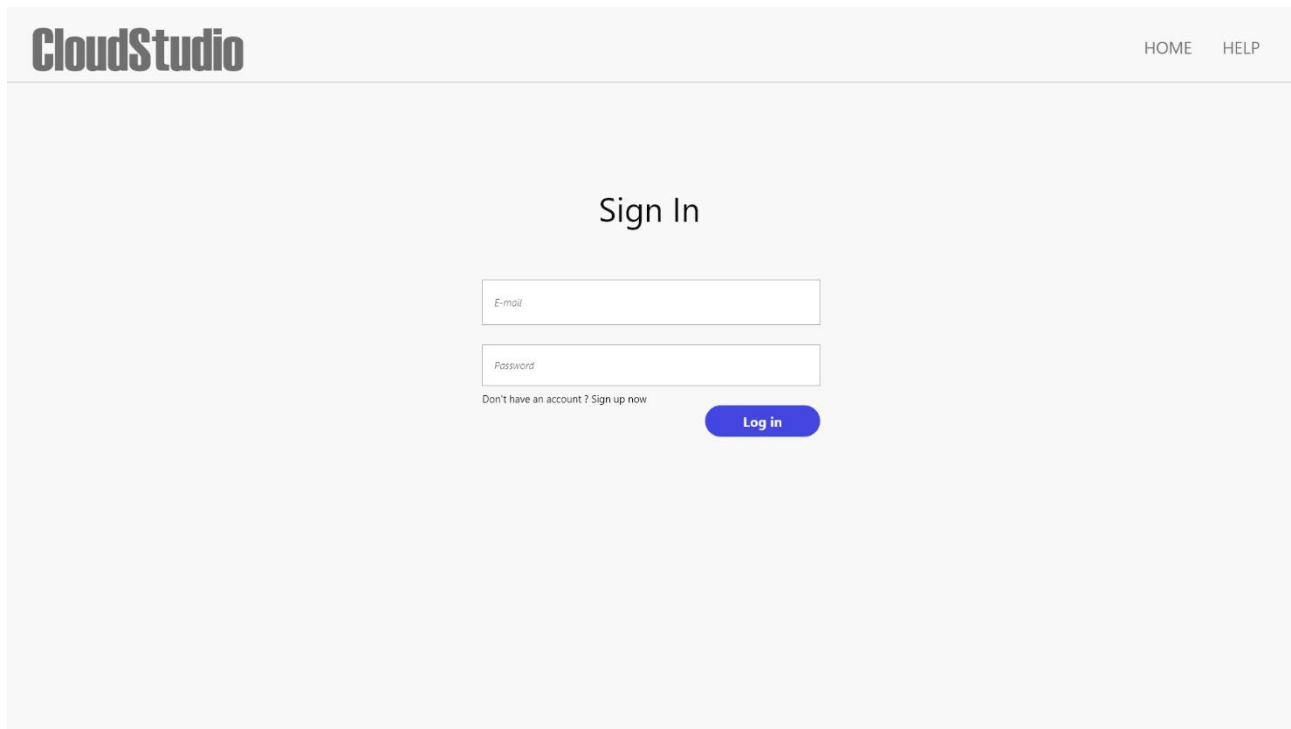
3. Solutions

3.1 Maquettes du site

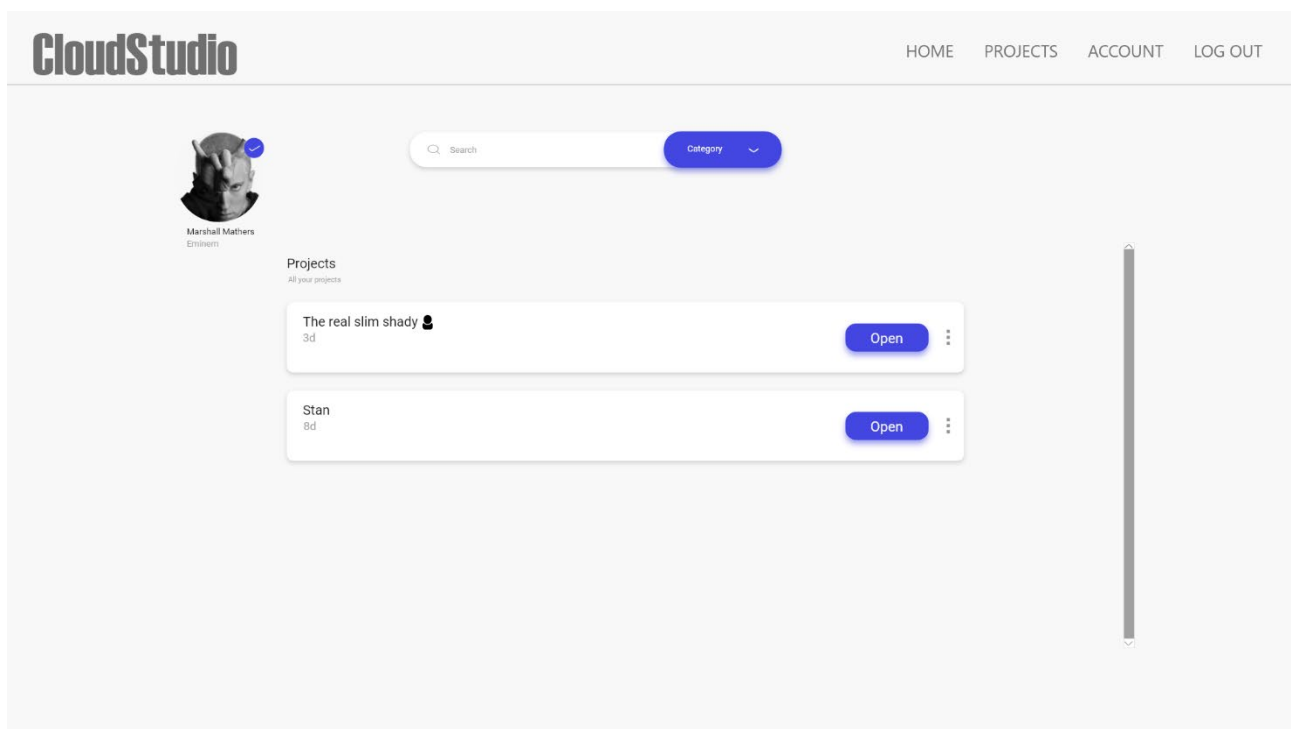
Voici tout d'abord les maquettes prototypes des différentes pages principales de notre futur site web :



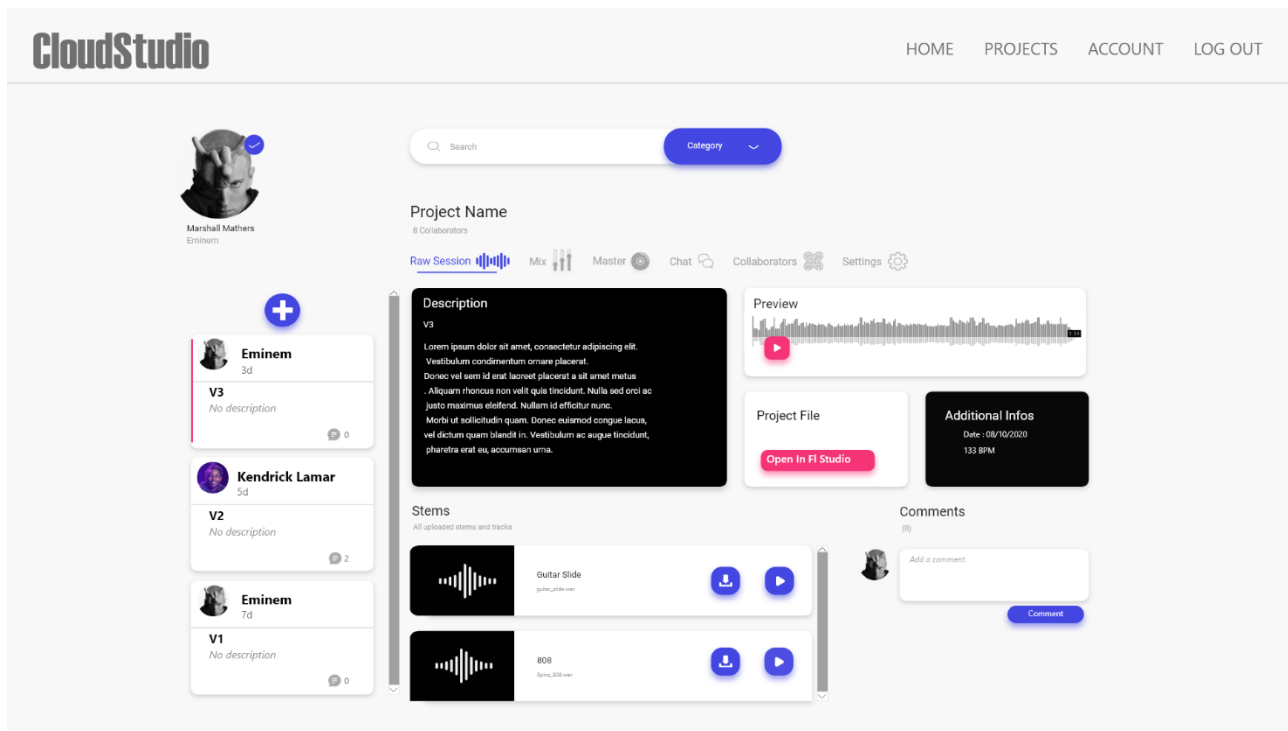
Page d'accueil



Page d'inscription



Page de sélection des projets



Page d'un projet

3.2 Technologies envisageables

Afin de réaliser ce site web, il y a un très grand nombre de solutions techniques envisageables, divisées en plusieurs types. Nous avons pour le moment des préférences sur chacun de ces types, mais nous pourrions très bien décider de changer de technologie si dans les quelques semaines à venir, nous trouvons une meilleure alternative pour résoudre nos difficultés de conception.

- Front end

Côté Client, il n'y a pas beaucoup de choix sur les langages à utiliser, qui sont HTML, CSS et JavaScript. Il y a cependant possibilité d'utiliser des CMS comme WordPress qui génèrent le code HTML, mais utiliser les langages directement permet de gagner en précision et en contrôle de l'apparence du site web pour le confort du client. Cependant il reste envisageable d'utiliser des Frameworks comme Bootstrap afin de faciliter le développement HTML et responsive du site.

- Back-end

Côté serveur, il existe un très grand nombre de technologies différentes, PHP, nodeJS, python, JAVA, etc. PHP semble le meilleur choix, car c'est le langage le plus utilisé aujourd'hui dans le développement web, il y a donc beaucoup de documentation et d'aide disponibles pour ce langage, sans oublier que nous l'apprenons tous en ce moment même, ce qui pourra nous permettre de nous entraider en cas de problème avec ce langage.

Il y a aussi les bases de données qui sont gérées côté backend, il existe donc plusieurs technologies : MySQL, PostgreSQL, SQLITE, etc.

Ici, nous souhaitons utiliser MySQL, car nous trouvons de nombreux tutoriels et cours en français sur MySQL, ce qui rendra l'utilisation de cette technologie plus simple pour nous que n'importe quelle autre technologie, de plus, cette technologie est gratuite, réputée et une des plus utilisées dans le monde pour les bases de données.

- **Gestion/conception du projet** : Afin de gérer le projet, il existe des outils de contrôle de versions comme Git et GitLab, qui sont indispensables afin de permettre à toute équipe d'être à jour sur le projet et de pouvoir revenir en arrière à tout moment si nous devons annuler ou recommencer l'implémentation d'une fonctionnalité. De plus, si les données sont perdues ou si le serveur web plante, les données du site seront toujours en sécurité sur le dépôt distant.

Lors de la conception du projet, nous aurons également besoin de réaliser des diagrammes UML, nous utiliserons donc Visual Paradigm, car nous avons déjà été initiés à ce logiciel l'an dernier et ça nous évitera donc d'apprendre à utiliser un autre logiciel.

Il y a également la question de l'hébergement qui se pose. En effet, le site web devra être stocké sur un serveur, constamment allumé et relié à internet afin de pouvoir fournir à tout moment les pages web aux clients.

La première possibilité est donc d'aller sur des services d'hébergement. Cependant, afin d'avoir un bon serveur qui peut gérer correctement des fichiers assez gros et nombreux ainsi que supporter un grand nombre de personnes connectées, il faudrait se diriger vers des services d'hébergement payants, car ceux gratuits risquent de ne pas fournir un serveur d'assez bonne qualité.

Azure ou AWS sont deux solutions que nous envisageons afin d'héberger notre site ainsi que notre base de données.

La deuxième possibilité, qui est donc celle vers laquelle nous nous dirigeons actuellement, est d'héberger notre site web nous-mêmes, sur un Raspberry PI à l'aide d'un serveur Apache de préférence, car c'est le plus utilisé.

En effet, ces nano ordinateurs consomment très peu d'énergie, ce qui leur permet de rester allumés 24h/24 facilement, et il est très facile de les contrôler à distance et d'échanger des fichiers avec eux de n'importe où, s'ils sont reliés à internet avec la fibre optique.

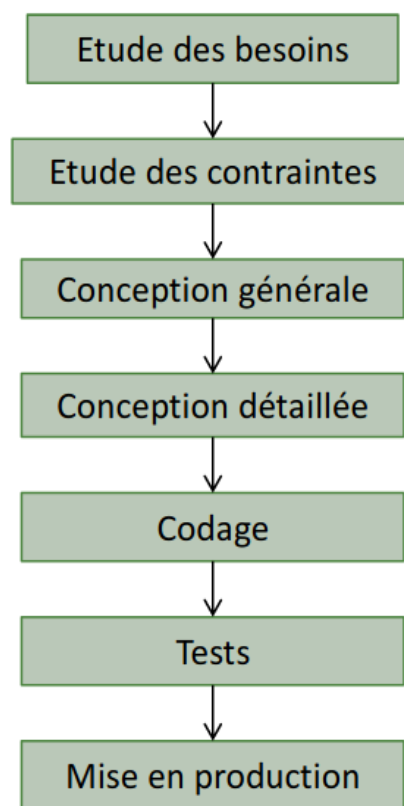
3.3 Organisation du travail

Dans notre projet, le modèle de développement que nous souhaitons utiliser est celui de la cascade.

Ce modèle nous permet de tous nous concentrer ensemble dans un premier temps sur la conception générale puis détaillée du projet après avoir étudié les besoins et les contraintes.

De cette manière, tout le monde a une vision globale du projet sous tous ses aspects avant de commencer à écrire la moindre ligne de code.

Cependant, il restera possible que nous réalisons que nous avons fait une erreur de conception pendant le codage, ce qui nous forcera à revenir en arrière sur l'étape de conception, mais comme nous ne serons pas très nombreux, il sera facile de prévenir et d'expliquer à tout le monde les modifications dans la conception et l'évolution des besoins.



Pour ce projet, la répartition des tâches sera la suivante :

4 étudiants feront les pages Web en HTML et CSS (Alain, William, Luka et Antoine), mais ils devront aussi gérer le PHP pour communiquer avec la base de données. Les 2 autres élèves (Eden et Emerik) s'occuperont de la base de données et de l'hébergement. L'analyse et la conception en amont se font avec tous les élèves réunis, afin que tout le monde ait une vision et une compréhension globale du projet.

Glossaire

- **Développement front-end** : conception de l'interface graphique utilisateur (ce que le client voit)
- **Développement back end** : code exécuté par le serveur pour générer la page web (ce que le client ne voit pas)
- **Raspberry** : nano ordinateur de la taille d'une carte de crédit, très peu puissant, mais très peu cher et consommant très peu d'énergie, très utile pour faire office de serveur.
- **Mix/Mixage** : Le mixage audio est l'opération technique et artistique par laquelle, dans les domaines de la musique, du cinéma, du jeu vidéo, de la télévision et de la radio, un certain nombre de sources audio sont mélangées afin de parvenir à un équilibre cohérent, en intervenant sur le niveau, légalisation, la dynamique et la spatialisation.
- **Mastering** : Le Mastering est le processus consistant à transférer un ensemble de registres pour en faire un programme sur un support physique ou un fichier informatique, lequel servira à une fabrication en série ou à la diffusion. Son but premier est de rendre homogène cet ensemble. Pour l'audio, l'approche diffère suivant que l'album est un original ou une compilation de différentes œuvres originales.
- **Framework** : En programmation informatique, un Framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.