

Goal

Enhance HURE Care with AI + workflow + billing/insurance readiness, while staying clinician-safe, audit-safe, and Kenya/East Africa practical. AI must always be assistive (clinician reviews + signs).

1) AI-Assisted Clinical Notes (SOAP) + ICD-10 Suggestions

What we want (User-facing)

In Patient → Clinical Notes:

- Toggle: **Record (Audio → Text) OR Text-only**
- If Record:
 - clinician records **after appointment**
 - transcript appears immediately
 - clinician edits transcript
 - clinician must click “**Mark transcript reviewed**” (required)
- AI generates SOAP:
 - supports both **Single Note** and **Broken-down SOAP**
 - Single Note can be **parsed into SOAP fields**
(Subjective/Objective/Assessment/Plan)
- ICD-10 suggestions:
 - AI suggests ICD-10 codes
 - clinician must **select** (no auto-assign)
- Save:
 - “Save Draft”
 - “Save & Sign” (final)
- Show clearly whether note was created by:
 - **Audio transcription or Text-only**

Frontend requirements

- In SOAP screen:
 - Record button + timer + stop
 - Transcript editor box
 - Button: **Mark transcript reviewed** (required before Save & Sign)
 - Buttons:
 - Generate SOAP (from transcript)
 - Parse Single Note → SOAP (if Single Note is used)
 - Rewrite for clarity (assist only; do not change meaning)
- Warnings / compliance UI:
 - If transcript not reviewed:
 - block Save & Sign
 - show message: “**AI transcription must be reviewed before saving.**”
- Security UX:
 - **90-second warning**
 - **2-minute screen lock**
 - “Last accessed” + “Last edited” shown
- One-time “legal safe” modal shown **only on first Save & Sign** (not on future edits)

Backend requirements

- Store:

- transcript text + SOAP fields
 - sourceType: AUDIO | TEXT
 - transcriptReviewed: true/false
 - reviewedAt, reviewedBy
- Audio handling:
 - audio is **temporary**
 - delete after transcript reviewed + note saved/signed (or after short TTL)
- Audit logs:
 - start/stop recording
 - transcript edited
 - transcript marked reviewed
 - SOAP generated
 - ICD-10 suggested + selected
 - saved draft / signed
 - edits after signing

Acceptance criteria

- Clinician cannot sign without marking transcript reviewed
- Audio is deleted automatically after save/sign (or TTL)
- Notes display “Created via Audio” vs “Text-only”
- ICD-10 suggestions require manual selection

2) AI End-of-Day Closure (Incomplete Notes Control)

What we want (User-facing)

Reduce “forgotten notes” without punishing providers:

- Provider sees same-day reminder **before logging off**
- Employer/admin sees uncompleted notes **after 24 hours** (grace period)

Frontend requirements

Provider view

- At end of day (or when logging out / dashboard banner):
 - “You have X incomplete notes today”
 - list of encounters with status:
 - No note started
 - Draft exists not signed
- Quick action: “Open chart” / “Complete now” / “Save draft”

Employer/Admin view (after 24h)

- Reporting view showing:
 - provider
 - encounter date
 - note status
 - aging buckets: 24–48h, 48–72h, 72h+
- Employer should **not** see note content here (only completion status)

Backend requirements

- A scheduled job / daily check:
 - identifies encounters older than same-day with incomplete notes
- Visibility rules:

- provider: same day
- employer: only after 24h
- Audit logs:
 - reminder shown
 - reminder dismissed
 - note completed after reminder

Acceptance criteria

- Provider can see same-day incomplete work
- Employer only sees after 24h
- Aging logic works correctly

3) AI-Assisted Referral Notes (fillable + downloadable)

What we want (User-facing)

Add “Referral Note” as a note type:

- AI can **generate referral note from existing clinical notes**
- Clinician edits before signing
- Save into patient history
- Output:
 - **Download PDF**
 - **Print**
 - **Copy email text** (clinic sends externally)

Source selection logic (important)

We must allow clinicians to choose the **visit date** source:

- Dropdown: “Source visit date”
- Default selection:
 - most recent **SIGNED SOAP** in selected visit
- If none signed:
 - use most recent draft (show warning)
- Extraction rules:
 - Broken SOAP → Subjective/Assessment/Plan/Objective mapping
 - Single Note → treat as chief complaint + summary; do not invent assessment/plan

Frontend requirements

Referral note screen includes:

- Receiving facility (free text)
- Urgency (Routine/Urgent/Emergency)
- Reason for referral (required)
- Clinical summary
- Investigations
- Treatment given
- Medications, allergies
- Requested action
- Signature line
- Buttons:
 - Generate from selected visit
 - Save Draft

- Save & Sign
- Download PDF / Print / Copy email text
Show:
 - “Source used: visit date + note type”

Backend requirements

- New note type: `REFERRAL_NOTE`
- Store both:
 - structured fields
 - PDF snapshot for downloads
- Audit logs:
 - generated from visit
 - edited
 - signed
 - downloaded / printed / copy email text

Acceptance criteria

- Visit dropdown works
- Referral PDF downloads and is stored in history
- Source is always displayed

4) Appointment Confirmation System + Badges (C/NC/LM + SD/ADV)

What we want (User-facing)

Keep appointment status clean + add simple visual indicators:

Appointment Status (system truth)

- Scheduled
- Confirmed
- Cancelled
- No-show
- Completed

Visual Indicators (badges)

Timing

- SD = Same day booking
- ADV = Advance booking

Confirmation

- C = Confirmed
- NC = Not confirmed (default)
- L/M = Left message (call attempt)

Clinics need these badges on the schedule cards.

Frontend requirements

- On each appointment card show:
 - [SD/ADV] and [C/NC/L/M] and Visit status (Scheduled/In Session etc.)
- Quick actions (from card click):
 - Mark confirmed
 - Left message
 - Set visit status (Scheduled/In Session/Completed/Cancelled)

- Tooltips optional: show last attempt time / method

Backend requirements

- Fields:
 - bookingType: SAME_DAY | ADVANCE
 - confirmationIndicator: C | NC | LM
 - confirmationMethod: SMS/WhatsApp/Phone/In-person
 - timestamps + attempt logs
- Reminder logic (Kenya-friendly):
 - ADV: 24–48h reminder + morning-of reminder
 - SD: same-day reminder only
 - reply “1” confirm, “0” cancel (future messaging integration)

Acceptance criteria

- Badges render correctly
- Quick actions update DB + audit log
- BookingType auto-set based on created date vs appointment date

5) Patient Intake Forms (Medical History + Dental Insurance) → returns into chart

What we want (User-facing)

Patient gets a secure link to fill a form on phone; submission returns into chart.

Security decisions (final)

- Link expires in **24 hours**
- Patient verifies identity using **last 4 digits of phone number**
- Submissions become **Pending Review** (do not auto-write into chart)

Frontend requirements

Staff side

- From Appointment or Patient Profile:
 - Send Medical History Form
 - Send Insurance Form
- Patient chart shows:
 - Intake Forms → Pending Review
 - buttons: Review / Accept / Reject

Patient side (mobile page)

- Enter last 4 digits of phone
- Fill form
- Submit
- Confirmation screen

Backend requirements

- Create `intakeFormInstance` records:
 - `patientId`, `appointmentId(optional)`, `formType`
 - `token`, `expiresAt (24h)`
 - `verification attempts + verifiedAt`
 - `answersJson`
 - `pdfSnapshotUrl`

- status: PENDING REVIEW / ACCEPTED / REJECTED
- When staff accepts:
 - update patient Demographics (insurance fields)
 - store original submission as PDF for audit

Acceptance criteria

- Link expires in 24h
- Phone last-4 validation works
- Pending review workflow works
- Accepted insurance updates Demographics (source of truth)

6) Insurance Claims-Ready Billing (no API integration yet, but future-ready)

What we want (User-facing)

Keep existing billing flow but make it claims-ready:

- Demographics holds insurance info (master record)
- Billing pulls insurance info read-only
- New action: **Prepare Insurance Claim**
- Generate **Claim Pack PDF** for manual submission today
- Later: “one-click” claim submission via connectors (future)

Frontend requirements

- In Patient → Demographics:
 - insurance fields remain the master record (you already have this)
- In Billing:
 - payer can be Patient or Insurance (already present)
 - if Insurance:
 - show insurance summary (read-only)
 - add button: **Prepare Insurance Claim**
- Create a new Claims list view (can be simple table):
 - Claim ID
 - Patient
 - Date
 - Status (Draft/Ready/Submitted Manual/Paid/Rejected)
 - Download Claim Pack

Backend requirements

- Claim Draft object:
 - snapshot insurance details used
 - ICD-10 diagnosis list
 - itemized services + totals
 - attachments references (clinical note, labs, referral)
 - status workflow
- Claim Pack generator:
 - PDF summary + attachments bundle
- Connector layer stubs (backend only for now):
 - EligibilityCheck()
 - PreAuth()

- SubmitClaim()
- CheckStatus()
- Remittance()
(disabled until credentials/APIs are available)

Acceptance criteria

- No disruption to cash billing
- Claim Draft can be created from a bill
- Claim Pack PDF downloads
- Historical claims preserve the snapshot used at submission time

Global requirements across all 6 enhancements

- **Audit trail everywhere** (medical + billing + claims actions)
- Role-based permissions:
 - clinicians vs billing staff vs admin
- Data integrity:
 - Demographics = single source of truth for insurance
 - Claims store snapshots so old claims aren't changed by new demographic edits
- "Assist, don't automate":
 - clinician review required for AI outputs