

Njangui

Guide Fonctionnel (Français)

Zekeng Ndadji Milliam Maxime

2026-04-06

Résumé

Ce guide fonctionnel présente Njangui du point de vue métier. Il décrit les parcours utilisateur, les règles de gouvernance, les workflows financiers et disciplinaires, ainsi que les mécanismes de notification et de reporting nécessaires à l'exploitation quotidienne du njangui.

Table des matières

1	Introduction	2
1.1	Pourquoi Njangui existe	2
1.2	Positionnement produit	2
1.3	Principes directeurs	3
1.4	Acteurs et responsabilités	3
1.5	Vocabulaire clé	3
1.6	Référence de version	3
2	Parcours Utilisateur	3
2.1	Vue d'ensemble	3
2.2	Parcours 1 - Nouveau membre	3
2.2.1	Étape A: accès	3
2.2.2	Étape B: entrée dans le njangui	4
2.2.3	Étape C: participation courante	4
2.3	Parcours 2 - Gestionnaire membres/staff	4
2.4	Parcours 3 - Gestionnaire tontines	4
2.5	Parcours 4 - Caissier / Censeur	4
2.5.1	Caissier (MANAGE_CAISSES)	4
2.5.2	Censeur (MANAGE_PENALTIES)	4
2.6	Parcours 5 - Admin/Créateur du njangui	4
2.7	Navigation transversale	5
2.8	Expérience continue	5
3	Modules Fonctionnels	5
3.1	Carte dynamique des domaines	5
3.2	1. Overview (Accueil du njangui)	5
3.3	2. Members & Staff	5
3.3.1	Gestion des membres	5
3.3.2	Gestion du staff	6
3.4	3. Tontines	6
3.4.1	Création d'une tontine	6
3.4.2	Tirage et classement	6
3.4.3	Rounds	6
3.5	4. Caisses	6
3.6	5. Events	6

3.7	6. Penalties	6
3.8	7. Finance	7
3.9	8. Settings	7
3.10	9. Notifications (inventaire métier)	7
4	Maquettes et Scénarios d’Écran	7
4.1	1. Carte de navigation	7
4.2	2. Écran Login	7
4.3	3. Overview (cockpit)	8
4.4	4. Tontines	8
4.5	5. Dialogues critiques	8
	4.5.1 Dialogue “Résultats de tirage”	8
	4.5.2 Dialogue “Transactions de caisse”	9
4.6	6. Members & Staff	9
4.7	7. Caisses / Events / Penalties / Finance	9
4.8	8. États transversaux	9
5	Support et FAQ	9
5.1	1. Avant de diagnostiquer un bug	9
5.2	2. Questions fréquentes	9
	5.2.1 Je ne vois pas une action attendue	9
	5.2.2 Je reçois des données “anciennes” après une action	10
	5.2.3 Les notifications ne tombent pas	10
	5.2.4 Une clôture de round échoue	10
	5.2.5 Pourquoi une sanction n’est pas visible par tout le monde	10
5.3	3. Guide rapide de résolution	10
	5.3.1 Problème de permissions	10
	5.3.2 Problème de schéma Firestore	10
	5.3.3 Problème d’export PDF	10
5.4	4. Bonnes pratiques opérationnelles	10
5.5	5. Escalade et reporting bug	10
6	Index Métier Transversal	11
6.1	1. Qui fait quoi, où, et pourquoi	11
6.2	2. Actions sensibles (confirmation impérative)	11
6.3	3. Référentiel notifications (vue fonctionnelle)	11
6.4	4. Contrat d’export (important)	12
6.5	5. Lecture rapide pour support/QA	12

1 Introduction

1.1 Pourquoi Njangui existe

Njangui répond à un besoin concret: permettre à un groupe de gérer, dans une seule application, toute sa vie de njangui/tontines sans éclater les informations entre plusieurs outils.

L’application couvre: - l’organisation des tontines, - la gestion des caisses, - les événements et mains levées, - la discipline (sanctions), - la gouvernance (staff, élections, droits), - la production de rapports fiables.

1.2 Positionnement produit

Njangui est pensé comme un **client riche**: - la logique métier est implémentée dans l’application mobile, - Firebase sert principalement de stockage/synchronisation, - l’architecture prépare un remplacement futur du backend sans casser l’expérience métier.

1.3 Principes directeurs

1. **Lisibilité métier:** l'utilisateur doit comprendre clairement ce qu'il fait et ses impacts.
2. **Traçabilité:** chaque opération sensible laisse une trace exploitable dans les rapports.
3. **Gouvernance réelle:** le staff et ses droits priment sur un modèle "créateur tout-puissant".
4. **Compatibilité de schéma:** les données historiques/staging restent exploitables même après évolution.
5. **Prévention des erreurs:** confirmations sur actions lourdes, validations de formulaires, états explicites.

1.4 Acteurs et responsabilités

- **Membre:** participe aux tontines, consulte ses données, vote si éligible, suit ses sanctions/finances.
- **Staff MANAGE_MEMBERS:** gère membres, invitations, staff, certains remplacements.
- **Staff MANAGE_TONTINES:** crée/modifie les tontines et les règles de tirage.
- **Staff MANAGE_CAISSSES:** opère les caisses, clôture certains workflows financiers.
- **Staff MANAGE_EVENTS:** pilote les événements et mains levées.
- **Staff MANAGE_PENALTIES (censeurs):** valide/suit les sanctions disciplinaires.
- **Créateur/Admin du njangui:** administration structurelle (nom, admins, suppression, actions de secours selon règles de mandat).

1.5 Vocabulaire clé

- **Njangui / Tontine Group:** l'espace du groupe (membres, staff, caisses, tontines, événements).
- **Tontine:** cycle de cotisations avec tirage/classement et rounds.
- **Round (tour):** période de contribution menant à des bénéficiaires.
- **Caisse:** compte financier individuel ou collectif.
- **Main levée:** collecte dédiée (obligatoire ou non) hors cycle régulier de tontine.
- **Sanction:** pénalité disciplinaire (montant et/ou texte), éventuellement liée à une tontine/round.

1.6 Référence de version

- versionName: 1.1.0
- versionCode: 4
- code name: AJEMI-UDs

2 Parcours Utilisateur

2.1 Vue d'ensemble

Un parcours complet passe généralement par les étapes suivantes:

1. authentification,
2. sélection du njangui actif,
3. exécution d'actions selon le rôle/droits,
4. réception de notifications et suivi des états,
5. consultation/export de rapports.

2.2 Parcours 1 - Nouveau membre

2.2.1 Étape A: accès

Depuis Login, l'utilisateur peut: - se connecter, - créer un compte, - réinitialiser son mot de passe.

2.2.2 Étape B: entrée dans le njangui

- Il reçoit une invitation.
- Il accepte ou refuse.
- S'il accepte, il devient membre actif et voit les modules autorisés.

2.2.3 Étape C: participation courante

- Consulte la tontine courante, ses tirages et son positionnement.
- Suit les appels à contribution.
- Consulte ses sanctions, ses caisses et ses rapports personnels.

2.3 Parcours 2 - Gestionnaire membres/staff

Acteur: staff avec `MANAGE_MEMBERS`.

Il peut: - inviter/supprimer des membres (selon règles d'état des tontines), - enregistrer un staff initial, - créer une élection quand autorisé, - gérer des remplacements de staff (avec approbations requises).

Points de contrôle: - existence et validité du mandat, - fenêtre temporelle après expiration de mandat, - confirmations avant actions lourdes.

2.4 Parcours 3 - Gestionnaire tontines

Acteur: staff avec `MANAGE_TONTINES`.

Cycle type:

1. créer une tontine,
2. configurer les préconditions de tirage,
3. piloter/fermer le tirage,
4. suivre les rounds jusqu'à clôture complète.

Règles importantes: - la date de début est contrainte par la durée de round, - le tirage peut être participant ou algorithmique, - les règles modifiées peuvent invalider les tirages précédents.

2.5 Parcours 4 - Caissier / Censeur

2.5.1 Caissier (`MANAGE_CAISSSES`)

- gère caisses individuelles/collectives,
- enregistre opérations comptables (crédit/débit),
- gère contributions et outflows des tours/événements,
- produit rapports financiers filtrés.

2.5.2 Censeur (`MANAGE_PENALTIES`)

- complète/valide les sanctions,
- suit le paiement,
- marque les sanctions payées,
- coordonne la clôture de recouvrement,
- produit rapports disciplinaires filtrés.

2.6 Parcours 5 - Admin/Créateur du njangui

- renomme le njangui (ID inchangé),
- gère les admins,

- peut supprimer le njangui,
- assure la continuité gouvernance quand le staff est absent/expiré selon règles.

2.7 Navigation transversale

Le menu donne accès à: - Overview - Tontines - Members & Staff - Caisses - Events - Finance - Penalties - Settings

Le changement de thème/langue est global et doit être instantané sans casser la navigation.

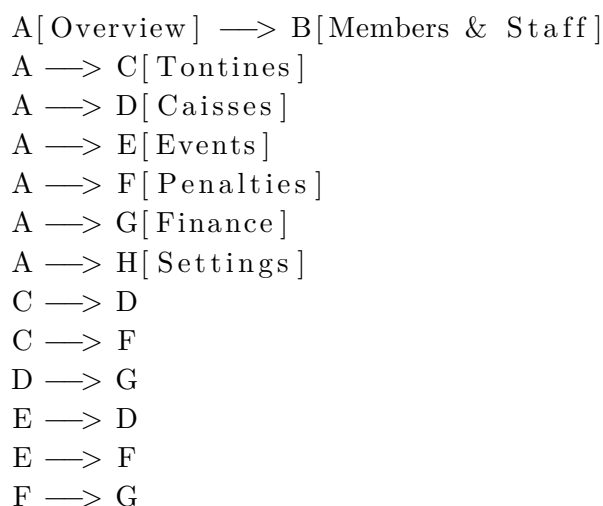
2.8 Expérience continue

L'application privilégiée: - loaders cohérents pendant les opérations, - refresh après chaque action critique, - pull-to-refresh pour recharger manuellement une page, - notifications locales même en arrière-plan (worker périodique).

3 Modules Fonctionnels

3.1 Carte dynamique des domaines

flowchart LR



3.2 1. Overview (Accueil du njangui)

Rôle: cockpit de pilotage.

Contient: - résumé membres/invitations/staff, - actions de gouvernance (staff, élections, votes), - actions d'administration du njangui (nom, devise, admins, suppression, quitter), - informations de rounds en cours (tour courant et bénéficiaires).

3.3 2. Members & Staff

3.3.1 Gestion des membres

- invitations,
- suivi membres actifs/inactifs,
- suppression d'un membre (avec impact différent selon l'état des tontines).

Règle de suppression: - tontines non commencées: membre retiré (y compris du tirage), - tontines en cours/passées: données historiques conservées.

3.3.2 Gestion du staff

- enregistrement d'un staff existant,
- création d'élection (postes, candidats, votants, mandat),
- remplacement de staff avec workflow d'approbation.

Règles d'accès: - basées sur le mandat actif/expiré, - fallback admin/creator après délai métier prévu.

3.4 3. Tontines

3.4.1 Création d'une tontine

Champs essentiels: - individualAmountPerRound, - startDate, - roundDuration (jours/semaines/mois), - maxWinnerPerRound, - participants (avec nombre de noms par participant).

3.4.2 Tirage et classement

Préconditions configurables: - multipleNamesStrategy (NONE / BALANCED), - positionning (positions fixes), - rankingType (RANDOM_ALGORITHM / PARTICIPANT_SELECTION), - deadlineForParticipantSelection.

Règles métier clés: - regroupement des bénéficiaires par maxWinnerPerRound, - clôture possible manuelle ou algorithmique pour manquants, - modification des règles => annulation des tirages existants + relance.

3.4.3 Rounds

- annonce début de round à tous,
- rappels de cotisation dans les derniers jours,
- clôture par les responsables de caisse,
- archivage en SuccessfulRound,
- fin automatique de la tontine quand tous ont bénéficié.

3.5 4. Caisses

Types: - **individuelles** (solde par membre), - **collectives** (solde unique partagé).

Fonctionnalités: - CRUD de caisses, - opérations comptables (crédit/débit), - désactivation temporaire d'alertes de déficit, - distribution d'une caisse collective entre membres/externes, - consultation/filtrage/export des transactions.

Règle devise: - la devise du njangui s'applique à tous les montants affichés.

3.6 5. Events

Deux familles: - **Événements**, - **Mains levées**.

Cas gérés: - contributions individuelles, - contributions collectives, - bénéficiaires internes et externes, - outflow avec priorités de caisses, - rollback d'outflow avant validation finale.

Effets automatiques: - débits sur caisses, - sanctions générées si déficit ou retard selon règles.

3.7 6. Penalties

Cycle de vie d'une sanction:

1. génération automatique ou création manuelle,
2. validation (automatique pour certaines créations manuelles selon règle),
3. notification des concernés,

4. paiement marqué,
5. recouvrement clôturé avec destination des montants.

Une sanction peut contenir: - des pénalités de type **montant**, - des pénalités de type **texte** (ex: suspension), - un contexte (tontine/round/libre), - une date limite.

3.8 7. Finance

Permet: - filtrage par période/contexte/membre, - agrégats lisibles (contributions, caisses, sanctions, événements, mains levées), - export PDF basé uniquement sur le filtre courant.

3.9 8. Settings

- profil utilisateur (noms/pseudo/photo),
- préférences (langue, thème, groupe par défaut),
- mot de passe,
- suppression logique de compte.

3.10 9. Notifications (inventaire métier)

Les notifications couvrent notamment: - invitations/retours d'invitation, - création tontine, - tirage (token, classement, deadline, clôture, changement de rang), - rounds (début, rappel, clôture, fin tontine), - gouvernance (élection, rappel vote, résultats, staff manquant, remplacements), - sanctions (revue, validation, échéance, recouvrement), - caisses (déficit), - événements/mains levées (outflow requis/appliqué, contributions, clôture).

Elles sont émises dans Firestore puis délivrées localement via worker périodique.

4 Maquettes et Scénarios d'Écran

Cette section propose des maquettes textuelles “vivantes”: au-delà de la structure, elles décrivent les intentions UX, les états et les actions attendues.

4.1 1. Carte de navigation

Auth (Login/Register/Reset)

- > Overview (Njangui actif)
- > Members & Staff
- > Tontines
- > Caisses
- > Events
- > Penalties
- > Finance
- > Settings

4.2 2. Écran Login

NJANGUI	
Sous-titre: Gérez votre communauté en confiance	
Email	<input type="text"/>
Mot de passe	<input type="password"/>

[Se connecter]	
Créer un compte	Mot de passe oublié

États clés: - loader pendant authentification, - message clair en cas de compte inexistant/désactivé, - navigation propre après succès.

4.3 3. Overview (cockpit)

Groupe actif [Select]	[Actualiser]
Cartes KPI: Membres actifs	Invitations Staff
Gouvernance: [Register Staff]	[Create Election]
	[Vote] [Approve Replacement]
Vie du groupe: [Rename]	[Currency] [Admins] [Leave]
Tontines en cours: round actuel	+ bénéficiaires

4.4 4. Tontines

Tontine courante [Select nom tontine]
Carte info: statut, round courant, bénéficiaires
Actions: [Gérer tirage] [Configurer règles] [...]
Liste de mes tontines (cards)
- Nom Statut Actions selon droits

Détails UX: - la tontine courante doit être identifiable sans ambiguïté, - pas de doublon d'actions entre header et cartes, - séparation visuelle nette entre blocs.

4.5 5. Dialogues critiques

4.5.1 Dialogue "Résultats de tirage"

Résultats du tirage	
... classement ...	
[Close]	[Clore le tirage]

Règles de style: - bouton secondaire (close) à gauche, - bouton dangereux/impactant en style accentué à droite, - confirmations avant actions irréversibles.

4.5.2 Dialogue “Transactions de caisse”

Filtres: dates, compte, type, recherche	
[Appliquer]	[Réinitialiser]
Liste transactions filtrées	
[Fermer]	[Exporter transactions]

Le bouton export doit rester visible, même sur petits écrans.

4.6 6. Members & Staff

Invitations membres	
[Inviter]	[Relancer]
Staff actif / Mandat	
[Enregistrer staff]	[Créer élection]
Postes (édition inline):	
- Nom du poste	
- Droits (chips)	
- Candidats / titulaires	

4.7 7. Caisses / Events / Penalties / Finance

Principe commun: - filtres en haut, - carte de synthèse agrégée, - liste détaillée, - actions contextualisées par droit, - export unique basé sur le filtre courant.

4.8 8. États transversaux

À prévoir sur tous les écrans: - loader d'initialisation, - loader d'action, - état vide guidé (quoi faire ensuite), - état erreur récupérable, - pull-to-refresh.

5 Support et FAQ

5.1 1. Avant de diagnostiquer un bug

Toujours vérifier: - le njangu actif sélectionné, - le rôle effectif de l'utilisateur, - l'état du mandat staff (actif/expiré), - la connectivité réseau, - la fraîcheur des données (pull-to-refresh).

5.2 2. Questions fréquentes

5.2.1 Je ne vois pas une action attendue

Cause fréquente: manque de droit ou règle temporelle non satisfaite.

Checklist: 1. l'utilisateur est-il membre actif? 2. possède-t-il le droit `MANAGE_*` attendu? 3. le mandat staff permet-il cette action maintenant? 4. la tontine/événement est-il dans le bon état?

5.2.2 Je reçois des données “anciennes” après une action

Actions recommandées: - glisser vers le bas pour rafraîchir, - sortir/revenir sur l'écran, - vérifier les loaders (si absents, logger le cas), - vérifier la persistance effective dans Firestore.

5.2.3 Les notifications ne tombent pas

Vérifier: - permission notification accordée, - utilisateur authentifié, - worker périodique planifié, - document notifications /<groupId> avec champs valides (type, recipients, contentData, etc.).

5.2.4 Une clôture de round échoue

Causes possibles: - tirage incomplet ou non clos, - droits insuffisants (MANAGE_CAISSES), - données contributions/outflows incomplètes, - round déjà clôturé.

5.2.5 Pourquoi une sanction n'est pas visible par tout le monde

Une sanction suit un cycle: - création, - validation, - notification, - paiement, - clôture de recouvrement. Selon son état, la visibilité et les actions changent.

5.3 3. Guide rapide de résolution

5.3.1 Problème de permissions

1. Ouvrir la fiche membre/staff.
2. Confirmer droits sur le poste.
3. Contrôler date de début/fin de mandat.
4. Reproduire avec un compte de test au même rôle.

5.3.2 Problème de schéma Firestore

1. Vérifier présence des champs attendus.
2. Vérifier fallback quand champ absent.
3. Tester migration douce sur données partielles.
4. Ajouter logs ciblés avant écriture/lecture.

5.3.3 Problème d'export PDF

1. Vérifier filtres actifs.
2. Vérifier présence de données filtrées.
3. Vérifier permissions stockage/partage.
4. Vérifier ouverture automatique post export.

5.4 4. Bonnes pratiques opérationnelles

- Toujours confirmer les actions lourdes (clôture, suppression, annulation de tirage).
- Garder les règles de tirage stables une fois les tirages lancés.
- Utiliser les rapports filtrés en réunion pour aligner le groupe.
- Éviter les opérations financières “hors workflow” non tracées.

5.5 5. Escalade et reporting bug

Quand un bug est remonté, capturer: - module + écran, - rôle de l'utilisateur, - action exacte déclenchée, - résultat attendu vs observé, - logs (stacktrace + horodatage), - identifiant du njangu/tontine concerné.

Ce format réduit fortement le temps de diagnostic.

6 Index Métier Transversal

Objectif: relier rapidement chaque règle métier à son usage concret, sans entrer dans le détail technique d'implémentation.

6.1 1. Qui fait quoi, où, et pourquoi

Domaine	Action clé	Qui peut le faire	Impact métier
Membres	Inviter ou retirer un membre	Responsable membres	Met à jour la composition réelle du njangui
Gouvernance	Créer une élection / enregistrer un staff	Selon état du mandat et rôle autorisé	Met en place les droits de pilotage
Tontines	Configurer le tirage et clore les étapes	Responsable tontines	Définit l'ordre des bénéficiaires
Rounds	Clore un tour de tontine	Responsable caisses	Archive le tour et déclenche sanctions si besoin
Caisses	Enregistrer crédit/débit	Responsable caisses	Assure la traçabilité financière
Événements	Clore contributions/outflows	Responsable caisses + événements	Impact direct sur soldes et discipline
Sanctions	Valider / marquer payé / clore recouvrement	Censeurs + caissiers	Met à jour l'état disciplinaire et financier
Rapports	Exporter un rapport	Utilisateur autorisé du module	Produit un document d'audit filtré

6.2 2. Actions sensibles (confirmation impérative)

Action	Pourquoi c'est sensible
Modifier les règles de tirage	Peut invalider des tirages déjà effectués
Clore un tirage incomplet	Lance un tirage automatique pour les manquants
Clore un round	Génère des archives et potentiellement des sanctions
Supprimer une caisse	Supprime aussi les écritures liées
Supprimer une tontine / le njangui	Impact structurel important pour le groupe
Clore un recouvrement de sanction	Valide définitivement l'affectation financière

6.3 3. Référentiel notifications (vue fonctionnelle)

Famille	Exemples	Public cible
Invitations	invitation, acceptation, refus	membres et responsables
Tontines	création, tirage, changement de position	participants concernés
Rounds	début, rappel, clôture	participants + responsables caisses
Gouvernance	élection, rappel de vote, résultats	membres et staff
Sanctions	validation, échéance, recouvrement	membre concerné + censeurs/caissiers

Famille	Exemples	Public cible
Caisses	alerte déficit	membres en défaut
Événements / mains levées	outflow requis, clôture	caissiers + membres concernés

6.4 4. Contrat d'export (important)

Principe unique dans l'application: - un seul bouton d'export par zone, - export basé strictement sur les données actuellement filtrées.

Module	Filtres usuels
Transactions de caisses	dates, compte, type, texte
Finance	période, membre, contexte
Sanctions	type, période, contexte, membre

6.5 5. Lecture rapide pour support/QA

1. Vérifier le njanguï actif.
2. Vérifier le droit réel de l'utilisateur.
3. Vérifier l'état de cycle (mandat, tirage, round, sanction).
4. Forcer un rafraîchissement avant conclusion.
5. Reproduire si possible sur un cas proche de la prod/staging.