

# Projet de synthèse : Une application d'échange local de biens et de compétences

## 1. Présentation du projet

**Objectif général du projet :** « Construire des communautés plus solidaires en créant une application web où les voisins peuvent partager facilement leurs biens physiques et leurs compétences, réduisant la consommation et favorisant les liens dans le monde réel. »

**Description du projet :** Le projet comprendra la conception, le développement et le déploiement d'une application web qui sert de réseau d'échange hyperlocal. Les utilisateurs peuvent répertorier les objets qu'ils souhaitent prêter ou donner (par exemple, des outils de bricolage, des appareils électroniques) ou les compétences qu'ils peuvent offrir (par exemple, « Je vous aiderai à monter vos meubles IKEA », « Je peux vous aider avec le jardinage », « Je vous apprendrai à jouer de la guitare »). D'autres utilisateurs à proximité peuvent parcourir ces annonces, envoyer des demandes et convenir d'un échange. Un système de réputation intégré, basé sur des avis, assure la confiance entre les utilisateurs.

Le projet nécessitera l'application de divers principes et pratiques de génie logiciel abordés tout au long du cours, notamment l'ingénierie des exigences, la conception, les tests et le déploiement de logiciels.

## 2. Fonctionnalités de l'application

### 1. Authentification des Utilisateurs

- **Inscription et Profil Utilisateur :** Un nouvel utilisateur peut s'inscrit en fournissant son email, son nom et sa localisation générale afin d'accéder à la plateforme.
- **Connexion Sécurisée :** Utilisateur enregistré peut se connecter afin que ses données et ses transactions soient privées.
- **Gestion du Profil :** Un utilisateur peut ajouter une photo de profil, une courte biographie et vérifier son numéro de téléphone pour instaurer la confiance au sein de la communauté.

### 2. Gestion des Annonces d'Objets

- **Créer une Annonce d'Objet :** Un utilisateur identifié peut créer une annonce pour un objet qu'il veut prêter ou donner en fournissant un titre, une description, une catégorie, des images, et en précisant sa disponibilité. Il peut également ajouter un commentaire supplémentaire indiquant l'objet ou la compétence avec lequel il souhaite échanger son objet.
- **Parcourir et Rechercher des Annonces :** Un utilisateur peut parcourir toutes les annonces disponibles, filtrer par catégorie et rechercher par mots-clés pour trouver ce qu'il recherche.
- **Voir les Détails d'une Annonce :** Un utilisateur peut voir une page détaillée pour toute annonce, affichant toutes les informations et le profil du créateur de l'annonce.

### 3. Gestion des Échanges de Compétences

- **Proposer une Compétence :** Un utilisateur enregistré peut créer une annonce pour une compétence qu'il offre en fournissant un titre, une description et ses disponibilités. Il peut également préciser dans un commentaire l'objet ou la compétence qu'il souhaite bénéficier en retour.
- **Parcourir les Compétences :** Un utilisateur peut parcourir et rechercher les compétences disponibles offertes par d'autres dans sa région.

### 4. Flux de Transaction d'Échange

- **Envoyer une Demande d'Échange :** En tant que bénéficiaire, je peux répondre à une annonce d'échange (une annonce d'un objet ou une compétence), en précisant l'objet ou la compétence que j'offre en retour et la date précise de l'échange.
- **Gérer les Demandes :** En tant que donneur, je peux voir mes demandes et accepter ou refuser des propositions d'échange.
- **Messagerie :** En tant qu'utilisateur, une fois une demande acceptée, je peux envoyer un message à l'autre partie pour finaliser la logistique (lieu de rendez-vous) en utilisant un système de chat intégré à l'application.

### 5. Système de Réputation et de Confiance

- **Avis et Notation :** Après qu'un échange est terminé, les deux utilisateurs peuvent laisser une notation publique (1 à 5 étoiles) et un avis l'un pour l'autre.

### 6. Système de notification

- **Notifications :** Les utilisateurs reçoivent des notifications par email ou sur l'application pour les nouvelles demandes, les messages et les mises à jour des demandes.

### 3. Phases du Cycle de Vie du Développement Logiciel (SDLC) et Tâches

Phase SDLC	Tâches pour l'Équipe
<b>1. Ingénierie des Exigences</b>	<ul style="list-style-type: none"> <li>- <b>Élicitation</b> : Collecter et catégoriser les différentes exigences.</li> <li>- <b>Spécification</b> : Créer un Document de <b>Spécification des Exigences Logicielles</b> (SRS).</li> </ul>
<b>2. Conception et Architecture Logicielle</b>	<ul style="list-style-type: none"> <li>- <b>Architecture</b> : Concevoir une architecture adaptée à l'application en se basant sur les <b>patterns d'architecture</b>.</li> <li>- <b>Conception</b> : Utiliser des <b>patrons de conception</b> pour concevoir les différentes fonctionnalités de l'application.</li> <li>- <b>Modélisation</b> : Créer des <b>diagrammes UML</b> pour représenter les composants de l'application et leurs interactions.</li> </ul>
<b>4. Implémentation</b>	<ul style="list-style-type: none"> <li>- <b>Mise en œuvre</b> : <ul style="list-style-type: none"> <li>- Utiliser un langage de programmation orienté objet (ex : Java, Python, ...) pour développer le backend et le frontend de l'application.</li> <li>- Implémenter les fonctionnalités de base en fonction des spécifications de conception et les intégrer dans l'application.</li> <li>- <b>Contrôle de Version</b> : Utiliser <b>Git</b> avec un workflow de branches de fonctionnalités. Chaque nouvelle fonctionnalité est développée dans sa propre branche et fusionnée via un Pull Request.</li> </ul> </li> </ul>
<b>5. Tests et Assurance Qualité</b>	<ul style="list-style-type: none"> <li>- <b>Tests Unitaires</b> : Effectuer des tests pour les fonctions backend individuelles (ex : logique d'inscription, validation de demande).</li> <li>- <b>Tests d'Intégration</b> : Tester les composants externes intégrés qui sont intégrés dans l'application (ex : S'assurer que la messagerie chat fonctionne correctement).</li> </ul>
<b>6. DevOps et CI/CD</b>	<ul style="list-style-type: none"> <li>- <b>Pipeline CI/CD</b> : Mettre en œuvre un pipeline CI/CD pour automatiser le processus de déploiement et surveiller les performances de l'application.</li> </ul>
<b>7. Sécurité</b>	<ul style="list-style-type: none"> <li>- <b>Authentification</b> : Hacher les mots de passe dans la base de données (ex : avec <b>bcrypt</b>).</li> <li>- <b>Confidentialité</b> : Ne pas exposer les adresses exactes des utilisateurs</li> </ul>
<b>8. Maintenance et Évolution</b>	<ul style="list-style-type: none"> <li>- <b>Documentation</b> : Rédiger un <code>README.md</code> complet avec les instructions d'installation.</li> <li>- <b>Déploiement</b> : Déployer l'application localement ou sur une URL publique.</li> </ul>

### 4. Critères d'Évaluation

Votre travail sera évalué à la fois sur le produit final et le processus utilisé pour le construire.

- **Processus d'Ingénierie Logicielle et Documentation**
  - **Qualité de la Documentation** : SRS, Diagrammes de Conception (UML), `README`.
  - **Preuve de Processus** : Historique des commits Git significatif.
  - **Respect du SDLC** : Avez-vous respecté les phases du SDLC ?
- **Qualité du Code et Implémentation Technique**
  - **Code Propre** : Le code est-il bien structuré, lisible et correctement commenté ?
  - **Architecture** : La séparation entre frontend et backend est-elle claire ?
  - **Choix Techniques** : Les bibliothèques et frameworks sont-ils utilisés de manière appropriée ?
- **Tests, DevOps et Sécurité**
  - **Tests** : Présence et qualité des tests unitaires et d'intégration.
  - **CI/CD** : Le pipeline est-il automatisé et fonctionnel ?
  - **Sécurité** : Implémentation des principes de sécurité de base (auth, protection des données).
- **Présentation Finale et Démo**
  - **Clarté** : Explication claire du projet et des décisions prises.
  - **Démo** : Une démonstration des fonctionnalités clés de l'application.
  - **Q&R** : Capacité à répondre aux questions techniques et de conception.