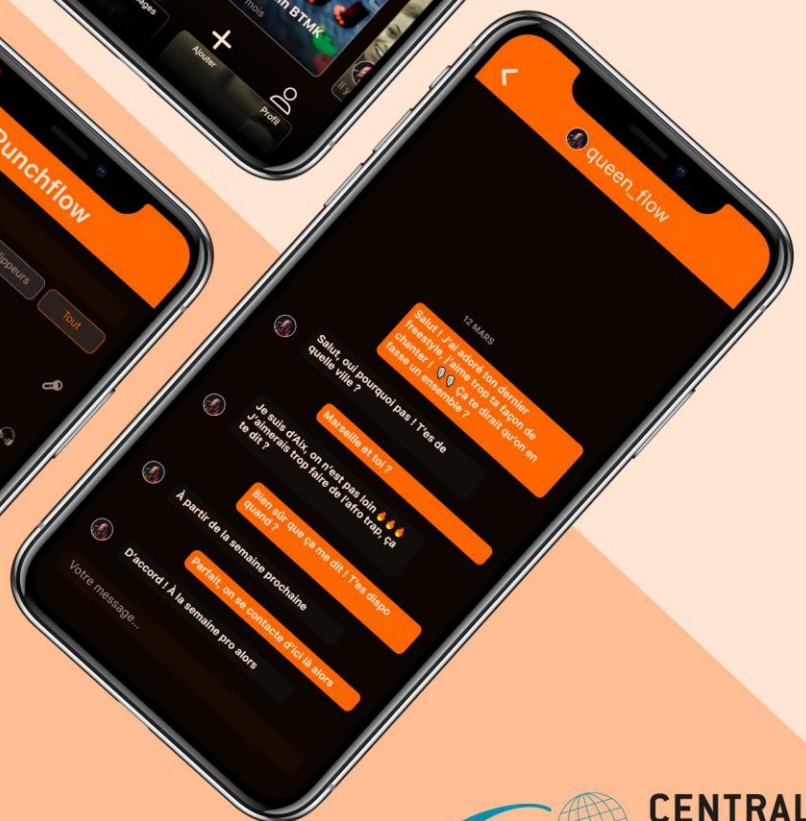
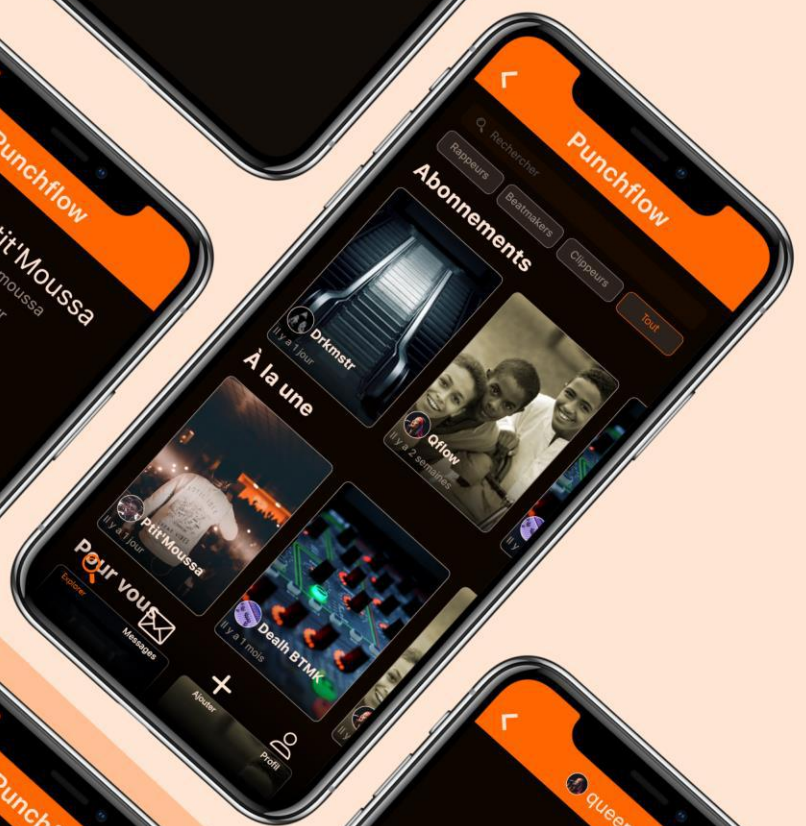
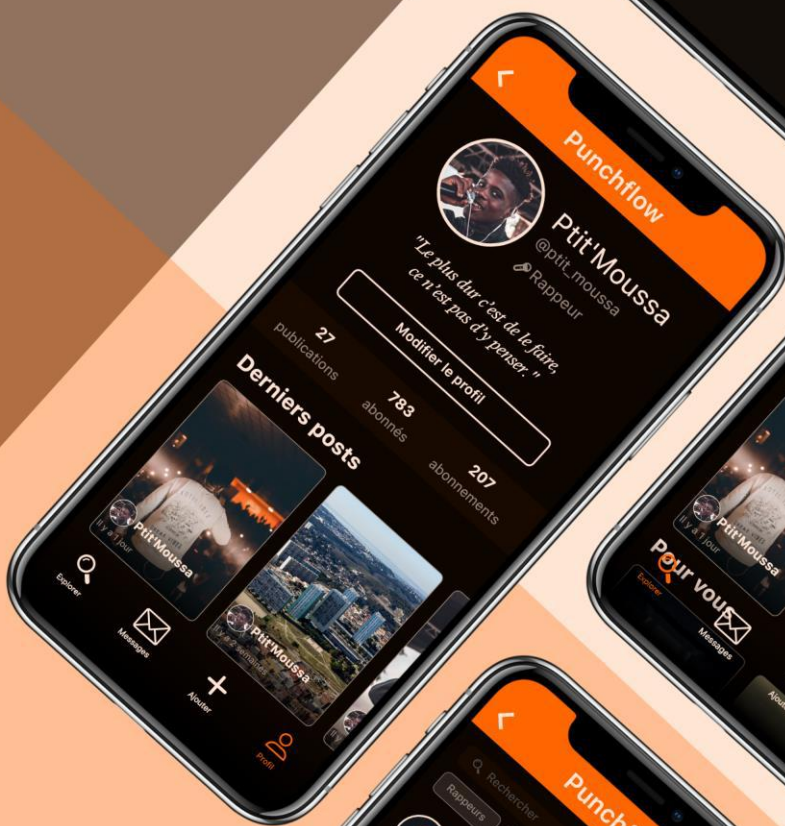


Punchflow

Dossier de présentation



Présentation du projet

Product Vision

Punchflow est un réseau social permettant à ses utilisateurs, à savoir des amateurs de rap (rappeurs, beatmakers ou clippeurs) de partager leurs travaux, ainsi que de commenter ceux des autres. Cette plateforme peut aussi permettre à ses utilisateurs, grâce à un fil d'actualité, de se faire repérer par des professionnels du domaine en recherche de nouveaux talents. Malgré les objectifs et buts vus précédemment, cette application s'efforce d'avoir des fonctionnalités simples telles que l'ajout de publications et de commentaires à celles-ci, une messagerie ainsi que le fait de s'abonner à un profil, ceci pour que le processus de mise en relation se fasse naturellement.

Poster de l'application



Product Backlog

Id	Story	Priorité
1	En tant qu'utilisateur, lorsque j'arrive sur l'application, je veux avoir le choix de me connecter ou de me créer un compte. Je pourrais donc me connecter sur l'application.	Must
2	En tant qu'utilisateur, après m'être connecté ou avoir créé un compte, je veux arriver sur l'écran explorer. Je pourrais donc voir les publications à la une, celles des utilisateurs auxquels je suis abonné, et les publications qui me sont proposés, en fonction de mes abonnements.	Must
3	En tant qu'utilisateur, lorsque je suis sur l'écran explorer, je veux pouvoir rechercher un profil avec son nom, grâce à une barre de recherche. Je pourrais donc trouver un profil grâce à son nom.	Must
4	En tant qu'utilisateur, lorsque je suis sur l'écran explorer, je veux pouvoir filtrer les publications en fonction des types de profils. Je pourrais donc voir les publications d'un ou de plusieurs types de profil.	Should
5	En tant qu'utilisateur, lorsque je suis sur l'application, je veux avoir le choix entre l'écran explorer, l'écran avec mes conversations, l'écran pour ajouter une publication et l'écran avec mon profil, grâce à une barre de navigation se trouvant sur le bas de l'écran.	Must
6	En tant qu'utilisateur, lorsque je suis sur l'application, et que j'appuie sur le bouton messages sur la barre de navigation, je veux arriver sur un écran avec une liste de mes conversations, et lorsque j'appuie sur l'une d'entre elles, je veux pouvoir voir, envoyer et recevoir des messages, je pourrais donc discuter avec un autre utilisateur.	Must
7	En tant qu'utilisateur, lorsque je suis sur l'écran des messages, je veux pouvoir faire une recherche pour trouver un profil avec lequel je suis en conversation, grâce à une barre de recherche. Je pourrais donc retrouver rapidement une conversation.	Should
8	En tant qu'utilisateur, lorsque je suis sur l'écran des messages, je veux pouvoir filtrer les utilisateurs avec lesquels je suis en conversation en fonction de leur type de profil.	Could
9	En tant qu'utilisateur, lorsque je suis sur l'application, et que j'appuie sur le bouton ajouter (une publication), je veux arriver sur l'écran de publication. Je pourrais donc ajouter une publication sur l'application.	Must
10	En tant qu'utilisateur, lorsque je suis sur l'application, et que j'appuie sur le bouton profil, je veux accéder à mon profil. Je pourrais donc voir mes publications ainsi que mes informations de profil.	Must
11	En tant qu'utilisateur, lorsque j'appuie sur l'image d'une publication, je veux pouvoir accéder à un écran avec le contenu de celle-ci. Je pourrais donc voir l'image, lire le texte correspondant, aimer et commenter la publication.	Must
12	En tant qu'utilisateur, lorsque j'appuie sur l'image de profil d'un autre utilisateur, je veux pouvoir accéder à son profil. Je pourrais donc voir les publications, m'abonner et/ou envoyer un message à cet utilisateur.	Must

Processus de Design-Thinking

Solution-Agnostic

Pour aboutir à l'idée de ce réseau social, j'ai eu d'abord 3 idées solution-agnostic Ce sont les suivantes :

- J'ai remarqué que les associations humanitaires ont du mal à récolter de l'argent
- J'ai remarqué que les monteurs vidéo amateurs ont du mal à se faire connaître
- J'ai remarqué que les jeunes chanteurs ont du mal à agrandir leur public

Ensuite, étant plus intéressé par les 2 dernières idées, j'ai décidé de les assembler tout en ciblant le monde du rap. Ceci a donné : « J'ai remarqué que les amateurs de rap ont du mal à travailler ensemble ».

Après reformulation, le problème fut affiné en plusieurs sous problèmes, et l'un d'eux fut dérivé en problème/solution, le résultat est donc devenu : « Grâce à une plateforme, les amateurs de rap qui ont du mal à travailler ensemble peuvent par exemple publier leurs travaux. ». En voici un résumé en image :

1) Idéation et votes



2) Reformulation



3) Affinement du problème

Les problèmes qu'ils rencontrent sont :



4) Résultat

Ajout du bénéfice + reformulation problème/solution :



Analyse pré-mortem



La plateforme pourrait ne pas marcher...

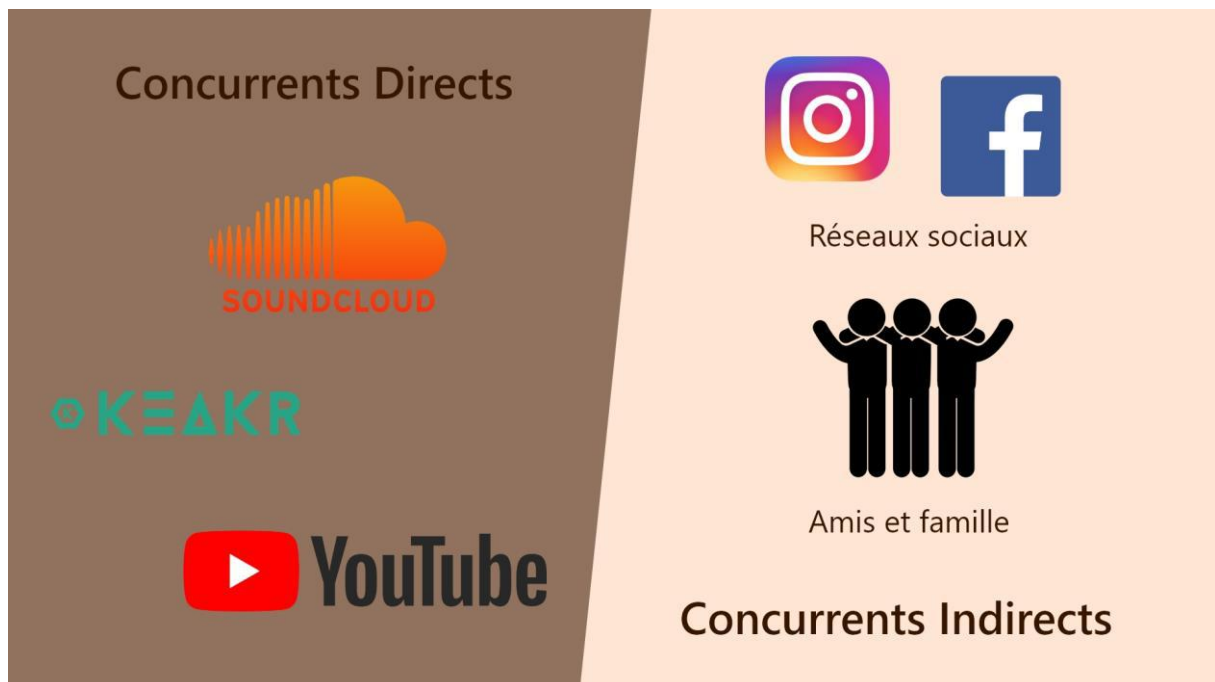
Cependant je n'ai pas trouvé de raison interne au réseau social lui-même.

Les seules raisons que j'ai trouvées et pour lesquelles le réseau pourrait ne pas survivre, sont les concurrents.

Il s'agit justement de la prochaine partie de ce dossier : l'analyse concurrentielle.

Analyse concurrentielle

Cette plateforme pourrait avoir plusieurs concurrents. Les concurrents directs sont les autres plateformes qui peuvent avoir les mêmes buts/utilisations que l'application et les concurrents indirects seraient plutôt le bouche à oreille, ainsi que les réseaux sociaux.



Persona

Voici un exemple de cible pour cette plateforme :



DAVID, 21 ans
Étudiant et beatmaker amateur
Aix-en-Provence, France

" J'adore faire mes propres productions musicales ! Ça serait cool de travailler avec des rappeurs et de faire des projets ensemble. "

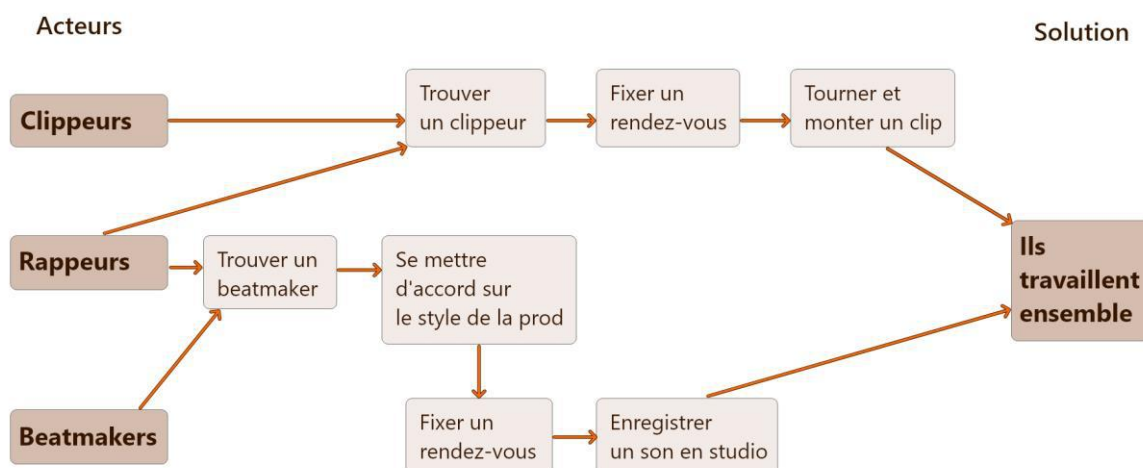
À propos
David est étudiant et fait des instrumentales durant son temps libre, dans sa chambre. Il a beaucoup de talent mais malheureusement son travail n'est pas reconnu de tous. Il ne veut pas forcément devenir célèbre mais veut quand même que ses productions soient utilisées par des rappeurs.

Comportements	Buts et besoins
Fais des productions musicales pendant son temps libre Ecoute beaucoup pour s'inspirer	Partager son travail à tout le monde Collaborer avec des rappeurs

Points de douleurs

Les principaux points de douleurs sont de trouver les bonnes personnes avec lesquelles travailler ensemble selon les styles et envies, et convenir d'un rendez-vous. Voici le diagramme de solution :

Diagramme de solution



Maquettes

Le design de cette application est découpé en 3 étapes : Le zoning, une maquette basse fidélité pour finir avec une maquette haute-fidélité. Voici un exemple du processus de maquettage de la page d'un profil utilisateur pour l'application :



Zoning

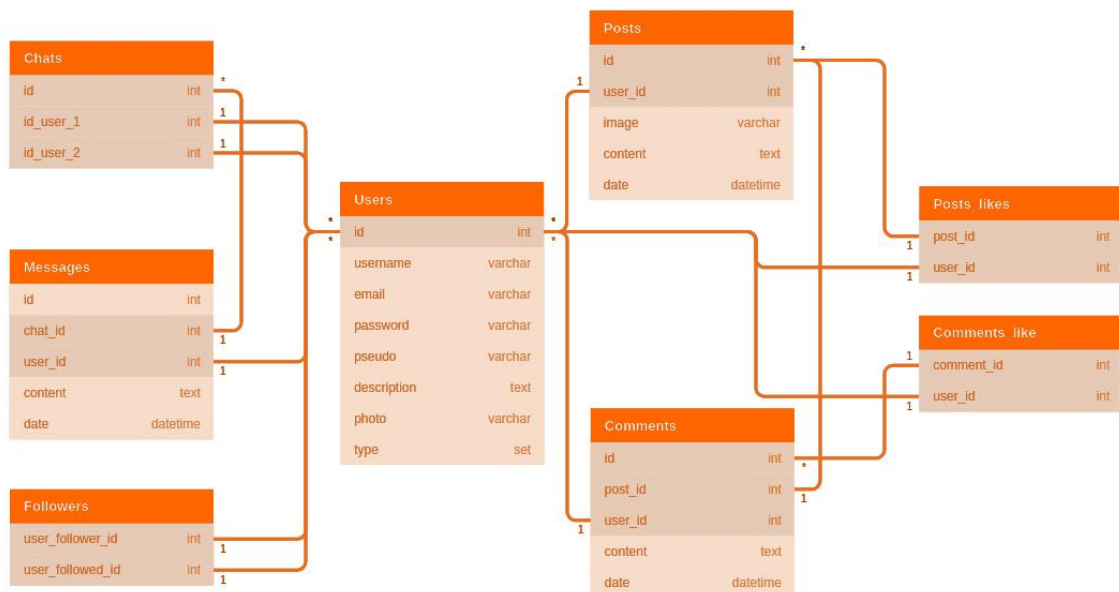


Basse fidélité



Haute-fidélité

Base de données



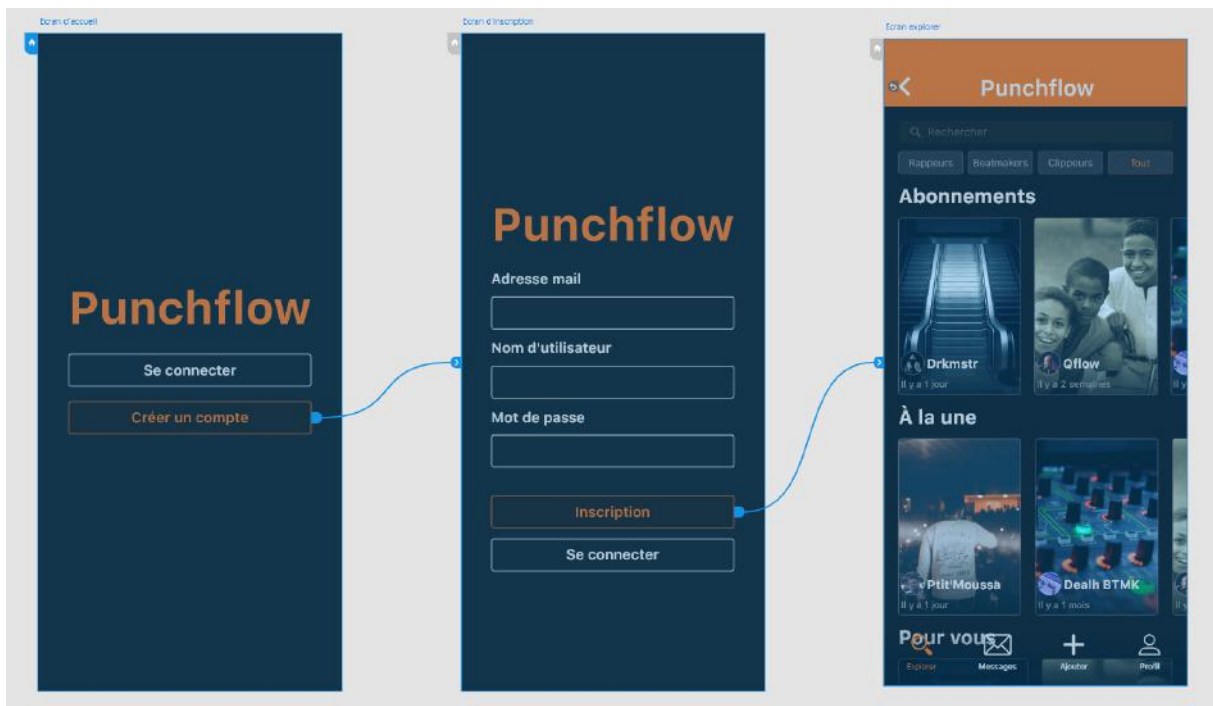
Présentation de fonctionnalités

1- Inscription

a) Description du Userflow

Étape	Action	Écran
1	L'utilisateur arrive pour la première fois sur l'application. L'utilisateur tape sur le bouton "Se créer un compte".	Page de choix entre connexion et inscription
2	La page d'inscription s'affiche, l'utilisateur la remplit et appuie sur le bouton "Inscription".	Page d'inscription
3	L'utilisateur est maintenant connecté avec son nouveau compte, il arrive sur la page "Explorer".	Page explorer

b) Userflow



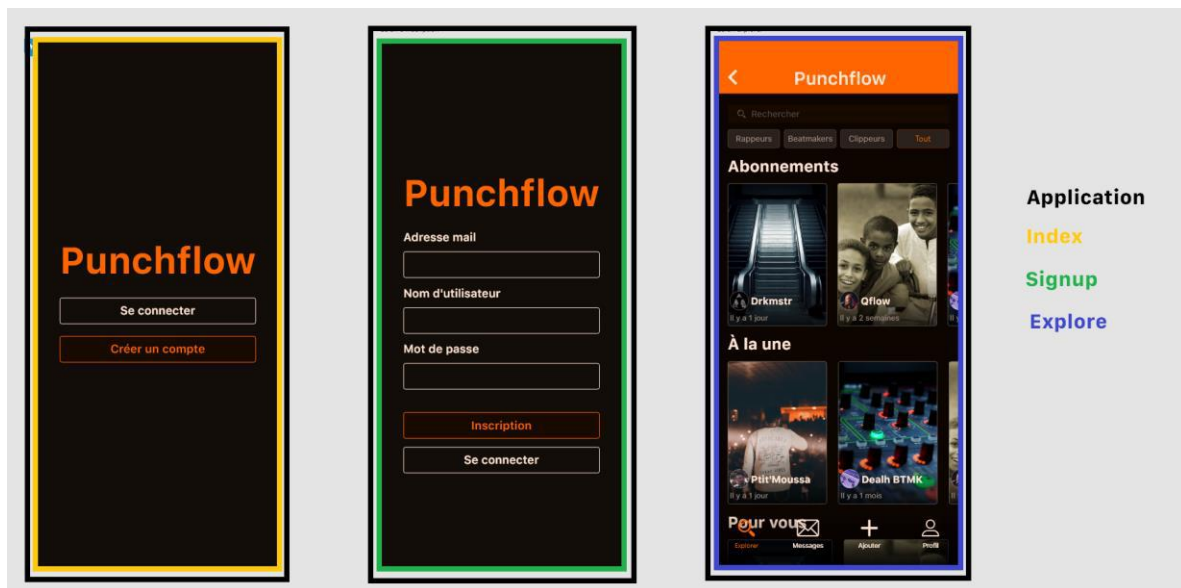
c) Kanban

À faire	En cours	Terminé
Back-end - Dans l'API, créer la route pour insérer un utilisateur		
Front-end - Créer le composant pour l'écran d'inscription		
Front-end - Créer le composant pour l'écran explorer		
Front-end - Faire la fonctionnalité avec la requête HTTP pour insérer l'utilisateur créé en BDD		

d) Components Angular

Component	Description
Écran d'accueil	Component avec les boutons et les boutons s'inscrire / se connecter
Écran d'inscription	Component avec le formulaire d'inscription
Écran explorer	Component avec les publications à la une, les publications des abonnements et les publications "pour nous"

e) Userflow components Angular



f) Requêtes HTTP

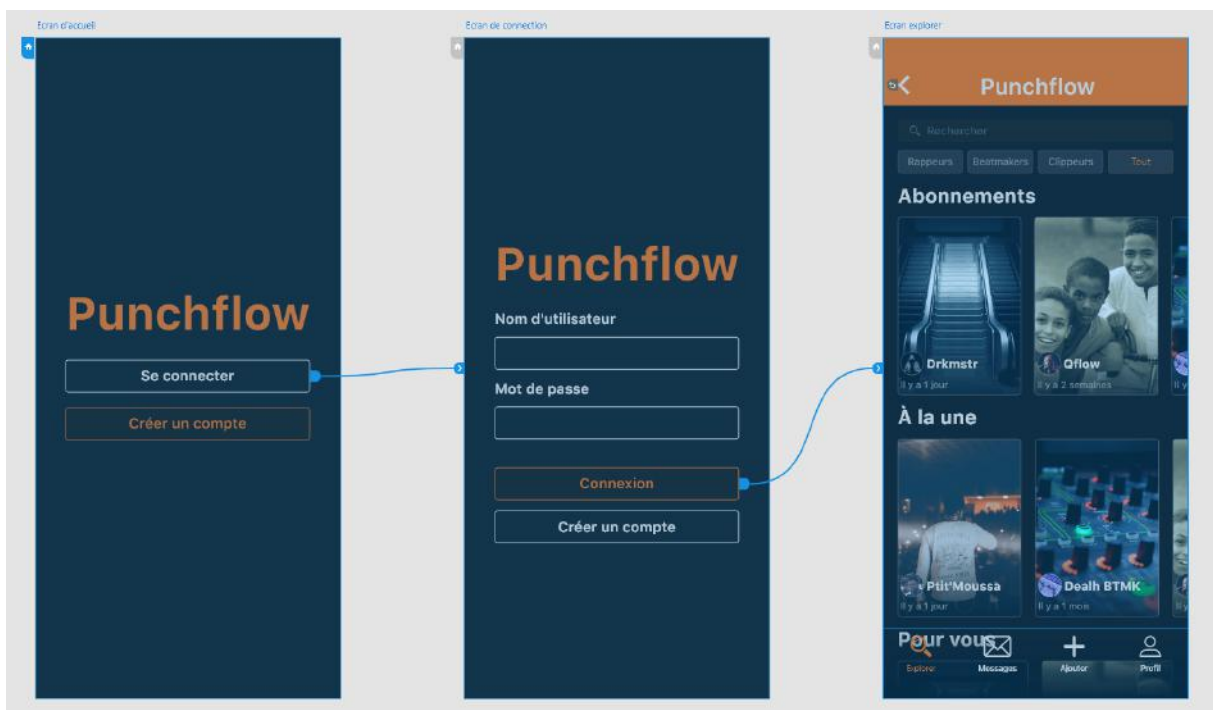
Route	/users/signup
Méthode HTTP	POST
Paramètres requis	username, email, password
Données retournées par le serveur	Pas de données retournées
Code de succès	201
Code d'échec	500
Requête MySQL	'INSERT INTO Users (username, email, password) VALUES (\${user.username}, \${user.email}, \${user.password})'
Description	Cette requête insert un nouvel utilisateur en base de données.

2- Connexion

a) Description du Userflow

Étape	Action	Écran
1	L'utilisateur est inscrit sur l'application	Page de choix entre connexion et inscription
2	La page d'inscription s'affiche, l'utilisateur la remplit et appuie sur le bouton "Inscription".	Page d'inscription
3	L'utilisateur est maintenant connecté avec son nouveau compte, il arrive sur la page "Explorer".	Page explorer

b) Userflow



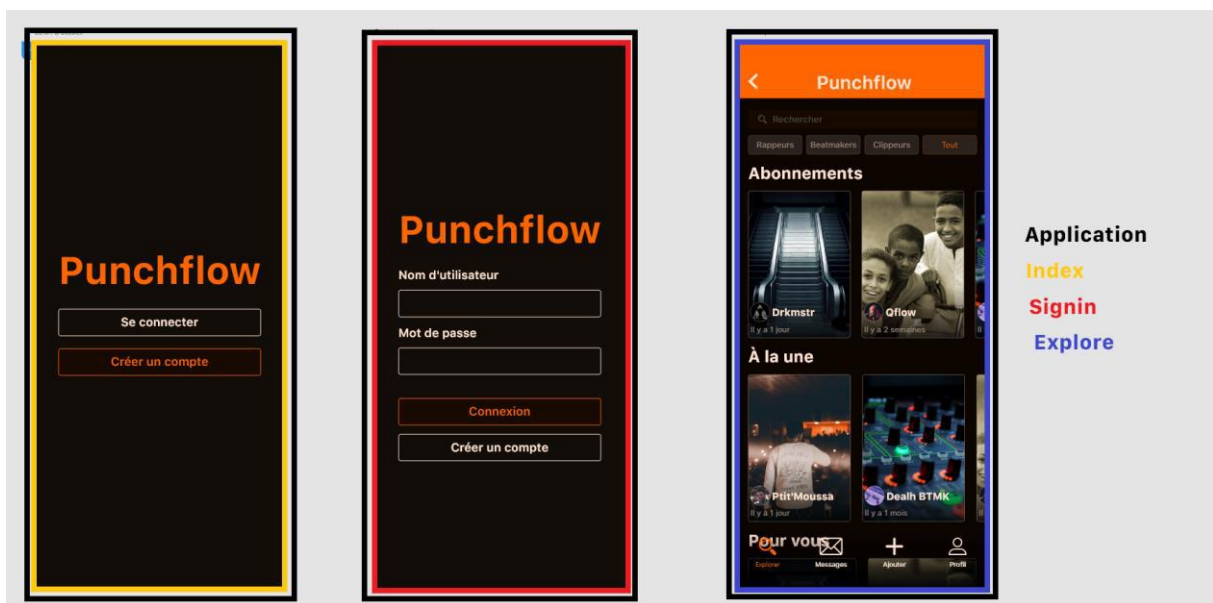
c) Kanban

À faire	En cours	Terminé
Back-end - Dans l'API, créer la route pour récupérer un utilisateur		
Front-end – Créer le composant pour l'écran de connexion		
Front-end - Faire la fonctionnalité avec la requête HTTP pour récupérer l'utilisateur en BDD		

d) Components Angular

Component	Description
Écran d'accueil	Component avec les boutons pour s'inscrire/ se connecter
Écran de connexion	Component avec le formulaire de connexion
Écran explorer	Component avec les publications à la une, les publications des abonnements et les publications "pour nous"

e) Userflow components Angular



f) Requêtes HTTP

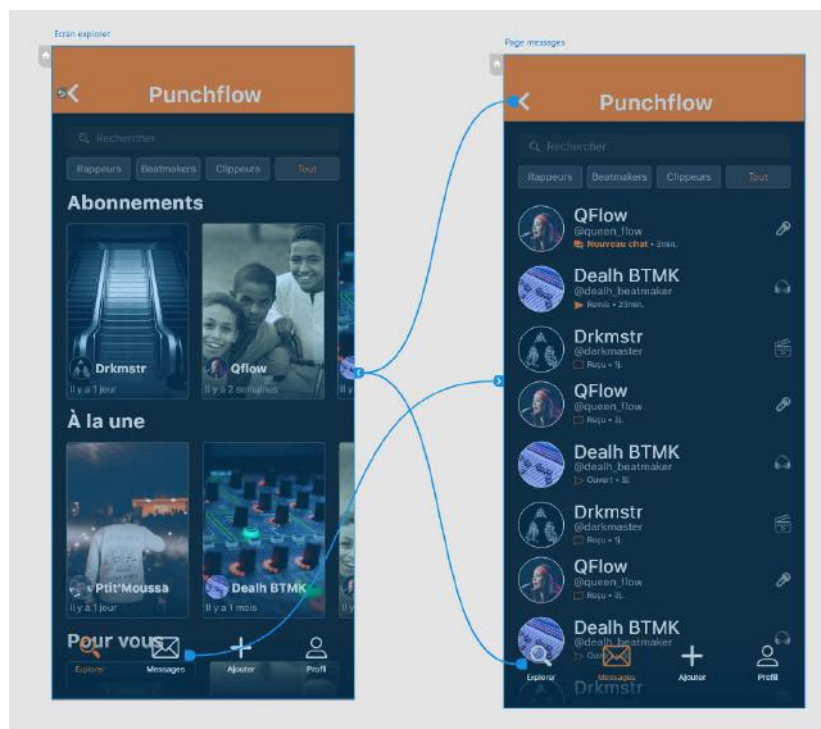
Route	/users/signin
Méthode HTTP	GET
Paramètres requis	username, password
Données retournées par le serveur	id de l'utilisateur
Code de succès	200
Code d'échec	404
Requête MySQL	'SELECT id FROM Users WHERE username = \${user.username} && password = \${user.password}'
Description	Cette requête récupère l'id d'un utilisateur en base de données, s'il existe.

3- Afficher ses conversations

a) Description du Userflow

Étape	Action	Écran
1	L'utilisateur est connecté dans l'application.	Écran explorer, écran profil, écran ajouter publication
2	L'utilisateur appuie sur le bouton message de la barre de navigation.	Écran explorer, écran profil, écran ajouter publication
3	L'utilisateur est maintenant sur son écran messages, contenant toutes ses conversations avec d'autres utilisateurs.	Page messages

b) Userflow



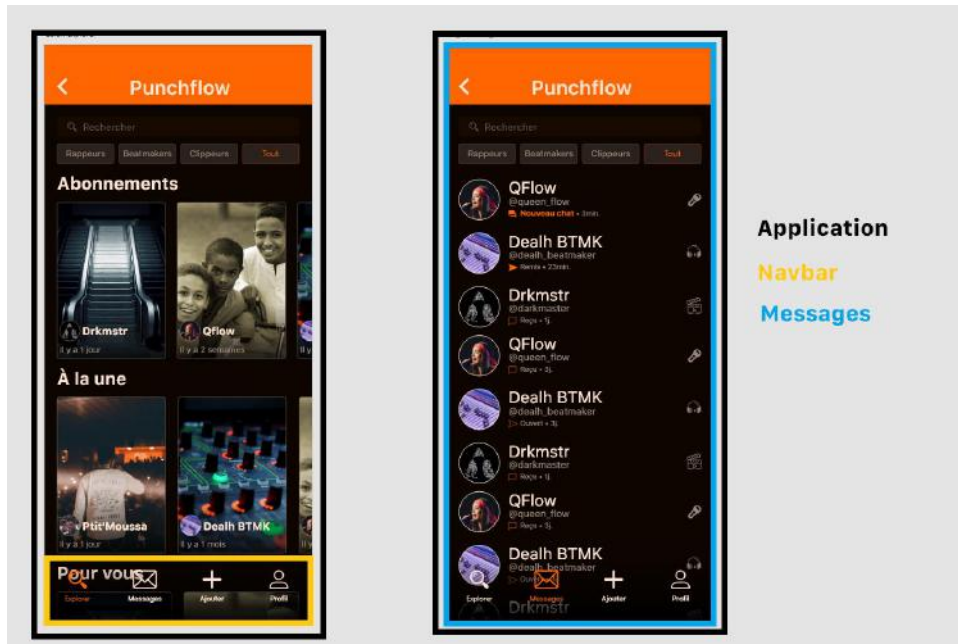
c) Kanban

À faire	En cours	Terminé
Back-end - Dans l'API, créer la route pour récupérer les conversations		
Front-end – Créer le composant de la barre de navigation et de l'écran messages		
Front-end - Faire la fonctionnalité avec la requête HTTP pour récupérer les conversations en BDD		

d) Components Angular

Component	Description
Barre de navigation	Component avec les boutons pour changer de page dans l'application
Écran messages	Component avec la liste des conversations

e) Userflow components Angular



f) Requête HTTP

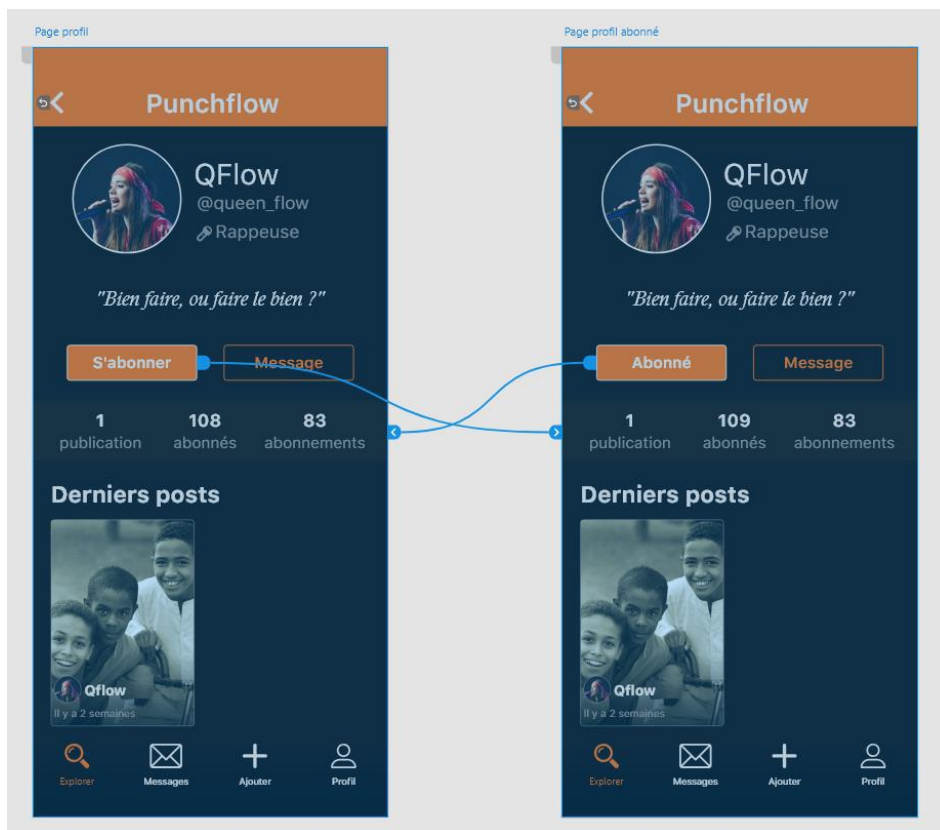
Route	/chats
Méthode HTTP	GET
Paramètres requis	id (identifiant de l'utilisateur connecté)
Données retournées par le serveur	Les conversations de l'utilisateur connecté
Code de succès	200
Code d'échec	404
Requête MySQL	'SELECT * FROM Chats WHERE \${user.id} = id_user_1 \${user.id} = id_user_2'
Description	Cette requête récupère les conversations dans lesquelles l'id de l'utilisateur connecté est dans l'une des colonnes id_user_1 ou id_user_2.

4- S'abonner / se désabonner à un utilisateur

a) Description du Userflow

Étape	Action	Écran
1	L'utilisateur est sur la page profil d'un autre utilisateur	Écran profil
2	L'utilisateur appuie sur le bouton s'abonner (ou abonné) pour s'abonner (ou se désabonner) à un utilisateur.	Écran profil

b) Userflow



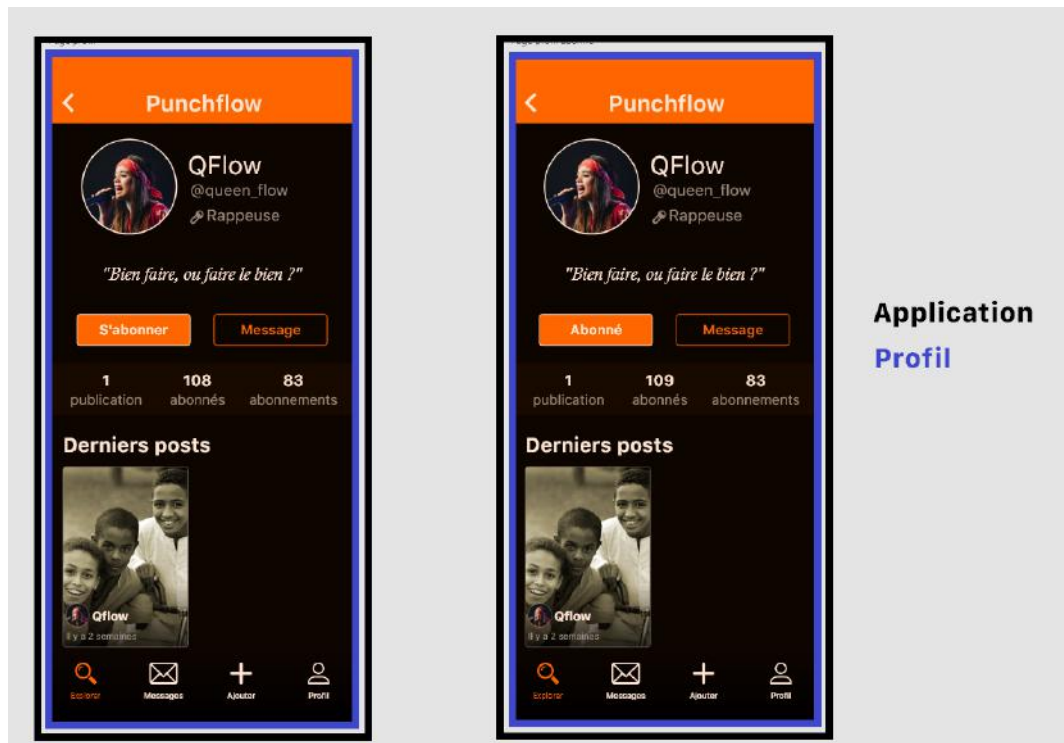
c) Kanban

À faire	En cours	Terminé
Back-end - Dans l'API, créer les routes pour s'abonner, se désabonner d'un profil		
Front-end – Créer le composant de la page profil avec le bouton pour s'abonner/ se désabonner		
Front-end - Faire la fonctionnalité avec la requête HTTP pour ajouter/supprimer un abonnement en BDD		

d) Components Angular

Component	Description
Page profil	Component avec les informations, les chiffres et les publications d'un profil

e) Userflow components Angular



f) Requêtes HTTP

Route	/follow/:id
Méthode HTTP	POST (s'abonner) – DELETE (se désabonner)
Paramètres requis	L'id de l'utilisateur connecté
Données retournées par le serveur	Pas de données retournées pas le serveur
Code de succès	201 (s'abonner) – 200 (se désabonner)
Code d'échec	500 (s'abonner) – 404 (se désabonner)
Requêtes MySQL	'INSERT INTO Followers VALUES (\${user.id}, \${:id})' - (s'abonner) 'DELETE FROM Followers WHERE user_follower_id = \${user.id} && user_followed_id = \${:id}' – (se désabonner) (:id est l'id du profil visité)
Description	Cette requête récupère les conversations dans lesquelles l'id de l'utilisateur connecté est dans l'une des colonnes id_user_1 ou id_user_2.

Pour terminer

Vous pouvez tester le prototype de la majorité des fonctionnalités de l'application sur ce lien : <https://xd.adobe.com/view/6aae51fe-785b-4386-7b4f-ca79e2e80893-1c9f/?fullscreen>