

# Projet Flutter 03— Application de réservation de ressources

Nom du projet : SUP4 DEV – FlutterBooking

---

## Objectif :

Développer une application mobile Flutter pour permettre aux utilisateurs de **réserver des ressources** comme :

- salles de réunion,
- véhicules,
- ordinateurs ou matériels divers,

...avec une **interface calendrier** pour visualiser les disponibilités.

---

## Fonctionnalités attendues

### Utilisateurs

- Authentification (avec rôles : utilisateur, administrateur, manager).
- Gestion des profils.

### Catalogue de ressources

- Liste des ressources (avec filtre par catégorie/type).
- Affichage : image, nom, description, capacité.
- CRUD ressources (côté admin uniquement).

### Réservation

- Vue calendrier des créneaux disponibles.
- Sélection d'un jour et d'un créneau horaire.
- Vérification des conflits de réservation (en temps réel).
- Possibilité d'imposer une validation par un manager.
- Annulation/modification possible par l'utilisateur.

### Notifications

- Notifications intégrées à l'app pour :
    - confirmation de réservation,
-

- annulation,
- modification par un admin.

### **Intégration API (optionnel)**

- Google Calendar API ou autre solution tierce pour synchroniser les événements.
- 

## **Aspects techniques Flutter**

### **Architecture**

- Architecture MVC / Clean
- Séparation logique : models, services, providers, views, widgets

### **Gestion de calendrier**

- Intégration de la lib Flutter : `table_calendar`, ou `syncfusion_flutter_calendar` pour affichage planning.

### **Backend / Base de données**

- API REST (NodeJS, Symfony ou Firebase Cloud Functions).
- Utilisation possible de **Firestore** ou base SQL pour la persistance des réservations et la détection des conflits.
- Transactions ou logique métier pour éviter les doubles réservations.

### **Gestion d'état**

- Provider, Riverpod ou Bloc pour gérer les utilisateurs, les ressources et les réservations.
- 

## **Maquette visuelle de l'application (obligatoire)**

Vous devez proposer une maquette visuelle de votre application (réalisée avec Figma, Adobe XD, Balsamiq).

- Fournissez un lien vers la maquette ou intégrez une capture d'écran claire de l'interface prévue.
  - La maquette doit refléter la structure de votre application (navigation, organisation des pages, composants essentiels).
  - Elle servira de base de réflexion et de développement tout au long du projet.
-

Cette étape est obligatoire et sera évaluée. Un bonus pourra être accordé si l'implémentation finale est fidèle à la maquette.

---

## Structure de l'application

lib/

- └─ models/ # User, Resource, Reservation
- └─ services/ # AuthService, ReservationService, NotificationService
- └─ providers/ # resourceProvider, authProvider, calendarProvider
- └─ views/
  - | └─ auth/ # Login, Signup
  - | └─ home/ # Liste des ressources
  - | └─ calendar/ # Agenda avec réservations
  - | └─ admin/ # Gestion des ressources, règles de réservation
- └─ widgets/ # CalendarWidget, ResourceCard, ReservationModal
- └─ main.dart

---

## Critères pour valider le projet

<b>Critère</b>	<b>Exigence minimum</b>
<i>Durée de développement</i>	1 mois
<i>Nombre de commits</i>	20 minimum
<i>UI claire et responsive</i>	Oui (type agenda / liste)
<i>Fonctionnalités principales</i>	Auth, ressource, calendrier, réservation
<i>Gestion des rôles</i>	Admin, utilisateur, manager
<i>Gestion des conflits</i>	Réservation unique par créneau
<i>Notification ou alerte</i>	Présente dans l'app
<i>Documentation (README.md)</i>	Incluse
<i>Déployé ou APK testable</i>	Oui recommandé

---

### **Bonus pour briller**

- Dashboard admin avec statistiques de réservation.
- Ajout d'une carte (Google Maps ou flutter\_map) pour visualiser les emplacements des ressources.
- PDF de confirmation (généré avec pdf package).
- Export iCal / intégration Google Calendar.