

Projet : nafankap

Sommaire :

1. Introduction.....	4
1.1 Objet du document.....	4
1.2 Portée du système.....	4
1.3 Public cible du document.....	4
1.4 Définitions et termes.....	5
2. Description générale.....	5
2.1 Vision produit.....	5
2.2 Types d'utilisateurs.....	6
2.3 Environnement de fonctionnement.....	7
2.4 Contraintes.....	7
3. Exigences fonctionnelles.....	7
3.1 Authentification & gestion des utilisateurs (F-AUTH).....	7
3.2 Gestion des tenants & utilisateurs internes (F-TENANT).....	8
3.3 Clients (CRM unifié) (F-CRM).....	8
3.4 Inbox omnicanale (F-INBOX).....	9
3.5 Produits, fournisseurs & lots d'achat (F-PROD / F-SUPPLIER / F-STOCK).....	10
3.6 Commandes & livraisons (F-ORD / F-DEL).....	11
3.7 Facturation (F-BILL).....	12
3.8 Automatisations (F-AUTO).....	13
3.9 Abonnements & paiements (F-SUB).....	13
3.10 Analytics & marges (F-ANALYT).....	14
4. Exigences non fonctionnelles (résumé).....	14
1. Architecture technique globale.....	15
2. Organisation des modules / dossiers (idée).....	16
3. Modèle de données (schéma logique).....	16
3.1 Tenants & utilisateurs.....	16
3.2 Clients & contacts.....	17
3.3 Conversations & messages.....	17
3.4 Fournisseurs, produits & lots d'achat.....	18
3.5 Commandes, livraisons & factures.....	18
3.6 Abonnements & paiements.....	19
4. Intégrations principales.....	19
4.1 WhatsApp Cloud API & Meta Graph API.....	19
4.2 Lemon Squeezy.....	20
4.3 Flutterwave (Mobile Money).....	20
5. Sécurité & multi-tenant.....	20
6. Principaux flux techniques.....	20
6.1 Flux : un client écrit sur WhatsApp.....	20
6.2 Flux : l'opérateur crée une commande et l'envoie au livreur.....	21
6.3 Flux : génération et envoi de facture.....	21
6.4 Flux : paiement d'abonnement via Lemon Squeezy.....	21



SRS – Spécification des Exigences Logiciel

1. Introduction

1.1 Objet du document

Ce document décrit la **Spécification des Exigences Logiciel (SRS)** de la plateforme **nafankap**, un SaaS destiné aux commerçants qui vendent principalement via **WhatsApp, Facebook et Instagram** (écosystème Meta).

Il précise :

- les **besoins métiers** couverts par nafankap,
- les **fonctionnalités** attendues,
- les **contraintes non fonctionnelles** (performance, sécurité, etc.),
- les grandes lignes du **modèle de données** et du **tunnel de vente** ciblé (prospect FB/IG → vente WhatsApp).

Ce document sert de base :

- à la **conception technique** (SDS),
- au **développement**,
- aux **tests** (recette),
- aux **évolutions futures** du produit.

1.2 Portée du système

nafankap est une application web **multi-tenant** qui permet à une boutique de :

- **Centraliser** l'ensemble des échanges avec un même client au sein d'un **Dossier Client** unique,
 - tout en **séparant les historiques** par canal (WhatsApp, Facebook, Instagram) : pas de timeline mélangée.
- **Suivre toutes les conversations** client via une **Vue Dossier** avec onglets (WhatsApp / Facebook / Instagram).
- **Créer et gérer des commandes** rattachées au Dossier Client.
- **Gérer un catalogue de produits** lié à des **fournisseurs** et à des **lots d'achat** (stock entrant avec prix d'achat).
- **Suivre les stocks** et calculer la **marge réelle** par commande / produit / fournisseur, grâce à un mécanisme d'**allocations de stock** (déstockage multi-lots invisible).
- **Générer et envoyer des factures** (PDF) aux clients.
- **Gérer les livraisons** (coordonnées pays / ville / quartier, livreur, bordereau).
- **Suivre les performances** (ventes, marges, géographie, fournisseurs).
- Fonctionner en **mode SaaS** avec des **plans d'abonnement payants**, réglés via **Lemon Squeezy (carte)** et **Flutterwave (Mobile Money)**.

1.3 Public cible du document

- Product owner / fondateur de nafankap
- Équipe de développement (frontend, backend, DevOps, intégrations)
- UX/UI designers
- Équipe de test et de recette
- Équipe support et exploitation

1.4 Définitions et termes

nafankap : nom de la plateforme SaaS.

Tenant : boutique / commerçant inscrit sur nafankap.

Utilisateur interne : personne qui se connecte à nafankap pour gérer une boutique (admin, opérateur, etc.).

Super admin : administrateur global de la plateforme nafankap (côté éditeur).

Client (Customer) : entité unique représentant une personne physique. Dans nafankap, c'est un **Dossier Client**.

Dossier Client / Vue Dossier Client :

- Fiche centrale qui regroupe toutes les **identités** d'une même personne (numéro WhatsApp, PSID Facebook, handle Instagram, e-mail, etc.).
- Interface qui permet de basculer entre **les onglets de conversation par canal** (WhatsApp, Facebook, Instagram), **sans jamais mélanger les messages** dans une seule timeline.

Conversation Canal : suite chronologique de messages sur un canal spécifique (ex. une conversation WhatsApp pour ce client).

Canal : WhatsApp, Facebook Messenger, Instagram Direct.

Fournisseur : entité auprès de laquelle la boutique achète ses produits.

Lot d'achat : stock entrant d'un produit donné, acheté auprès d'un fournisseur, avec un **prix d'achat unitaire** et une **quantité initiale / restante**.

Allocation de stock : lien technique entre une ligne de commande client et un ou plusieurs lots d'achat, utilisé pour calculer la **marge réelle** (déstockage multi-lots transparent pour l'utilisateur).

Livraison : processus d'acheminement de la commande chez le client, éventuellement via un partenaire de livraison.

OTP (One-Time Password) : code à usage unique (envoyé par SMS ou WhatsApp) pour authentifier un utilisateur via son numéro de téléphone.

Abonnement : contrat de service entre un tenant et nafankap, avec un plan, une période et un prix.

2. Description générale

2.1 Vision produit

Le flux typique (notamment au Cameroun) est le suivant :

1. Un prospect voit une publicité sur **Facebook / Instagram**.
2. Il contacte la boutique via **Messenger ou Instagram DM**.
3. Après échange, il fournit son **numéro de téléphone** (WhatsApp).
4. La vente se conclut sur **WhatsApp** (paiement, localisation, livraison).

nafankap doit **faciliter ce passage** :

- Capturer le prospect sur Facebook / Instagram.
- Lier son **numéro de téléphone** dès qu'il le donne.
- Retrouver ce même client sur WhatsApp pour finaliser la commande, tout en gardant l'historique Messenger/Instagram accessible dans le **Dossier Client**.
- Permettre à l'opérateur de **passer de l'onglet Facebook à l'onglet WhatsApp** dans la même vue, sans perte de contexte.

Objectif : faire de nafankap **l'outil central** des vendeurs WhatsApp / Facebook / Instagram pour :

- avoir une **vue 360° par client**,
- éviter les pertes d'informations entre conversations, commandes et stocks,
- suivre les marges en tenant compte du **prix d'achat réel** (via lots d'achat),
- identifier les **meilleurs fournisseurs**,
- automatiser les **confirmations, remerciements et relances**,
- piloter l'activité sur plusieurs **pays / villes / quartiers**.

2.2 Types d'utilisateurs

Super admin nafankap

- Gère la plateforme globale, les plans d'abonnement, le support, le monitoring.

Admin boutique (tenant admin)

- Configure la boutique (pays, ville, quartier, canaux Meta).
- Gère les utilisateurs internes.
- Gère le catalogue produits, fournisseurs, stocks.
- Suit les commandes, factures et analytics.
- Gère l'abonnement nafankap.

Opérateur boutique

- Répond aux clients depuis l'inbox.
- Crée et suit les commandes.
- Assigne les livreurs.
- Gère les interactions quotidiennes.

(Futur) Livreur avec accès restreint

- Accès limité aux commandes qui lui sont assignées.

Client final

- N'a pas d'accès direct à nafankap.
- Interagit via WhatsApp, Facebook ou Instagram.

2.3 Environnement de fonctionnement

Application web responsive (desktop, mobile).

Frontend et backend hébergés dans le cloud.

Base de données relationnelle (PostgreSQL).

Intégrations externes :

- WhatsApp Cloud API.
- Meta Graph API (Facebook Messenger, Instagram DM).
- Lemon Squeezy (paiements par carte).
- Flutterwave (paiements Mobile Money).
- Provider SMS/WhatsApp pour OTP.

2.4 Contraintes

- Dépendance à la disponibilité des API externes (Meta, Lemon Squeezy, Flutterwave, SMS provider).
- Respect des limitations de taux (rate limits).
- Respect des exigences de protection des données (type RGPD : confidentialité, minimisation, accès par tenant).

3. Exigences fonctionnelles

Les exigences sont regroupées par module.

3.1 Authentification & gestion des utilisateurs (F-AUTH)

F-AUTH-01 – Connexion par téléphone (OTP)

- L'utilisateur peut se connecter uniquement avec son numéro de téléphone.
- Un OTP est envoyé par SMS ou WhatsApp.
- La connexion n'est validée qu'après saisie du bon OTP.

F-AUTH-02 – Connexion par e-mail + mot de passe

- L'utilisateur peut créer un compte avec e-mail + mot de passe.
- Il peut ensuite se connecter avec ces identifiants.

F-AUTH-03 – Envoi & validation de l'OTP

- L'OTP a une durée de validité limitée (ex. 5 minutes).
- Le nombre de tentatives erronées est limité (ex. 3).

F-AUTH-04 – Réinitialisation de mot de passe

- Possibilité de demander un lien de réinitialisation par e-mail.

F-AUTH-05 – Crédit de tenant à la première inscription

- Lorsqu'un nouvel utilisateur crée une boutique, le système crée automatiquement :
 - un nouvel utilisateur,
 - un nouveau tenant.

F-AUTH-06 – Rôles

- Rôles supportés :
 - SUPER_ADMIN (plateforme),

- ADMIN (admin boutique),
- OPERATOR (opérateur boutique).

3.2 Gestion des tenants & utilisateurs internes (F-TENANT)

F-TENANT-01 – Profil boutique

L'admin boutique peut définir / modifier :

- nom de la boutique,
- pays (obligatoire),
- ville (obligatoire),
- quartier (optionnel),
- téléphone, e-mail, logo, fuseau horaire.

F-TENANT-02 – Invitations par e-mail

- Invitation d'un utilisateur interne par e-mail (envoi d'un lien d'invitation).

F-TENANT-03 – Invitations par téléphone

- Invitation par numéro de téléphone (lien ou code envoyé par SMS / WhatsApp).

F-TENANT-04 – Acceptation d'invitation

- L'invité peut :
 - soit créer un compte e-mail + mot de passe,
 - soit valider son numéro par OTP.

F-TENANT-05 – Gestion des utilisateurs internes

- L'admin boutique peut :
 - consulter la liste des utilisateurs de la boutique,
 - voir leurs rôles,
 - changer les rôles (ADMIN / OPERATOR),
 - activer / désactiver des comptes.

F-TENANT-06 – Multi-tenant par utilisateur (extension)

- Un même utilisateur peut appartenir à plusieurs tenants (boutiques).

3.3 Clients (CRM unifié) (F-CRM)

F-CRM-01 – Fiche client unique (Dossier Client)

- Une fiche client unique par personne, regroupant :
 - ses **méthodes de contact** (WhatsApp, Facebook, Instagram, e-mail...),
 - toutes ses **conversations**,
 - ses **commandes et factures**.

F-CRM-02 – Méthodes de contact client

- Chaque client peut avoir plusieurs méthodes de contact :
 - numéro de téléphone / WhatsApp,
 - ID Facebook Messenger,
 - ID Instagram,

- e-mail (facultatif).

F-CRM-03 – Données géographiques client

- La fiche client contient :
 - pays (obligatoire),
 - ville (obligatoire),
 - quartier (optionnel).

F-CRM-04 – Crédit automatique de client

- À la réception d'un message provenant d'un contact inconnu, nafankap crée automatiquement une fiche client minimale.

F-CRM-05 – Fusion de clients

- Un utilisateur interne peut fusionner deux fiches client lorsqu'il s'agit de la même personne, pour regrouper :
 - conversations,
 - commandes,
 - factures.

F-CRM-06 – Recherche client

- Recherche par :
 - nom,
 - numéro de téléphone,
 - e-mail,
 - ville,
 - éventuellement canal d'origine.

F-CRM-07 – Liaison d'identité Facebook → WhatsApp (extension)

- Depuis une conversation Facebook/Instagram :
 - l'opérateur peut ajouter un numéro WhatsApp au dossier.
 - Si ce numéro est déjà utilisé par un autre dossier, le système propose une **fusion**.
 - Si le numéro est nouveau, il est ajouté au dossier courant et l'onglet WhatsApp devient disponible.

3.4 Inbox omnicanale (F-INBOX)

F-INBOX-01 – Liste des conversations globales

Le système doit afficher une liste de conversations avec, pour chaque ligne :

Liste de conversations avec :

- client,
- canal (icône),
- dernier message
- date du dernier message,
- statut (non lu, en cours, résolu),
- opérateur assigné (optionnel).

F-INBOX-02 – Conversations par client

- Depuis la fiche client (Dossier Client), l'utilisateur voit l'ensemble des conversations du client.
- La vue utilise des **onglets par canal** (WhatsApp, Facebook, Instagram) dans la **Vue Dossier**.

- L'utilisateur peut basculer d'un onglet à l'autre sans quitter le dossier.

F-INBOX-03 – Détail conversation

Chaque onglet de conversation affiche :

- tous les messages du canal correspondant,
- dans l'ordre chronologique,
- avec indication du sens (entrant/sortant),
- la date et l'heure.

Les messages de différents canaux ne sont jamais mélangés dans une même timeline.

F-INBOX-04 – Envoi de messages

L'utilisateur peut envoyer des messages texte, images, documents depuis nafankap.

Le système route automatiquement le message vers le canal approprié.

F-INBOX-05 – Lien vers CRM et commandes

Depuis une conversation, l'utilisateur doit pouvoir :

- accéder à la fiche client,
- consulter la liste des commandes,
- créer une nouvelle commande.

F-INBOX-06 – Modèles de réponses

Possibilité de configurer et utiliser des modèles de réponses (messages rapides, scripts).

F-INBOX-07 – Contrainte 24h WhatsApp (extension)

- Sur les conversations WhatsApp :
 - Si le dernier message du client date de plus de 24h, la saisie libre est **bloquée**.
 - L'utilisateur doit sélectionner un **Template WhatsApp** validé par Meta pour relancer la discussion.

F-INBOX-08 – Médias pérennes (extension)

- Les images, preuves de paiement, voaux reçus sont **téléchargés** et stockés de façon pérenne par nafankap.
- L'interface doit permettre de les consulter même après expiration des liens Meta.

3.5 Produits, fournisseurs & lots d'achat (F-PROD / F-SUPPLIER / F-STOCK)

F-SUPPLIER-01 – Fournisseurs

Un fournisseur contient au minimum :

- nom,
- pays (obligatoire),
- ville (obligatoire),
- quartier (optionnel),
- téléphone (optionnel),
- notes (optionnel).

F-PROD-01 – Produits

Un produit contient au minimum :

- nom,
- description (optionnelle),
- image (optionnelle),
- prix de vente conseillé,
- catégorie (optionnelle).

F-STOCK-01 – Lot d'achat

Un lot d'achat (stock entrant) contient :

- produit,
- fournisseur,
- quantité achetée,
- prix d'achat unitaire,
- date d'achat,
- quantité restante,
- éventuellement pays / ville / quartier (localisation du stock).

F-STOCK-02 – Déstockage et choix de lot

- Lors d'une vente :
 - l'utilisateur peut **laisser le système choisir automatiquement** les lots (stratégie par défaut : FIFO),
 - ou, si besoin, choisir explicitement les lots à consommer.
- **Pour l'opérateur et le client**, la commande n'affiche qu'une **ligne produit** (ex. « Article × 5 »), même si en interne plusieurs lots sont consommés.
- La répartition multi-lots est gérée **en arrière-plan** pour le calcul des marges.

F-STOCK-03 – Mise à jour des stocks

- Lorsque la commande atteint un statut de validation (ex. Confirmé), les quantités vendues décrémentent les quantités restantes des lots concernés.

3.6 Commandes & livraisons (F-ORD / F-DEL)

F-ORD-01 – Commande

Une commande doit être liée à :

- un client (Dossier Client),
- une ou plusieurs lignes produit,
- des frais de livraison éventuels,
- des informations de livraison (pays, ville, quartier).

F-ORD-02 – Adresse de livraison

Adresse minimale :

- pays (obligatoire),
- ville (obligatoire),
- quartier (optionnel),
- commentaire (optionnel).

F-ORD-03 – Statuts de commande

- Nouveau,

- Confirmé,
- Transmis au livreur,
- En livraison (optionnel, peut être simplifié),
- Livré,
- Annulé.

F-DEL-01 – Livreurs / partenaires de livraison

Un livreur contient :

- nom,
- pays (obligatoire),
- ville (obligatoire),
- quartier (optionnel),
- téléphone / WhatsApp,
- notes (optionnel).

F-DEL-02 – Assignation

- Une commande peut être assignée à un livreur.

F-DEL-03 – Récap de livraison (bordereau)

- nafankap génère un récap texte contenant :
 - client, téléphone, pays/ville/quartier,
 - détail des produits,
 - montants,
- pour être envoyé au livreur (ex. via WhatsApp).

3.7 Facturation (F-BILL)

F-BILL-01 – Génération de facture PDF

Pour chaque commande validée, le système permet de générer une facture PDF contenant :

- Infos de la boutique (nom, pays, ville, quartier éventuel).
- Infos du client (nom, pays, ville).
- Détail des produits (nom, quantité, prix de vente unitaire).
- Frais de livraison.
- Total.

F-BILL-02 – Stockage facture

- La facture (URL PDF) est stockée et rattachée à la commande.

F-BILL-03 – Envoi de facture

- Envoi via WhatsApp (document ou lien).
- Ou par e-mail (si disponible).

F-BILL-04 – Historique

- Consultation de l'historique des factures par :
 - date,
 - client,
 - commande.

3.8 Automatisations (F-AUTO)

F-AUTO-01 – Confirmation automatique

- Quand une commande passe à Confirmé, le système peut envoyer automatiquement un message de confirmation au client sur son canal principal.

F-AUTO-02 – Remerciement

- Quand une commande passe à Livré, le système peut envoyer un message de remerciement / satisfaction.

F-AUTO-03 – Relances

- Possibilité de définir des règles, par exemple :
 - « Si un message client reste sans réponse pendant X heures, alors envoyer un rappel ou notifier un opérateur. »

F-AUTO-04 – Notifications abonnement

- En cas d'abonnement proche de l'expiration ou expiré, l'admin boutique est notifié (WhatsApp, e-mail, bannière).

3.9 Abonnements & paiements (F-SUB)

F-SUB-01 – Plans d'abonnement

- nafankap propose plusieurs plans (Starter, Pro, Business, ...) avec tarifs et limitations (nombre d'utilisateurs, fonctionnalités, quotas...).

F-SUB-02 – Paiement via Lemon Squeezy (carte)

- L'admin boutique peut payer son abonnement par carte via Lemon Squeezy (checkout externe).

F-SUB-03 – Paiement via Flutterwave (Mobile Money)

- L'admin boutique peut payer via Mobile Money (Flutterwave), adapté aux pays africains.

F-SUB-04 – Webhooks paiement

À la réception d'un paiement (webhook Lemon Squeezy / Flutterwave), le système doit :

- mettre à jour le **statut d'abonnement** du tenant,
- mettre à jour la **période de validité**,
- créer un enregistrement de **facture d'abonnement**.

F-SUB-05 – Notifications & restrictions si non payé (extension)

- Le système notifie l'admin **3 jours avant** l'expiration de l'abonnement (WhatsApp, e-mail).
- Si l'abonnement est expiré à la date J, une **période de grâce de 3 jours** est appliquée.
- À J+3 :
 - le tenant passe en **mode Lecture Seule** :
 - consultation de l'historique toujours possible,
 - **création de nouvelles commandes et nouveaux messages bloquée**.

3.10 Analytics & marges (F-ANALYT)

F-ANALYT-01 – Marges

- Calcul de la marge brute pour :
 - chaque commande,
 - chaque ligne de commande,
 - chaque produit,
 - chaque fournisseur.

F-ANALYT-02 – Filtres géographiques

- Analyses filtrables par :
 - pays,
 - ville,
 - quartier.

F-ANALYT-03 – Statistiques par fournisseur

- Pour chaque fournisseur :
 - volume d'achat,
 - volume de vente associé,
 - marge totale,
 - taux de commandes livrées / annulées.

F-ANALYT-04 – Dashboard

- Indicateurs clés :
 - nombre de commandes sur une période,
 - chiffre d'affaires,
 - marge globale,
 - top produits,
 - top fournisseurs.

3.11 Administration globale (F-ADMIN)

- Liste des tenants avec : pays, ville, plan, statut (actif, suspendu, expiré).
- Gestion des plans : création, modification, désactivation.
- Suspension / réactivation d'un tenant.
- Accès aux logs techniques et événements clés (webhooks, erreurs, etc.).

4. Exigences non fonctionnelles (résumé)

Performance :

- Temps de réponse des pages principales < 1 seconde en usage normal.
- Traitement des webhooks < 2 secondes.

Disponibilité :

- Objectif de disponibilité $\geq 99\%$ (hors maintenances planifiées).

Sécurité :

- Toutes les communications via HTTPS.
- Isolation stricte des données par tenant.
- OTP stockés de manière sécurisée (hash + expiration).
- Webhooks vérifiés (signature / secret partagé).

Confidentialité :

- Les données clients sont visibles uniquement par leur tenant.
- Pas de stockage de données sensibles (OTP, tokens d'API...) en clair dans les logs.

Utilisabilité :

- Interface responsive, mobile-friendly.
- Langue par défaut : français.
- Simplicité d'usage pour des commerçants non techniques.

SDS – Spécification Technique nafankap

1. Architecture technique globale

- Frontend + Backend léger :
 - Next.js (App Router)
 - Pages : dashboard, inbox, clients, produits, fournisseurs, stock, commandes, factures, abonnements, paramètres.
- Base de données :
 - NeonDB (PostgreSQL serverless).
- ORM / Migrations :
 - Drizzle ORM.
- Authentification :
 - BetterAuth :
 - auth e-mail + mot de passe,
 - provider custom pour téléphone + OTP.
 - support multi-tenant.
- Stockage de fichiers :
 - UploadThing : factures PDF, images produits, médias clients (images, audios).
- Automatisation / intégrations :
 - n8n (auto-hébergé) pour orchestrer certains workflows :
 - téléchargement asynchrone des médias Meta,
 - relances automatiques,
 - rappels d'abonnement,
 - traitements de webhooks.
- Messagerie :
 - WhatsApp Cloud API,
 - Meta Graph API (Facebook Messenger, Instagram DM).

- Paiements :
 - Lemon Squeezy (cartes),
 - Flutterwave (Mobile Money).

2. Organisation des modules / dossiers (idée)

Application (App Router) structurée en sections :

- Dashboard.
- Inbox (Vue Dossier + conversations).
- Clients (CRM).
- Produits.
- Fournisseurs.
- Stock.
- Commandes.
- Factures.
- Abonnements / facturation SaaS.
- Paramètres (boutique, intégrations Meta, paiements, etc.).

API internes :

- Authentification.
- Webhooks WhatsApp / Meta.
- Webhooks Lemon Squeezy / Flutterwave.

Couche de services métier :

- Gestion des clients (création, fusion, affectation d'identités).
- Gestion des conversations & messages.
- Gestion des produits / lots / allocations.
- Gestion des commandes / livraisons / factures.
- Gestion des abonnements & facturation SaaS.
- Calcul d'analytics et de marges.

3. Modèle de données (schéma logique)

3.1 Tenants & utilisateurs

Tenants (boutiques)

- Identifiant unique.
- Nom.
- Pays, ville, quartier.
- Coordonnées (téléphone, e-mail).
- Plan d'abonnement, statut d'abonnement (actif, en retard, expiré, annulé).
- Dates de création / mise à jour.

Utilisateurs

- Identifiant unique.
- Référence vers un tenant (ou null pour SUPER_ADMIN).
- Téléphone, e-mail.
- Rôle (SUPER_ADMIN, ADMIN, OPERATOR).

- Statut actif / inactif.
- Dates de création / mise à jour.

OTP (table technique)

- Téléphone cible.
- Hash du code OTP.
- Date d'expiration.
- Compteur de tentatives.

3.2 Clients & contacts

Clients (Dossiers Client)

- Identifiant.
- Tenant associé.
- Nom du client.
- Pays, ville, quartier.
- Notes internes.
- Date de création.

Méthodes de contact

- Référence au client.
- Type (téléphone, WhatsApp, Facebook Messenger, Instagram DM, e-mail).
- Valeur (numéro, ID, adresse e-mail).
- Indicateur « principal » ou non.

3.3 Conversations & messages

Conversations

- Identifiant.
- Tenant.
- Client.
- Canal (WhatsApp, Facebook, Instagram).
- Statut (ouverte, en attente, fermée).
- Utilisateur assigné (optionnel).
- Date du dernier message.

Messages

- Référence à une conversation.
- Tenant.
- Direction (entrant, sortant).
- Type d'expéditeur (client, utilisateur, système).
- Contenu (texte ou structure pour médias).
- Canal.
- Date d'envoi / réception.
- Référence technique externe (ID Meta) pour déduplication.

3.4 Fournisseurs, produits & lots d'achat

Fournisseurs

- Tenant.

- Nom.
- Pays, ville, quartier.
- Téléphone.
- Notes.

Produits

- Tenant.
- Nom.
- Description.
- URL d'image.
- Prix de vente de base.

Lots d'achat (entrées de stock)

- Tenant.
- Produit concerné.
- Fournisseur d'origine.
- Prix d'achat unitaire.
- Quantité achetée.
- Quantité restante.
- Localisation (pays, ville, quartier).
- Date d'achat.

Mouvements de stock (optionnel, audit)

- Tenant.
- Lot concerné.
- Type (vente, ajustement).
- Quantité.
- Date.
- Raison (commentaire).

3.5 Commandes, livraisons & factures

Commandes

- Tenant.
- Client.
- Statut (nouvelle, confirmée, transmise au livreur, en livraison, livrée, annulée).
- Pays, ville, quartier de livraison.
- Commentaires de livraison.
- Frais de livraison.
- Montant total de vente.
- Coût total d'achat (somme des coûts d'achat des items, pour marge).
- Dates de création / mise à jour.

Lignes de commande (order items)

- Commande associée.
- Produit.
- Quantité.
- Prix de vente unitaire.

- Liens internes vers un ou plusieurs lots d'achat consommés (allocation de stock) avec coûts d'achat unitaires.

Partenaires de livraison (livreurs)

- Tenant.
- Nom.
- Pays, ville, quartier.
- Téléphone.
- Notes.

Factures de commande

- Commande associée.
- Tenant.
- URL du PDF.
- Montant.
- Date de création.

3.6 Abonnements & paiements

Abonnements

- Tenant.
- Plan.
- Statut (actif, en retard, annulé, expiré).
- Période actuelle : début / fin.
- Fournisseur de paiement (Lemon Squeezy, Flutterwave).
- Identifiant externe d'abonnement.

Factures SaaS (abonnement nafankap)

- Tenant.
- Fournisseur (Lemon Squeezy / Flutterwave).
- Identifiant externe de facture.
- Montant.
- Devise.
- Statut (payée, échouée, en attente).
- Date de création.

4. Intégrations principales

4.1 WhatsApp Cloud API & Meta Graph API

- Webhook dédié pour la réception des messages WhatsApp.
- Webhook dédié pour la réception des messages Facebook/Instagram.

Traitement typique d'un message entrant :

1. Vérifier la signature / authenticité.
2. Identifier le tenant (via configuration de l'app Meta).

3. Identifier ou créer le client et sa méthode de contact correspondante.
4. Créer ou trouver la conversation correspondante (par canal).
5. Insérer un message entrant.
6. Mettre à jour la date du dernier message de la conversation.

4.2 Lemon Squeezy

Utilisation de liens de checkout par plan.

Webhook de paiement pour les événements de type :

- création d'abonnement,
- paiement d'abonnement,
- création de commande.

Mise à jour des abonnements et des factures SaaS à partir des événements reçus.

4.3 Flutterwave (Mobile Money)

- Utilisation de Payment Links / checkout pour Mobile Money.
- Webhook de paiement pour mise à jour des abonnements.
- Rattachement au tenant via des métadonnées (identifiant de boutique dans le paiement).

5. Sécurité & multi-tenant

- Tous les endpoints API récupèrent le tenant à partir de la session utilisateur (ou de la configuration de l'app pour les webhooks).
- Toutes les requêtes base de données sont filtrées par tenant (et idéalement renforcées par des règles au niveau de la base).
- Les webhooks sont associés à un tenant via l'identifiant de l'app Meta ou des métadonnées de paiement.
- Validation stricte des payloads.
- Vérification systématique des signatures des webhooks (Lemon Squeezy, Flutterwave, Meta).

6. Principaux flux techniques

6.1 Flux : un client écrit sur WhatsApp

1. Le client envoie un message à la boutique sur WhatsApp.
2. Meta appelle /api/webhooks/whatsapp.
3. L'API nafankap :
 - vérifie la signature,
 - identifie le tenant,
 - cherche une méthode de contact correspondante (numéro/ID),
 - crée au besoin un client et une méthode de contact,
 - crée ou retrouve la conversation,
 - insère un message entrant.

4. L'interface utilisateur (inbox) est rafraîchie, et l'opérateur voit le nouveau message.

6.2 Flux : l'opérateur crée une commande et l'envoie au livreur

1. Dans une conversation client, l'opérateur clique “Créer une commande”.
2. Il sélectionne les produits, les quantités.
3. Pour chaque produit, le système propose les lots d'achat disponibles (par défaut : FIFO).
4. L'opérateur choisi (ou accepté la proposition automatique), le système calcule :
 - o le coût d'achat (purchase_price_unit),
 - o la marge pour chaque ligne.
5. Il renseigne l'adresse de livraison : pays, ville, quartier.
6. La commande est créée (statut Nouveau).
7. Lorsqu'il assigne un livreur et passe en Transmis au livreur, nafankap génère un récap texte.
8. L'opérateur envoie ce récap au livreur (via WhatsApp par exemple).

6.3 Flux : génération et envoi de facture

1. Quand la commande est confirmée, l'opérateur demande une facture.
2. Le backend génère un PDF (via jsPDF / autre).
3. Le PDF est uploadé sur UploadThing → URL stockée dans order_invoices.
4. L'opérateur envoie la facture au client (WhatsApp ou e-mail).

6.4 Flux : paiement d'abonnement via Lemon Squeezy

1. L'admin boutique choisit un plan dans la page Abonnement”.
2. nafankap le redirige vers le checkout Lemon Squeezy correspondant.
3. Le paiement est effectué.
4. Lemon Squeezy envoie un webhook à /api/webhooks/lemonsqueezy.
5. nafankap :
 - o vérifie le secret,
 - o met à jour subscriptions (status ACTIVE, dates),
 - o créer un saas_invoices (status PAID).
6. L'abonnement est actif, les restrictions éventuelles sont levées.