

# Projet : Gestion de tontine

Membres : AGBETONYO Joël – KAYI Yao Maxime

Encadreur : BATANA Ferdinand

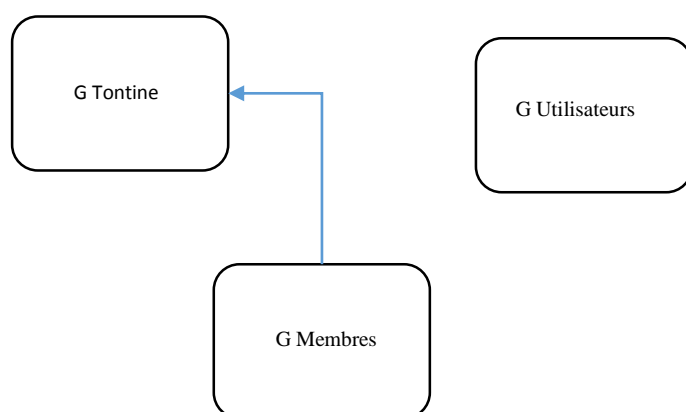
## Analyse fonctionnelle

L'objectif de cette partie est de découvrir en profondeur notre cahier de charge. Elle consistera à lister et de décrire en profondeur les grandes fonctionnalités et services offertes par l'application. Dans la suite nous aurons à voir les actions possibles d'un utilisateur, les contraintes sur chacune de ses actions à travers des cas d'utilisation qui seront modélisé avec le langage UML. Nous y définiront les grandes parties à l'aide des diagrammes en creusant loin les parties majeures. Ensuite viendra la définition de nos données qui nous conduira au diagramme de classe pour aboutir au modèle relationnel.

### 1. Décomposition de l'application

Cette section étudiera les grandes parties de l'application et fournira des descriptions textuelles de chaque partie. Sur toute la plateforme, les actions courantes sont la création d'entité, modification, mis à jour et supprimer; rien de plus que cela.

En regroupant toutes les actions définissant chaque utilisateur, on vient alors à trois grandes packages : gestion des groupes de tontine, Gestion des membres et Gestion des utilisateurs.



La description textuelle de chaque package

Gestion des groupes de tontine	<p>Elle effectue les tâches comme créer un groupe de tontine, supprimer un groupe, Mettre à jour l'état de cotisation d'un membre.</p> <p>Un administrateur peut initier une invitation à partir d'un id fourni par l'aspirant membre (plus de détail dans la partie modèle de donnée)</p> <p>Ici un administrateur de groupe crée un cycle de cotisation à chaque fois quand besoin y est Et il ne peut créer qu'un cycle à la fois. Et il pourra aussi spécifier que le cycle est terminé.</p> <p>Pour avoir une main sur les membres du groupes, le package utilise les services du package Gestion des membres</p>
Gestion des membres	Ajouter, Retirer un membre du groupe de tontine
Gestion des utilisateurs	Elle concerne les administrateurs de la plateforme

Remarquons que la gestion de compte d'un utilisateur et le système d'authentification de la plateforme constitue aussi des packages à détailler mais c'est presque la même chose qu'on retrouve sur les plateformes permettant l'authentification.

### 1. 1 Gestion des groupes de tontine

Comme dit plus haut, elle servira à manipuler les groupes de tontine. Pour un utilisateur authentifié il pourra avoir l'autorisation de d'utiliser n'importe quel service de ce package.

La contrainte est que l'utilisateur soit authentifié et ait un rôle d'utilisateur simple.

On suppose que l'utilisateur s'est authentifié et respecte la contrainte apposée.

### Diagramme de cas d'utilisation

Cas d'utilisations :

- créer un groupe (spécifier un nom ou générer un id automatique qui n'a rien à avoir avec l'id du groupe dans la table de la BDD).
- supprimer un groupe :

Contraintes : être le créateur du groupe et s'assurer que toutes les cotisations sont levées, que aucun cycle de cotisation n'est en cours.

- changer l'état d'approbation de cotisation d'un membre : checker que le membre là à cotiser

- créer un cycle :

Contraintes : s'assurer qu'il n'y a pas de cycle en cours.

- créer une invitation

Contraintes : avoir reçu une sorte de clé provenant de l'aspirant membre

NB : Au plus un cycle simultanément.

- spécifier la fin d'un cycle :

Contraintes : s'assurer que tous les membres ont cotisés sur ce cycle

#### Description des cas d'utilisation

Cas 1 Création d'un groupe (Package Gestion de Tontine)

#### 1. 2. Gestion des membres d'un groupe de tontine

Un administrateur de groupe à la possibilité de retirer ou d'ajouter un utilisateur ç son groupe de cotisation.

La contrainte est que l'utilisateur soit authentifié et ait un rôle d'utilisateur simple. Ici chaque action est sous à la contrainte qu'il n'y ait aucun cycle de cotisation en cours.

On suppose que l'utilisateur s'est authentifié et respecte les contraintes apposée.

#### Diagramme de cas d'utilisation

Cas d'utilisations :

- ajouter un nouveau membre
- retirer un membre

#### Description des cas d'utilisation

Cas 1 Gestion d'un groupe (Package Gestion des membres)
---


### 1. 3. Gestion des utilisateurs

Tous les utilisateurs sont gérés par les super administrateurs. Ils ont presque le droit sur tout ce qui compose la plateforme. Ainsi ils peuvent contrôler les utilisateurs et les groupes de tontines.

La contrainte est que l'utilisateur soit authentifié et ait un rôle de super administrateur. La suppression d'un groupe de tontine doit avoir tous ces cycles de cotisation soient terminés.

On suppose que l'utilisateur s'est authentifié et respecte les contraintes apposées.

Diagramme de cas d'utilisation

Cas d'utilisations :

- retirer un groupe de tontine
- retirer un utilisateur

Description des cas d'utilisation

Cas 1 Retirer un utilisateur (Package Gestion des membres)

Apports :

Comment un utilisateur peut-il rejoindre un groupe de tontine ?

L'entente se fait entre l'administrateur et le nouveau désirant à être membre.

L'administrateur décide d'ajouter le membre en saisissant l'id du membre qui lui a été fourni par le l'aspirant membre. Ensuite un lien est envoyé à l'aspirant qui confirme ou rejette l'invitation.

### 2. Modèle de donnée

Le but de cette section est de présenter les grands modèles de données obtenue et de les décrire un à un. Ensuite on établira le diagramme de classe qui aboutira sur le modèle relationnel.

En analysant les informations établies sur notre plateforme, on en dégage 6 grands modèles de données. On

- le modèle User : ceci contient les informations d'un simple utilisateur de la plateforme. On retrouve dans ce modèle les champs comme id, nom, prénom, téléphone, .... Dans ce modèle un autre champ d'une suite de caractère qui sera le hash d'une donnée ou plusieurs données fournies par l'utilisateur qui servira aussi d'identifiant unique. Son utilité c'est de pouvoir inviter des utilisateurs à rejoindre des groupes de tontines.

! Elle sera mise à jour à chaque fois que l'utilisateur le fournira pour recevoir une invitation d'un administrateur de groupe.

- le modèle invitation : renferme les informations sur une invitation initiée par un administrateur. On y trouve les champs comme id de l'utilisateur invité, l'id du groupe d'invitation, le statut d'acceptation.

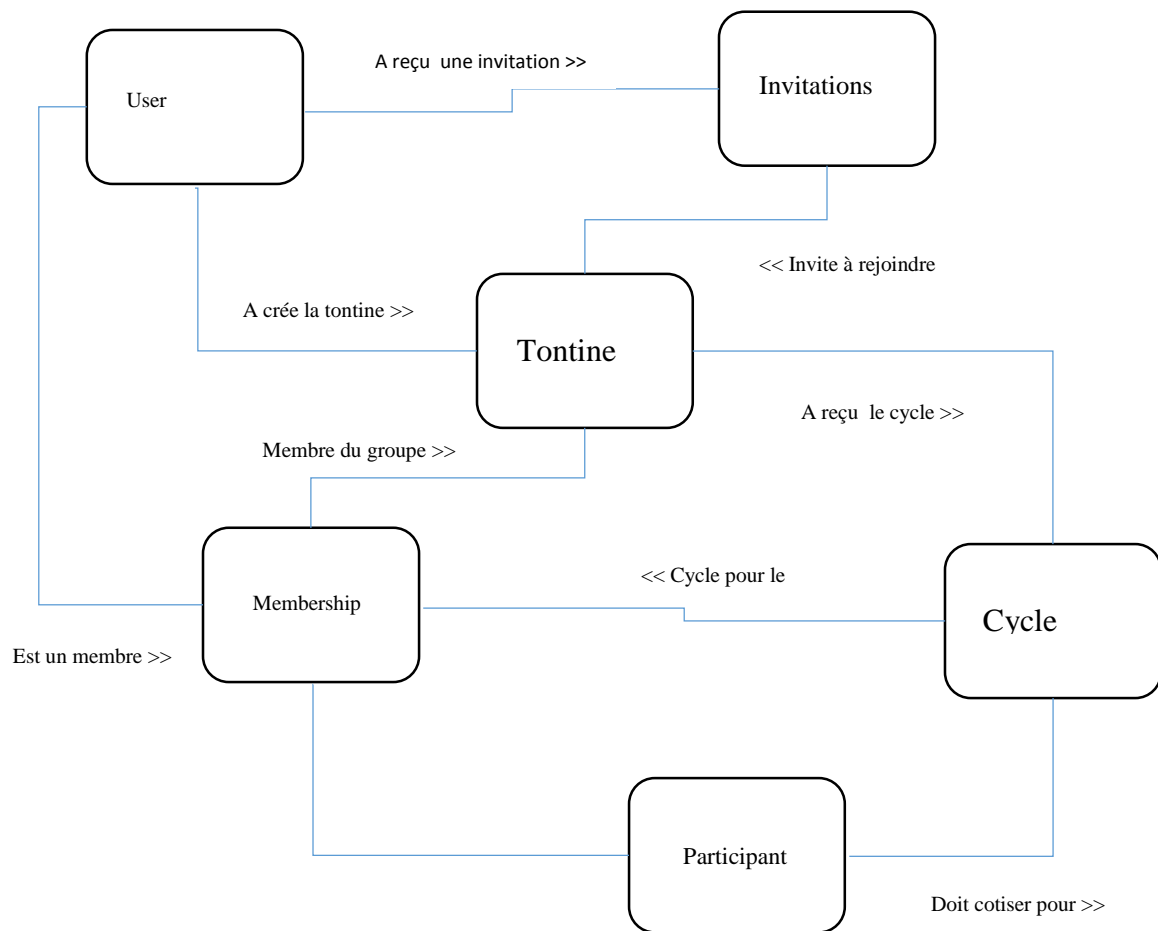
- le modèle Membership : ce modèle définit l'état d'appartenance d'un utilisateur à un groupe de tontine. On y trouve l'id de l'utilisateur et l'id du groupe. On y trouve un utilisateur que si le lien d'invitation a été accepté au préalable.

- le modèle cycle : ce modèle représente un tour de cotisation. Il regroupe l'id du groupe de tontine, l'id du membre à qui ira la cotisation totale ; le montant unitaire.

- le modèle participant : dans ce modèle, on regroupe l'id de l'utilisateur et l'id du groupe duquel il est membre.

- le modèle tontine : on définit les attributs d'un groupe de tontine ; il contient id de l'utilisateur qui l'a créé.

Voici une description des relations entre les entités de notre structure de données



## 2. 1. Les entités

Le but de cette section sera d'étudier les relations entre chaque modèle de données de notre structure. Et on terminera par une représentation UML avec le modèle relationnel.

On suppose que chaque entité a une clé primaire ou identifiant unique « id ».

Entités	Champs	Description
User	lastname	STRING
	firstname	STRING
	phone	STRING
	Invitation_hash_id	BIGINTEGER: servira pour générer la prochaine clé d'invitation.....  S'incrémante de 1 à chaque fois qu'une clé d'invitation est générée

	Contraintes : Il ne peut y avoir deux utilisateurs avec le même nom, prénom et téléphone	
	Relations : - Peut avoir plusieurs groupes de tontine - Peut avoir plusieurs invitations - Peut être membre de plusieurs groupes	
Invitation	user_invited_key	STRING : permet d'identifier l'invitation initiée
		Contraintes : UNIQUE
	user_id	BIGINTEGER : l'id de l'utilisateur que l'on a invité a rejoindre un groupe
	tontine_id	BIGINTEGER : l'id du groupe de tontine
	Contraintes :	
	Relations : - Une invitation n'a qu'un seul utilisateur invité - un seul groupe d'invitation	
Membership	user_id	BIGINTEGER : id de l'utilisateur
		Contraintes :
	tontine_id	BIGINTEGER : id du groupe de tontine
	Contraintes	
	Relations : - Plusieurs membres peuvent avoir le même groupe de tontine - Plusieurs membership peuvent appartenir à n seul utilisateur	
Tontine	name	STRING : nom du groupe ou une auto id qui pourra être changé plus tard

	admin_user_id	BIGINTEGER : id de l'utilisateur qui a créé le groupe
	Contraintes	
	Relations : - Peut avoir un ou plusieurs membres - Peut envoyer un ou plusieurs invitations - N'a qu'un seul administrateur - Peut créer plusieurs cycles de cotisation	
Cycle	checkout_member_id	BIGINTEGER : l'id du membre qui recevra le cotisation totale
	quota	INTEGER : la somme à payer par chaque participant du groupe
	Contraintes :	
	Relations : Plusieurs cycles peuvent avoir le même groupe de tontine - Un cycle peut avoir 1 ou plusieurs participants - Un cycle n'a qu'un checkout member (celui qui recevra la totalité cotisée)	
Participant	cycle_id	BIGINTEGER : cycle auquel participe le membre
	member_id	BIGINTEGER : id du membre qui participe au cycle de cotisation
	checkout	BOOLEAN : état de remplissage du quota du cycle
	Contraintes	
	Relations :	



	- Plusieurs participants peuvent cotiser sur plusieurs cycles	
--	---	--

### 2.1.2 Schématisation avec UML