

Sujet de projet : KeepNote - Application de notes et de tâches

Introduction

Ce projet consiste à développer une application mobile moderne de prise de notes et de gestion de tâches, inspirée de Google Keep. L'application permettra aux utilisateurs de créer, organiser et gérer des notes et des tâches, avec un système de catégorisation par couleur et la possibilité d'associer des tâches à des notes spécifiques.

L'application sera développée avec Expo et React Native, et communiquera avec une API REST backend déjà mise en place.

Date de rendu : 29 avril 2025

Objectifs pédagogiques

- Développer une application React Native complète avec Expo
- Maîtriser la navigation entre les écrans avec Expo Router
- Implémenter un système d'authentification
- Gérer les états d'application avec les Hooks React
- Optimiser les communications avec l'API (défi de limitation des requêtes)
- Implémenter des stratégies de mise en cache et de stockage local
- Créer des interfaces utilisateur réactives et esthétiques
- Implémenter le mode sombre/clair

Fonctionnalités requises

Authentification

- Écran de connexion avec email et mot de passe
- Stockage local du jeton d'authentification
- Option de déconnexion
- **Important** : Pas d'inscription dans l'application. Les comptes doivent être créés via le site administrateur à l'adresse :
<https://keep.kevindupas.com/admin/register>

Notes

- Affichage des notes dans une interface de type grille/liste
- Création de nouvelles notes avec titre et contenu
 - Possibilité d'associer des catégories lors de la création
- Modification des notes existantes
 - Modifier le titre et le contenu
 - Ajouter/supprimer des catégories associées
- Suppression de notes existantes
- Filtrage des notes par catégorie
- Recherche de notes par texte

Tâches

- Affichage des tâches dans une liste

- Création de nouvelles tâches avec description
 - Possibilité d'associer une note lors de la création (optionnel)
 - Possibilité d'ajouter des sous-tâches lors de la création
- Modification des tâches existantes
 - Modifier la description principale
 - Ajouter/supprimer/modifier des sous-tâches
 - Associer ou dissocier une note
- Marquage des tâches comme complétées/non-complétées
 - Marquage individuel pour chaque sous-tâche
 - Possibilité de marquer la tâche principale comme complétée
- Suppression des tâches et sous-tâches
- Filtrage des tâches (par exemple : complétées/non-complétées, par note associée)

Catégories

- Affichage des catégories existantes
- Création de nouvelles catégories avec nom et couleur
- Association des catégories aux notes (à la création et à l'édition)
- Modification et suppression des catégories personnalisées

Fonctionnalités supplémentaires

- Mode sombre / mode clair
- Indicateurs de chargement et gestion des erreurs
- Interface utilisateur réactive

Structure du projet

```

app/
├── (tabs)/
│   ├── index.tsx          # Écran principal des notes
│   ├── tasks.tsx          # Écran principal des tâches
│   ├── settings.tsx       # Écran des paramètres (déconnexion, switch thème)
│   └── _layout.tsx        # Layout des onglets
├── tasks/
│   ├── create.tsx         # Page de création de tâche
│   └── [id].tsx           # Page d'édition de tâche
├── notes/
│   ├── create.tsx         # Page de création de note
│   └── [id].tsx           # Page d'édition de note
├── auth/
│   └── login.tsx          # Page de connexion
├── _layout.tsx            # Layout principal
└── +not-found.tsx        # Page 404

components/
├── notes/
│   ├── NoteList.tsx       # Composant de liste des notes
│   └── ...
├── tasks/
│   ├── TaskList.tsx       # Composant de liste des tâches
│   └── ...

```

```
├─ ui/
│   └─ ...
└─ ...

hooks/
├─ useColorScheme.ts    # Hook pour le thème
└─ ...

contexts/
└─ AuthContext.tsx      # Contexte d'authentification
```

Défi : Optimisation des requêtes API

Important : L'application doit effectuer moins de 10 000 requêtes vers le backend. Une pénalité de 2 points sera appliquée si ce seuil est dépassé.

Pour respecter cette contrainte, vous devez implémenter des stratégies telles que :

- Stockage local (AsyncStorage) des données
- Mise en cache intelligente
- Requêtes par lot (batch requests)
- Synchronisation périodique plutôt que constante
- Gestion optimisée des mises à jour

Le backend permet de consulter le nombre de requêtes effectuées par votre application. Vous pouvez suivre cette métrique via le tableau de bord admin.

API et Backend

- L'API est accessible à l'adresse fournie dans les instructions du projet
- La documentation complète de l'API est fournie séparément
- Vous pouvez visualiser vos données (notes, tâches, catégories) et le nombre de requêtes API via l'interface admin après création de compte :
<https://keep.kevindupas.com/admin>

Barème d'évaluation (20 points)

Fonctionnalités (12 points)

- Authentification (2 points)
 - Connexion fonctionnelle (1 point)
 - Gestion des tokens et déconnexion (1 point)
- Gestion des notes (4 points)
 - Affichage, création, modification, suppression (2 points)
 - Catégorisation et recherche (2 points)
- Gestion des tâches (4 points)
 - Affichage, création, modification, suppression (2 points)
 - Sous-tâches et marquage (2 points)
- Gestion des catégories (1 point)
- Interface utilisateur (1 point)
 - Mode sombre/clair (0,5 point)
 - Indicateurs de chargement et gestion des erreurs (0,5 point)

Architecture et qualité du code (5 points)

- Organisation du projet selon la structure recommandée (1,5 points)
- Bonnes pratiques React Native et Expo (1,5 points)
- Gestion propre des états et utilisation correcte des hooks (1 point)
- Lisibilité et maintenabilité du code (1 point)

Optimisation des requêtes API (3 points)

- Respect de la limite de 10 000 requêtes (critère éliminatoire : -2 points si dépassé)
- Mise en place de stratégies efficaces (3 points)

Contraintes techniques

- Implémenter une stratégie efficace de mise en cache et de stockage local
- Effectuer moins de 10 000 requêtes API au total

Bonne chance pour ce projet !