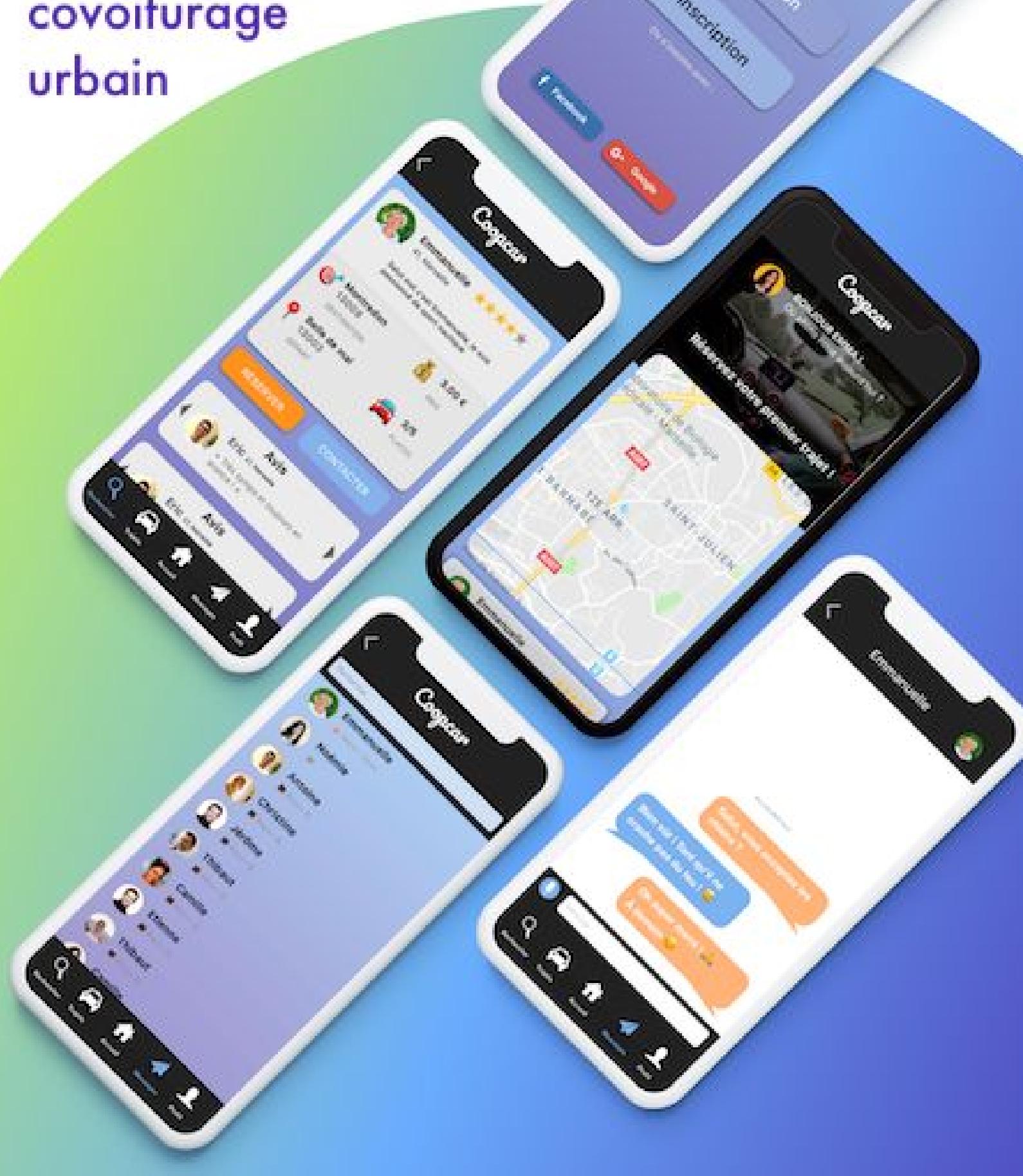


Coopcar

Le réseau social de
covoiturage
urbain



PRÉSENTATION DU PROJET

1. Product Vision

Coopcar est une application web de réseau social qui permet aux habitants de Marseille de simplifier leurs déplacements en trouvant des covoitages à petit prix, plus adaptés à leurs trajets. L'application permet aux personnes véhiculées de proposer des trajets urbains et coopératifs.

2. Poster de l'application

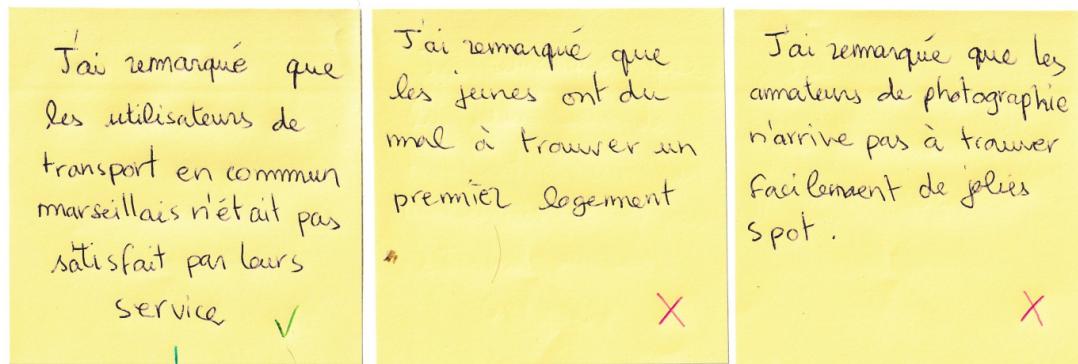


3. Product Backlog

ID	STORY	PRIORITY
1	En tant qu'utilisateur quand j'arrive sur l'application je dois avoir le choix de me connecter ou de m'inscrire, je veux donc pouvoir me déconnecter et me reconnecter	Must
2	En tant qu'utilisateur quand je me suis connecté sur l'application, je veux faire une recherche afin de trouver un trajet	Must
3	En tant qu'utilisateur quand je suis connecté sur l'application, je souhaite pouvoir communiquer avec un autre utilisateur par message	Must
4	En tant qu'utilisateur conducteur quand je suis connecté, je veux pouvoir publier un trajet	Must
5	En tant qu'utilisateur quand j'ai effectué un trajet avec un autre utilisateur, je souhaite pouvoir lui mettre une note	Should
6	En tant qu'utilisateur quand je suis sur mon profil, je souhaite pouvoir modifier mes informations.	Should
7	En tant qu'utilisateur quand je suis connecté sur l'application, je souhaite pouvoir voir tous les trajets que j'ai déjà effectué	Should

PROCESSUS DE DESIGN THINKING

1. Idées solution Agnostic



→ LES PROBLÈMES QU'ILS RENCONTRENT SONT:

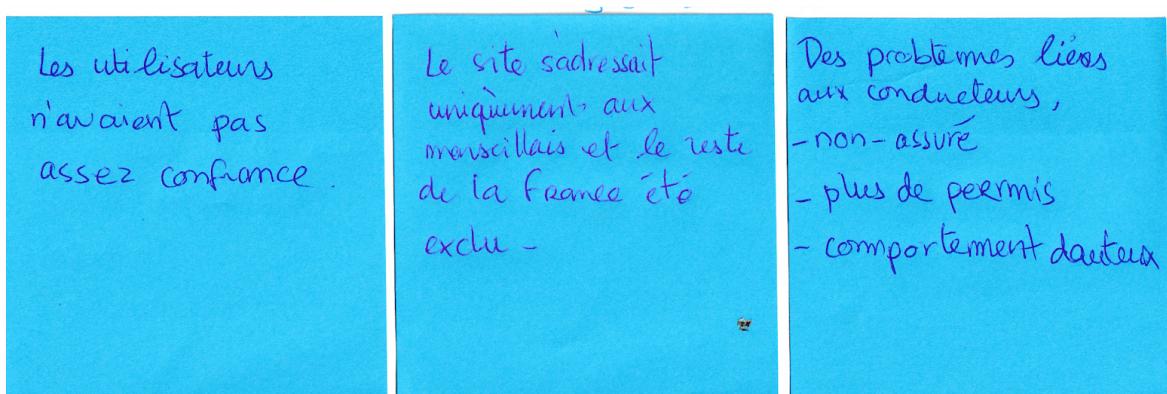
Les prix des titres de transport sont trop cher.	Serices mal déscrivis :	- metro - tram - bus.	les horaires ne sont pas souvent réellement respectés
--	-------------------------	-----------------------------	---

2. Analyse pré-mortem

MERCREDI 03/04/2020 à 11H51

Victime de son exclusivité, le réseau social de covoiturage marseillais, n'a pas réussi à convaincre.

Presque 1 an après la sortie de ce nouveau réseau sociale, les chiffres le montrent, une trop faible activité sur la plateforme due à son exclusivité marseillaise, force l'application Coopcar à fermer.

le réseau fut déourné en plateforme de VTC (Vigile)

Trop de concurrence avec UBER, TAXI ...

3. Analyse concurrentielle

Pour la mise en place ce projet, j'ai dû procéder à une veille concurrentielle.

En effet, de nombreux services de covoitages ou de transports existent déjà, j'ai donc dû m'informer à propos des concurrents possibles que mon réseau pourrait avoir en listant les concurrents directs et indirects.

Voici l'analyse concurrentielle qui en découle.

Concurrents directs



Concurrents indirects

4. Persona

Voici par exemple de profil cible correspondant au utilisateurs possible de mon application. **Illustré ci-dessous.**



Aline, 26

Aide soignante, et amatrice de mode
Saint-Antoine 13016, Marseille, France.

« J'ai encore attendu mon bus 15min ce matin...je te raconte pas la galère avec mon responsable quand je suis arrivée sérieux ! »

À propos

Jeune diplômée en médecine travaillant en tant qu'aide-soignante. Sa première passion est la mode, amatrice de haute couture, sa culture de la mode et très variées. N'étant pas véhiculée elle utilise quotidiennement les transports en commun qui ne respectent pas les horaires, à cause de quoi elle est souvent en retard, chose qu'elle ne peut pas se permettre.

Comportements

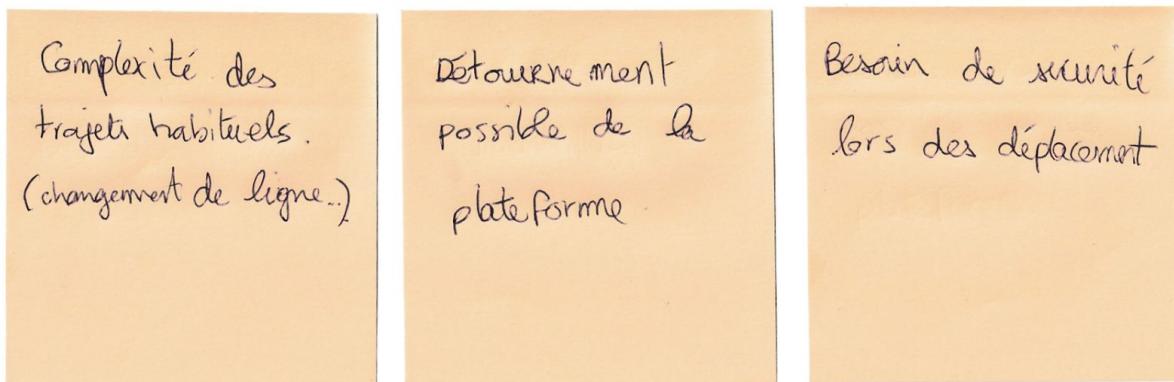
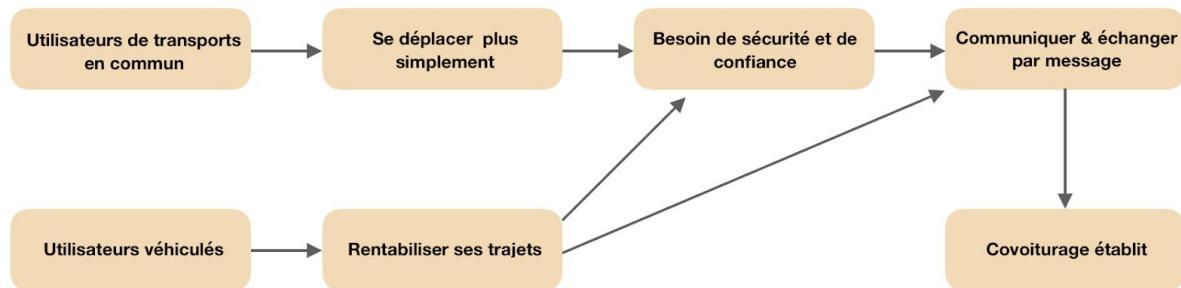
- À une vie très chargée
- Ne peux pas être en retard
- Eviter les trajets à rallonges

Buts & besoins

- Trajets rapide
- Sécurité lors de ces déplacements
- Un moyen simple de se déplacer

5. Points de douleurs - Diagramme de solutions

Le principal point de douleur de cette application est le besoin de confiance que ressentent les utilisateurs de réseaux sociaux coopératif comme celui ci. Afin de pallier à ce problème la solution qui émerge est donc la communication. **Ci-dessous le diagramme de solution.**



6. Zoning



7. Wireframe basse-fidélité

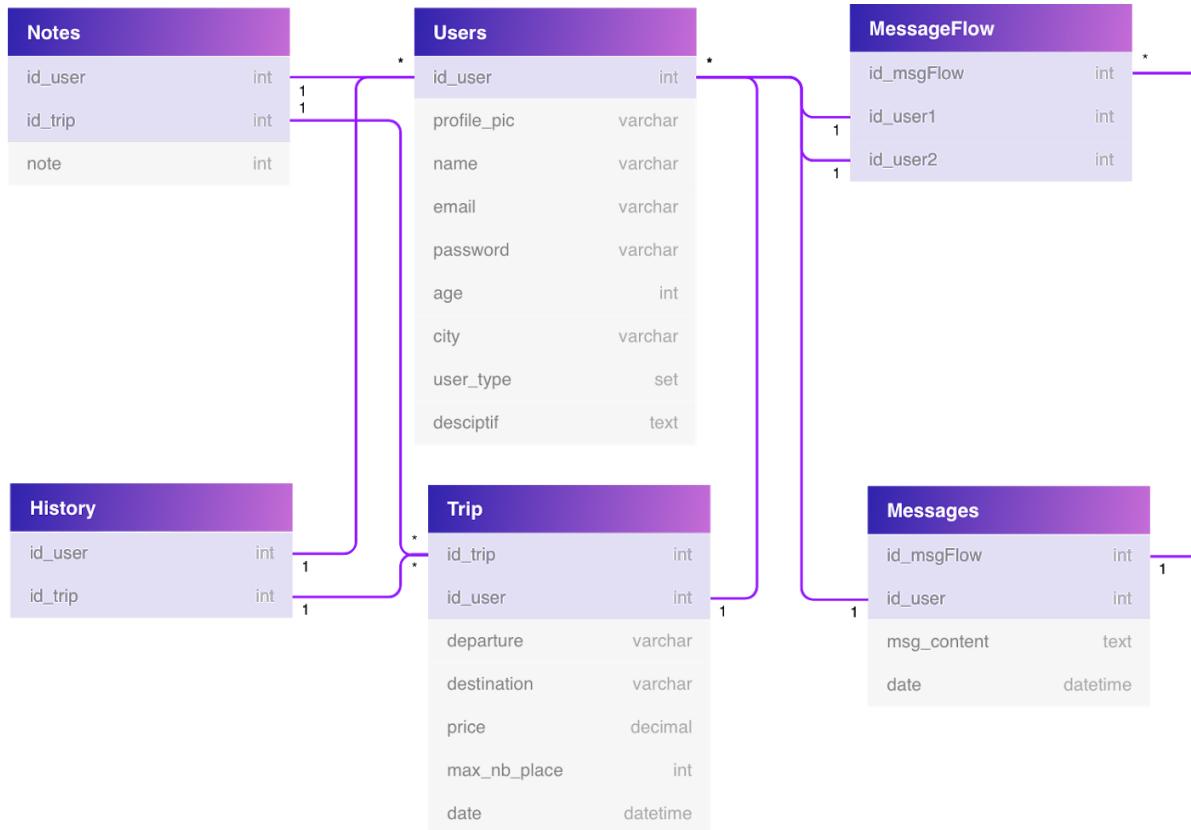


8. Wireframe haute-fidélité

The wireframe grid consists of 8 panels:

- Panel 1 (Top Left):** Welcome screen with "Bienvenue sur Coopcarr!", a "Connexion" button, an "Inscription" button, and social login buttons for Facebook and Google.
- Panel 2 (Top Middle):** Registration screen titled "Inscription" with fields for Name, Email, and Password, and a "S'inscrire" button.
- Panel 3 (Top Right):** Login screen titled "Se connecter" with fields for Email and Password, and a "Connexion" button.
- Panel 4 (Bottom Left):** Driver profile screen for "Emmanuelle" (41, Marseille) with a 5-star rating, departure location "Belle de mai 13008", destination "Montredon 13008", 3/5 places available, and a price of 3.00 €. Includes a map and a "RÉSERVER" button.
- Panel 5 (Bottom Middle):** Driver profile screen for "Emmanuelle" (41, Marseille) with a 5-star rating, departure location "Belle de mai 13008", destination "Montredon 13008", 3/5 places available, and a price of 3.00 €. Includes a map and a "RÉSERVER" button.
- Panel 6 (Bottom Right):** Driver profile screen for "Emmanuelle" (41, Marseille) with a 5-star rating, departure location "Belle de mai 13008", destination "Montredon 13008", 3/5 places available, and a price of 3.00 €. Includes a map and a "RÉSERVER" button.
- Panel 7 (Top Middle):** Driver profile screen for "Emmanuelle" (41, Marseille) with a 5-star rating, departure location "Belle de mai 13008", destination "Montredon 13008", 3/5 places available, and a price of 3.00 €. Includes a map and a "RÉSERVER" button.
- Panel 8 (Top Right):** Driver profile screen for "Emmanuelle" (41, Marseille) with a 5-star rating, departure location "Belle de mai 13008", destination "Montredon 13008", 3/5 places available, and a price of 3.00 €. Includes a map and a "RÉSERVER" button.

PRÉSENTATION DE LA BASE DE DONNÉES



PRÉSENTATION DES FONCTIONNALITÉS

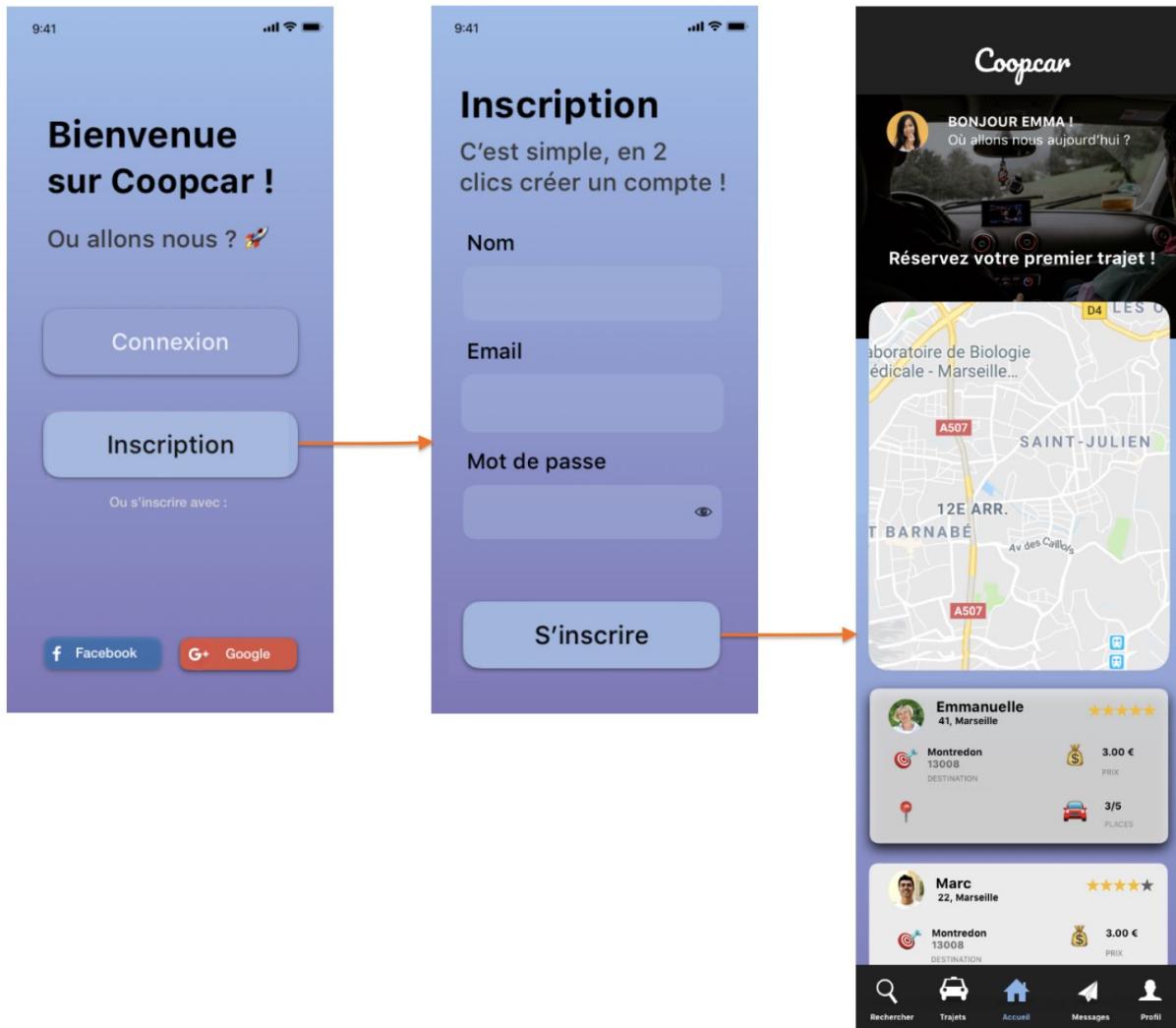
1. Inscription

1.1 Description du User flow

Étape	Action	Type d'écran	Exemple
1	L'utilisateur ouvre l'application et appui sur "Inscription"	First screen	Écran d'accueil
2	L'écran d'inscription s'ouvre et l'utilisateur entre ses informations	Entité	Formulaire d'inscription

3	L'utilisateur arrive sur le Feed	Interaction	Feed
---	----------------------------------	-------------	------

1. 2 User flow haute fidélité



1. 3 Kanban

À faire		En cours		Terminer	
Back-end	Front-end	Back	Front	Back	Front
Après avoir créer l'API, créer la route permettant l'ajout d'un utilisateur	Créer le component nécessaire à l'écran d'inscription				
	Créer le component nécessaire à l'écran d'accueil				
	Créer la fonctionnalité et la				

	requête HTTP pour insérer le nouvel utilisateur en BDD				
--	--	--	--	--	--

1. 4 Description des components

Component	Description
FirstScreen Component	Component contenant les boutons
SignUp Component	Component contenant le formulaire d'inscription
Home Component	Contenant les différents components de mon application

1. 5 User flow des components

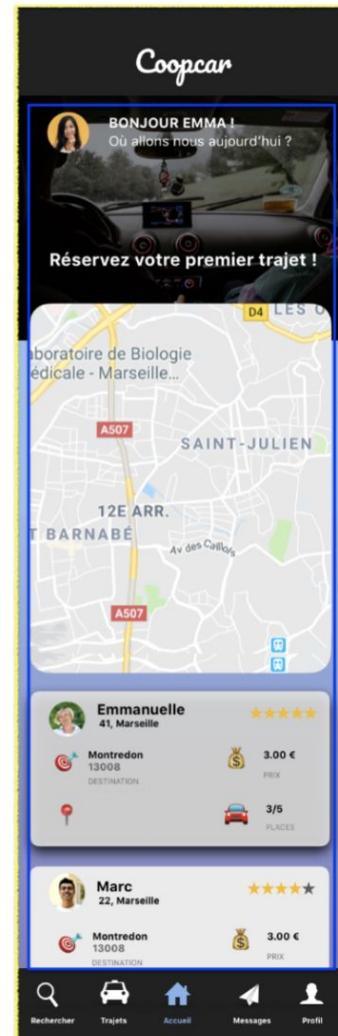


App component



SignUp component

FirstScreen component



Home component

1. 6 Requêtes HTTP

Route	/users/signup
Méthode	POST
Paramètres requis	name, email, password
Données retournées	Pas de données retournées par le serveur
Code de succès	201
Code d'échec	500
Requête MySQL	INSERT INTO Users (name, email, password)

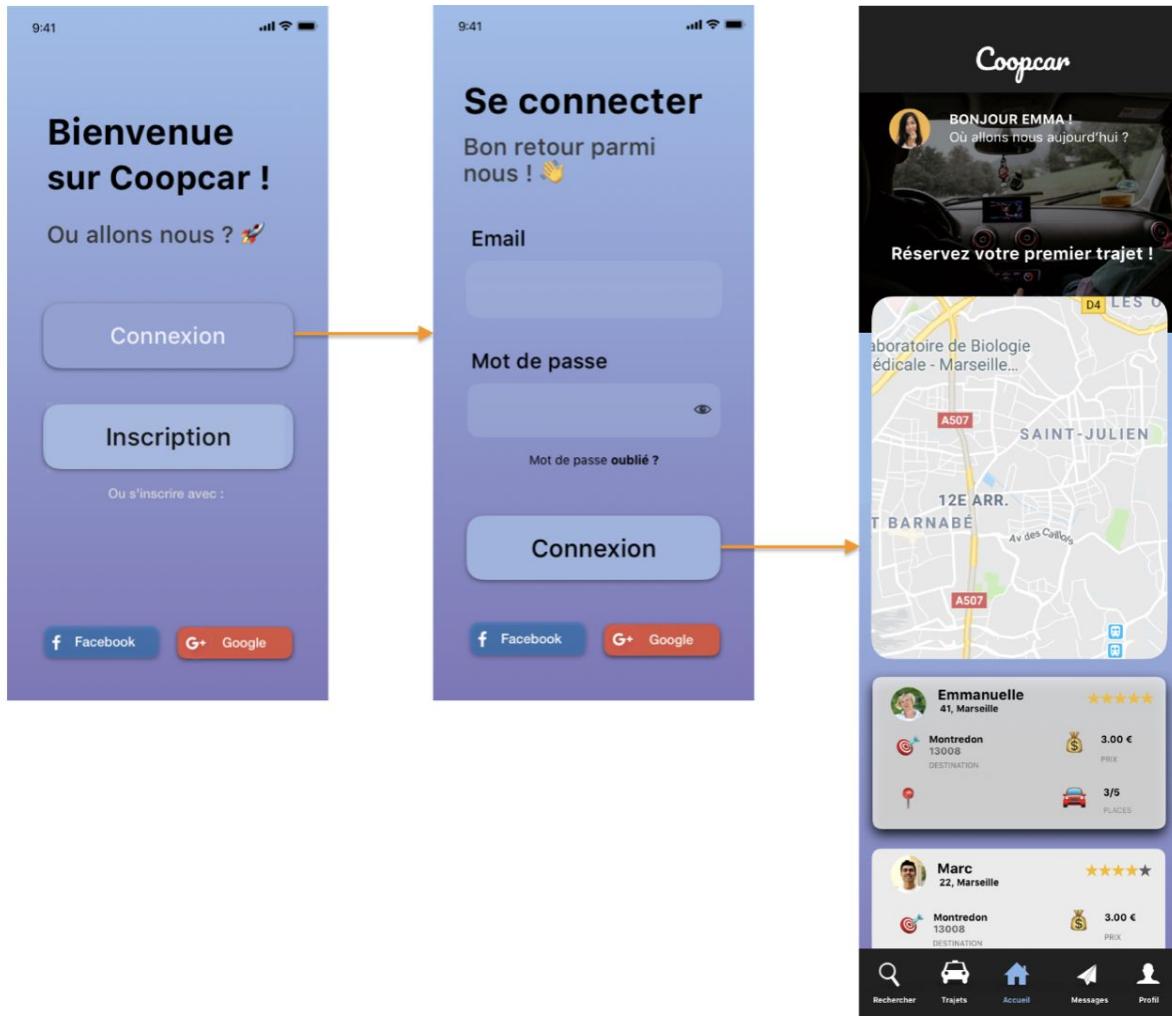
	VALUES ([nom],[email],[mot de passe]);
Description de la requête	Permet d'insérer un nouvel utilisateur en base de données

2. Connexion

2.1 Description du User flow

Étape	Action	Type d'écran	Exemple
1	L'utilisateur ouvre l'application et appui sur "Connexion"	First screen	Écran d'accueil
2	L'écran de connexion s'ouvre et l'utilisateur entre son mot de passe	Entité	Formulaire de connexion
3	L'utilisateur arrive sur le Feed	Interaction	Feed

2. 2 User flow haute fidélité



2. 3 Kanban

À faire		En cours		Terminer	
Back-end	Front-end	Back	Front	Back	Front
Créer la route permettant d'accéder au compte que nous avons créé	Créer le component nécessaire à l'écran de connexion				
	Créer la fonctionnalité et la requête HTTP pour accéder à son compte				

2. 4 Description des components

Component	Description
FirstScreen Component	Component contenant les boutons
SignIn Component	Component contenant le formulaire de connexion
Home Component	Contenant les différents components de mon application

2. 5 User flow des components



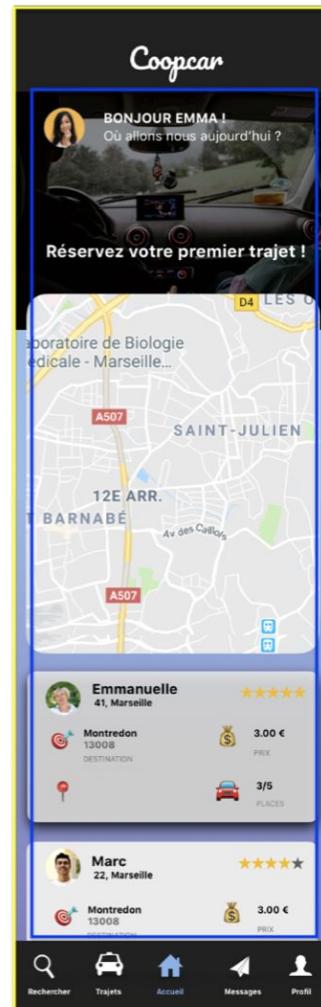
App component

FirstScreen component



SignIn component

Home component



2. 6 Requêtes HTTP

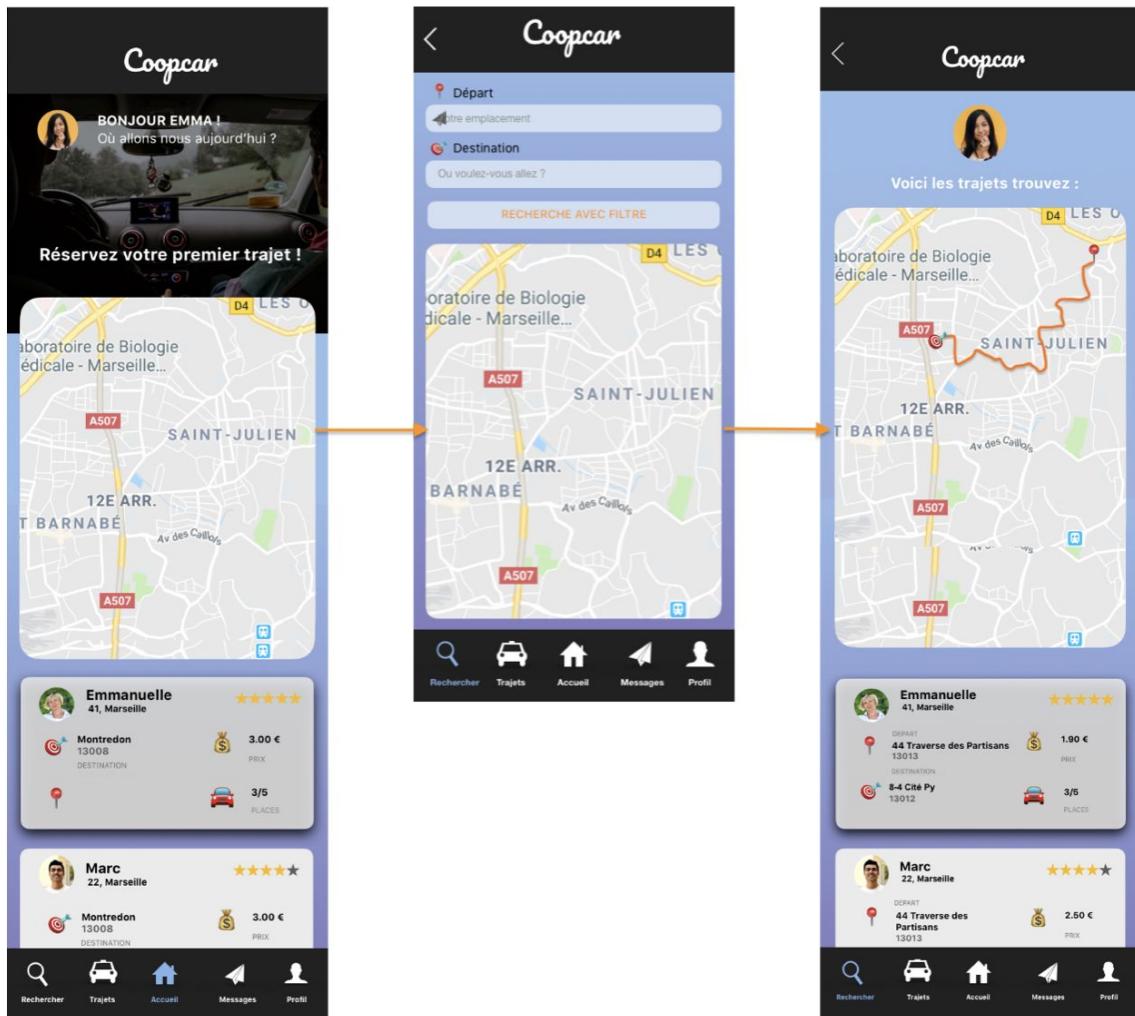
Route	/users/signin
Méthode	GET
Paramètres requis	email, password
Données renvoyées	id de l'utilisateur
Code de succès	200
Code d'échec	404
Requête MySQL	'SELECT id FROM Users WHERE email = \$(user.email) && password =\$(user.password)'
Description de la requête	Permet de récupérer l'id d'un utilisateur si il existe en base de données

3. Recherche

3. 1 Description du User flow

Étape	Action	Type d'écran	Exemple
1	L'utilisateur appuie sur la carte pour faire une recherche	Page d'accueil	Page Home/Feed
2	L'écran de recherche apparaît, l'utilisateur entre sa recherche	Entité	Recherches
3	Il trouve une liste de trajets en fonction	Interaction	Résultat de recherche

3. 2 User flow haute fidélité



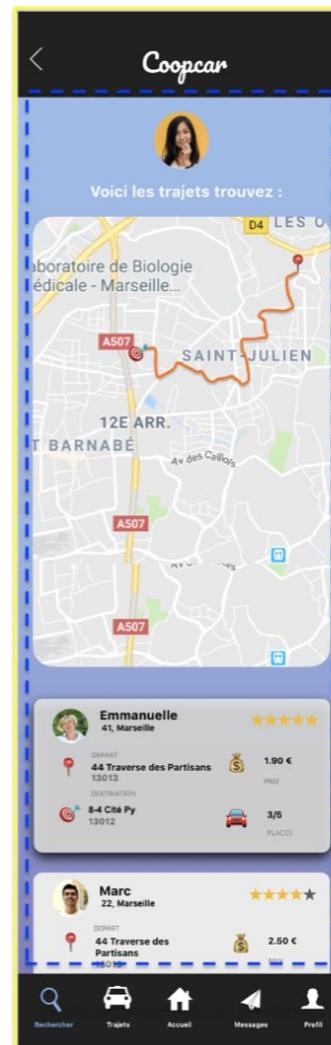
3. 3 Kanban

À faire		En cours		Terminer	
Back-end	Front-end	Back	Front	Back	Front
Créer la route permettant de faire une recherche	Créer le component nécessaire à l'écran de recherche				
	Créer la fonctionnalité et la requête HTTP pour rechercher un trajet				

3. 4 Description des components

Component	Description
Home Component	Component contenant les différents components de mon l'application
Research Component	Component contenant le formulaire de recherche

3. 5 User flow des components



App component

Research component

Home component

3. 6 Requêtes HTTP

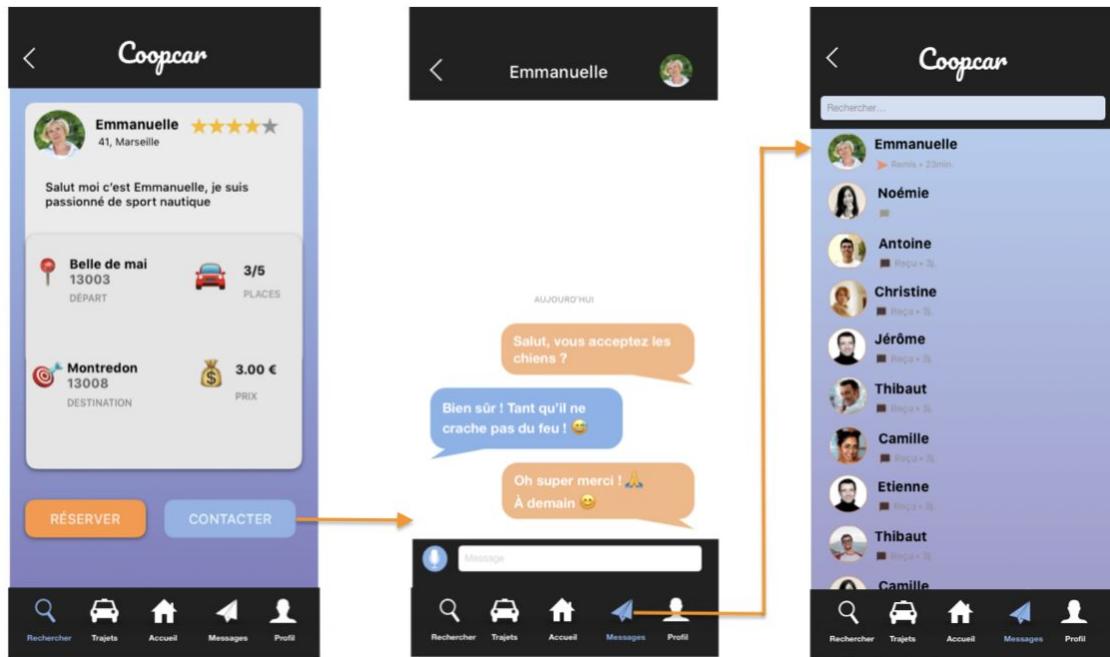
Route	/trips
Méthode	GET
Paramètres requis	departure, destination
Données retournées	id des publications
Code de succès	200
Code d'échec	404
Requête MySQL	SELECT id FROM Trips WHERE departure = \${departure} && destination = \${destination}'
Description de la requête	Permet de rechercher un trajet en base de données

4. Message

4. 1 Description du User flow

Étape	Action	Type d'écran	Exemple
1	L'utilisateur appui sur le bouton "Contacter" présent sur une offre	Publication d'un utilisateur	Résultat de recherche
2	Il accède à une nouvelle conversation	Entité	Conversations
3	Il affiche ses messages	Interaction	Liste des conversations

4. 2 User flow haute fidélité



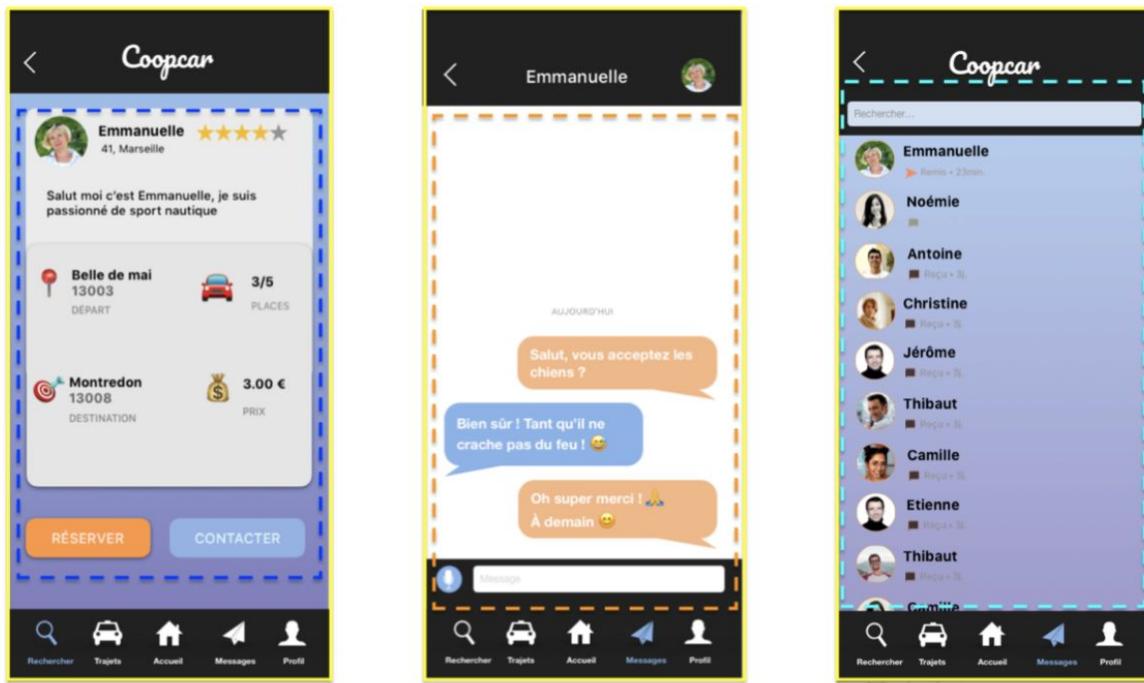
4. 3 Kanban

À faire		En cours		Terminer	
Back-end	Front-end	Back	Front	Back	Front
Créer la route permettant d'afficher et de lister ses messages	Créer le component nécessaire à l'écran de publication				
	Créer le component nécessaire à l'écran de conversation				
	Créer la fonctionnalité et la requête HTTP pour afficher ses messages				

4. 4 Description des components

Component	Description
Trip Component	Component contenant un trajet
Msg Component	Component contenant une conversation
MsgFlow Component	Component contenant toutes les conversations

4. 5 User flow des components



App component

MsgFlow component

Trip component

Msg component

4. 6 Requêtes HTTP

Route	/msg/:id
Méthode	GET
Paramètres requis	id du destinataire
Données retournées	id de l'utilisateur qui a envoyé le message, le contenu du

	message, la date de chaque message
Code de succès	200
Code d'échec	404
Requête MySQL	'SELECT id_user, msg_content, date FROM Messages WHERE id_msgFlow = (SELECT id_msgFlow FROM MessageFlow WHERE (id_user1 = \${user.id} && id_user2 = \${:id}) (id_user1 = \${:id} && id_user2 = \${user.id}))'
Description de la requête	Permet de récupérer la liste des messages d'une conversation en base de données

INVISION - PRÉSENTATION

<https://invis.io/5GSB75WKNC>

DOSSIER GITHUB DU PROJET

<https://github.com/Janis-Lobry/Coopcar-Projet-4>