

Project-404 Requirements, UX Flow & AI (Voice) Draft v1

Date: August 28, 2025

Target MVP kickoff: September 2025

Leads: Dominic (Frontend/UI/UX)

Team: Yusupha (Frontend) Ibrahim & Yaya (Full-stack), Cherno (AI/ML + Web), Abdoulie (DB/AI/Backend)

1) Product Vision

Build a community powered platform to **report, find, and recover missing persons** in The Gambia (and beyond), combining **structured case management, search & alerts**, and **AI assistance** (face + voice + image enhancement) while protecting privacy and preventing misuse.

Primary outcomes

- Faster awareness & wider reach for new cases.
- Better data quality for authorities and families.
- Smart assistance: recognizing faces/voices and improving low quality media.

2) MVP Scope (September–October 2025)

Must have

1. **Auth:** Email/phone login, signup, reset password.
2. **Report Case:** Create case with person details, last seen, contact, photos, and evidence.
3. **Search & Browse:** Filter by age, gender, location, date; view case profile.
4. **Tips:** Submit confidential tips with optional attachments.
5. **Alerts:** Opt in notifications for nearby/new cases.
6. **Basic Admin/Moderator:** Approve/flag/close cases; audit log.
7. **Data Privacy & Safety:** Consent workflow; reporting abuse; watermarking sharables.

Nice to have (Stretch)

- **Face matching (candidate suggestions)** on uploaded photos.
- **Voice module (ingest + clean + embed)** to attach a voiceprint to a case and search similar voices from public submissions.
- **Image enhancement** (denoise, super resolution).
- **Shareable Poster** auto generated per case.

3) Personas & Core User Stories

A) Family/Reporter

- *As a family member*, I can file a report with photos, last seen map, and contact so others can help.
- *As a reporter*, I can update status (new info / found) and receive alerts/messages.
- *As a reporter*, I can upload a **voice sample** (from video/audio) to help AI match.

B) Public Helper

- I can browse/search open cases and submit a **tip** with optional photo/audio.
- I can subscribe to alerts around my location.

C) Moderator/Police Partner

- I can verify/approve new reports, redact sensitive data, and close or escalate cases.
- I can view an audit trail and export a case file.

D) Admin

- I can manage users/roles, rate limits, and integration keys.
- I can configure AI features (enable/disable, thresholds).

4) App Flow (High Level)

1. **Onboarding** → **Auth** (Login/Signup/Guest view).
2. **Dashboard** with 4 quick actions: *Search, Report, Alerts, My Reports*.
3. **Search** (filters) → **Case Profile** → share poster / send tip.
4. **Report Form** → upload photos & optional **voice/audio** → submit → moderation.
5. **Notifications** for cases near you and status changes.
6. **Moderator Console** → review queue → approve/flag/close.

5) ASCII Wireframes (Draft)

5.1 Login

```
+-----+
|   Project-404   |
| Find Them, Bring Them Home |
|-----|
| [ Email / Phone ] |
| [ Password   ]   |
| ( Log In )      |
| Forgot?  Create account |
+-----+
```

5.2 Dashboard

```
+-----+
| Hello, [Name]    |
|-----|
| [ Search Cases ] [ Report ] |
| [ Alerts   ] [ My Reports ] |
+-----+
```

5.3 Search

```
+-----+
| [ Search... ] (Filters) |
| Age | Gender | Date | Radius |
|-----|
| [Img] Name • Age • Last seen |
| [Img] Name • Age • Last seen |
| [Img] Name • Age • Last seen |
+-----+
```

5.4 Report Case

```
+-----+
| Report Missing Person |
|-----|
| Name: [_____]        |
+-----+
```

```

| Age: [__] Gender: [ ] |
| Last Seen: [Map/Address] |
| Date: [__/__/____] |
| Contact: [_____] |
| Photos: [ + Add ] |
| Voice/Audio: [ + Upload/Record ]|
| Notes: [_____] |
| ( Submit for Review ) |
+-----+

```

5.5 Case Profile

```

+-----+
| [ Main Photo ] Status: Missing|
| Name, Age |
| Last seen: Place • Date |
| Contact: xxx-xxx |
|-----|
| ( Share Poster ) ( Send Tip ) |
| ( Listen to Voice Sample ) |
+-----+

```

5.6 Moderator Queue

```

+-----+
| Pending Reports (n) |
|-----|
| [Case #123] Open Flag Approve|
| [Case #124] Open Flag Approve|
+-----+

```

6) Data Model (Draft)

User(id, name, email, phone, role[public,reporter,mod,admin], location_optin, created_at)

Case(id, title, person_name, age, gender, last_seen_place, last_seen_geo, last_seen_time, description, status[missing,found,closed], reporter_id, created_at, updated_at)

Media(id, case_id, type[photo,video,audio,poster], url, metadata, created_at)

VoicePrint(id, case_id, media_id, embedding_vector_ref, quality_score, created_at)

Tip(id, case_id, user_id_nullable, message, attachments[], created_at, status[new,reviewed,escalated])

AlertSubscription(id, user_id, radius_km, home_geo, keywords[], created_at)

AuditLog(id, actor_id, action, target_type, target_id, diff_json, created_at)

7) API Surface (Draft)

GraphQL (preferred)

Queries

cases(filter: CaseFilter, paging: Paging): [Case]

case(id: ID!): Case

Mutations

createCase(input: CreateCaseInput!): Case

uploadMedia(caseld: ID!, file: Upload!, type: MediaType!): Media

createTip(input: CreateTipInput!): Tip

updateCaseStatus(id: ID!, status: CaseStatus!): Case

subscribeAlerts(input: AlertInput!): AlertSubscription

AI hooks

createVoicePrint(caseld: ID!, mediald: ID!): VoicePrint

searchByVoice(file: Upload!): [CaseMatch]

searchByFace(file: Upload!): [CaseMatch]

REST (for uploads/webhooks)

- POST /api/cases/{id}/media
- POST /api/ai/voice/print (case_id, media_id)
- POST /api/ai/voice/search (audio file)
- POST /api/ai/face/search (image file)

8) AI Voice Module — Feature Design

Goal: Allow families to attach a **voice sample** of the missing person (from any video/voice note). The system extracts a **speaker embedding** (voiceprint) to:

- Help match tips/recordings from the public that might contain the same voice.
- Allow *search by voice* (optional, privacy aware).
- Provide a playable cleaned clip on the case profile (if consented).

Pipeline

1. **Ingest:** Upload audio or video → extract audio track.
2. **Quality Check:** Detect speech vs. noise; estimate SNR; require minimum duration (e.g., ≥5s speech).
3. **Cleaning:** Denoise, dereverb, loudness normalize; optional band limit removal.
4. **Diarization** (for multi speaker clips): separate speakers; let the user choose which segment matches the person.
5. **Embedding:** Compute **speaker embedding** (e.g., ECAPA TDNN/x-vector style) → store as vector in **vector DB** (FAISS/pgvector).
6. **Matching:** Cosine similarity search against other embeddings with a calibrated threshold; return candidate cases.
7. **Human in the loop:** Moderators review AI suggestions before anything is shown publicly.
8. **Privacy:** Opt-in from reporter; encrypt at rest; restrict who can run voice searches (mods/LE only by default).

UI Impacts

- **Report Case:** “Upload voice or video (optional)” + guidance (best quality, single speaker if possible).
- **Case Profile:** (Listen to sample) only if consented & approved.
- **Moderator Tooling:** Visual waveform, speaker segments, quality score, (Set primary speaker segment) button.

API Additions

- POST /api/ai/voice/ingest (file → returns media_id + qc metrics)
- POST /api/ai/voice/embed (media_id → voiceprint_id, quality_score)
- POST /api/ai/voice/match (file → [case_id, score])

Storage

- Raw file (S3/minio)
- Processed WAV (mono 16kHz)
- Embeddings table + vector index

- Strict access policies & auditing.

Risks & Mitigations

- **False matches** → conservative thresholds + human review.
- **Impersonation/abuse** → watermarking, upload provenance, rate limiting, report-abuse flow.
- **Privacy** → consent capture, revocation, encryption, access logs.

9) AI Face/Image Module — Feature Design (Preview)

- **Image QC** (blur, occlusion), **enhancement** (super-res, denoise) for poster quality.
- **Face detection & embeddings** (vector DB); **candidate matches** with moderator gate.
- **Age progression** (later): only by opt-in; clearly marked as AI generated with disclaimers.

10) Non-Functional Requirements

- **Security:** OWASP ASVS L1+; 2FA (admin/mod); strict RBAC; rate limits; input validation; audit logging.
- **Privacy:** Explicit consent; right to removal; PII minimization; redact sensitive fields for public view.
- **Performance:** p95 < 800ms for common queries; uploads < 50MB (MVP).
- **Availability:** 99% MVP; daily backups; signed URLs for media.
- **Compliance:** Clear ToS/Privacy Policy; data retention policy; parental/guardian consent for minors.

11) Initial Milestones & Timeline

Now → Aug 31, 2025

- Finalize requirements & wireframes v1 (this doc).

September 1–30, 2025 (M1: MVP Build, Sprint 1–2)

- Scaffold backend (Flask + GraphQL); DB schema (MongoDB + MySQL).
- Frontend shell (Flutter or React Native/React Web for prototype).
- Case report, search, profile, tips; basic mod console.

October 1–31, 2025 (M2: AI Hooks & Pilot)

- Integrate voice ingest → embedding → secure storage.

- Image QC/enhancement pipeline.
- Pilot with limited users; feedback; bugfixes.

November 2025 (M3: Field Test)

- Add alerts/notifications; poster generator; refine mod tools.
- Prepare for partnerships/outreach.

12) GitHub Projects — Starter Ticket Backlog

Frontend/UI (Domzzy, Yusupha)

- Create wireframes in Figma from this doc
- Implement screens: Auth, Dashboard, Search, Report, Case Profile
- Poster share view; Alerts opt-in modal
- Bootstrap Next.js + Tailwind skeleton.
- Implement CaseForm (Name, Age, Photo upload).
- Implement CaseList (basic table).
- Implement CaseDetails page (static fetch).
- API integration with backend endpoints.
- Setup lint + prettier config.
- Add **responsive design** for mobile/desktop.
- Add error & loading states (spinners, error banners).
- Add role based UI logic (Admin vs Public).

Backend/API (Ibrahim, Yaya)

- Setup **Flask skeleton** with virtualenv/venv and requirements.txt.
- Implement **GraphQL endpoints** for:
 - Auth (email/phone login/signup, JWT, RBAC)
 - Case CRUD (Create, Read, Update, Delete)
 - Media upload (photos, videos, audio, posters via signed URLs)
 - Tips submission & moderation
 - Alerts subscription & management
- Implement **REST endpoints** for media uploads/webhooks if needed.
- Integrate **MongoDB + MySQL** for structured and flexible storage.
- Add **basic logging + error handling**.
- Write **unit tests** with pytest for key endpoints.
- Setup **feature branch workflow + CI/CD** integration.

AI/ML (Cherno)

- Research & prototype **face recognition pipeline** (DeepFace / FaceNet / OpenCV).
- Implement **image enhancement module** for low quality photos (denoise, super-resolution, color correction).
- Implement **disguise/augmentation pipeline** (glasses, hats, beards, hairstyles) for candidate matching.
- Develop **age-progression module** for long term missing persons (optional, clearly labeled as AI generated).
- Integrate **3D reconstruction pipeline** (from photo → 3D mesh preview, optional MVP).
- Expose **AI services as Flask endpoints** for backend integration (GraphQL mutations/hooks).
- Build **embedding storage & retrieval** (vector DB: FAISS / pgvector) for faces and optionally voices.
- Write **basic evaluation scripts** (accuracy, false positives, confidence scores).
- Collaborate with backend for **secure storage & access control**.

(Abdoulie) Database / Backend Support

- Design **MongoDB + MySQL schemas** for cases, users, media, tips, alerts, voiceprints.
- Implement **database migrations** and seeding scripts.
- Build **GraphQL resolvers** for backend API endpoints.
- Integrate **voiceprint storage & search** (linking Cherno's embeddings to database).
- Implement **audit logging** and moderation support tables.
- Optimize queries for **search performance** (indexing, pagination).
- Collaborate with Cherno on **AI/ML data pipelines** (face/voice embeddings storage).
- Set up **DB backup, recovery, and security policies**.
- Write **integration tests** for backend + database interactions.