

Document de conception du projet TPT

TROCODEAL

Août 2024

Membres du groupe :

- INALIARIJAONA Ony Mirana - ETU 1097
- RAKOTOARIMANANA Manoa Iharivola - ETU 1130
- RANDRIAMAHEFA Miantsofiana Fanantenana - ETU 1167
- RAVELOJAONA Lafatra Ravaka Ftitia - ETU 1192

1. Introduction

1.1. Objectif du Document

Ce document de conception détaille l'architecture, les diagrammes UML, les interfaces utilisateur (IHM), et la base de données pour le projet de création d'un site d'échange d'objets entre particuliers, ainsi que son application mobile et client lourd.

1.2. Portée

Le projet vise à développer un système complet d'échange d'objets, comprenant un site web (front office et back office), une application mobile, et une application client lourd. Le système permet aux utilisateurs de proposer et d'accepter des échanges d'objets sans transactions financières.

2. Diagrammes UML

2.1. Diagramme de Cas d'Utilisation

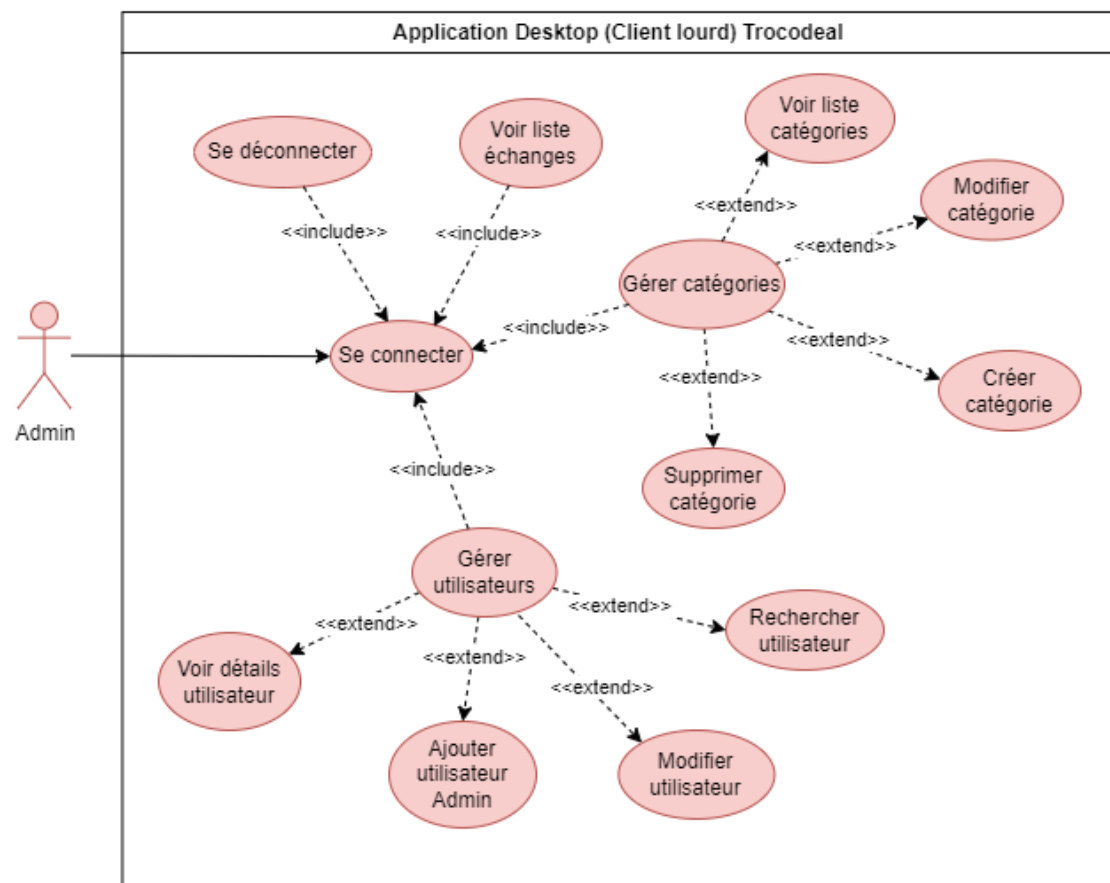
Les diagrammes de cas d'utilisation ci-dessous décrivent les principales interactions entre les acteurs et le système pour les différentes interfaces de votre projet : le site web Trocodeal, l'application desktop, et l'application mobile.

Site Web Trocodeal

Acteurs :

- **Utilisateur** : Accède aux fonctionnalités liées à la gestion des objets, aux échanges, et à la navigation sur le site.
- **Admin** : Gère les catégories, les utilisateurs, et supervise les échanges via le dashboard.

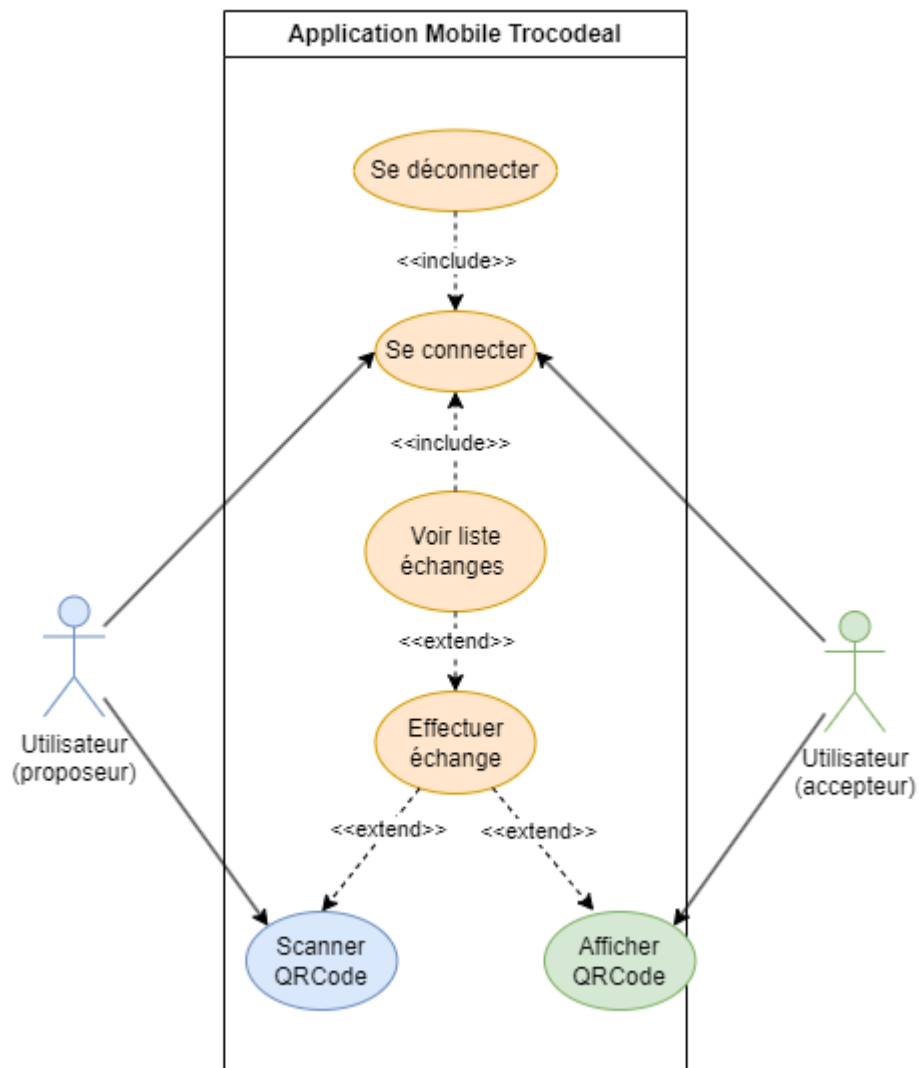
Application Desktop Trocodeal



Acteur :

- **Admin** : Accède aux fonctionnalités de gestion des catégories et des utilisateurs, et à la supervision des échanges.

Application Mobile Trocodeal

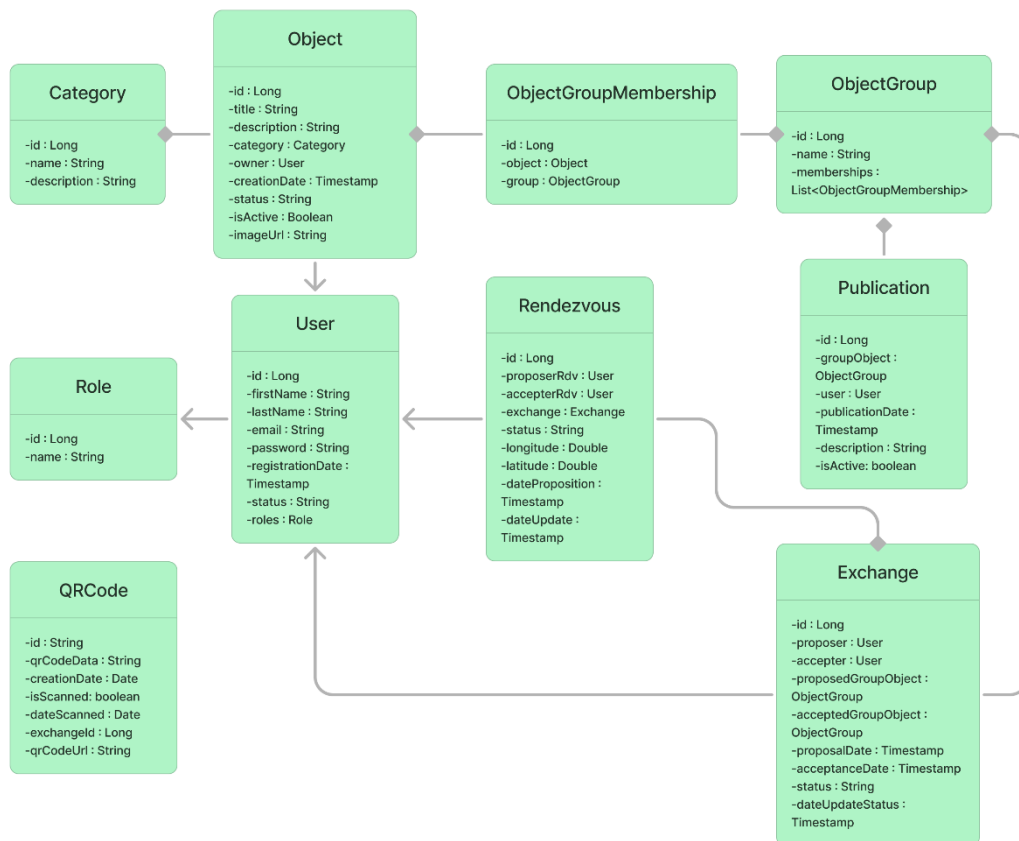


Acteurs :

- **Utilisateur (proposeur)** : Effectue des échanges en proposant des objets.
- **Utilisateur (accepteur)** : Accepte les échanges proposés.

2.2. Diagramme de Classes

Le diagramme de classes ci-dessous modélise les principales entités du système Trocodeal ainsi que leurs relations. Ce diagramme est essentiel pour comprendre la structure des données et les interactions entre les différentes classes du système.



2.3. Diagramme de Séquence

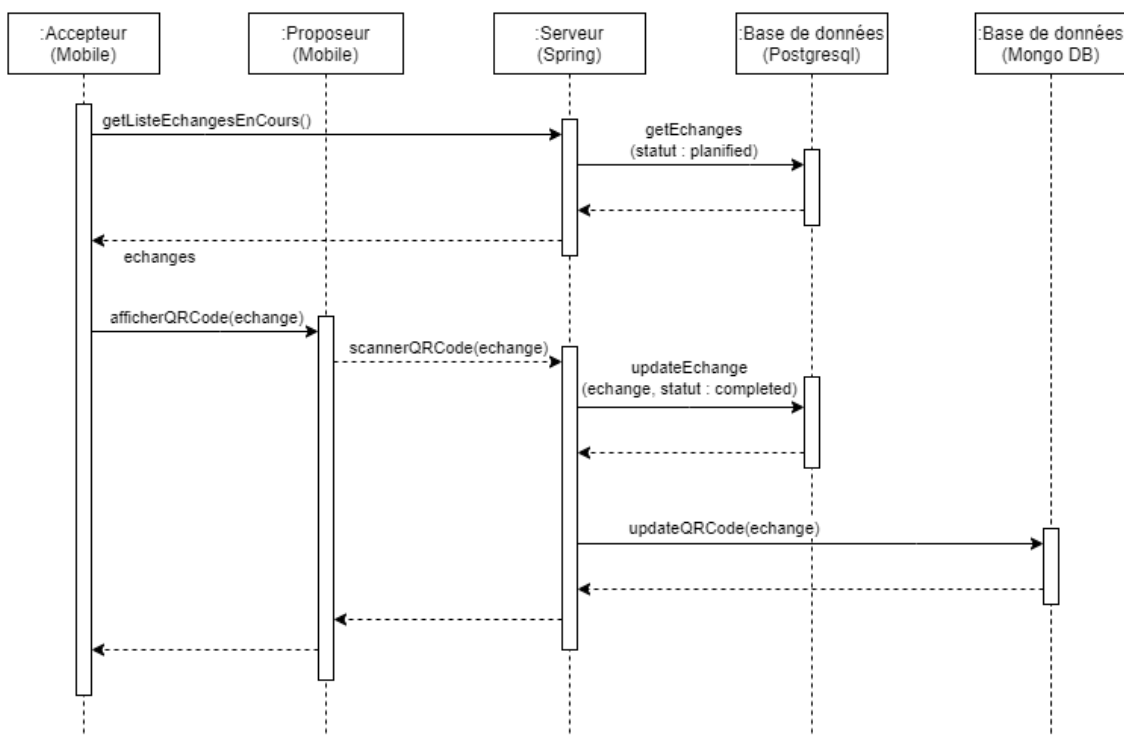
Les diagrammes de séquence suivants décrivent les interactions entre les différents acteurs (utilisateurs, système, bases de données) pour diverses opérations dans le système Trocodeal. Ces diagrammes permettent de visualiser le flux des messages et l'ordre d'exécution des actions pour chaque scénario.

Scénario 1 : Validation d'un Échange

Ce diagramme montre les interactions entre l'accepteur, le proposeur, le serveur (Spring), et les bases de données (PostgreSQL et MongoDB) pour valider un échange via QR code.

Étapes :

1. L'accepteur récupère la liste des échanges en cours.
2. L'accepteur affiche le QR code associé à l'échange.
3. Le proposeur scanne le QR code pour valider l'échange.
4. Le serveur met à jour l'état de l'échange et du QR code dans les bases de données.

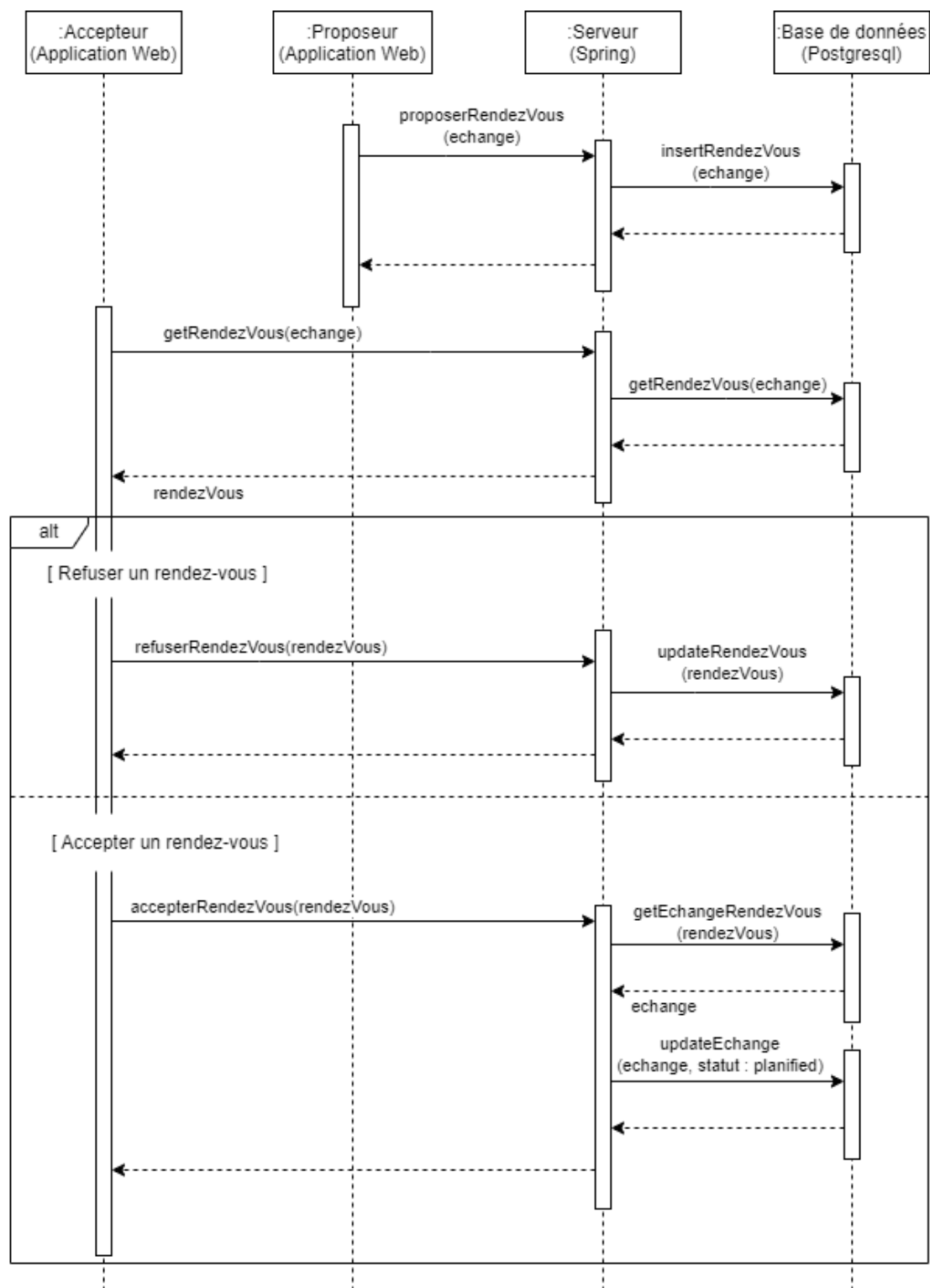


Scénario 2 : Gestion d'un Rendez-vous

Ce diagramme montre le processus de proposition, d'acceptation ou de refus d'un rendez-vous entre le proposeur et l'accepteur, avec la mise à jour des informations dans la base de données.

Étapes :

1. Le proposeur propose un rendez-vous.
2. Le serveur insère le rendez-vous dans la base de données.
3. L'accepteur consulte la proposition de rendez-vous.
4. Selon la décision de l'accepteur (accepter ou refuser), le serveur met à jour le statut du rendez-vous et de l'échange.

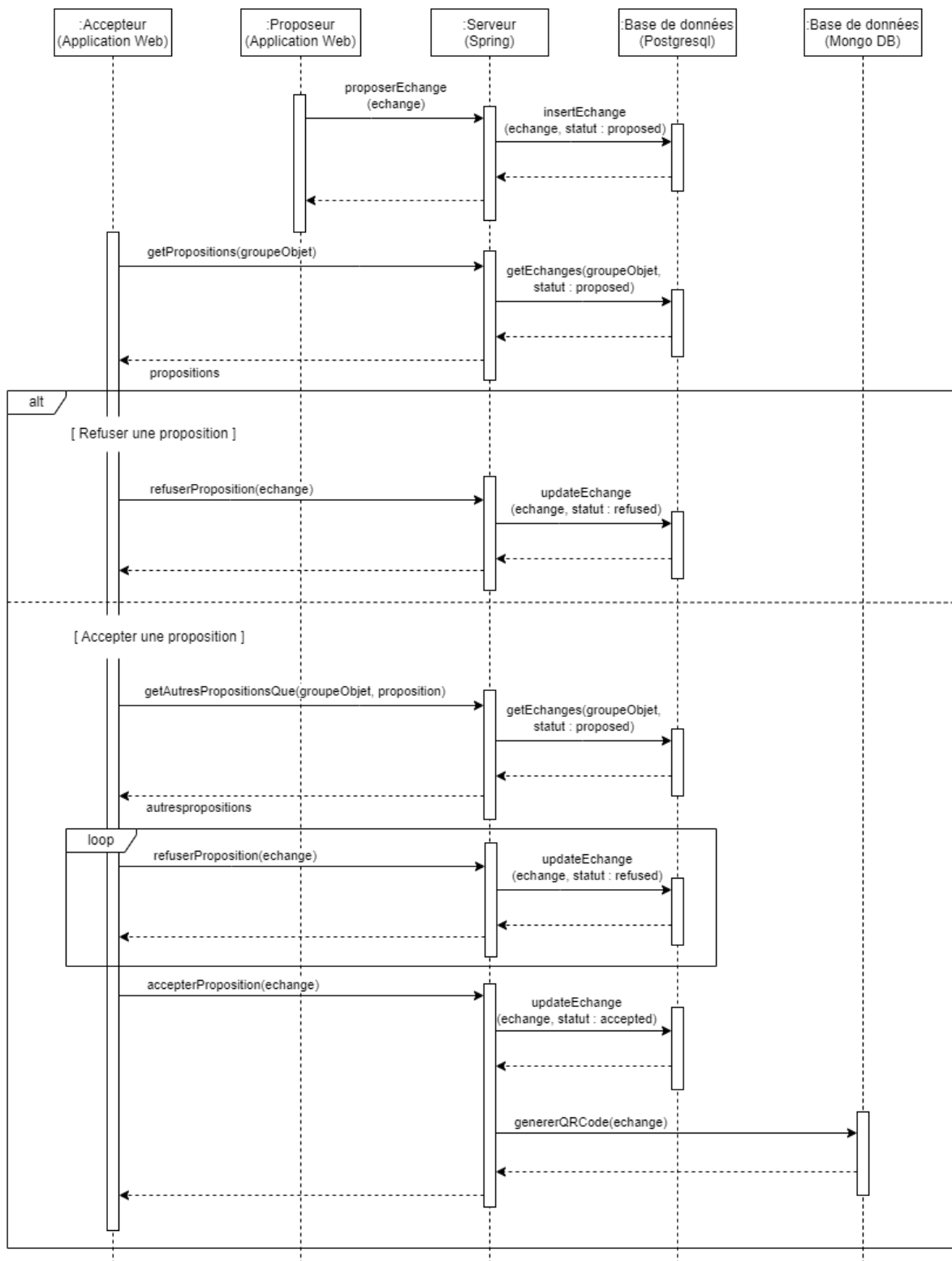


Scénario 3 : Proposition et Validation d'un Échange

Ce diagramme décrit le processus de proposition d'échange d'un objet et les interactions qui suivent, incluant l'acceptation ou le refus de la proposition.

Étapes :

1. Le proposeur soumet une proposition d'échange.
2. Le serveur enregistre la proposition dans la base de données.
3. L'accepteur consulte la proposition et décide de l'accepter ou de la refuser.
4. Le serveur met à jour le statut de l'échange en fonction de la réponse de l'accepteur.



3. Interface Homme-Machine (IHM)

3.1. Wireframes / Maquettes

Les maquettes suivantes ont été créées sur Figma pour illustrer l'interface utilisateur des différentes sections de l'application Trocodeal. Elles fournissent une représentation visuelle des concepts décrits précédemment dans cette section.

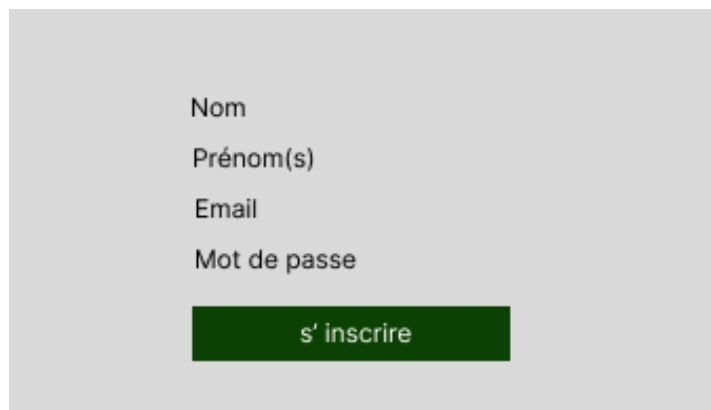
3.1.1. Application Web (Site Trocodeal)

- **Page d'inscription et de login**

Objectif : Permettre aux utilisateurs de créer un compte ou de se connecter à leur compte existant.

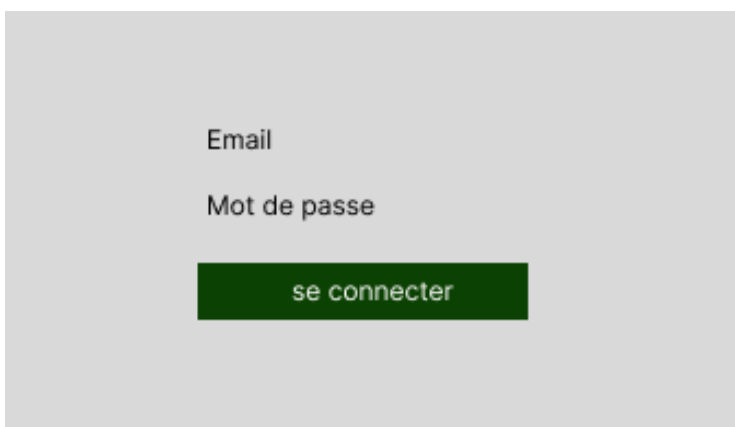
- **Composants :**

- **Formulaire d'inscription :**



A wireframe of a registration form on a light gray background. It features four text input fields stacked vertically, labeled "Nom", "Prénom(s)", "Email", and "Mot de passe". Below these fields is a dark green rectangular button with the white text "s'inscrire".

- **Formulaire de login :**



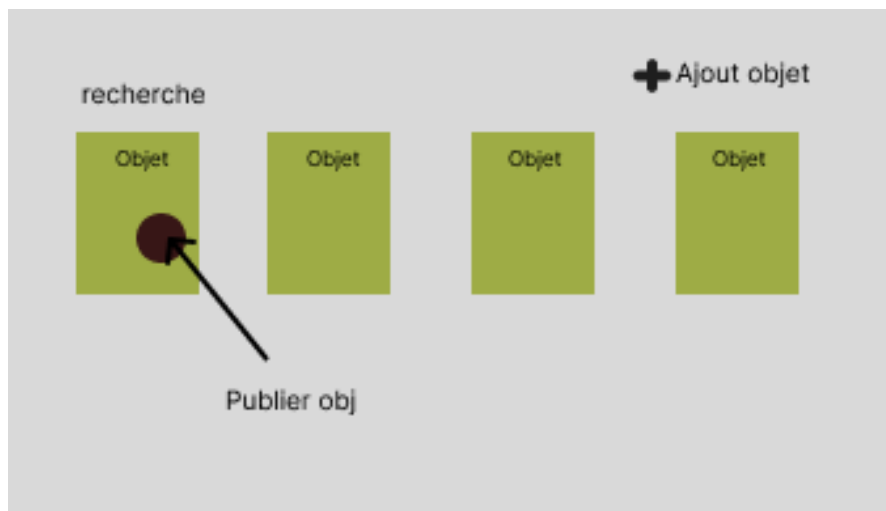
A wireframe of a login form on a light gray background. It features two text input fields stacked vertically, labeled "Email" and "Mot de passe". Below these fields is a dark green rectangular button with the white text "se connecter".

- **Actions :**

- Validation du formulaire d'inscription, avec un message d'erreur en cas de champs manquants ou de mot de passe non conforme.
- Validation du login, redirection vers la page d'accueil en cas de succès, ou affichage d'un message d'erreur en cas d'échec.

- **3.1.2. Page de détail d'objet et publication**

- Objectif : Afficher les informations détaillées d'un objet et permettre aux utilisateurs de publier un objet qu'ils souhaitent échanger.



- **Actions :**

- Lorsqu'un utilisateur clique sur "Proposer un échange", un formulaire de proposition est affiché.
- Le bouton "Publier l'objet" permet de mettre l'objet en ligne et de le rendre disponible pour l'échange.

- **Page de gestion des objets**

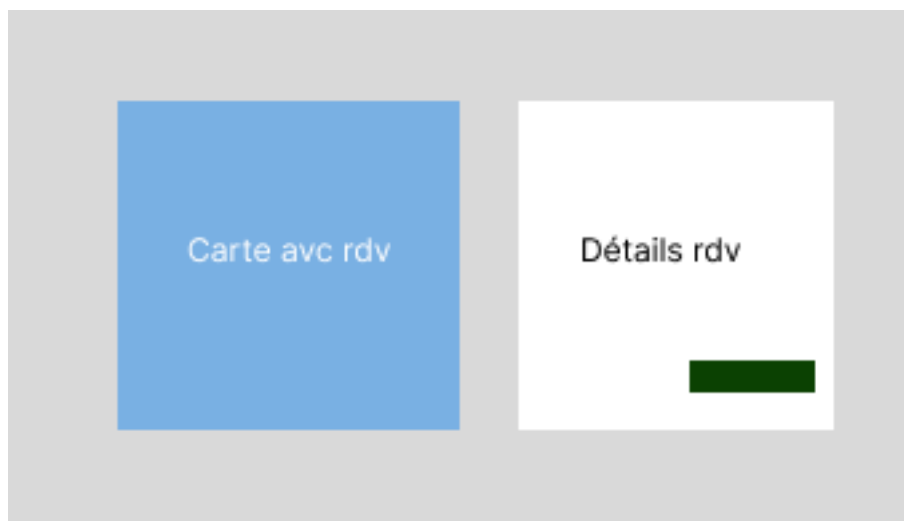
- Objectif : Permettre aux utilisateurs de gérer les objets qu'ils ont ajoutés, avec des options pour modifier, archiver ou supprimer ces objets.

- **Actions :**

- L'utilisateur peut modifier les informations d'un objet via le formulaire de modification.
- L'archivage d'un objet le rend indisponible pour les autres utilisateurs.
- La suppression d'un objet le retire définitivement de la plateforme.

- **Page de gestion des rendez-vous**

- Objectif : Permettre aux utilisateurs de fixer un rendez-vous pour finaliser l'échange d'objets en personne.
- Composants :
 - **Géolocalisation :**
 - Carte intégrée pour choisir un lieu de rendez-vous.
 - **Détails du rendez-vous :**
 - Affichage des coordonnées des deux parties (Nom, Email).
 - Option pour annuler ou reprogrammer le rendez-vous.
 - Bouton "Confirmer le rendez-vous".



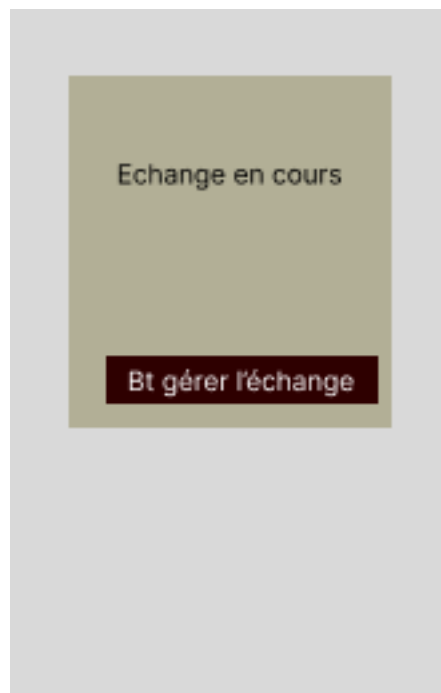
- **Actions :**
 - Sélectionner une date, une heure, et un lieu pour le rendez-vous.
 - Confirmer, annuler ou reprogrammer le rendez-vous.

3.1.2. Application Mobile

- **Page de connexion :**

- Composants : Formulaire de connexion avec champs pour l'email et le mot de passe, bouton de connexion.
- Actions : Connexion de l'utilisateur, récupération de mot de passe.

- **Page de liste des échanges en cours :**



- Composants : Liste des échanges en cours, statut de chaque échange, option pour afficher le QR code.
- Actions : Visualisation des échanges, affichage et scan de QR code.

- **Page de validation d'échange :**

- Composants : QR code pour l'échange, bouton pour confirmer ou annuler l'échange.
- Actions : Validation ou annulation de l'échange via QR code.

3.1.3. Application Desktop

- **Page de gestion des catégories :**



- Composants : Liste des catégories, options pour ajouter, modifier, ou supprimer une catégorie.
 - Actions : Ajout, modification, suppression des catégories d'objets.
- **Page de gestion des utilisateurs :**
 - Composants : Liste des utilisateurs avec rôles, options pour ajouter un utilisateur, modifier ses informations.
 - Actions : Ajout de nouveaux utilisateurs (admins), modification des informations d'un utilisateur existant.

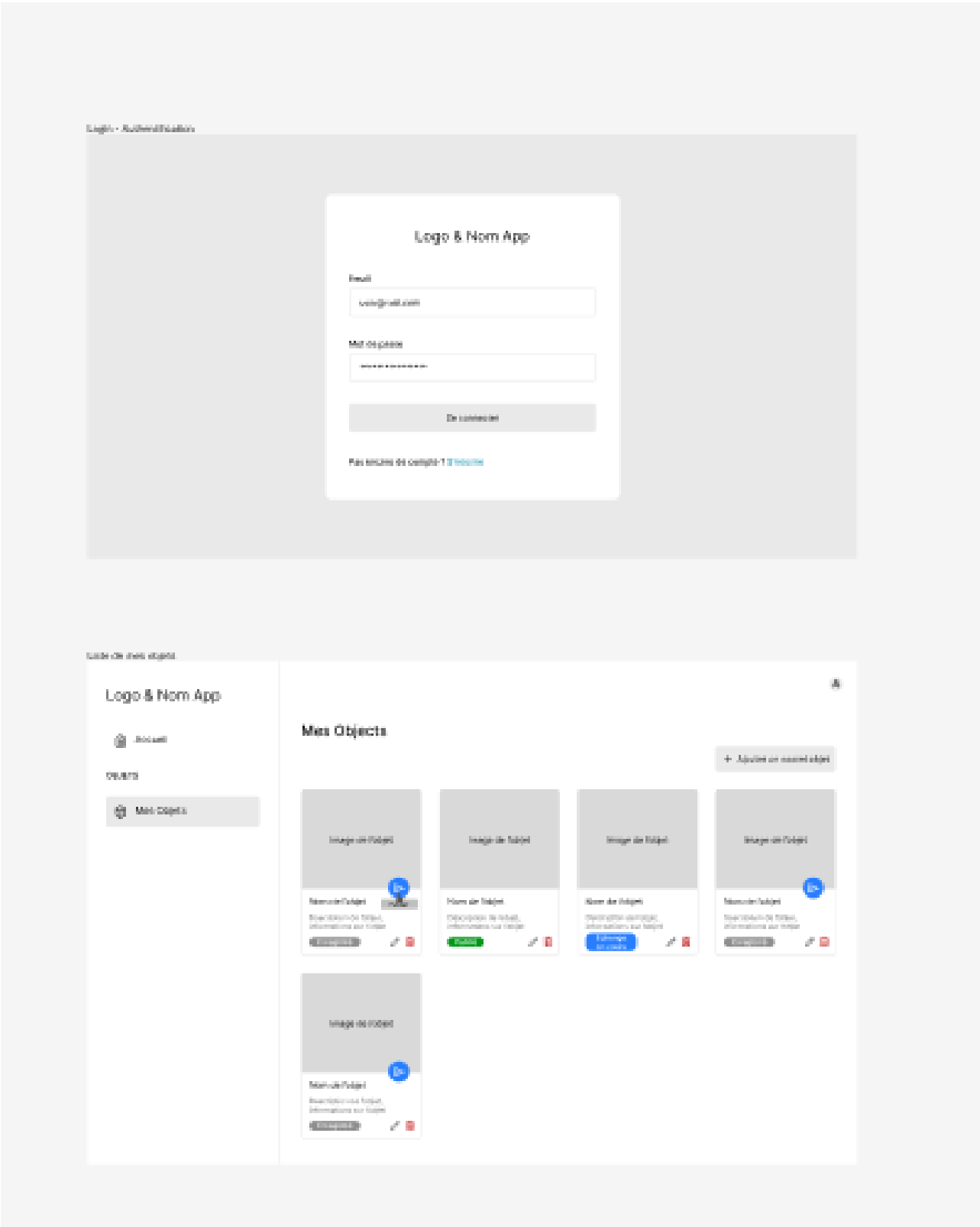
3.2. Diagramme de Navigation

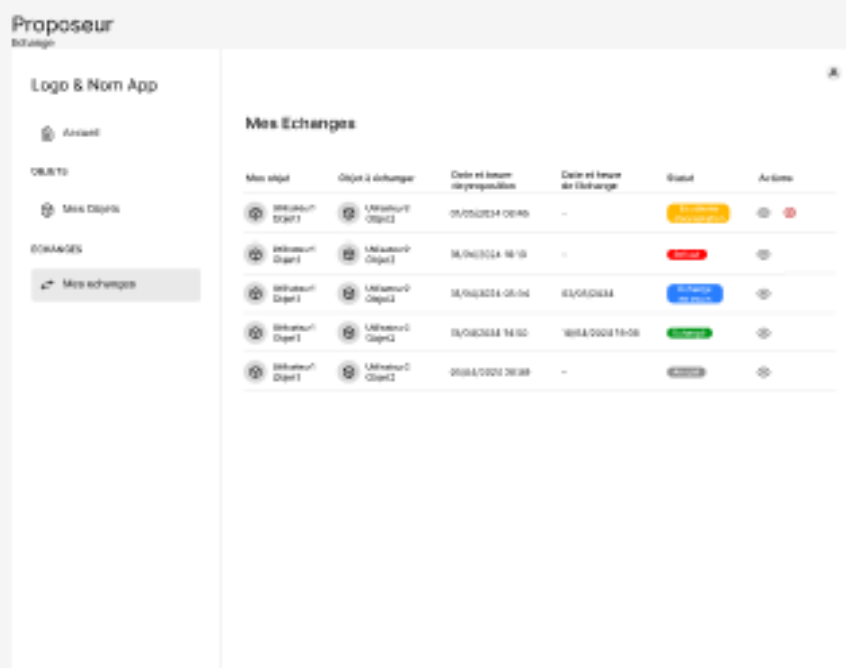
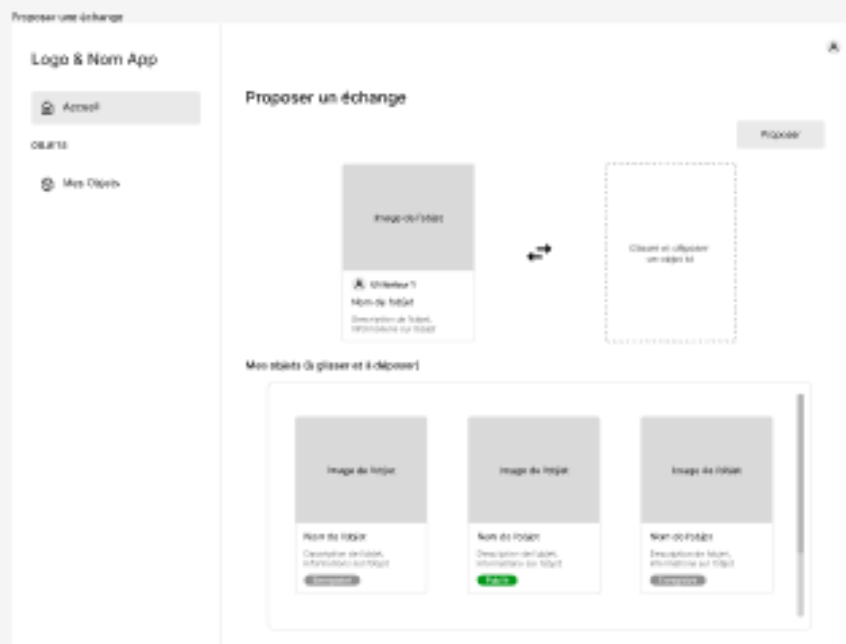
Les pages d'inscription et de login, de détail d'objet, de gestion des objets, de proposition d'échange, de négociation, et de gestion des rendez-vous sont intégrées dans un flux de navigation cohérent à travers l'application. Voici une représentation textuelle de la navigation :

- **Inscription/Connexion → Page d'accueil → Recherche d'objets → Détail d'objet → Proposition d'échange → Négociation → Rendez-vous.**

3.3. Prototypes

Les maquettes suivantes ont été créées sur Figma pour illustrer l'interface utilisateur des différentes sections de l'application Trocodeal. Elles fournissent une représentation visuelle des concepts décrits précédemment dans cette section.

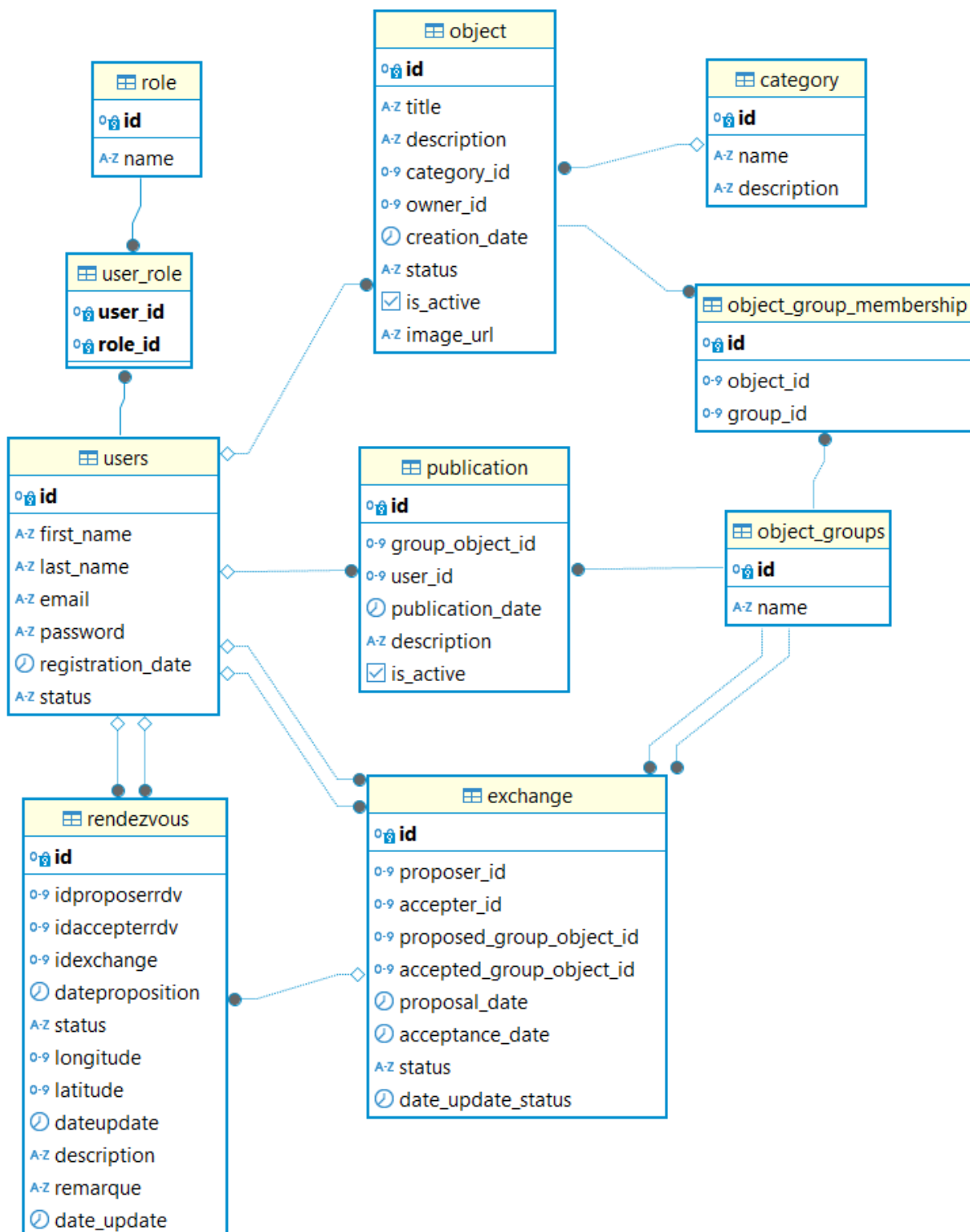




4. Base de Données

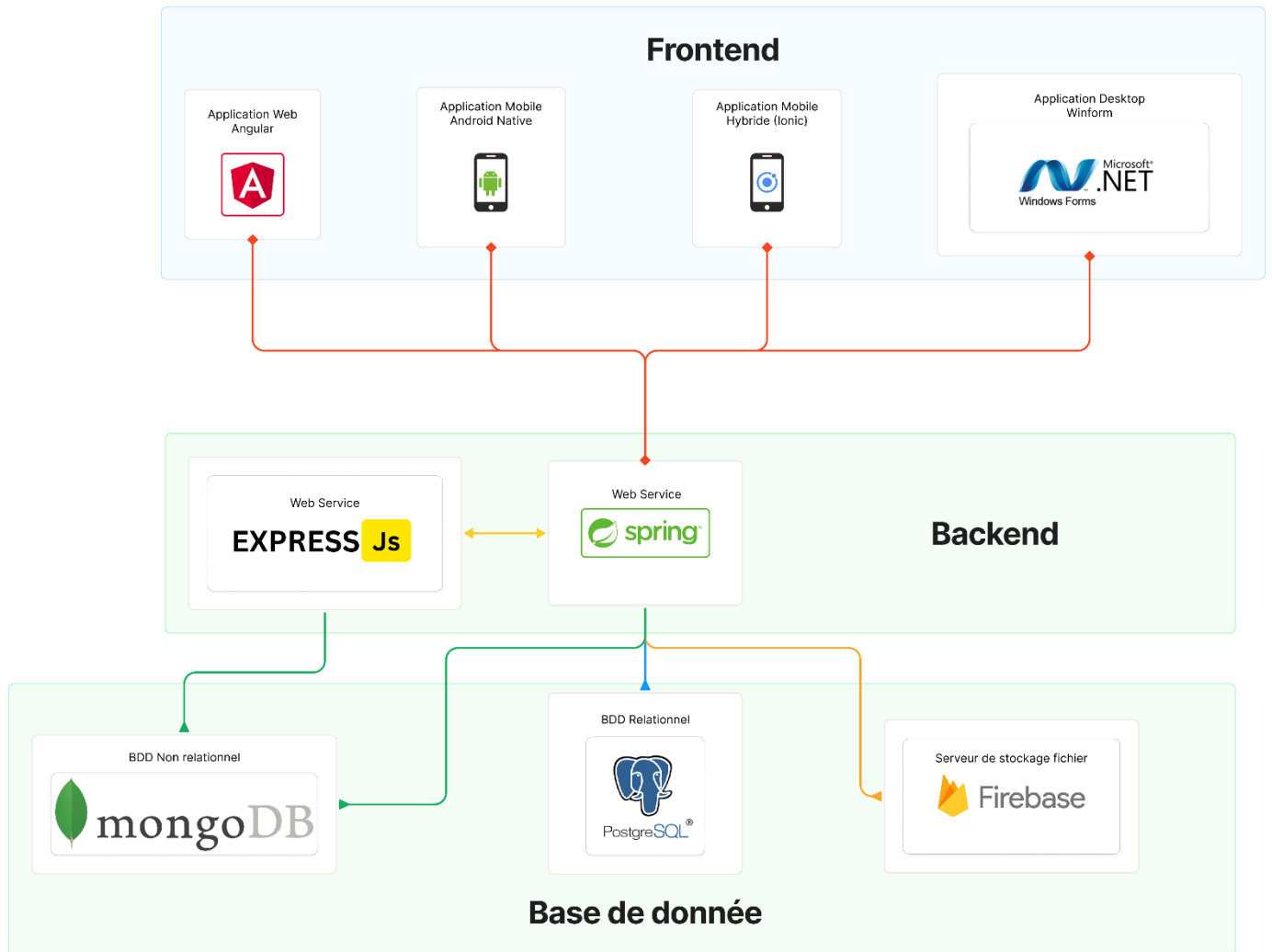
Modèle Conceptuel de Données (MCD)

Le MCD suivant représente les entités, leurs attributs principaux, et les relations entre elles pour le projet Trocodeal. Ce modèle est essentiel pour comprendre la structure des données et les interactions entre les différentes entités du système.



5. Architecture Globale

L'architecture du projet Trocodeal se compose de plusieurs couches, chacune ayant un rôle bien défini dans le fonctionnement global du système. Ce modèle en couches permet de séparer les responsabilités, ce qui facilite la maintenance, la scalabilité, et l'évolution du système.



Description des Composants :

1. Frontend :

- **Application Web Angular** : Fournit l'interface utilisateur principale pour les utilisateurs finaux. Développée avec Angular, cette application gère l'interaction utilisateur, la navigation, et les appels API vers le back-end.

- **Application Mobile Native (Android)** : Version native Android de l'application mobile, offrant des performances optimisées et un accès aux fonctionnalités natives du système Android.
- **Application Mobile Hybride (Ionic)** : Développée avec Ionic, cette application est déployable sur plusieurs plateformes (iOS, Android), offrant une expérience utilisateur uniforme.
- **Application Desktop (WinForms .NET)** : Application client lourd destinée à la gestion administrative, développée en WinForms pour une interface riche sur des environnements Windows.

2. Backend :

- **Web Service Spring Boot** : Gère les API RESTful pour la gestion des utilisateurs, des objets, des échanges, etc. Spring Boot est utilisé pour sa robustesse et ses capacités d'intégration.
- **Web Service Express.js** : Un autre service web utilisé pour des fonctionnalités spécifiques, interagissant principalement avec MongoDB.

3. Base de Données :

- **MongoDB** : Utilisée pour stocker des données non relationnelles, telles que les historiques d'échanges, les logs, ou d'autres données JSON.
- **PostgreSQL** : SGBD relationnel utilisé pour stocker les données structurées, telles que les utilisateurs, objets, catégories, et échanges.
- **Firebase** : Utilisé comme serveur de stockage de fichiers, par exemple pour stocker les images des objets, les fichiers partagés entre utilisateurs, etc.

Flux de Données :

1. Interaction Utilisateur :

- Les utilisateurs interagissent avec les interfaces utilisateur via des applications web, mobiles, ou desktop. Ces interactions déclenchent des requêtes HTTP/HTTPS vers les services web.

2. Traitement Serveur :

- Les services web (Spring Boot et Express.js) traitent les requêtes, interagissent avec les bases de données (PostgreSQL, MongoDB), et effectuent des opérations de stockage ou de récupération de données.

3. Stockage et Gestion des Données :

- **PostgreSQL** est utilisé pour les opérations CRUD sur les entités relationnelles.
- **MongoDB** gère les données documentaires et non structurées.
- **Firebase** assure le stockage des fichiers associés aux objets et autres données volumineuses.

