

# **CAHIER DES CHARGES**

## **Application de Troc**

RAHAMISON Fiaritia Naliarilala - N°16 - ETU 1258  
RAKOTOARIMALALA Rovantsoa Tsioriorinantenaina - N°18 - ETU 1370  
RAMILISON Solofoniaina Mendrika Fitahiana - N°35 - ETU 1302

# TABLES DE MATIERES

<b>1. Introduction</b>	<b>4</b>
1.1. Contexte	4
1.2. Objectif	4
1.3. Objectif spécifique	4
1.4. Description générale	4
<b>2. Infrastructure</b>	<b>5</b>
1.1. Architecture logicielle du système	5
1.1.1. Architecture desktop pour le Back Office (Administration)	5
1.1.2. Serveur d'application	6
1.1.3. Backend	6
1.1.4. Base de données	7
1.1.5. Front Office (Interface utilisateur)	8
1.1.6. Application mobile	9
1.1.7. Webservice REST	9
1.1.8. Module de cartographie	10
1.2. Choix technologiques	10
1.2.1. Plateforme de développement	10
1.2.2. Système de gestion de base de données (SGBD)	13
<b>2. Infrastructure et sécurité</b>	<b>14</b>
2.1. Sécurité	15
2.2. Gestion des versions	15
<b>3. Modules fonctionnels</b>	<b>15</b>
3.1. Gestion des utilisateurs et authentification	15
3.2. Gestion des annonces	16
3.3. Gestion des échanges d'objets	16
3.4. Service de mapping	16
3.5. Service des statistiques	16
<b>4. Fonctionnalités par profil</b>	<b>16</b>
4.1. Administrateur	17
4.2. Troqueur	17
<b>5. Spécifications détaillées des modules</b>	<b>17</b>
5.1. Gestion des utilisateurs et authentification	18
5.1.1. Métiers concernés	18
5.1.2. Descriptions fonctionnalités	18
5.2. Gestion des annonces	19
5.2.1. Création d'annonces:	19

5.2.2. Modification et suppression d'annonces:	19
5.2.3. Visualisation des Annonces Publiées:	19
5.2.4. Recherche et Filtrage des Annonces:	20
5.2.5. Consultation des Détails des Annonces:	20
5.2.6. Upload de Photos:	20
5.2.7. Masquer les Annonces Signalées:	20
5.2.8. Gestion des Catégories d'Objets:	20
5.3. Gestion des échanges d'objets	20
5.3.1. Métiers concernés	20
5.3.2. Descriptions fonctionnalités	21
5.4. Service de mapping	22
5.4.1. Affichage des Objets sur une Carte:	22
5.4.2. Recherche et Filtrage Géographique:	22
5.4.3. Détails de l'Objet:	22
5.4.4. Gestion des Localisations :	23
5.5. Service des statistiques	23
5.5.1. Nombre d'objets échangés:	23
5.5.2. Catégories Populaires d'Objets Échangés:	23
5.5.3. Répartition Géographique:	23
<b>6. Conclusion</b>	<b>25</b>

# 1. Introduction

## 1.1. Contexte

Dans un monde où la réutilisation et la réduction des déchets deviennent de plus en plus importantes, le troc d'objets entre particuliers gagne en popularité. Ce projet a pour but de créer une plateforme en ligne et une application mobile pour permettre aux gens d'échanger des objets sans utiliser d'argent.

## 1.2. Objectif

Le but de ce projet est de développer un site web et une application mobile où les utilisateurs peuvent échanger des objets. L'idée est de rendre ces échanges simples, sécurisés et accessibles à tous.

## 1.3. Objectif spécifique

- **Créer un site web convivial** : Une interface facile à utiliser pour que les utilisateurs puissent créer des annonces, gérer leurs objets, et effectuer des échanges.
- **Développer un back office** : Un espace de gestion pour les administrateurs du site afin de superviser les activités et maintenir l'ordre.
- **Lancer une application mobile** : Disponible pour Android et iOS, ainsi qu'une version pour Windows, permettant aux utilisateurs de gérer leurs échanges où qu'ils soient.
- **Utiliser des bases de données fiables**
- **Intégrer des technologies web modernes** : Java pour le serveur, JavaScript côté serveur, et des frameworks comme React, Angular ou VueJS pour le front-end.
- **Ajouter des fonctionnalités avancées** : Services REST, un module de cartographie pour localiser les objets, et des QR codes pour faciliter les échanges.

## 1.4. Description générale

Le site d'échange permettra aux utilisateurs de :

- **Créer un compte** pour accéder à toutes les fonctionnalités.
- **Ajouter des objets** qu'ils souhaitent échanger, avec des descriptions et des photos.
- **Proposer et accepter des échanges** d'objets avec d'autres utilisateurs.
- **Retirer des objets** de la liste d'échange si nécessaire.

L'application mobile offrira les mêmes fonctionnalités, avec des interfaces adaptées pour les appareils Android et iOS, et une version client lourd pour Windows. Les données seront synchronisées entre le site web, l'application sur Windows et l'application mobile pour une expérience utilisateur fluide.

## 2. Infrastructure

### 1.1. Architecture logicielle du système

Pour la réalisation du projet d'échange d'objets entre particuliers, nous nous appuierons sur une architecture solide et bien définie. Cette architecture inclut un serveur d'application, une base de données, des services web, des interfaces utilisateur variées, et des applications mobiles.

#### 1.1.1. Architecture desktop pour le Back Office (Administration)

Une application de bureau (client lourd) sera développée en C# avec Winforms pour servir de panneau d'administration du projet. Cette application permettra une gestion complète du contenu et des utilisateurs du site via des WebServices. Elle nécessitera une installation sur chaque ordinateur administrateur.

##### a. Avantages

- **Gestion centralisée et performante** : Facilite la gestion du contenu et des utilisateurs avec une interface desktop réactive.
- **Fonctionnalités natives** : Accès aux fonctionnalités natives de la plateforme, offrant plus de possibilités pour les tâches administratives complexes.
- **Sécurité renforcée** : L'accès limité aux ordinateurs administrateurs autorisés renforce la sécurité du back office.
- **Interface riche** : Possibilité de créer une interface utilisateur plus sophistiquée et interactive pour les tâches d'administration.

##### b. Inconvénients

- **Maintenance et compatibilité** : Des problèmes peuvent survenir lors des mises à jour de l'OS ou des changements de plateforme, nécessitant des révisions de l'application.

- **Mises à jour** : L'application desktop est plus difficile à mettre à jour comparé aux applications web, ce qui peut compliquer la maintenance du back office.
- **Mises à jour** : L'application desktop est plus difficile à mettre à jour comparé aux applications web, ce qui peut compliquer la maintenance du back office.
- **Limitation d'accès** : L'administration est limitée aux ordinateurs sur lesquels l'application est installée, réduisant la flexibilité d'accès.
- **Développement spécifique** : Nécessité de développer des versions différentes si l'administration doit être accessible sur plusieurs types de plateformes.

### 1.1.2. Serveur d'application

Le serveur d'application sera développé en Java, ce qui assure une robustesse et une scalabilité accrues. Il gèrera les services métiers, les API REST, ainsi que toute la logique applicative nécessaire au bon fonctionnement du site et des applications.

#### a. Avantages

- **Centralisation de la logique métier** : Facilite la gestion et la maintenance des règles métier.
- **Scalabilité** : Peut être mis à l'échelle pour gérer une forte charge de trafic.
- **Sécurité** : Centralisation des contrôles de sécurité et de l'authentification.

#### b. Inconvénients

- **Complexité** : Peut nécessiter une infrastructure complexe pour gérer la montée en charge.
- **Maintenance** : Les mises à jour peuvent affecter l'ensemble du système si elles ne sont pas bien gérées.

### 1.1.3. Backend

Le backend sera implémenté en JavaScript via Node.js/Express. Cela permettra une communication rapide et efficace avec le serveur d'application et les bases de données, grâce à la nature asynchrone de Node.js/Express.

**a. Avantages**

- **Performance** : Modèle asynchrone adapté aux applications en temps réel et aux nombreuses requêtes simultanées.
- **Flexibilité** : Facilité d'intégration avec divers types de bases de données et services externes.
- **Partage de code** : Possibilité de partager du code entre le frontend et le backend.

**b. Inconvénients**

- **Complexité du débogage** : La gestion des callbacks et des promesses peut rendre le débogage difficile.
- **Performance CPU-bound** : Moins performant pour les tâches intensives en calcul.

#### 1.1.4. Base de données

Le projet sera conçu avec deux bases de données distinctes; PostgreSQL est utilisé pour le stockage des données structurées. PostgreSQL offre une robustesse transactionnelle et une performance élevée pour les requêtes complexes. Et MongoDB est utilisée pour le stockage des données non structurées et semi-structurées. MongoDB permet une flexibilité de schéma et une performance élevée pour les opérations de lecture/écriture.

**a. Avantages**

- Les bases de données sont optimisées pour gérer des requêtes complexes et des volumes importants de données de manière efficace.
- Les bases de données sont optimisées pour gérer des requêtes complexes et des volumes importants de données de manière efficace.
- Les bases de données disposent de mécanismes de sauvegarde et de restauration, ainsi que de récupération après sinistre pour garantir la continuité des opérations.

#### **b. Inconvénients**

- La configuration, l'optimisation, la sauvegarde et la récupération des bases de données requièrent des compétences spécialisées, ce qui peut augmenter les coûts de gestion.
- Les solutions de bases de données peuvent être coûteuses, en particulier les versions commerciales avec support technique. Les coûts matériels pour les bases de données volumineuses peuvent également être élevés.
- Les bases de données peuvent rencontrer des problèmes de concurrence lorsqu'un grand nombre de transactions sont effectuées simultanément, nécessitant des mécanismes sophistiqués de gestion des verrous et de l'isolation des transactions.

#### **1.1.5. Front Office (Interface utilisateur)**

Le front office est la partie visible de l'application avec laquelle les utilisateurs interagissent directement. Il est conçu pour offrir une expérience utilisateur intuitive et fluide, facilitant l'accès aux différentes fonctionnalités du site. Cette interface utilisateur réactive et attrayante est développée en utilisant des technologies modernes telles que React, garantissant une navigation rapide et agréable.

#### **a. Avantages**

- **Expérience utilisateur** : Interface utilisateur intuitive et responsive.
- **Accessibilité** : Accessible depuis n'importe quel navigateur web.

#### **b. Inconvénients**

- **Performance** : Peut être limité par les capacités des navigateurs et la qualité de la connexion internet.
- **Sécurité** : Exposé aux attaques web (XSS, CSRF, etc.).



#### 1.1.6. Application mobile

Les applications mobiles Android et iOS permettent aux utilisateurs d'accéder facilement aux fonctionnalités principales du site. Les applications incluront un stockage local des données pour permettre une utilisation hors ligne.

##### a. Avantages

- **Accessibilité** : Les utilisateurs peuvent accéder aux fonctionnalités principales depuis leurs appareils mobiles.
- **Utilisation hors ligne** : Stockage local des données pour une utilisation sans connexion internet.

##### b. Inconvénients

- **Développement multiplateforme** : Nécessite des efforts supplémentaires pour supporter Android et iOS.
- **Maintenance** : Besoin de mises à jour régulières pour s'adapter aux nouvelles versions des systèmes d'exploitation mobiles.

#### 1.1.7. Webservice REST

Les WebServices REST sont utilisés pour la communication entre le front office, le back office, les applications mobiles et le client lourd. Ils assureront l'échange sécurisé et efficace des données entre les différentes parties du système.

##### a. Avantages

- **Interopérabilité** : Facilite la communication entre différentes parties du système, peu importe la technologie.
- **Modularité** : Permet de développer et de déployer indépendamment les différentes fonctionnalités.

##### b. Inconvénients

- **Performance** : Peut introduire une latence supplémentaire en raison des appels réseau.
- **Sécurité** : Nécessite une gestion rigoureuse des aspects de sécurité (authentification, autorisation, chiffrement).

#### 1.1.8. Module de cartographie

Un module de cartographie sera intégré pour permettre aux utilisateurs de localiser les objets disponibles pour échange. Cela facilitera les échanges entre utilisateurs proches géographiquement.

##### a. Avantages

- **Utilité** : Améliore l'expérience utilisateur en fournissant des visualisations géographiques.
- **Praticité** : Facilite la localisation des objets et des utilisateurs.

##### b. Inconvénients

- **Complexité** : Intégration et maintenance peuvent être complexes, surtout si des services tiers sont utilisés.
- **Coûts** : Peut engendrer des coûts supplémentaires si des services cartographiques payants sont utilisés.

## 1.2. Choix technologiques

Les choix des technologies nécessaires à la réalisation de l'application s'avèrent très importantes dans la réussite d'une telle mission.

#### 1.2.1. Plateforme de développement

Pour le développement du projet, nous avons opté pour ces choix technologiques; des technologies pour le backend, frontend, mobile, et client lourd.

### **1) Node.js**

Pour la partie backend du projet, Node.js sera le framework utilisé. Node.js permet un développement backend rapide et efficace grâce à son modèle asynchrone, idéal pour les applications en temps réel.

#### **a. Avantages**

- Node.js offre une communication rapide et asynchrone, ce qui est idéal pour les applications en temps réel et les nombreuses requêtes simultanées.

#### **b. Inconvénient**

- La gestion des erreurs et des callbacks dans Node.js peut être complexe, et les performances peuvent être limitées pour les tâches intensives en calcul.

### **2) Leaflet**

Pour le mapping, leaflet sera utilisé. Leaflet est une bibliothèque JavaScript légère utilisée pour l'intégration de cartes interactives dans les applications web.

#### **a. Avantages**

- Leaflet est léger, flexible et facile à intégrer, offrant une solution efficace pour les applications de cartographie.

#### **b. Inconvénient**

- Leaflet peut manquer de fonctionnalités avancées présentes dans d'autres bibliothèques de cartographie plus complètes.

### **3) Java**

Le serveur d'application sera développé en Java. Java est utilisé pour les fonctionnalités backend critiques nécessitant robustesse et sécurité, telles que la gestion des utilisateurs et l'authentification.

Java sera également utilisé pour le développement d'une application Android du projet. Java est utilisé pour le développement natif des applications Android, offrant une performance optimale et un accès complet aux fonctionnalités de la plateforme.

#### **a. Avantages**

- Java offre une robustesse et une sécurité élevées, idéales pour la gestion des utilisateurs, des objets et des échanges.
- Le développement en Java pour Android offre des performances natives élevées et un accès complet aux fonctionnalités de la plateforme.

#### **b. Inconvénient**

- Le développement en Java peut être complexe, ce qui peut allonger les temps de développement.
- Le développement en Java pour Android peut être plus long et complexe, surtout en raison de la diversité des versions d'Android à supporter.

### **4) React**

Pour la partie frontend c'est-à-dire pour le backoffice et le frontoffice, React sera le framework utilisé pour le développement. React est utilisé pour le développement de l'interface utilisateur web, offrant une expérience utilisateur interactive et réactive.

#### **a. Avantages**

- React permet de créer des interfaces utilisateur performantes et réactives, avec une modularité et une réutilisation du code élevées.

#### **b. Inconvénient**

- La gestion de l'état dans React peut être complexe, et nécessite une bonne compréhension de JavaScript et des bibliothèques associées.

### **5) React Native**

Pour la partie mobile IOS, React Native sera le framework utilisé pour le développement. React Native permet de développer des applications mobiles pour iOS et Android en utilisant JavaScript, facilitant le partage de code entre les plateformes.

#### **a. Avantages**

- React Native permet un développement rapide et le partage de code entre iOS et Android, avec une large communauté de développeurs pour le support.

#### **b. Inconvénient**

- Les performances des applications React Native peuvent être légèrement inférieures à celles des applications natives, et dépendent souvent de bibliothèques tierces.

### **6) C# WinForms**

Pour la partie client lourd c'est-à-dire le desktop du projet, C# WinForms sera le framework utilisé pour le développement. C# avec WinForms est utilisé pour le développement d'applications de bureau, permettant une interface utilisateur riche et réactive.

#### **a. Avantages**

- Les applications de bureau développées avec C# WinForms offrent une performance élevée et un accès complet aux fonctionnalités de Windows.

#### **b. Inconvénient**

- Ces applications sont limitées à la plateforme Windows et nécessitent des mises à jour fréquentes pour maintenir la compatibilité et les performances.

### **1.2.2. Système de gestion de base de données (SGBD)**

Il est à préciser qu'il s'agit également d'un élément important pour réussir à concevoir une application vu qu'il persiste toutes les informations et se charge d'assurer l'intégrité des données. C'est pour cela que la performance de l'application globale se base surtout sur sa gestion des données, en d'autres termes, de son SGBD.

#### **1) PostgreSQL**

PostgreSQL est une base de données relationnelle robuste offrant des garanties ACID, ce qui assure une intégrité et une fiabilité des données exceptionnelles. Elle est particulièrement performante pour les requêtes complexes

et les transactions nécessitant une forte cohérence des données. PostgreSQL sera utilisé pour le stockage des données structurées du projet.

**a. Avantages**

- PostgreSQL offre une grande robustesse et une conformité ACID, garantissant des transactions fiables et des requêtes complexes efficaces.

**b. Inconvénients**

- La gestion de PostgreSQL peut être complexe et nécessite des compétences spécialisées, ce qui peut augmenter les coûts de maintenance.

## **2) MongoDB**

MongoDB est une base de données NoSQL flexible permettant de gérer des données non structurées et semi-structurées avec une grande flexibilité de schéma. MongoDB sera utilisée pour le stockage des données non structurées et semi-structurées du projet.

**a. Avantages**

- MongoDB permet une flexibilité de schéma et une scalabilité horizontale, offrant des performances élevées pour les opérations de lecture/écriture.

**b. Inconvénients**

- Le support pour les transactions complexes est limité, et la gestion de la consistance des données peut être plus complexe.

## **2. Infrastructure et sécurité**

Pour garantir la qualité et la sécurité du projet, nous respecterons plusieurs contraintes techniques tout au long du développement.

## 2.1. Sécurité

- **Authentification et Autorisation** : Mise en place de mécanismes robustes pour protéger les données des utilisateurs et assurer la confidentialité des échanges.
- **Chiffrement des Données** : Les données sensibles doivent être chiffrées, aussi bien en transit qu'au repos, pour prévenir tout accès non autorisé.

## 2.2. Gestion des versions

- **Dépôt de Code** : Utilisation de GitLab ou GitHub pour le contrôle de version. Le projet doit être nommé sous la forme mbdsp10-troc-1258-1302-1370.
- **Contributions** : Tous les membres de l'équipe devront participer activement au dépôt de code avec des commits réguliers, permettant une progression continue et transparente du projet.

# 3. Modules fonctionnels

L'application web recommandée aura les caractéristiques ci-après :

- l'application intégrera et agrègera toutes les informations provenant de tous les modules disponibles.
- l'accès sera protégé par un nom d'utilisateur (login) et un mot de passe,
- elle sera composée de plusieurs modules qui comportera plusieurs fonctionnalités (les détails seront dans la partie qui suit),
- chaque utilisateur aura accès à des fonctionnalités configurées par rapport à leur profil d'appartenance,
- l'application devra être accessible à partir d'un navigateur web, un mobile et sur Windows

## 3.1. Gestion des utilisateurs et authentification

Le module de services d'authentification et gestion des utilisateurs est une composante essentielle de notre plateforme, permettant aux utilisateurs de se connecter de manière sécurisée, de gérer leurs comptes et d'interagir avec les profils d'autres utilisateurs. Ce module offre une variété de fonctionnalités pour assurer une expérience utilisateur fluide et sécurisée

### 3.2. Gestion des annonces

Le module de gestion d'annonces est une composante essentielle de notre plateforme d'échange d'objets entre particuliers. Il permet aux utilisateurs de créer, modifier, supprimer et gérer leurs annonces, ainsi que de rechercher et filtrer les objets disponibles à l'échange.

### 3.3. Gestion des échanges d'objets

Le module de services d'échanges d'objets est conçu pour faciliter les transactions d'échange entre utilisateurs sur notre plateforme. Il permet aux utilisateurs de proposer, gérer, et finaliser des échanges d'objets de manière sécurisée et efficace.

### 3.4. Service de mapping

Le service de mapping permettra aux utilisateurs de:

- localiser les objets disponibles pour échange sur une carte interactive
- visualiser géographiquement les objets, facilitant ainsi la recherche d'objets proches et la planification des échanges.

### 3.5. Service des statistiques

Le module de statistiques est essentiel pour fournir des insights détaillés sur l'utilisation et les activités de la plateforme d'échange d'objets. Il permet de suivre et d'analyser divers aspects du comportement des utilisateurs et des échanges effectués, offrant ainsi des informations précieuses pour améliorer la plateforme et optimiser l'expérience utilisateur.

## 4. Fonctionnalités par profil

Les utilisateurs sont organisés en profils. Ainsi, les autorisations d'accès à chaque fonctionnalité seront configurées pour chaque profil.

Chaque utilisateur aura des droits assignés au profil auquel il appartient.



## 4.1. Administrateur

Il aura la possibilité de:

- Gérer et modérer les localisations des objets sur la carte.
- Assurer la demande de consultation des objets grâce à une liste.
- Accéder à l'ensemble des fonctionnalités de configuration et de maintenance du module de mapping.
- Superviser et analyser les données géographiques pour optimiser le service.

## 4.2. Troqueur

Il aura la possibilité de:

- Visualiser les objets disponibles sur la carte interactive.
- Utiliser les services de recherche et de filtrage géographique pour trouver des objets à échanger.
- Ajouter ou mettre à jour la localisation de ses propres objets.
- Accéder aux détails des objets en cliquant sur les marqueurs de la carte.
- Utiliser les services de navigation pour planifier les itinéraires vers les objets.

# 5. Spécifications détaillées des modules

Avant d'illustrer les spécifications de chaque module, les règles générales suivantes sont à appliquer selon besoin :

- L'affichage des informations liées aux objets et aux échanges est contrôlé par le niveau d'accès du compte.
- Des champs libres, dynamiques et paramétrables sont mis à disposition des gestionnaires de cette application pour des informations spécifiques. La mise en place de ces champs devrait être en accord avec les deux parties.
- Les différentes dates de création sont automatiques. Leur modification dépend de la règle imposée au moment de l'exécution de l'ouvrage.
- Un système de notification accompagne chaque module. Cette notification est déclenchée à chaque nouvel enregistrement, modification et suppression.

## 5.1. Gestion des utilisateurs et authentification

### 5.1.1. Métiers concernés

- Utilisateur

### 5.1.2. Descriptions fonctionnalités

#### 5.1.2.1. **Authentification sécurisée :**

- Connexion :

Les utilisateurs se connectent en utilisant leur adresse e-mail et un mot de passe sécurisé.

- Déconnexion :

Les utilisateurs peuvent se déconnecter en cliquant sur un bouton de déconnexion, mettant fin à leur session active.

#### 5.1.2.2. **Création de compte utilisateur :**

- Formulaire d'inscription :

Les utilisateurs doivent remplir un formulaire d'inscription comprenant des champs obligatoires comme l'adresse e-mail, le mot de passe, nom d'utilisateur, Numéro de téléphone, Adresse

#### 5.1.2.3. **Visualisation et gestion des comptes utilisateurs :**

- Tableau de bord des utilisateurs :

Les administrateurs peuvent visualiser une liste de tous les utilisateurs inscrits avec des options pour trier et filtrer.

- Suspension de compte :

Les administrateurs peuvent suspendre des comptes utilisateurs, empêchant ainsi leur accès temporaire.

- Suppression de compte :

Les administrateurs peuvent supprimer définitivement des comptes utilisateurs.

- Réactivation de compte :

Les comptes suspendus peuvent être réactivés après une approbation par les administrateurs.

#### **5.1.2.4. Gestion des rôles des utilisateurs:**

- Attribution des rôles :

Les administrateurs peuvent attribuer des rôles spécifiques (utilisateur standard ou administrateur) aux utilisateurs.

#### **5.1.2.5. Modification des informations de profil :**

- Mise à jour des informations personnelles :

Les utilisateurs peuvent mettre à jour leurs informations personnelles telles que l'adresse e-mail, le mot de passe, nom d'utilisateur, Numéro de téléphone, Adresse.

#### **5.1.2.6. Visualisation du profil public d'autres utilisateurs :**

Les utilisateurs peuvent consulter les profils publics d'autres membres, incluant les informations partagées publiquement

## **5.2. Gestion des annonces**

### **5.2.1. Création d'annonces:**

- Créer une nouvelle annonce pour un objet à échanger en saisissant un titre, une description, en ajoutant des photos et en sélectionnant une catégorie.

### **5.2.2. Modification et suppression d'annonces:**

- Modifier les détails d'une annonce existante.
- Supprimer une annonce existante de la plateforme.
- Modifier ou supprimer des annonces non conformes ou signalées.

### **5.2.3. Visualisation des Annonces Publiées:**

- Voir toutes les annonces publiées par l'utilisateur dans un tableau de bord.

#### 5.2.4. Recherche et Filtrage des Annonces:

- Rechercher des objets en utilisant des mots-clés.
- Filtrer les résultats de recherche par catégorie, date de publication, etc.
- Trier les résultats de recherche par pertinence, date, etc.

#### 5.2.5. Consultation des Détails des Annonces:

- Voir les détails complets d'une annonce, y compris le titre, la description, les photos, et les informations de contact de l'utilisateur ayant publié l'annonce.

#### 5.2.6. Upload de Photos:

- Télécharger des photos pour les annonces créées.

#### 5.2.7. Masquer les Annonces Signalées:

- Signaler des annonces non conformes. Les annonces signalées peuvent être masquées ou supprimées par les administrateurs.

#### 5.2.8. Gestion des Catégories d'Objets:

- Ajouter de nouvelles catégories pour les objets.
- Modifier les catégories existantes.
- Supprimer les catégories obsolètes ou inutiles.

### 5.3. Gestion des échanges d'objets

#### 5.3.1. Métiers concernés

- Echange
- Utilisateur

### 5.3.2. Descriptions fonctionnalités

#### 5.3.2.1. **Proposition d'échange :**

- Sélection des objets à proposer en échange :

Les utilisateurs choisissent parmi les objets pour en proposer un échange.

#### 5.3.2.2. **Réception et gestion des propositions reçues :**

- Réception des propositions :

Les utilisateurs reçoivent des notifications lorsqu'ils reçoivent de nouvelles propositions d'échange.

- Gestion des propositions :

Une section dédiée permet aux utilisateurs de voir, organiser, et suivre les propositions d'échange reçues.

#### 5.3.2.3. **Acceptation ou refus des propositions d'échange**

- Acceptation :

Les utilisateurs peuvent accepter une proposition d'échange, confirmant ainsi leur accord pour procéder à l'échange.

- Refus :

Les utilisateurs peuvent refuser une proposition d'échange, avec la possibilité de fournir des commentaires.

- Historique des transactions :

Un historique des échanges acceptés et refusés est conservé pour référence.

#### 5.3.2.4. **Notifications pour les nouvelles propositions et acceptations d'échange**

- Nouvelles propositions :

Les utilisateurs reçoivent des notifications instantanées pour chaque nouvelle proposition d'échange.

- Acceptations :

Des notifications sont envoyées lorsqu'une proposition d'échange est acceptée

#### **5.3.2.5. Génération et scan de QR codes pour faciliter les échanges d'objets**

- Génération de QR codes :  
Pour chaque échange accepté, un QR code unique est généré pour sécuriser et faciliter l'échange.
- Scan de QR codes :  
Les utilisateurs peuvent scanner le QR code de l'autre partie pour valider et confirmer l'échange d'objets.

### **5.4. Service de mapping**

#### **5.4.1. Affichage des Objets sur une Carte:**

- Intégration d'une carte interactive sur le site web et l'application mobile.
- Affichage des objets disponibles pour échange avec des marqueurs sur la carte.
- Chaque marqueur fournira des informations de base sur l'objet, telles que le nom, la catégorie, et une image miniature.
- 

#### **5.4.2. Recherche et Filtrage Géographique:**

- Fonctionnalité de recherche permettant aux utilisateurs de trouver des objets dans une zone géographique spécifique.
- Filtrage des résultats de la carte par distance, catégorie d'objets, et autres critères pertinents.
- Option pour afficher uniquement les objets disponibles dans un rayon défini autour de la position actuelle de l'utilisateur.
- 

#### **5.4.3. Détails de l'Objet:**

- Clic sur un marqueur pour afficher une fenêtre pop-up avec des détails complets sur l'objet, y compris une description, des photos, et les coordonnées du propriétaire.

- Lien direct depuis la pop-up pour proposer un échange pour l'objet sélectionné.

#### 5.4.4. Gestion des Localisations :

- Interface pour que les utilisateurs puissent ajouter ou mettre à jour la localisation de leurs objets lors de la création ou de la modification d'une annonce.
- Utilisation de l'API de géolocalisation du navigateur ou de l'appareil mobile pour simplifier la saisie de la localisation.

### 5.5. Service des statistiques

#### 5.5.1. Nombre d'objets échangés:

- Visualiser le nombre total d'objets échangés par l'utilisateur.
- Voir les statistiques des échanges par période (jour, semaine, mois, année).
- Voir le nombre d'objets échangés en général par jour, semaine, mois, et année pour toute la plateforme.

#### 5.5.2. Catégories Populaires d'Objets Échangés:

- Voir les catégories d'objets les plus échangées par l'utilisateur.
- Voir les catégories d'objets les plus populaires échangées sur la plateforme.
- 

#### 5.5.3. Répartition Géographique:

- Accéder à une vue générale de la répartition géographique des utilisateurs, si disponible publiquement.





## 6. Conclusion

La réussite de ce projet d'application de troc entre particuliers apportera des avantages importants. En créant une plateforme intégrée, nous pouvons gérer les échanges de manière efficace et sans complications, améliorant ainsi l'expérience des utilisateurs.

Ce projet fournira une base solide pour l'ajout des différents modules. Les recommandations sur l'architecture et les technologies utilisées assureront la disponibilité continue du système et sa capacité à évoluer selon les besoins futurs des utilisateurs.

En fin de compte, cette application de troc aidera à promouvoir la réutilisation des objets et à réduire les déchets, tout en créant une communauté active d'échange et de partage.