

Dossier de projet

CONCEPTEUR D'APPLICATION WEB ET MOBILE



Réalisation de l'application Web et Mobile

Présenté par Ahmed Sékou Magassouba

SOMMAIRE

Introduction	3
Présentation Personnelle	3
Présentation du projet en Anglais	3
Compétences couvertes par le projet	3
Organisation et cahier des charges	5
Analyse de l'existant	5
1. Applications de chat populaires	5
2. Points forts et points faibles de l'existant	5
3. Innovations récentes	6
Composition de l'application	6
1. Partie web pour les administrateurs	6
2. Partie mobile pour les utilisateurs	7
Les fonctionnalités attendues	8
Application mobile	8
1. Page de chat général :	8
2. Page de profil :	8
3. Page d'annuaire :	8
4. Page de connexion et d'inscription :	9
5. Page du listing des conversations d'un utilisateur :	9
6. Page de chat privé :	9
1. gestion du channels :	9
2. Bannissement d'utilisateurs :	10
3. Droits d'administration :	10
4. Suppression du channel :	10
5. Possibilité de quitter le channel :	10
Context technique	11
CONCEPTION DU PROJET	12
Choix de développement	12
1. Choix des langages	12
JavaScript en front-end :	12
JavaScript en back-end :	13
2. Choix des framework	16
1. React Native avec Expo (Frontend) :	16
2. Express (Backend) :	17
3. En résumé	17
Outils utilisés	17

Organisation du projet	18
1. Diagrammes de Gantt	18
2. Rôles et responsabilités	23
3. GitHub Kanban	24
Utilisation des Milestone (jalons)	25
Utilisation des Labels	26
Les différents type de label utilisé dans notre méthodes :	27
Evaluation des priorités	28
Estimation de la vélocité	29
Architecture logicielle	29

Introduction

Présentation Personnelle

Je m'appelle MAGASSOUBA Ahmed Sékou, j'ai trente-six ans. Anciennement technicien électronique R&D, dans l'innovation énergétique. Voulant faire évoluer mon profil de compétence. J'ai décidé d'intégrer un boot camp en Février 2021 au sein d'Epitech, qui m'a permis de découvrir les langages liés au web. J'ai intégré le cursus de la CodingSchool de la plateforme en Septembre 2021. J'ai obtenu mon titre de développeur web et mobile . Aujourd' hui je suis en Bachelor 3 afin de préparer mon titre de concepteur d'application web et mobile en alternance au sein de mon centre de formation La Plateforme.

Présentation du projet en Anglais

Compétences couvertes par le projet

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

Organisation et cahier des charges

Analyse de l'existant

Dans le cadre de la conception de notre application de chat, il est essentiel de réaliser une analyse approfondie de l'existant afin de comprendre les fonctionnalités déjà disponibles sur le marché et d'identifier les opportunités d'amélioration. Cette analyse nous permettra de définir les objectifs de notre application et de déterminer les fonctionnalités clés que nous devons inclure pour offrir une expérience utilisateur exceptionnelle.

1. Applications de chat populaires

Nous avons étudié les applications de chat les plus populaires et les mieux établies du marché, telles que WhatsApp, Facebook Messenger et Telegram. Ces applications offrent un large éventail de fonctionnalités, notamment la messagerie instantanée, les appels vocaux et vidéo, les groupes de discussion, les émoticônes, la possibilité de partager des fichiers et des images, etc. Il est important d'examiner attentivement ces fonctionnalités pour déterminer celles que nous devons intégrer dans notre application.

2. Points forts et points faibles de l'existant

L'analyse de l'existant nous a permis d'identifier les points forts et les points faibles des applications de chat existantes. Parmi les points forts, on peut citer :

- Interface utilisateur conviviale et intuitive.
- Capacité à envoyer des messages instantanés à des contacts individuels ou à des groupes.
- Possibilité de passer des appels vocaux et vidéo de haute qualité.
- Fonctionnalités avancées de confidentialité et de sécurité.
- Prise en charge multiplateforme (Web, iOS, Android).

Cependant, nous avons également identifié certains points faibles que nous devons prendre en compte pour améliorer notre application :

-
- Manque de personnalisation des conversations et des profils d'utilisateurs.
 - Difficultés à trouver rapidement des messages ou des informations spécifiques.
 - Limitations dans le partage de fichiers et de médias.
 - Absence de certaines fonctionnalités de modération et de filtrage du contenu.

3. Innovations récentes

Nous avons également examiné les dernières innovations dans le domaine des applications de chat. Certaines tendances émergentes incluent :

- Intégration de chatbots pour une assistance automatisée.
- Intégration de l'intelligence artificielle pour des suggestions de réponses rapides.
- Utilisation de la réalité augmentée pour des conversations plus immersives.
- Fonctionnalités de chiffrement de bout en bout pour une sécurité accrue.

Ces innovations peuvent servir de sources d'inspiration pour le développement de notre application de chat.

Composition de l'application

Notre application de chat est conçue pour répondre aux besoins spécifiques de deux catégories d'utilisateurs : les administrateurs et les utilisateurs. Afin de fournir une expérience optimale à chaque groupe, nous avons développé deux interfaces distinctes : une partie web dédiée aux administrateurs et une partie mobile destinée aux utilisateurs.

1. Partie web pour les administrateurs

La partie web de notre application est spécialement conçue pour permettre aux administrateurs de gérer et de superviser le fonctionnement global du système de chat. Les fonctionnalités clés de la partie web comprennent :

-
- Gestion des utilisateurs : Les administrateurs peuvent créer, modifier et supprimer des comptes utilisateur, ainsi que gérer les autorisations d'accès.
 - Modération des contenus : Les administrateurs ont la possibilité de surveiller les conversations, d'appliquer des filtres de contenu et de prendre des mesures appropriées en cas de violation des règles d'utilisation.
 - Statistiques et rapports : La partie web offre des fonctionnalités de génération de rapports et de visualisation des données pour permettre aux administrateurs de suivre l'activité des utilisateurs, les performances du système et d'obtenir des informations précieuses pour améliorer l'expérience globale.

2. Partie mobile pour les utilisateurs

La partie mobile de notre application est conçue pour offrir une expérience conviviale et intuitive aux utilisateurs finaux. Les fonctionnalités principales de la partie mobile comprennent :

- Messagerie instantanée : Les utilisateurs peuvent envoyer et recevoir des messages en temps réel, individuellement ou dans des groupes de discussion.
- Appels vocaux et vidéo : L'application permet aux utilisateurs d'effectuer des appels vocaux et vidéo de haute qualité avec leurs contacts.
- Partage de fichiers et de médias : Les utilisateurs peuvent partager des fichiers, des images, des vidéos et d'autres types de contenus avec leurs contacts.
- Personnalisation du profil : Les utilisateurs ont la possibilité de personnaliser leur profil en ajoutant des photos, des informations personnelles et des préférences de confidentialité.

En séparant les fonctionnalités destinées aux administrateurs de celles destinées aux utilisateurs finaux, nous garantissons une interface adaptée aux besoins spécifiques de chaque groupe. Les administrateurs bénéficieront d'outils puissants pour gérer et superviser le système, tandis que les utilisateurs pourront profiter d'une expérience fluide et engageante sur leurs appareils mobiles.

Les fonctionnalités attendues

Application mobile

Notre application de chat vise à offrir une expérience conviviale et pratique pour les utilisateurs. Voici les principales fonctionnalités que nous prévoyons d'inclure :

1. Page de chat général :

Cette page constitue l'espace principal de conversation pour les utilisateurs. Elle permet aux utilisateurs connectés d'interagir avec d'autres membres de la communauté, d'échanger des messages en temps réel et de participer à des discussions de groupe. Les utilisateurs pourront consulter les messages précédents, envoyer des réponses, partager des médias et utiliser des fonctionnalités de modération appropriées.

2. Page de profil :

La page de profil permet à chaque utilisateur de créer et de gérer son propre profil.. Ils auront également la possibilité de gérer les paramètres liés à leur compte.

3. Page d'annuaire :

L'annuaire de l'application regroupe tous les utilisateurs inscrits. Cette page permet aux utilisateurs de découvrir de nouveaux contacts, de rechercher des utilisateurs inscrits. L'utilisateur peut sélectionner un ou plusieurs membre(s) pour lancer une conversation privée.

4. Page de connexion et d'inscription :

Les pages de connexion et d'inscription offriront aux utilisateurs la possibilité de créer un nouveau compte ou de se connecter à leur compte existant. Les utilisateurs pourront fournir leurs informations personnelles, choisir un nom d'utilisateur et un mot de passe sécurisé. Si un utilisateur dispose déjà d'un token valide, il sera automatiquement redirigé vers la page du chat général lorsqu'il ouvrira l'application. Sinon, il sera dirigé vers la page de connexion pour se connecter ou s'inscrire.

5. Page du listing des conversations d'un utilisateur :

Cette page listera tous les channels de conversation auxquels l'utilisateur est abonné ou auquel il a accès. L'utilisateur pourra visualiser les channels existants. Cette fonctionnalité permettra aux utilisateurs de lister ses interactions avec les autres utilisateurs.

6. Page de chat privé :

La page de chat privé permet aux utilisateurs d'engager des conversations individuelles ou collectives avec d'autres utilisateurs. Voici les fonctionnalités spécifiques de cette page :

1. gestion du channels :

Lorsqu'un utilisateur crée un channel de chat privé, il devient le super administrateur de ce channel. En tant que super administrateur, l'utilisateur a les droits spéciaux pour gérer le channel. Il peut bannir un utilisateur spécifique de la conversation, en lui interdisant l'accès au channel. De plus, le super administrateur a la possibilité de nommer un autre utilisateur en tant qu'administrateur du channel. Les administrateurs ont des droits étendus pour gérer les membres et les paramètres du channel. Enfin, le super administrateur a le pouvoir de supprimer complètement le channel si nécessaire.

2. Bannissement d'utilisateurs :

En tant que super administrateur, l'utilisateur a le pouvoir de bannir un utilisateur spécifique du channel de chat privé. Lorsqu'un utilisateur est banni, il perd l'accès au channel et ne peut plus participer à la conversation. Cette fonctionnalité permet de maintenir la sécurité et la gestion appropriées des channels de chat privés.

3. Droits d'administration :

Le super administrateur a la possibilité de nommer d'autres utilisateurs en tant qu'administrateurs du channel. Les administrateurs ont des droits étendus, tels que la gestion des membres, la modification des paramètres du channel et la modération des messages. Cette fonctionnalité permet de déléguer la responsabilité de la gestion du channel à des utilisateurs de confiance.

4. Suppression du channel :

En tant que super administrateur, l'utilisateur a le pouvoir de supprimer complètement le channel de chat privé. Cette action entraînera la suppression de toutes les conversations, des fichiers partagés et des paramètres du channel. Cette fonctionnalité permet aux super administrateurs de gérer de manière appropriée ses channels inactifs ou obsolètes.

5. Possibilité de quitter le channel :

Chaque utilisateur, y compris les super administrateurs et les administrateurs, a la liberté de quitter un channel de chat privé à tout moment. Lorsqu'un utilisateur quitte le channel, il n'aura plus accès aux messages et aux activités ultérieures du channel. Cette fonctionnalité donne aux utilisateurs la liberté de contrôler leur participation aux conversations privées.

En combinant ces fonctionnalités, notre application offre un environnement de chat privé flexible et personnalisé, où les super administrateurs peuvent gérer les

membres, les droits et les paramètres des channels. Les utilisateurs peuvent ainsi engager des conversations privées tout en conservant le contrôle sur leur propre participation.

Nous cherchons à fournir une expérience de chat complète, alliant à la fois des conversations de groupe dynamiques et des échanges privés personnalisés.

L'application offrira ainsi aux utilisateurs une plateforme conviviale.

Contrainte

Context technique

L'application mobile devra être accessible sur tous les systèmes d'exploitation Android et IOS. L'application web devra être accessible sur tous les navigateurs.

CONCEPTION DU PROJET

Choix de développement

1. Choix des langages



JavaScript est un langage de programmation polyvalent qui peut être utilisé à la fois côté client (front-end) et côté serveur (back-end). Voici les avantages de l'utilisation de JavaScript dans ces deux domaines :

JavaScript en front-end :

1. Interaction avec l'interface utilisateur : JavaScript est principalement utilisé pour rendre les pages Web interactives. Il vous permet de manipuler le contenu HTML, de réagir aux actions de l'utilisateur (clics de souris, pressions de touche, etc.) et de mettre à jour dynamiquement l'interface utilisateur sans recharger la page entière.
2. Validation des données côté client : JavaScript peut être utilisé pour valider les données côté client avant de les envoyer au serveur. Cela permet de fournir une expérience utilisateur plus réactive en évitant des allers-retours coûteux avec le serveur pour valider les entrées.
3. Amélioration des performances : JavaScript permet d'effectuer des calculs et des manipulations de données côté client, ce qui réduit la charge sur le serveur et améliore la performance globale de l'application.
4. Manipulation du DOM : JavaScript offre des fonctionnalités puissantes pour manipuler le Document Object Model (DOM), qui représente la structure HTML d'une page. Cela permet de modifier dynamiquement le contenu, le style et le comportement des éléments de la page.

JavaScript en back-end :

1. Partage de code : En utilisant JavaScript à la fois côté client et côté serveur, vous pouvez partager certaines parties de votre code entre les deux. Cela permet une meilleure réutilisation du code et facilite le développement et la maintenance de l'application.
2. Gestion des requêtes et des réponses : JavaScript côté serveur, avec des environnements tels que Node.js, permet de gérer les requêtes HTTP entrantes et de générer des réponses dynamiques. Vous pouvez construire des API, gérer des sessions utilisateur, effectuer des opérations de base de données et d'autres tâches côté serveur.
3. Manipulation des données : JavaScript offre des bibliothèques et des frameworks robustes pour la manipulation des données côté serveur. Vous pouvez accéder à des bases de données, effectuer des opérations de lecture/écriture, analyser et formater des données, et même effectuer des calculs complexes.
4. Évolutivité et performances : L'utilisation de JavaScript côté serveur permet de tirer parti d'une architecture événementielle et non bloquante. Cela signifie que vous pouvez gérer simultanément de nombreuses connexions sans bloquer le thread principal. Cela améliore les performances de l'application, notamment lors de la gestion d'un grand nombre de requêtes en parallèle.

En utilisant JavaScript à la fois côté client et côté serveur, vous bénéficiez d'une expérience de développement plus homogène et d'une plus grande flexibilité dans la création d'applications Web modernes et interactives.



Node.js est un environnement d'exécution JavaScript côté serveur qui permet de développer des applications web et des services backend.

1. Événementiel et non bloquant : Node.js est construit sur le moteur JavaScript V8 de Chrome, qui utilise un modèle événementiel et non bloquant. Cela signifie que Node.js peut gérer un grand nombre de connexions simultanées sans bloquer le thread principal, offrant ainsi une scalabilité élevée et une meilleure réactivité de l'application. Dans un système de chat, où de nombreuses connexions en temps réel doivent être gérées, cette capacité est essentielle.
2. JavaScript côté serveur : En utilisant Node.js comme plateforme côté serveur, vous pouvez utiliser JavaScript à la fois côté client et côté serveur. Cela permet de partager du code entre le front-end et le back-end, ce qui facilite la réutilisation du code et améliore l'efficacité du développement. En outre, si vous avez déjà de l'expérience en JavaScript pour le développement front-end, vous pouvez appliquer vos compétences existantes pour le back-end, réduisant ainsi la courbe d'apprentissage.
3. Écosystème et modules : Node.js dispose d'un écosystème dynamique et d'une vaste bibliothèque de modules, disponibles via le gestionnaire de paquets npm. Ces modules prêts à l'emploi facilitent la mise en place rapide de fonctionnalités supplémentaires dans votre système de chat. Par exemple, vous pouvez utiliser des modules tels que Socket.io pour la communication en temps réel, Express

pour la gestion des routes et des requêtes HTTP, ou encore MongoDB pour la persistance des données.

4. Performances : Node.js est connu pour ses performances élevées, grâce à son modèle de traitement asynchrone et à la gestion efficace des E/S. Cela permet d'optimiser le temps de réponse de votre système de chat, assurant une expérience fluide et réactive pour les utilisateurs.
5. Communauté active : Node.js bénéficie d'une communauté active et engagée, ce qui signifie que vous pouvez trouver facilement du support, des tutoriels, des exemples de code et des solutions aux problèmes que vous pourriez rencontrer lors du développement. La documentation de Node.js est également bien entretenue, ce qui facilite l'apprentissage et la compréhension de la technologie.

En résumé, le choix de Node.js en tant que technologie pour le back-end offre une scalabilité élevée, une réactivité accrue, un partage de code avec le front-end, un écosystème riche en modules et des performances optimisées. Ces facteurs combinés font de Node.js un choix solide pour le développement d'applications en temps réel telles qu'un système de chat.

2. Choix des framework



1. React Native avec Expo (Frontend) :

React Native est un framework de développement d'applications mobiles multiplateformes. Il permet de créer des applications mobiles natives pour iOS et Android en utilisant JavaScript et React, une bibliothèque JavaScript pour la création d'interfaces utilisateur. React Native permet aux développeurs de créer une seule base de code et de la partager entre les plateformes, ce qui permet de réduire le temps et les efforts nécessaires pour développer et maintenir des applications pour différentes plateformes.

Expo est un ensemble d'outils et de services qui facilite le développement d'applications mobiles avec React Native. Il fournit des fonctionnalités supplémentaires et des composants prêts à l'emploi, ainsi qu'un ensemble d'outils de développement, tels qu'un environnement de développement local, un système de build et un ensemble de services cloud pour faciliter le déploiement et la distribution des applications. Expo simplifie le processus de développement d'applications React Native en fournissant une expérience de développement plus fluide et en réduisant la complexité technique.

En utilisant React Native avec Expo pour le frontend de notre application, nous bénéficions d'un cadre de développement puissant et flexible qui nous permet de créer des applications mobiles multiplateformes efficacement, en tirant parti de la réutilisation du code et en offrant une expérience utilisateur native.

2. Express (Backend) :

Express est un framework web pour Node.js qui simplifie le développement d'applications web et d'API. Il fournit des fonctionnalités et des outils pour la création de routes, la gestion des requêtes et des réponses, le traitement des middlewares et la gestion des sessions, entre autres. Express est très populaire en raison de sa simplicité, de sa flexibilité et de sa légèreté, ce qui en fait un choix courant pour la création de services backend.

En utilisant Node.js avec Express pour le backend de notre application, nous pouvons créer des API robustes, gérer les requêtes et les réponses, interagir avec une base de données, gérer l'authentification et l'autorisation des utilisateurs, et fournir des fonctionnalités serveur nécessaires à notre application de chat.

3. En résumé

L'utilisation de ces frameworks nous permet de créer une architecture complète pour notre application, en reliant le frontend (React Native avec Expo) et le backend (Express JS) pour offrir une expérience utilisateur fluide et des fonctionnalités de communication en temps réel entre les utilisateurs.

Ces frameworks bien établis offrent une combinaison puissante pour le développement de votre application de chat, vous permettant de créer une interface utilisateur réactive et riche en fonctionnalités, tout en gérant les aspects du backend pour la communication et le stockage des données.

Outils utilisés

Dans le cadre de ce projet , j'ai utilisé différents outils:

- Visual Studio Code comme éditeur de code (IDE);
- Postman pour tester les requêtes API;
- GitHub pour le versionning de notre code;
- Milestone (jalons) sur GitHub pour organiser notre travail;

-
- NPM pour installer les paquets;
 - Figma pour la création de nos maquettes;
 - Draw.io pour la conception de notre base de données;
 - Expo pour émuler mon application mobile sur mon téléphone;
 - Midjourney pour la création du logo.

Organisation du projet

Ce projet a été réalisé durant notre année de formation, la totalité du mois de janvier à été dédiée à la conception de notre application. Donc l'organisation de notre travail à été essentielle. Nous avons commencé le projet par un listing des tâches principales et essentielles . Pour ce faire , nous avons mis en place les procédés ci-dessous.

1. Diagrammes de Gantt

Le projet de développement d'une application de chat suit un plan structuré en utilisant des diagrammes de Gantt.

gantt

dateFormat YYYY-MM-DD

title Projet de développement d'une application de chat

section Conception

Évaluation du périmètre et cahier des charges :a1, 2023-01-02, 2023-01-05

Choix de la stack technique et agenda :a2, 2023-01-02, 2023-01-05

Conception et réalisation du WireFrame :a3, 2023-01-05, 2023-01-07

Modélisation de la base de données :a4, 2023-01-05, 2023-01-07

Cérémonie Agile SCRUM :a5, 2023-01-06, 2023-01-07

section Mise en place de la BDD et réflexion sur l'architecture du Frontend

Découpage des User Stories :b1, 2023-01-07, 2023-01-08

Développement du Backend avec Node.js :b2, 2023-01-07, 2023-01-13

Réflexion et mise en place de la structure du Frontend en React Native :b3, 2023-01-08, 2023-01-13

Test approfondi de l'API et documentation :b4, 2023-01-09, 2023-01-13

Réunion de synchronisation :b5, 2023-01-12, 2023-01-13

section Évolution du Frontend

Sprint Frontend :c1, 2023-01-15, 2023-01-21

Mise en place de Redux :c2, 2023-01-13, 2023-01-15

Mise en place des stack :c3, 2023-01-13, 2023-01-15

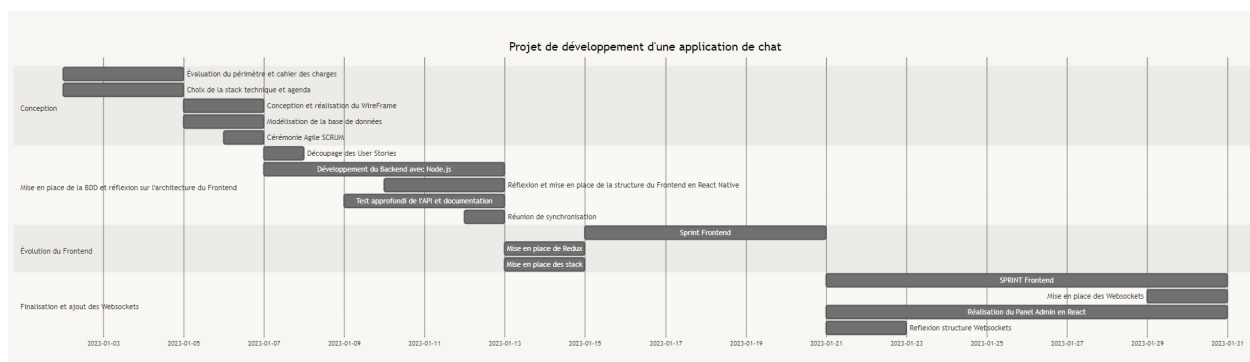
section Finalisation et ajout des Websockets

SPRINT Frontend :d1, 2023-01-21, 2023-01-31

Mise en place des Websockets :d2, 2023-01-29, 2023-01-31

Réalisation du Panel Admin en React :d3, 2023-01-21, 2023-01-31

Reflexion structure Websockets :d4, 2023-01-21, 2023-01-23



- Il commence par une **première semaine** de conception où l'équipe évalue le périmètre, établit un cahier des charges et choisit la stack technique. En parallèle, des membres spécifiques de l'équipe travaillent sur la conception du wireframe et la modélisation de la base de données.

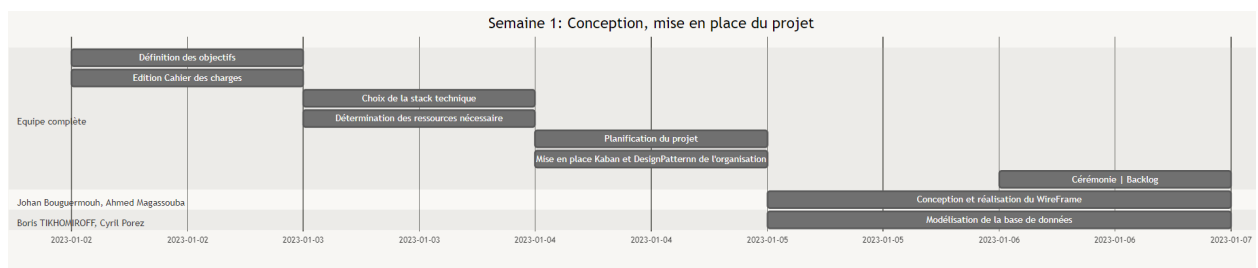
gantt

```
dateFormat YYYY-MM-DD
title Semaine 1: Conception, mise en place du projet

section Equipe complète
Définition des objectifs :a0, 2023-01-02, 2023-01-03
Edition Cahier des charges :a1, 2023-01-02, 2023-01-03
Choix de la stack technique :a2, 2023-01-03, 2023-01-04
Détermination des ressources nécessaire :a3, 2023-01-03, 2023-01-04
Planification du projet :a4, 2023-01-04, 2023-01-05
Mise en place Kaban et DesignPatternn de l'organisation :a5, 2023-01-04, 2023-01-05
Cérémonie | Backlog :a6, 2023-01-06, 2023-01-07

section Johan Bouguermouh, Ahmed Magassouba
Conception et réalisation du WireFrame :b1, 2023-01-05, 2023-01-07

section Boris TIKHOMIROFF, Cyril Porez
Modélisation de la base de données :c1, 2023-01-05, 2023-01-07
```



- La **deuxième semaine** se concentre sur la mise en place de la base de données, le développement du backend et la réflexion sur l'architecture du frontend. Des tests approfondis de l'API sont effectués, et une réunion de synchronisation est organisée.

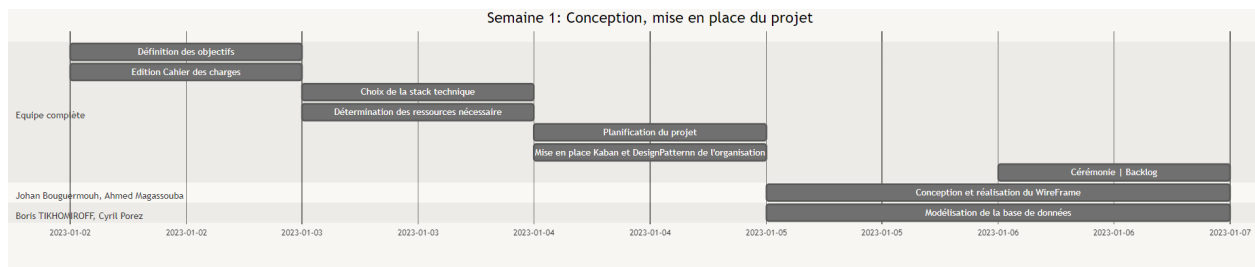
ganttt

```
dateFormat YYYY-MM-DD
title Semaine 1: Conception, mise en place du projet

section Equipe complète
Définition des objectifs :a0, 2023-01-02, 2023-01-03
Edition Cahier des charges :a1, 2023-01-02, 2023-01-03
Choix de la stack technique :a2, 2023-01-03, 2023-01-04
Détermination des ressources nécessaire :a3, 2023-01-03, 2023-01-04
Planification du projet :a4, 2023-01-04, 2023-01-05
Mise en place Kaban et DesignPatternnn de l'organisation :a5, 2023-01-04, 2023-01-05
Cérémonie | Backlog :a6, 2023-01-06, 2023-01-07

section Johan Bouguermouh, Ahmed Magassouba
Conception et réalisation du WireFrame :b1, 2023-01-05, 2023-01-07

section Boris TIKHOMIROFF, Cyril Porez
Modélisation de la base de données :c1, 2023-01-05, 2023-01-07
```



- La **troisième semaine** est dédiée à l'évolution du frontend, avec un sprint spécifique.

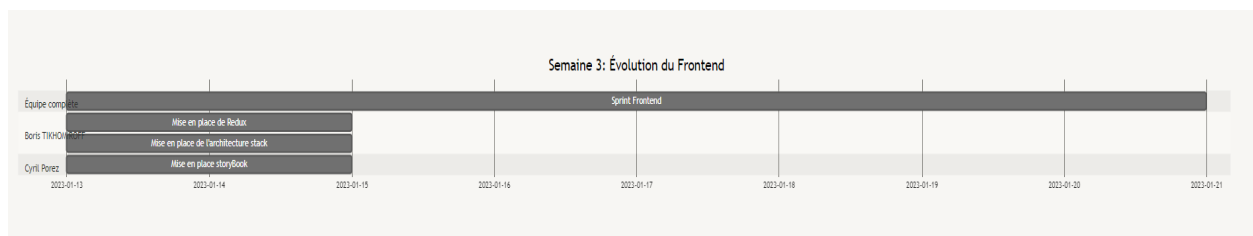
gantt

```
dateFormat YYYY-MM-DD
title Semaine 3: Évolution du Frontend

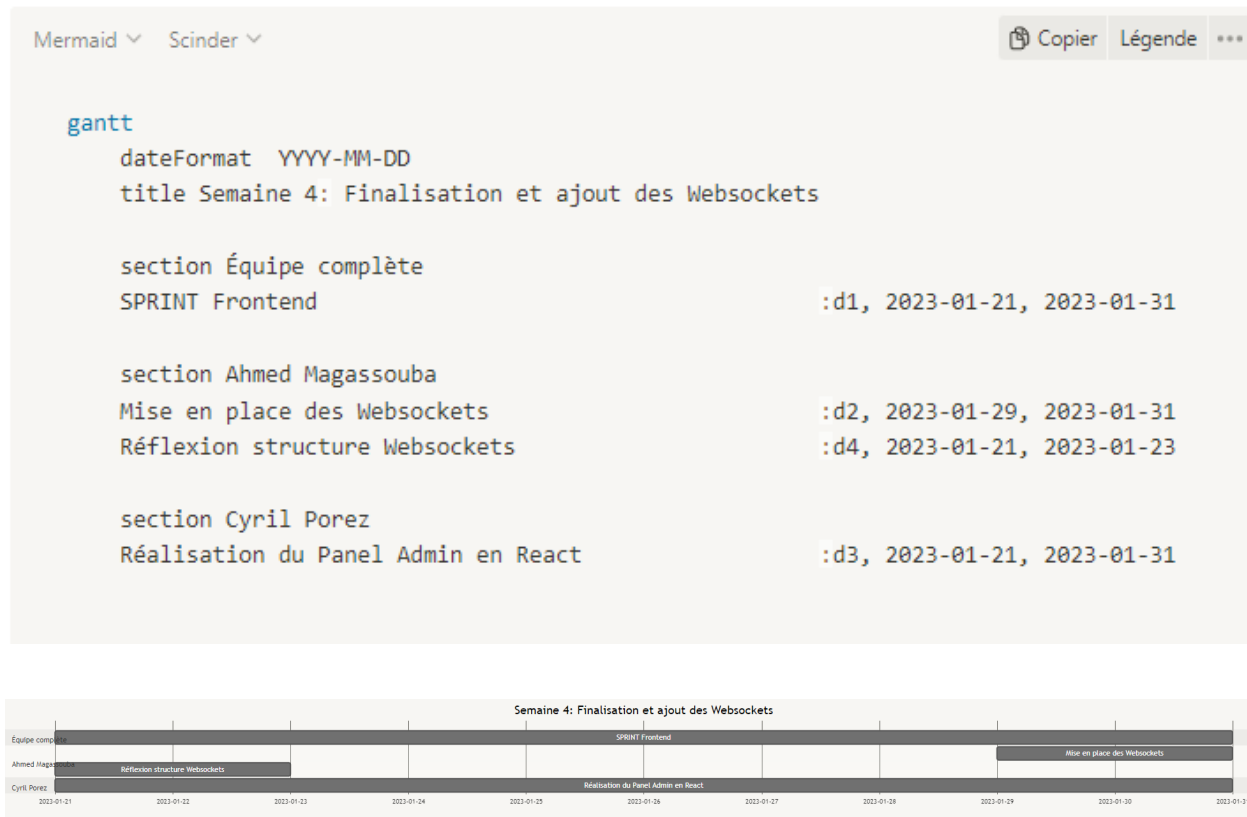
section Équipe complète
Sprint Frontend                                     :c1, 2023-01-13, 2023-01-21

section Boris TIKHOMIROFF
Mise en place de Redux                               :c2, 2023-01-13, 2023-01-15
Mise en place de l'architecture stack                 :c3, 2023-01-13, 2023-01-15

section Cyril Porez
Mise en place storyBook                             :c3, 2023-01-13, 2023-01-15
```



- La **quatrième semaine** est réservée à la finalisation du projet et à l'ajout des websockets.



Dans l'ensemble, le projet suit une approche agile, où chaque membre de l'équipe a des tâches spécifiques à accomplir, conformément à la planification établie.

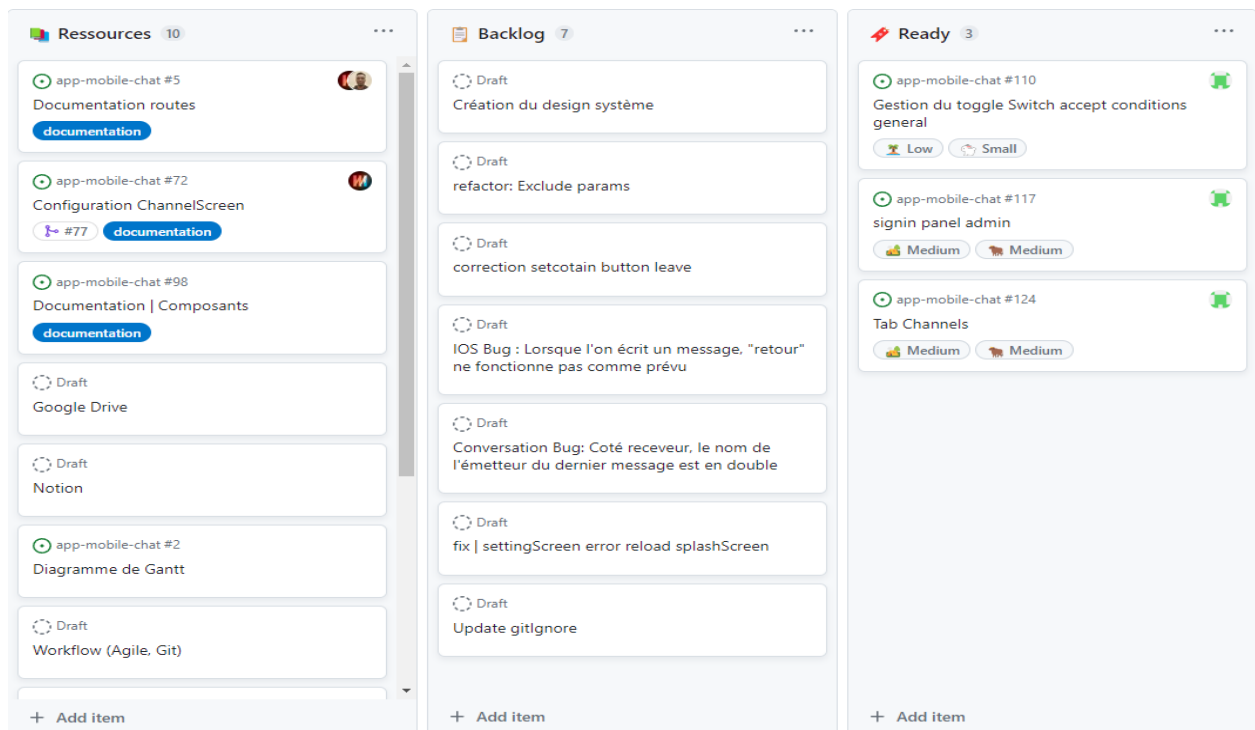
2. Rôles et responsabilités

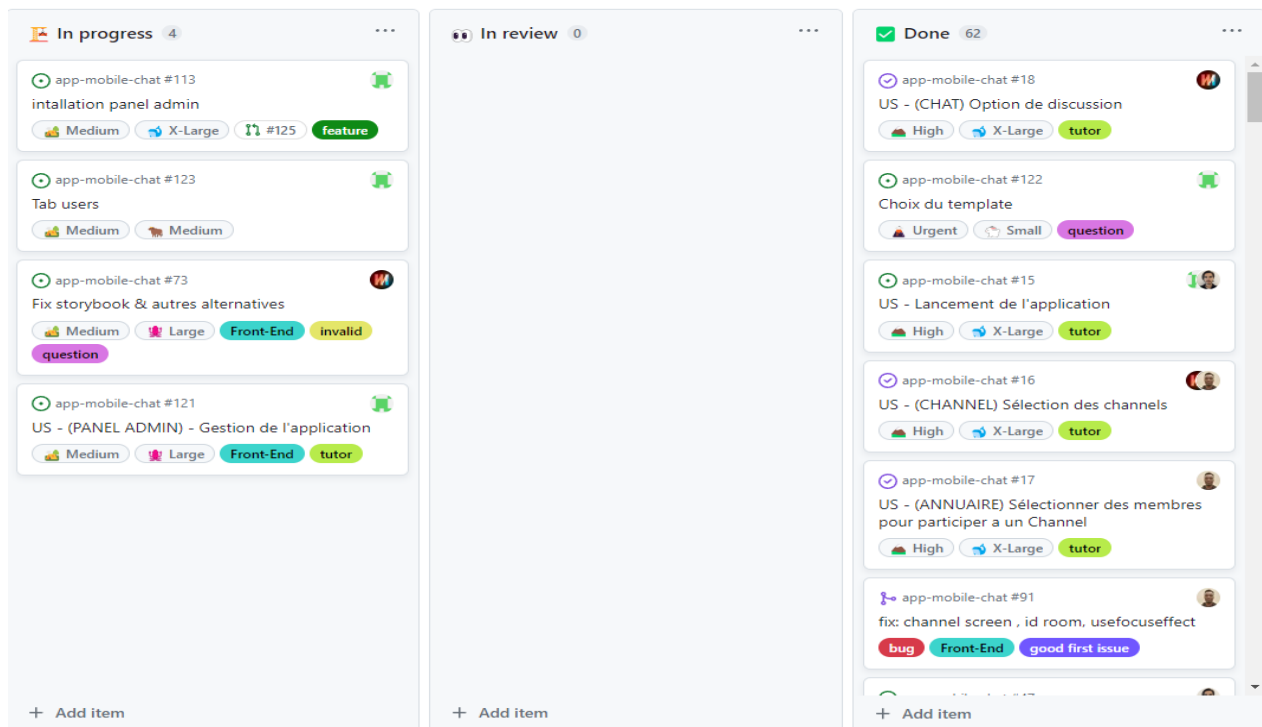
- @Johan Bouguermouh & @Ahmed Magassouba seront en charge de la conception et de la réalisation de la maquette de l'application WireFrame
- @Boris TIKHOMIROFF & @Cyril Porez seront en charge de la modélisation de la base de données et de sa mise à jour en cas de changements

- @Boris TIKHOMIROFF, @Cyril Porez, @Ahmed Magassouba et @Johan Bouguermouh seront en charge du développement de l'application mobile avec React Native
- @Boris TIKHOMIROFF, @Cyril Porez, @Ahmed Magassouba et @Johan Bouguermouh seront en charge du développement du back-end avec Node.js
- @Boris TIKHOMIROFF, @Cyril Porez, @Ahmed Magassouba et @Johan Bouguermouh seront successivement en charge de la gestion de projet et de la coordination des différentes tâches avec Trello et les diagrammes de Gantt
- XX sera en charge de la gestion de version avec Git et GitHub

3. GitHub Kanban

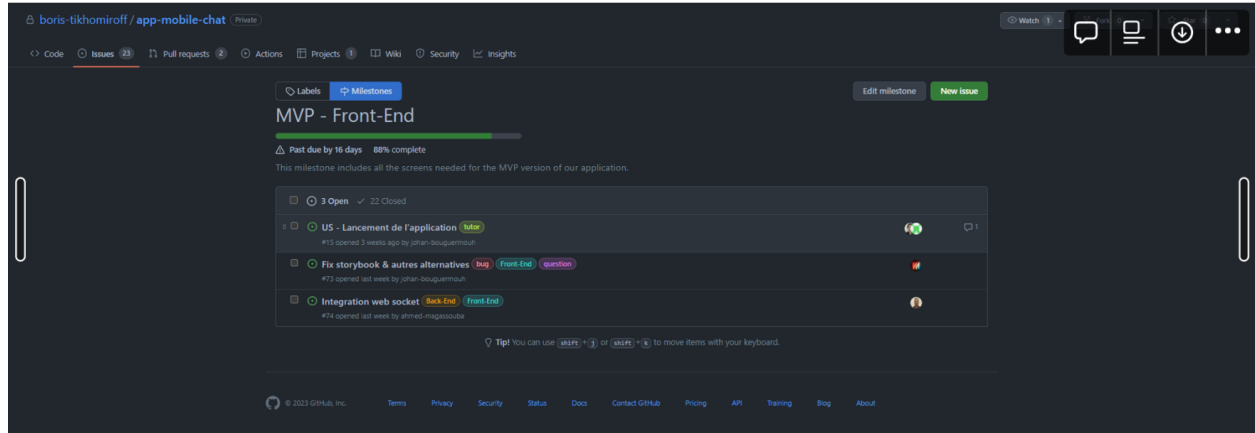
J' ai utilisé le GitHub Kanban afin de lister les tâches à effectuées.





Utilisation des Milestone (jalons)

Un milestone (jalon) sur GitHub est un outil de gestion de projet qui permet de regrouper les tâches ou les objectifs en fonction de leur échéance. Il sert à organiser les tâches à réaliser dans un projet en les regroupant par étape ou par échéance. Les utilisateurs peuvent également utiliser les milestones pour suivre les progrès d'un projet et identifier les tâches qui doivent être achevées avant une date spécifique. En utilisant Github, les milestones sont créés sur les repository, et peuvent être associés à des issues pour les regrouper.



Utilisation des Labels

Les labels sur GitHub sont des étiquettes utilisées pour classer les tâches ou les demandes de fonctionnalités dans un projet. Ils permettent de catégoriser les tâches en fonction de leur statut, de leur priorité ou de leur type. Les utilisateurs peuvent créer des labels personnalisés en fonction des besoins de leur projet et les appliquer aux tâches ou aux demandes de fonctionnalités pour les organiser de manière efficace. Les labels permettent de filtrer les tâches et de les regrouper selon des critères spécifiques facilitant ainsi la gestion de projet.

Les différents type de label utilisé dans notre méthodes :

i tutors


Le Labels tutors définis une issue comme étant un parentes a un ensemble de features à apportées. Il est souvent liées a un milestones.


i documentation

Le label documentations défini une issue destinées seulement offrir un ensemble de données étant utile aux développeurs. Elle rassemble des éléments centraux permettant a ces derniers d'y apposé des grande modification ou des changement majeurs devant être consulté par les autres.


i question


Le label question est dédié au discussions de groupes sur un sujet précis. Les sujets peuvent êtres, l'utilisations d'une bibliothèque, l'implémentation d'une nouvelle fonctionnalité ou encore la révision d'étapes organisée. Elle servent a concilié la volonté de chacun sur une vision commune de manière proactive.

 Back-End
Issues en relations avec le dossier
back


 Front-End
Issue en relation avec le dossier
front

 Feature
L'issue est associé a une amélioration

 Bug
Lier a un fix

 Duplicate
L'issue est un doublon et a été délaissé au profit d'une autre.
Lors de l'utilisation de duplicate on lie la nouvelle issue a celle-ci

- # [n°issue]

 Help Wanted
L'issue est associé a une amélioration L'issue est en attente d'informations supplémentaire

Evaluation des priorités

Lors du choix d'une issue dans le **ready** ou la concertation dans le **backlog**, nous priorisons du plus élevé au moins urgent.



Urgent



High



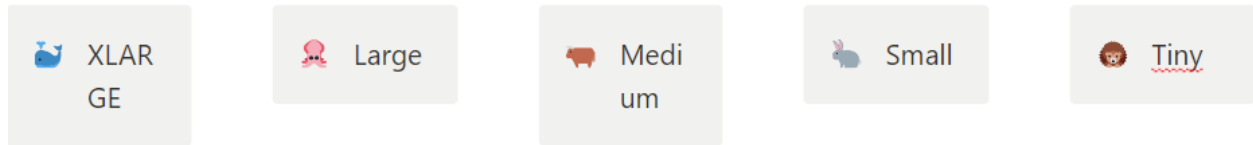
Medium



Low

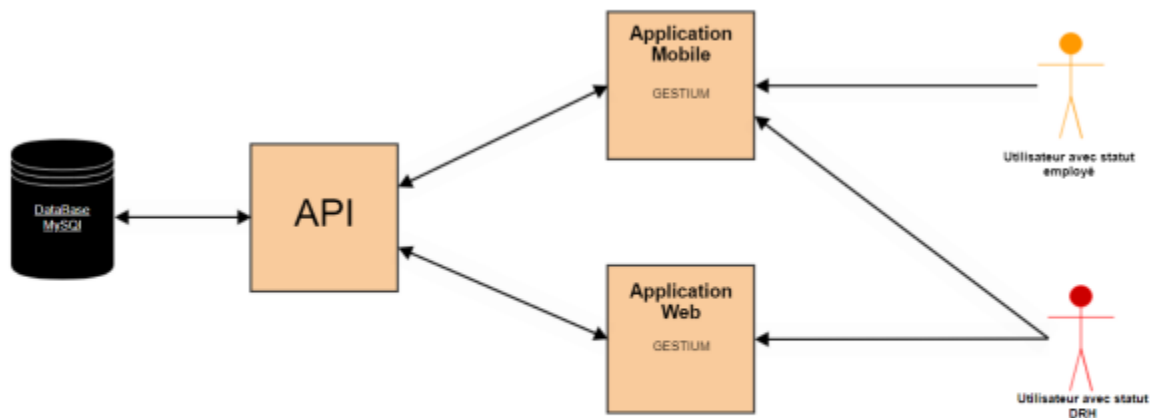
Estimation de la vélocité

A la récupération d'une issue nous évaluons la vélocité de la tâche.



Architecture logicielle

Les utilisateurs qui ont un statut **USER** auront uniquement accès à l'application mobile. Les utilisateurs avec un statut **ADMIN** auront accès à la fois à l'application mobile et au site web. L'application mobile et le site web utilisent la même API. L'API communique avec ma base de données MySQL.

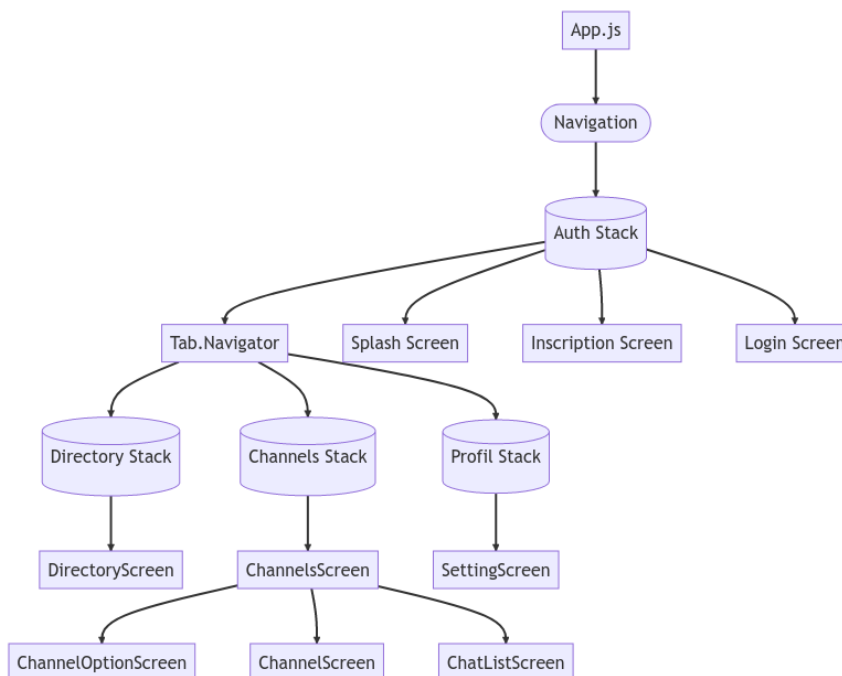


Schémas de l'architecture logiciel

CONCEPTION FRONT-END

Nous avons choisi Figma pour créer la maquette et la charte graphique en raison de sa capacité à faciliter la collaboration en temps réel, de sa facilité d'utilisation et de son accessibilité multiplateforme. Grâce à ses fonctionnalités avancées de conception responsive, de bibliothèques de composants et de styles réutilisables, ainsi que de prototypage interactif, Figma permet de créer des interfaces cohérentes, de simuler l'expérience utilisateur et d'obtenir des commentaires précieux avant le développement. En résumé, Figma offre une solution complète pour concevoir et collaborer efficacement tout au long du processus de création d'une application.

Arborescence du projet



Charte graphique

Notre charte graphique définit les principes visuels et les éléments graphiques qui donneront vie à notre application. Elle guide l'utilisation de couleurs, de typographies et de motifs, créant ainsi une identité visuelle cohérente et attrayante. Cette charte graphique reflète notre vision et nos valeurs, et servira de référence essentielle pour assurer une expérience utilisateur harmonieuse et mémorable



Charte Graphique Sent

Couleurs



#00C2FF



#A8B500



#5B1B9B



#FFA022



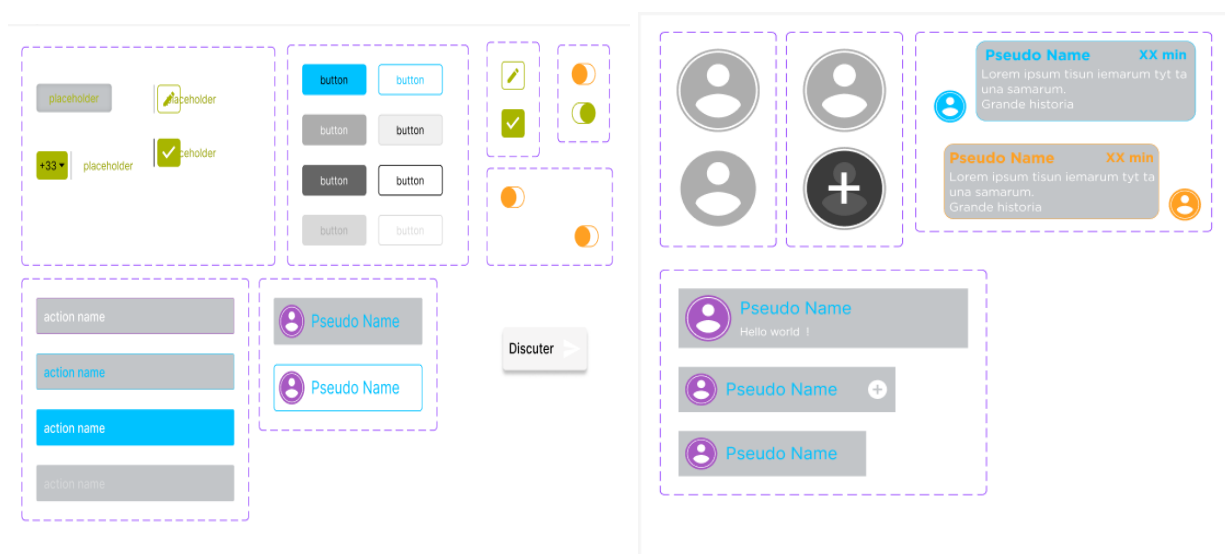
#000000

Polices

ROBOTO

Design System

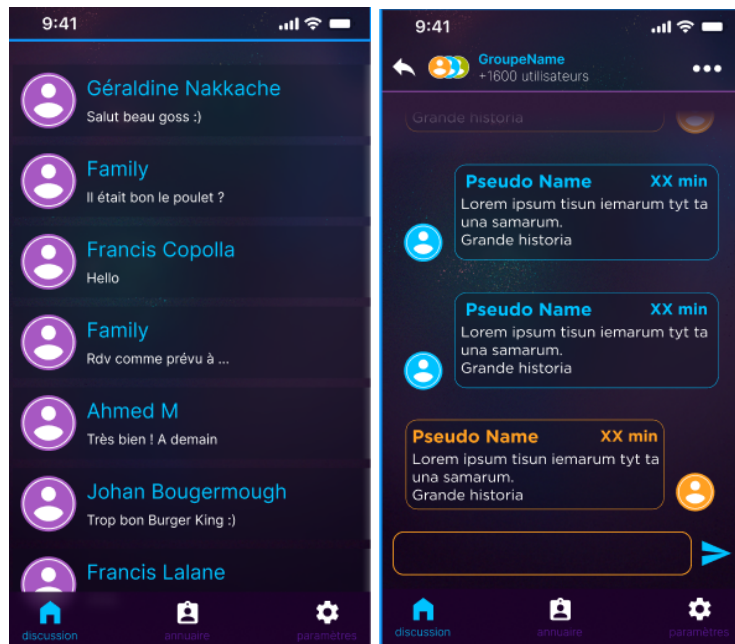
Notre design système est un ensemble cohérent de composants, de règles et de directives qui définit l'apparence et l'interaction de notre application. Il permet de créer une expérience utilisateur harmonieuse et conviviale en favorisant la réutilisation des éléments de conception et en garantissant une cohérence visuelle à travers l'ensemble de l'application. Le design système facilite la collaboration entre les membres de l'équipe, accélère le processus de conception et renforce l'identité visuelle de notre marque



*** Veuillez vous référer à l'annexe de ce dossier pour la totalité du design system.**

Maquettage

Les maquettes ont été créées à l'aide de Figma .



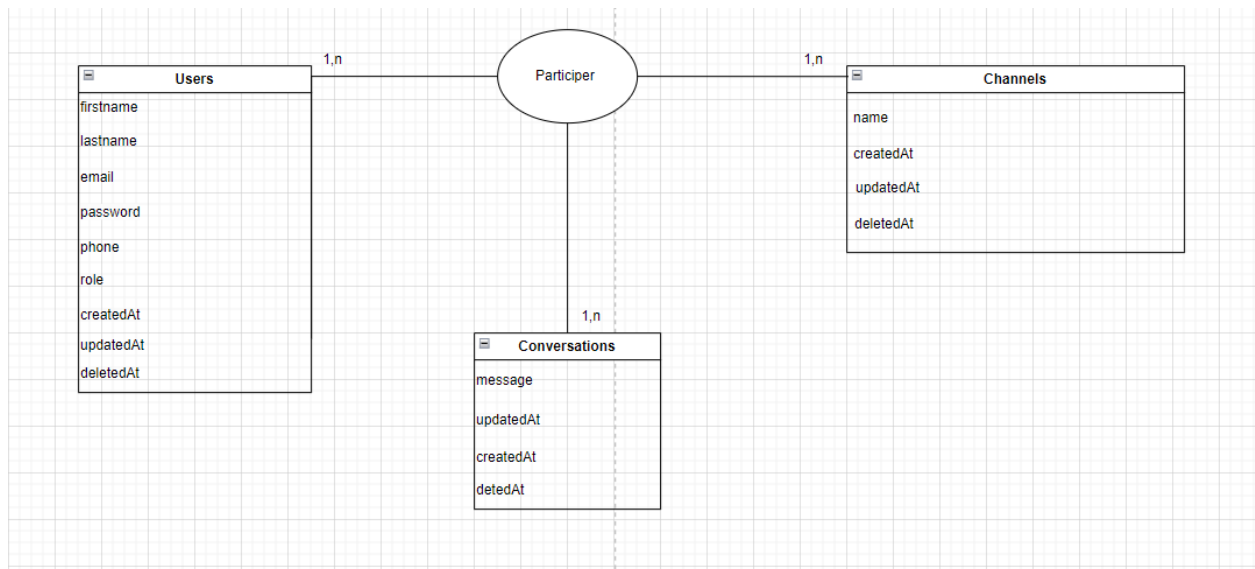
*** Veuillez vous référer à l'annexe de ce dossier pour la totalité du maquettage.**

Conception backend de l'application

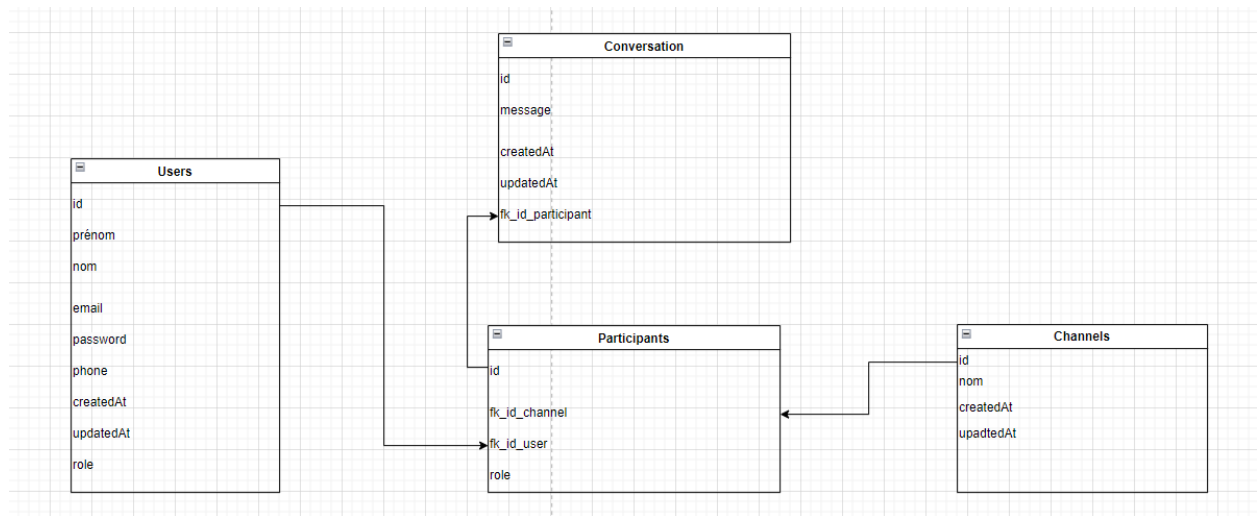
La base de données

Modèle Conceptuel de donnée (MCD)

Dans le cadre de la conception de notre application de chat, nous avons élaboré un modèle conceptuel de données (MCD) qui reflète la structure et les relations entre les différentes entités de notre système. Le MCD constitue un schéma essentiel qui nous permet de visualiser et de comprendre la logique de notre base de données. Il est composé de trois entités et du verbe de liaison, qui joue un rôle crucial dans la gestion des relations entre les utilisateurs, les canaux de conversation et les conversations elles-mêmes.



Modèle Logique de donnée (MLD)



Modèle Physique de données

Pour cette dernière étape, j'ai analysé les relations entre mes tables. Dans ce projet, je n'ai pas de cardinalité de type N/N, je n'ai donc pas de table de liaison. J'ai ensuite créé les clés étrangères. La clé étrangère est une contrainte qui garantit l'intégrité référentielle entre deux tables. J'ai ensuite typé mes champs.

