Core Pathways

### **Services Overview**

### Mediconnect is a mobile-first, online-by-default care platform that meets people where they already are: on Android and on WhatsApp. Starting in Nairobi, Kenya, the app provides a clear, end-to-end path from first symptoms to resolution—on-demand GP care, safe medicines, diagnostics, records, and when needed, coordinated advanced care—without forcing patients to fight the system. Authentication is handled entirely through WhatsApp verification; users who struggle with the app can request a WhatsApp callback to be guided through care.

#### Access & Channels

### Android-first app for Nairobi, Kenya with a simple, modern interface designed to reduce cognitive load and work reliably on common devices and connections.

### WhatsApp-native verification replaces traditional SMS OTP for sign-up, login, and high-signal notifications (consult start, prescription ready, results posted).

### App experiences are online-first. For users who can’t navigate the app, a WhatsApp callback flow connects them to a clinician or support for guided care.

### Languages: English and Swahili for patient-facing journeys and core content.

#### GP Consults & Care Navigation

### On-demand GP consultations with two modes tailored to real-life use:

### Synchronous chat/voice for immediate issues, with clear expectations on session length for throughput and fairness.

### Asynchronous chat for follow-ups and non-urgent issues, with notifications on new messages and clinician responses.

### Triage assist inside the chat helps collect structured symptoms and suggests short, patient-friendly educational videos; low-confidence or risky cases are escalated to a GP.

### The GP is the front door for care: they assess, treat common conditions, and guide next steps (medicines, labs, or a specialist referral).

### Integrated referral pathways to local specialists for complex cases; appointments can be requested in-app with patient context pre-shared (with consent) to reduce repeat questioning and delays.

#### Prescriptions & Pharmacy (Verified)

### Digital prescriptions are issued from the consult and available instantly in-app as a secure artifact with a scannable code.

### Patients can find nearby verified pharmacies through a simple map/list. The pharmacy network is curated and licensed to reduce counterfeit risk.

### At the counter, the verified pharmacy scans the prescription to claim and dispense. Item details are visible; personally identifiable information remains limited by policy.

### Delivery options can be enabled with partner logistics (where available). Patients can choose pickup or delivery, with cash-on-delivery or mobile money options.

### Built-in checks help flag look-alike/sound-alike drug risks and common interactions; the GP can adjust and reissue on the spot.

### Patients can mark issues with a dispense (e.g., suspected counterfeit) for fast support review.

#### Diagnostics & Lab Work

### Patients can book tests referred by a clinician, choosing home sample collection or visits to a partner diagnostic center.

### Live status updates and reminders help patients complete tests on time (booked, on the way, collected, processing, completed).

### Results are delivered digitally to the patient and the ordering clinician; sensitive results can be scoped to the ordering specialist by default, with patient-led sharing to others.

### All reports (PDF/images) are securely stored and retrievable in the patient’s Health Locker.

#### Health Locker (Patient Record)

### A single place in the app for the patient’s key health information: consult history, prescriptions, lab results, allergies, and immunizations.

### Easy export to a sharable PDF for second opinions or cross-border care.

### Patient controls what is shared and with whom; sharing is time-bound and auditable.

#### Education & Self-Care

### Short, clinician-reviewed videos and guides help patients understand symptoms, simple self-care, when to see a doctor, and what to expect next.

### Content is tagged by symptom and surfaced contextually during consults and triage.

### Videos are optimized for low data usage and can be cached for reliable viewing.

#### Specialists & Advanced Care

### Curated local specialist referrals with clear expectations on timelines and costs.

### For cases that require advanced treatment not readily available locally, the platform can facilitate guidance and coordination to appropriate providers abroad. This includes access to verified Indian doctors and hospital partners—subject to Kenyan regulations and the rules of destination countries—so patients can receive high-quality, transparent care when local options are constrained.

#### Payments, Insurance, and Transparency

### Simple, upfront consult pricing with clear receipts. Subscription options can provide predictable costs for families.

### Where insurer collaboration exists, patients can see real-time claim updates and any out-of-pocket expectations before visits.

### For bundled procedures (local or cross-border), milestone-based quotes and itemized statements reduce surprises.

#### Partner & Stakeholder Portals

### Pharmacy portal to securely sign in, scan prescriptions, claim, and manage dispense status.

### Diagnostics portal to receive orders with minimal necessary information, upload results, and notify patients.

### Clinician and admin tools for consult routing, notes, referrals, partner onboarding, and issue resolution—kept simple to minimize training friction.

#### Governance, Privacy, and Safety (Concise)

### Patient data is protected and only shared with consent and purpose. Access is role-based and auditable.

### Consult transcripts, prescriptions, and results are retained in line with clear policies; patients can request exports.

### Design choices prioritize safety: plain language summaries, smart defaults, guardrails for risky medication combinations, and clear alerts for critical results.

#### Launch Scope and Expansion

### Initial launch: Nairobi, Kenya, with focus on urban and peri-urban users who already rely on Android and WhatsApp.

### Expansion to additional Kenyan regions and then to other countries will follow the same model: local clinician networks, verified pharmacies and labs, and content localized to language and context.

#### Operating Guidance (Pre-MVP Build, Brief)

### Online-first delivery and WhatsApp-only verification minimize onboarding friction and reduce SMS costs.

### Keep operations lean: a small core team for app, partner ops, and 24/7 clinician coverage with clear session policies.

### Budget guidance while shipping the first release: keep cloud and monitoring costs tight (target sub-$500/month for core infrastructure, excluding WhatsApp conversation fees), prefer managed services, and add complexity only when usage justifies it.

### Stakeholder authentication (clinicians, pharmacies, labs) will use a separate validation approach—final scheme to be decided with compliance counsel and partner input.

### 

### **Care Pathways — Updated for HaaS Model**

#### **Core stance**

* HaaS delivery, not a marketplace: Patients don’t “shop.” The GP drives the flow and offers concise, curated options when needed.
* Minimal patient steps: AI intake first, GP consult, GP-led referral selection, coordinated booking via WhatsApp/app notifications.
* Single channel mindset: App-first with WhatsApp for auth, coordination, reminders, and video calls.

#### **1) AI Intake Gate (pre-GP)**

* Before connecting to a GP, patient answers a brief adaptive intake (symptoms, onset, severity, key red flags).
* Outputs:
  + Patient-facing: confirmation of what will be shared.
  + GP-facing: “AI Intake Summary” + concise history snapshot (recent consults, active meds/allergies).
* Goal: reduce discovery time; GP starts with context.

#### **2) GP Consult (online; 2 concurrent max per GP)**

* Baseline: 15 minutes, with 12-minute prompt to wrap or escalate.
* WhatsApp Video option: GP can initiate WA video within the consult; event is logged (no media captured).
* If specialist referral is required AND the patient accepts:
  + Consult budget extends by +10 minutes (total up to 25 minutes) to:
    - Explain options (local/online),
    - Present 3–4 specialist options (curated by GP/system),
    - Capture patient’s selection,
    - Trigger booking coordination.

#### **3) Specialist Referral (curated by GP; not patient-driven browsing)**

* Choice model:
  + Local (in-person): GP offers 3 options (availability, location).
  + Online specialist: connect to an available specialist (likely Indian or local; whichever is available per roster).
* Booking and coordination:
  + Appointment coordination is executed by the system/Ops via WhatsApp notifications and in-app notifications.
  + Patient receives confirmed time, channel, and prep instructions.
* Data sharing:
  + With consent, a compact packet (GP summary, relevant uploads) is sent to the specialist.

#### **4) Pharmacy Flow (verified network + controlled fallback)**

* After GP issues Rx, patient sees:
  + 2–3 suggested verified pharmacies (geo-based) and a map of all verified partners.
  + Option A: Select a specific verified pharmacy (preferred). QR is retained. Pharmacy scan claims Rx; med list only is shown.
  + Option B: Walk-in to any verified pharmacy on the map and show QR.
  + Non-onboarded fallback: Patient can download the Rx PDF to use anywhere. Once PDF is downloaded, the app permanently disables QR regeneration for that Rx to prevent duplication.
* Liability notice:
  + When choosing PDF/non-verified route, patient sees and accepts a disclaimer that counterfeit risks outside the verified network are not on Mediconnect.

#### **5) Diagnostics (order → result)**

* Specialist orders tests.
* Lab admin sees minimum PII: Patient name, Order ID, Test list, Masked phone (per your decision).
* Results become visible to the patient and the ordering specialist by default; GP access remains policy-controlled.

#### **6) Notifications and reminders (WhatsApp + app)**

* Events: consult assignment, referral choices/confirmations, appointment times, Rx issued, pharmacy reminder, diagnostics order/result, escalations.
* Language: English + Swahili; opt-out respected.

#### **7) SLAs and capacity (aligned to HaaS)**

* GP concurrency: 2 active online consults per GP.
* Online consult cap: 15 minutes; 25 minutes if referral is required and accepted (only in that scenario).
* Offline Q&A: 3–6 hour SLA to first clinician response (unchanged).
* Pharmacy: 90% of partner claims completed within 30 minutes during working hours.
* Diagnostics: SLA displayed per partner; delays notify patient via WhatsApp.

### **Policies and PII Decisions (applied)**

* Diagnostics PII to lab: Patient name + Order ID + Test list + Masked phone.
* Pharmacy PII:
  + Pharmacist UI: No patient PII (med list only), even when the patient pre-selects a specific pharmacy.
  + System record: The selection and claim are recorded and linked to the pharmacy’s ledger/anonymized token for reconciliation and audit. PII remains hidden in pharmacy UI; only Ops (with purpose and audit) can access identifiable data if needed.

### **Architecture/Schema Deltas (minimal, testable)**

* ai\_intake(id, consult\_id, summary\_json, red\_flags, created\_at)
* consult fields:
  + consult.type ENUM['online','offline']
  + consult.capacity\_cap\_minutes DEFAULT 15
  + consult.referral\_extension\_applied BOOL DEFAULT false
* referral(id, consult\_id, kind ENUM['local','online'], options\_json, selected\_specialist\_id, status ENUM['proposed','booked','completed','cancelled'])
* appointment(id, referral\_id, provider\_id, channel ENUM['in\_person','online\_wa','online\_app'], scheduled\_at, confirmed\_at, notifier ENUM['wa','app'], status)
* video\_call\_event(id, consult\_id, started\_at, ended\_at, channel='whatsapp', metadata)
* prescription fields:
  + prescription.qr\_enabled BOOL DEFAULT true
  + prescription.pdf\_downloaded\_at TIMESTAMP NULL (if set → qr\_enabled=false)
  + prescription.non\_verified\_disclaimer\_accepted\_at TIMESTAMP NULL
* pharmacy\_selection(id, prescription\_id, pharmacy\_id NULL, selection\_source ENUM['suggested','map\_walkin','none'], selected\_at)
* claim ledger:
  + pharmacy\_claim adds external\_ref\_token (anonymized) for reconciliation without exposing PII to pharmacy UIs.

RLS and audit notes

* Keep pharmacist views item-only; block PII columns via RLS.
* Log every state change (intake submitted, WA video start/stop, referral selection, PDF download disabling QR, claim attempts).

### **Operational Notes**

* HaaS execution:
  + GP orchestrates choices; patient confirms from curated options only.
  + Ops ensures rosters for online specialists and local partners; booking is system/Ops-led, never free-form by the patient.
* Reduced geography emphasis:
  + Document is phrased for pilot(s) with Kenya prioritized but without repeatedly naming Nairobi; per your guidance we keep it generic while honoring current constraints.

operating model final

### **Operating Model — MVP (Healthcare-as-a-Service)**

#### **Core stance**

* Service-led, not a marketplace: We deliver care through our own network (GPs, specialists, pharmacies, diagnostics) with minimal patient actions.
* MVP channels:
  + Synchronous consults: WhatsApp video calls (primary).
  + Intake, records, Rx, and notifications: App + WhatsApp notifications.
  + In-app chat and in-app video are deferred to a future phase.
* Patient flow: AI intake → GP consult → GP-curated referral (if needed) → Pharmacy/Diagnostics → Follow-ups.

#### **Roles and responsibilities**

* Patient: Completes brief AI intake; attends GP consult (WhatsApp video); reviews Rx; visits verified pharmacies; completes diagnostics as ordered.
* GP: Handles consults; issues prescriptions; curates specialist options (local or online); extends session for referral discussion when accepted.
* Specialist (local or online): Provides consult via in-person or WhatsApp video; reviews GP packet; returns plan/recommendations.
* Pharmacy admin (verified): Scans Rx QR; views med list only; dispenses and updates claim status.
* Diagnostics admin: Fulfills orders with minimal PII; uploads results.
* Ops admin: Manages rosters, partner onboarding, booking coordination, escalations; no PHI by default (metadata only).
* Support: Time-boxed, masked impersonation for troubleshooting; handles escalations and notifications.

#### **Channels and touchpoints (MVP)**

* AI Intake (in-app): Mandatory short adaptive questionnaire; generates “AI Intake Summary” for GP context.
* GP Consult (synchronous): WhatsApp video call initiated and logged by the system/GP; no media capture.
* Asynchronous: Offline questions remain supported in-app; patient is notified via WhatsApp.
* Notifications: All key events mirrored via WhatsApp + in-app (where applicable); English + Swahili.

#### **Care pathways**

* AI Intake Gate
  + Patient completes a brief, adaptive intake before any GP connection.
  + Outputs shared: Patient gets a summary preview; GP gets an “AI Intake Summary” and recent history snapshot.
* GP Consult (WhatsApp video, 2-session capacity model)
  + Baseline consult: 15 minutes; 12-minute prompt to wrap or escalate.
  + Capacity: Each GP may hold up to 2 active sessions; only one can be in live video at any time; the other slot is for pre/post-video tasks or asynchronous follow-ups.
  + If referral is required and accepted: Session extends by +10 minutes (up to 25 minutes total) to present curated options and confirm selection.
* Specialist Referral (curated; GP-driven)
  + Choices presented by GP (no free browsing by patient):
    - Local (in-person): 3 vetted options based on availability/location.
    - Online specialist: Connected via WhatsApp video to the available roster (may be Indian or local).
  + Booking: System/Ops coordinates; confirmations and instructions via WhatsApp + in-app notifications.
  + Data sharing: With consent, a compact clinical packet (GP summary + relevant attachments) is shared.
* Pharmacy Flow (verified-first with controlled fallback)
  + After Rx issuance:
    - Suggested: 2–3 nearby verified pharmacies; plus a map of all verified partners.
    - Option A (preferred): Select a specific verified pharmacy; retain QR; pharmacist scans and claims; med list only is shown.
    - Option B: Walk-in to any verified partner and present QR.
    - Fallback (non-verified): Patient may download the PDF to use anywhere. Once the PDF is downloaded, QR regeneration is permanently disabled for that Rx to prevent duplication.
  + Disclaimer: If using a non-verified route, the patient accepts a liability notice regarding counterfeit risk outside the verified network.
* Diagnostics (order → result)
  + Specialist orders tests; diagnostics admin sees minimum PII to fulfill.
  + Results are visible to the patient and the ordering specialist by default; GP visibility is policy-controlled.
  + Delays are communicated via WhatsApp notifications.

#### **SLAs and capacity**

* GP consults (WhatsApp video): 15-minute cap; up to 25 minutes when referral is accepted.
* GP concurrency: 2 active sessions per GP; at most one simultaneous live video; other slot used for pre/post work or async updates.
* Offline Q&A: First clinician response within 3–6 hours.
* Pharmacy claims: 90% of verified partner claims completed within 30 minutes during working hours.
* Diagnostics: Partner-published turnaround times; patient notified on delays.

#### **Authentication and communications**

* Auth: WhatsApp verification only (authentication templates). OTP validation with TTL and attempt limits; all attempts audited.
* Notifications: Consult assignment, referral choices/confirmations, appointment times, Rx issued, pharmacy reminders, diagnostics orders/results, and escalations via WhatsApp + in-app.

#### **Data handling, privacy, and compliance**

* Diagnostics PII to lab (confirmed): Patient name + Order ID + Test list + masked phone.
* Pharmacy PII (confirmed):
  + Pharmacist UI: No patient PII; medication items only.
  + System: If a pharmacy is selected or claims via QR, the event is recorded and linked via anonymized references for reconciliation and audit. Ops may access PII for dispute resolution with purpose-based audit.
* Security: Strict RLS on PHI, masked support views, reason prompts, and 24-month audit retention.
* Retention (unchanged): Chat/media hot for 90 days then archive; prescriptions hot for 24 months then archive pointers; on-demand restore with audit.
* Geography emphasis: Document kept pilot-oriented without over-highlighting a single city; policies are applicable to initial markets with Kenya prioritized.

#### **Observability and incident response**

* Metrics: Video session success rate, first-response times, SLA hit rate (offline), OTP success (WhatsApp), Rx claim success, archive restore success, audit completeness.
* Runbooks:
  + GP coverage gaps → reassign/queue and notify.
  + Pharmacy downtime → suggest nearby verified alternatives; notify patient.
  + WhatsApp disruption → fallback to in-app notifications and Ops outreach; post-incident review.

Confirmed decisions captured:

* MVP synchronous consults default to WhatsApp video (GP and online specialists).
* Diagnostics PII minimal set: Name + Order ID + Test list + masked phone.
* Pharmacist UI: No patient PII; system records anonymized settlement data; Ops-only PII access under audit.
* Referral-driven extension: +10 minutes when the patient accepts a specialist referral (total up to 25 minutes).

flows verbal and data/rla

### **Algorithms & Sequence Flows — Verbal Model (MVP, Healthcare‑as‑a‑Service) New:-**

### **Global guardrails**

### Channels: WhatsApp verification; WhatsApp video for synchronous consults in MVP; in‑app text chat is enabled (AI intake + GP/patient thread in one place). Other video providers allowed per specialist comfort (Ops will arrange and record metadata).

### Intake first: Every consult begins with the short AI intake; the AI summary is injected into the consult’s persistent message thread as a system message.

### Capacity & timers: GP can hold 2 active sessions; baseline 15 minutes; extend to 25 minutes only when a specialist referral is required and accepted.

### Privacy: Minimum PII to labs; no PII in pharmacy UI; all privileged access audited.

### WhatsApp verification (unchanged)

### Generate OTP → send via WhatsApp template → verify with TTL/attempt caps → create session → audit outcome.

### AI intake gate + thread initialization

### Patient taps “Talk to a GP.”

### Adaptive questions capture symptoms/onset/severity/red flags.

### System:

### Creates or reuses a consult record (mode=online if going live; offline otherwise).

### Writes an AI Intake Summary into ai\_intake AND posts a system message into the consult’s message thread (so the GP and patient see the intake context in-line).

### If red flags, prioritize routing; else continue.

### GP assignment + live session start (WA video) + in‑app chat

### Scheduler assigns a GP whose active\_sessions < 2 and with ≤1 live video currently.

### Create/confirm consult(mode='online', capacity\_cap\_minutes=15).

### Start WhatsApp video (or other provider if required by specialist later; for GP it’s WA video) and log a video\_call\_event.

### During/around the call:

### GP can use the in‑app chat thread for pre/post notes, links, and clarifications; patient can respond in the same thread.

### 12‑minute prompt warns of time; auto‑wrap at 15 unless moving to specialist referral.

### Specialist referral (curated; GP‑driven; booking via Ops/automation)

### Trigger: GP determines escalation; explains and patient accepts.

### Extend consult (+10 minutes, total up to 25).

### Options the GP presents (patient does not browse):

### Local (in‑person): 3 vetted specialists.

### Online: connect to the available specialist roster (may be Indian or local).

### Capture patient selection in‑thread (structured quick‑reply + confirmation message).

### Create referral and appointment (status 'booked' or 'pending'), and post a summary message in the thread.

### Appointment channel:

### Default: WhatsApp video.

### Alternate: Other video provider per specialist preference (Ops arranges; details posted to the thread and via WhatsApp). In‑app text chat is also available for pre/post consult communications.

### Consent: With patient consent, send the compact clinical packet to the specialist; confirm via message in the thread.

### Specialist consultation flow (new)

### Pre‑consult:

### Specialist reviews the AI summary, GP notes, attachments in their portal.

### Specialist may order pre‑appointment diagnostics (lab\_order tied to referral) based on the in‑app chat or pre‑consult review; patient notified in thread + WhatsApp.

### During consult:

### If online: WhatsApp video (or alternate provider) is used; operational metadata is logged. Clarifications and follow‑ups happen in the in‑app consult thread.

### If local: Details (time, location) appear in thread; any updates are mirrored via WhatsApp.

### Post‑consult:

### Specialist records recommendations in portal; a summary is posted to the thread (patient‑readable).

### If a procedure is required, proceed to §5.

### Procedure decisioning (inpatient/outpatient; local vs foreign) — specialist portal drives, patient confirms in thread

### Specialist (working via the specialist/admin portal) proposes procedure options:

### Attributes: inpatient/outpatient, local/foreign, facility, estimated costs, prep, timeline.

### System posts the options in the patient’s consult thread with concise cards and a selection action.

### Patient selects the preferred option (local or foreign). Ops coordinates booking/logistics; confirmations and instructions are posted to the thread and via WhatsApp.

### All steps are auditable; any docs (invitation letters, prep guides) are attached and linked in thread.

### Prescription + pharmacy flow (verified-first; controlled fallback)

### GP issues prescription at consult end.

### System posts an Rx summary card in the thread with:

### Suggested 2–3 nearby verified pharmacies and a “View verified map” link.

### “Select pharmacy” action (preferred) or “Use QR at any verified pharmacy.”

### “Download PDF for non‑verified pharmacy” fallback with disclaimer.

### Behaviors:

### If a verified pharmacy is selected, QR remains enabled; first verified claim wins; pharmacist sees med items only (no PII).

### If patient downloads PDF, QR is permanently disabled for that Rx; disclaimer acceptance is logged; responsibility note displayed.

### Diagnostics (order → result) — specialist‑owned

### Specialist raises lab\_order (may be pre‑appointment or during/after consult).

### Lab admin sees minimum PII: patient name, order ID, test list, masked phone.

### When results are uploaded, the system:

### Notifies the patient (thread + WhatsApp).

### Grants view to patient and ordering specialist by default; GP visibility per policy.

### The thread receives a result notification message with a secure link.

### Offline Q&A (async care)

### Patient can post questions in the same consult thread (or open a new offline consult thread).

### SLA: 3–6 hours to first clinician response.

### Scheduler assigns; GP replies in the thread; status changes reflected as system messages.

### Notifications (mirrored)

### All key events (auth, assignment, call start, referral, appointment, Rx, claims, diagnostics, procedures) are posted in the thread and mirrored to WhatsApp where relevant. Provider message IDs are tracked for delivery status and audit.

### **OLD:-**

#### **Global guardrails**

* Channel: WhatsApp video for synchronous consults; WhatsApp + in‑app notifications. In‑app chat/video deferred.
* Intake first: Every live consult begins with a short AI intake that generates a GP‑ready summary.
* Capacity: Each GP has up to 2 active sessions; only 1 can be a live video at a time.
* Timers: Online consults are 15 minutes; extend to 25 minutes only if a specialist referral is required and accepted during the session.
* Privacy: Minimum PII to labs; zero PII to pharmacies (UI). All access audited.

#### **0) WhatsApp verification (auth)**

* Trigger: User enters phone; requests sign‑in.
* Steps:
  1. Backend creates a 6‑digit code, stores hash with TTL (e.g., 5 minutes), max attempts (e.g., 5).
  2. Send via WhatsApp auth template; store provider message\_id.
  3. User submits code; backend verifies hash+TTL+attempts; on success, creates a session tied to user.id; audit the outcome.
  4. Lock/rotate on excessive failures; expose a “try again later” schedule.
* Result: Authenticated session; OTP artifacts retained for audit, not for reuse.

#### **1) AI intake gate**

* Trigger: User taps “Talk to a GP.”
* Steps:
  1. Adaptive questions (symptoms, onset, severity, red flags).
  2. Generate an “AI Intake Summary” (structured JSON, red\_flags boolean/list).
  3. If red flags → prioritize routing; otherwise continue normal flow.
* Result: Intake summary attached to a pending consult request; visible to GP at assignment.

#### **2) GP assignment and live session (WhatsApp video)**

* Trigger: Intake completed; user requests live consult.
* Steps:
  1. Capacity check: Assign a GP whose active\_sessions < 2 and not already in 1 live video.
  2. Create consult(mode='online', capacity\_cap\_minutes=15).
  3. Start WhatsApp video invite; log start/stop times as call events.
  4. 12‑minute prompt to GP to wrap or consider referral; auto‑wrap at 15 unless referral accepted (see §3).
  5. GP composes clinical notes and issues next steps (Rx, advice, referral).
* Result: Consult closed with structured summary; follow‑ups queued as necessary.

Fallback

* If no GP is available immediately: create offline consult ticket (mode='offline', SLA=3–6h) and notify user via WhatsApp.

#### **3) Specialist referral (curated; GP‑driven)**

* Trigger: GP determines referral is needed, presents options; user accepts.
* Steps:
  1. Extend the same consult’s timebox to 25 minutes (referral\_extension\_applied=true).
  2. GP offers curated choices:
     + Local in‑person: 3 vetted options.
     + Online specialist: the available roster (may be Indian or local).
  3. Capture user’s selection; create referral record and an appointment (booked or pending scheduling).
  4. Share a compact packet (GP summary + relevant attachments) with consent.
  5. Send confirmations and instructions via WhatsApp + in‑app; handle rescheduling via Ops if needed.
* Result: Appointment scheduled; patient receives details; consult wraps with clear next steps.

#### **4) Prescription and pharmacy flow**

* Trigger: GP prescribes at consult end.
* Steps:
  1. Create prescription + items; generate QR token (server‑side, store hash).
  2. Suggest 2–3 verified pharmacies near the patient; also show map of verified partners.
  3. User chooses one:
     + Preferred: Select a verified pharmacy (pharmacy\_selection). QR remains enabled; pharmacy scans and claims; pharmacy UI shows med list only.
     + Or: Walk into any verified partner and show QR; first verified claim wins.
     + Fallback: Download Rx PDF to use anywhere. On first PDF download, permanently disable QR (qr\_enabled=false); capture liability disclaimer acceptance.
  4. Verified pharmacy claim:
     + Staff logs into scanner; POST token to claim.
     + If unclaimed and QR enabled, link claim to that pharmacy and reveal item‑only med list; otherwise reject.
* Result: Dispensing captured via claim ledger; patient sees claim status; zero PII in pharmacy UI; disputes resolvable via Ops under audit.

#### **5) Diagnostics (order → result)**

* Trigger: Specialist orders tests.
* Steps:
  1. Create lab\_order with tests; expose to lab admin with minimum PII: patient name, order ID, test list, masked phone.
  2. Lab uploads result file(s); result becomes visible to the patient and ordering specialist by default.
  3. Notify patient via WhatsApp; GP visibility is governed by policy (off by default unless shared).
* Result: Results delivered; downstream care continues per specialist advice.

#### **6) Offline Q&A (async)**

* Trigger: User submits a question instead of a live consult.
* Steps:
  1. Create offline ticket with SLA 3–6 hours to first response.
  2. Scheduler assigns when GP active\_sessions < 2 and not in call.
  3. GP responds; thread closes after resolution/timeout.
* Result: Async care honored with SLA; notifications sent at updates.

#### **7) Notifications (WA + in‑app)**

* Event‑driven sends for: auth codes, consult assignment, video start link, referral options/confirmation, appointment details, Rx issued, pharmacy reminders, diagnostics orders/results, delay notices, and escalations.
* Message states are tracked (queued, sent, delivered, failed) with provider message IDs.

### **Data Model — MVP (key tables and fields)**

Users and roles

* user(id, phone, role ENUM['patient','gp','specialist','pharmacist','support','ops\_admin','super\_admin'], locale, created\_at)
* clinician(id PK FK user.id, type ENUM['GP','Specialist'], facility\_id NULL, active BOOL, approved\_by, approved\_at)
* facility(id, name, type ENUM['hospital','lab','pharmacy'], region, active)

Authentication and notifications

* wa\_auth\_code(id, phone, code\_hash, ttl\_expires\_at, attempts, status ENUM['pending','verified','expired','locked'], wa\_message\_id, created\_at, verified\_at)
* notification(id, user\_id, channel ENUM['wa','app'], template\_key, payload\_json, provider\_message\_id, status ENUM['queued','sent','delivered','failed'], created\_at, sent\_at, delivered\_at, failed\_at)

Consults and intake

* consult(id, patient\_id, assigned\_clinician\_id, mode ENUM['online','offline'], status ENUM['open','closed','cancelled'], started\_at, ended\_at, capacity\_cap\_minutes DEFAULT 15, referral\_extension\_applied BOOL DEFAULT false)
* ai\_intake(id, consult\_id, summary\_json, red\_flags BOOL, created\_at)

Live session telemetry

* video\_call\_event(id, consult\_id, channel ENUM['whatsapp'], started\_at, ended\_at, metadata\_json)

Referrals and appointments

* referral(id, consult\_id, kind ENUM['local','online'], options\_json, selected\_specialist\_id NULL, status ENUM['proposed','booked','completed','cancelled'], created\_at, updated\_at)
* appointment(id, referral\_id, provider\_id, channel ENUM['in\_person','online\_wa'], scheduled\_at, confirmed\_at, notifier ENUM['wa','app'], status ENUM['scheduled','completed','no\_show','cancelled'])

Prescriptions and pharmacy

* prescription(id, consult\_id, prescriber\_id, pdf\_url, qr\_token\_hash, qr\_enabled BOOL DEFAULT true, pdf\_downloaded\_at TIMESTAMP NULL, non\_verified\_disclaimer\_accepted\_at TIMESTAMP NULL, status ENUM['new','claimed','dispensed','expired'], expires\_at)
* prescription\_item(id, prescription\_id, drug\_name, strength, form, quantity, instructions)
* pharmacy(id, name, onboarded BOOL, username, password\_hash, location\_geo, active)
* pharmacy\_selection(id, prescription\_id, pharmacy\_id NULL, selection\_source ENUM['suggested','map\_walkin','none'], selected\_at)
* pharmacy\_claim(id, prescription\_id, pharmacy\_id, claimed\_at, status ENUM['ready','dispensed'], external\_ref\_token, notes)

Diagnostics

* lab\_order(id, patient\_id, specialist\_id, lab\_id, tests\_json, status ENUM['ordered','uploaded','viewed'], created\_at)
* lab\_result(id, lab\_order\_id, file\_url, uploaded\_by, uploaded\_at, viewed\_by\_patient\_at, viewed\_by\_specialist\_at)

Content and records

* attachment(id, entity\_type ENUM['consult','lab\_result','referral'], entity\_id, file\_url, mime, size)
* audit\_log(id, actor\_id, role, table\_name, row\_id, action ENUM['read','create','update','delete'], purpose, ts, masked\_view BOOL)

Retention/ops (configurable constants can be stored in a config table if needed)

* config(key, value\_json) for retention and feature flags.

Note: “message” table is optional for offline Q&A if you keep threaded replies; if so:

* message(id, consult\_id, sender\_id, kind ENUM['text','image','audio'], content, created\_at)

### **RLS and Access Control — Policy Summary**

General principles

* Default deny; allow by role and relationship to the row.
* Column‑level protection for PII where needed (e.g., pharmacy views).
* All privileged reads/writes are audited with purpose; support has time‑boxed masked views.

Users

* user: a user can read/update only their own profile. Admin roles can list minimal fields for ops (e.g., phone, role) when necessary.

Consults and intake

* consult:
  + Patient can read where patient\_id = auth.uid().
  + Assigned clinician (GP or specialist) can read when assigned\_clinician\_id = auth.uid().
  + Ops/Admin: metadata‑only view (no conversation content) via restricted view.
* ai\_intake:
  + Same visibility as consult; support sees masked summary (no raw free‑text if present) under time‑boxed impersonation.

Live session events

* video\_call\_event:
  + Visible to patient and assigned clinician; Ops/Admin metadata‑only view (timestamps, not content).

Referrals and appointments

* referral:
  + Patient and assigned GP can read full; selected specialist can read once status >= 'booked'.
  + Ops/Admin: metadata‑only view (status, timestamps, provider\_id) for coordination; no clinical notes by default.
* appointment:
  + Patient and the involved provider (specialist) can read details.
  + Ops/Admin: read scheduling metadata; PII minimized.

Prescriptions and pharmacy

* prescription:
  + Patient and prescriber can read full.
  + Once a claim exists, the claimed pharmacy gets item‑only read via a restricted projection/view; all PII fields (name, phone, address if any) are excluded.
  + Ops/Admin: can read metadata (status, timestamps); PII requires elevated reason‑prompt and audit.
* prescription\_item:
  + Patient, prescriber: full.
  + Claimed pharmacy: item‑only view joined by claim, no patient PII.
* pharmacy\_selection:
  + Patient: can read their own selections.
  + Pharmacy: no direct access.
  + Ops/Admin: metadata for coordination and reconciliation.
* pharmacy\_claim:
  + Claimed pharmacy: can read its own claims, item‑only join; no patient PII.
  + Patient: may see status of their prescription claim(s).
  + Ops/Admin: full claim metadata; PII only via reason‑prompted audited view.

Diagnostics

* lab\_order:
  + Patient and ordering specialist: full.
  + Lab admin: minimum PII view (patient name, order ID, test list, masked phone) where lab\_id matches.
  + GP (referring) has no access by default unless explicitly shared.
* lab\_result:
  + Patient and ordering specialist: full result access.
  + Lab admin: can read only what they uploaded for operational checks (no additional PHI).
  + Ops/Admin: metadata only.

Notifications and auth

* wa\_auth\_code:
  + System‑only; no end‑user access.
* notification:
  + A user can read their own notifications.
  + Ops/Admin: can read metadata and delivery states for incident resolution.

Audit

* audit\_log:
  + Super Admin full access.
  + Ops/Admin can query aggregated views; row‑level details require elevated reason and are audited.

Support impersonation

* support:
  + Must open a time‑boxed session (e.g., 15 minutes) with purpose string; PII masked by default; unmasking requires explicit reason and is fully audited.

## Changes to data and rla

Key changes highlighted:

* In‑app chat is now first‑class. The AI intake summary is also written into the thread as a system message for continuity.
* Specialist consultation lifecycle includes pre‑appointment diagnostics, procedure proposals (inpatient/outpatient; local/foreign), and patient selection captured in‑thread.
* Appointment channels include WhatsApp video (default) and alternate providers; details recorded as metadata and posted to the thread.

Core entities (selected)

Users and roles

* user(id, phone, role ENUM['patient','gp','specialist','pharmacist','support','ops\_admin','super\_admin'], locale, created\_at)
* clinician(id PK FK user.id, type ENUM['GP','Specialist'], facility\_id NULL, active, approved\_by, approved\_at)
* facility(id, name, type ENUM['hospital','lab','pharmacy'], region, active)

Auth and notifications

* wa\_auth\_code(id, phone, code\_hash, ttl\_expires\_at, attempts, status ENUM['pending','verified','expired','locked'], wa\_message\_id, created\_at, verified\_at)
* notification(id, user\_id, channel ENUM['wa','app'], template\_key, payload\_json, provider\_message\_id, status ENUM['queued','sent','delivered','failed'], created\_at, sent\_at, delivered\_at, failed\_at)

Consults, intake, and chat

* consult(id, patient\_id, assigned\_clinician\_id, mode ENUM['online','offline'], status ENUM['open','closed','cancelled'], started\_at, ended\_at, capacity\_cap\_minutes DEFAULT 15, referral\_extension\_applied BOOL DEFAULT false)
* ai\_intake(id, consult\_id, summary\_json, red\_flags BOOL, created\_at)
* message(  
   id, consult\_id, sender\_id, sender\_role ENUM['patient','gp','specialist','support','system'],  
   kind ENUM['text','image','audio','system','link','ai\_summary'],  
   content TEXT NULL, attachment\_url NULL, created\_at  
   )
  + Note: When AI intake completes, insert a message(kind='ai\_summary', sender\_role='system') with a concise generated summary.

Live session telemetry

* video\_call\_event(id, consult\_id, channel ENUM['whatsapp','other'], provider\_name NULL, started\_at, ended\_at, metadata\_json)

Referrals, appointments, and procedures

* referral(id, consult\_id, kind ENUM['local','online'], options\_json, selected\_specialist\_id NULL, status ENUM['proposed','booked','completed','cancelled'], created\_at, updated\_at)
* appointment(id, referral\_id, provider\_id, channel ENUM['in\_person','online\_wa','online\_other'], other\_provider\_name NULL, scheduled\_at, confirmed\_at, notifier ENUM['wa','app'], status ENUM['scheduled','completed','no\_show','cancelled'])
* procedure\_option(  
   id, referral\_id, type ENUM['inpatient','outpatient'], location ENUM['local','foreign'],  
   facility\_id NULL, estimate\_json, prep\_json, supporting\_docs\_url NULL, status ENUM['proposed','selected','booked','completed','cancelled'], created\_at, updated\_at  
   )
* procedure\_selection(id, referral\_id, selected\_option\_id, selected\_at, consent\_artifact\_url NULL)

Prescriptions and pharmacy

* prescription(id, consult\_id, prescriber\_id, pdf\_url, qr\_token\_hash, qr\_enabled BOOL DEFAULT true, pdf\_downloaded\_at TIMESTAMP NULL, non\_verified\_disclaimer\_accepted\_at TIMESTAMP NULL, status ENUM['new','claimed','dispensed','expired'], expires\_at)
* prescription\_item(id, prescription\_id, drug\_name, strength, form, quantity, instructions)
* pharmacy(id, name, onboarded BOOL, username, password\_hash, location\_geo, active)
* pharmacy\_selection(id, prescription\_id, pharmacy\_id NULL, selection\_source ENUM['suggested','map\_walkin','none'], selected\_at)
* pharmacy\_claim(id, prescription\_id, pharmacy\_id, claimed\_at, status ENUM['ready','dispensed'], external\_ref\_token, notes)

Diagnostics

* lab\_order(id, patient\_id, specialist\_id, lab\_id, referral\_id NULL, tests\_json, status ENUM['ordered','uploaded','viewed'], created\_at)
* lab\_result(id, lab\_order\_id, file\_url, uploaded\_by, uploaded\_at, viewed\_by\_patient\_at, viewed\_by\_specialist\_at)

Attachments and audit

* attachment(id, entity\_type ENUM['consult','message','lab\_result','referral','procedure\_option'], entity\_id, file\_url, mime, size)
* audit\_log(id, actor\_id, role, table\_name, row\_id, action ENUM['read','create','update','delete'], purpose, ts, masked\_view BOOL)

Config/flags

* config(key, value\_json) for retention, feature flags (e.g., enable\_online\_other\_video=true).

RLS policies (summary)

Users

* user: self‑read/update only; ops/super\_admin minimal list views where required (no PHI by default).

Consults and intake

* consult:
  + Patient: read where patient\_id = auth.uid().
  + Assigned GP/Specialist: read where assigned\_clinician\_id = auth.uid() OR linked via referral/appointment to the consult.
  + Ops/Admin: metadata‑only via restricted view (no message content).
* ai\_intake:
  + Same as consult; support sees masked fields under time‑boxed impersonation.

Messages (in‑app chat)

* message:
  + Patient: read/write on own consult threads (write limited to allowed kinds).
  + Assigned GP/Specialist: read/write where consult\_id is assigned or linked via referral/appointment.
  + Support: read masked content during time‑boxed impersonation; unmask requires reason prompt; all actions audited.
  + Ops/Admin: no message content by default; may read metadata (timestamps, sender\_role, kind) via restricted view.

Live session events

* video\_call\_event:
  + Patient + assigned clinician(s): read.
  + Ops/Admin: metadata‑only view (no PII beyond timestamps/provider names).

Referrals, appointments, procedures

* referral:
  + Patient + assigned GP: read; selected specialist gains read when status >= 'booked'.
  + Ops/Admin: read scheduling metadata; clinical notes excluded by default.
* appointment:
  + Patient + provider: read details.
  + Ops/Admin: read scheduling metadata; no additional PHI.
* procedure\_option / procedure\_selection:
  + Patient: read proposed options and own selection.
  + Specialist/GP: read for their cases.
  + Ops/Admin: coordination metadata; any PII/clinical detail access requires elevated reason and audit.

Prescriptions and pharmacy

* prescription/prescription\_item:
  + Patient + prescriber: full read.
  + Claimed pharmacy: item‑only projection where a claim exists and matches pharmacy\_id; no patient PII.
  + Ops/Admin: metadata; PII only via reason‑prompted audited access.
* pharmacy\_selection:
  + Patient: read own selection(s).
  + Pharmacy: no access.
  + Ops/Admin: metadata for reconciliation; no PII exposed to pharmacies.
* pharmacy\_claim:
  + Pharmacy: read own claims (with item‑only join); no patient PII.
  + Patient: see status of own prescription claim(s).
  + Ops/Admin: full metadata; PII restricted to audited access.

Diagnostics

* lab\_order:
  + Patient + ordering specialist: full read.
  + Lab admin: minimum PII view where lab\_id matches (name, order ID, test list, masked phone).
  + GP: no read by default unless explicitly shared.
* lab\_result:
  + Patient + ordering specialist: full read.
  + Lab admin: read own uploads for operational checks only.
  + Ops/Admin: metadata.

Support impersonation and audit

* Support must open a time‑boxed (e.g., 15‑minute) session with purpose; masked views by default; unmasking triggers reason prompts and audit logs.
* All privileged actions write to audit\_log with purpose and masked\_view flags.

Retention (unchanged)

* Chat/media hot 90 days then archive; prescriptions hot 24 months; on‑demand restore with audit.

MoSCoW

### **MoSCoW — Must Have (Mo) Requirements for MVP**

Notes

* Model: Healthcare-as-a-Service (HaaS). GP orchestrates care; patient actions are minimal.
* Channels: WhatsApp verification and WhatsApp video for synchronous consults; in‑app text chat thread is enabled (AI intake + consult thread). In‑app video is deferred.
* Geography: Pilot-focused without over‑emphasizing a single city; policies tuned for initial markets with Kenya prioritized.

#### **1) WhatsApp‑only Authentication (OTP)**

* Description: Users authenticate via a 6‑digit OTP delivered using WhatsApp Cloud API authentication templates. No SMS.
* Scope:
  + Generate OTP; store hash with TTL (e.g., 5 minutes) and attempt limits (e.g., 5).
  + Deliver via WhatsApp template; validate and issue session.
  + Audit every attempt and outcome (success/failed/expired/locked).
* Acceptance criteria:
  + OTP issuance succeeds within 5 seconds for ≥95% requests.
  + TTL and attempt caps are strictly enforced; lockout behavior documented and tested.
  + Audit log entries include actor, purpose=auth, outcome, timestamps.
* Data/RLS:
  + wa\_auth\_code table; no user‑level read; system‑only.
  + audit\_log entries for attempts and verifications.
* Dependencies:
  + WhatsApp Cloud API (templates pre‑approved).
  + Backend (Edge Functions/API), DB.
* Edge cases:
  + Duplicate request within TTL returns rate‑limited response but does not reset TTL.
  + Delivery failure → user sees retry guidance; no code leakage in UI.
* Out of scope:
  + Email/SMS-based auth; social login.

#### **2) AI Intake Gate + Thread Initialization**

* Description: Every consult starts with a brief adaptive intake. The summary is saved and injected as a system message into the consult’s persistent in‑app message thread.
* Scope:
  + Adaptive Q&A (symptoms, onset, severity, red flags).
  + Create or bind to a consult; write ai\_intake and post message(kind='ai\_summary').
* Acceptance criteria:
  + completion time ≤90 seconds (p50).Intake
  + AI summary visible to patient and assigned GP in the same thread.
  + Red‑flag flagging available to scheduler for priority routing.
* Data/RLS:
  + ai\_intake table; message(kind='ai\_summary') in thread.
  + Visibility mirrors consult access; support masked by default.
* Dependencies:
  + In‑app messaging module, consult scheduler.
* Edge cases:
  + Patient abandons intake → allow resume; stale intakes (>24h) are not auto‑submitted.

#### **3) In‑App Messaging Thread (Patient ↔ GP/Specialist/Support)**

* Description: Persistent, auditable message thread per consult. Used for AI summary, pre/post consult notes, offline Q&A, attachments, and system updates.
* Scope:
  + Text, images, audio, links, system messages (status changes, actions).
  + Patient and assigned clinicians can post; support can assist with masked view.
* Acceptance criteria:
  + Thread loads ≤1.5 seconds (p75) for the last 50 messages.
  + System messages recorded for major events (intake summary, assignment, call start/stop, referral proposal/selection, Rx issued, claim status, diagnostics events).
* Data/RLS:
  + message table with sender\_role; RLS by consult membership; Ops/Admin metadata‑only view.
* Dependencies:
  + Storage for attachments; audit/logging.
* Edge cases:
  + Attachment failures queue and retry; failed sends are visible to the sender.

#### **4) GP Synchronous Consult via WhatsApp Video (+ in‑thread chat)**

* Description: GP conducts live consult over WhatsApp video; pre/post clarifications occur in the in‑app thread. One live video at a time per GP; 2 active sessions overall.
* Scope:
  + Start/stop call; log video\_call\_event with provider=whatsapp.
  + Timers: 15‑minute cap with a 12‑minute prompt.
* Acceptance criteria:
  + Call start link delivered within 5 seconds in ≥95% cases.
  + Auto‑wrap at 15 minutes unless referral accepted (see #5); wrap posts a summary system message.
* Data/RLS:
  + consult.capacities; video\_call\_event rows.
* Dependencies:
  + WhatsApp Cloud API; scheduler; notifications.
* Edge cases:
  + Call drop → GP can re‑initiate within the consult window; events stitched.

#### **5) Specialist Referral (GP‑curated, Not Patient‑Browse) with Time Extension**

* Description: If escalation is needed and the patient accepts, GP presents curated choices and consult time extends by +10 minutes (max 25 minutes).
* Scope:
  + Options: Local in‑person (3 vetted) OR Online specialist (roster; may be local or Indian).
  + Patient selection captured in‑thread; referral and appointment created.
  + Consent capture before sharing the packet.
* Acceptance criteria:
  + Selecting an option posts confirmation to thread and creates referral and appointment records.
  + capacity\_cap\_minutes updated to 25 with referral\_extension\_applied=true.
* Data/RLS:
  + referral, appointment tables; procedure entities if applicable later.
* Dependencies:
  + Ops/automation for booking; notifications.
* Edge cases:
  + Patient declines referral → consult wraps with advice; no extension applied.

#### **6) Specialist Consultation Lifecycle (Online or Local) — Baseline**

* Description: Specialist reviews packet, may order pre‑appointment diagnostics, consults via WhatsApp video (or alternate provider), posts summary, and triggers next steps.
* Scope:
  + Online channel default WhatsApp video; alternate provider supported; details posted to thread.
  + Local visit details posted; changes mirrored via WhatsApp.
* Acceptance criteria:
  + Specialist receives access only after status >=‘booked’.
  + Post‑consult summary available in thread to the patient within 1 hour of consult.
* Data/RLS:
  + referral.status transitions; appointment.channel; message summaries.
* Dependencies:
  + Specialist portal, notifications.
* Edge cases:
  + Reschedule flows update thread and notifications atomically.

#### **7) Procedure Decisioning (Inpatient/Outpatient; Local/Foreign) — Specialist‑Initiated, Patient Confirms**

* Description: Specialist proposes procedure options via portal; system posts cards in thread; patient selects; Ops coordinates.
* Scope:
  + Attributes include type (IP/OP), location (local/foreign), facility, estimate, prep.
  + Selection captured; consent artifacts attached; booking managed by Ops.
* Acceptance criteria:
  + Patient selection is logged; procedure\_selection created; confirmations posted.
* Data/RLS:
  + procedure\_option, procedure\_selection entities; attachments.
* Dependencies:
  + Ops workflows; hospital partners (later phases may deepen).
* Edge cases:
  + If estimates expire, system marks option stale and re‑requests confirmation.

#### **8) Prescription: PDF + QR; Verified‑First; PDF Download Disables QR**

* Description: GP issues Rx. Patient sees suggested verified pharmacies (2–3) and verified map; may select one, walk into any verified pharmacy, or download PDF for non‑verified use.
* Scope:
  + QR token generation (hash stored); first verified claim wins.
  + Downloading PDF permanently disables QR; disclaimer acceptance required.
* Acceptance criteria:
  + After PDF download, qr\_enabled=false and cannot be re‑enabled; thread shows a system message reflecting the change.
  + Verified pharmacy claim reveals item list only; pharmacist cannot see patient PII.
* Data/RLS:
  + prescription.qr\_enabled, pdf\_downloaded\_at, disclaimer timestamp; pharmacy\_claim; item‑only view for pharmacies.
* Dependencies:
  + Pharmacy scanner portal; geo suggestions; notifications.
* Edge cases:
  + Attempted verified claim after QR disabled → reject with clear reason.

#### **9) Verified Pharmacy Claim (No PII in UI; Item‑Only)**

* Description: Partner pharmacist logs in, scans QR to claim if unclaimed; sees medication items only; statuses: ready → dispensed.
* Scope:
  + Claim ledger ties pharmacy to prescription via token; pharmacies get anonymized references for reconciliation.
* Acceptance criteria:
  + Only the first verified claim succeeds per Rx; subsequent attempts are rejected with an explanatory message.
  + Pharmacist UI never renders patient name/phone.
* Data/RLS:
  + pharmacy\_claim with external\_ref\_token; item‑only join view; audit logs on claim.
* Dependencies:
  + Next.js scanner app; auth; Ops reconciliation tools.
* Edge cases:
  + Non‑onboarded scan → show read‑only med list with disclaimer; does not create a claim.

#### **10) Diagnostics (Order → Result) — Specialist‑Owned; Minimum PII to Lab**

* Description: Specialist orders labs; lab admin sees minimum PII to fulfill; uploads results; patient and specialist can view; GP access is policy‑controlled.
* Scope:
  + Lab PII: patient name, order ID, test list, masked phone (per decision).
* Acceptance criteria:
  + Result upload posts a thread message and sends WhatsApp notification within 1 minute.
  + Lab admin cannot access unrelated orders (RLS by lab\_id).
* Data/RLS:
  + lab\_order (with referral\_id when applicable); lab\_result; lab‑scoped views.
* Dependencies:
  + Diagnostics partner portal; storage; notifications.
* Edge cases:
  + Partial results supported as multiple uploads linked to same order.

#### **11) Offline Q&A (Async, 3–6h SLA)**

* Description: Patients can post questions asynchronously in thread; scheduler assigns GP for first response within SLA.
* Scope:
  + Threaded responses; status system messages.
* Acceptance criteria:
  + ≥90% of offline tickets receive first clinician response within 6 hours.
* Data/RLS:
  + consult(mode='offline'); message entries; audit of assignments.
* Dependencies:
  + Scheduler; notifications.
* Edge cases:
  + Auto‑close after inactivity (configurable), with re‑open on patient reply.

#### **12) Notifications (WhatsApp + In‑App)**

* Description: Mirror key events to WhatsApp and in‑app: auth, assignment, call start, referral, appointment, Rx, claims, diagnostics, procedure decisions, escalations.
* Scope:
  + Track provider message IDs and delivery states.
* Acceptance criteria:
  + ≥95% of notifications queued within 2 seconds; delivery status tracked.
* Data/RLS:
  + notification table with provider\_message\_id and status fields.
* Dependencies:
  + WhatsApp Cloud API; template management; in‑app notifier.
* Edge cases:
  + WA outage → in‑app only + Ops manual outreach playbook.

#### **13) Security: RLS, Masking, Audit, Purpose‑Based Access**

* Description: Enforce least privilege via RLS; pharmacies see item‑only; labs see minimum PII; support uses time‑boxed masked impersonation; all privileged actions audited with purpose.
* Scope:
  + Column‑level projections for pharmacy/lab views; reason prompts for Ops/Super Admin PII access.
* Acceptance criteria:
  + Attempted unauthorized access is denied and logged.
  + Support impersonation requires purpose and auto‑expires (e.g., 15 min).
* Data/RLS:
  + RLS policies on consult, message, prescription/item, pharmacy\_claim, lab\_order/result, referral/appointment.
  + audit\_log with masked\_view flag where applicable.
* Dependencies:
  + DB policy definitions; admin tooling for audits.
* Edge cases:
  + Emergency unmasking requires named approver and reason; logged.

#### **14) Retention & Archive (Hot/Cold)**

* Description: Hot storage for chat/media 90 days; prescriptions 24 months; archive thereafter with pointers; on‑demand restore with audit.
* Scope:
  + Nightly archiver; restore flow with signed URLs and expiry.
* Acceptance criteria:
  + Archive jobs complete daily; restore completes within minutes for typical cases; all actions audited.
* Data/RLS:
  + archive pointers in DB; audit restore events.
* Dependencies:
  + Object storage; scheduler; monitoring.
* Edge cases:
  + Restores expire (e.g., 48h) and require re‑request.

#### **15) Minimal Ops Console (Roster, Bookings, Escalations)**

* Description: Ops can manage GP/specialist rosters, confirm/edit appointments, and handle escalations; PHI minimized by default.
* Scope:
  + Roster visibility, appointment calendar, status transitions; purpose‑gated PII access.
* Acceptance criteria:
  + Ops can complete referral → appointment booking without needing PHI fields by default.
* Data/RLS:
  + Restricted views; purpose‑based unmask controls.
* Dependencies:
  + Admin app (Next.js); auth/MFA for admin roles.
* Edge cases:
  + Conflict resolution for double bookings with clear error handling.

#### **16) Localization and Legal Copy (MVP)**

* Description: App UI supports English + Swahili; legal copy (Terms, Privacy, Consent) in English.
* Scope:
  + Notification templates for both languages; language selection at profile.
* Acceptance criteria:
  + Language switch persists; templates render correctly; consent surfaced pre‑consult and for data sharing.
* Dependencies:
  + i18n scaffolding; template management.

Non‑functional baselines (apply to all above)

* Performance: P75 thread load ≤1.5s for last 50 messages; P95 action API latency ≤500ms for core flows (auth verify, message send, claim).
* Availability: Core APIs target ≥99.5% during pilot hours.
* Budget: Keep infra within lean monthly envelope excluding variable WhatsApp fees; avoid adding new vendors unless essential.
* Privacy/Compliance: Align with applicable data protection rules for the pilot (consent, minimization, retention, cross‑border safeguards).

### **MoSCoW — Should Have (S) Requirements for MVP**

Notes

* Model: Healthcare-as-a-Service (HaaS). Keep patient actions minimal; GP/ops orchestrate.
* Channels: WhatsApp-only auth; WhatsApp video for live consults; in‑app text chat thread (AI intake + consult).
* Scope: “Should” items are important but not launch blockers. They enhance reliability, efficiency, compliance, or UX without changing core flows.

1. Appointment Scheduling & Rescheduling Automation (GP + Specialist)

* Description: System-managed time slots, confirmations, reschedules, and reminders for GP follow-ups and specialist appointments.
* Scope:
  + Slot catalogs by provider; book/cancel/reschedule; buffer rules; double-booking prevention.
  + Reminders via WhatsApp + in-app; timezone-aware.
* Acceptance criteria:
  + 0 double-bookings in test; ≥95% confirmations delivered within 5s.
  + Reschedule flow updates thread + notifications atomically.
* Data/RLS:
  + appointment fields extended with buffers/policies; provider availability tables (provider\_availability, blackout\_dates).
  + Patient + provider read; Ops coordination metadata; PHI minimized.
* Dependencies: Ops console; notifications; provider calendars.
* Edge cases: No‑show status updates; provider unavailability triggers auto-offer of nearest alternative.

1. Smart Roster & Load Balancing

* Description: Fair GP assignment using skills/tags and queue depth; priority for red-flag intakes.
* Scope:
  + Skills tags (e.g., pediatrics), language tags; red-flag prioritization.
  + Cooldown to avoid rapid reassignments.
* Acceptance criteria:
  + Red-flag intake waiting time ≤50% of normal median.
  + GP workload variance stays within configured bound (e.g., Gini coefficient threshold).
* Data/RLS: clinician.tags, clinician\_langs; scheduler logs; audit.
* Dependencies: Scheduler; AI intake flags.
* Edge cases: Manual override by Ops with audit.

1. Patient Health Profile (Light PHR)

* Description: Maintain key profile: conditions, meds, allergies, vitals snapshot, and past consult summaries.
* Scope:
  + Editable profile sections; GP can update with consent.
  + Quick summary card shown at consult start.
* Acceptance criteria:
  + Profile card renders ≤500ms; changes audited.
* Data/RLS: patient\_profile(id, user\_id, conditions\_json, meds\_json, allergies\_json, vitals\_json, updated\_by).
  + Patient/assigned clinician read; edits by patient/clinician; Ops no PHI by default.
* Dependencies: UI forms; audit.
* Edge cases: Versioning and rollback of erroneous clinical entries.

1. Scoped Consent & Data Sharing Management

* Description: Versioned consents with scopes (care, referral sharing, foreign transfer).
* Scope:
  + Consent artifacts linked to actions (referral, foreign transfer).
  + Thread posts “consent captured” system messages.
* Acceptance criteria:
  + Actions requiring consent are blocked until consent present.
  + Consent versions retrievable for audit.
* Data/RLS: consent(id, user\_id, scope ENUM[…], version, artifact\_url, granted\_at, revoked\_at).
* Dependencies: Legal copy; storage.
* Edge cases: Revocation cascades to block future shares; prior actions remain auditable.

1. Ops Console v2 (Calendar, Queues, Templates)

* Description: Enhanced ops tools for rosters, appointment calendars, referral queues, canned responses.
* Scope:
  + Calendar view; referral pipeline; escalation macros; message templates.
* Acceptance criteria:
  + Ops can book/modify ≥95% cases without PHI access.
* Data/RLS: restricted admin views; reason-prompt for any PHI unmask.
* Dependencies: Admin MFA; audit.
* Edge cases: Conflict detection on concurrent edits.

1. Partner Onboarding Workflows (Pharmacy/Diagnostics/Specialist)

* Description: Guided onboarding with checklists, document capture, and approval gates.
* Scope:
  + License docs, point-of-contact, training confirmation, test scan/upload.
* Acceptance criteria:
  + Each partner completes end-to-end test (QR claim or result upload) before “active.”
* Data/RLS: partner\_onboarding(id, facility\_id, checklist\_json, status, approved\_by, approved\_at), documents in attachment.
* Dependencies: Admin portal, storage, audit.
* Edge cases: Auto-expire stale onboarding; re-verification reminders.

1. Pharmacy Enhancements: Substitution Policy & Stock Status

* Description: Capture allowed substitutions and out-of-stock redirection.
* Scope:
  + Prescriber substitution policy; pharmacy marks OOS; system suggests nearest partner with availability.
* Acceptance criteria:
  + If OOS, thread receives alternative suggestion within 60s.
* Data/RLS: prescription.substitution\_policy ENUM['no\_sub','generic\_ok','pharmacist\_judgement']; pharmacy\_inventory\_snapshot(optional).
* Dependencies: Partner cooperation; geo services.
* Edge cases: Patient consent required for substitution if policy restricted.

1. Claim Dispute & Reconciliation

* Description: Ops-managed flow to resolve pharmacy claim issues.
* Scope:
  + Open dispute; attach evidence; adjust status under audit; notify parties.
* Acceptance criteria:
  + All status reversals require reason and dual-ack (Ops + pharmacy).
* Data/RLS: claim\_dispute(id, pharmacy\_claim\_id, reason, evidence\_url, resolution, resolved\_by).
* Dependencies: Ops console; audit.
* Edge cases: Fraud flags escalate to Super Admin; block offending pharmacy temporarily.

1. Diagnostics Enhancements: Catalogs & Estimates

* Description: Maintain lab test catalogs, indicative prices, prep notes; support partial results.
* Scope:
  + tests\_master; lab-specific availability; per-test prep.
* Acceptance criteria:
  + Order form autocompletes tests; prep shown in thread upon order.
* Data/RLS: test\_master, lab\_test\_map(price, tat, prep); lab\_result supports multiple files.
* Dependencies: Diagnostics partners.
* Edge cases: Test not available → suggest nearest lab supporting it.

1. Thread UX: Audio Notes, Transcription, Gallery

* Description: Allow audio messages; optional server transcription; attachment gallery per consult.
* Scope:
  + Upload audio; background transcription; gallery view for attachments.
* Acceptance criteria:
  + Audio uploads retry offline; transcription accuracy baseline documented (if enabled).
* Data/RLS: message.kind includes 'audio'; transcription\_text masked for support by default.
* Dependencies: Storage; optional transcription service (can be deferred if not available).
* Edge cases: PII in transcripts handled per masking policies.

1. Notifications Enhancements: Quiet Hours & Preferences

* Description: Respect user quiet hours; per-channel language and frequency preferences.
* Scope:
  + Preference UI; scheduler defers non-urgent messages.
* Acceptance criteria:
  + Non-urgent notifications respect quiet hours; urgent overrides are tagged and audited.
* Data/RLS: user\_pref(id, user\_id, quiet\_hours\_json, lang\_pref, wa\_opt\_in BOOL).
* Dependencies: Notifier scheduler.
* Edge cases: Consent-required notices bypass quiet hours with minimal copy.

1. Security Hardening: Admin MFA, Session Controls, Rate Limits

* Description: Strengthen admin access and protect APIs.
* Scope:
  + MFA for admin roles; session timeout/refresh rules; per-endpoint rate limits; IP allowlist for admin.
* Acceptance criteria:
  + Admin MFA enforced 100%; rate-limit breaches logged with actor and IP.
* Data/RLS: admin\_session logs; security\_events table.
* Dependencies: Auth provider for admin; gateway/WAF.
* Edge cases: Break-glass access with dual approval and full audit trail.

1. Backup & DR Level 2 (PITR + Monthly Restore Tests)

* Description: Improve resilience beyond minimal snapshots.
* Scope:
  + DB PITR/hourly WAL; storage versioning; monthly restore drill with report.
* Acceptance criteria:
  + Achieve RPO ≤1h, RTO ≤4h; monthly drill report archived.
* Data/RLS: backup\_job, restore\_drill logs.
* Dependencies: DB/backups; storage; runbooks.
* Edge cases: Cross-provider restore playbook stored offline.

1. Feature Flags by Region/Partner

* Description: Toggle features per geography or partner cohort.
* Scope:
  + Flags for specialist online availability, diagnostics types, notification templates.
* Acceptance criteria:
  + Flag changes reflect in ≤60s; logged with actor and reason.
* Data/RLS: feature\_flag(key, scope, value\_json, updated\_by).
* Dependencies: Config service; cache invalidation.
* Edge cases: Safe defaults on flag misconfig.

1. Localization Enhancements

* Description: Enrich Swahili medical terminology; fallback chains; template QA.
* Scope:
  + Medical glossary; review loop with clinicians; template preview/testing.
* Acceptance criteria:
  + No placeholder/fallback strings in production templates; clinician sign-off for sensitive terms.
* Data/RLS: i18n resources; template versions.
* Dependencies: Content ops; clinician reviewers.
* Edge cases: Dynamic content (names, dates) formatted per locale.

1. Compliance Reporting (Ops Metrics, Anonymized)

* Description: Export anonymized operational KPIs for internal/regulatory review.
* Scope:
  + SLA adherence, response times, claim success, audit completeness.
* Acceptance criteria:
  + Monthly export generated and archived; no PHI in reports.
* Data/RLS: report\_job logs; aggregate views with PII removed.
* Dependencies: Analytics queries on operational data (not product analytics).
* Edge cases: Redaction verified by automated checks.

## Could have-Co

### **MoSCoW — Could Have (Co) Requirements for MVP+ (patient‑facing)**

Notes

* Model: HaaS. Keep patient actions minimal; no marketplace browsing.
* Channels: WhatsApp auth; WhatsApp video for live consults; in‑app text chat thread is enabled (AI intake + consult).
* Co items are valuable enhancements for patient experience and engagement, but not launch blockers.

1. Contextual Health Education Cards (in‑thread)

* Description: After intake/diagnosis, serve concise, clinician‑approved education cards (English/Swahili) in the chat thread.
* Scope:
  + Trigger by intake tags, GP diagnosis codes, or specialist notes.
  + Content types: what-to-expect, home care tips, red‑flags, when to seek urgent help.
  + Patient can bookmark cards in thread.
* Acceptance criteria:
  + Card appears within 30 seconds of trigger; correct language; no contradictory advice to GP plan.
  + Bookmark/reopen cards from a “Resources” mini-index per consult.
* Data/RLS:
  + education\_card(id, key, lang, content\_json, version).
  + message(kind='link'|'system') posts references; patient/assigned clinician read; Ops metadata only.
* Dependencies: Content ops workflow; clinician review/approval.
* Edge cases: If GP advice conflicts, content is suppressed and flagged for review.

1. Medication Reminders & Adherence Check‑ins

* Description: Optional reminder schedule for prescribed meds with “taken/skip/side‑effect” quick replies.
* Scope:
  + Reminders via WhatsApp + in‑app; adherence summary posted weekly to thread and visible to GP.
  + Side‑effect quick reply opens a GP message template in thread.
* Acceptance criteria:
  + ≥95% reminders sent on schedule; adherence summary accurate to within 1 reminder cycle.
* Data/RLS:
  + med\_reminder(id, prescription\_item\_id, schedule\_json, opt\_in BOOL).
  + adherence\_event(id, user\_id, prescription\_item\_id, status ENUM['taken','skipped','issue'], ts).
  + Patient/GP read; Ops metadata; support masked.
* Dependencies: Scheduler; notifier.
* Edge cases: Timezone changes auto‑recalculate schedule with notice.

1. Follow‑up Nudge & Self‑Reschedule Request (bounded)

* Description: Patient can request a new time for a booked follow‑up/specialist appointment; Ops approves.
* Scope:
  + One active request per appointment; proposed time windows; thread updates on approval/denial.
* Acceptance criteria:
  + Request posts in thread; decision posted within SLA (e.g., 4 hours).
* Data/RLS:
  + reschedule\_request(id, appointment\_id, requested\_slots\_json, status, decided\_by, decided\_at).
* Dependencies: Ops console, provider availability data.
* Edge cases: Multiple patients request the same slot → first approved wins; others offered alternatives.

1. Pre‑Visit Prep Checklists and One‑Tap Check‑In

* Description: For specialist/local visits, send prep steps and allow “I’m on my way/Checked‑in” actions.
* Scope:
  + Checklists from specialist portal; patient confirmation stored; arrival signal shared with Ops/provider.
* Acceptance criteria:
  + Prep card delivered ≥24h prior when possible; check‑in recorded with timestamp.
* Data/RLS: prep\_checklist(id, appointment\_id, items\_json, posted\_at); checkin\_event(id, appointment\_id, type).
* Dependencies: Specialist portal; notifier.
* Edge cases: Late booking posts prep immediately and flags short‑notice.

1. Procedure Cost & Logistics Preview (patient‑facing)

* Description: When specialist proposes procedure options, show estimate ranges, travel/logistics, docs needed; “request assistance” action.
* Scope:
  + Cards per option; links to facility location; upload area for required docs.
* Acceptance criteria:
  + Selection and assistance requests are captured and acknowledged in thread.
* Data/RLS: extend procedure\_option.estimate\_json/prep\_json; assistance\_request table.
* Dependencies: Ops workflows; facility metadata.
* Edge cases: Quote stale → mark “estimate expired” and notify for refresh.

1. Family/Dependent Profiles (basic)

* Description: Consult “on behalf of” a dependent (child/elder) with guardian consent.
* Scope:
  + Add dependent with relationship; select at consult start; thread labeled with dependent’s name.
* Acceptance criteria:
  + RLS isolates dependent records; GP sees correct profile.
* Data/RLS:
  + dependent(id, guardian\_user\_id, name, dob, consent\_artifact\_url, active).
  + consult.dependent\_id NULLABLE; visibility to guardian and assigned clinicians; Ops metadata only.
* Dependencies: Consent capture; profile UI.
* Edge cases: Dependent turns 18 → ownership transfer workflow.

1. Voice Notes for Intake and Chat + Optional Transcription

* Description: Allow audio messages; server‑side transcription stored as masked field.
* Scope:
  + Upload with network retries; transcript displayed to clinicians, masked for support.
* Acceptance criteria:
  + Upload success ≥98% on stable connections; transcript latency ≤60s for 1‑min audio.
* Data/RLS: message.kind='audio'; message.transcript\_text (masked for support).
* Dependencies: Storage; transcription service (feature‑flagged).
* Edge cases: Patient opts out of transcription → skip transcript generation.

1. Photo Uploads for Symptoms (with Safety Prompts)

* Description: Patients attach images (rash/wound) with guidance prompts; EXIF stripped.
* Scope:
  + Size and type validation; redaction prompt for identifiable regions.
* Acceptance criteria:
  + EXIF removed 100%; upload succeeds with progressive feedback.
* Data/RLS: attachment(entity\_type='message'); RLS same as message.
* Dependencies: Storage; image processing.
* Edge cases: Sensitive content flagged → clinician‑only visibility tag.

1. Care Plan Summary & Patient PDF Export

* Description: Auto‑generate a readable care plan after the GP/specialist consult; patient can save/share.
* Scope:
  + Includes diagnosis, meds, tests, follow‑ups, red‑flags; watermark “Not a prescription” when no Rx.
* Acceptance criteria:
  + Plan generated within 2 minutes post‑consult; accessible in thread and profile.
* Data/RLS: care\_plan(id, consult\_id, version, file\_url); patient/assigned clinicians read; Ops metadata only.
* Dependencies: Document renderer.
* Edge cases: Corrections produce new version; prior version retained with reason.

1. Wellness Nudges (opt‑in)

* Description: Light lifestyle suggestions (hydration, sleep, activity) aligned to clinical plan.
* Scope:
  + 2–3 per week max; opt‑in/out control.
* Acceptance criteria:
  + No nudge contradicts prescribed care; clinician‑approved library used.
* Data/RLS: wellness\_pref(id, user\_id, enabled BOOL, categories\_json).
* Dependencies: Content ops; scheduler.
* Edge cases: Acute conditions suppress unrelated nudges automatically.

1. Data Saver Mode (low bandwidth)

* Description: Optimize for poor connectivity: compress uploads, defer large downloads, placeholder images.
* Scope:
  + Toggle in settings; default on when detected low bandwidth.
* Acceptance criteria:
  + 30–50% reduction in data usage for typical consult thread.
* Data/RLS: user\_pref.data\_saver BOOL; delivery logic in clients.
* Dependencies: Client logic; CDN.
* Edge cases: Override when downloading critical medical files.

1. Post‑Consult Feedback Micro‑Survey

* Description: 2–3 question survey (CSAT/NPS) in thread post‑consult.
* Scope:
  + One survey per consult; reminders once after 24h.
* Acceptance criteria:
  + ≥20% response rate target during pilot; results aggregated without PHI.
* Data/RLS: survey\_response(id, consult\_id, scores\_json, comments, ts); analytics view anonymized.
* Dependencies: Analytics; notifier.
* Edge cases: Negative feedback triggers Ops review task.

1. Multi‑language Expansion (phaseable)

* Description: Expand patient UI and templates beyond English/Swahili (feature‑flagged).
* Scope:
  + Add locale packs; clinician‑reviewed medical terms.
* Acceptance criteria:
  + No mixed‑locale strings; fallbacks logged during QA only.
* Data/RLS: i18n resources, template versions.
* Dependencies: Content ops; i18n framework.
* Edge cases: Patient switches language mid‑consult → subsequent messages respect new locale.

1. Second‑Opinion Request (curated, Ops‑routed)

* Description: Patient taps “Request second opinion” in thread; Ops assigns another specialist (not patient‑browse).
* Scope:
  + Captures reason; creates a new referral; original GP is notified.
* Acceptance criteria:
  + Second opinion scheduled or declined with reason within SLA (e.g., 24–48h).
* Data/RLS: referral(second\_opinion\_flag BOOL); message system post; usual RLS.
* Dependencies: Ops routing; specialist capacity.
* Edge cases: Conflicting advice → flagged for clinical governance review.

1. Waitlist Opt‑In for Earlier Slots

* Description: Patient opts into an earlier slot waitlist; auto‑offers if cancellations occur.
* Scope:
  + Offer presented in thread; holds slot for a short window (e.g., 15 minutes) pending patient accept.
* Acceptance criteria:
  + No double booking; offer expiry handled cleanly.
* Data/RLS: waitlist(id, appointment\_id, user\_id, status, offered\_at, expires\_at).
* Dependencies: Scheduler; notifier.
* Edge cases: Multiple waitlisted patients → priority by clinical urgency then FIFO.

1. Caregiver Read‑Only Access (secure share)

* Description: Patient can share a read‑only thread link with OTP to a caregiver for 72 hours.
* Scope:
  + No posting; can view messages/attachments except PHI tagged “clinician‑only.”
* Acceptance criteria:
  + OTP required; access revocable by patient; all views audited.
* Data/RLS: share\_link(id, consult\_id, token\_hash, expires\_at, revoked\_at); audit\_log of views.
* Dependencies: Link service; OTP sender.
* Edge cases: Screenshots cannot be prevented; show caution banner.

1. Symptom Trend Check‑ins (for chronic cases)

* Description: Lightweight daily/weekly check‑ins for chronic conditions (pain score, BP logs if self‑entered).
* Scope:
  + Summaries posted to thread; GP sees trend sparkline at consult start.
* Acceptance criteria:
  + Completion rate ≥50% for opted‑in users.
* Data/RLS: checkin(id, user\_id, condition\_tag, metrics\_json, ts); trend view; RLS to patient/assigned clinicians.
* Dependencies: Scheduler; charts rendering.
* Edge cases: Missed check‑ins pause after threshold to avoid spam.

## Won’t have-W

### **MoSCoW — Won’t Have (W) for MVP**

Notes

* Model: Healthcare-as-a-Service (HaaS), GP‑orchestrated, minimal patient actions.
* Channels: WhatsApp-only authentication; WhatsApp video for live consults; in‑app text chat thread (AI intake + consult). In‑app video is deferred.
* Purpose: Make explicit what is out of scope for MVP/pilot to protect timelines, safety, and budget. For each item: what, why, interim alternative (if any), and risks of forcing it into MVP.

1. In‑App Video Calling (WebRTC/SDK)

* What: Native in‑app video between patient↔GP/specialist.
* Why: Complexity, vendor choice, QA, bandwidth adaptation. MVP standardizes on WhatsApp video.
* Interim alternative: WhatsApp video (default) or another provider per specialist comfort coordinated by Ops; details posted in thread.
* Risk if forced: Delivery delay, call instability, security surface expansion.

1. SMS/Email/Social Logins

* What: Any auth channel other than WhatsApp OTP.
* Why: Increases auth surface, cost, and template management; conflicts with “WhatsApp-only” decision.
* Interim alternative: WhatsApp OTP with TTL/attempt caps.
* Risk if forced: Fragmented identity, more abuse vectors.

1. Patient Marketplace Browsing (Doctors/Pharmacies/Labs)

* What: Open directories and self‑service comparisons.
* Why: HaaS model requires curated flows and GP‑led referrals; browsing adds cognitive load and safety risk.
* Interim alternative: GP presents 3–4 curated specialist options; pharmacies are suggested (2–3 nearby) with verified map.
* Risk if forced: Safety drift, poor decisions, partner QA burden.

1. Patient Self‑Booking With Arbitrary Specialists

* What: Direct specialist selection and booking by the patient.
* Why: Breaks GP‑orchestrated model and curation.
* Interim alternative: GP‑curated referral with Ops booking and confirmations.
* Risk if forced: Mismatches, no‑shows, quality variance.

1. Video/Audio Recording of Consults

* What: Capturing media of WhatsApp or any call.
* Why: Legal, privacy, storage, and consent complexities.
* Interim alternative: Structured clinical notes and care plan summaries in thread.
* Risk if forced: Regulatory exposure, storage cost, consent disputes.

1. Pharmacy View of Patient PII

* What: Showing name/phone/address to pharmacists.
* Why: Privacy decision: item‑only UI for pharmacies.
* Interim alternative: Anonymized claim/ledger references; Ops can unmask under audit if a dispute arises.
* Risk if forced: Data leaks, trust erosion, partner misuse.

1. QR Regeneration After PDF Download

* What: Re‑enabling QR once a PDF prescription is downloaded.
* Why: Anti‑duplication/fraud policy; liability clarity.
* Interim alternative: Verified network via QR or non‑verified via PDF (one‑way switch).
* Risk if forced: Double dispensing, fraud exposure.

1. Home Delivery/Last‑Mile Logistics for Medications

* What: In‑app ordering/delivery and courier tracking.
* Why: Operational and regulatory scope expansion; partner contracts needed.
* Interim alternative: Patient picks up at verified pharmacies.
* Risk if forced: SLA failures, cold‑chain and liability issues.

1. Insurance Eligibility, Benefits, or Claims Adjudication

* What: Health insurance integrations or claims handling.
* Why: Complex payer APIs and regulatory workflows; not needed for MVP pilot.
* Interim alternative: Self‑pay assumptions in care plan summaries.
* Risk if forced: Large delays, compliance overhead.

1. Payments and Billing Inside the App

* What: Collecting patient payments, refunds, or partner settlements in‑app.
* Why: PCI/compliance, refunds, and reconciliation add complexity.
* Interim alternative: Out‑of‑band payment handling by partners; internal ledger only for pharmacy claims.
* Risk if forced: Compliance and ops burdens; payment disputes.

1. Automatic Specialist Discovery Outside Curated Network

* What: Pulling specialists from public registries or third‑party marketplaces.
* Why: Quality and credentialing risks.
* Interim alternative: Rostered specialists vetted by Ops.
* Risk if forced: Clinical governance risk.

1. Full EHR Interoperability (HL7/FHIR Sync)

* What: Bidirectional medical record syncing with external providers.
* Why: Heavy integration and consent management.
* Interim alternative: Attachments and summaries in thread; exportable care plan PDF.
* Risk if forced: Data mapping errors, privacy breaches.

1. Wearable/Device Integrations (BP cuffs, glucometers, watches)

* What: Automatic ingestion of device data.
* Why: Partner SDKs, data quality, and consent handling expand scope.
* Interim alternative: Patient‑entered vitals in thread (optional, later).
* Risk if forced: Data reliability, support load.

1. Open‑Ended Patient↔Specialist Messaging Beyond Consult Context

* What: Unlimited DMs with specialists outside an active consult/referral thread.
* Why: Safety, workload, and medico‑legal boundaries.
* Interim alternative: All communications occur in the consult thread; new concerns start a new consult.
* Risk if forced: Burnout, unmanaged advice, liability.

1. Public Forums/Communities

* What: User‑to‑user groups, comments, or broadcast channels.
* Why: Moderation, misinformation risks, and scope creep.
* Interim alternative: Clinician‑approved education cards in thread (Co item).
* Risk if forced: Safety, moderation overhead.

1. Automated Foreign Procedure Logistics

* What: Visa/travel, cross‑border insurance, hospital financial clearances.
* Why: Complex and jurisdiction‑specific; early phase Ops cannot scale this.
* Interim alternative: Specialist/Admin proposes procedure options; Ops coordinates minimally and posts instructions.
* Risk if forced: Travel/medical legal exposure.

1. Lab Sees More Than Minimal PII

* What: Exposing full demographics or contact info to labs.
* Why: Privacy decision to restrict lab view to: patient name, order ID, test list, masked phone.
* Interim alternative: Ops‑mediated contact when absolutely needed, under audit.
* Risk if forced: PII leakage, regulatory risk.

1. GP Concurrency >2 and Unlimited Live Video Slots

* What: More than 2 active sessions per GP or multiple simultaneous live videos.
* Why: Quality and safety; protects consultation quality.
* Interim alternative: Smart rostering/load balancing (S item).
* Risk if forced: Reduced care quality, timeout breaches.

1. AI‑Only Diagnosis/Prescription Without Clinician

* What: Autonomous clinical decisions or medication orders.
* Why: Safety and regulation; violates HaaS model.
* Interim alternative: AI intake aids context; clinicians decide.
* Risk if forced: Harm and regulatory non‑compliance.

1. Marketing/Ads Surfaces in Patient Thread

* What: Sponsored content or partner ads.
* Why: Trust and focus on essential care.
* Interim alternative: Clinician‑approved education only (Co).
* Risk if forced: Trust erosion, distraction.

1. Push to Non‑WhatsApp External Channels

* What: Telegram, SMS, email notifications for care events.
* Why: Channel proliferation and consent management complexity.
* Interim alternative: WhatsApp + in‑app only.
* Risk if forced: Fragmented comms, template/consent sprawl.

1. Native Offline Mode for Full Consult Threads

* What: Full offline operation with sync.
* Why: Complex conflict resolution and storage encryption concerns.
* Interim alternative: Basic retries and data‑saver mode (Co); critical notices cached minimally.
* Risk if forced: Data conflicts, privacy issues.

1. Full Clinical Coding/Claims Taxonomy

* What: ICD/LOINC/SNOMED at production grade across all flows.
* Why: Adds documentation burden and QA; not needed for pilot outcomes.
* Interim alternative: Lightweight internal tags for education/triggers.
* Risk if forced: Slows clinicians, increases errors.

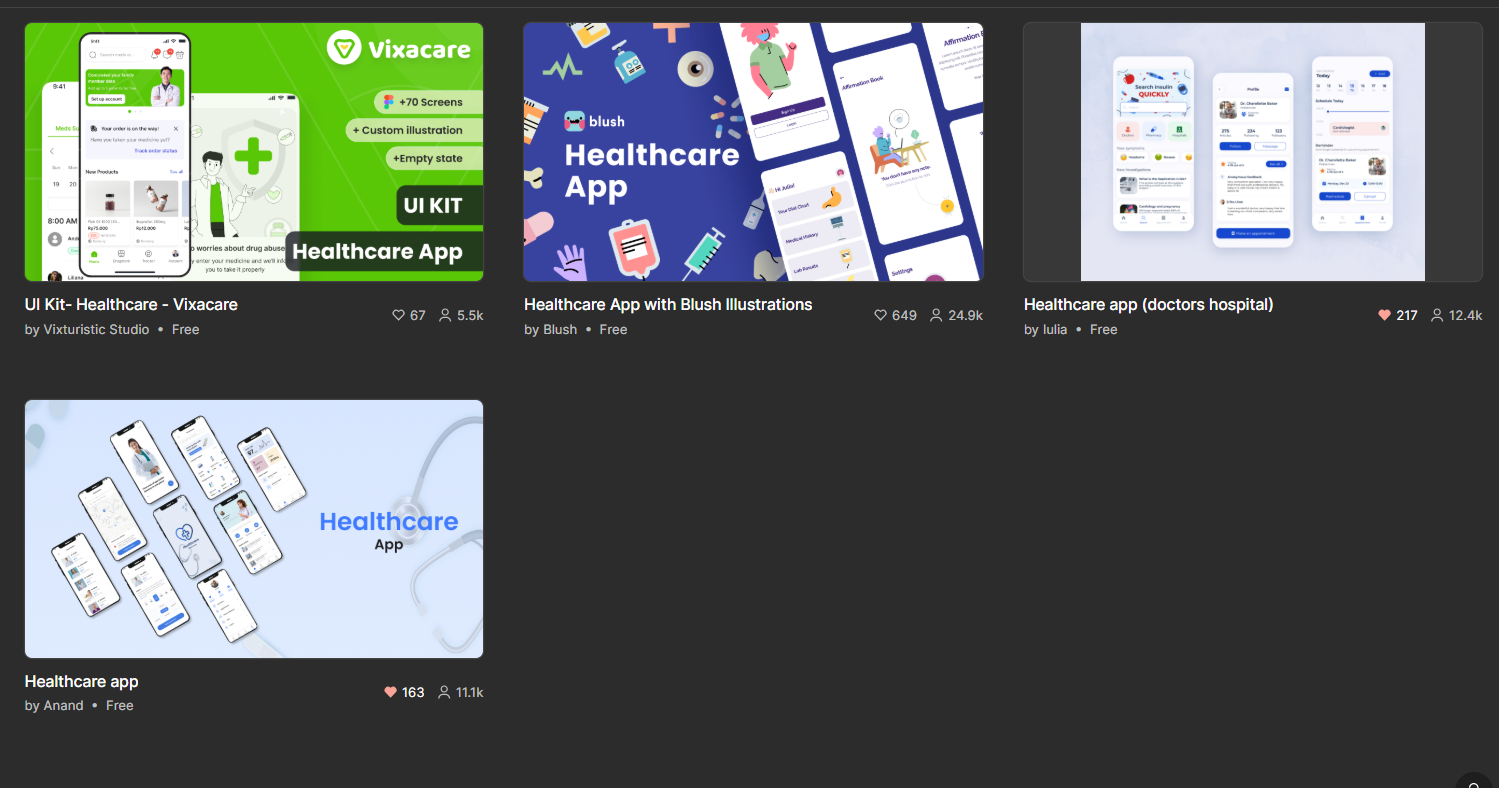
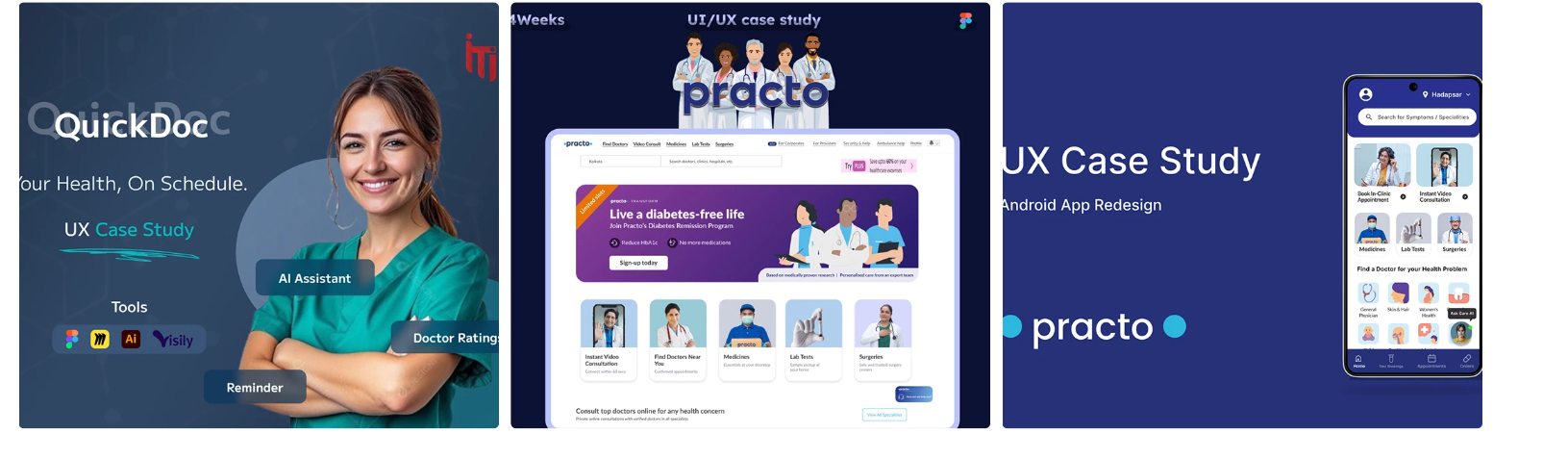
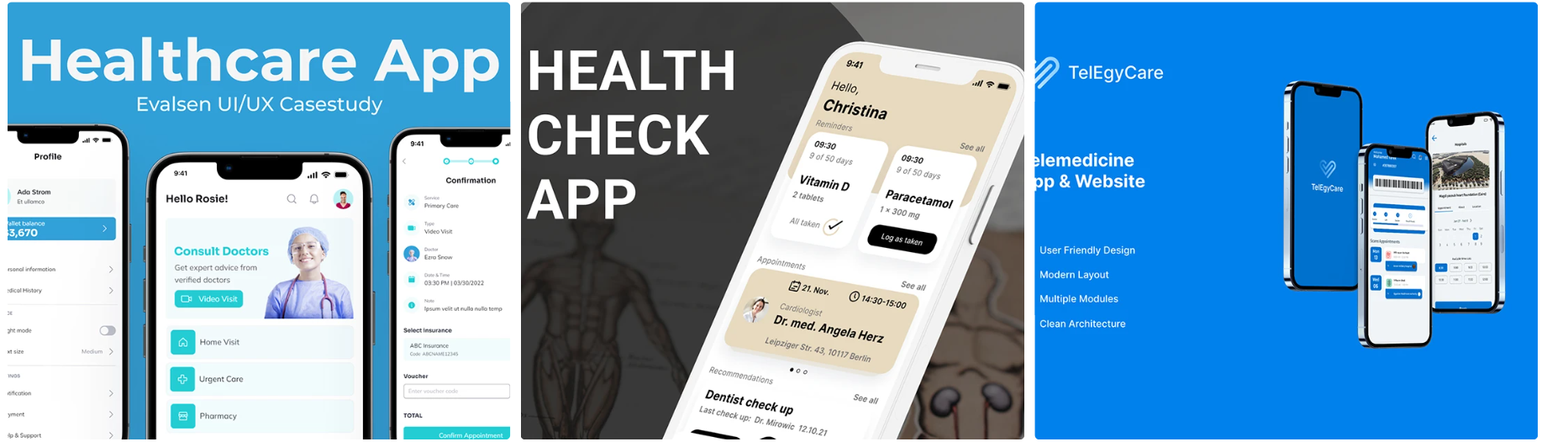
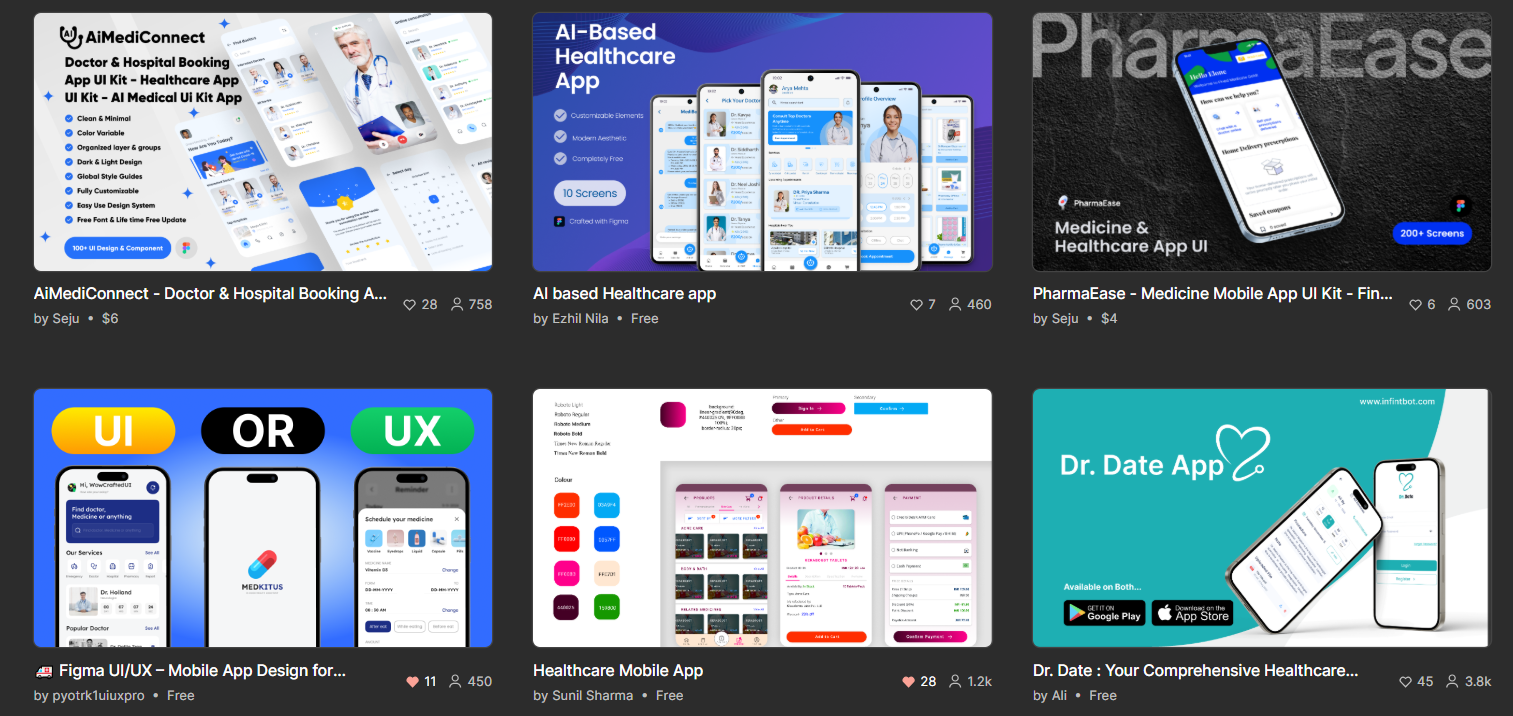
1. End‑to‑End Automated Procurement for Partners

* What: Inventory, purchase orders, reconciliations for pharmacies/labs.
* Why: ERP complexity out of scope.
* Interim alternative: Minimal claim ledger and Ops reconciliation for pharmacy claims.
* Risk if forced: Significant build with little MVP value.

1. Over‑Emphasis on Specific City Rollout Mechanics

* What: Deep, city‑specific ops playbooks embedded in product.
* Why: Keep document pilot‑oriented without locking to a single city.
* Interim alternative: Generic pilot policies with Kenya prioritized.
* Risk if forced: Hard‑to‑port architecture.

Ui/ux documentation

DESIGN INSPIRATIONS  
  




DESIGN UNDERSTANDINGS AND RESEARCH PARTLY

WireFrames

#### **Roles and responsibilities**

* Patient: Completes brief AI intake; attends GP consult (WhatsApp video); reviews Rx; visits verified pharmacies; completes diagnostics as ordered

Type of interaction:-App only; Android focused

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

* GP: Handles consults; issues prescriptions; curates specialist options (local or online); extends session for referral discussion when accepted.  
    
  Type of interaction:-Web-based and desk type

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

* Specialist (local or online): Provides consultation via in-person or WhatsApp video; reviews GP packet; returns plan/recommendations.

Type of interaction:- web based, for both the mobile and non

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

* Pharmacy admin (verified): Scans Rx QR; views med list only; dispenses and updates claim status.
* Type of interaction:- mobile web based login/scanner

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

* Diagnostics admin: Fulfills orders with minimal PII; uploads results.  
    
  Type of interaction:- web based login and dashboard (mobile according to final)

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

* Ops admin: Manages rosters, partner onboarding, booking coordination, escalations; no PHI by default (metadata only).  
    
  Type of interaction:- completely web based- heavy interaction- options focused operations dash/control-board

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

* Support: Time-boxed, masked impersonation for troubleshooting; handles escalations and notifications.  
    
  Type of interaction:- gp type support of tech- also web based only

User expectations:-

No of pages to interact:-

Page names and functionalities(with prioritization score):-

Guidelines

#### **1) Role-to-capability mapping (RBAC) → page and permission design**

What it is:

* Start from your roles and responsibilities, then specify the capabilities each role needs in the product. This is the backbone for which pages exist, who sees what, and what each user can do (view, edit, approve, etc.).

How to do it:

1. List roles (e.g., End User, Operations, Service Provider).
2. For each role, list key business tasks and the data they touch.
3. Turn tasks into capabilities using CRUD + action verbs (Create Order, Approve Refund, Assign Technician, Export CSV).
4. For each capability, attach permissions (view/create/update/delete/approve) and constraints (e.g., can edit only items they own).
5. Group capabilities into coherent screens/pages (or sections) and label per‑role visibility.
6. Capture cross‑role constraints (separation of duties; e.g., an Ops user can refund but cannot approve their own refund).

Deliverables to produce:

* Role→Capability Matrix (by role, list capabilities and permission levels)
* Permissioned Page Inventory (every page lists which roles see it and what each role can do on it)
* Access rules for sensitive actions (e.g., dual‑approval, read‑only for external providers)

Sources and examples:

* The NIST RBAC model is the industry standard, adopted as ANSI/INCITS 359 (updated 2012). See NIST’s overview and role‑engineering material: [NIST RBAC project page](https://csrc.nist.gov/projects/role-based-access-control) and [Role Engineering & RBAC Standards](https://csrc.nist.gov/projects/role-based-access-control/role-engineering-and-rbac-standards). These pages summarize the standard and link to case studies (including a European bank RBAC project that discovered ~1,300 roles) and practical role‑engineering processes.

Why this matters now:

* Getting RBAC right early prevents “role explosion,” avoids security rework, and directly informs which pages exist and the controls they must include.

#### **2) Task-centered information architecture → flows, sitemap, and screen inventory**

What it is:

* Use your roles as “actors” and map their highest‑value tasks into user flows. From flows, derive your sitemap and a screen inventory with purpose, inputs, outputs, states, and components. This keeps wireframes focused on helping users complete real tasks, not just “pages.”

How to do it:

1. For each role, pick top tasks (e.g., “Service Provider accepts a new job,” “Ops triages an issue,” “User tracks order”).
2. Draft step‑by‑step flows for each task (start state → branching decisions → success/failure states).
3. From flows, enumerate the minimal set of screens needed (one screen per decision or data‑entry step).
4. Create a Screen Inventory:
   * Screen name and URL stub
   * Role(s) who access it
   * Task the screen enables (and KPI/success metric)
   * Core data entities and CRUD operations
   * Components needed (filters, tables, forms, modals)
   * States to design (initial/empty, loading, error, success, permission‑denied)
5. Assemble a simple sitemap that groups screens into logical navigation (per role).

Deliverables to produce:

* User flows per role (diagrams or bullet sequences)
* Sitemap (per role and global)
* Screen Inventory (one concise row per screen/page)

Sources and examples:

* Nielsen Norman Group (NN/g) shows practical wireframing conventions and a step‑by‑step approach from navigation to components: [How to Draw a Wireframe](https://www.nngroup.com/articles/draw-wireframe-even-if-you-cant-draw/).
* NN/g’s deliverables study shows flows, site maps, and prototypes are among the most effective artifacts for engineers and stakeholders: [Which UX Deliverables Are Most Commonly Created and Shared?](https://www.nngroup.com/articles/common-ux-deliverables/).

Why this matters now:

* This practice turns your role docs into a concrete list of “screens we actually need” and minimizes “mystery pages” that don’t support a real flow.

#### **3) MoSCoW prioritization of pages, flows, and features (with acceptance criteria)**

What it is:

* Apply the MoSCoW method—Must, Should, Could, Won’t—for the project, the first release/increment, and each iteration/timebox. Tie each item to acceptance criteria so it’s testable.

How to do it:

1. Put every page/flow/feature into a prioritization backlog. For each, write a one‑line outcome and acceptance criteria (use Given/When/Then).
2. Categorize with MoSCoW at three levels:
   * Project-level
   * Increment/Release-level
   * Current timebox/sprint
3. Keep Must‑Have effort under ~60% and maintain a ~20% pool of Could‑Haves as contingency; re‑balance each iteration.
4. Lock Won’t‑Haves “this time” to control scope creep and align expectations.

Deliverables to produce:

* Prioritized Requirements List (PRL) with MoSCoW tags at the three levels above
* Acceptance criteria per item
* A visible rule of thumb (≤60% Musts; ~20% Coulds) and decision rules for Should vs. Could

Sources and examples:

* Official DSDM/Agile Business Consortium guidance (originators of MoSCoW) with recommended balancing and multi‑level use: [Agile Business Consortium: MoSCoW Prioritisation (DSDM Handbook Chapter)](https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html).
* Short video primer from the same body: [Mastering MoSCoW Prioritisation](https://www.agilebusiness.org/resource/video-mastering-moscow-prioritisation.html).

Why this matters now:

* MoSCoW gives you a defensible way to right‑size v1 while preserving a realistic path to iterate, avoiding “everything is a Must.”

#### **4) Annotated wireframes with realistic content, states, and interaction rules**

What it is:

* Move beyond boxes: annotate wireframes with content, validation, rules, permission behaviors, and system states. Use realistic (or AI‑generated but realistic) content instead of lorem ipsum to get better feedback.

How to do it:

1. For each wireframe, annotate:
   * Purpose/KPI of the screen
   * For each component: data source, rules, validations, and empty/error/loading states
   * Permission behavior per role (e.g., button hidden vs. disabled vs. shows a “permission denied” inline message)
   * Navigation and “next step” actions
2. Use content‑first placeholders for labels, table columns, error text, and examples. Avoid lorem ipsum.
3. If helpful, generate realistic placeholder content programmatically/with AI and note the prompt/assumptions in the wireframe annotations.
4. Where a wireframe supports multiple roles, include “role overlays” that show what changes for each role.

Deliverables to produce:

* Annotated low‑ to mid‑fidelity wireframes (paper/Figma) with a consistent annotation legend
* A component checklist to ensure no state is forgotten (loading/empty/error/success/permission‑denied)

Sources and examples:

* Why realistic content matters and how to document it in early designs: NN/g’s “Promptframes” approach (content‑first wireframes): [Promptframes: Evolving the Wireframe for the Age of AI](https://www.nngroup.com/articles/promptframes/).
* Practical wireframing steps and conventions: [NN/g: How to Draw a Wireframe](https://www.nngroup.com/articles/draw-wireframe-even-if-you-cant-draw/).
* Which artifacts work best for different audiences (prototypes for stakeholders; flows/site maps/specs for devs): [NN/g deliverables study](https://www.nngroup.com/articles/common-ux-deliverables/).

Why this matters now:

* Annotated, content‑realistic wireframes reduce rework and let engineering implement faster because edge cases and states are already clarified.

### **Put it all together: a fast, concrete workflow you can run now**

1. Inventory roles and responsibilities (you’ve done this).
2. Role→Capability Matrix (RBAC)
   * For each role, list capabilities with CRUD and constraints.
3. Task flows per role
   * For the top 3–5 tasks per role, write the flow and decisions.
4. Derive the Screen Inventory and Sitemap
   * One line per screen with role access, purpose, data, components, and states.
5. MoSCoW the inventory
   * Tag Must/Should/Could/Won’t at project, release, and sprint levels; attach acceptance criteria.
6. Wireframe in low fidelity with annotations
   * Include states, rules, and permissions in the wireframe; use realistic labels/data.
7. Prototype the Must/Should flows for stakeholder/dev handoff
   * Use interactive prototypes for reviews; keep annotations visible for devs.

### **Minimal templates you can copy/paste**

Role→Capability Matrix (snippet)

Role: Operations

- View Tickets [read]

- Assign Ticket [update] (constraint: cannot assign to self)

- Approve Refund [approve] (SoD: cannot approve if creator)

- Export Ticket List [export] (constraint: only filtered subset; PII masked)

Role: Service Provider

- View Assigned Jobs [read]

- Accept/Decline Job [update]

- Update Job Status [update] (allowed statuses: en route, on-site, completed)

Screen Inventory (one row per screen)

Screen: Ticket Details (/tickets/:id)

Roles: Ops (full), Service Provider (partial), End User (read-only)

Purpose/KPI: Resolve within SLA; reduce back-and-forth

Data: Ticket, Customer, Job, SLA, Comments

Components: Header meta, status badge, editable fields (Ops), activity timeline, comments, assignment widget

States: Loading, Not Found, SLA breach warning, Permission Denied (if non-assigned provider)

Rules:

- Service Provider sees only assigned tickets; fields X/Y hidden.

- Ops can reassign; reassign triggers notification.

- End User sees status + comments, cannot edit.

MoSCoW Backlog item (with acceptance criteria)

Item: Assign Ticket (Ops)

Priority:

- Project: Must

- Release 1: Must

- Sprint 1: Should

Acceptance:

- Given I’m an Ops user on a ticket, When I assign to a provider, Then the provider sees it in "Assigned Jobs" within 60s and receives a notification.

- Given I’m the assigned provider, When I view the ticket, Then I can change status but cannot reassign.

- Given I’m the creator of a refund request, Then I cannot approve it (SoD).

Wireframe annotation checklist (paste on each frame)

* Purpose/KPI:
* Roles and permission behavior:
* Inputs/validations:
* Empty/loading/error/success states:
* Side effects (events/notifications):
* Navigation (next steps/deeplinks):
* Content notes (no lorem ipsum):

### **Quick “online examples” to study the craft**

* RBAC standards and role engineering case material (how organizations structure roles and permissions, and how that drives UI surface): [NIST RBAC project page](https://csrc.nist.gov/projects/role-based-access-control), [Role Engineering & RBAC Standards](https://csrc.nist.gov/projects/role-based-access-control/role-engineering-and-rbac-standards).
* Wireframe conventions, examples, and what stakeholders find most effective (flows, site maps, prototypes): [NN/g wireframing how‑to](https://www.nngroup.com/articles/draw-wireframe-even-if-you-cant-draw/), [NN/g deliverables study](https://www.nngroup.com/articles/common-ux-deliverables/).
* Content‑first wireframing with realistic examples (the Blue Apron promo page example shows how content populates a wireframe and why it improves feedback): [NN/g promptframes](https://www.nngroup.com/articles/promptframes/).
* MoSCoW official method, including guidance on balancing Must/Should/Could and applying it at project/increment/timebox levels: [Agile Business Consortium (DSDM) MoSCoW](https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html), plus [video explainer](https://www.agilebusiness.org/resource/video-mastering-moscow-prioritisation.html).

If you’d like, share your three primary roles and top 5 tasks per role. I can draft your first Role→Capability Matrix, Screen Inventory, and a MoSCoW v1 in your exact domain.

sam wire frame verbal

screen: SplashLoading

platform: Mobile.Android

viewport:

orientation: portrait

width: 360dp

grid: 8dp

layout:

scroll: none

safeArea: true

background: "#FFFFFF"

components:

- id: BrandBackdrop

type: Container

anchor: { top: parent, left: parent, right: parent, bottom: parent }

props:

backgroundGradient:

from: "#F5FBFF"

to: "#FFFFFF"

- id: CenterLogo

type: Image

anchor: { centerX: parent, centerY: parent }

size: { width: 112dp, height: 112dp }

props:

asset: "logo\_mark"

- id: AppName

type: Text

anchor: { top: CenterLogo, below: 16dp, centerX: parent }

props:

text: "Mediconnect"

style: title

- id: Tagline

type: Text

anchor: { top: AppName, below: 8dp, centerX: parent }

props:

text: "Care that connects"

style: caption

color: "#6B7280"

- id: ProgressSpinner

type: Spinner

anchor: { top: Tagline, below: 24dp, centerX: parent }

size: { width: 32dp, height: 32dp }

- id: StatusText

type: Text

anchor: { top: ProgressSpinner, below: 12dp, centerX: parent }

props:

textBinding: SplashStatusMessage

style: caption

color: "#6B7280"

- id: RetryButton

type: Button

visibility: when SplashState == "error" or NetworkStatus == "offline"

anchor: { top: StatusText, below: 12dp, centerX: parent }

props:

label: "Retry"

emphasis: secondary

actions:

onTap: SplashRetry

behaviors:

flow:

- onAppear: CheckPrereqs

- if Locale is unset: Navigate(LanguageGate)

- else if AgeConfirmed is false: Navigate(AgeGate)

- else if ConsentAccepted is false: Navigate(ConsentGate)

- else if AuthSession is none: Navigate(AuthStart)

- else: Navigate(CareHome)

network:

- if NetworkStatus == "offline": set SplashStatusMessage = "No connection"

- if NetworkStatus == "online" and SplashState == "loading": set SplashStatusMessage = "Preparing your app"

retry:

- SplashRetry: RecheckNetwork and restart flow

assets:

- logo\_mark  
  
  
  
screen: CareHome

platform: Mobile.Android

viewport:

orientation: portrait

width: 360dp

grid: 8dp

layout:

scroll: vertical

safeArea: true

components:

- id: TopAppBar#1

type: AppBar

anchor: { top: parent, left: parent, right: parent }

height: 56dp

props:

title: "Mediconnect"

children:

- id: EmergencyButton#2

type: TextButton

anchor: { right: parent, centerY: parent }

props:

label: "Emergency"

style: danger

actions:

onTap: OpenEmergencyRequest

- id: NotificationsButton#3

type: IconButton

anchor: { rightOf: EmergencyButton#2, marginLeft: 8dp, centerY: parent }

props:

icon: "bell"

badgeBinding: NotificationBadgeCount

actions:

onTap: OpenNotificationCenter

- id: ProfileSwitcher#4

type: SegmentedControl

anchor: { top: TopAppBar#1, below: 8dp, left: parent, right: parent }

padding: 12dp

props:

itemsBinding: DependentList # [{id,name,relation,avatar}]

selectedIdBinding: ActivePatientId

allowAdd: false # add dependent flow lives in Profile

actions:

onChange: SetActivePatient(DependentId)

- id: ConnectivityBanner#5

type: Banner

anchor: { below: ProfileSwitcher#4, left: parent, right: parent }

visibility: when Offline == true or SyncQueueCount > 0

props:

style: info

textBinding: ConnectivityMessage # e.g., "Offline. Actions will sync when connected."

- id: CareHeroCard#6

type: Card

anchor:

below: ConnectivityBanner#5 ? ConnectivityBanner#5 : ProfileSwitcher#4

marginTop: 8dp

left: parent

right: parent

padding: 16dp

props:

title: "How can we help today?"

subtitle: "Start with AI or talk to a GP"

icon: "stethoscope"

- id: PrimaryCTA\_StartAIOrGP#7

type: Button

anchor: { below: CareHeroCard#6, marginTop: 12dp, left: 16dp, right: 16dp }

props:

label: "Start AI Chat or Consult a GP"

emphasis: primary

fullWidth: true

actions:

onTap: OpenAIIntakeThread

- id: SecondaryCTA\_AskQuestion#8

type: Button

anchor: { below: PrimaryCTA\_StartAIOrGP#7, marginTop: 8dp, left: 16dp, right: 16dp }

props:

label: "Ask a Question"

emphasis: secondary

fullWidth: true

actions:

onTap: OpenOfflineTicketComposer

- id: ContinueCareCard#9

type: Card

visibility: when HasInProgressConsult == true

anchor: { below: SecondaryCTA\_AskQuestion#8, marginTop: 12dp, left: 16dp, right: 16dp }

padding: 12dp

children:

- id: ContinueCare\_Title#9a

type: Text

props:

textBinding: ContinueCareTitle # e.g., "Continue with Dr. {name}"

style: subtitle

- id: ContinueCare\_Meta#9b

type: Text

props:

textBinding: ContinueCareMeta # e.g., "{mode} • {ETA or UnreadCount}"

style: caption

- id: ContinueCare\_Chevron#9c

type: Icon

anchor: { right: parent, centerY: parent }

props: { icon: "chevron-right" }

actions:

onTap: OpenThread(ConsultId)

- id: QuickActionsRow#10

type: GridRow

anchor:

below: ContinueCareCard#9 ? ContinueCareCard#9 : SecondaryCTA\_AskQuestion#8

marginTop: 16dp

left: 16dp

right: 16dp

props:

columns: 4

itemHeight: 88dp

children:

- id: QA\_Specialists#10a

type: QuickTile

props: { icon: "user-md", label: "Specialists" }

actions: { onTap: GoTab(Specialists) }

- id: QA\_Pharmacy#10b

type: QuickTile

props: { icon: "pill", label: "Pharmacy" }

actions: { onTap: GoTab(Pharmacy) }

- id: QA\_Diagnostics#10c

type: QuickTile

props: { icon: "lab-flask", label: "Diagnostics" }

actions: { onTap: GoTab(Diagnostics) }

- id: QA\_Education#10d

type: QuickTile

props: { icon: "play-circle", label: "Education" }

actions: { onTap: GoEducationHome }

- id: SpecialistTeaser#11

type: Card

anchor: { below: QuickActionsRow#10, marginTop: 12dp, left: 16dp, right: 16dp }

padding: 12dp

contentVariant:

when HasAssignedSpecialist == false:

- id: SpecTeaser\_Empty#11a

type: Text

props: { text: "Your specialist will appear here once assigned by your GP." }

when HasAssignedSpecialist == true:

- id: SpecTeaser\_Row#11b

type: Row

children:

- id: SpecTeaser\_Avatar#11c

type: Avatar

props: { initialsBinding: SpecialistInitials }

- id: SpecTeaser\_Info#11d

type: Column

children:

- id: SpecTeaser\_Name#11e

type: Text

props: { textBinding: SpecialistName, style: body }

- id: SpecTeaser\_Specialty#11f

type: Text

props: { textBinding: SpecialistSpecialty, style: caption }

- id: SpecTeaser\_Chevron#11g

type: Icon

anchor: { right: parent, centerY: parent }

props: { icon: "chevron-right" }

actions:

onTap: GoTab(Specialists)

- id: PharmacyTeaser#12

type: Card

anchor: { below: SpecialistTeaser#11, marginTop: 12dp, left: 16dp, right: 16dp }

padding: 12dp

children:

- id: PharmTeaser\_Title#12a

type: Text

props: { text: "Nearby pharmacies (verified)", style: subtitle }

- id: PharmTeaser\_List#12b

type: List

props:

itemsBinding: NearestPharmaciesTop3 # [{name,distance}]

itemHeight: 48dp

itemTemplate:

- id: PharmItem\_Name#12c

type: Text

props: { textBinding: name, style: body }

- id: PharmItem\_Distance#12d

type: Text

anchor: { right: parent, centerY: parent }

props: { textBinding: distance, style: caption }

actions:

onTap: GoTab(Pharmacy)

- id: EducationCarousel#13

type: Carousel

anchor: { below: PharmacyTeaser#12, marginTop: 12dp, left: parent, right: parent }

height: 160dp

props:

itemsBinding: VideoRecs # [{id,title,tag,isCached}]

showDownloadBadge: true

actions:

onItemTap: OpenEducationDetail(VideoId)

- id: BottomNav#14

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props:

activeTab: "Care"

tabs:

- { key: "Care", icon: "home" }

- { key: "Specialists", icon: "user-md" }

- { key: "Pharmacy", icon: "pill" }

- { key: "Diagnostics", icon: "lab-flask" }

- { key: "Profile", icon: "user" }

actions:

onSelect:

Care: SwitchTab(Care)

Specialists: SwitchTab(Specialists)

Pharmacy: SwitchTab(Pharmacy)

Diagnostics: SwitchTab(Diagnostics)

Profile: SwitchTab(Profile)

behaviors:

permissions:

- when LocationPermission != granted:

PharmacyTeaser#12: showInlinePrompt: true

emptyStates:

- if !HasAssignedSpecialist: SpecialistTeaser#11 shows SpecTeaser\_Empty#11a

offline:

- ConnectivityBanner#5 visible

- New actions queue: OpenAIIntakeThread, OpenOfflineTicketComposer

notes:

- No diagnostics teaser on this page; diagnostics accessible via QuickActions and BottomNav only.

- Dependents: switch only; creation/edit lives in Profile.

- Call launches occur from consult flow; buttons labeled “… via WhatsApp” on relevant pages, not here.  
  
Ai page   
  
screen: AIIntakeChat

platform: Mobile.Android

viewport:

orientation: portrait

width: 360dp

grid: 8dp

layout:

scroll: none

safeArea: true

background: #FFFFFF

components:

- id: TopAppBar#1

type: AppBar

anchor: { top: parent, left: parent, right: parent }

height: 56dp

props:

titleBinding: ActivePatientDisplayName

children:

- id: BackButton#1a

type: IconButton

anchor: { left: parent, centerY: parent }

props: { icon: "arrow-left" }

actions: { onTap: NavigateBack }

- id: EmergencyButton#1b

type: TextButton

anchor: { right: parent, centerY: parent }

props: { label: "Emergency", style: danger }

actions: { onTap: OpenEmergencyRequest }

- id: RiskBanner#2

type: Banner

anchor: { below: TopAppBar#1, left: parent, right: parent }

visibility: when RiskFlag == "high"

props:

style: warning

text: "Your symptoms may need urgent attention."

actions:

onTap: OpenEmergencyRequest

- id: SuggestedPrompts#3

type: ChipGroup

anchor: { below: RiskBanner#2 ? RiskBanner#2 : TopAppBar#1, left: 8dp, right: 8dp }

padding: 8dp

props:

chipsBinding: SuggestedPrompts # ["Fever", "Headache", "Stomach pain", ...]

scroll: horizontal

actions:

onSelect: InsertTextIntoComposer(ChipText)

- id: IntakeFormButton#4

type: Button

anchor: { below: SuggestedPrompts#3, marginTop: 8dp, left: 16dp, right: 16dp }

props:

label: "Add symptom details"

emphasis: tertiary

fullWidth: true

actions:

onTap: Open(IntakeFormSheet#4a)

- id: ThreadList#5

type: List

anchor:

below: IntakeFormButton#4

marginTop: 8dp

left: 0dp

right: 0dp

bottom: ComposerBar#9

props:

itemsBinding: Messages # [{id, sender, kind, text, attachments[]}]

scroll: vertical

reverse: true # newest at bottom; list grows upward

itemTemplate:

- when sender == "user":

- id: UserBubble#5u

type: Bubble

props:

align: right

bgColor: "#DCFCE7"

textBinding: text

- id: UserAttachments#5ua

type: AttachmentStrip

visibility: when attachments.length > 0

props: { itemsBinding: attachments }

actions: { onItemTap: OpenAttachment }

- when sender == "ai":

- id: AIBubble#5a

type: Bubble

props:

align: left

bgColor: "#F3F4F6"

textBinding: text

- id: AIAdviceCardInline#5ac

type: Card

visibility: when adviceInline == true

padding: 12dp

children:

- id: AdviceTitle#5act

type: Text

props: { text: "Suggested steps", style: subtitle }

- id: AdviceList#5acl

type: BulletedList

props: { itemsBinding: adviceItems } # ["Stay hydrated", ...]

- id: AcceptAdviceBtn#5aca

type: Button

props: { label: "Accept advice", emphasis: primary, fullWidth: true }

actions: { onTap: AcceptAIAdvice }

- id: EscalateToGPBtn#5acb

type: Button

props: { label: "Consult a GP", emphasis: secondary, fullWidth: true }

actions: { onTap: OpenGPQueue }

- when sender == "system":

- id: SystemNote#5s

type: Caption

props: { textBinding: text, align: center, color: "#6B7280" }

- id: EscalateCard#6

type: Card

anchor: { above: ComposerBar#9, left: 16dp, right: 16dp, bottom: ComposerBar#9, marginBottom: 8dp }

visibility: when ShowEscalation == true

padding: 12dp

children:

- id: EscalateTitle#6a

type: Text

props: { text: "Talk to a GP", style: subtitle }

- id: EscalateMeta#6b

type: Text

props: { textBinding: EscalationMeta } # e.g., "Recommended based on symptoms"

- id: EscalatePrimary#6c

type: Button

props: { label: "Consult a GP", emphasis: primary, fullWidth: true }

actions: { onTap: OpenGPQueue }

- id: EscalateSecondary#6d

type: Button

props: { label: "Keep chatting", emphasis: secondary, fullWidth: true }

actions: { onTap: DismissEscalation }

- id: ComposerBar#9

type: Composer

anchor: { bottom: BottomNav#10, left: 0dp, right: 0dp }

height: 56dp

props:

placeholder: "Type your symptoms..."

allowAttach: true

allowMic: false # future enables voice

actions:

onSend: SendMessage

onAttachTap: OpenSystemFilePicker # images/pdf

onHeightChange: AdjustThreadInset

- id: BottomNav#10

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props:

activeTab: "Care"

tabs:

- { key: "Care", icon: "home" }

- { key: "Specialists", icon: "user-md" }

- { key: "Pharmacy", icon: "pill" }

- { key: "Diagnostics", icon: "lab-flask" }

- { key: "Profile", icon: "user" }

actions:

onSelect:

Care: SwitchTab(Care)

Specialists: SwitchTab(Specialists)

Pharmacy: SwitchTab(Pharmacy)

Diagnostics: SwitchTab(Diagnostics)

Profile: SwitchTab(Profile)

overlays:

- id: IntakeFormSheet#4a

type: BottomSheet

height: 60%

props: { title: "Symptom details" }

children:

- id: Field\_Symptom#4a1

type: TextField

props: { label: "Main symptom", required: true }

- id: Field\_Duration#4a2

type: Select

props: { label: "Duration", options: ["<24 hours","1-3 days",">3 days"] }

- id: Field\_Severity#4a3

type: Slider

props: { label: "Severity", min: 1, max: 10 }

- id: Field\_Allergies#4a4

type: TextField

props: { label: "Allergies (if any)" }

- id: Field\_Meds#4a5

type: TextField

props: { label: "Current medications" }

- id: SheetActions#4a6

type: Row

children:

- { id: CancelForm#4a6a, type: Button, props: { label: "Cancel" }, actions: { onTap: Close(IntakeFormSheet#4a) } }

- { id: ApplyForm#4a6b, type: Button, props: { label: "Apply", emphasis: primary }, actions: { onTap: SubmitIntakeFields } }

- id: ConsentDialog#11

type: Dialog

visibility: when NeedsConsent == true

props:

title: "Telemedicine and privacy"

message: "By continuing you agree to telemedicine and data processing terms."

primaryLabel: "Agree"

secondaryLabel: "Review"

actions:

onPrimary: AcceptConsent

onSecondary: OpenLegalDocs

behaviors:

connectivity:

- when OnlineStatus == false:

ThreadList#5: allowReadCache: true

ComposerBar#9: queueSends: true

showBanner: "Offline. Messages will send when connected."

files:

- allowedTypes: ["image/jpeg","image/png","application/pdf"]

- maxSizeMB: 10

- maxCountPerConsult: 5

ai:

- detectRedFlags: true

- if redFlag: set RiskFlag = "high", show RiskBanner#2 and EscalateCard#6

- generateAdvice: true

escalation:

- OpenGPQueue: navigateTo "OnlineGPQueueAndAssignment" with ThreadId

consent:

- firstThreadOnly: show ConsentDialog#11 before sending

mvp:

- Text chat with AI

- Structured intake sheet

- AI advice card

- Escalate to GP (navigates to GP queue; call via WhatsApp later)

- Attach images/PDF

- Emergency button

- Bottom nav present

future:

- Voice input in composer

- In-app audio/video call

- Symptom timeline and trend view

- Local language toggle inside chat

- Save advice to Care Plan  
  
  
Pharmacy pages   
  
pharmsy pages

screen: PharmacyHome

platform: Mobile.Android

viewport:

orientation: portrait

width: 360dp

grid: 8dp

layout:

scroll: vertical

safeArea: true

background: "#FFFFFF"

components:

- id: TopAppBar#1

type: AppBar

anchor: { top: parent, left: parent, right: parent }

height: 56dp

props: { title: "Pharmacy" }

children:

- id: EmergencyButton#1a

type: TextButton

anchor: { right: parent, centerY: parent }

props: { label: "Emergency", style: danger }

actions: { onTap: OpenEmergencyRequest }

- id: NotificationsButton#1b

type: IconButton

anchor: { rightOf: EmergencyButton#1a, marginLeft: 8dp, centerY: parent }

props: { icon: "bell", badgeBinding: NotificationBadgeCount }

actions: { onTap: OpenNotificationCenter }

- id: LocationPrompt#2

type: Banner

anchor: { below: TopAppBar#1, left: parent, right: parent }

visibility: when LocationPermission != "granted"

props:

style: info

text: "Allow location to find nearby verified pharmacies"

actions:

onTap: RequestLocationPermission

- id: MapCard#3

type: Card

anchor:

below: LocationPrompt#2 ? LocationPrompt#2 : TopAppBar#1

marginTop: 8dp

left: 16dp

right: 16dp

padding: 8dp

children:

- id: PharmacyMap#3a

type: MapView

height: 200dp

props:

centerBinding: CurrentLocationOrDefaultNairobi

markersBinding: VerifiedPharmacies # [{id,name,lat,lng,isTop3}]

showUserLocation: true

- id: MapControlsRow#3b

type: Row

anchor: { below: PharmacyMap#3a, marginTop: 8dp, left: parent, right: parent }

children:

- id: RecenterBtn#3b1

type: Button

props: { label: "Recenter" }

actions: { onTap: RecenterMap }

- id: ViewToggle#3b2

type: SegmentedControl

anchor: { right: parent }

props:

items: ["Top 3", "All"]

selectedBinding: MapFilterSelection

actions:

onChange: SetMapFilter(MapFilterSelection)

- id: Top3List#3c

type: List

anchor: { below: MapControlsRow#3b, marginTop: 8dp, left: parent, right: parent }

props:

itemsBinding: NearestPharmaciesTop3 # [{id,name,distance}]

itemHeight: 56dp

itemTemplate:

- id: PharmRow#3cItem

type: Row

children:

- { id: PharmName#3c1, type: Text, props: { textBinding: name, style: body } }

- { id: PharmDistance#3c2, type: Text, anchor: { right: parent, centerY: parent }, props: { textBinding: distance, style: caption } }

actions:

onItemTap: Open(PharmacyDetail, { pharmacyId: id })

- id: CurrentRxPanel#4

type: Card

anchor: { below: MapCard#3, marginTop: 12dp, left: 16dp, right: 16dp }

padding: 12dp

visibility: when HasActivePrescription == true

children:

- { id: CurrentRxTitle#4a, type: Text, props: { text: "Your prescription", style: subtitle } }

- id: RxQRCode#4b

type: QRCode

height: 160dp

props: { valueBinding: ActivePrescriptionToken }

- id: RxActionsRow#4c

type: Row

anchor: { below: RxQRCode#4b, marginTop: 8dp }

children:

- { id: ShowQRBtn#4c1, type: Button, props: { label: "Show QR at pharmacy" }, actions: { onTap: MaxBrightnessAndFullscreenQR } }

- { id: DownloadPDFBtn#4c2, type: Button, props: { label: "Download PDF" }, actions: { onTap: DownloadPrescriptionPDF(ActivePrescriptionId) } }

- { id: ShareBtn#4c3, type: Button, props: { label: "Share" }, actions: { onTap: OpenShareSheet(ActivePrescriptionPDF) } }

- id: DeliveryCTA#4d

type: Button

visibility: when FeatureDeliveryEnabled == true

anchor: { below: RxActionsRow#4c, marginTop: 8dp }

props:

label: "Order delivery"

emphasis: secondary

fullWidth: true

actions:

onTap: Open(DeliveryOrder, { prescriptionId: ActivePrescriptionId })

- id: PrescriptionsHeader#5

type: Row

anchor: { below: CurrentRxPanel#4 ? CurrentRxPanel#4 : MapCard#3, marginTop: 16dp, left: 16dp, right: 16dp }

children:

- { id: PrescriptionsTitle#5a, type: Text, props: { text: "Prescriptions", style: subtitle } }

- id: FilterTabs#5b

type: SegmentedControl

anchor: { right: parent }

props:

items: ["Active", "All"]

selectedBinding: RxFilter

actions:

onChange: SetPrescriptionFilter(RxFilter)

- id: PrescriptionList#6

type: List

anchor: { below: PrescriptionsHeader#5, marginTop: 8dp, left: 16dp, right: 16dp, bottom: BottomNav#9 }

props:

itemsBinding: PrescriptionItems # [{id,date,status,pharmacyName?}]

itemHeight: 72dp

itemTemplate:

- id: RxItemRow#6a

type: Row

children:

- id: RxItemInfoCol#6a1

type: Column

children:

- { id: RxDate#6a1a, type: Text, props: { textBinding: date, style: body } }

- { id: RxStatus#6a1b, type: Chip, props: { labelBinding: status } } # New, Claimed, Dispensed, Expired

- id: RxRightCol#6a2

type: Row

anchor: { right: parent, centerY: parent }

children:

- id: ExpiredWatermark#6a2a

type: Watermark

visibility: when status == "Expired"

props: { text: "EXPIRED" }

- id: OpenBtn#6a2b

type: IconButton

props: { icon: "chevron-right" }

actions:

onItemTap: Open(PrescriptionDetail, { prescriptionId: id })

- id: BottomNav#9

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props:

activeTab: "Pharmacy"

tabs:

- { key: "Care", icon: "home" }

- { key: "Specialists", icon: "user-md" }

- { key: "Pharmacy", icon: "pill" }

- { key: "Diagnostics", icon: "lab-flask" }

- { key: "Profile", icon: "user" }

actions:

onSelect:

Care: SwitchTab(Care)

Specialists: SwitchTab(Specialists)

Pharmacy: SwitchTab(Pharmacy)

Diagnostics: SwitchTab(Diagnostics)

Profile: SwitchTab(Profile)

behaviors:

permissions:

- if LocationPermission != "granted":

MapCard#3: show simplified state without user dot

offline:

- allow viewing cached pharmacies and prescriptions

- queue PDF downloads/share until online

states:

- if HasActivePrescription == false:

hide CurrentRxPanel#4

screen: PrescriptionDetail

platform: Mobile.Android

viewport:

orientation: portrait

width: 360dp

grid: 8dp

layout:

scroll: vertical

safeArea: true

background: "#FFFFFF"

components:

- id: TopAppBar#1

type: AppBar

anchor: { top: parent, left: parent, right: parent }

height: 56dp

props: { title: "Prescription" }

children:

- { id: BackBtn#1a, type: IconButton, anchor: { left: parent, centerY: parent }, props: { icon: "arrow-left" }, actions: { onTap: NavigateBack } }

- { id: EmergencyButton#1b, type: TextButton, anchor: { right: parent, centerY: parent }, props: { label: "Emergency", style: danger }, actions: { onTap: OpenEmergencyRequest } }

- id: RxHeader#2

type: Row

anchor: { below: TopAppBar#1, left: 16dp, right: 16dp, marginTop: 8dp }

children:

- { id: RxDate#2a, type: Text, props: { textBinding: RxDateLabel, style: subtitle } }

- { id: RxStatus#2b, type: Chip, anchor: { right: parent }, props: { labelBinding: RxStatusLabel } }

- id: QRBlock#3

type: Card

anchor: { below: RxHeader#2, marginTop: 8dp, left: 16dp, right: 16dp }

padding: 12dp

children:

- { id: QR#3a, type: QRCode, height: 200dp, props: { valueBinding: RxToken } }

- id: QRButtons#3b

type: Row

anchor: { below: QR#3a, marginTop: 8dp }

children:

- { id: ShowFull#3b1, type: Button, props: { label: "Show QR" }, actions: { onTap: MaxBrightnessAndFullscreenQR } }

- { id: DownloadPDF#3b2, type: Button, props: { label: "Download PDF" }, actions: { onTap: DownloadPrescriptionPDF(PrescriptionId) } }

- { id: SharePDF#3b3, type: Button, props: { label: "Share" }, actions: { onTap: OpenShareSheet(PrescriptionPDF) } }

- id: ItemsCard#4

type: Card

anchor: { below: QRBlock#3, marginTop: 12dp, left: 16dp, right: 16dp }

padding: 12dp

children:

- { id: ItemsTitle#4a, type: Text, props: { text: "Medicines", style: subtitle } }

- id: ItemsList#4b

type: List

props:

itemsBinding: RxItems # [{drug,strength,form,quantity,instructions}]

itemHeight: 56dp

itemTemplate:

- id: ItemRow#4b1

type: Row

children:

- { id: Drug#4b1a, type: Text, props: { textBinding: drug, style: body } }

- { id: Dose#4b1b, type: Text, anchor: { right: parent }, props: { textBinding: strength, style: caption } }

- { id: Instructions#4b2, type: Text, props: { textBinding: instructions, style: caption, color: "#6B7280" } }

- id: RenewalOrDelivery#5

type: Column

anchor: { below: ItemsCard#4, marginTop: 12dp, left: 16dp, right: 16dp }

children:

- id: RenewalBar#5a

type: Banner

visibility: when RxStatusLabel == "Expired"

props:

style: info

text: "This prescription has expired"

actions:

onTap: RequestRenewalFromGP

- id: DeliveryCTA#5b

type: Button

visibility: when FeatureDeliveryEnabled == true

props:

label: "Order delivery"

emphasis: secondary

fullWidth: true

actions:

onTap: Open(DeliveryOrder, { prescriptionId: PrescriptionId })

- id: BottomNav#9

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props:

activeTab: "Pharmacy"

tabs:

- { key: "Care", icon: "home" }

- { key: "Specialists", icon: "user-md" }

- { key: "Pharmacy", icon: "pill" }

- { key: "Diagnostics", icon: "lab-flask" }

- { key: "Profile", icon: "user" }

actions:

onSelect:

Care: SwitchTab(Care)

Specialists: SwitchTab(Specialists)

Pharmacy: SwitchTab(Pharmacy)

Diagnostics: SwitchTab(Diagnostics)

Profile: SwitchTab(Profile)

behaviors:

offline:

- allow viewing cached QR and items

- queue download or share until online

screen: PharmacyDetail

platform: Mobile.Android

viewport:

orientation: portrait

width: 360dp

grid: 8dp

layout:

scroll: vertical

safeArea: true

background: "#FFFFFF"

components:

- id: TopAppBar#1

type: AppBar

height: 56dp

props: { titleBinding: PharmacyName }

children:

- { id: BackBtn#1a, type: IconButton, anchor: { left: parent, centerY: parent }, props: { icon: "arrow-left" }, actions: { onTap: NavigateBack } }

- { id: EmergencyButton#1b, type: TextButton, anchor: { right: parent, centerY: parent }, props: { label: "Emergency", style: danger }, actions: { onTap: OpenEmergencyRequest } }

- id: MiniMap#2

type: MapView

anchor: { below: TopAppBar#1, left: 16dp, right: 16dp }

height: 160dp

props:

centerBinding: PharmacyLatLng

markers: [{ id: "self", latlngBinding: PharmacyLatLng }]

- id: InfoCard#3

type: Card

anchor: { below: MiniMap#2, marginTop: 8dp, left: 16dp, right: 16dp }

padding: 12dp

children:

- { id: Address#3a, type: Text, props: { textBinding: PharmacyAddress, style: body } }

- { id: Distance#3b, type: Text, props: { textBinding: PharmacyDistance, style: caption, color: "#6B7280" } }

- { id: Hours#3c, type: Text, props: { textBinding: PharmacyHours, style: caption } }

- id: ActionsRow#4

type: Row

anchor: { below: InfoCard#3, marginTop: 8dp, left: 16dp, right: 16dp }

children:

- { id: DirectionsBtn#4a, type: Button, props: { label: "Get directions" }, actions: { onTap: OpenMapsIntent(PharmacyLatLng) } }

- { id: CallBtn#4b, type: Button, props: { label: "Call" }, actions: { onTap: DialPhone(PharmacyPhone) } }

- id: DeliveryCTA#5

type: Button

visibility: when FeatureDeliveryEnabled == true

anchor: { below: ActionsRow#4, marginTop: 8dp, left: 16dp, right: 16dp }

props:

label: "Request delivery"

emphasis: secondary

fullWidth: true

actions:

onTap: Open(DeliveryOrder, { pharmacyId: PharmacyId })

- id: BottomNav#9

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props:

activeTab: "Pharmacy"

tabs:

- { key: "Care", icon: "home" }

- { key: "Specialists", icon: "user-md" }

- { key: "Pharmacy", icon: "pill" }

- { key: "Diagnostics", icon: "lab-flask" }

- { key: "Profile", icon: "user" }

actions:

onSelect:

Care: SwitchTab(Care)

Specialists: SwitchTab(Specialists)

Pharmacy: SwitchTab(Pharmacy)

Diagnostics: SwitchTab(Diagnostics)

Profile: SwitchTab(Profile)

screen: DeliveryOrder

platform: Mobile.Android

layout:

scroll: vertical

safeArea: true

components:

- id: TopAppBar#1

type: AppBar

props: { title: "Order delivery" }

children:

- { id: BackBtn#1a, type: IconButton, props: { icon: "arrow-left" }, actions: { onTap: NavigateBack } }

- id: RxSummary#2

type: Card

anchor: { below: TopAppBar#1, left: 16dp, right: 16dp, marginTop: 8dp }

padding: 12dp

children:

- { id: RxDate#2a, type: Text, props: { textBinding: RxDateLabel, style: body } }

- { id: ItemsCount#2b, type: Text, props: { textBinding: ItemsCountLabel, style: caption } }

- id: PharmacyPicker#3

type: Select

anchor: { below: RxSummary#2, left: 16dp, right: 16dp, marginTop: 8dp }

props:

label: "Pharmacy"

optionsBinding: EligibleDeliveryPharmacies

- id: AddressSection#4

type: Card

anchor: { below: PharmacyPicker#3, left: 16dp, right: 16dp, marginTop: 8dp }

padding: 12dp

children:

- { id: AddressTitle#4a, type: Text, props: { text: "Delivery address", style: subtitle } }

- { id: AddressPicker#4b, type: Select, props: { label: "Saved addresses", optionsBinding: SavedAddresses } }

- { id: AddressEdit#4c, type: Button, props: { label: "Add or edit address" }, actions: { onTap: Open(ProfileAddresses) } }

- id: SlotSection#5

type: Card

anchor: { below: AddressSection#4, left: 16dp, right: 16dp, marginTop: 8dp }

padding: 12dp

children:

- { id: SlotTitle#5a, type: Text, props: { text: "Delivery time", style: subtitle } }

- { id: SlotSelect#5b, type: Select, props: { label: "Choose a slot", optionsBinding: DeliverySlots } }

- id: CostSummary#6

type: Card

anchor: { below: SlotSection#5, left: 16dp, right: 16dp, marginTop: 8dp }

padding: 12dp

children:

- { id: ItemsTotal#6a, type: KeyValue, props: { key: "Items", valueBinding: ItemsTotal } }

- { id: DeliveryFee#6b, type: KeyValue, props: { key: "Delivery", valueBinding: DeliveryFee } }

- { id: Total#6c, type: KeyValue, props: { key: "Total", valueBinding: OrderTotal, highlight: true } }

- id: PaymentCTA#7

type: Button

anchor: { below: CostSummary#6, left: 16dp, right: 16dp, marginTop: 12dp }

props:

label: "Proceed to payment"

emphasis: primary

fullWidth: true

actions:

onTap: Navigate(ProfilePayments, { context: "pharmacy\_order", orderId: TempOrderId })

- id: BottomNav#9

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props: { activeTab: "Pharmacy", tabs: [ {key:"Care"},{key:"Specialists"},{key:"Pharmacy"},{key:"Diagnostics"},{key:"Profile"} ] }

screen: DeliveryTracking

platform: Mobile.Android

layout:

scroll: vertical

safeArea: true

components:

- id: TopAppBar#1

type: AppBar

props: { title: "Order tracking" }

children:

- { id: BackBtn#1a, type: IconButton, props: { icon: "arrow-left" }, actions: { onTap: NavigateBack } }

- id: StatusTimeline#2

type: Timeline

anchor: { below: TopAppBar#1, left: 16dp, right: 16dp, marginTop: 8dp }

props:

steps:

- { key: "received", label: "Order received" }

- { key: "preparing", label: "Preparing" }

- { key: "out", label: "Out for delivery" }

- { key: "delivered", label: "Delivered" }

activeStepBinding: CurrentStatusKey

- id: CourierInfo#3

type: Card

anchor: { below: StatusTimeline#2, left: 16dp, right: 16dp, marginTop: 8dp }

padding: 12dp

children:

- { id: CourierName#3a, type: Text, props: { textBinding: CourierName, style: body } }

- { id: CourierPhone#3b, type: Button, props: { label: "Call courier" }, actions: { onTap: DialPhone(CourierPhone) } }

- id: ActionsRow#4

type: Row

anchor: { below: CourierInfo#3, left: 16dp, right: 16dp, marginTop: 8dp }

children:

- { id: ContactPharmacy#4a, type: Button, props: { label: "Contact pharmacy" }, actions: { onTap: DialPhone(PharmacyPhone) } }

- { id: CancelOrder#4b, type: Button, props: { label: "Cancel order" }, actions: { onTap: OpenCancelOrderDialog } }

- id: BottomNav#9

type: BottomNavigation

anchor: { bottom: parent, left: parent, right: parent }

props: { activeTab: "Pharmacy", tabs: [ {key:"Care"},{key:"Specialists"},{key:"Pharmacy"},{key:"Diagnostics"},{key:"Profile"} ] }

model 1 - Sathanandh

# **Page Names + Functionalities (with prioritization score):**

# Splash/Auth (M) → WhatsApp login, prefilled country code, support link.

# Home/Dashboard (M) → Dynamic status cards, persistent FAB for consult.

# AI Intake (M) → Adaptive questions, summary confirmation.

# Waiting/Queue (S) → Animated reassurance, educational content.

# Active Consult (M) → Chat, structured GP questions, WhatsApp video.

# Referral Selection (S) → GP-curated options, confirm and book.

# Prescription & Pharmacy (M) → QR hero, pharmacy selection, PDF fallback.

# Health Locker (S) → Tabs (consults, prescriptions, labs), share control.

# Education Library (C) → Symptom grid, offline videos.

# 

# **Page Categorization by User Type**

### **1. Patient**

* **Login / Auth Page** (mobile-first entry)
* **Home / Dashboard**
* **AI Intake Form** (fills before GP sees summary)
* **Waiting / Queue Status**
* **Active Consult** (joins GP session)
* **Referral Selection** (chooses specialist)
* **Prescription & Pharmacy** (select pharmacy / fallback PDF)
* **Diagnostics / Lab Orders** (view results, upload if needed)
* **Health Locker** (stores & shares records)
* **Education Content Library**

### **2. General Practitioner (GP)**

* **Consult Queue** (pull next consult, triage)
* **Active Consult** (drives session, structured notes, video option)
* **Referral Selection** (curates options for patient)
* **Prescription & Pharmacy** (issues Rx to patient)

### **3. Specialist**

* **Consult Queue** (only for referred cases)
* **Active Consult** (for deeper consults after referral)
* **Referral Selection** (receives patient confirmation → starts engagement)
* **Health Locker** (access shared records)

### **4. Pharmacy**

* **Prescription & Pharmacy** (scan QR, validate, dispense)
* (Ops-initiated onboarding handled under Ops/Admin, not pharmacy UI)

### **5. Diagnostics**

* **Diagnostics / Lab Orders** (upload results, manage order queue)
* **Health Locker** (patient-shared results)

### **6. Ops / Support**

* **Login / Auth Page** (web entry)
* **Consult Queue** (metadata-only for monitoring)
* **Ops / Support Dashboards** (rosters, escalations, onboarding, impersonation)
* **Health Locker** (masked access for troubleshooting)
* **Education Content Library** (support may recommend to patients)