Cahier des Charges pour l'Application "BusTicketsBooking"

Description

L’application **BusTicketsBooking** permet aux utilisateurs de rechercher des trajets de bus, de visualiser les détails des trajets, de réserver des billets, et de gérer leurs réservations. L’application intègre la gestion de l’état via le pattern BLoC pour séparer la logique métier de l’interface utilisateur, assurant ainsi une réactivité et une architecture robuste.

Objectifs du Projet

1. **Rechercher des bus** selon des critères spécifiques (ville de départ, ville d’arrivée, date, heure).
2. **Visualiser les informations de trajets** (horaires, compagnies, arrêts).
3. **Réserver et annuler des billets**.
4. **Consulter l’historique et le statut des réservations**.

Architecture de l’Application

- **Backend API** : L’application se connectera à une API qui fournira les données des trajets, des bus, et des réservations.
- **Gestion de l’état avec BLoC** : Chaque fonctionnalité (recherche, réservation, affichage des trajets) sera gérée par un BLoC séparé.

Fonctionnalités et Tâches à Réaliser

1. Écran de Connexion et d’Inscription

- **Tâches** :
    - Création d’un formulaire de connexion et d’inscription.
    - Gestion des erreurs (mot de passe incorrect, utilisateur non trouvé).

    - Intégration de l’authentification avec une API.
- **Bloc** :
    - Créer un AuthBloc pour gérer l’état de connexion et d’inscription, les succès et erreurs.

2. Écran de Recherche de Trajet

- **Tâches** :
    - Formulaire de recherche (ville de départ, ville d’arrivée, date).
    - Bouton de soumission pour déclencher la recherche.
    - Affichage des résultats avec les détails de chaque trajet.
- **Bloc** :
    - SearchBloc pour traiter les événements de recherche et diffuser les résultats.

3. Écran de Résultats de Recherche

- **Tâches** :
    - Liste des trajets avec informations (horaires, prix, disponibilité).
    - Sélection d’un trajet pour afficher les détails et options de réservation.
- **Bloc** :
    - Utiliser SearchBloc pour actualiser les résultats et gérer la sélection d’un trajet.

4. Écran de Détails du Trajet

- **Tâches** :
    - Affichage des informations détaillées : itinéraire, prix, disponibilité des places.
    - Bouton pour réserver un billet.
- **Bloc** :
    - Créer un BookingBloc pour gérer l’ajout d’une réservation avec vérification des places.

5. Écran de Réservation

- **Tâches** :
    - Formulaire de réservation (sélection des places, informations du passager).
    - Validation de la réservation avec retour de confirmation.
    - Gestion des erreurs (places indisponibles, problème de réseau).
- **Bloc** :
    - Utiliser BookingBloc pour valider et soumettre la réservation.

6. Écran de Confirmation de Réservation

- **Tâches** :
    - Afficher les détails de la réservation confirmée (trajet, numéro de billet, code QR).
    - Option pour annuler la réservation.
- **Bloc** :
    - Intégrer un état de confirmation et gérer les annulations dans BookingBloc.

7. Écran d'Historique des Réservations

- **Tâches** :
    - Liste des réservations passées et en cours.
    - Option pour voir les détails et annuler les réservations en cours.
- **Bloc** :
    - Créer un HistoryBloc pour gérer les états d'affichage des réservations, actualiser la liste, et gérer les annulations.

8. Écran de Profil et Paramètres

- **Tâches** :

    - Modifier les informations de profil (nom, email, numéro de téléphone).
    - Gérer les notifications, préférences de langue.
- **Bloc** :
    - ProfileBloc pour mettre à jour les informations utilisateur et gérer l'état des préférences.

9. Notifications et Mise à Jour des États

- **Tâches** :
    - Notification push pour les mises à jour de réservation, alertes d'annulation, rappel de trajet.
- **Bloc** :
    - Utiliser un NotificationBloc pour déclencher les notifications en fonction des événements du backend.

Backend et Gestion des Données

- **API Backend** pour gérer les informations de trajets, les réservations, et l'authentification.
- **Stockage Local** pour sauvegarder temporairement les données (SharedPreferences ou SQLite) en cas d'absence de connexion.

Tâches Additionnelles pour l'Implémentation

- **Tests unitaires et d'intégration** pour chaque BLoC afin de garantir la robustesse de la logique.
- **Documentation et Guidelines** pour assurer la maintenabilité du code.