



GESTION ASSOCIATIVE

PROJET : NUMÉRISATION DE LA GESTION ADMINISTRATIVE
D'UNE ASSOCIATION - CHINGU



ETUDIANTS :

YAYE AWA DIOUF
FAIZI PINEAU

ANNÉE UNIVERSITAIRE : 2024 - 2025

INTRODUCTION

Dans un contexte où les associations cherchent à moderniser leurs outils de gestion, la mise en place d'un système d'information performant devient un levier essentiel pour optimiser leur fonctionnement.

Ce projet vise à concevoir et structurer une application web de gestion associative nommé **CHINGU_ASSO** permettant de centraliser la gestion des membres, des cotisations, des événements et des participations.

L'objectif est double :

- digitaliser les processus administratifs encore souvent réalisés manuellement,
- et améliorer la visibilité, la fiabilité et la réactivité de l'association dans la gestion de ses activités.

Ce travail a pour ambition d'illustrer une démarche complète, structurée et professionnelle, similaire à celle menée dans les organisations lors de la mise en place d'un nouveau système d'information.

1. PHASE INITIALE



1.1. Cahier de charges



1.2. Étude de faisabilité



1.3 Plan de projet

1.1. CAHIER DES CHARGES

Objectif du système

Numériser et centraliser l'ensemble des processus administratifs afin de :

- Simplifier la gestion des **adhérents**,
- Automatiser le suivi des **cotisations**,
- Organiser efficacement les **événements associatifs**,
- Accroître la visibilité des activités.


Acteurs principaux

- Administrateur de l'association
- Membres (adhérents)
- Trésorier / Bureau
- Responsables d'événements

1.1. CAHIER DES CHARGES



Fonctionnalités attendues

- **Gestion des membres**
 - Création, modification, suppression, listing
 - Import/Export CSV
 - Suivi de l'état d'adhésion
- **Gestion des cotisations**
 - Enregistrement des paiements et Statut : Payée / En retard / A régulariser
 - Historique de paiement
- **Gestion des événements**
 - Création d'un événement
 - Description, lieu, date, nombres places
- Consultation publique / privée
- **Gestion des participations**
 - Inscription à un événement
 - Présence / absence
 - Statistiques d'engagement
- **Gestion des événements**
 - Création d'un événement
 - Description, lieu, date, nombre de places
 - Consultation publique / privée
- **Gestion des participations**
 - Inscription à un événement
- Présence / absence
- Statistiques d'engagement
- **Tableaux de bord**
 - Indicateurs clés :
 - Nombre de membres actifs
 - Taux de cotisation payé
 - Participation aux événements
-  **Contraintes**
 - Interface intuitive
 - Sécurité des données
 - Système accessible sur PC / tablette / mobile
 - Stockage sécurisé (SQLite)

1.2. ÉTUDE DE FAISABILITÉ

Faisabilité Technique

- Framework **Flask** parfaitement adapté
- Possibilité de **déployer sur serveur** mutualisé ou local
- Base de données **SQLite** → robuste, fiable, adaptée au contexte associatif
- Architecture **MVC** = maintenance simplifiée

➔ **Faisabilité validée**

Faisabilité Économique

- Développement non coûteux
- Hébergement minimal (< 50€/an)
- Pas de licences payantes
- Maintenance simple

➔ **Projet économiquement réaliste**

Faisabilité Organisationnelle

- Formation légère pour les membres
- Processus cohérents avec les pratiques existantes
- Amélioration immédiate de la gestion interne

➔ **Projet organisationnellement bénéfique**

2. PHASE ANALYSE



**2.1. SFG → Spécifications
Fonctionnelles Générales**



**2.2. SFD → Spécifications
Fonctionnelles Détaillées**



**2.3. Spécifications
techniques**



**2.4. Spécifications
techniques**

2.1 SFG – SPÉCIFICATIONS FONCTIONNELLES GÉNÉRALES

🎯 Vision globale du système

Le système doit permettre à l'association :

- de gérer ses adhérents,
- de suivre les paiements,
- d'organiser des événements,
- et d'enregistrer les participations.

🌐 Vue d'ensemble

- Le logiciel se décompose en 4 modules principaux :
 - Module Membres
 - Module Cotisations
 - Module Événements
 - Module Participations
- Chaque module comporte :
 - Consultation
 - Ajout
 - Modification
 - Suppression
 - Exportation
 - Statistiques

1.3. PLAN DE PROJET

Découpage en phases

Phase	Durée estimée (semaine)	Livrables
Étude & Analyse	1 à 2	Cahier des charges, SFG
Conception	2	UML, Maquettes
Développement	4	Code fonctionnel
Tests	1	Plan + Rapport de tests
Préparation à la production	1	Guide installation

Ressources :

- 1 développeur
- 1 responsable association pour retours fonctionnels
- 1 référent pour tests

Risques :

- Données membres sensibles → prévoir sauvegardes
- Délais dépendant de la validation du bureau
- Acceptation du changement (transition numérique)

2.2 SFD – SPÉCIFICATIONS FONCTIONNELLES DÉTAILLÉES

✓ Module 1 : Membres

- Fonctions :
 - Créer un membre (nom, email, statut, date adhésion...)
 - Modifier ses informations
 - Désactiver un membre
 - Rechercher un membre
 - Export PDF/Excel
- Règles métiers :
 - Email obligatoire et unique
 - Statut actif = adhésion valide

✓ Module 2 — Cotisations

- Fonctions :
 - Enregistrer un paiement
 - Gérer les statuts (Payé / Non payé / En attente)
 - Éditer reçus
 - Filtrer par année
- Règles :
 - Un membre = 1 cotisation par an
 - Le montant dépend de la politique associative

2.2 SFD – SPÉCIFICATIONS FONCTIONNELLES DÉTAILLÉES

✓ Module 3 — Événements

- **Fonctions :**

- Créer un événement
- Publier/archiver
- Limiter nombre de participants
- Lister événements passés / futurs

- **Règles :**

- Aucun événement ne peut être créé sans date valide
- Nombre de places optionnel mais contrôlé

✓ Module 4 — Participations

- **Fonctions :**

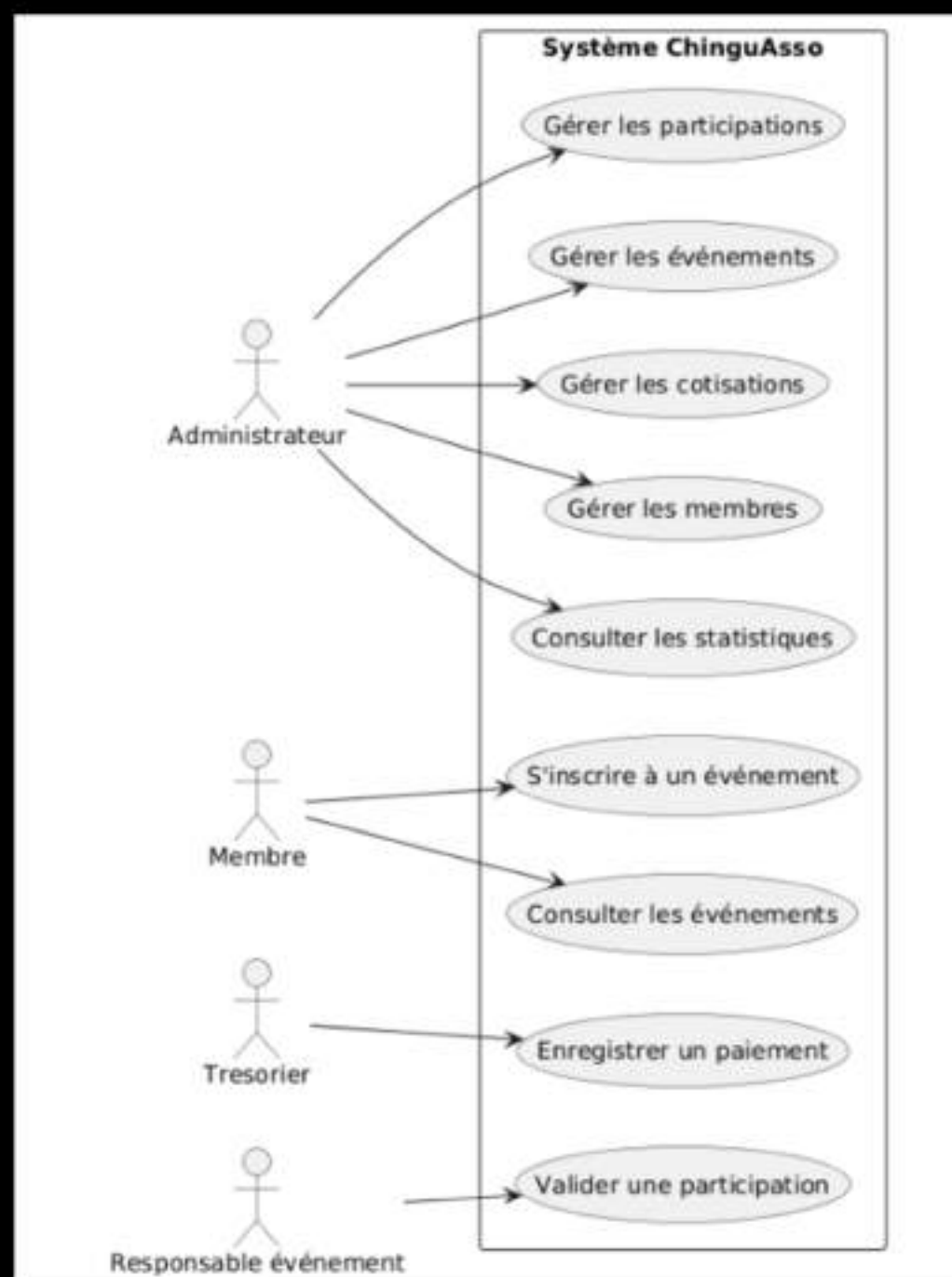
- Inscription d'un membre
- Marquer présence / absence
- Liste participants
- Statistiques

- **Règles :**

- Un membre ne peut s'inscrire qu'une fois par événement
- Si l'événement est complet → inscription refusée

2.2 SFD – SPÉCIFICATIONS FONCTIONNELLES DÉTAILLÉES


✓ Le comportement attendu du système
Étape 1 — Diagramme UML



2.2 SFD – SPÉCIFICATIONS FONCTIONNELLES DÉTAILLÉES

✓ Le comportement attendu du système

Étape 2 — Maquette synthétique



ChinguAsso

DashboardMembresÉvénementsCotisationsParticipations

Dashboard — Vue d'ensemble

Aperçu rapide des activités principales de l'association.

INFO

21 Membres actifs

Nombre total de membres inscrits et actifs dans l'association.

STATS

54.55 % de cotisations payées

Taux global de cotisations réglées pour l'année en cours.

ACTIONS

11 Événements à venir

Événements programmés et non encore passés.

Derniers membres inscrits

Nom	Email	Inscription	Cotisation
Isaac Barnes	carla39@example.net	2025-08-22	✓ Payée
Kaitlin Hampton	angelachavez@example.org	2025-07-19	✓ Payée
Timothy Lewis	smorris@example.org	2025-06-11	✗ Non payée
Angela Moody	smithnatalie@example.com	2025-05-18	✗ Non payée

2.2 SFD – SPÉCIFICATIONS FONCTIONNELLES DÉTAILLÉES

✓ Le comportement attendu du système

Étape 3 — User Flow

1. Ajouter un membre

Tableau de bord → Membres → Ajouter membre → Formulaire
→ Valider

2. Enregistrer une cotisation

Membres → Sélectionner → Cotisations → Ajouter paiement →
Enregistrer

3. Créer un événement

Tableau de bord → Événements → Créer → Formulaire →
Publier

4. Valider une participation

Événements → Détails → Participants → Valider → Confirmé

2.3. SPÉCIFICATIONS TECHNIQUES

Technologies

- Flask (Python),
- SQLite
- HTML / CSS / Python
- Framework : Flask 3.0
- Jinja2 Templates

Sécurité

- Authentification
- Hashing des mots de passe (Bcrypt)
- Validation des formulaires
- Protection CSRF - Cross-Site Request Forgery

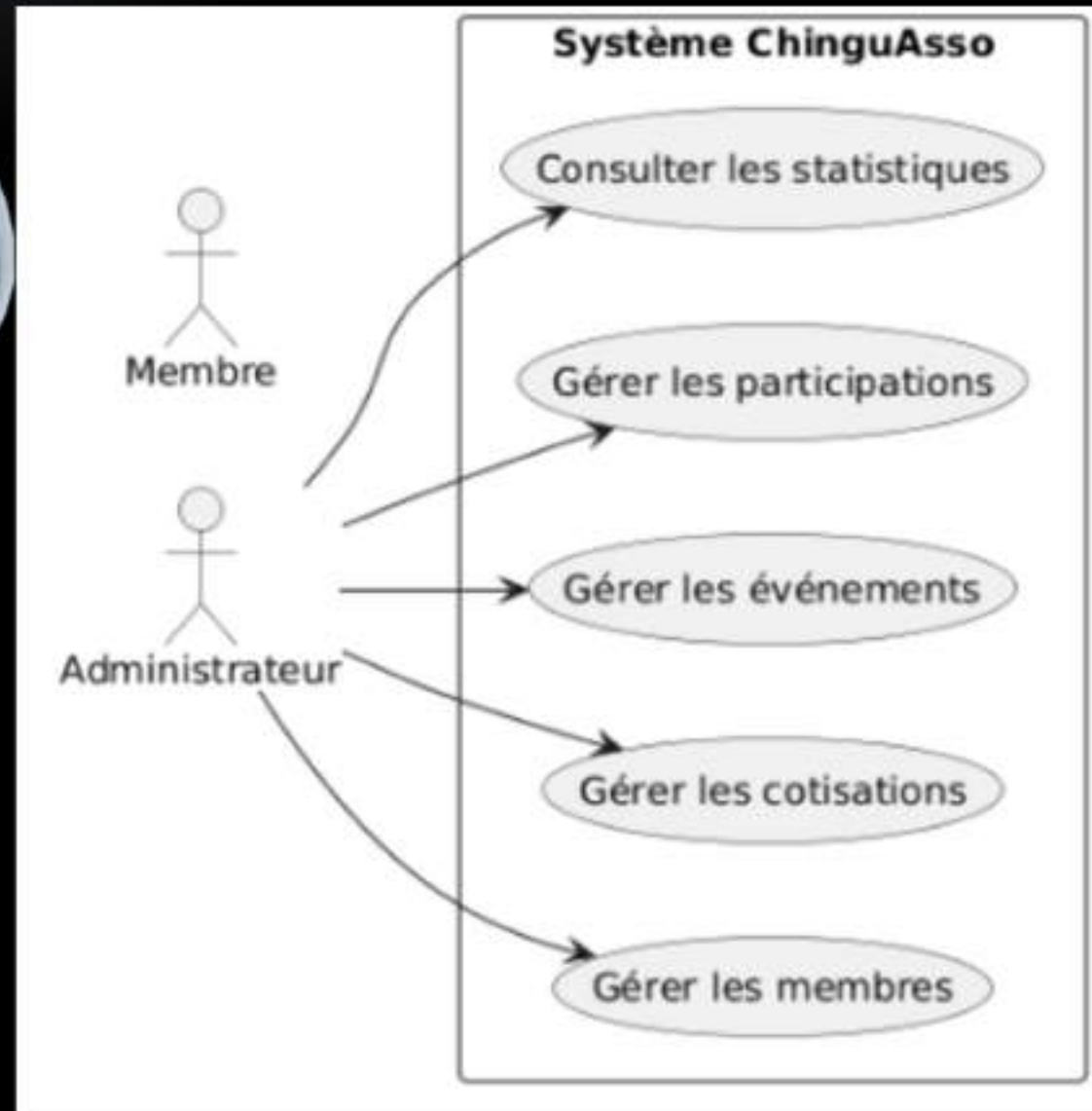
(La protection CSRF empêche un site externe d'envoyer des requêtes à ta place sur un site où tu es connecté.)

Architecture

- MVC
- Routes centralisées (web.php)
- Contrôleurs organisés par module
- Modèles reliés JPA Eloquent

2.4. MODÈLES UML

📌 Diagramme de cas d'utilisation



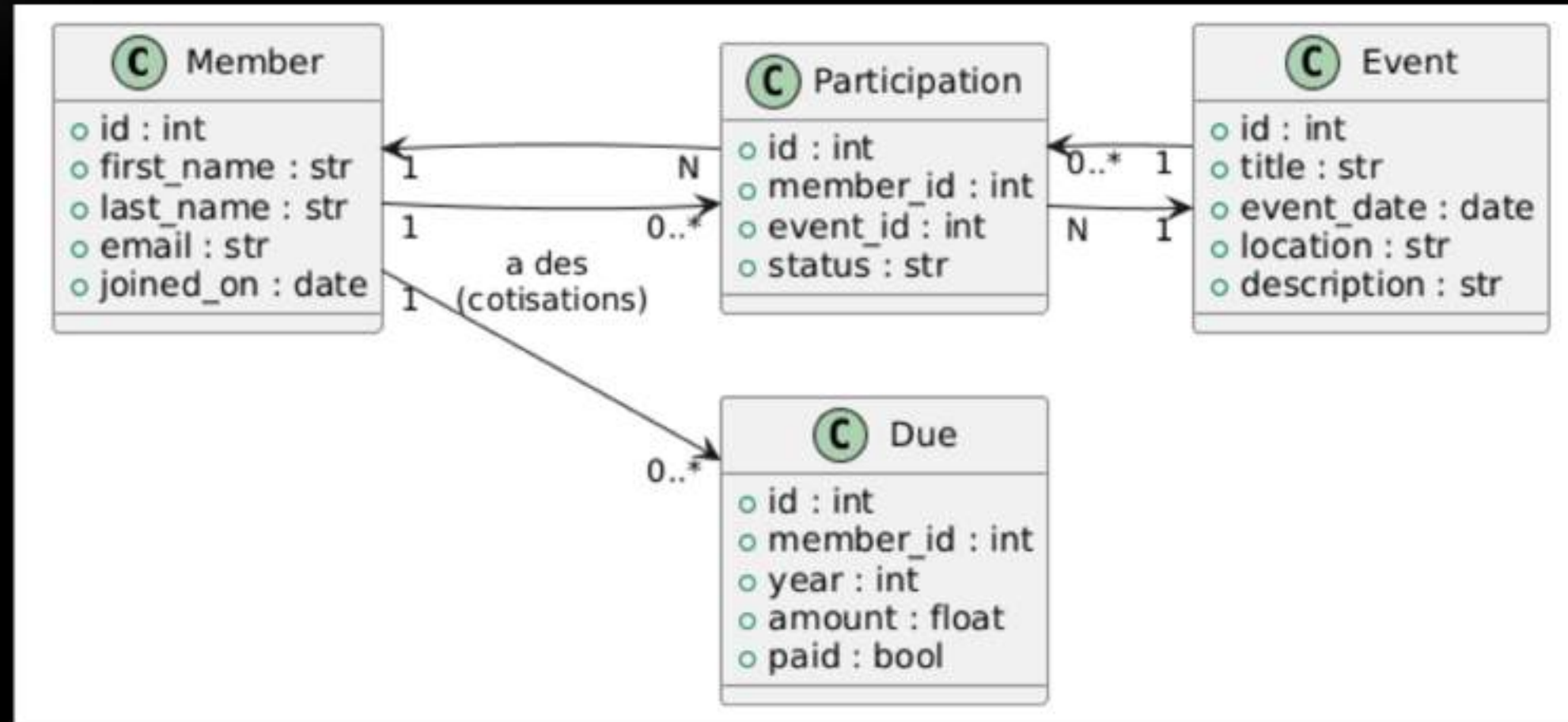
Acteurs : Membre, Administrateur

Cas principaux :

- Gérer membres
- Gérer cotisations
- Gérer événements
- Gérer participations
- Consulter statistiques

2.4. MODÈLES UML

📌 Diagramme de classes (simplifié)



Classes principales :

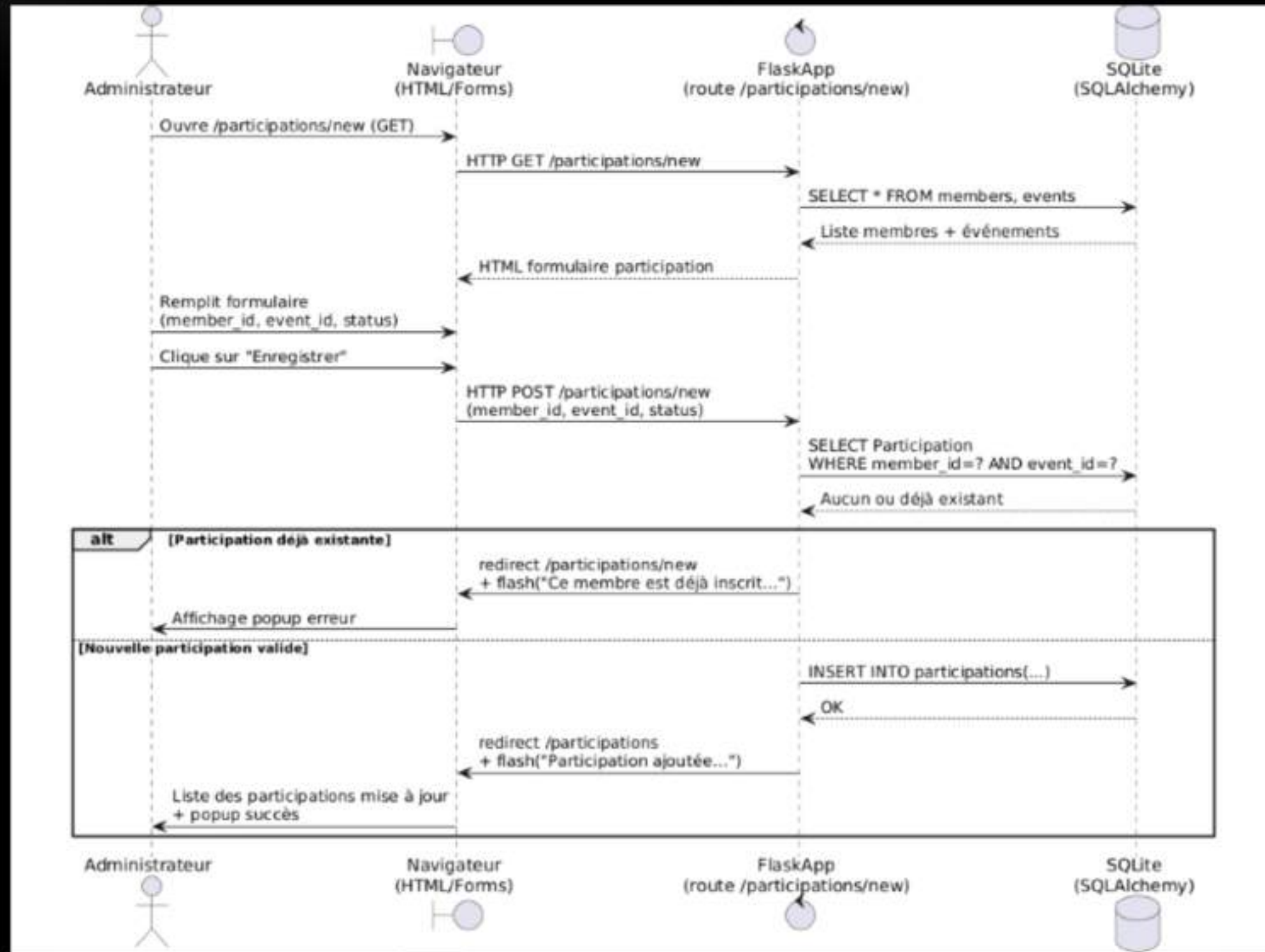
- Membre
- Cotisation
- Événement
- Participation

Relations :

- Membre 1—N Cotisations
- Membre N—N Événements (via Participation)

2.4. MODÈLES UML

📌 Diagramme de séquence (exemple : ajout d'un membre)



3. PHASE CONCEPTION



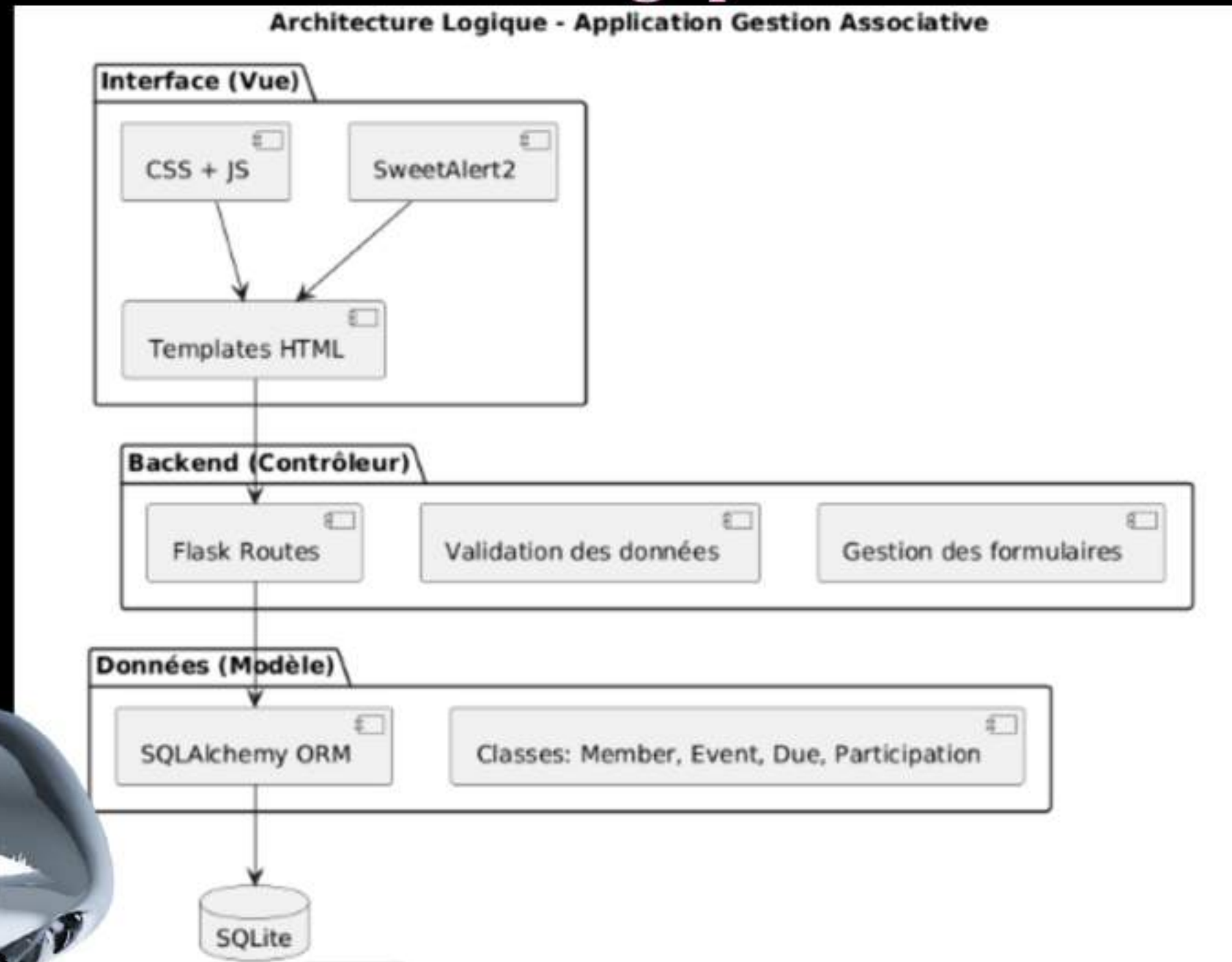
**3.1. Conception
architecturale**



**3.2. Maquettes /
Prototypes**

3.1. CONCEPTION ARCHITECTURALE

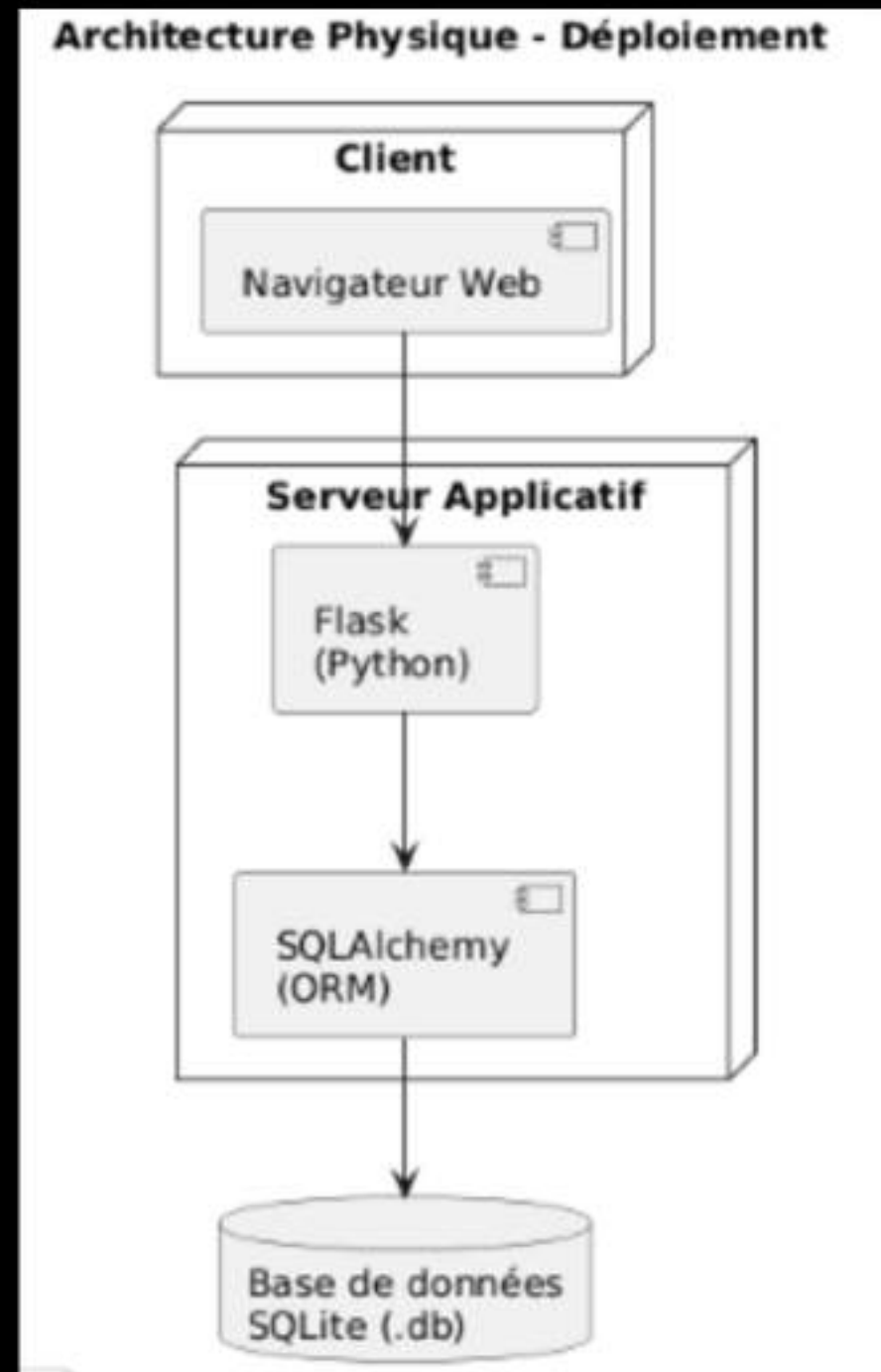
- **Architecture logique :**



- Couche Présentation (Blade)
- Couche Métier (Controllers & Services)
- Couche Données (Eloquent ORM + SQLite)

3.1. CONCEPTION ARCHITECTURALE

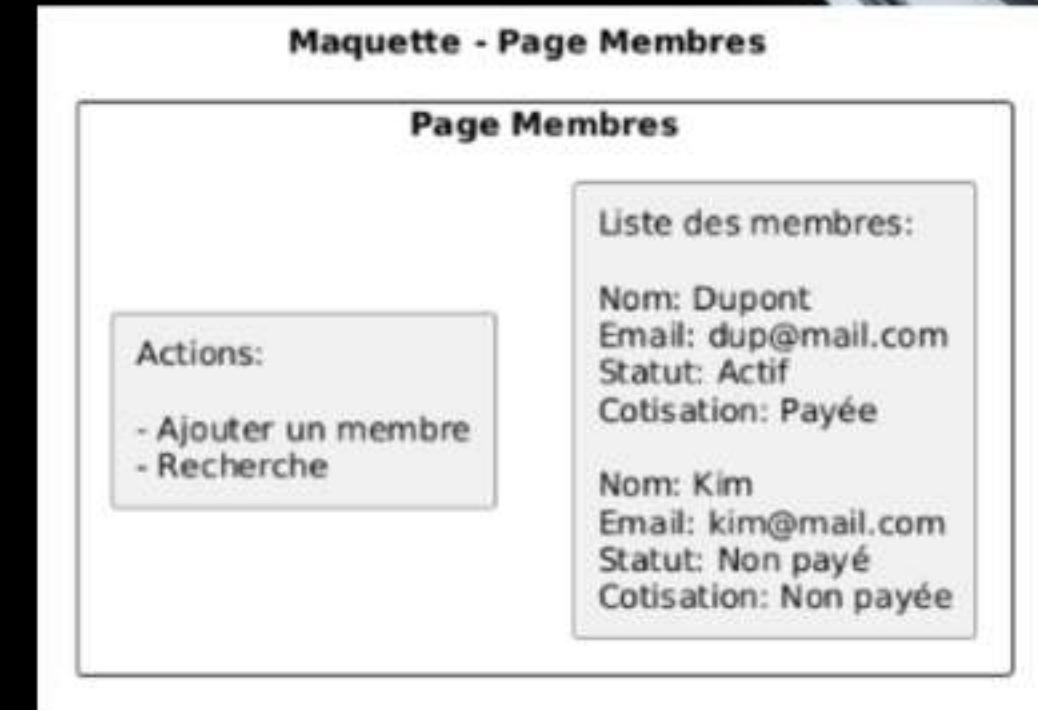
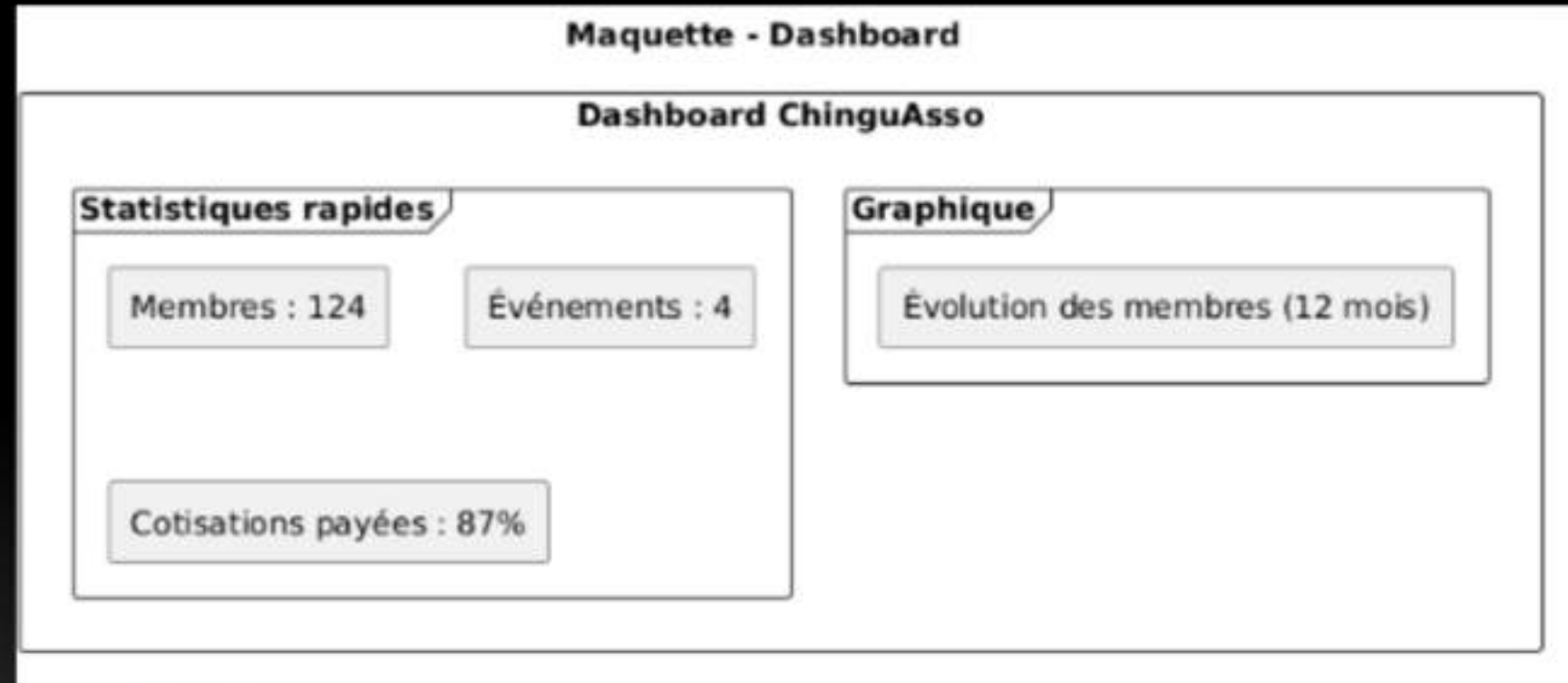
- **Architecture physique :**



- Serveur Web : Apache
- Base SQLite
- Déploiement sur serveur mutualisé / local

3.2. MAQUETTES / PROTOTYPES

- Architecture logique :



4. DÉVELOPPEMENT & TEST



**4.1. Documentation
technique**



4.2. Plan de tests



4.3. Rapport de tests

4.1. DOCUMENTATION TECHNIQUE

La documentation comprend:

- Structure du projet (app.py, models.py, templates/, static/).
- Description de chaque modèle SQLAlchemy et Configuration de la base
- Routes Flask et leurs méthodes HTTP.
- Instructions d'installation et de lancement (pip install -r requirements.txt, flask run).

4.2. PLAN DE TESTS



Types de tests prévus :

- **Unitaires** : test des fonctions CRUD pour chaque modèle.
- **Intégration** : test de cohérence entre membres, cotisations et participations.
- **Fonctionnels** : test du parcours utilisateur complet (ajouter un membre → créer un événement → inscrire à un événement).
- **Critères d'acceptation** : absence d'erreur, validation des formulaires, données cohérentes.
- **Tests de sécurité** (authentification, rôles)
- **Tests UI** (formulaires, erreurs, validations)

4.2. PLAN DE TESTS

📱 Critères d'acceptation :

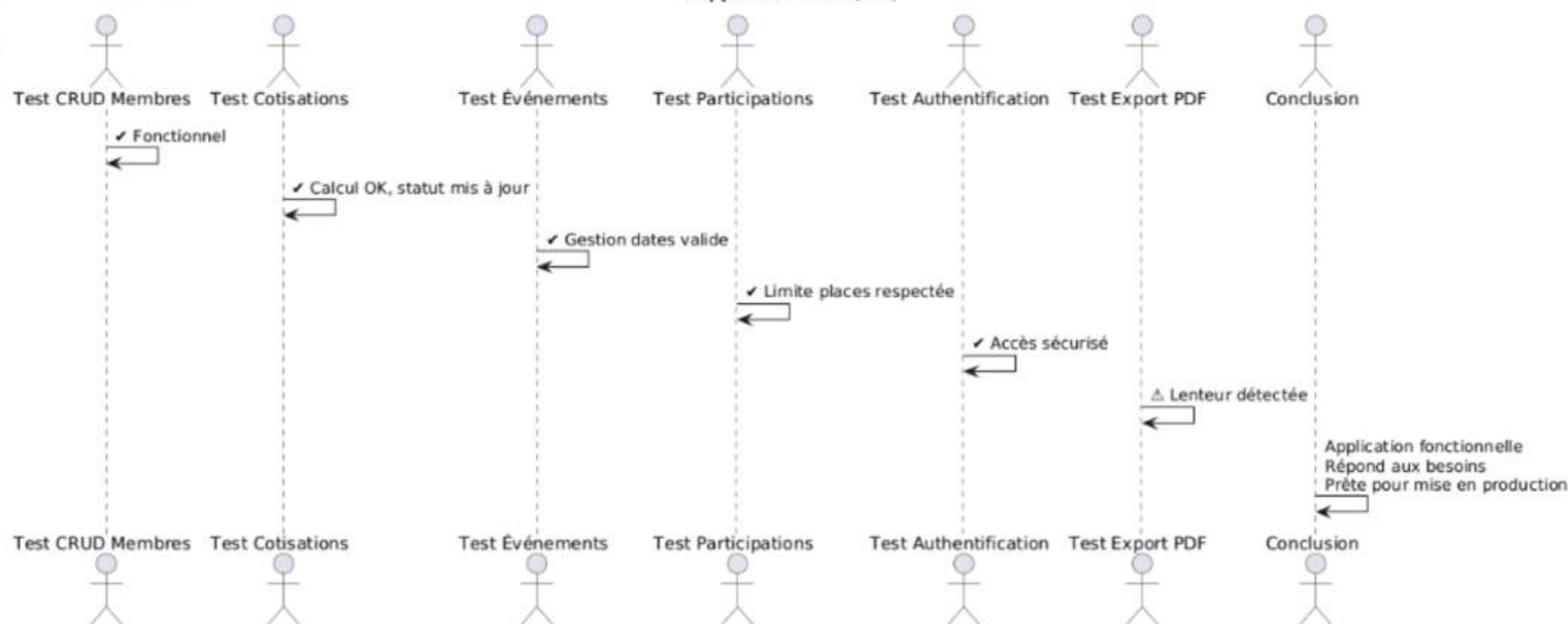
- Tous les modules CRUD fonctionnent correctement.
- Les contraintes de base (unicité email, cotisation par année) sont respectées.
- L'application démarre sans erreur sur Windows et Mac.
- Tests utilisateur concluants sur Chrome et Edge.
- Quelques anomalies mineures (saisie vide) corrigées via flash().
- Résultat final : L'application est fonctionnelle, stable et conforme au cahier des charges.

4.3 RAPPORT DE TESTS



Résultats synthétiques:

Rapport de tests (4.3)



CONCUSION

Le projet ChinguAsso montre qu'un système d'information bien conçu peut transformer la gestion d'une association.

La solution développée facilite le suivi des membres, des paiements et des événements, tout en garantissant une utilisation simple et sécurisée.

Grâce aux livrables produits (SFG, SFD, UML, architecture, tests), l'application est structurée, cohérente et prête à évoluer selon les futurs besoins de l'association.



TEAM

CHINGU_ASSO



Faizi PINEAU

Developer

Le cœur et l'âme de l'entreprise.

C'est aussi une personne qui a une idée et qui la transforme ensuite en réalité.



Yaye Awa DIOUF

Developer

Le moteur créatif de l'entreprise : celui qui imagine, conçoit et transforme les idées en solutions concrètes et innovantes.

CONTACT US

An abstract sculpture made of highly reflective, metallic material, possibly silver or chrome. It features several large, rounded, bulbous shapes that are interconnected, creating a complex, organic form. The surface is extremely shiny, reflecting light in bright, sharp highlights and deep shadows, giving it a three-dimensional, almost liquid appearance. The sculpture is set against a dark, solid background, which makes the metallic sheen stand out prominently.

E-mail

awa.diouf.yaye@gmail.com
faizi6@gmail.com

Website

www.ChinguAsso.com

Phone

+33 0000000000

Address

UNIVERSITE PARIS NANTERRE



THANK YOU

CHINGU_ASSO