Saramedico Platform Master Development Specification

Project Type: End-to-End HIPAA-Compliant Medical AI SaaS

Compliance: HIPAA (USA), HITECH, SOC2 Readiness

Development: Web Based and Playstore App

Target Audience: Solo Practitioners, Small Clinics, Enterprise Networks

📑 Table of Contents

Project Scope Statement: Saramedico AI Ecosystem

Project Overview Design, develop, and launch Saramedico, a full-stack, HIPAA-compliant Medical AI SaaS platform. The system will consist of a high-performance public marketing interface and a secure, multi-tenant web application. It is designed to serve a spectrum of users from solo practitioners to enterprise healthcare networks under a unified subscription model featuring a 10-day automated free trial.

The project encompasses end-to-end delivery: architecture design, full-stack development, security hardening, cloud deployment, and production launch.

Core Functional Deliverables

A. **Public Web Platform (saramedico.com)** *A production-grade, SEO-optimized marketing interface serving as the primary acquisition channel.*

Content Architecture: Dedicated pages for Product Overview, Solutions (Reader/Listener Modules), Pricing, and Security/Trust.

Conversion Funnels: Integrated free trial onboarding, lead capture forms, and enterprise demo booking workflows.

Trust & Compliance: Prominent HIPAA disclosures, Trust Center, and legal frameworks (Privacy Policy, Terms, BAA Request forms).

Support: Integrated customer support ticketing and knowledge base.

B. **Secure Medical AI Web Application (The SaaS Product)** *A HIPAA-compliant, role-based application delivering two core functional modules:*

Module A: Document Intelligence (The Reader):

Secure ingestion and OCR of PDF/Image medical records.

AI-powered chart summarization, chronological timeline extraction, and RAG-based Medical Q&A.

Automated PII redaction and citation linking.

Module B: Ambient Clinical Agent (The Listener):

Real-time, browser-based audio recording of clinical consultations.

Medical-grade multi-speaker transcription.

Automated generation of structured clinical documentation (SOAP notes).

Intelligent clinical tagging (e.g., "Anxious", "Non-compliant") and remedy suggestions.

Patient Management:

Encrypted patient history and unified profile views (Documents + Visits).

Secure patient communication via SMS and Email (via Twilio/SendGrid).

ICD-10 billing code suggestions.

C. Subscription & Billing Infrastructure

Trial Logic: Automated 10-day free trial with strict usage gating (e.g., 3 consult limit).

Stripe Integration: Recurring billing logic for tiered plans (Standard, Premium, Clinic Team).

Enterprise Handling: Support for per-seat billing, custom invoicing, and dunning management (payment failure handling).

Backend & Infrastructure

API Architecture: High-performance, API-driven backend services handling authentication, AI orchestration, billing events, and notifications.

Data Layer: Multi-tenant architecture utilizing encrypted relational databases (PostgreSQL) and vector storage (pgvector) for AI memory.

AI Orchestration: scalable pipelines integrating Large Language Models (LLMs) and Speech-to-Text engines.

Storage: Secure, encrypted object storage (S3) for audio and document retention.

Security, Compliance & Enterprise Readiness

Encryption Standard: AES-256 encryption for data at rest and TLS 1.3 for data in transit.

Access Control: Mandatory Multi-Factor Authentication (MFA) and strict Role-Based Access Control (RBAC).

Auditability: Immutable audit logging for every PHI access event (View, Edit, Export).

Tenant Isolation: Logical separation of data across different organizations to prevent leakage.

Enterprise Features: Readiness for Single Sign-On (SSO), dedicated tenant infrastructure, and automated compliance exports.

Scalability & Deployment

Architecture: Cloud-native, containerized architecture designed to auto-scale based on traffic load.

Availability: Deployment across multiple availability zones for high redundancy.

Integration: API-first design to facilitate future third-party EMR/EHR integrations.

1. Executive Summary

Saramedico is a scalable, multi-tenant AI platform designed to automate healthcare administration. The platform is built on a **Modular Architecture**, allowing users to subscribe to specific capabilities based on their needs.

The system is composed of two core functional modules:

- **Module A (Standard):** A retrospective document analysis engine for reviewing PDF medical records.
- **Module B (Premium):** A real-time ambient intelligence agent for recording and summarizing live patient consultations.

2. Business Strategy: Modules & Pricing Logic

The platform separates features into "Standard" and "Premium" to capture different market segments. This logic must be enforced by the backend API.

2.1 The Module Split

- **Standard Plan (The "Reader"):** Targeted at administrative staff and legal nurse consultants. Includes full access to PDF Uploads, OCR, Timeline Extraction, and RAG Chat. **Does NOT include Audio Recording.**
- **Premium Plan (The "Listener"):** Targeted at practicing physicians. Includes everything in Standard **PLUS** Live Audio Recording, Multi-speaker Transcription, and SOAP Note Generation.

2.2 Detailed Subscription Tiers

Tier Name	Target User	Monthly Price	Standard Features (Docs)	Premium Features (Audio)	Team Features
Free Trial	New User	**$0** (7 Days)	5 Uploads (100 pgs)	3 Live Visits	Single User
Solo Standard	Admin / Reviewer	**$29**	**2,500 Pages/mo**	❌ Locked	Single User
Solo Premium	Doctor	**$99**	**Unlimited**	**Unlimited**	Single User
Clinic Team	Small Practice	**$89 / user**	**Unlimited**	**Unlimited**	Shared Folders, Admin Dashboard
Enterprise	Hospital	**Custom**	Custom Retention	Custom Integration	SSO, Dedicated Support

3. Visual Design System (UI/UX Guidelines)

- **Color Palette:**
    - **Primary:** Deep Medical Blue (#0F172A) – Used for primary buttons, headers, and active states.
    - **Secondary:** Soft Sky Blue (#E0F2FE) – Used for backgrounds on active tabs and "trust" highlights.
    - **Accent:** Success Green (#10B981) – Used for "Approve," "Safe," and "Complete" indicators.
    - **Alert:** Clinical Red (#EF4444) – Used solely for "Stop Recording" and destructive actions.
    - **Background:** Clean White (#FFFFFF) and Light Gray (#F8FAFC) for dashboard panels.
- **Typography:**
    - **Font Family:** Inter or Plus Jakarta Sans. Clean, sans-serif, highly legible at small sizes.

    - **Sizing:** Base size 16px. Headers 24px-32px.
- **Design Principle:** "High Information Density, Low Clutter." Doctors need to see a lot of data without feeling overwhelmed.

4. Public Website Architecture (Detailed Page-by-Page)

The public site must be SEO-optimized and built using Next.js (Static Site Generation) for speed.

4.1 Landing Page (Home)

- **Hero Section:**
    - **Headline:** "The Operating System for Modern Medical Practice."
    - **Sub-headline:** "One secure platform to automate your clinical notes and summarize your patient history."
    - **Visual:** A high-quality device mockup showing the **Mobile App** (Recording) next to the **Desktop Dashboard** (Reviewing).
    - **CTA:** "Start Free 10-Day Trial" (Primary) and "View Pricing" (Secondary).
- **Problem/Solution Grid:**
    - *Problem:* "Drowning in Paperwork?" -> *Solution:* "Upload 500 pages, get a summary in seconds."
    - *Problem:* "Late Nights Charting?" -> *Solution:* "Record your visit, let AI write the SOAP note."
- **Social Proof:** Logos of compliant standards (HIPAA, HITECH, SOC2). User testimonials carousel.

4.2 Features & Solutions

- **Tabbed Interface:** Users toggle between **"For Reviewers"** and **"For Clinicians"**.
- **"For Reviewers" Content:** Deep dive into OCR, Timeline View, and PII Redaction. Animated GIFs showing a document being scanned.
- **"For Clinicians" Content:** Deep dive into Ambient Recording, Multi-language support, and EMR integration. Audio samples showing transcription accuracy.

4.3 Pricing & Plan Selection

- **Toggle Switch:** [Bill Monthly] vs [Bill Yearly (Save 20%)].
- **The Comparison Table:** A detailed row-by-row comparison.
    - *Row 1:* "PDF Analysis" (Checked for All).
    - *Row 2:* "Live Audio Recording" (❌ Standard | ✅ Premium).
    - *Row 3:* "Team Collaboration" (❌ Solo | ✅ Clinic).
- **FAQ Section (Pricing Specific):** "Do I need a credit card?" "Can I upgrade later?"

4.4 About Us & Mission

- **Mission Statement:** "To restore patient-doctor connection by removing the keyboard from exam room."
- **Leadership/Team:** Photos and bios of key team members (CEO, CTO, Medical Advisors).
- **Values:** "Privacy First," "Clinical Accuracy," "Speed."

4.5 Security & Compliance Center

- **Trust Badge Area:** High-res badges for AES-256, TLS 1.3, and HIPAA.
- **Architecture Diagram:** A simplified visual showing how data flows securely into the encrypted cloud (AWS).
- **Legal Documents:** Links to "Terms of Service," "Privacy Policy," and a **"Request BAA"** form for covered entities.

4.6 FAQ & Knowledge Base

- **Search Bar:** "How can we help?"
- **Categories:** *Getting Started*, *Security*, *Billing*, *Troubleshooting*.
- **Common Questions:**
    - "Is my data used to train AI?" (Answer: No, we have a Zero-Retention policy).
    - "What happens to my data if I cancel?" (Answer: You can export everything, then it is permanently deleted).

4.7 Contact & Support

- **Contact Form:** Fields for Name, Email, Subject (Sales vs Support), and Message.
- **Direct Contact:** Support Email (help@saramedico.com) and Sales Email (sales@...).
- **Office Address:** Physical HQ address (adds legitimacy).

5. Secure Application Architecture (The Product)

Access requires secure Login. This is a Single Page Application (SPA).

5.1 Authentication & Onboarding

- **Login Screen:** Email/Password. "Forgot Password" flow.
- **MFA Challenge:** User enters 6-digit SMS/Auth code. **Mandatory.**
- **Onboarding Wizard:** First-time users see a 3-step tour:
    1. "Select your specialty."
    2. "Upload a sample document."
    3. "Test your microphone."

5.2 The Unified Dashboard

- **Top Header:** Global Search (Patients/Docs), Notification Bell, Profile Menu.
- **Left Sidebar (Navigation):**
    - Home (Dashboard Widgets)
    - Live Consult (Microphone Icon)
    - Chart Review (Document Icon)
    - Patients (Directory)
    - Team (If Clinic Tier)
    - Settings

5.3 Module A: Document Intelligence Studio (Standard)

- **Layout:** Split Screen (50% Viewer / 50% AI Panel).

- **Features:**
    - **Smart Uploader:** Drag & Drop. Checkbox for [ ] Auto-Redact PII.
    - **AI Timeline:** A vertical list of extracted dates (2023-01-10: MRI Scan). Clicking a date scrolls the PDF to that page.
    - **Citation Chat:** When AI answers a question, it appends [Page 4]. Clicking the link jumps the viewer to the exact highlight.

5.4 Module B: Live Consultation Agent (Premium)

- **Recording Mode:** Large, distraction-free "Start/Stop" button. Waveform visualizer. Language selector.
- **Review Mode (3-Column Layout):**
    - **Col 1 (Transcript):** Real-time text stream. Speaker labels (Dr. Smith, Patient).
    - **Col 2 (SOAP Note):** AI-generated clinical note. Editable text areas.
    - **Col 3 (Assist):**
        - **Tags:** AI suggests [Anxious], [Chronic Pain]. Click to accept.
        - **Remedies:** Search bar for home remedies (e.g., "Ice, Rest"). Click to add to Plan.
- **Action Bar:** Save to EMR | Copy Text | Email Patient.

5.5 Patient Management Directory

- **Unified View:** A table showing Patient Name, DOB, MRN.
- **Patient Profile:** Clicking a patient opens their profile containing two tabs:
    - **"Visits":** List of audio recordings/SOAP notes.
    - **"Documents":** List of uploaded PDFs.

5.6 Settings, Billing & Admin

- **My Profile:** Change password, update MFA.
- **Billing:** View current plan, update credit card (Stripe integration), download invoices.

- **Team (Clinic Only):** Invite new users via email. Set permissions (Admin vs Member).
- **Audit Logs:** A read-only table showing all account activity (Login, View Patient, Export) for compliance.

6. Technical Infrastructure & Scalability

6.1 Scalable Cloud Stack (AWS)

- **Compute:** AWS Fargate (Serverless Containers). Autoscales based on traffic.
- **Database:** PostgreSQL (AWS RDS) with pgvector. Multi-AZ deployment for redundancy.
- **Storage:** AWS S3 (Standard Class). Lifecycle policies to move old data to Glacier (cheaper).

6.2 AI Pipeline

- **Orchestration:** Python (FastAPI + LangChain).
- **Models:** AWS Bedrock (Claude 3.5 Sonnet) for reasoning; AWS Transcribe Medical for audio.
- **Security:** Private VPC. No public internet access for the database.

7. Development Roadmap & Phasing

This roadmap is strictly sequential. Phase 2 and Phase 3 can run in parallel if two separate development teams (Frontend vs. Backend) are available.

1.0 Phase 1: The Secure "Zero-Trust" Foundation

Goal: Establish the HIPAA-compliant infrastructure before writing application code.

1.1 AWS Infrastructure Setup (DevOps)

- [ ] **Account Hardening:** Create a fresh AWS account. Activate MFA on Root User. Create Admin IAM group with enforced MFA.
- [ ] **VPC Configuration:**
    - Create a Virtual Private Cloud (VPC) spanning 2 Availability Zones (e.g., us-east-1a, us-east-1b).
    - **Public Subnets:** Deploy NAT Gateways and Application Load Balancers (ALB) here.

    - **Private Subnets:** Restrict all Compute (Fargate), Database (RDS), and Cache (Redis) resources here. No direct internet access.
- [ ] **KMS Encryption Setup:**
    - Create a Customer Managed Key (CMK) in AWS KMS for "Saramedico-PHI".
    - Define Key Policy to allow usage only by specific IAM Roles (e.g., ecsTaskExecutionRole).
- [ ] **HIPAA Artifact:** Navigate to AWS Artifact and accept the **Business Associate Agreement (BAA)**.

1.2 Database & Storage Initialization

- [ ] **PostgreSQL RDS:**
    - Deploy RDS Postgres 15+ instance in Private Subnets.
    - Enable Storage Encryption using the KMS Key created above.
    - Enable IAM Database Authentication (to avoid hardcoding passwords).
    - Install pgvector extension: CREATE EXTENSION vector;.
- [ ] **S3 Bucket Strategy:**
    - Create saramedico-uploads-private bucket.
    - Enable "Block All Public Access".
    - Enable "Default Encryption" using KMS.
    - Enable "Object Lock" (WORM compliance) for Audit Logs.

1.3 Shared Backend Kernel (FastAPI)

- [ ] **Boilerplate:** Initialize FastAPI with Pydantic settings management.
- [ ] **Authentication Middleware:**
    - Integrate **AWS Cognito** as the Identity Provider (IdP).
    - Write a dependency get_current_user that decodes the JWT, verifies the signature against Cognito's JWKS, and checks for MFA_ENABLED=True.
- [ ] **Database ORM:**
    - Setup **SQLAlchemy** (Async Engine) or **Prisma**.

    - **Crucial:** Create a custom TypeDecorator for EncryptedString. This ensures that fields like patient_name are encrypted *by the application* before they even touch the database.

Phase 2: Frontend Core & The "Reader" Module

Goal: Build the Document Intelligence Engine (Module A).

2.1 Next.js Application Shell

- [ ] **Layout Architecture:**
    - Implement Next.js App Router (/app).
    - Create (auth) layout for Login/Register pages.
    - Create (dashboard) layout with the Persistent Sidebar and Header.
- [ ] **State Management:**
    - Initialize **Zustand** store for User Session (useUserStore) and UI State (useSidebarStore).
- [ ] **ShadcnUI Implementation:**
    - Install core components: Button, Dialog, Table, Toast, Form.
    - Customize tailwind.config.js with the Saramedico Color Palette (#0F172A).

2.2 Feature: Secure Document Upload

- [ ] **Frontend Uploader:**
    - Build a "Drag & Drop" zone using react-dropzone.
    - Implement **Presigned URLs**: The frontend requests a "One-Time Upload URL" from the Backend, then PUTs the file directly to S3 (bypassing the API server to reduce load).
- [ ] **PII Redaction Toggle:**
    - Add a checkbox state isRedactionMode.
    - If True, trigger a Lambda function on upload to run AWS Comprehend Medical, identify Names/DOBs, and draw black boxes over those coordinates.

2.3 Feature: The "Split-View" Workspace

- [ ] **PDF Rendering:**
    - Implement react-pdf.
    - **Security measure:** Render pages as <canvas> layers, not selectable DOM elements, to prevent browser-extension scraping.
- [ ] **AI Chat Interface (RAG):**
    - Build the Chat UI (User Bubble right, AI Bubble left).
    - Implement **Streaming Response**: Use EventSource to display the AI's answer character-by-character as it generates.
    - **Citation Layer:** When the backend returns a citation {page: 4, bbox: [x,y,w,h]}, render a transparent yellow div overlay on the PDF canvas at those coordinates.

2.4 Backend: RAG Pipeline Implementation

- [ ] **Ingestion Worker (Background Task):**
    - Trigger: S3 ObjectCreated event.
    - Process: AWS Textract extracts text -> LangChain splits text into 1000-token chunks.
    - Embedding: Send chunks to **AWS Titan Embeddings G1**.
    - Storage: INSERT INTO document_vectors (embedding, text, page) VALUES (...).
- [ ] **Retrieval Endpoint:**
    - Receive user query -> Embed query -> Run Cosine Similarity search on pgvector -> Send top 5 chunks + Query to **Claude 3.5 Sonnet**.

Phase 3: The "Listener" Module (Live Audio)

Goal: Build the Ambient Clinical Intelligence Engine (Module B).

3.1 Frontend: High-Fidelity Audio Recorder

- [ ] **Browser Media API:**
    - Implement ExtendableMediaRecorder (Standard MediaRecorder is buggy on some browsers).

    - Config: Codec audio/wav, Sample Rate 16000Hz (Required by AWS Transcribe).
- [ ] **Visualizer:**
    - Connect the AudioContext analyzer node to a <canvas> to draw the real-time waveform.
- [ ] **Chunked Upload Strategy:**
    - **Do not** record the whole hour in RAM.
    - Logic: Every 30 seconds, slice the Blob, send it to backend via WebSocket or Multipart POST. This prevents data loss if the browser crashes.

3.2 Backend: Audio Processing Pipeline

- [ ] **Transcription Service:**
    - Integrate **AWS Transcribe Medical**.
    - Enable Speaker Diarization (identifying "Clinician" vs "Patient").
- [ ] **SOAP Generation Service:**
    - Input: Raw Transcript.
    - LLM Process: Send to Bedrock (Claude 3.5) with a specific System Prompt: *"You are a scribe. Categorize this text into Subjective, Objective, Assessment, Plan."*
    - Output: Return structured JSON, not just a string.

3.3 Feature: Post-Visit Editor

- [ ] **Rich Text Editors:**
    - Implement 4 separate TipTap or Quill editors for S, O, A, P sections.
- [ ] **Tagging UI:**
    - Build the TagInput component.
    - Backend logic: Run a lightweight NLP classifier on the transcript to output suggested tags (["Anxious", "Smoker"]).

4.0 Phase 4: Business Logic & Integrations

Goal: Monetize the platform and handle tenant isolation.

4.1 Stripe Subscription Gating

- [ ] **Stripe Setup:**
    - Create 2 Products in Stripe Dashboard: "Standard Tier" and "Premium Tier".
    - Setup **Webhooks**: Create an endpoint /api/webhooks/stripe to listen for invoice.payment_succeeded and customer.subscription.deleted.
- [ ] **Permission Guard:**
    - Create a React HOC (Higher Order Component) <RequireTier tier="premium">.
    - Wrap the "Microphone Button" in this HOC. If the user is on Standard, show a "Upgrade to Unlock" modal instead of recording.

4.2 Tenant Isolation Logic

- [ ] **Row-Level Security (RLS):**
    - Ensure *every single database query* includes WHERE organization_id = :current_org_id.
    - Write a Pytest unit test that attempts to fetch Org B's patient using Org A's token. It *must* fail.

4.3 Notification System

- [ ] **Twilio Integration:**
    - Setup a verified Sender ID.
    - Create a Template: *"Dr. [Name] has updated your care plan. View securely here: [Link]"*.
    - **Security Note:** Do NOT send clinical details in the SMS. Only send a link to a secure portal view.

5.0 Phase 5: Launch Readiness & Security Audit

Goal: Verify compliance before public traffic.

5.1 Automated Testing

- [ ] **Unit Tests:** pytest for all backend logic (Aim for 80% coverage).
- [ ] **E2E Tests:** Playwright scripts that simulate a full doctor workflow: Login -> Create Patient -> Record -> Save.

5.2 Security Hardening

- [ ] **Penetration Scan:** Run OWASP ZAP against the staging API to find vulnerabilities.
- [ ] **Cloud Security Posture:** Run AWS Security Hub automated checks. Fix any red flags (e.g., "S3 bucket public read").
- [ ] **Audit Log Verification:** Manually perform an action (View Patient) and verify a row appeared in the audit_logs table.

5.3 Deployment Pipeline (CI/CD)

- [ ] **GitHub Actions:**
    - **Build Stage:** Lint code, run tests, build Docker images.
    - **Deploy Stage:** Push images to AWS ECR. Trigger ECS Fargate service update "Rolling Deployment" (Zero downtime).