

Cahier des Charges - Application de Gestion Scolaire

DEVMOB-ApcPedagogie-21 - Application Mobile de Gestion des Notes Lycée

1. Fiche d'Identité du Projet

Élément	Détails
Nom du projet	DEVMOB-EduLycee
Version	1.0
Type	Application Mobile Multiplateforme
Public cible	Lycées, élèves, professeurs, administration
Durée de développement	7 semaines minimum
Technologies principales	Flutter, Dart, Firebase, Node.js
Architecture	Clean Architecture + BLoC

2. Objectif Principal

Développer une application mobile complète de gestion scolaire pour lycée permettant :

- La centralisation numérique des notes et évaluations
- La communication efficace entre professeurs, élèves et parents
- Le suivi personnalisé de la progression des élèves
- La simplification des processus administratifs

3. Personnes et Rôles Utilisateurs

a. Élève

- Consulte ses notes et moyennes
- Accède à son emploi du temps
- Reçoit les devoirs et annonces
- Consulte son carnet de correspondance

b. Professeur

- Saisit les notes et appréciations
- Gère les classes et matières
- Communique avec élèves et parents
- Planifie les évaluations

c. Parent

- Suit la scolarité de son enfant
- Consulte les notes et absences
- Échange avec les professeurs
- Reçoit les alertes importantes

d. Administration

- Gère les utilisateurs et classes
- Configure l'établissement
- Génère les bulletins et statistiques
- Supervise l'utilisation de la plateforme

e. Vie Scolaire

- Gère les absences et retards
- Traite les sanctions
- Coordonne les événements

4. Fonctionnalités Détaillées

a. Fonctionnalités Globales

1. Gestion des Notes

- Saisie des notes avec coefficients
- Calcul automatique des moyennes
- Historique complet par matière
- Graphiques de progression
- Système de compétences (LOMFR)

2. Emploi du Temps

- Visualisation hebdomadaire/mensuelle
- Alertes de modifications
- Salles et professeurs associés
- Export vers calendrier personnel

3. Communication

- Messagerie sécurisée par rôle
- Annonces de l'établissement
- Carnet de correspondance numérique
- Notifications push importantes

4. Vie Scolaire

- Gestion des absences/retards
- Cahier de textes numérique
- Devoirs et rendus en ligne
- Événements scolaires

b. Espace Élève

Modèles de données

```
class Eleve {
    String uid;
    String nom;
    String prenom;
    String classeld;
    String photoUrl;
    List<String> parentsIds;
    DateTime dateInscription;
    StatutScolaire statut;
}

class Note {
    String id;
    String eleveld;
    String matiereId;
    String professeurId;
    double valeur;
    double coefficient;
    String typeEvaluation;
    String commentaire;
    DateTime date;
    String competenceld;
}
```

Fonctionnalités Spécifiques Élève

- Tableau de bord personnalisé
- Relevé de notes détaillé
- Préviation des moyennes
- Alertes devoirs à rendre

c. Espace Professeur

Outils Pédagogiques

- Saisie rapide des notes par classe
- Grilles d'évaluation par compétences
- Statistiques de classe
- Appréciations personnalisées
- Gestion des devoirs

d. Espace Administration

Tableau de Bord Admin

- Métriques établissement (taux réussite, absentéisme)
- Alertes (notes manquantes, problèmes techniques)
- Rapports périodiques automatiques
- Supervision de l'activité

Gestion Administrative

- Utilisateurs : Création, import, gestion des droits
- Classes : Configuration, professeurs principaux
- Périodes : Trimestres, semestres, années
- Bulletins : Génération, personnalisation, export

5. Architecture Technique

Stack Technologique

Couche	Technologie
Frontend	Flutter 3.x, Dart 3.x
Backend	Firebase + Node.js (Cloud Functions)
Base de données	Cloud Firestore + Cache Redis
Authentification	Firebase Auth (multi-rôles)
Stockage	Firebase Storage (documents, photos)
Notifications	Firebase Cloud Messaging
Analytics	Firebase Analytics

Structure du Projet

```
lib/
├── domain/      # Métier et entités
│   ├── entities/
│   │   ├── eleve.dart
│   │   ├── note.dart
│   │   ├── classe.dart
│   │   └── evaluation.dart
│   └── repositories/
├── data/        # Accès données
│   ├── models/
│   ├── datasources/
│   └── repositories/
├── presentation/ # UI + Logique métier
│   └── blocs/
├── notes/
├── emploi_du_temps/
├── communication/
│   └── auth/
├── pages/
├── widgets/
├── themes/
├── core/        # Commun
│   ├── constants/
│   ├── utils/
│   ├── errors/
│   └── network/
├── injection/   # DI
│   └── dependency_injection.dart
```

6. Design et Expérience Utilisateur

Chartre Graphique

- **Style** : Épuré et professionnel, inspiré environnement éducatif
- **Couleurs** : Palette bleu marine/orange (couleurs éducation nationale)
- **Typographie** : Lisible et accessible (Roboto, OpenDyslexic)
- **Accessibilité** : Respect WCAG, mode daltonien

Navigation par Profil

Élève/Parent

```
BottomNavigationBar(  
  items: [  
    BottomNavigationBarItem(icon: Icon(Icons.dashboard), label: 'Tableau de bord'),  
    BottomNavigationBarItem(icon: Icon(Icons.grade), label: 'Notes'),  
    BottomNavigationBarItem(icon: Icon(Icons.schedule), label: 'Emploi du temps'),  
    BottomNavigationBarItem(icon: Icon(Icons.assignment), label: 'Devoirs'),  
    BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profil'),  
  ],  
)
```

Professeur

dart

```
BottomNavigationBar(  
  items: [  
    BottomNavigationBarItem(icon: Icon(Icons.dashboard), label: 'Classes'),  
    BottomNavigationBarItem(icon: Icon(Icons.grade), label: 'Saisie notes'),  
    BottomNavigationBarItem(icon: Icon(Icons.chat), label: 'Messages'),  
    BottomNavigationBarItem(icon: Icon(Icons.analytics), label: 'Statistiques'),  
    BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profil'),  
  ],  
)
```

Écrans Principaux

1. Connexion multi-profil
2. Tableau de bord personnalisé
3. Relevé de notes détaillé
4. Emploi du temps interactif
5. Messagerie scolaire
6. Saisie des notes (professeurs)
7. Administration complète

7. Sécurité et Confidentialité

Règles de Sécurité Firestore

```
rules_version = '2';  
service cloud.firestore {  
  match /databases/{database}/documents {  
    // Les élèves voient seulement leurs notes  
    match /notes/{document} {  
      allow read: if request.auth != null  
        && (request.auth.uid ==  
          resource.data.elevelId  
            || hasRole('professeur',  
              resource.data.matiereId))
```

```

    || hasRole('admin')
    || isParentOf(resource.data.eleveld));
  allow write: if hasRole('professeur',
resource.data.matiereId)
    || hasRole('admin');
}

// Restrictions données personnelles
match /eleves/{document}{
  allow read: if request.auth != null
    && (request.auth.uid == resource.data.uid
      || hasRole('professeur')
      || isParentOf(resource.data.uid)
      || hasRole('admin'));
}

function hasRole(role, matiereId = null) {

```

```

  let userData =
get(/databases/${database}/documents/utilisat
eurs/${request.auth.uid});
  return userData.data.role == role
    && (matiereId == null ||
userData.data.matiereId.contains(matiereId));
}

function isParentOf(eleveld) {
  let eleveData =
get(/databases/${database}/documents/eleves/
${eleveld});
  return
eleveData.data.parentsIds.contains(request.aut
h.uid);
}
}
}

```

Matrice des Permissions

Action	Élève	Parent	Professeur	Admin
Voir ses notes	✓	✓	✗	✓
Voir notes classe	✗	✗	✓	✓
Saisir notes	✗	✗	✓	✓
Gérer utilisateurs	✗	✗	✗	✓
Voir statistiques	Limited	Limited	✓	✓

8. Fonctionnalités Avancées

Intégrations Pédagogiques

- **LOMFR** : Référentiel de compétences
- **Export PDF** : Bulletins, certificats
- **API Pronote** : Synchronisation (optionnelle)
- **QRCodes** : Présence, documents

Fonctionnalités Intelligentes

- **Alertes automatiques** : Chute de notes, absentéisme
- **Recommandations** : Ressources pédagogiques
- **Analytics prédictifs** : Risque de décrochage
- **Chatbots** : Assistance FAQ

Modules Spécialisés

- **Orientation** : Suivi Parcoursup
- **Vie de classe** : Délégués, projets
- **Documentation** : CDI numérique
- **Restaurant scolaire** : Menus, réservations

9. Critères d'Évaluation

Obligatoires (100 points)

Critère	Détails	Pondération
Fonctionnel	Gestion notes, communication, emploi du temps	35%
Sécurité	Protection données, authentification multi-rôle	25%
Performance	Rapidité interface, calculs temps réel	15%
UX/UI	Interface adaptée à chaque profil	15%
Code qualité	Architecture clean, tests, documentation	10%

Bonus (+25 points max)

- Tests complets (unitaires, intégration, widget) : +10%
- Module compétences LOMFR : +5%
- Synchronisation Pronote : +5%
- Mode hors-ligne avancé : +5%

10. Planning de Développement

Phase 1 - Cadrage (Semaine 1)

- Architecture technique et modèles de données
- Setup environnement et CI/CD
- Authentification multi-rôles

Phase 2 - Core Élève (Semaines 2-3)

- Tableau de bord élève/parent
- Consultation notes et statistiques
- Emploi du temps et devoirs

Phase 3 - Core Professeur (Semaines 4-5)

- Saisie et gestion des notes
- Outils pédagogiques
- Communication avec les familles

Phase 4 - Administration (Semaine 6)

- Backoffice administration
- Reporting et analytics
- Gestion des utilisateurs

Phase 5 - Finalisation (Semaine 7)

- Optimisations performance
- Tests de charge et sécurité
- Documentation et déploiement

11. Métriques de Succès

Techniques

- Temps de réponse < 1 seconde
- Disponibilité > 99.8%
- Support 1000+ utilisateurs simultanés

Pédagogiques

- Réduction temps de saisie notes > 40%
- Augmentation communication famille-école > 60%
- Satisfaction utilisateurs > 4.5/5

Administratifs

- Réduction impression bulletins > 80%
- Automatisation tâches répétitives > 50%
- Conformité RGPD et protection données

12. Livrables

Développement

- Code source documenté et versionné
- API documentation (Swagger)
- Environnements de test et production
- Scripts de déploiement et migration

Design & UX

- Design system complet
- Maquettes interactives toutes interfaces
- Guide d'utilisation par profil

Qualité & Sécurité

- Audit de sécurité et RGPD
- Plan de test et résultats
- Documentation technique et utilisateur
- Procédures de sauvegarde et recovery

Ce cahier des charges constitue la base contractuelle pour le développement de l'application DEVMOB-EduLycee. Toute modification des spécifications fera l'objet d'un avenant formel validé par l'établissement et l'équipe de développement.