

Guide de Crédation d'Utilisateurs

Application Plan de Classe

Version 2.0 - Décembre 2025

1. Vue d'ensemble

Le système de gestion des utilisateurs de l'application Plan de Classe est **complètement indépendant de auth.users** de Supabase. Cette architecture permet une gestion simplifiée et un contrôle total sur les données utilisateurs.

Principes clés

- Tous les utilisateurs sont stockés dans la table **profiles** avec leurs identifiants
- Les mots de passe sont hachés en SHA256 côté serveur
- Chaque établissement possède son propre code d'accès unique
- Les identifiants sont générés automatiquement au format *prenom.nom123*

2. Structure des tables

Le système repose sur plusieurs tables interconnectées dans Supabase :

2.1 Table profiles (table principale)

Cette table centralise tous les comptes utilisateurs avec leurs informations d'authentification.

Colonne	Type	Description
<code>id</code>	<code>uuid</code>	Clé primaire auto-générée
<code>establishment_id</code>	<code>uuid</code>	Référence vers <code>establishments.id</code>
<code>role</code>	<code>enum</code>	'vie-scolaire' 'professeur' 'delegue'
<code>username</code>	<code>text</code>	Identifiant unique de connexion
<code>password_hash</code>	<code>text</code>	Mot de passe haché SHA256
<code>first_name</code>	<code>text</code>	Prénom de l'utilisateur
<code>last_name</code>	<code>text</code>	Nom de famille
<code>email</code>	<code>text</code>	Adresse email (optionnel)
<code>phone</code>	<code>text</code>	Téléphone (optionnel)
<code>can_create_subrooms</code>	<code>boolean</code>	Permission de créer des sous-salles
<code>created_at</code>	<code>timestamp</code>	Date de création
<code>updated_at</code>	<code>timestamp</code>	Date de dernière modification

2.2 Table students (élèves)

Stocke les informations spécifiques aux élèves délégues et éco-délégues.

Colonne	Type	Description
<code>id</code>	<code>uuid</code>	Clé primaire auto-générée
<code>profile_id</code>	<code>uuid</code>	Référence vers <code>profiles.id</code>
<code>establishment_id</code>	<code>uuid</code>	Référence vers <code>establishments.id</code>
<code>first_name</code>	<code>text</code>	Prénom
<code>last_name</code>	<code>text</code>	Nom
<code>email</code>	<code>text</code>	Email (optionnel)

Colonne	Type	Description
phone	text	Téléphone (optionnel)
class_id	uuid	Référence vers classes.id
role	enum	'delegue' 'eco-delegue'
can_create_subrooms	boolean	Permission sous-salles

2.3 Table teachers (professeurs)

Contient les données des professeurs avec leur matière enseignée.

Colonne	Type	Description
id	uuid	Clé primaire auto-générée
profile_id	uuid	Référence vers profiles.id
establishment_id	uuid	Référence vers establishments.id
first_name	text	Prénom
last_name	text	Nom
email	text	Email (optionnel)
subject	text	Matière enseignée

2.4 Table teacher_classes (liaison professeurs-classes)

Table de liaison permettant d'associer un professeur à plusieurs classes.

Colonne	Type	Description
id	uuid	Clé primaire auto-générée
teacher_id	uuid	Référence vers teachers.id
class_id	uuid	Référence vers classes.id
created_at	timestamp	Date de création

3. Comment créer un utilisateur

3.1 Prérequis : Exécuter le script SQL

Avant toute création d'utilisateur, exécutez le script **013_remove_auth_dependency.sql** dans l'éditeur SQL de Supabase. Ce script :

- Supprime la dépendance à *auth.users*
- Crée la fonction *hash_password()*
- Met à jour les politiques RLS (Row Level Security)

3.2 Créer un élève via l'interface

1. Accédez à **Dashboard → Élèves**
2. Cliquez sur le bouton "**Ajouter un élève**"
3. Remplissez le formulaire avec les informations requises
4. Validez en cliquant sur "**Ajouter**"

Champs du formulaire élève :

Champ	Statut	Description
Prénom	Requis	Prénom de l'élève
Nom	Requis	Nom de famille
Email	Optionnel	Adresse email
Téléphone	Optionnel	Numéro de téléphone
Classe	Requis	Sélection dans la liste déroulante
Rôle	Requis	Délégué ou Éco-délégué
Permission sous-salles	Optionnel	Toggle activé/désactivé

Ce qui se passe automatiquement :

- Un enregistrement est créé dans **profiles** avec username généré (format: prenom.nom123)
- Un mot de passe aléatoire de 8 caractères est généré et haché en SHA256
- Un enregistrement lié est créé dans **students**
- **Les identifiants sont affichés dans une notification - notez-les immédiatement !**

3.3 Créer un professeur via l'interface

1. Accédez à **Dashboard → Professeurs**
2. Cliquez sur "**Ajouter un professeur**"
3. Complétez les informations (Prénom, Nom, Email, Matière, Classes)
4. Validez la création

Processus automatique :

- Création dans **profiles** avec le rôle 'professeur'
- Création dans **teachers** avec la matière
- Création des liaisons dans **teacher_classes** pour chaque classe sélectionnée

3.4 Créer un compte Vie Scolaire (SQL)

Les comptes vie-scolaire doivent être créés directement en SQL dans l'éditeur Supabase :

```
INSERT INTO profiles (
    establishment_id,
    role,
    username,
    password_hash,
    first_name,
    last_name,
    email,
    can_create_subrooms
) VALUES (
    'ID_ETABLISSEMENT',          -- UUID de l'établissement
    'vie-scolaire',              -- Rôle fixe
    'vs.nom',                   -- Format recommandé: vs.prenom
    hash_password('MotDePasse'), -- Mot de passe sécurisé
    'Prénom',
    'Nom',
    'email@établissement.fr',
    true                         -- Permission sous-salles
);
```

4. Connexion des utilisateurs

4.1 Connexion standard (Élèves et Professeurs)

5. Accédez à la page de connexion de l'application
6. Saisissez le **code établissement** (ex: stm001, vh001)
7. Sélectionnez votre **rôle** (Délégué ou Professeur)
8. Entrez votre **identifiant** (ex: jean.dupont123)
9. Saisissez votre **mot de passe**

4.2 Connexion administrateur

~~Pour les accès administrateurs, utilisez le bouton "Connexion Admin" et entrez le code approprié :~~

Code Admin	Rôle	Établissement
cpdc001	Délégué	ST-MARIE
cpdc002	Professeur	ST-MARIE
cpdc003	Vie Scolaire	ST-MARIE

Se référer à l'annexe A. Le bouton « connexion admin » a été supprimé.

5. Gestion des identifiants

5.1 Consulter/Modifier les identifiants d'un élève

10. Dans la liste des élèves, cliquez sur le menu "..."
11. Sélectionnez "**Configurer l'accès**"
12. Options disponibles : voir l'identifiant, modifier, générer nouveau mot de passe, imprimer en PDF

5.2 Réinitialiser un mot de passe (SQL)

Pour réinitialiser un mot de passe manuellement :

```
UPDATE profiles
SET password_hash = hash_password('nouveau_mot_de_passe')
WHERE username = 'identifiant_utilisateur';
```

6. Vérification des données

6.1 Voir tous les profils

```
SELECT id, username, role, first_name, last_name, establishment_id
FROM profiles
ORDER BY created_at DESC;
```

6.2 Voir les élèves avec leurs profils

```
SELECT
  s.*,
  p.username,
  p.role as profile_role,
  c.name as class_name
FROM students s
JOIN profiles p ON s.profile_id = p.id
LEFT JOIN classes c ON s.class_id = c.id
ORDER BY s.last_name;
```

6.3 Voir les professeurs avec leurs classes

```
SELECT
  t.*,
  p.username,
  array_agg(c.name) as classes
FROM teachers t
JOIN profiles p ON t.profile_id = p.id
LEFT JOIN teacher_classes tc ON t.id = tc.teacher_id
LEFT JOIN classes c ON tc.class_id = c.id
GROUP BY t.id, p.username
ORDER BY t.last_name;
```

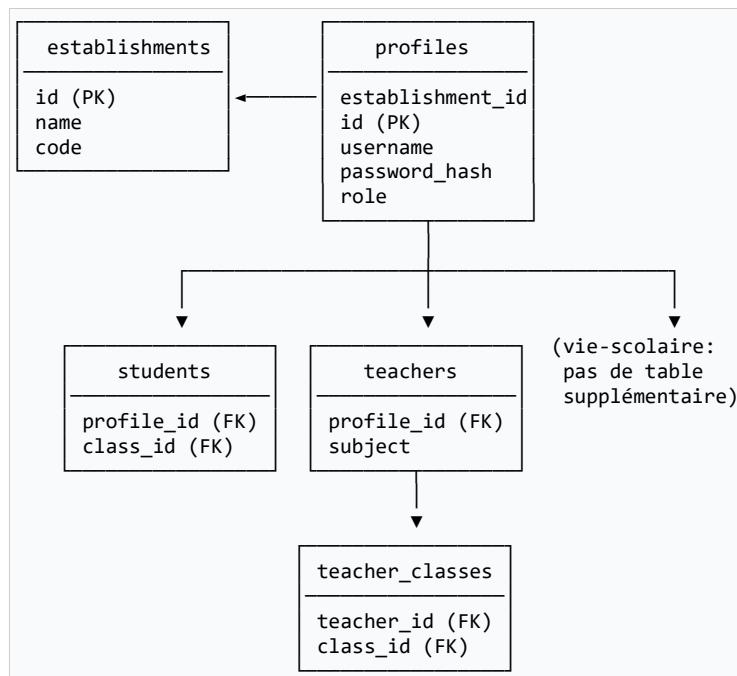
7. Points importants à retenir

⚠️ IMPORTANT

- Notez les identifiants lors de la création - ils ne seront plus affichés ensuite
- Les mots de passe sont hachés - impossible de les récupérer, uniquement les réinitialiser
- Les usernames sont uniques - pas de doublons possibles dans tout le système
- Pas besoin de Supabase Auth - tout fonctionne avec la base de données uniquement
- Le module lib/user-management.ts gère automatiquement toute la logique de création

8. Schéma des relations

Voici comment les tables sont interconnectées :



Annexe A – Analyse de l'Architecture d'Authentification

Note technique – Décembre 2025

A.1 Constat : Formulaire de connexion actuel

Le formulaire de connexion actuel demande **4 informations**:

1. Code établissement (ex: stm001)
2. Rôle (selecteur : Vie Scolaire / Professeur / Délégué)
3. Identifiant (username)
4. Mot de passe

Le bouton « Connexion Admin » a été supprimé lors de la migration du 05/12/2025, mais le formulaire conserve les champs « Code établissement » et « Rôle ».

A.2 Problème : Redondance des informations

Ces informations sont **redondantes** avec ce qui est déjà stocké dans la base de données :

Information demandée	Déjà présente dans profiles
Code établissement	profiles.establishment_id
Rôle	profiles.role
Username	profiles.username
Mot de passe	profiles.password_hash

Le code actuel (custom-auth.ts) utilise le rôle sélectionné pour router vers différentes tables :

- vie-scolaire → cherche dans profiles
- professeur → cherche dans teachers
- delegue → cherche dans students

A.3 Impacts de l'architecture actuelle

Problème	Description
UX dégradée	L'utilisateur doit se souvenir de son rôle et du code établissement alors que ces informations lui sont attribuées à la création
Risque d'erreur	Un professeur qui sélectionne « Délégué » par erreur ne pourra pas se connecter (même avec les bons credentials)
Duplication	Les credentials sont stockés à la fois dans profiles ET dans teachers/students (colonnes username, password_hash dupliquées)
Sécurité	Le rôle devrait être une propriété intrinsèque de l'utilisateur, pas un choix au moment de la connexion

A.4 Solution standard recommandée

L'approche conforme aux standards d'authentification serait de simplifier le formulaire à 2 champs :

CONNEXION STANDARD	
Identifiant : []	
Mot de passe : []	

Le système devrait :

1. Rechercher l'utilisateur **uniquement par username** (unique dans toute la base)
2. Vérifier le mot de passe

3.Récupérer automatiquement: rôle, établissement, permissions depuis profiles

A.5 Comparatif actuel vs. proposé

Aspect	Actuel	Proposé
Champs login	4	2
Tables à interroger	3 (selon rôle)	1 (profiles)
Duplication données	Oui	Non
Risque erreur utilisateur	Élevé	Nul

A.6 Recommandations

Court terme : Documenter la limitation actuelle et former les utilisateurs sur le bon usage du formulaire.

Moyen terme : Refactorer custom-auth.ts pour :

- Supprimer les paramètres establishmentCode et role du formulaire
- Centraliser l'authentification sur la table profiles uniquement
- Supprimer les colonnes username/password_hash de teachers et students
- Utiliser profile_id comme seule référence pour les credentials

Annexe ajoutée le 08 décembre 2025