

DOCUMENTATION TECHNIQUE DE RÉFÉRENCE

PROJET : JOUAN-SUGU

Version : 1.0 **Date :** 21 Janvier 2026 **Cible :** Équipe de Développement, Chefs de Projet, Parties Prenantes

I. VISION ET PÉRIMÈTRE

1.1 Description du Projet

Jouan-Sugu est une "Super-Application" hybride combinant E-commerce, Investissement boursier et Services bancaires. Elle vise à démocratiser l'accès au capital pour les entreprises locales et à offrir de la liquidité immédiate aux particuliers via le nantissement d'actifs numériques.

1.2 Objectif du Prototype (Livraison : Août)

L'objectif est de produire un **MVP (Minimum Viable Product)** fonctionnel en circuit fermé.

- **Interne** : Toutes les logiques (Achats, Bourse, Prêts, Nantissement) sont réelles et codées.
- **Externe** : Les partenaires financiers (Banques, Mobile Money) sont **simulés** via des interfaces factices (Mock Servers) pour prouver la viabilité du concept sans dépendance contractuelle immédiate.

II. ARCHITECTURE TECHNIQUE

2.1 Stack Technologique

- **Backend (API & Logique Métier)** : Laravel 11 (PHP). Choisi pour sa robustesse, sa sécurité native et sa facilité à gérer des tâches complexes (Jobs, Queues).
- **Frontend Mobile** : Flutter (Dart). Pour une application native unique (Android/iOS) fluide et performante.
- **Frontend Web (Admin & Dashboard)** : Laravel Blade (ou React.js/Vue.js consommant l'API) pour les interfaces d'administration.
- **Base de Données** : MySQL 8.0 (ou MariaDB).
- **Cache & Queues** : Redis. Essentiel pour gérer les transactions boursières en temps réel.

2.2 Architecture Système

Le système repose sur une architecture **API RESTful** :

1. **Client Mobile (Flutter)** : Envoie des requêtes HTTPS sécurisées.
2. **API Gateway (Laravel)** : Authentifie (via Sanctum), valide les données et distribue les tâches.

3. Modules de Service :

- *Marketplace Service* : Gestion des produits/commandes.
- *Finance Engine* : Gestion des actions, calculs de risques, nantissement.
- *Banking Simulator* : Module simulant les réponses des banques et opérateurs Mobile Money.

2.3 Sécurité

- **Authentification** : Laravel Sanctum (Token based).
 - **Rôles (RBAC)** : Utilisation de `spatie/laravel-permission` pour gérer les rôles : Admin, Particulier, Entreprise, Banquier (Simulé).
 - **Transactions** : Toutes les opérations financières doivent être atomiques (Database Transactions) pour éviter les incohérences.
-

III. MODULES FONCTIONNELS (Épiques)

Module 1 : Gestion des Utilisateurs (KYC & Auth)

- **Rôle** : Enrôler les utilisateurs et vérifier leur identité.
- **Pourquoi** : Indispensable pour la partie bancaire (lutte contre la fraude).
- **Fonctionnalités** : Incription, Login, Upload de pièces d'identité, Validation compte Entreprise vs Particulier.

Module 2 : La Marketplace "Sugu" (Commerce)

- **Rôle** : Lieu d'échange de biens et services.
- **Pourquoi** : Génère du flux et permet aux entreprises de réaliser le chiffre d'affaires qui valorisera leurs actions.
- **Fonctionnalités** :
 - **C2C** : Petites annonces particuliers.
 - **B2B/B2C** : Boutiques officielles entreprises.
 - **Appels d'Offres** : Une entreprise publie un besoin, les autres postulent.

Module 3 : Finance & Bourse (Investissement)

- **Rôle** : Permettre aux entreprises de lever des fonds et aux utilisateurs d'investir.
- **Pourquoi** : C'est le cœur innovant du projet.

- **Fonctionnalités :**

- Émission d'actions (Entreprise).
- Achat/Vente d'actions (Marché secondaire).
- Carnet d'ordres simple (Offre vs Demande).
- Calcul dynamique de la valeur de l'action (Basé sur l'activité Marketplace).

Module 4 : Banque & Nantissement (Crédit)

- **Rôle :** Transformer les actions en liquidités (Cash).

- **Pourquoi :** Apporte de la valeur immédiate à l'investissement.

- **Fonctionnalités :**

- **Simulateur de Scoring :** Calcule combien on peut prêter (LTV - ex: 60% de la valeur des actions).
- **Nantissement :** "Geler" les actions de l'utilisateur tant que le prêt n'est pas remboursé.
- **Liquidation :** Vente automatique des actions si l'utilisateur ne rembourse pas.

Module 5 : E-Wallet & Transactions

- **Rôle :** Le grand livre comptable interne.
 - **Pourquoi :** Centraliser l'argent avant qu'il ne sorte vers le Mobile Money.
 - **Fonctionnalités :** Rechargement, Retrait (vers Mobile Money simulé), Transfert P2P, Paiement Marketplace.
-

IV. MODÉLISATION DES DONNÉES (MCD)

Voici les tables critiques de la base de données. Pour chaque table, l'explication du choix technique est fournie.

A. Tables Transversales

Table	Description	Pourquoi cette table ? (Justification)
users	ID, Nom, Email, Role, KYC_Status	Table centrale pour l'authentification et lier toutes les actions à une personne ou entité.
wallets	ID, User_ID, Balance, Currency	Séparation des responsabilités. On ne met pas l'argent dans la table users pour des raisons de sécurité et pour permettre d'avoir plusieurs devises plus tard.

B. Tables Marketplace

Table	Description	Pourquoi cette table ? (Justification)
products	ID, Seller_ID, Title,	Stocke le catalogue. Le champ Type permet de

Table	Description	Pourquoi cette table ? (Justification)
	Price, Type (Bien/Service)	distinguer un objet physique d'un service.
orders	ID, Buyer_ID, Total_Amount, Status	Gère le cycle de vie d'un achat. Essentiel pour calculer le chiffre d'affaires des entreprises (et donc la valeur de leur action).

C. Tables Finance & Bourse (Complexe)

Table	Description	Pourquoi cette table ? (Justification)
companies	ID, User_ID (Owner), Valuation, Share_Price	Extension du profil "Entreprise". Contient la valeur officielle de l'action à l'instant T.
company_shares	ID, Company_ID, Total_Shares_Issued	Définit combien d'actions existent au total pour une entreprise (le "gâteau" total).
portfolios	ID, User_ID, Company_ID, Quantity_Owned	Cruciale. C'est le registre de propriété. Elle dit "Moussa possède 10 actions de Bama-Tech". C'est cette table qu'on interroge pour savoir si quelqu'un peut vendre.
market_orders	ID, User_ID, Type (Buy/Sell), Price_Limit, Status	Le carnet d'ordres. Stocke les intentions d'achat ou de vente avant qu'elles ne soient exécutées.

D. Tables Banque & Risque

Table	Description	Pourquoi cette table ? (Justification)
loans	ID, User_ID, Amount, Interest_Rate, Due_Date, Status	Le dossier de crédit classique. Suit si le prêt est "En cours", "Remboursé" ou "En défaut".
collaterals	ID, Loan_ID, Portfolio_ID, Locked_Quantity	La table la plus importante du système de risque. Elle lie une dette (Loan) à des actions (Portfolio). Tant que cette ligne existe, le système empêche l'utilisateur de vendre les actions ciblées dans portfolios.
transactions	ID, Sender_Wallet,	Audit Trail. Historique immuable de tous les

Table	Description	Pourquoi cette table ? (Justification)
	Receiver_Wallet, Type, Amount	mouvements d'argent. Indispensable pour la comptabilité et le débogage.

V. ORGANISATION DU PROJET (MÉTHODE AGILE & CLICKUP)

5.1 Structure ClickUp

Nous utiliserons la hiérarchie suivante dans ClickUp pour organiser le travail :

- **Space** : Développement Jouan-Sugu
- **Folder** : Sprint (ex: Février - Backend Core)
- **List** : Backlog, To Do, In Progress, Review, QA (Test), Done.

5.2 Les Épiques (Grands blocs de tâches)

1. **EPC-01 : Infrastructure & Base** : Setup Laravel/Flutter, Auth, Design System.
2. **EPC-02 : Marketplace Core** : CRUD Produits, Panier, Commandes.
3. **EPC-03 : Trading Engine** : Logique d'achat/vente d'actions.
4. **EPC-04 : Banking Module** : Logique de prêt, Nantissement, Mock Mobile Money.
5. **EPC-05 : Frontend Integration** : Liaison des écrans Flutter avec l'API.

5.3 Exemple de User Story (Modèle à suivre)

Chaque tâche dans ClickUp doit suivre ce format :

Titre : Demander un prêt sur nantissement **En tant que** : Investisseur (Particulier) **Je veux** : Bloquer mes actions "Bama-Tech" pour obtenir un prêt immédiat **Afin de** : Avoir du cash sur mon Mobile Money sans vendre mes actions

Critères d'acceptation (Definition of Done) :

1. Le système vérifie que j'ai bien les actions disponibles.
2. Le système calcule le montant max (60% de la valeur).
3. Une entrée est créée dans la table **collaterals**.
4. Mon wallet est crédité.
5. Mes actions sont marquées comme "Locked" (Invendables).

5.4 Calendrier Prévisionnel (Roadmap)

- **Phase 1 (Fév-Mars)** : Backend Laravel, Base de données, API Marketplace.

- **Phase 2 (Avril-Mai)** : Moteur financier (Bourse + Prêts), Mock Servers.
 - **Phase 3 (Juin-Juil)** : Développement Mobile Flutter, Intégration API.
 - **Phase 4 (Août)** : Tests de charge, Débogage, Livraison Prototype.
-

Redigé par : Traore Don Manuel Yiritié Mawuli Jouanelle.