**📑 Functional Requirements Document (FRD)**

**Project**: WhatsApp SaaS AI Agent  
**Version**: Draft v1.0 (Sep 2025)  
**Owner**: abhish1479  
**Status**: Work-in-progress

**1. Introduction**

**1.1 Purpose**

This FRD describes the functional requirements for a **SaaS platform** that enables small and medium businesses (SMBs) to deploy AI-powered WhatsApp agents. These agents assist with:

**• Customer onboarding**

**• Lead conversion**

**• Product/service inquiries**

**• Business-owner controlled RAG updates**

**1.2 Scope**

The system includes:

**• Business Owner Onboarding App** (Flutter-based wizard)

**• Backend SaaS Platform** (FastAPI + DB + Workers)

**• AI & RAG Layer** (LLM integration with knowledge base)

**• WhatsApp Integration** (via WhatsApp Business API)

**• Billing & Credits System**

**• Monitoring & Analytics**

**2. System Overview**

**1 Business Owner Onboarding**

◦ Owner signs up, provides business details, selects type, uploads docs, sets up services/products, configures workflows.

◦ Payment setup is required to activate.

**2 Knowledge Base (RAG)**

◦ Owner uploads documents, FAQs, services, or updates via WhatsApp.

◦ Documents processed, embedded, stored in vector DB.

◦ Owner can add/remove/update RAG data through backend UI or WhatsApp.

**3 Customer Conversations**

◦ Customers reach business via WhatsApp.

◦ AI agent retrieves context from RAG, generates responses via LLM.

◦ Handles lead qualification and onboarding flows using templates.

**4 Billing & Credits**

◦ Each business operates on a credit/subscription model.

◦ Credits deducted per API call, message, or AI token usage.

◦ Wallet balance accessible via dashboard.

**5 Monitoring & Analytics**

◦ Track metrics: conversions, active users, credits consumed, latency.

◦ Dashboards for business owners and internal ops.

**3. Functional Requirements**

**3.1 User Management**

**• FR-1**: Business owner can register/login via email/WhatsApp authentication.

**• FR-2**: Multi-tenant architecture ensures each business has isolated data.

**• FR-3**: Roles: Owner, Admin, Customer.

**3.2 Business Onboarding**

**• FR-4**: Owner must complete onboarding wizard (business info, services, workflows, payment).

**• FR-5**: Owner can review and activate account.

**• FR-6**: Onboarding progress stored in DB.

**3.3 Knowledge Base (RAG)**

**• FR-7**: Owner can upload documents (PDF, DOCX, TXT, website URLs).

**• FR-8**: System extracts, embeds, and indexes documents.

**• FR-9**: Owner can update knowledge base directly via WhatsApp.

**• FR-10**: RAG policies ensure content moderation before indexing.

**3.4 Conversations & Leads**

**• FR-11**: Customer messages are received via WhatsApp webhook.

**• FR-12**: AI agent retrieves context, generates reply, and sends back via WhatsApp API.

**• FR-13**: System distinguishes between owner messages (RAG updates) and customer inquiries.

**• FR-14**: Leads are created/updated based on inquiries.

**• FR-15**: Conversations auto-expire after inactivity threshold.

**3.5 Billing & Wallet**

**• FR-16**: Credits deducted per conversation/session.

**• FR-17**: Wallet balance exposed via API and dashboard.

**• FR-18**: Subscription/payment integration with Stripe/PayPal.

**• FR-19**: Credit ledger logs all transactions.

**3.6 Templates & Workflows**

**• FR-20**: Predefined session templates for onboarding, FAQs, promotions.

**• FR-21**: Business owner can customize workflows.

**3.7 Analytics & Monitoring**

**• FR-22**: Track KPIs: daily conversations, conversion rate, active leads.

**• FR-23**: Admin dashboards display aggregated metrics.

**• FR-24**: Prometheus scrapes system metrics; Grafana dashboards visualize them.

**4. Non-Functional Requirements (NFRs)**

**• NFR-1**: Scalability — Support 1,000+ concurrent tenants.

**• NFR-2**: Security — Enforce HTTPS, JWT-based auth, encrypted data at rest.

**• NFR-3**: Compliance — Adhere to WhatsApp Business Policy, GDPR.

**• NFR-4**: Performance — 95% responses under 2 seconds.

**• NFR-5**: Availability — 99.9% uptime SLA.

**5. System Architecture (Summary)**

**1 Frontend (Flutter)** → calls **Backend APIs**

**2 Backend (FastAPI)** → manages onboarding, billing, RAG, conversations

**3 Workers** → async tasks (doc ingestion, summarization, webhook processing)

**4 Vector DB** → RAG storage

**5 WhatsApp API** → message exchange

**6 LLM API** → AI-powered responses

**7 Billing system** → Stripe/PayPal + internal credit ledger

**8 Monitoring** → Prometheus + Grafana

**6. Assumptions**

• Owners have an active WhatsApp Business number.

• All customers interact via WhatsApp (no web chat).

• LLM provider (e.g., OpenAI) has stable APIs and pricing.

**7. Open Questions**

• Should system support **multi-language conversations** (auto-detect and translate)?

• Should business owners have a **web dashboard** in addition to Flutter app?

• How will refunds/chargebacks be handled in billing?

• Which vector DB (Pinecone, Weaviate, PGVector) is final choice?