

Application Mobile

Application Mobile de Gestion des Absences

Présentation

Cette application mobile Android permet aux enseignants de gérer les absences des élèves, consulter leurs classes, et accéder à diverses statistiques. Elle est conçue pour fonctionner en complément d'une API REST qui gère les données côté serveur.

Informations de connexion

Pour tester l'application, utilisez les identifiants suivants :

- **Email/Username** : teacher@teacher.fr
- **Mot de passe** : admin123

Architecture de l'application

L'application est développée selon le pattern MVVM (Model-View-ViewModel) et utilise les composants d'architecture Android :

- **Model** : Représente les données et la logique métier
- **View** : Activités et fragments qui affichent l'interface utilisateur
- **ViewModel** : Gère les données liées à l'UI et les interactions avec le modèle

- **Repository** : Couche intermédiaire qui gère les sources de données (API, base de données locale)

Structure des packages

```
com.example.exam/  
├─ adapters/      # Adaptateurs pour les RecyclerViews  
├─ api/           # Services API et configuration Retrofit  
├─ models/       # Classes de modèles de données  
├─ repository/   # Repositories pour accéder aux données  
├─ ui/           # Activités et fragments organisés par fonctionnalité  
│ └─ absences/   # Gestion des absences  
│ └─ classes/    # Gestion des classes  
│ └─ home/       # Écran d'accueil et tableau de bord  
│ └─ ...  
└─ utils/        # Classes utilitaires
```

Fonctionnalités principales

1. Authentification

- Connexion avec email/mot de passe
- Stockage sécurisé du token d'authentification
- Déconnexion

2. Tableau de bord (Dashboard)

- Vue d'ensemble des statistiques (nombre d'élèves, classes, absences)
- Graphique des absences par mois
- Liste des absences récentes
- Fonctionnalité de rafraîchissement (swipe-to-refresh)

3. Gestion des classes

- Liste des classes de l'enseignant
- Détails d'une classe avec liste des élèves
- Statistiques par classe (taux d'absence, nombre d'élèves)

4. Gestion des absences

- Consultation des absences
- Saisie d'une nouvelle absence
- Possibilité de joindre un justificatif (document)
- Filtrage des absences

Flux d'utilisation

Connexion et navigation

1. L'utilisateur lance l'application et arrive sur l'écran de connexion
2. Après authentification réussie, l'utilisateur est redirigé vers le tableau de bord
3. La navigation entre les différentes sections se fait via le menu latéral (Navigation Drawer)

Consultation des classes

1. L'utilisateur accède à la section "Classes" depuis le menu
2. La liste des classes s'affiche avec des informations de base
3. En cliquant sur une classe, l'utilisateur accède aux détails et à la liste des élèves
4. Depuis cette vue, l'utilisateur peut saisir une absence ou générer un rapport

Saisie d'une absence

1. Depuis la liste des classes ou les détails d'une classe, l'utilisateur peut cliquer sur "Saisir une absence"
2. Un formulaire s'affiche permettant de sélectionner l'élève, la date, le statut et la raison
3. L'utilisateur peut optionnellement joindre un document justificatif

- Après validation, l'absence est enregistrée et l'utilisateur est redirigé vers l'écran précédent

Technologies utilisées

Bibliothèques principales

- **AndroidX** : Composants d'architecture Android (ViewModel, LiveData)
- **Material Design** : Composants UI modernes et cohérents
- **Retrofit** : Client HTTP pour les appels API
- **OkHttp** : Client HTTP sous-jacent avec intercepteurs pour logging
- **Gson** : Sérialisation/désérialisation JSON
- **MPAndroidChart** : Visualisation de données sous forme de graphiques
- **Lottie** : Animations vectorielles
- **Shimmer** : Effets de chargement
- **SwipeRefreshLayout** : Fonctionnalité de rafraîchissement par glissement

Gestion des données

- **SharedPreferences** : Stockage des informations de session (token, identifiants)
- **LiveData** : Notification des changements de données aux composants UI
- **ViewModel** : Persistance des données lors des changements de configuration

Détails techniques

Authentification et sécurité

L'application utilise un système d'authentification par token JWT :

1. L'utilisateur s'authentifie avec ses identifiants
2. Le serveur renvoie un token JWT
3. Ce token est stocké dans les SharedPreferences
4. Un intercepteur OkHttp ajoute automatiquement le token à chaque requête API

Gestion des erreurs

L'application gère différents types d'erreurs :

- Erreurs réseau (pas de connexion)
- Erreurs d'authentification (token expiré, identifiants incorrects)
- Erreurs serveur (500, 404, etc.)

Des écrans d'erreur spécifiques sont affichés avec possibilité de réessayer.

Optimisations UI/UX

- **Shimmer Effect** : Affichage d'un effet de chargement pendant les requêtes API
- **Animations** : Transitions fluides entre les écrans
- **Swipe-to-refresh** : Actualisation des données par glissement vers le bas
- **Mise en cache** : Certaines données sont mises en cache pour améliorer les performances

API et endpoints

L'application communique avec une API REST hébergée à l'adresse `https://api.slam.qoyri.fr/api/`.

Principaux endpoints utilisés

- **Authentification** : `/auth/login`
- **Dashboard** : `/teacher/dashboard`
- **Classes** : `/teacher/classes`
- **Détails d'une classe** : `/teacher/classes/{id}`
- **Absences** : `/teacher/absences`
- **Création d'absence** : `/teacher/absences/create`

Installation et déploiement

Prérequis

- Android Studio 4.0+
- JDK 11+
- SDK Android 33+ (Android 13)

Compilation et installation

1. Cloner le dépôt
2. Ouvrir le projet dans Android Studio
3. Synchroniser avec Gradle
4. Compiler et installer sur un appareil ou émulateur

Configuration

L'URL de l'API est configurée dans la classe `ApiClient.java`. Pour changer l'environnement (développement, production), modifiez la constante `BASE_URL`.

Dépannage

Problèmes courants

1. **Erreur de connexion** : Vérifiez votre connexion internet et que l'API est accessible
2. **Erreur d'authentification** : Vérifiez vos identifiants ou réessayez de vous connecter
3. **Données non affichées** : Utilisez la fonction de rafraîchissement (swipe-to-refresh)

Évolutions futures

- Implémentation du mode hors ligne avec synchronisation
- Ajout de notifications push pour les nouvelles absences
- Intégration de la biométrie pour l'authentification
- Support du mode sombre

Revision #1

Created 16 May 2025 17:51:39 by qoyri

Updated 16 May 2025 17:51:53 by qoyri