

SHARE

Parisot Clément, Boudry Mélanie, Poteaux Agathe, Bruyé Antoine

Sommaire :

Présentation du projet	1
Fonctionnalités de l'application	2
→ Authentification et gestion des utilisateurs	2
→ Tableau de bord	2
→ Gestion des Documents	2
→ Notifications	2
→ Intégration avec l'API Symfony existante	2
→ Interface utilisateur intuitive	2
Spécifications techniques	3
Plateformes	3
Version Minimale	3
Compatibilité avec les résolutions	3
Langages et Frameworks	3
Services tiers	3
Interface utilisateur	4
Contraintes	4
Livrables	4
Évolutivité	4

Présentation du projet

GitHub du projet : git clone [git@github.com:MelB04/Share.git](https://github.com/MelB04/Share.git)

Figma :

<https://www.figma.com/file/tuCGGdxZzqtx7fsacMUIxO/Untitled?type=design&node-id=0%3A1&mode=dev&t=SjfsAAjdTBZpvd0u-1>

Nous avons créé une application avec le Framework Symfony 6.3 s'appelant "Share". Cette application est un projet du BTS dans le cadre de l'Épreuve E5 SLAM. Share est un site web permettant de partager des fichiers entre utilisateurs. Via ce site web, nous pouvons totalement gérer nos partages avec les autres utilisateurs.

Fonctionnalités de l'application

→ Authentification et gestion des utilisateurs

Mise en place d'un système d'authentification sécurisé.

Permettre aux utilisateurs de s'inscrire, de se connecter et de récupérer leur mot de passe.

→ Tableau de bord

Afficher une vue d'ensemble des documents partagés.

Indiquer les documents récemment ajoutés ou partagés.

→ Gestion des Documents

Uploader des documents depuis le téléphone.

Afficher une liste des documents existants.

Partager des documents avec d'autres utilisateurs..

→ Notifications

Envoyer des notifications pour informer les utilisateurs des documents partagés, etc.

→ Intégration avec l'API Symfony existante

Connecter l'application React Native à l'API Symfony existante pour assurer la cohérence des données.

→ Interface utilisateur intuitive

Concevoir une interface utilisateur moderne et conviviale, respectant les meilleures pratiques de conception d'interface mobile.

Spécifications techniques

Plateformes

L'application cible exclusivement la plateforme Android, étant donné les contraintes de temps. Cela signifie que l'application sera développée spécifiquement pour les appareils Android.

Version Minimale

Pour garantir une large compatibilité et une adoption optimale, l'application sera conçue pour être compatible avec la version minimale d'Android la plus répandue dans l'industrie à ce moment-là. Cela permettra d'assurer que l'application peut être utilisée par un large éventail d'utilisateurs sans exclusion majeure.

Compatibilité avec les résolutions

L'application doit être conçue de manière à s'adapter aussi bien aux téléphones qu'aux tablettes Android. Cela implique d'assurer une mise en page et une expérience utilisateur optimisées pour les deux types d'appareils, en tenant compte des différentes tailles d'écran et des résolutions.

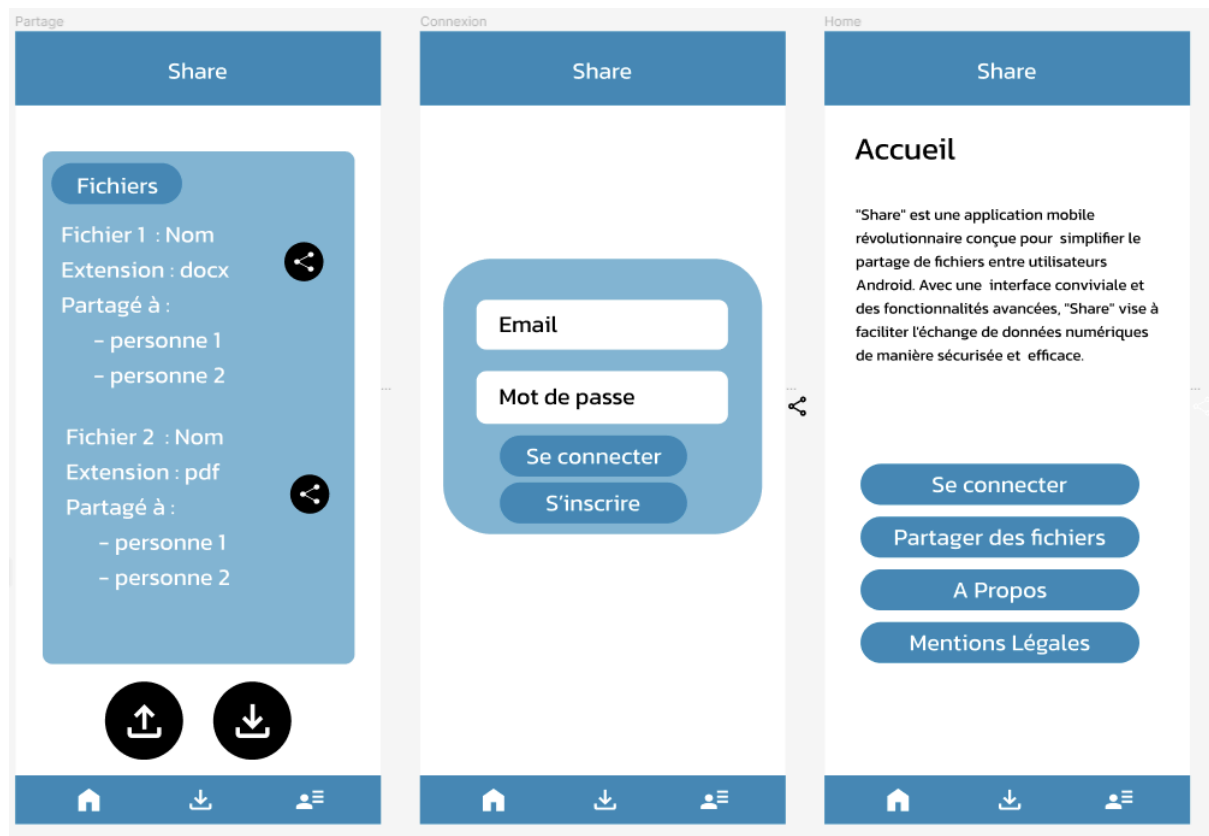
Langages et Frameworks

L'application sera développée en utilisant React Native, un framework JavaScript populaire pour le développement d'applications mobiles multiplateformes. Cela permettra de rationaliser le processus de développement et de partager une grande partie du code entre les versions Android de l'application.

Services tiers

L'application nécessite une connexion avec une API hébergée pour accéder aux données nécessaires à son fonctionnement. Cette API sera hébergée sur un serveur distant et fournira les données requises à l'application en temps réel. Les requêtes vers cette API seront gérées de manière sécurisée et efficace pour assurer la fluidité de l'expérience utilisateur.

Interface utilisateur



Contraintes

Assurer la sécurité des données utilisateur.

Assurer la compatibilité avec les versions récentes d'Android

Livrables

- Application mobile React Native fonctionnelle.
- Une documentation détaillée.

Évolutivité

- Mettre en place un espace de dessin (Canva JavaScript) pour permettre de partager en immédiat