

Développement d'une application mobile avec React Native & Expo

L'objectif de ce TP est de concevoir et développer une application mobile multiplateforme (iOS et / ou Android) en utilisant le framework React Native et l'écosystème Expo. Vous apprendrez à gérer la navigation complexe, la consommation d'API REST, la géolocalisation et la gestion des permissions.

1. Configuration et Environnement

- **Framework :** React Native avec Expo (*Managed Workflow*).
 - **Langage :** TypeScript.
 - **Navigation :** React Navigation / Expo Router.
 - **Icônes :** @expo/vector-icons.
-

2. Cahier des Charges

A. Architecture et Navigation

Vous devez mettre en place une structure de navigation ergonomique comprenant :

- Une navigation par **onglets (Tab Navigation)** pour les sections principales.
- Ou un **menu latéral (Drawer Navigation)**.
- *Optionnel* : Une combinaison des deux (ex: un Drawer contenant un Tab Navigator ou inversement).

B. Module "Explorateur de Données" (API REST)

Choisissez une API publique de votre choix (ex: [PokeAPI](#), [Star Wars API](#), [Rick and Morty API](#)) ou toute autre API, y compris une API que vous avez créée vous-même (hors TP).

1. **Liste** : Affichez les éléments sous forme de liste (`FlatList`).
2. **Pagination** : Gérez le chargement de données supplémentaires lors du défilement (`Infinite Scroll`).
3. **Détails** : Lors d'un clic sur un élément, naviguez vers une vue détaillée affichant plus d'informations.

C. Module "Météo & Gastronomie" (Capteurs et Logique)

Ce module doit proposer une recette de cuisine adaptée au climat actuel de l'utilisateur.

1. **Permissions** : Demandez l'autorisation d'accéder à la position de l'appareil via `expo-location`.
2. **Géolocalisation** : Récupérez les coordonnées (latitude/longitude) de l'utilisateur.
3. **Météo** : Utilisez une API météo (ex: [OpenWeatherMap](#)) pour obtenir le temps actuel.

-
4. **Recette** : En fonction de la météo (ex: s'il fait froid -> une soupe ; s'il fait chaud -> une salade), consommez une API de recettes (ex: [TheMealDB](#)) pour afficher une suggestion adaptée.
-

3. Contraintes Techniques

- **Gestion d'état** : Utilisation des Hooks (`useState`, `useEffect`).
 - **Composants** : Utilisation de composants natifs (`View`, `Text`, `Pressable`, `Image`, `ActivityIndicator`).
 - **Style** : Utilisation de `StyleSheet`. L'application doit être "Responsive" et visuellement propre.
 - **Erreurs** : Gérez les cas d'erreur (absence de connexion internet, refus de géolocalisation, API indisponible).
-

4. Fonctionnalités Bonus (Optionnel)

Pour valoriser votre travail, vous pouvez ajouter :

- **Cartographie** : Afficher la position de l'utilisateur sur une carte via `react-native-maps`.
 - **Design** : Personnalisation de la `StatusBar` et écran de chargement (Splash Screen).
 - **Capteurs** : Utilisation de l'accéléromètre ou de la caméra.
 - **Partage** : Permettre de partager une recette via le module `Share` de React Native.
 - Ou toute autre fonctionnalité native que vous jugez pertinente.
-

5. Livrables attendus

1. Le code source complet (dépôt Git ou archive .zip sans le dossier `node_modules`).
2. Un court fichier `README.md` expliquant comment lancer le projet et les APIs choisies.
3. Quelques captures d'écran de l'application en fonctionnement.