

Dossier de projet professionnel



Réalisation de l'application web et mobile

–

Deki Eco-logistic

Présenté par Fabien PONZIO

SOMMAIRE :

1 / Introduction :

1.1 Présentation personnelle	4
1.2 Présentation de la Coding School La Plateforme	4
1.3 Présentation des projets en anglais	5

2 / Présentation des projets :

2.1 Présentation du contexte du projet	5
2.2 Liste des compétences couvertes par les projets	6
2.3 Analyse du problème existant	6
2.4 Résumé du projet	7

3 / Conception de l'application mobile

3.1 Cahier des charges du projet 8

3.1 a) Objectifs de l'application	8
3.1 b) Les cibles / utilisateurs du projet	8
3.1 c) l'équipement des cibles	9

3.2 Caractéristiques

3.2 a) Produits	9
3.2 b) Services	10

3.3 Contexte technique

3.3 a) Choix des langages	
3.3 b) Choix des frameworks	
3.3 c) Logiciels et autres outils	

3.4 Maquette

3.4 b) Charte graphique et UX	12
3.4 d) Prototype	13
3.4 e) outils	18
3.4 f) difficultés rencontrées et axes d'améliorations	19

4 / Conception et persistance des données en intégrant les recommandations de sécurité

4.1 Recommandations de sécurité	19
4.2 MCD	22
4.3 MLD	23

5/ Développement Back-end d'une application multi-couches en intégrant les recommandations de sécurité Deki

5.1 Collaboration à la gestion de projet

5.1 a) Fonctionnalités significatives	24
5.1 b) Organisation	25
5.1 c) Versioning et outils	25
5.1 d) Exemples de problèmes rencontrés	26

6/ Développement de la partie front-end d'une application : présentation de fonctions significatives Deki

6.1 Fonctionnalités significatives : Librairie Reactour	28
6.2 Problèmes rencontrés	33

8/ Conclusion

33

9/ Annexes

Introduction :

1.1/ Présentation personnelle :

Je m'appelle Fabien PONZIO et je suis en **reconversion d'études**. En effet, c'est après avoir étudié dans le **commerce** que j'ai intégré l'univers du **web** depuis bientôt 2 ans, c'est un choix qui a été naturel pour moi, de plus ce choix se trouve être en complète adéquation avec mon cursus précédent. Mon orientation vers ce milieu s'est faite grâce à ma curiosité et l'envie de disposer d'une autonomie en termes d'**informatique** et de **développement web**. J'ai suivi le cursus de la **Coding School** de la plateforme en 2020 et j'ai obtenu mon titre de développeur web et web mobile . Aujourd' hui je suis en Coding School 2 afin de préparer mon titre de concepteur / développeur d'application web et en **alternance au sein de Deki**.

1.2/ Présentation La Plateforme_ Coding School :

La Coding School est une formation web qui s'adresse à tous ceux qui souhaitent s'ouvrir les portes des métiers du numérique. Le modèle pédagogique unique de l'École la Plateforme, membre du label GEN, Grande École du Numérique, s'adapte aux besoins de chacun. Les évaluations se font par des contrôles continus sur des projets webs concrets réalisés seuls ou en groupes. L'intégration professionnelle est réalisée au travers de projets professionnels tutorés, portés par des entreprises. La coding school revendique une pédagogie active et inductive centrée sur l'apprenant et orientée projet.

1.3/ Présentation du projet en anglais :

- First project is : **Ultimate List** which was a **group project**. The main idea was to make an application that we can **create a Tier List** with. A tier list is a concept in video game culture where playable characters or other in-game elements are subjectively ranked by their respective viability as part of a list. Characters listed high on a tier list of a specific game are considered to be powerful characters compared to lower-scoring characters, and are therefore more likely to be used in high-level competitive settings like tournaments. In our case we wanted to adapt this concept to the music universe where users can classify their favorite artists, by genre, by albums etc.. This app also has a **social media aspect**, in fact, once a list is done it will appear on users' feed and can be liked, commented or even shared on other media etc ... Lists can be searched by using filters.

Deki is useful in order to reduce global carbon emission in France and maybe Worldwide. I mainly worked on the app that is delivered to clients using Stacks such as React.

- Deki is the project I've mostly worked on due to the fact that this is where I'm doing my **apprenticeship**. Deki's a 2 years old **startup** company based in the city center of Marseille, Deki's commercializing a **technologic tool** and a logistic solution for transport companies. By the end of 2022 French minister is planning to establish **ZFE** "zone à faible émission" which is the meaning of "Low Carbon Emission Zone" these "ZFE" will define areas in the city center of every french cities in order to lower motorized vehicles traffic during specific hours. Deki's been working towards this law approval. In fact, by choosing **decarbonated transport** and working with ecologic transport companies in a supply chain process, Deki's aspiring to have a 100% decarbonated transport model in big cities. My company receives parcels from multiple clients in their own hub and then works together with multiple delivery companies that are using bike fleet to deliver to the final client in the most respectful way towards the planet. The tool is deliver to dispatching companies, and final clients such as retailers, restaurants etc... Deki is useful in order to **reduce global carbon emission** in whole France and maybe Worldwide, I mainly worked on the app that is delivered to clients using Stacks such as React.

2. Présentation des projets :

2.1 Présentation du contexte des projets

Après avoir principalement fait du **développement web** au cours de ma **première année** de formation, la **deuxième année** allait porter sur le **développement d'applications**. Connaissant l'importance qu'occupent les applications mobiles sur le marché j'étais curieux à l'idée de pouvoir en développer une. Dans le cadre de notre deuxième année nous avons donc développé en équipe **UltimateList**, une **application mobile** de **classements musicaux**. Cette deuxième année de formation impliquait également un contrat d'**apprentissage en entreprise**, que j'ai effectué auprès de **Deki**. Tout au long de ce rapport j'alternerai donc entre **UltimateList** et **Deki** pour démontrer chacune des compétences requises au titre.

2.2 Liste des compétences couvertes par les projets

L'ensemble des compétences ont été couvertes par deux projets différents:

Le premier se nomme Ultimate List c'est une application réalisée en groupe dans le cadre de ma deuxième année au sein de la Plateforme_ c'est une application de création de Tier-List

Ce projet couvre les compétences du titre suivantes :

- Maquetter une application
- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web
- Développer la partie back-end d'une interface utilisateur web
- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données
- Concevoir une application
- Développer des composants métier
- Construire une application organisée en couches
- Développer une application mobile
- Préparer et exécuter les plans de tests d'une application
- Préparer et exécuter le déploiement d'une application

2.3 Analyse du problème existant

A chaque période de fin d'année des applications de **streaming** fournissent des **classements de l'année** en fonction de ce que nous avons écouté. A part partager à tout le monde ce classement généré automatiquement à une seule période de l'année il y a peu de moyen de faire part de ses goûts musicaux spontanément tout au long de l'année. C'est pourquoi nous avons décidé de développer cette application.

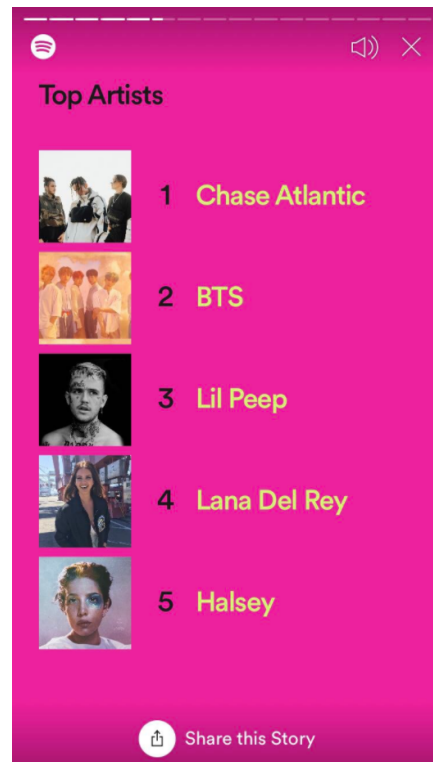
Deki réponds à une problématique bien plus préoccupante au quotidien, celle de la **pollution de l'environnement** et de la circulation de véhicules en centre ville de grandes agglomérations.

2.4 Résumé du projet

Ultimate List permet la création de **classements musicaux** ainsi que son **partage**. Une fois la liste publiée, d'autres utilisateurs qui possèdent un compte pourront interagir avec à la manière d'un **réseau social** (commentaires, likes, partages).

Après s'être inscrit puis connecté à l'application il sera possible d'accéder à la page de création de liste, à la page de fil d'actualités (feed), la page de recherche, et bien sûr le profil. Cette application n'est pas destinée à être commercialisée.

L'application s'inspire à la fois des réseaux sociaux traditionnels et des applications de musique. Nous voulions une **application ludique**, avec les mêmes codes que les réseaux sociaux sur lesquels nous nous rendons tous les jours voir toutes les heures, tout en apportant une **plue-value**, le suivi de son état d'esprit, sans rentrer dans des considérations scientifiques qui apporteraient un peu trop de sérieux à la démarche.



3 / Conception de l'application mobile

3.1 cahier des charges du projet (Ultimate List):

Il est nécessaire de définir ce qu'est une tier-list pour continuer.

*Une **tier-list** est un classement concernant tous les personnages jouables d'un jeu vidéo par ordre de viabilité. Elle permet alors de **classer**, de manière purement théorique, les personnages en fonction de leur potentiel, pour des joueurs les contrôlant de niveau parfaitement égal.*

*Dans notre cas, nous avons décidé de **revisiter le concept** en l'appliquant à un grand nombre de thèmes, afin de proposer aux utilisateurs de partager plus simplement ses différents avis et goûts.*

*Le thème de départ est la **musique**, pouvoir faire des tier list sur d'autres thèmes est un axe d'évolution que nous envisageons fortement.*

3.1 a) Objectifs du projet :

Dans sa première version, l'application permettra de créer des **tier-lists d'artistes, groupes de musique** et même **d'albums**. Elle sera amenée à évoluer sur d'autres thématiques, telles que des films ou acteurs de cinéma, recettes de cuisines, etc... Le but premier de cette version est de créer le système permettant la génération et le partage de ces tier list dans un **objectif communautaire**. En effet Ultimate List disposera à terme d'un volet de **réseau social**, les listes pourront être partagées facilement sur n'importe quel réseau ou même par message.

3.1 b) Public cible :

Le public ciblé est tout d'abord les fans de **musique**, peu importe le genre écouté. Permettre le partage de ces listes sur les **réseaux sociaux**, par message ou par mail peut par exemple permettre de générer des débats au sein d'une **communauté** de fans, générer des **flux commerciaux**, par exemple si un artiste partage sa liste d'artistes favoris.

3.1 c) Equipement des cibles :

Les millénials est une génération ultra connectée, en effet appelée parfois « **Digital Natives** », vit depuis toujours avec internet, le téléphone collé à l'oreille et connectée aux réseaux sociaux. En effet leurs taux d'équipement en ordinateurs, smartphones et tablettes dépasse de 5 à 10 points ceux de la génération précédente. Quant aux médias plus classiques comme la télévision et la radio, ils sont aujourd'hui jugés «

obsolètes». Les entreprises doivent donc utiliser les nouveaux moyens de communication pour atteindre leur cible.

La génération Y ayant grandi au milieu de services et d'applications nomades, rapides et individualistes, accorde une grande importance aux critères d'illimité et de rapidité.

3.2 Caractéristiques :

3.2 a) produit:

L'application se veut communautaire dans sa facilité de partage puis plus tard commenter et s'abonner et de commentaire. Les caractéristiques principales sont la **création et le partage de tier-list**. La possibilité de partager sa liste sur les différents réseaux sociaux des utilisateurs

Afin de nous démarquer dans le futur nous avons donc décidé de donner la possibilité à nos utilisateurs de pouvoir s'abonner entre eux afin de prendre connaissance de leurs différentes tier-list pour pouvoir étendre leurs connaissances, **découvertes musicales** et flux de streaming musical.

Les utilisateurs pourront à l'avenir également **aimer** et **commenter** des tier-list, pouvoir partager leur avis, débattre et rencontrer des personnes autour d'une **affinité musicale commune**. Ils pourront également s'intéresser aux goûts musicaux de leurs personnalité favorites ou même **découvrir** les préférences de ses **amis** ou **collègues de travail**, la musique est un sujet de conversation majeur et reflète beaucoup d'une personne.

3.2 b) Services:

Listes de nos différents services présent dans l'application :

Utilisateurs:

- Inscription / Connexion
(L'inscription et la connexion pourront s'effectuer via Google / facebook / Apple ainsi que par formulaire)
- Insertion d'informations supplémentaires (Lieu / présentation supplémentaires)
- Modification des informations
- Suppression des informations
- Affichage des utilisateurs

Tier-list:

- Création d'une tier-list
- Modification d'une tier-list
- Suppression d'une tier-list
- Affichage des tier-list
- Partage de tier-list

Feed:

- Affichage des tier-list
- Affichage des meilleures tier-list
(Les meilleures tier-list seront sélectionnées soit par rapport aux nombres de likes / soit aux nombres de commentaires)
- Affichage des meilleurs tier-list
(Les meilleurs tier-list seront sélectionnés par rapport aux nombres d'abonnements.)

Commentaires:

- Commenter une tier-list
- Modifier le commentaire
- Supprimer le commentaire
- Afficher un commentaire

Abonnement:

- S'abonner à un utilisateur.
- Se désabonner à un utilisateur.

Like:

- Liker une tier-list.
- Unlike une tier-list

Search Bar:

- Recherche dans les utilisateurs
- Recherche dans les styles musicales
- Recherche dans les artistes

services musique:

- API:

- TheAudioDB

style musicale: Rock / Rap / Hip-hop / Electro / Pop ...

Artistes : nom / image / biographie / discographie.

Album: nom / image / titres / année.

3.3 Contexte technique

3.3 a) Choix des langages et des frameworks

Pour réaliser ce projet, nous avons décidé d'utiliser uniquement du **Javascript** et **Node Js** comme environnement de développement. Pourquoi ce choix? Tout simplement car Javascript est un **langage riche** avec de nombreux concepts, nous permettant une **montée en compétences**. C'est un langage beaucoup utilisé par les géants du web, ces derniers créent de nombreuses bibliothèques de code **open source** facilitant ainsi le développement de certaines fonctionnalités. Il est présent dans toutes les applications web mais aussi mobiles. Il n'existe à ce jour plus aucune page web qui n'utilise pas cette technologie pour dynamiser son contenu.

ExpressJS:

Pour le développement de notre Api, nous avons choisi d'utiliser Express Js qui est un framework de NodeJs car il comporte un ensemble de paquets, pour des fonctionnalités, des outils qui aident à simplifier le développement. Express Js permet de développer, de façon rapide et efficace, une API. Ce qui correspond parfaitement à notre besoin.

Historique Express JS :

ExpressJs est le framework le plus populaire pour NodeJs. C'est un framework **minimaliste** permettant de garder un certain contrôle dans le développement du projet et apporte peu de surcouche permettant ainsi de garder des performances optimales et une exécution rapide.

React Native:

Pour la création de mon application mobile, nous avons choisi le framework react native car il est écrit en javascript et qu'il permet le développement d'un seul code pour les plateformes **iOS** et **Android**. Ce qui nous permet d'optimiser notre temps de développement par rapport à notre besoin.

Historique React Native:

React Native est un framework d'applications mobiles open source créé par Facebook. Il est utilisé pour développer des applications pour Android, iOS et UWP en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes.

3.3 c) Logiciels et autres outils

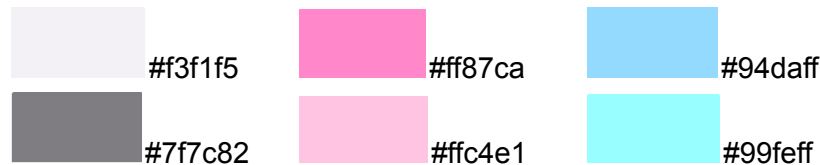
Dans le cadre de ce projet, nous avons également utilisés différents logiciels et outils:

- LucidSpark, pour la partie réflexion / brainstorming
- Trello, pour l'organisation du travail
- LucidChart, pour le maquettage de la base de données
- MongoDB Compass, pour la visualisation de notre base de données
- Figma, pour la création des maquettes
- Adobe Illustrator, pour la création du logo, du background ainsi que la charte graphique
- Yarn, pour la gestion des packages
- GitLab et ligne de commande, pour le versionning du code
- Expo, pour émuler l'application sur les téléphones
- XCode et le simulateur, pour l'environnement de développement pour iOS
- Visual Studio Code, pour éditer le code.

3.4 Maquette (Ultimate List)

3.4 a) Charte graphique et UX

Palette:



Police:

KGRedHands.ttf

Ultimate List

KGRedHandsOutline.ttf

Ultimate List

Titre : KGRedHands.ttf

ExtraLight 200

UltimateList

Light 300

UltimateList

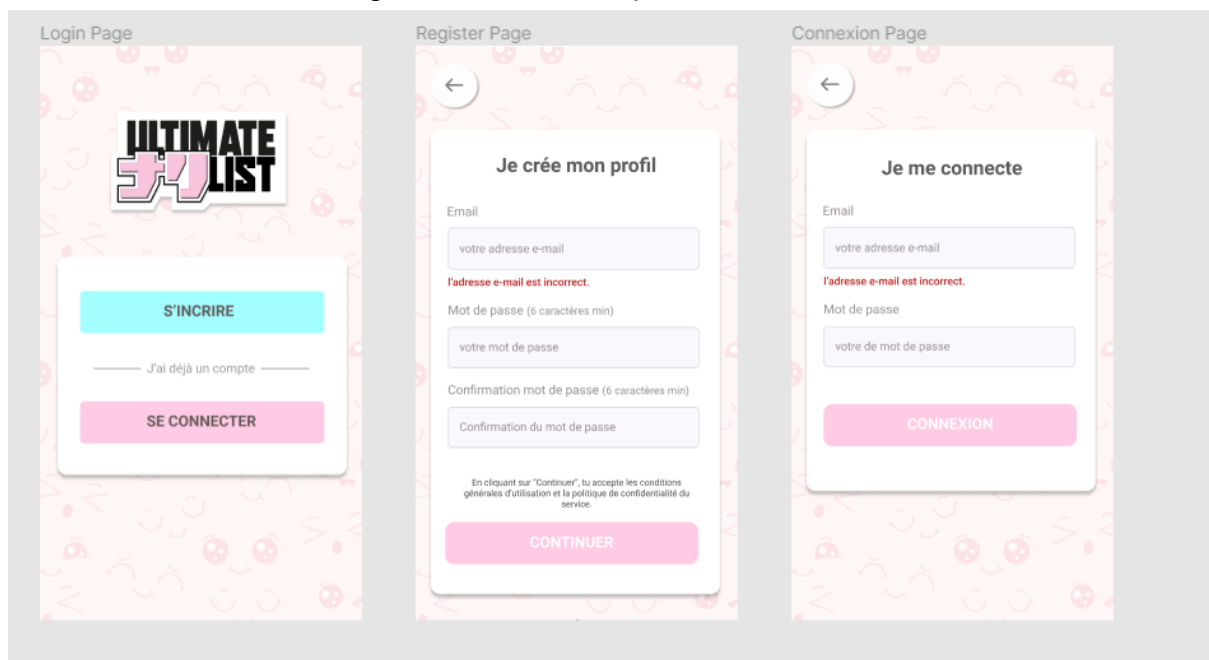
texte : Inter

Les premières minutes d'utilisation d'une application mobile sont cruciales pour assurer la pérennité d'un projet mobile dans le temps. L'expérience de l'application doit faire la différence. L'UX Design (ou User Experience Design) et les couleurs adoptées sont des couleurs enfantines et douces appliquées dans le but de faire vivre une expérience plaisante et enrichissante à l'utilisateur.

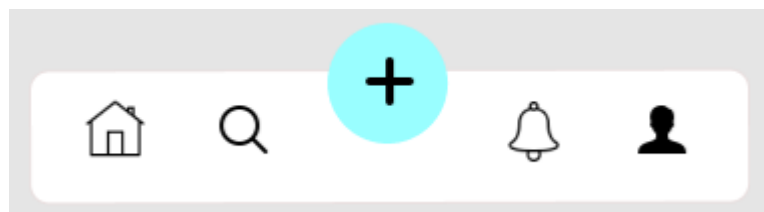
3.4 d) Wireframe et prototype

Une ambiance “**Kawaii**” sera adoptée, des **couleurs pastels** seront utilisées, ce qui pourra parfois pouvoir décrédibiliser les tiers lists les plus "sérieuses" à l'image des cabines de photo japonaises **purikura** qui peuvent parfois décrédibiliser les personnes les plus effrayantes. Nous avons décidé d'adopter ce thème car notre **public** est principalement "**millennial**". Une possibilité de changement de thème sera implanté pour pouvoir s'adapter à des utilisateurs qui ne font pas partie de cette tranche d'âge.

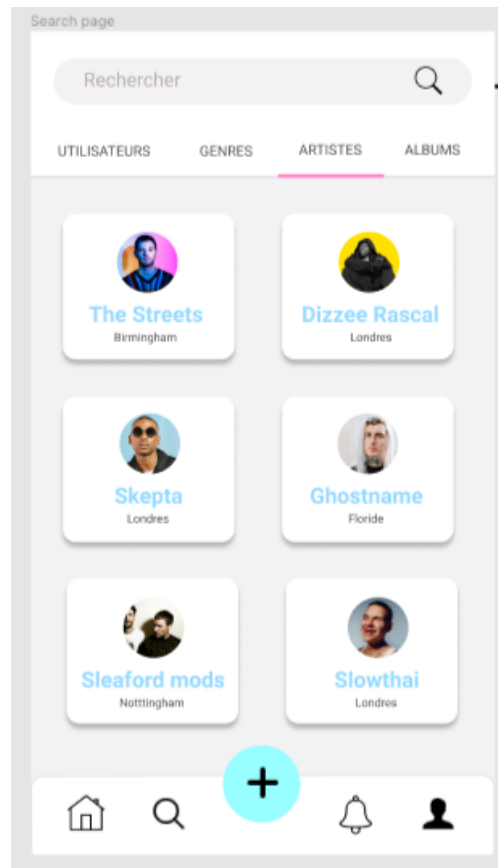
Pages Accueil / Inscription / Connexion



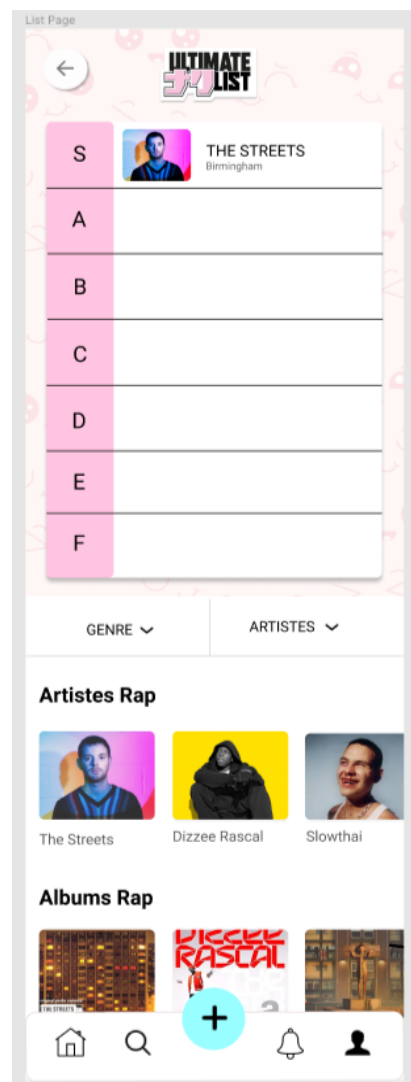
La Navbar va permettre d'aller sur la création de liste, le feed, la page de recherche, les notifications et le profil.



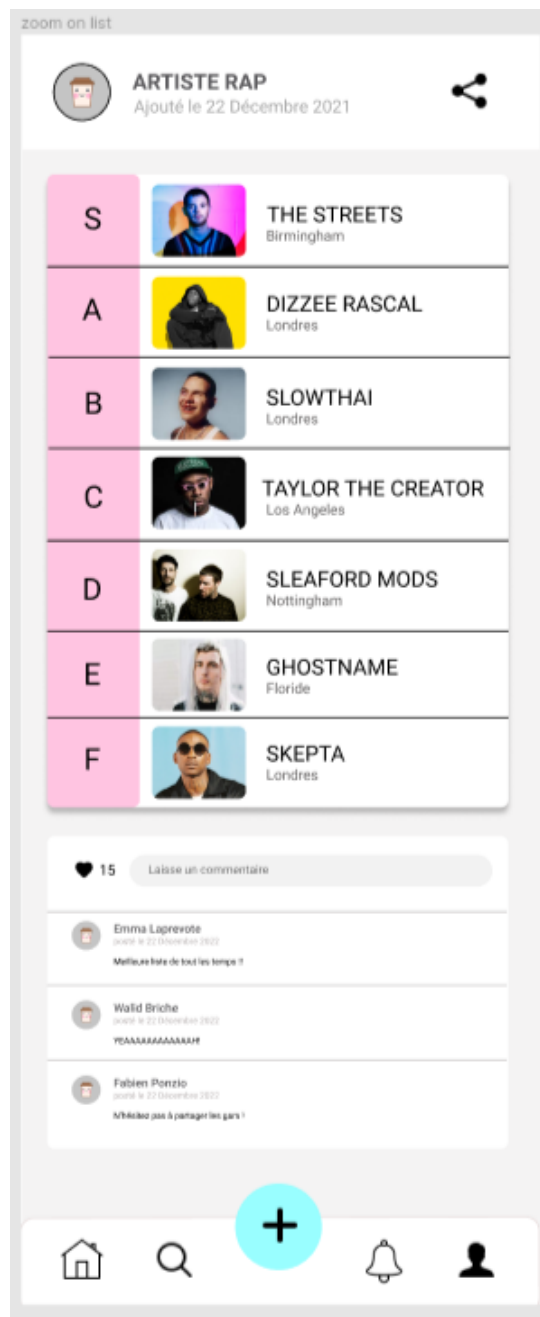
La barre de recherche en haut va permettre de rechercher par utilisateurs, par genres, par artistes et par albums.

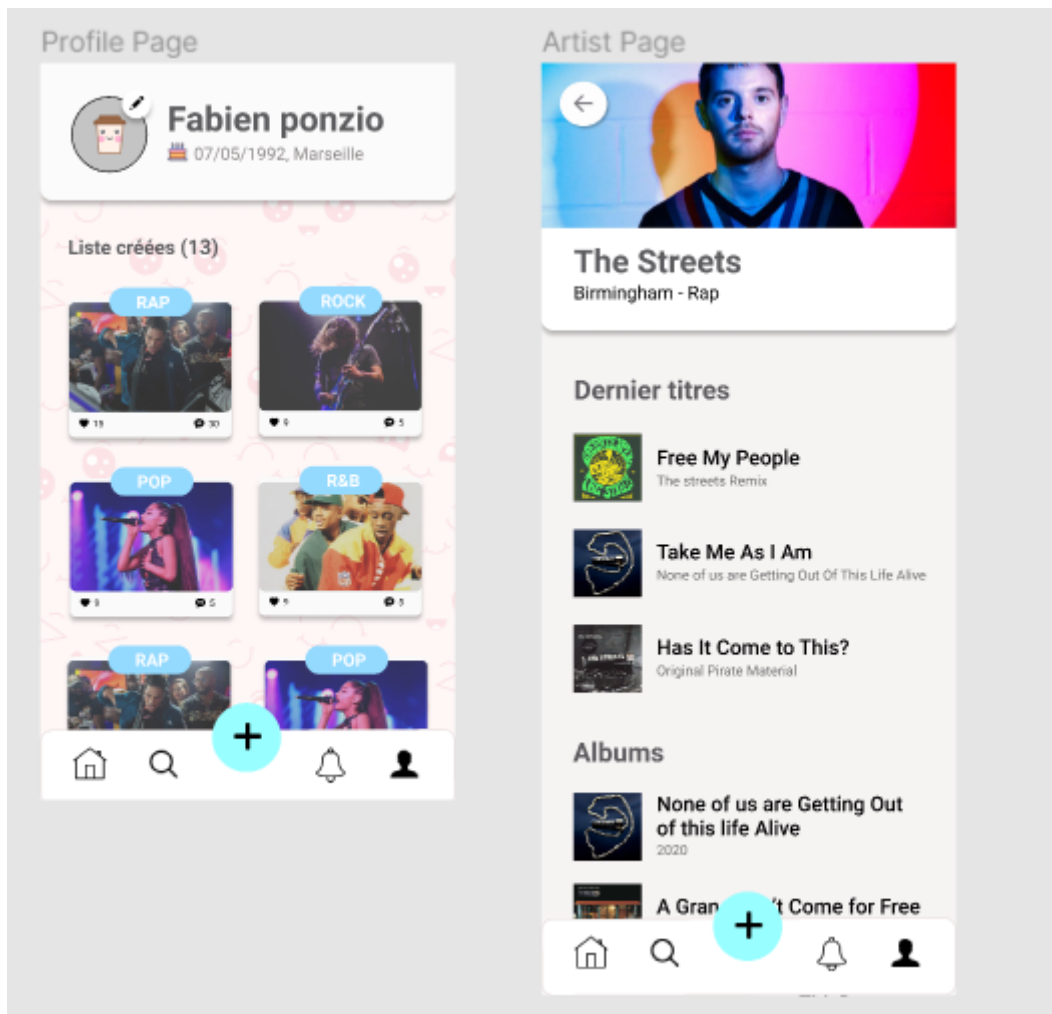


Pour créer une liste il faudra, choisir le style musical de la liste, choisir si ça sera une liste d'artistes, d'albums ou de morceaux. **L'établissement du classement se fera par glissé/déposé.**



Le bouton d'action permettra de partager la liste sur les réseaux sociaux. Il sera possible d'interagir avec les listes, via un système de like et de commentaires.





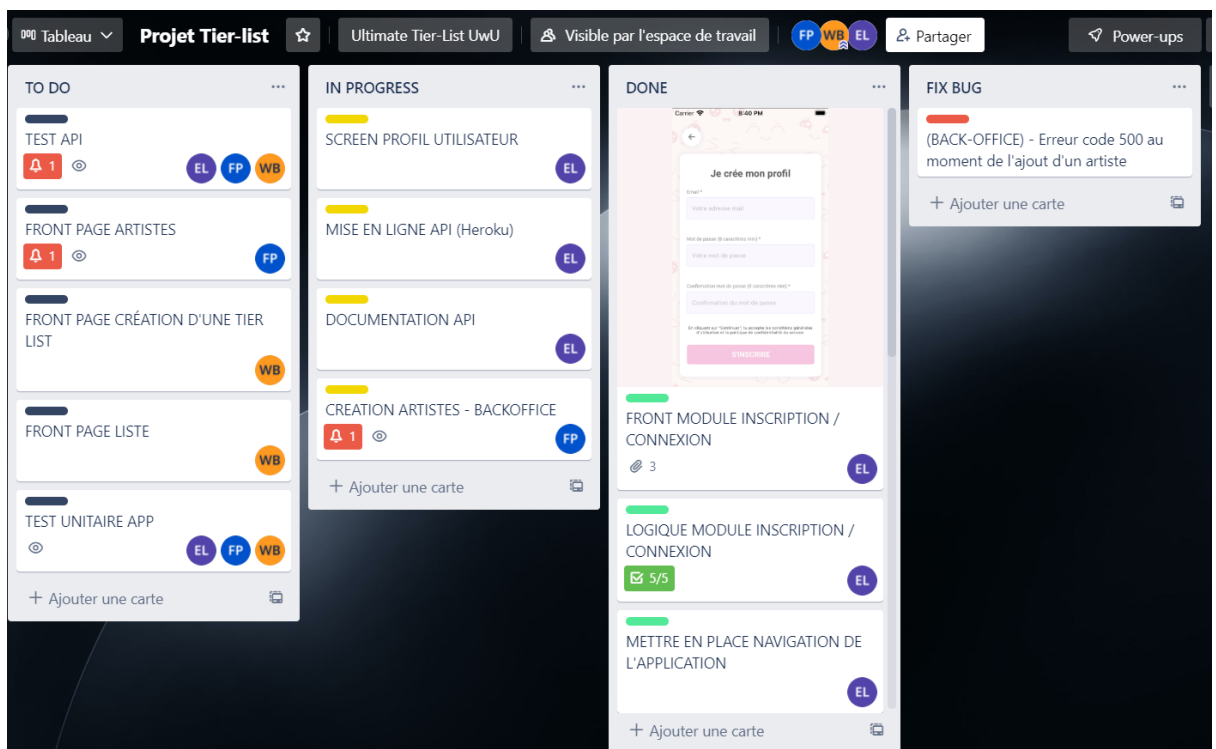
A gauche la page profil contenant les informations de l'utilisateur (nom complet, date de naissance et avatar) et les listes créées ainsi que le genre musical associé.

A droite, la page artiste contenant les informations sur l'artiste recherché ainsi que sa discographie.

3.4 e) outils

Dans le cadre de ce projet , j'ai dû utiliser d'autres outils:

- **Visual Studio Code** pour écrire mon code;
- **Postman** pour effectuer les requêtes API;
- **GitHub** et **GitLab** pour le versionning de mon code;
- **Trello** pour organiser mon travail;
- **NPM** pour installer les paquets;
- **Figma** pour la création de mes maquettes;
- **LucidChart** pour le maquettage de ma base de données;
- **Expo** pour émuler mon application mobile sur mon téléphone;
- **Color Hunt** pour la création de ma charte graphique.



Nous avons créé un **tableau de bord** sur Trello afin de lister les tâches à accomplir et suivre leur **avancement** par les différents membres de l'équipe.

Trello est une application qui permet aux utilisateurs d'avoir accès à un tableau sous forme de cartes pour gérer des projets collaboratifs. C'est un outil **intuitif** et **simple** d'utilisation.

Trello nous permet de suivre plus facilement l'ensemble du processus du projet en attribuant des **tâches** à des **collaborateurs**. Nous avons une colonne pour les tâches à réaliser, une seconde pour les tâches en cours et une dernière pour les tâches terminées.

3.4 f) difficultés rencontrées et axes d'améliorations

La difficulté principale lors de la réalisation de la maquette a été de définir un parti pris **esthétique** et **graphique** qui définisse la marque "UltimateList". Nous nous sommes basés sur une **esthétique japonaise** forte en couleur et connue dans le monde entier.

4 / Conception et persistance des données en intégrant les recommandations de sécurité

4.1 Recommandations de Sécurité

JWTToken :

JWT (JSON Web Tokens) est un standard ouvert défini dans la RFC 75191. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. C'est actuellement l'un des moyens les plus pratiques et utilisé pour sécuriser l'accès à une API. En effet, comme chaque jeton a une durée de vie. Même s'ils sont compromis, ces jetons seront considérés comme "expirés" après un certain temps. Bien sûr, la génération d'une nouvelle clé publique et privée invalide tous les jetons précédemment créés.

Ce jeton est composé de trois parties:

- **Un header** : identifie quel algorithme a été utilisé pour générer la signature
- **un payload** : est la partie qui contient les informations de l'utilisateur, sous forme de chaîne de caractères hashé en base 64. Pour des mesures de sécurité , je n'insère aucunes données sensibles telles que des mots de passe ou des informations personnellement identifiables.
- **la signature** : Elle est créée à partir du header et du payload générés et d'un secret. Une signature invalide implique systématiquement le rejet du token. La signature du jeton a une importance fondamentale , il sert à vérifier que les informations connues sont inchangées.

Lorsqu'un utilisateur essaie de se connecter à son espace , une demande est envoyée au serveur. Si les informations envoyées sont correctes , le serveur renvoie une réponse sous forme de JSON dans le quel s'y trouve le jeton . Celui-ci contient des informations concernant la personne connectée (son id , son mail et son rôle). Le client enverra ce jeton avec toutes les demandes qui suivront. Ainsi, le serveur n'aura pas à stocker d'informations sur la session.

Credential stuffing :

Le credential stuffing (ou bourrage d'identifiant) consiste à réaliser, à l'aide de logiciels ou de façon manuelle, des tentatives d'authentification massives sur des sites et services web à partir de couples identifiants/mots de passe (généralement, une adresse électronique et un mot de passe).

Elle est due à différents facteurs :

L'identifiant d'un utilisateur correspond souvent à son adresse électronique, notamment sur le web.

Les utilisateurs se servent souvent des mêmes couples identifiants et mot de passe sur plusieurs sites, sans renouveler leur mot de passe.

Des listes contenant des centaines de millions de ces couples email/mot de passe sont disponibles au sein du web caché (dark web) suite à des violations de données.

Si le credential stuffing ne permet pas forcément et facilement de cibler un compte en particulier, il permet en revanche :

de trouver des couples identifiant/mot de passe valides (un faible pourcentage de plusieurs centaines de millions de comptes représente tout de même un volume de données qui peuvent être compromises) ;

de déjouer les mesures de sécurité les plus simples. En effet, les attaquants recourent généralement à une architecture distribuée (réseaux de machines zombies ou botnets) afin de ne pas être détectés. Ils peuvent également utiliser des machines ayant la capacité d'outrepasser les mesures de sécurité basiques telles que la résolution des **CAPTCHA** les plus simples.

Ce type d'attaque se rapproche de celui par force brute (bruteforce attack) dont le taux de réussite est limité aujourd'hui.

Comment y remédier ?

Utiliser une connexion multifacteur

Les accès aux comptes depuis le portail web peuvent être sécurisés, en plus du couple identifiant et mot de passe, par l'**utilisation d'une connexion multifacteur**, par exemple par l'envoi d'un SMS contenant un code à usage unique sur le terminal mobile de l'utilisateur empêchant finalement toute connexion frauduleuse depuis le web.

Ces mesures peuvent permettre au responsable de traitement de ne pas subir de violation de données importante et d'assurer la confidentialité des données confiées par ses clients.

Les mesures mises en œuvre correctement permettent de lutter efficacement contre des attaques, parfois anciennes et parfois plus récentes. Il est important pour les équipes en charge de la sécurité des systèmes d'information d'effectuer une veille

active sur les méthodes utilisées par les attaquants et d'adapter, en fonction, leur dispositif de défense.

Instaurer des mesures de sécurité fortes pour les utilisateurs

Afin de réduire le taux de réussite de l'attaquant, il est utile de mettre en place des mesures de sécurité adaptées, par exemple :

- un **CAPTCHA** ;
- un couple identifiant et mot de passe où **l'identifiant n'est pas basé sur l'adresse courriel de l'utilisateur**.

La violation du trimestre

Le credential stuffing



1 Des listes d'identifiants et de mots de passe sont publiées, généralement suite à une **violation de données**.

2 L'attaquant **recupère ces listes**, en partant du principe que les utilisateurs se servent souvent **du même mot de passe et du même identifiant** (l'adresse courriel) pour différents services.



3 Par l'intermédiaire de « robots », l'attaquant cible des sites souvent peu sécurisés et tente une **grande quantité de connexions**.



Les sites **peu sécurisés** peuvent avoir **des difficultés à distinguer un robot d'un véritable utilisateur**.

4 L'attaquant, qui a réussi à se connecter à un compte, peut alors **changer de mot de passe** pour que l'utilisateur ne puisse plus utiliser son compte, **faire des achats** (si la carte bancaire est enregistrée), etc.



Gestion des droits :

Définir des profils d'habilitation dans les systèmes en séparant les tâches et les domaines de responsabilité, afin de limiter l'accès des utilisateurs aux seules données strictement nécessaires à l'accomplissement de leurs missions.

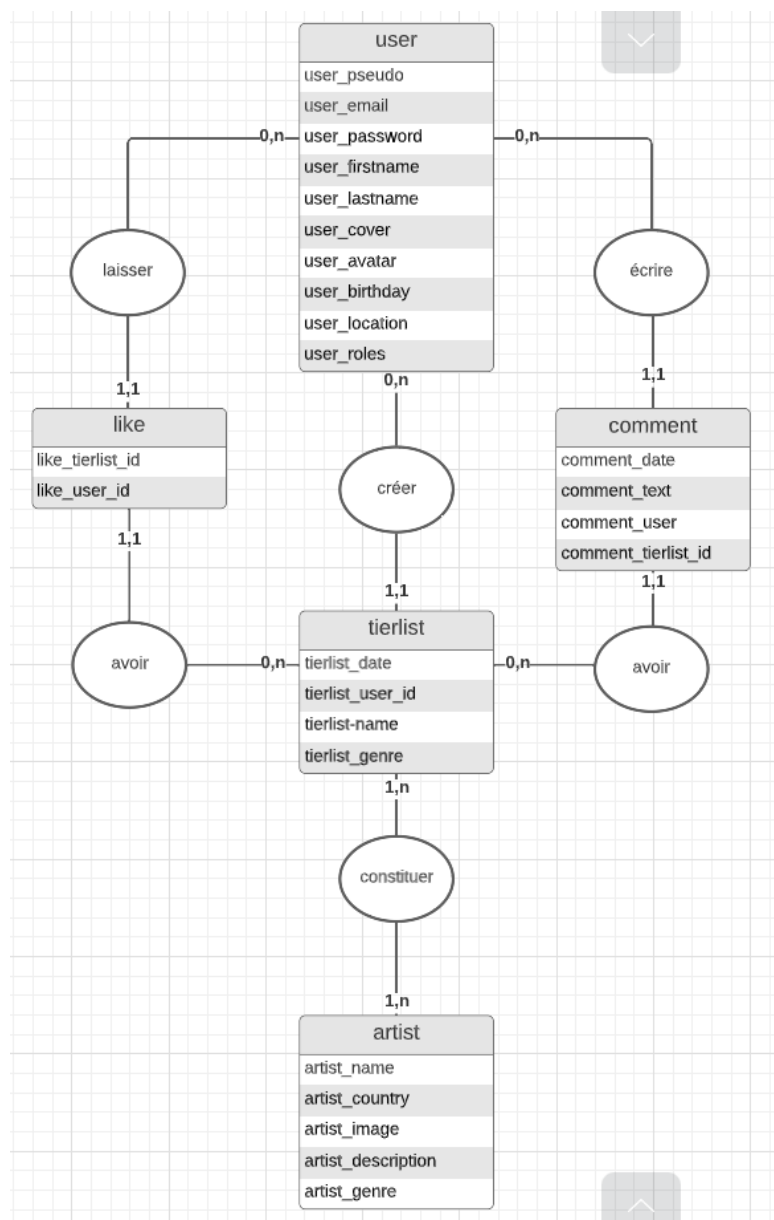
Supprimer les permissions d'accès des utilisateurs dès qu'ils ne sont plus habilités à accéder à un local ou à une ressource informatique, ainsi qu'à la fin de leur contrat.

4.2 Mise en place et création de la base de données (Ultimate List)

MongoDB a été adopté pour plus de **flexibilité** et **d'accessibilité**. En effet, lorsqu'on fait du SQL il est nécessaire de faire une base de données fiable. Par exemple si l'on veut ajouter un champ "sexe" en cours de projet dans notre table utilisateur et qu'un utilisateur est créé au moment où le champ n'existait pas alors il sera proposé à l'utilisateur de définir son sexe. Le NoSQL fonctionne en **cycle itératif**, en effet les nouveaux documents créés vont avoir aucune incidence sur les collections actuelles ou précédentes.

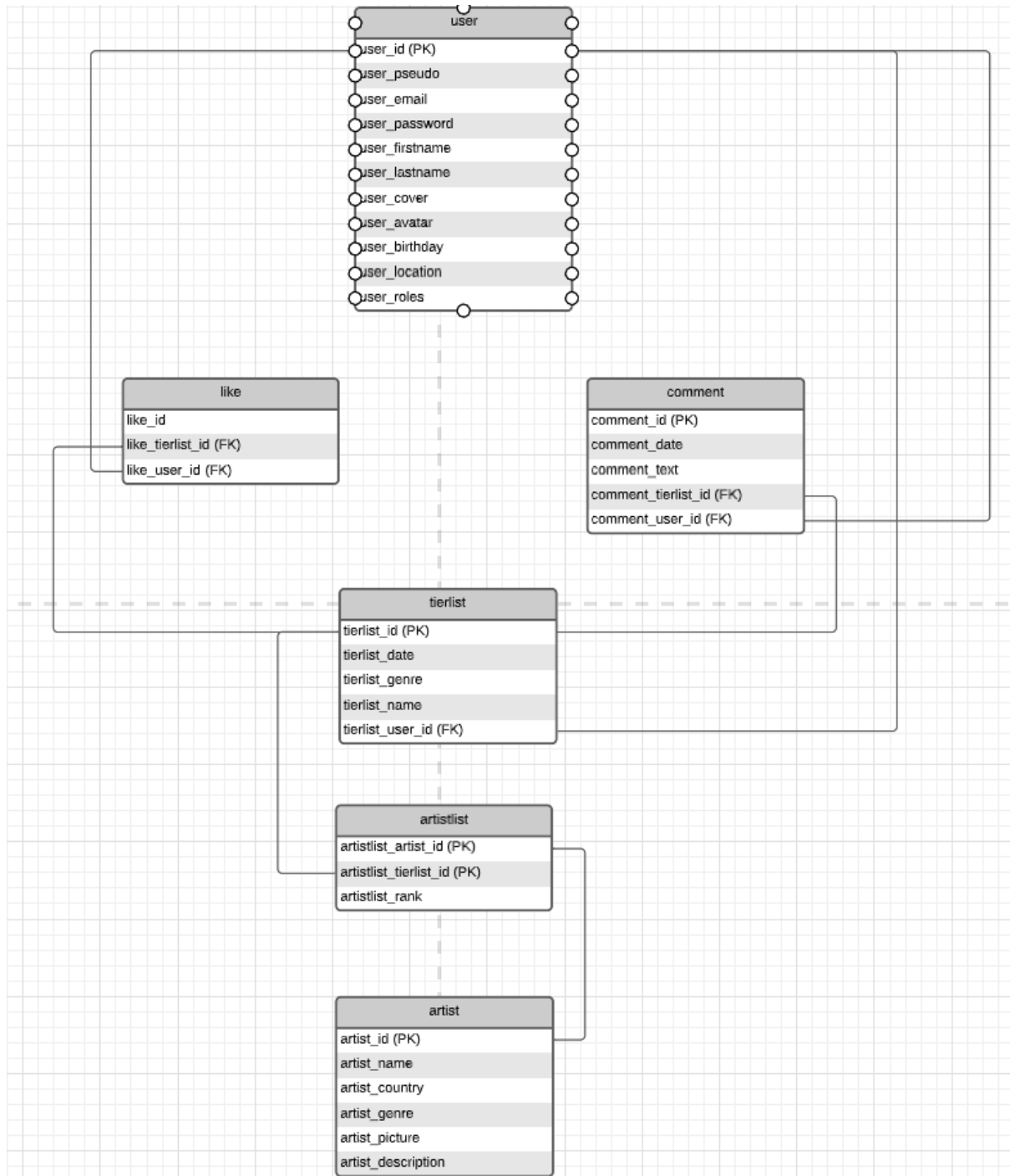
Des schémas de la méthode de Merise ont tout de même été effectués.

4.3 MCD



Nous pouvons distinguer deux grandes catégories de tables : la table user et la table tierlist. Le point d'entrée de notre base de données est la table User.

4.3 MLD



5/ Développement Back-end d'une application multi-couches en intégrant les recommandations de sécurité (Deki)

5.1 Collaboration à la gestion de projet

5.1 a) Fonctionnalités significatives - Ré-attribution de course

Dans le cadre de mon contrat d'apprentissage j'ai pu participer à la conception d'un **outil back-end**. En effet il s'agissait d'un système de ré-assignation automatique des courses si l'entreprise de livraison préalablement assignée n'a pas accepté la course dans le temps imparti.

A la fin de la création d'une course nous avons pour finalité d'attribuer une course à un dispatcher, celui-ci se voyait attribuer la course en fonction de son poids, de son lieu de livraison ou même de son client final.

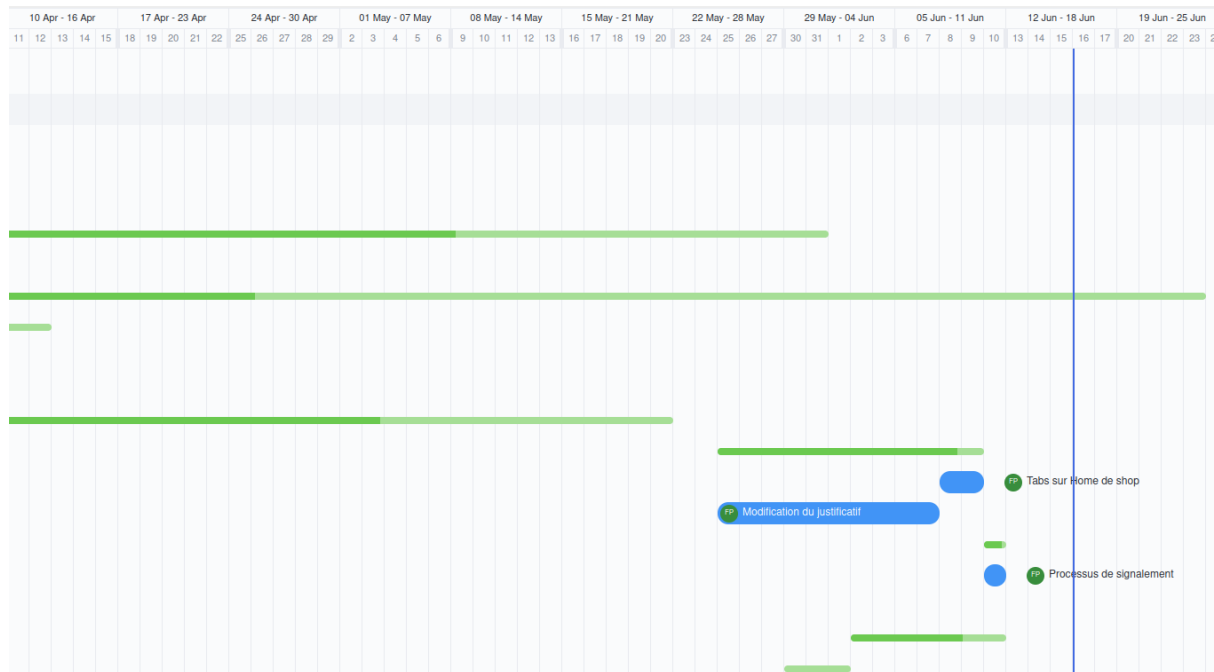
Cependant il était parfois possible que le dispatcher n'accepte pas ou ne refuse pas sa course pour différentes raisons, une **solution de repli** était dans ce cas difficilement trouvable à temps et donc d'effectuer nos livraisons dans les délais imposés.

Pour répondre à cette problématique et aux attentes du client nous avons donc décidé de mettre en place un **outil de ré-assignation automatique** des courses pour que nous puissions gagner du temps et être plus **prévoyant**.

L'outil en question est déclenché via un **appel API** qui va collecter dans un premier temps toutes les courses dont le délai imparti est écoulé, puis, pour chaque courses l'identifiant de la société de livraison qui n'a pas accepté à temps la course sera rajoutée dans une liste, afin d'éviter une nouvelle ré-attribution automatique vers cette même société.

Un mail lui est envoyé pour lui signaler que la course ne lui est plus conférée et la course repasse vers l'outil d'attribution automatique afin qu'elle soit prise en charge par une nouvelle société de livraison. Cet outil est automatisé avec un **CRON JOB** qui déclenche automatiquement un appel vers cet outil **toutes les cinq minutes**. Dans le cas ou aucune société de livraison est disponible, cela en vient à l'administrateur de l'entrepôt de marchandises de décider.

5.1 b) Organisation



Le diagramme de **Gantt** permet à toutes les personnes travaillant sur le projet d'avoir une meilleure organisation et visibilité sur le projet. Pour avancer au mieux il a été essentiel d'organiser nos tâches, il permet de suivre l'avancement du projet et de faire des ajustements en cours de route. La **planification** nous donne une meilleure vue d'ensemble.

5.1 c) Versioning et outils

Le travail de deki est stocké sur **Gitlab**. Pour ce qui est du versioning nous créons une branche par fonctionnalités. Il était habituel de push en fin de journée afin de stocker le travail effectué. Une véritable distinction est faite entre l'environnement de développement et de production pour que l'on puisse travailler sans crainte de casser le système. A chaque fin de tâche il est nécessaire d'effectuer une demande de merge, celle-ci va permettre au CTO de récupérer mon travail et l'ajouter à la branche de production et déployer les changements effectués.

5.2 Problèmes rencontrés

Les principaux problèmes rencontrés sont du au formatage de la date ainsi que de l'heure. Pour résoudre ce problème nous avons décidé d'utiliser la librairie moment.js.

```
1  const services = require("app/services/course");
2  const queryAPI = require("app/helpers/queryBuilder");
3  const mailer = require("app/services/mail");
4  const assignator = require("app/helpers/autoAssign");
5
6  module.exports.assign = async () => {
7    try {
8      const coursesWaiting = await queryAPI.coursesWaiting();
9      for (let i = 0, l = coursesWaiting.length; i < l; i++) {
10       const course = coursesWaiting[i];
11       if (course.team_id) {
12         const teamId = course.team_id;
13         await services.deny({
14           course_id: course.id,
15           team_id: course.team_id,
16         });
17         const mail = await queryAPI.getMailByTeamID(teamId);
18         mailer.sendMail([
19           {
20             type: "re-assign-task",
21             toSend: {
22               email: mail,
23               subject: "Course réattribué",
24             },
25           },
26         ]);
27       } else {
28         const merchant = await queryAPI.getMerchantByCourseId(course.id);
29         await assignator.assign([course.id], merchant);
30       }
31     }
32     return [0, { success: true }];
33   } catch (err) {
34     console.log(err);
35     throw err;
36   }
37 };
38
```

Assign

Modify
Pending events
Processed events
Invocation logs

Name ⓘ

Webhook ⓘ

Cron Schedule ⓘ

Build a cron expression

Frequently used crons

Payload ⓘ

1	{}
---	----

> Advanced

Save
Delete

Interface graphique d'**Hasura** notre **API** qui permet de visualiser notre base de données et de créer des CRON JOB. Dans cette capture d'écran c'est la route "assign" va être exécutée.

<div> <div>✓</div> <div>✗</div> </div>	Thomas Boyon	Collecte	Selectionnez un coursier	Non attribué	30/03/2022	13:34	→	13:34	-	0.1	-	-
	GAASWATT	Livraison	Selectionnez un coursier	Non attribué	31/03/2022	13:34	→	13:34	-	0.1	0.57/89.14	-

Partie visuelle de l'interface destinée au dispatcher où se trouve une livraison automatique attribuée. Un bouton d'acceptation ou de refus est mis à disposition du dispatcher lui permettant d'accepter ou de refuser cette course. Dans le cas où le dispatcher refuse, la course repasse automatiquement dans le système.

6/ Développement de la partie front-end d'une application : présentation de fonctions significatives (Deki)

6.1 Librairie Reactour

Au cours de mon apprentissage au sein de mon entreprise j'ai pu y développer plusieurs fonctionnalités **Front-End** mais la plus significative reste la mise en place de la **librairie Reactour** au sein du projet de l'entreprise.

En effet, cette fonction est très significative car elle répond à un besoin important de l'entreprise: **rendre l'outil accessible** à chaque personnes allant être destinées à utiliser l'application.

L'outil proposé à nos clients est complet mais pas forcément simple d'utilisation, au fur et à mesure que notre outil était utilisé par nos clients testeurs, beaucoup de leurs retour nous faisaient part d'une certaine incompréhension ou de beaucoup de questions, ce qui nous a amenés à la conclusion suivante : une **formation à l'outil** est nécessaire.

Reactour est la librairie qui répond à notre problématique, en effet elle permet de proposer **un guide activable** au clic en mettant des zones importantes de l'écran en surbrillance, à la manière d'un **tutoriel** afin que l'utilisateur puisse exploiter au mieux l'outil.

Dans le cadre de notre projet nous avons insisté sur des **Inputs importants** qui font partie du cycle d'utilisation de notre outil.

Installation:

Voici la commande d'initialisation de la librairie Reactour à insérer dans le terminal :

```
npm i -S @reactour/tour @emotion/react  
# or  
yarn add @reactour/tour @emotion/react
```

```
ReactDOM.render(  
  <ApolloProvider client={client}>  
    <Provider store={store} context={ReactReduxContext}>  
      <Fragment>  
        <GlobalLoader>  
          <Tour>  
            <Router context={ReactReduxContext}/>  
          </Tour>  
        </GlobalLoader>  
      </Fragment>  
    </Provider>  
  </ApolloProvider>,  
  document.getElementById('root')  
) ;
```

Intégration de Reactour via `<Tour></Tour>` dans le DOM de l'application afin que toute l'application puisse utiliser et accéder à Reactour et ses différents composants.

```

export const stepsHome = [
  {
    selector: '[data-tour="step-1"]',
    content: <p className='step-text'>Cliquez ici pour commencer la création d'une nouvelle course</p>,
  },
  {
    selector: '[data-tour="step-2"]',
    content: <p className='step-text'>Triez vos courses par statut ici </p>,
  },
  {
    selector: '[data-tour="step-3"]',
    content: <p className='step-text'>Vous pouvez aussi trier vos courses sur une période donnée</p>,
  },
  {
    selector: '[data-tour="step-4"]',
    content: <p className='step-text'>Recherchez vos courses en fonction des clients</p>,
  },
  {
    selector: '[data-tour="step-5"]',
    content: <p className='step-text'>Vous trouverez ici le détail de toutes vos livraisons</p>,
  },
  {
    selector: '[data-tour="step-6"]',
    content:
      <ul className='icon-list'>
        <li>
          <p className='step-text'><QrcodeOutlined className='pop-icons'></QrcodeOutlined>: Afficher QR Code</p>
        </li>
        <li>
          <p className='step-text'><MailOutlined className='pop-icons'></MailOutlined>: Recevoir le QR Code par mail</p>
        </li>
        <li>
          <p className='step-text'><DownloadOutlined className='pop-icons'></DownloadOutlined>: Imprimer le QR Code</p>
        </li>
        <li>
          <p className='step-text'><PrinterOutlined className='pop-icons'></PrinterOutlined>: Imprimer l'étiquette</p>
        </li>
        <li>
          <p className='step-text'><FileDoneOutlined className='pop-icons'></FileDoneOutlined>: Afficher la signature</p>
        </li>
      </ul>
  },
  {
    selector: '[data-tour="step-7"]',
    content: <p className='step-text'>Vous pouvez nous contacter en cliquant ici!</p>,
  },
];

```

Les étapes sont toutes recueillies dans un même fichier, lui-même divisé en **tableaux** qui correspondent aux différentes pages où les visites guidées vont être faites.

Fonction “displaySteps” qui permet d’afficher les “steps” correspondants à la page sur laquelle nous nous trouvons avec l’aide d’un “switch” qui est une fonction javascript permettant de faire appel à des cas différents.

```
export function displaySteps(routerPath, activeCourse) {
  switch(routerPath){
    case '/': return stepsHome;
    case '/courseMerchant':
      if (activeCourse === 0) {
        return collectSteps
      } else if (activeCourse === 1) {
        return deliverySteps
      } else {
        return recapSteps
      }
    case '/crmShop': return stepsCrm;
    default: return null;
  }
}
```

Création du **bouton** permettant l'**activation** du tour grâce à la propriété Javascript onClick. en cliquant sur l'image la fonction displaySteps va être appelée et afficher les bonnes étapes en fonction du router.location.pathname.

```
import { useTour } from '@reactour/tour';
import { displaySteps } from '../../utils/file-tools';
import { stepsHome, stepsCrm, stepsCourseMerchant } from
  '../../utils/steps';

return <Tooltip placement="right" title="Suivez le guide">
  <TourImage
    height='37'
    width='37'
    fill='#fff'

    onClick={() => {
      const test = displaySteps(store.getState().router.location.pathname, activeCourse);
      if (test !== null) {
        setSteps(displaySteps(store.getState().router.location.pathname, activeCourse))
        setIsOpen(true)
        setCurrentStep(0)
      }
    }}
    className="tour-image"
  >
  </TourImage>
</Tooltip>
```



Suivez le guide

En appuyant sur ce bouton qui figure dans toutes les pages, nous allons déclencher une visite guidée de la page sur laquelle nous sommes.

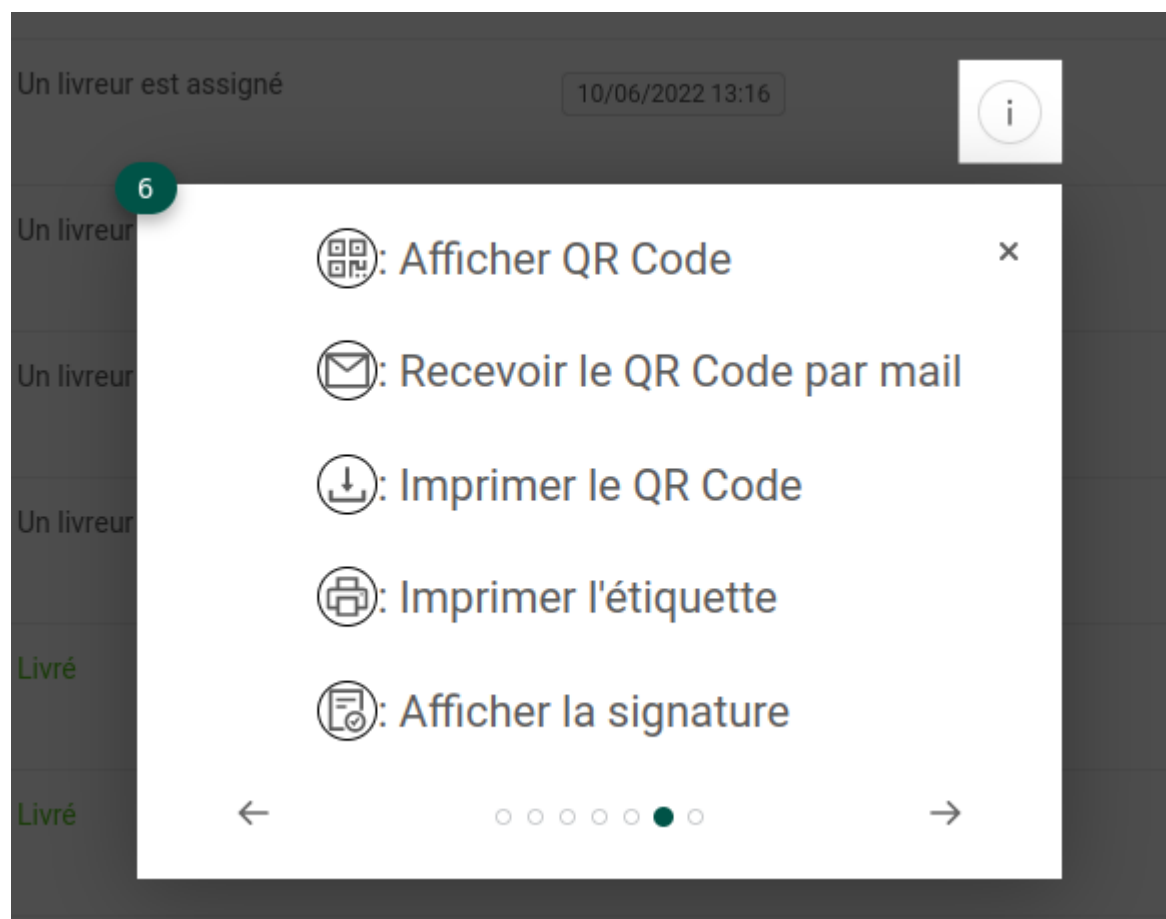
Ici on explique au dispatcher qu'il va falloir cliquer sur ce bouton pour créer une course :

The screenshot shows the 'Livraison à domicile by DEKI' interface. At the top, there's a header with 'Accueil', 'Livraison à domicile by DEKI', and 'Bonjour Thomas Boyon'. Below the header, there's a section titled 'Toutes vos courses'. A yellow button labeled '+ Nouvelle course' is highlighted. A modal window is open, displaying the text: 'Cliquez ici pour commencer la création d'une nouvelle course'. The modal has a close button (X) and navigation arrows. Below the modal, there's a table with columns: 'Devis', 'Date de livraison prévue', 'Colis', 'Statut', 'Dernière M.A.J', 'Informations', and 'Contacter DEKI'. The table contains several rows of course data.

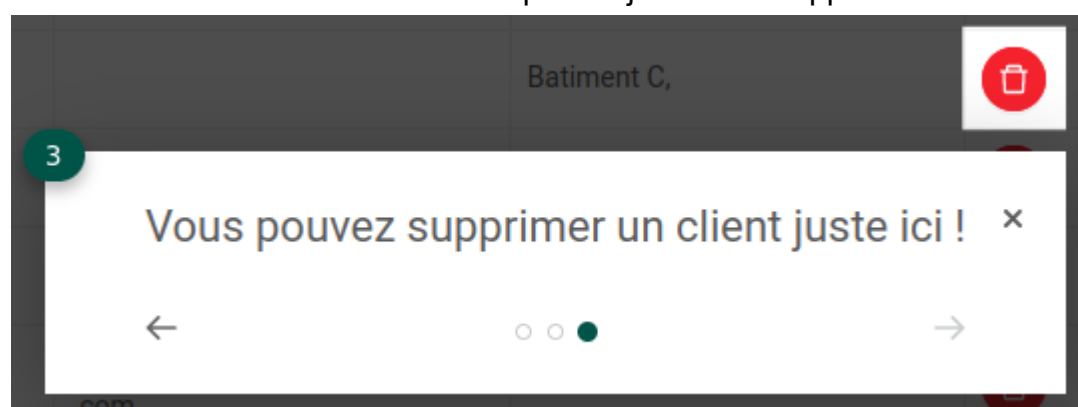
Ici on explique où se trouve tous les clients du dispatcher :

The screenshot shows the 'Livraison à domicile by DEKI' interface. At the top, there's a header with 'Livraison à domicile by DEKI' and 'Bonjour Thomas Boyon'. Below the header, there's a section titled 'Tous les clients'. A modal window is open, displaying the text: 'En cliquant sur une ligne vous pourrez modifier les informations d'un client'. The modal has a close button (X) and navigation arrows. Below the modal, there's a table with columns: 'Société', 'Prénom - Nom', 'Téléphone', 'Colis par défaut', 'Adresse', 'Email', 'Notes', and 'Action'. The table contains several rows of client data.

Ici on explique toutes les opérations possibles contenues dans le bouton information



On met ici en évidence l'une des étapes majeures : la suppression de client.



6.2 Problèmes rencontrés

Les principales difficultés rencontrées ont principalement été au niveau de l'organisation en amont afin de proposer un rendu propre auprès de mon responsable. Il était important tout au long de cette tâche de rester organisé en définissant sur chaque pages les étapes à implémenter et sur quelles parties de l'écran. Me familiariser avec la librairie afin de l'adapter au mieux au projet fait partie de mes problèmes rencontrés.

Conclusion :

Pour conclure , ce projet m'a permis de découvrir de nouveaux frameworks et aussi de pouvoir monter en compétence sur javascript.

Ce projet qui s'est déroulé sur l'année de formation m'a appris à organiser mon travail.

En effet , durant cette année je n'ai pas seulement développé ce projet , j'ai aussi collaboré sur le développement d'une autre application mobile avec des personnes de ma promotion. Une application sur la gestion des humeurs développée avec symfony pour l'API et react Native pour l'application.

Celle-ci m'a permis de développer des compétences transverses comme le travail d'équipe.

De plus, le projet présenté dans ce dossier m'a permis de voir que je pouvais m'adapter aux différentes situations.