

# Projet Flutter 07– Application de Support Client / Ticketing

Nom du projet : DEVMOB SUPPORTCLIENT

Technologies : Flutter (application mobile)

---

## Objectif général

Développer une application mobile permettant à des clients de soumettre des incidents ou demandes (tickets), suivre leur état, échanger avec l'équipe support, et permettre aux agents/admins de gérer efficacement ces tickets.

---

## Modules principaux de l'application Flutter

---

### 1. Authentification & rôles

- Connexion via e-mail/mot de passe.
- Deux profils :
  - **Client** : crée et suit ses tickets.
  - **Support/Administrateur** : traite, affecte, résout les tickets.

Gestion des rôles via tokens JWT ou session API.

---

### 2. Création de tickets

- Formulaire de création :
  - Titre.
  - Description.
  - Priorité.
  - Joindre un ou plusieurs fichiers (images, captures d'écran).
- Sélection de la catégorie du problème.

Joindre un fichier avec `file_picker` ou `image_picker`.

---

### 3. Suivi des statuts

- Affichage des tickets avec leur statut :
-

- Nouveau, En cours, Résolu, Fermé.
- Badge de couleur selon le statut.

Changement de statut côté support uniquement.

---

#### 4. Système de discussion (commentaires)

- Chaque ticket contient un fil de discussion.
- Le client et le support peuvent y ajouter des messages.
- Notifications push ou locales à chaque réponse.

Peut être géré comme un chat simplifié dans chaque ticket.

---

#### 5. Interface Support / Admin

- Affectation d'un ticket à un membre de l'équipe.
  - Changement de statut, gestion des priorités.
  - Vue globale sur tous les tickets.
  - Filtrage : par catégorie, date, statut, utilisateur.
- 

#### 6. Historique et reporting

- Affichage :
  - Nombre de tickets par statut.
  - Temps moyen de résolution.
  - Graphique par type de problème.

Utilisation de fl\_chart ou syncfusion\_flutter\_charts.

---

#### 7. FAQ et assistant automatique (optionnel)

- Ajout d'une page de **FAQ interactive**.
- Suggestions automatiques à partir d'un texte (ex: « mot de passe oublié » ⇒ proposer article).

Possibilité future d'intégrer un chatbot Flutter simple.

---

---

## Technologies Flutter à utiliser

<b>Besoin</b>	<b>Outils recommandés</b>
<i>Authentification</i>	http, jwt_decode, provider ou riverpod
<i>Navigation / gestion d'état</i>	go_router, flutter_bloc, riverpod
<i>Upload fichiers</i>	image_picker, file_picker, dio
<i>PDF ou captures</i>	flutter_pdfview, open_file, path_provider
<i>Graphiques et stats</i>	fl_chart, syncfusion_flutter_charts
<i>Notifications</i>	flutter_local_notifications, Firebase si besoin
<i>Filtrage &amp; recherche</i>	search_delegate, filter_list
<i>UI</i>	flutter_slidable, badges, animated_list

---

## 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 sera accordé si l'interface finale est fidèle à la maquette proposée.
- 

## Structure Flutter conseillée

lib/

└─ main.dart

└─ models/

| └─ ticket.dart

| └─ user.dart

---

```

| └─ comment.dart
└─ services/
| └─ api_service.dart
└─ providers/
| └─ auth_provider.dart
| └─ ticket_provider.dart
└─ screens/
| └─ login_screen.dart
| └─ dashboard_screen.dart
| └─ ticket_list_screen.dart
| └─ ticket_detail_screen.dart
| └─ create_ticket_screen.dart
| └─ faq_screen.dart
| └─ admin_dashboard_screen.dart
└─ widgets/
    └─ ticket_card.dart
    └─ comment_bubble.dart
    └─ stat_widget.dart

```

---

## Planning suggéré

### **Semaine**    **Tâches**

<i>Semaine 1</i>	Authentification + rôles + navigation
<i>Semaine 2</i>	Création de ticket avec upload de fichiers
<i>Semaine 3</i>	Fil de discussion dans ticket + mise à jour statut
<i>Semaine 4</i>	Interface admin + filtrage + stats
<i>Semaine 5</i>	Intégration FAQ / Assistant + tests & design final

---

## Fonctionnalités minimums à livrer

<i>Fonctionnalité</i>	<b>Obligatoire ?</b>
<i>Authentification / rôles</i>	<input checked="" type="checkbox"/>
<i>Création de ticket</i>	<input checked="" type="checkbox"/>
<i>Discussion dans ticket</i>	<input checked="" type="checkbox"/>
<i>Suivi des statuts</i>	<input checked="" type="checkbox"/>
<i>Interface admin / support</i>	<input checked="" type="checkbox"/>
<i>Filtres et recherche</i>	<input checked="" type="checkbox"/>
<i>Statistiques de base</i>	<input checked="" type="checkbox"/>

---

## Bonus si tu veux aller plus loin

- **Notifications push Firebase** pour nouvelles réponses.
- **Offline mode** : consultation locale des tickets déjà chargés.
- **Dark Mode.**
- **Badges de notification dynamiques.**

## Annexe

### C'est quoi un "ticket" dans ce contexte ?

Un **ticket** dans une application de **support** ou de **gestion d'incidents** est **une demande créée par un utilisateur** (client, employé, etc.) pour signaler :

- un **problème technique** (bug, panne, difficulté d'accès...),
  - une **demande de service** (aide, installation, question...),
  - ou une **suggestion d'amélioration**.
- 

### Exemple dans la vie réelle

Imaginons que tu sois :

#### Étudiant ou employé

Tu rencontres un problème :

- Tu ne peux plus te connecter à ton compte.
  - Tu veux demander un certificat.
  - Tu as repéré une erreur dans ton dossier.
-

Tu ouvres **l'application de support**, et tu **crées un ticket** :

"Je ne peux plus me connecter à mon espace depuis hier."

Ce ticket est **envoyé à l'équipe support**, qui peut :

- Te répondre,
  - Te demander plus de détails,
  - Changer son statut (ex: "En cours", "Résolu"),
  - Clôturer le ticket.
- 

#### **Pourquoi ce système est utile ?**

- Pour **centraliser toutes les demandes** : plus d'e-mails éparpillés.
  - Pour **suivre le traitement** de chaque problème.
  - Pour **prioriser les urgences**.
  - Pour **avoir des statistiques** (combien de demandes par mois, quels types de problèmes sont fréquents, etc.).
- 

#### **Où on utilise ce genre d'appli ?**

- Les **universités** : gestion des demandes des étudiants.
- Les **entreprises** : demandes internes (RH, informatique...).
- Les **fournisseurs de services** (banques, opérateurs...).
- Les **centres de support technique** (helpdesk).