

# Documentation TeacherAssistant - Diagrammes de Séquence

## Vue d'ensemble de l'application

**TeacherAssistant** est une application mobile hybride (iOS principalement) permettant aux enseignants de gérer leur classe, élèves, matières et évaluations. L'application fonctionne entièrement en temps réel avec synchronisation automatique.

## Architecture générale

- **Client** : Application mobile hybride (mode portrait uniquement)
- **Authentification** : Google Sign-In avec rafraîchissement automatique du token
- **Backend** : TeacherAssistantAPI (REST API)
- **Stockage** : Serveur uniquement (temps réel, pas de stockage local)
- **Autorisation** : Token Google → email utilisateur → droits sur classe spécifique

## Contraintes fonctionnelles

- Un utilisateur = une classe à la fois
- Une classe = une période active à la fois
- Photos : 1 maximum par interrogation/élève
- Moyennes calculées par l'API en temps réel
- Export bulletins en PDF

---

## 1. Flux d'authentification et autorisation

mermaid

## sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant Google as Google Auth

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Lance l'application

App->>U: Affiche écran de connexion

U->>App: Clique "Se connecter avec Google"

App->>Google: Initie Google Sign-In

Google->>U: Affiche interface d'authentification

U->>Google: Saisit identifiants

Google->>App: Retourne token + email

Note over App: Stockage token pour auto-refresh

App->>API: POST /auth/verify (token)

API->>Google: Vérifie validité token

Google->>API: Confirme token + email

API->>DB: Recherche utilisateur par email

DB->>API: Retourne profil + droits classe

API->>App: Session + droits utilisateur

App->>U: Redirige vers tableau de bord

## 2. Gestion du profil utilisateur

mermaid

## sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Accède au profil

App->>API: GET /user/profile (token)

API->>Google: Vérifie token

API->>DB: Récupère profil utilisateur

DB->>API: Données profil

API->>App: Profil utilisateur

App->>U: Affiche profil actuel

alt Modification profil

U->>App: Modifie pseudo/image

App->>API: PUT /user/profile (token, données, image?)

API->>Google: Vérifie token + droits

API->>DB: Met à jour profil

DB->>API: Confirmation

API->>App: Profil mis à jour

App->>U: Confirme modification

end

## 3. Création et configuration de classe

mermaid

## sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Crée nouvelle classe

App->>U: Formulaire classe

U->>App: Saisit infos classe

App->>API: POST /classes (token, infos classe)

API->>Google: Vérifie token

API->>DB: Crée classe + lie à utilisateur

DB->>API: Classe créée

API->>App: Confirmation + ID classe

App->>U: Classe créée avec succès

## loop Ajout matières

U->>App: Ajoute matière

App->>API: POST /classes/{id}/subjects (token, matière, note\_max)

API->>DB: Ajoute matière à la classe

DB->>API: Confirmation

API->>App: Matière ajoutée

end

## loop Ajout élèves

U->>App: Ajoute élève

App->>API: POST /classes/{id}/students (token, infos élève)

API->>DB: Ajoute élève à la classe

DB->>API: Confirmation

API->>App: Élève ajouté

end

## 4. Gestion des périodes

mermaid

#### sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Démarre nouvelle période

App->>API: POST /classes/{id}/periods (token)

API->>Google: Vérifie token + droits classe

API->>DB: Crée période pour classe

DB->>API: Période créée

API->>App: Confirmation période active

App->>U: Interface période active

Note over U,DB: Une seule période active par classe

## 5. Saisie des interrogations avec photos

mermaid

#### sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Sélectionne matière + élève

App->>U: Interface saisie interrogation

U->>App: Saisit note + prend photo (optionnel)

alt Avec photo

App->>API: POST /interrogations (token, élève\_id, matière\_id, note, photo)

API->>Google: Vérifie token + droits

API->>API: Stockage photo sur serveur

API->>DB: Sauvegarde interrogation + lien photo

Note over API: 1 photo max par interrogation/élève

else Sans photo

App->>API: POST /interrogations (token, élève\_id, matière\_id, note)

API->>DB: Sauvegarde interrogation

end

DB->>API: Confirmation + recalcul moyennes

API->>App: Interrogation sauvée + nouvelles moyennes

App->>U: Mise à jour interface temps réel

## 6. Ajout de remarques et appréciations

mermaid

sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Sélectionne élève

App->>U: Interface remarques

U->>App: Saisit remarque (progrès, comportement...)

App->>API: POST /students/{id}/remarks (token, remarque, type)

API->>Google: Vérifie token + droits

API->>DB: Sauvegarde remarque

DB->>API: Confirmation

API->>App: Remarque ajoutée

App->>U: Mise à jour interface

## 7. Consultation bulletin en cours de période

mermaid

sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Accède gestion classe

U->>App: Sélectionne élève

App->>API: GET /students/{id}/bulletin/current (token)

API->>Google: Vérifie token + droits

API->>DB: Récupère interrogations + calcule moyennes

DB->>API: Données bulletin temps réel

API->>App: Bulletin avec moyennes actuelles

App->>U: Affiche bulletin en cours

**Note over** API,DB: Moyennes calculées en temps réel par l'API<br/>Normalisation selon note\_max par matière

## 8. Clôture de période et bulletin final

mermaid

## sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Clique "Clôturer période"

App->>U: Liste des élèves à clôturer

loop Pour chaque élève

U->>App: Sélectionne élève

App->>API: GET /students/{id}/bulletin/final (token)

API->>DB: Récupère données complètes

DB->>API: Bulletin + moyennes finales

API->>App: Bulletin pré-rempli

App->>U: Interface bulletin final

U->>App: Ajoute commentaires finaux

App->>API: PUT /students/{id}/bulletin/finalize (token, commentaires)

API->>DB: Finalise bulletin élève

DB->>API: Élève clôturé

Note over DB: Plus d'ajout d'interrogations possible<br/>pour cet élève dans cette période  
end

API->>App: Période totalement clôturée

App->>U: Confirmation clôture

## 9. Export des bulletins PDF

mermaid

## sequenceDiagram

participant U as Utilisateur

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant DB as Database

U->>App: Demande export bulletin

App->>U: Options d'export (élève, classe complète)

U->>App: Sélectionne option

alt Export individuel

App->>API: GET /students/{id}/bulletin/export (token, format=PDF)

else Export classe complète

App->>API: GET /classes/{id}/bulletins/export (token, format=PDF)

end

API->>Google: Vérifie token + droits

API->>DB: Récupère données bulletins

DB->>API: Données complètes

API->>API: Génération PDF

API->>App: Fichier PDF généré

App->>U: Téléchargement/Partage PDF

Note over API: Génération PDF côté serveur<br/>Format standardisé avec moyennes<br/>et commentaires

## 10. Synchronisation temps réel

mermaid



## sequenceDiagram

participant App as TeacherAssistant App

participant API as TeacherAssistantAPI

participant Google as Google Auth

loop Rafraîchissement automatique token

App->>Google: Refresh token automatique

Google->>App: Nouveau token valide

Note over App: Token mis à jour pour les prochaines requêtes

end

loop Synchronisation données

App->>API: Requêtes avec token à jour

API->>Google: Vérification token

alt Token valide

Google->>API: Token confirmé

API->>App: Données actualisées

else Token expiré

API->>App: Erreur 401 - Token expiré

App->>Google: Demande refresh token

Google->>App: Nouveau token

App->>API: Nouvelle requête avec token fresh

end

end

## Endpoints API principaux

### Authentification

- `POST /auth/verify` - Vérification token Google
- `GET /auth/refresh` - Rafraîchissement session

### Profil utilisateur

- `GET /user/profile` - Récupération profil
- `PUT /user/profile` - Modification profil

### Gestion classes

- `POST /classes` - Création classe
- `GET /classes/{id}` - Détails classe
- `POST /classes/{id}/subjects` - Ajout matière
- `POST /classes/{id}/students` - Ajout élève

## Gestion périodes

- `POST /classes/{id}/periods` - Démarrage période
- `PUT /periods/{id}/close` - Clôture période

## Interrogations et évaluations

- `POST /interrogations` - Ajout interrogation + photo
- `GET /students/{id}/interrogations` - Liste interrogations élève
- `POST /students/{id}/remarks` - Ajout remarques

## Bulletins

- `GET /students/{id}/bulletin/current` - Bulletin en cours
  - `GET /students/{id}/bulletin/final` - Bulletin final
  - `PUT /students/{id}/bulletin/finalize` - Finalisation bulletin
  - `GET /students/{id}/bulletin/export` - Export PDF individuel
  - `GET /classes/{id}/bulletins/export` - Export PDF classe
- 

## Considérations techniques

### Sécurité

- Vérification token Google à chaque requête API
- Validation des droits utilisateur sur la classe
- Upload sécurisé des photos (1 max par interrogation/élève)

### Performance

- Calculs de moyennes optimisés côté API
- Pas de cache local, synchronisation temps réel
- Gestion des photos avec compression/redimensionnement

### Gestion d'erreurs

- Token expiré : refresh automatique
- Perte de connexion : retry automatique
- Conflits de données : synchronisation forcée

### Contraintes métier

- Un utilisateur = une classe active

- Une classe = une période active
- Clôture définitive empêche nouvelles interrogations
- Moyennes normalisées selon note\_max par matière