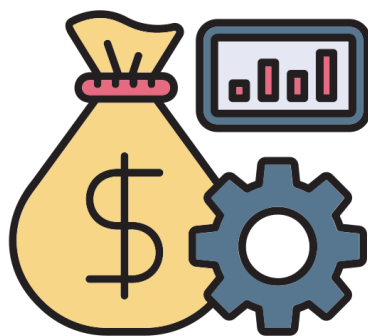


Documentation CAF

Système de Gestion des Cotisations

Cotisations AD Fidèles



Version 1.0

27 juillet 2025

Application développée avec KivyMD
Framework Python pour interfaces Material Design

Table des matières

1	Vue d'ensemble	2
1.1	Objectifs principaux	2
2	Architecture technique	2
2.1	Technologies utilisées	2
2.2	Structure des fichiers	2
3	Structure de la base de données	3
3.1	Tables utilisateur	3
4	Guide d'utilisation	3
4.1	Première utilisation	3
4.1.1	Lancement de l'application	3
4.1.2	Création d'un compte	3
4.2	Navigation principale	3
4.2.1	Histoire des Cotisations	3
4.2.2	Cotisations	4
4.2.3	Ajouter un membre	4
4.2.4	Compte total	4
4.3	Fonctionnalités avancées	5
4.3.1	Interface utilisateur	5
4.3.2	Sécurité	5
5	Fonctionnalités techniques	5
5.1	Gestion des écrans	5
5.2	Calculs automatiques	5
5.3	Gestion des erreurs	5
6	Installation et déploiement	5
6.1	Prérequis	5
6.2	Structure minimale requise	6
6.3	Lancement	6
7	Maintenance et personnalisation	6
7.1	Ajout de nouvelles fonctionnalités	6
7.2	Sauvegarde des données	6
7.3	Personnalisation de l'interface	6
8	Limitations connues	6
9	Support et développement	7
9.1	Structure du code	7
9.2	Les fonctionnalités prochaines	7
10	Conclusion	7

1 Vue d'ensemble

CAF (Cotisations AD Fidèles) est une application de bureau développée avec KivyMD pour la gestion des cotisations et des contributions financières. L'application permet de gérer plusieurs utilisateurs, leurs mises, cotisations et de générer des rapports détaillés.

1.1 Objectifs principaux

- Gestion centralisée des cotisations
- Suivi des membres et de leurs contributions
- Génération de rapports financiers
- Interface utilisateur moderne et intuitive
- Stockage sécurisé des données

2 Architecture technique

2.1 Technologies utilisées

Framework principal

KivyMD (interface utilisateur Material Design)

Base de données

SQLite3 (stockage local)

Langage

Python 3

Structure

Architecture MVC avec séparation des interfaces (.kv) et de la logique (.py)

2.2 Structure des fichiers

Listing 1 – Arborescence du projet

```
1 CAF /
2 | -- main.py           # Logique principale
3 | -- base.db           # Base de données SQLite
4 | -- Pages /
5 |   | -- Page1.kv      # Ecran d'accueil
6 |   | -- Page2.kv      # Authentification
7 |   | -- Page3.kv      # Historique cotisations
8 |   | -- Page4.kv      # Ajout de membres
9 |   | -- Page5.kv      # Gestion cotisations
10 |   \-- Page6.kv       # Rapports et totaux
11 \-- Assets /
12 | -- Icon.png          # Icône application
13 | -- Cot.jpeg          # Image fond cotisations
14 | -- Lock.jpeg         # Image fond authentification
15 \-- Font.ttf           # Police personnalisée
```

3 Structure de la base de données

3.1 Tables utilisateur

Chaque utilisateur possède sa propre table nommée `{identifiant}_{mot_de_passe}` avec la structure suivante :

Colonne	Type	Description
Name	TEXT	Nom du membre cotisant
Mise	TEXT	Montant de la mise unitaire
Date	TEXT	Date de la cotisation (format MM/DD/YY)
Heur	TEXT	Heure de la cotisation (format HH :MM :SS)
Nbre_de_mise	TEXT	Nombre de mises effectuées
Payer_with	TEXT	Montant payé
Remise	TEXT	Monnaie rendue

4 Guide d'utilisation

4.1 Première utilisation

4.1.1 Lancement de l'application

1. Exécuter `python main.py`
2. Cliquer sur "**Commencer**" depuis l'écran d'accueil
3. Créer un nouveau compte ou se connecter

4.1.2 Création d'un compte

1. Cliquer sur "**New Compte**" depuis l'écran de connexion
2. Remplir les champs :
 - Nom complet
 - Identifiant unique
 - Mot de passe (confirmation requise)
3. Confirmer la création

4.2 Navigation principale

Une fois connecté, quatre options principales sont disponibles :

4.2.1 Histoire des Cotisations

But Consulter l'historique complet des cotisations

Navigation

Mise → Membre → Détails des transactions

Fonctionnalités

- Visualisation par mise

- Filtrage par membre
- Tableau détaillé des transactions

4.2.2 Cotisations

But Enregistrer de nouvelles cotisations

Processus

1. Sélectionner une mise existante
2. Choisir le membre
3. Saisir le nombre de mises
4. Indiquer le montant payé
5. Calcul automatique de la remise

Validation

Confirmation avant enregistrement

4.2.3 Ajouter un membre

But Inscrire de nouveaux membres dans une mise

Informations requises

- Montant de la mise
- Nom et prénom du membre

Note

Un membre ne peut être ajouté qu'une fois par mise

4.2.4 Compte total

Deux types de rapports sont disponibles :

Total sur somme

- Vue d'ensemble par mise
- Nombre de participants par mise
- Total des montants collectés
- Résumé général

Inventaire sur cotisation

- Détail par membre et par mise
- Nombre total de cotisations par membre
- Montants individuels et totaux

4.3 Fonctionnalités avancées

4.3.1 Interface utilisateur

- **Thème** : Basculer entre mode sombre et clair
- **Couleurs** : Choix parmi 19 palettes de couleurs
- **Navigation** : Menu hamburger accessible depuis toutes les pages

4.3.2 Sécurité

- Authentification par identifiant/mot de passe
- Isolation des données par utilisateur
- Validation des saisies

5 Fonctionnalités techniques

5.1 Gestion des écrans

L'application utilise un système de navigation multi-niveaux :

- **Niveau 1** : Pages principales (Page1 à Page6)
- **Niveau 2** : Sous-écrans par page (Page1, Page2, Page3 par page)

5.2 Calculs automatiques

Les formules de calcul utilisées dans l'application :

$$\text{Total de cotisation} = \text{mise unitaire} \times \text{nombre de mises} \quad (1)$$

$$\text{Remise} = \text{montant payé} - \text{total cotisation} \quad (2)$$

$$\text{Condition} : \text{montant payé} \geq \text{total cotisation} \quad (3)$$

5.3 Gestion des erreurs

- Validation des champs obligatoires
- Vérification des formats numériques
- Messages d'erreur informatifs
- Confirmation des actions importantes

6 Installation et déploiement

6.1 Prérequis

Listing 2 – Installation des dépendances

```
1 pip install kivymd
2 pip install kivy
```

6.2 Structure minimale requise

```
1      CAF/  
2      |-- main.py  
3      |-- Pages/  
4      |    \-- (tous les fichiers .kv)  
5      \-- Assets/  
6      \-- (images et police)
```

6.3 Lancement

```
1      python main.py
```

7 Maintenance et personnalisation

7.1 Ajout de nouvelles fonctionnalités

1. **Nouvelle page** : Créer PageX.kv et ajouter la classe correspondante
2. **Nouvelle méthode** : Ajouter dans la classe CAF
3. **Mise à jour de l'interface** : Modifier les fichiers .kv appropriés

7.2 Sauvegarde des données

- La base de données `base.db` contient toutes les informations
- Sauvegarder ce fichier pour préserver les données

7.3 Personnalisation de l'interface

Images

Remplacer les fichiers dans le dossier Assets

Police

Remplacer Font.ttf

Couleurs

Modifier les palettes dans `change_color()`

8 Limitations connues

1. **Stockage local uniquement** : Pas de synchronisation cloud
2. **Un utilisateur à la fois** : Pas de gestion multi-utilisateur simultanée
3. **Pas de suppression** : Aucune fonction de suppression des données
4. **Format de date fixe** : Format américain MM/DD/YY

9 Support et développement

9.1 Structure du code

Classe principale

CAF(MDApp) - Point d'entrée et logique métier

Classes d'écran

Page1 à Page6 - Contrôleurs d'interface

Méthodes utilitaires

Gestion de base de données, navigation, validation

10 Conclusion

Cette documentation couvre les aspects essentiels de l'application CAF. Pour des modifications avancées, consulter directement le code source avec cette documentation comme référence.

L'application CAF représente une solution complète pour la gestion des cotisations, offrant une interface moderne et des fonctionnalités robustes pour le suivi des contributions financières.