TerraFerm Africa (TFA) — Farm Operating System (FOS) PRD v1.0

AI-Assisted Remote Farm Operations Monitoring & Command Center

Product Vision: "*Your boss is data*" — *Building a digital twin of TFA's cactus nursery operations that transforms unstructured field data into real-time intelligence for ExCo decision-making*.

EXECUTIVE SUMMARY

Problem Statement

TFA's 13 ha Steelpoort pilot nursery generates critical operational data through WhatsApp messages, voice notes, images, and field observations from low-literacy workers in rural Limpopo. This unstructured data cannot inform rapid decision-making at the ExCo level, creating a disconnect between ground truth and strategic execution.

Solution

A lightweight, AI-powered Farm Operating System that:

1. **Ingests** unstructured field data via WhatsApp/mobile
2. **Transforms** it into structured metrics using Claude AI
3. **Visualizes** operations in real-time via SpaceX Mission Control-inspired dashboard
4. **Enables** data-driven decisions: "You have no boss, your boss is data"

Success Metrics

- **Operational:** 100% field activity captured within 24 hours
- **Efficiency:** Reduce ExCo reporting time from 3 days → 15 minutes
- **Accuracy:** <5% variance between planned vs. actual planting rates
- **Adoption:** 80%+ daily active usage by field teams within 4 weeks

---

PART 1: STRATEGIC CONTEXT

Design Philosophy Alignment (Musk Methodology)

Principle	FOS Implementation
First-Principles	Start from ground truth: Field Activity → Data Capture → AI Processing → Insights →

Principle	FOS Implementation
Thinking	Decisions
Vertical Integration	Unified system: WhatsApp → Database → AI → Dashboard → Field App
Radical Simplicity	One tap to log work. Zero training for workers. Instant insights for ExCo.
Ambitious Scale	Design for 13 ha pilot, scale to 1,700 ha in 28 months
Engineering-Centric	Surface metrics that matter: plants/day, R/ha, survival %, labor output
Real-Time Data	SpaceX Mission Control aesthetic: Live operations, zero lag

TFA Brand Identity Integration

3. Color System

Brand Colors

Primary — "Cactus Green"

```css
--tfa-primary: #2B7035;
--tfa-primary-light: #3D8F47;
--tfa-primary-dark: #1F5227;
--tfa-primary-50: rgba(43, 112, 53, 0.1);    /* 10% opacity */
--tfa-primary-100: rgba(43, 112, 53, 0.2);   /* 20% opacity */
```

Swatch	Hex	Usage
🟢	#2B7035	Logo mark, success states, checkmarks
🟢	#3D8F47	Hover states, highlights
🟢	#1F5227	Dark accents, gradients

Secondary — "TerraFerm Blue"

```css
--tfa-secondary: #025373;
--tfa-secondary-light: #0A6B8F;
--tfa-secondary-dark: #094C6A;
```

Swatch	Hex	Usage
🔵	#025373	Headlines, slide titles, contact names
🔵	#0A6B8F	Light variant
🔵	#094C6A	Dark variant

Accent — "Teal Cyan"

```css
--tfa-accent: #01E3C2;
--tfa-accent-light: #33EACD;
--tfa-accent-dark: #01AB93;
--tfa-accent-muted: #018F7B;
```

Swatch	Hex	Usage
🔷	#01E3C2	Metrics, highlights, CTAs, active states (dark mode)
🔷	#01AB93	Accent text on light backgrounds
🔷	#018F7B	Muted accent

Tertiary — "Earth Gold"

```css
--tfa-tertiary: #A37A51;
--tfa-tertiary-light: #B8936A;
--tfa-tertiary-dark: #8B6644;
```

Swatch	Hex	Usage
🟡	#A37A51	Underlines, subtitles, warning accent
🟡	#B8936A	Light variant
🟡	#8B6644	Dark variant

Semantic Colors

```css
--tfa-error: #D94848;
--tfa-error-dark: #C03030;
--tfa-warning: #D35230;
--tfa-warning-dark: #B84520;
--tfa-success: #2B7035;      /* Same as primary */
--tfa-info: #025373;         /* Same as secondary */
```

State	Color	Hex	Usage
Error		#D94848	Pain points, negative metrics, PDF button
Warning		#D35230	Caution states, PPTX button
Success		#2B7035	Positive outcomes, checkmarks
Info		#025373	Informational

Surface Colors

Dark Mode (Default)

```css
html.theme-dark {
  --tfa-bg-primary: #0A0F0A;     /* Base background */
  --tfa-bg-secondary: #141A14;   /* Cards, panels */
  --tfa-bg-tertiary: #1E261E;    /* Elevated surfaces, inputs */
  --tfa-bg-elevated: #283028;    /* Highest elevation */

  --tfa-text-primary: #FFFFFF;   /* Headlines, body text */
  --tfa-text-secondary: #C8D0C8; /* Secondary text */
  --tfa-text-muted: #9AA89A;     /* Muted text, labels */

  --tfa-border: #2A352A;         /* Card borders */
  --tfa-border-light: #3A453A;   /* Subtle borders */
}
```

Light Mode

```css
```

```
html.theme-light {
  --tfa-bg-primary: #F5F7F5;     /* Base background */
  --tfa-bg-secondary: #FFFFFF;   /* Cards, panels */
  --tfa-bg-tertiary: #E8EDE8;    /* Elevated surfaces */
  --tfa-bg-elevated: #FFFFFF;    /* Highest elevation */

  --tfa-text-primary: #1A1F1A;   /* Headlines, body text */
  --tfa-text-secondary: #3A4A3A; /* Secondary text */
  --tfa-text-muted: #4A5A4A;     /* Muted text, labels */

  --tfa-border: #D0D8D0;         /* Card borders */
  --tfa-border-light: #E0E8E0;   /* Subtle borders */

  --tfa-accent: #01AB93;         /* Darker accent for contrast */
}
```

Gradients

```
css

/* Primary gradient (Cactus Green) */
--tfa-gradient-primary: linear-gradient(135deg, #2B7035 0%, #1F5227 100%);

/* Accent gradient (Teal) */
--tfa-gradient-accent: linear-gradient(135deg, #01E3C2 0%, #01AB93 100%);

/* Hero/Background gradient (Dark mode) */
--tfa-gradient-hero: linear-gradient(180deg, #0A0F0A 0%, #141A14 50%, #1E261E 100%);

/* Energy flow gradient (Brand spectrum) */
--tfa-gradient-energy: linear-gradient(90deg, #2B7035 0%, #01AB93 33%, #025373 66%, #A37A51 100%);
```

Typography:

- Primary: Inter (clean, readable on mobile)
- Data/Metrics: JetBrains Mono (precision, engineering feel)

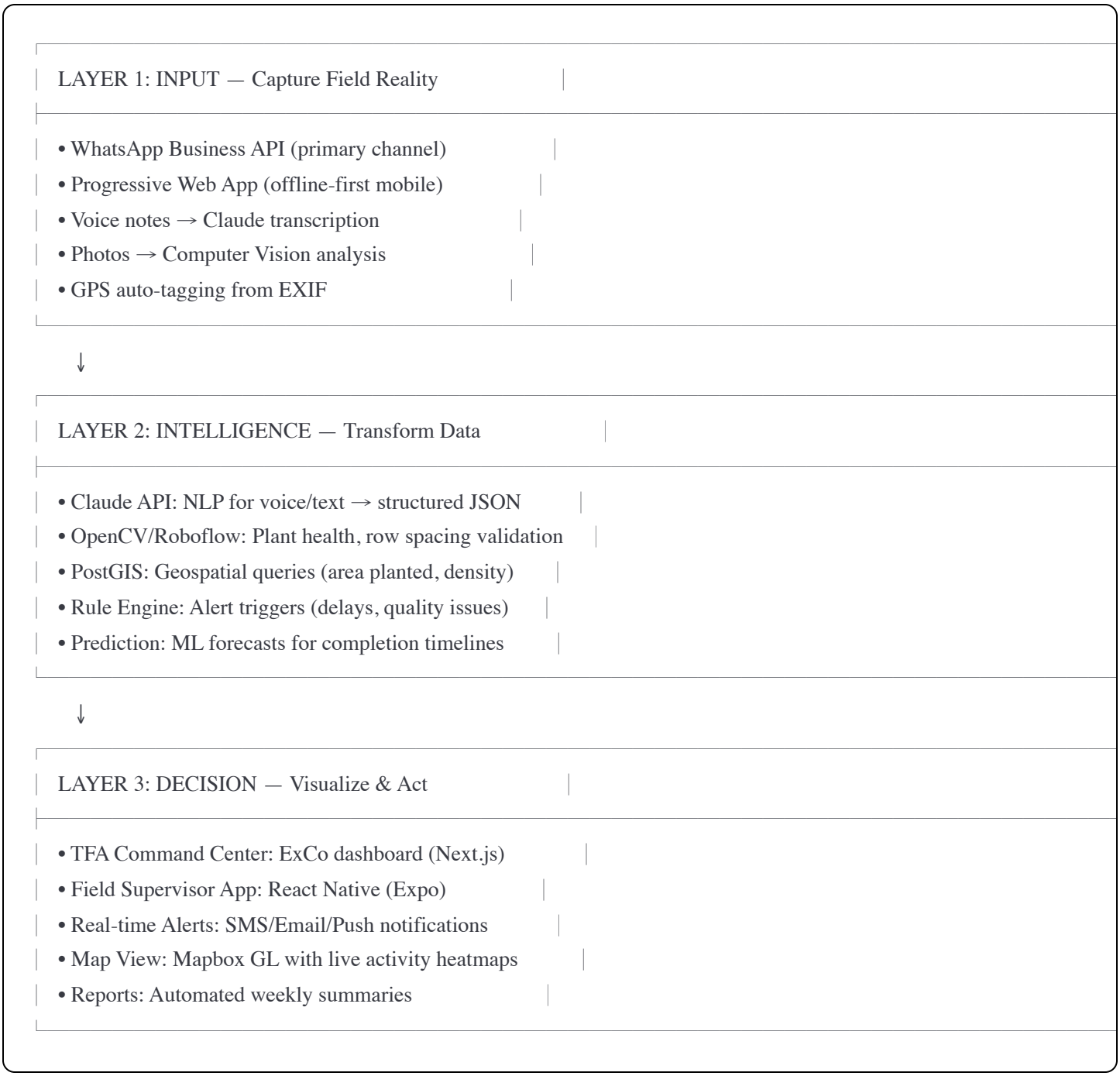
Visual Language:

- Dark mode by default (field visibility + energy focus)
- Information density balanced with clarity
- Neumorphic cards for hierarchy

- Glow effects on critical alerts (teal/gold shadows)

PART 2: SYSTEM ARCHITECTURE

Three-Layer Conceptual Model



Tech Stack (MVP)

Frontend:

- **Dashboard:** Next.js 14 (App Router), Tailwind CSS, shadcn/ui
- **Mobile:** Progressive Web App (PWA) with offline support

- **Maps:** Mapbox GL JS (free tier: 50k loads/month)

Backend:

- **Database:** Supabase (PostgreSQL + PostGIS + Auth + Storage)
- **API:** Next.js API Routes (serverless functions)
- **AI Processing:** Anthropic Claude API (Sonnet 4)
- **Image Analysis:** OpenCV + Roboflow (optional Phase 2)

Infrastructure:

- **Hosting:** Vercel (frontend), Supabase Cloud (backend)
- **File Storage:** Supabase Storage (images, voice notes)
- **Auth:** Supabase Auth (magic links for simplicity)

Integrations:

- **WhatsApp:** Twilio WhatsApp Business API (R1,500/month)
- **Weather:** OpenWeatherMap API (free tier)
- **Notifications:** Supabase Realtime + Push API

PART 3: FEATURE SPECIFICATIONS

3.1 TFA Command Center (ExCo Dashboard)

Primary View: Operations Overview

TFA STEELPOORT NURSERY — OPERATIONS COMMAND CENTER
Week 4, 2026 | Last Updated: 26 Jan 14:23 SAST

🎯 CRITICAL METRICS (Week-over-Week)

Metric	Current	Target	Δ
Area Planted	1.2 ha	2.0 ha	-40% ⚠️
Planting Rate	850/day	1,200/day	-29% ⚠️
Plant Density	11.8k/ha	12k/ha	-1.7% ✅
Survival Rate	94%	95%	-1% ✅
Cost/Hectare	R182,400	R179,400	+1.7% ⚠️
Labor Efficiency	283/day	400/day	-29% ⚠️

📍 GEOSPATIAL VIEW

[Interactive Mapbox map showing:]
- Plot boundaries (GeoJSON polygons)
- Color-coded by completion: Green (done), Yellow (in-progress), Gray (pending)
- Heatmap layer: Planting density
- Markers: Recent activities (last 48h)
- Weather overlay: Current conditions + 7-day forecast

Feature Breakdown:

Feature	Description	Priority
Live KPI Cards	Real-time metrics with trend indicators (↑↓) and color coding	P0
Interactive Map	Mapbox GL with plot boundaries, activity markers, density heatmap	P0
Activity Timeline	Reverse-chronological feed of field activities (planting, clearing, inspections)	P0
Alert Center	System-generated alerts (delays, quality issues, resource shortages)	P0
Labor Analytics	Team productivity, attendance, cost analysis	P1
Resource Tracker	Cladode inventory, equipment status, compost usage	P1
Weather Dashboard	7-day forecast, rainfall totals, planting windows	P1
Photo Gallery	AI-tagged image library (searchable by plot, date, activity type)	P1
Weekly Reports	Auto-generated PDF summaries for ExCo meetings	P2
Predictive Analytics	ML-based completion forecasts, budget variance projections	P2

3.2 Field Worker App (PWA)

Design Constraints:

- Target users: Low-literacy, rural workers (no prior smartphone experience)
- Connectivity: Intermittent 3G/4G, must work offline
- Device: Budget Android phones (3-4 year old models)

Core Features:

1. Daily Check-In (Voice-First)

```
┌─────────────────────────────────┐
│ Good Morning, Ansi! 🌄          │
│                                 │
│ [🎤 Record Daily Update]        │ ← Big button, auto-transcribe
│                                 │
│ Today's Goal: Plant 400         │
│ Current Progress: 147 ✅        │
│                                 │
│ [📷 Take Photos]                │ ← Auto-upload when online
│ [✅ Mark Tasks Complete]        │ ← Simple checklist
└─────────────────────────────────┘
```

2. Photo Capture with Auto-Tagging

- One-tap photo capture
- GPS auto-extracted from EXIF
- Prompt: "What are you photographing?" (voice response)
- Offline queue, auto-upload when connected

3. Task Checklist

- Simple checkboxes for assigned daily tasks
- Visual progress bar (gamification)
- Push notification reminders (9 AM, 3 PM)

4. Voice Notes

- Big red button: "Report an Issue"
- Auto-sent to supervisor + transcribed by Claude
- Common prompts: "Describe what you see", "What's the problem?"

3.3 AI Processing Pipeline

WhatsApp → Claude NLP Flow

```
python
```

```
# Pseudo-code for AI extraction

INPUT: WhatsApp message from field
"Hi Nick, planted 400 cladodes in Plot 2A today.
Had 6 workers. Rows look good but spacing a bit tight.
Weather was hot, need more water tomorrow."

CLAUDE PROCESSING:
1. Entity extraction:
   - Activity: "planting"
   - Quantity: 400 cladodes
   - Location: "Plot 2A"
   - Labor: 6 workers
   - Date: Today (infer from timestamp)

2. Issue detection:
   - Problem: "spacing tight"
   - Severity: Medium
   - Recommendation: "Adjust row spacing"

3. Resource needs:
   - Item: "Water"
   - Urgency: High (next day)
   - Quantity: Not specified (ask follow-up)

OUTPUT (Structured JSON):
{
  "activity_type": "planting",
  "plot_id": "2A",
  "cladodes_planted": 400,
  "workers": 6,
  "date": "2026-01-26",
  "issues": [{
    "type": "spacing_error",
    "severity": "medium",
    "description": "Rows too close together",
    "action_required": "Adjust spacing in remaining plots"
  }],
  "resources_needed": [{
    "item": "water",
    "urgency": "high",
    "requested_by": "field_team"
  }],
```

```
  "weather_conditions": "hot",
  "sentiment": "concerned",
  "confidence": 0.92
}
```

Image Analysis Flow

```
python

# Computer Vision pipeline

INPUT: Photo of planted row

ANALYSIS:
1. Row spacing measurement (OpenCV edge detection)
2. Plant count estimation (YOLO object detection)
3. Plant health scoring (color analysis: green ratio)
4. Weed pressure assessment (texture analysis)

OUTPUT:
{
  "row_spacing_cm": 248,  # Target: 250cm
  "plants_visible": 23,
  "plant_health_score": 0.87,  # 0-1 scale
  "weed_pressure": "moderate",
  "recommended_action": "Herbicide treatment in 7 days",
  "confidence": 0.78
}
```

3.4 Alert System

Trigger Rules:

Alert Type	Condition	Severity	Action
Planting Behind Schedule	Actual rate < 80% of target for 2+ days	High	SMS to COO + WhatsApp supervisor
Plant Density Variance	<11,000 or >13,000 plants/ha	Medium	Flag for inspection
Survival Rate Drop	<90% survival in any plot	High	Agronomist alert + photo review
Cost Overrun	Actual cost >105% of R179,400/ha	Medium	Email to Finance + COO
Quality Issue	Row spacing >10% off target	Medium	Supervisor review required
Resource Shortage	Cladode inventory <3 days supply	High	Procurement alert
Weather Risk	>30mm rain forecast in 48h	Low	Delay planting recommendation

---

PART 4: DATA MODEL

Database Schema (PostgreSQL + PostGIS)

Core Tables:

sql

```sql
-- 1. PLOTS (GIS-enabled)
CREATE TABLE plots (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  plot_code VARCHAR(10) UNIQUE NOT NULL, -- e.g., "2A", "3B"
  geometry GEOMETRY(POLYGON, 4326), -- GeoJSON polygon
  area_ha DECIMAL(10,2),
  planned_density INTEGER, -- target plants/ha
  status VARCHAR(20), -- 'pending', 'clearing', 'planting', 'completed'
  start_date DATE,
  target_completion_date DATE,
  actual_completion_date DATE,
  created_at TIMESTAMPTZ DEFAULT NOW()
);

-- 2. ACTIVITIES (Field operations log)
CREATE TABLE activities (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  plot_id UUID REFERENCES plots(id),
  activity_type VARCHAR(50), -- 'site_clearing', 'planting', 'inspection', 'weeding'
  activity_date DATE NOT NULL,
  created_at TIMESTAMPTZ DEFAULT NOW(),

  -- Planting-specific
  cladodes_planted INTEGER,
  workers_count INTEGER,
  hours_worked DECIMAL(5,2),

  -- Measurements
  area_covered_ha DECIMAL(10,4),
  row_spacing_cm INTEGER,
  plant_spacing_cm INTEGER,
  actual_density INTEGER,

  -- Metadata
  reported_by VARCHAR(100), -- Name or phone number
  report_method VARCHAR(20), -- 'whatsapp', 'app', 'manual'
  gps_location GEOGRAPHY(POINT, 4326),
  notes TEXT,

  -- AI extraction metadata
  ai_extracted BOOLEAN DEFAULT FALSE,
  ai_confidence DECIMAL(3,2),
  source_message_id VARCHAR(100) -- Link to original WhatsApp/media
```

```
);

-- 3. FIELD_OBSERVATIONS (Issues, quality checks)
CREATE TABLE field_observations (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  activity_id UUID REFERENCES activities(id),
  plot_id UUID REFERENCES plots(id),
  observation_date DATE NOT NULL,
  created_at TIMESTAMPTZ DEFAULT NOW(),

  -- Issue tracking
  observation_type VARCHAR(50), -- 'quality_issue', 'pest', 'weed', 'spacing_error'
  severity VARCHAR(20), -- 'low', 'medium', 'high', 'critical'
  description TEXT,
  action_required TEXT,
  status VARCHAR(20) DEFAULT 'open', -- 'open', 'in_progress', 'resolved'
  resolved_at TIMESTAMPTZ,

  -- Media
  photos JSONB, -- Array of Supabase Storage URLs
  voice_notes JSONB,

  -- AI analysis
  ai_detected BOOLEAN DEFAULT FALSE,
  ai_analysis JSONB -- Full Claude response
);

-- 4. PLANT_HEALTH (Tracking survival and growth)
CREATE TABLE plant_health (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  plot_id UUID REFERENCES plots(id),
  assessment_date DATE NOT NULL,
  created_at TIMESTAMPTZ DEFAULT NOW(),

  -- Health metrics
  plants_alive INTEGER,
  plants_dead INTEGER,
  survival_rate DECIMAL(5,2), -- Percentage
  avg_height_cm DECIMAL(5,2),
  health_score DECIMAL(3,2), -- 0-1 from CV analysis

  -- Issues
  pest_detected BOOLEAN,
  disease_detected BOOLEAN,
```

```
  weed_pressure VARCHAR(20), -- 'low', 'moderate', 'high'

  -- Source
  assessed_by VARCHAR(100),
  photos JSONB
);

-- 5. LABOR (Team tracking)
CREATE TABLE labor_logs (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  work_date DATE NOT NULL,
  created_at TIMESTAMPTZ DEFAULT NOW(),

  -- Worker details
  worker_name VARCHAR(100),
  worker_phone VARCHAR(20),
  role VARCHAR(50), -- 'planter', 'digger', 'supervisor'

  -- Work done
  activity_id UUID REFERENCES activities(id),
  hours_worked DECIMAL(5,2),
  tasks_completed JSONB, -- Array of tasks
  output_quantity INTEGER, -- e.g., plants planted

  -- Attendance
  check_in_time TIME,
  check_out_time TIME,
  present BOOLEAN DEFAULT TRUE
);

-- 6. RESOURCES (Inventory tracking)
CREATE TABLE resource_inventory (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  resource_type VARCHAR(50), -- 'cladode', 'compost', 'water', 'equipment'
  quantity DECIMAL(10,2),
  unit VARCHAR(20), -- 'units', 'kg', 'liters', 'hours'
  location VARCHAR(100),

  -- Transactions
  transaction_type VARCHAR(20), -- 'received', 'used', 'transferred'
  transaction_date DATE NOT NULL,
  related_activity_id UUID REFERENCES activities(id),
  notes TEXT,
```

```
  created_at TIMESTAMPTZ DEFAULT NOW()
);

-- 7. WEATHER (Environmental data)
CREATE TABLE weather_data (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  recorded_at TIMESTAMPTZ DEFAULT NOW(),

  -- Location
  latitude DECIMAL(10,6),
  longitude DECIMAL(10,6),

  -- Conditions
  temperature_c DECIMAL(4,1),
  humidity_percent INTEGER,
  rainfall_mm DECIMAL(5,2),
  wind_speed_kmh DECIMAL(5,2),
  conditions VARCHAR(50), -- 'clear', 'cloudy', 'rain'

  -- Source
  source VARCHAR(50), -- 'openweathermap', 'manual'
  is_forecast BOOLEAN DEFAULT FALSE,
  forecast_date DATE
);

-- 8. ALERTS (System notifications)
CREATE TABLE alerts (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  created_at TIMESTAMPTZ DEFAULT NOW(),

  -- Alert details
  alert_type VARCHAR(50),
  severity VARCHAR(20), -- 'low', 'medium', 'high', 'critical'
  title VARCHAR(200),
  description TEXT,

  -- Context
  related_plot_id UUID REFERENCES plots(id),
  related_activity_id UUID REFERENCES activities(id),

  -- Status
  status VARCHAR(20) DEFAULT 'active', -- 'active', 'acknowledged', 'resolved'
  acknowledged_by VARCHAR(100),
  acknowledged_at TIMESTAMPTZ,
```

```
  resolved_at TIMESTAMPTZ,

  -- Delivery
  notification_sent BOOLEAN DEFAULT FALSE,
  notification_channels JSONB -- ['sms', 'email', 'push']
);

-- 9. WHATSAPP_MESSAGES (Raw message log)
CREATE TABLE whatsapp_messages (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  received_at TIMESTAMPTZ DEFAULT NOW(),

  -- Message metadata
  from_number VARCHAR(20),
  message_id VARCHAR(100) UNIQUE,
  message_type VARCHAR(20), -- 'text', 'image', 'audio', 'video'

  -- Content
  body TEXT,
  media_url VARCHAR(500),
  media_content_type VARCHAR(50),

  -- Processing
  processed BOOLEAN DEFAULT FALSE,
  processed_at TIMESTAMPTZ,
  extracted_data JSONB, -- Claude extraction result
  linked_activity_id UUID REFERENCES activities(id),

  -- Storage
  media_stored_path VARCHAR(500) -- Supabase Storage path
);
```

Key Indexes:

```
sql
```

```
CREATE INDEX idx_activities_plot_date ON activities(plot_id, activity_date);
CREATE INDEX idx_activities_type ON activities(activity_type);
CREATE INDEX idx_observations_status ON field_observations(status);
CREATE INDEX idx_labor_date ON labor_logs(work_date);
CREATE INDEX idx_alerts_severity_status ON alerts(severity, status);
CREATE INDEX idx_whatsapp_processed ON whatsapp_messages(processed);

-- GIS indexes
CREATE INDEX idx_plots_geometry ON plots USING GIST(geometry);
CREATE INDEX idx_activities_location ON activities USING GIST(gps_location);
```

PART 5: API SPECIFICATION

REST Endpoints (Next.js API Routes)

Authentication: Supabase Auth (JWT tokens)

Base URL: `https://fos.terraferm.africa/api`

1. Dashboard Data

```
GET /api/dashboard/overview
```

Returns current metrics for Command Center homepage.

Response:

```
json
```

```
{
  "summary": {
    "total_area_planted_ha": 1.2,
    "total_area_target_ha": 13.0,
    "planting_progress_percent": 9.2,
    "current_planting_rate_per_day": 850,
    "target_planting_rate_per_day": 1200,
    "avg_plant_density_per_ha": 11800,
    "overall_survival_rate_percent": 94,
    "cost_per_hectare_actual": 182400,
    "cost_per_hectare_budget": 179400,
    "labor_productivity_per_worker": 283
  },
  "kpis": [
    {
      "metric": "Area Planted",
      "current": "1.2 ha",
      "target": "2.0 ha",
      "delta_percent": -40,
      "trend": "down",
      "status": "warning"
    }
    // ... more KPIs
  ],
  "recent_activities": [
    {
      "id": "uuid",
      "activity_type": "planting",
      "plot_code": "2A",
      "date": "2026-01-26",
      "description": "Planted 400 cladodes with 6 workers",
      "status": "completed"
    }
  ],
  "active_alerts": [
    {
      "id": "uuid",
      "severity": "high",
      "title": "Planting Behind Schedule",
      "description": "Current rate 29% below target for 3 consecutive days"
    }
  ],
```

```
  "last_updated": "2026-01-26T14:23:00Z"
}
```

2. Plot Management

```
GET /api/plots
GET /api/plots/:id
POST /api/plots
PUT /api/plots/:id
```

POST /api/plots Example:

```
json

{
  "plot_code": "3C",
  "geometry": {
    "type": "Polygon",
    "coordinates": [[[28.123, -24.567], ...]]
  },
  "area_ha": 0.5,
  "planned_density": 12000,
  "target_completion_date": "2026-02-15"
}
```

3. Activity Logging

```
POST /api/activities
GET /api/activities?plot_id=uuid&date=2026-01-26
PUT /api/activities/:id
```

POST /api/activities Example:

```
json
```

```
{
  "plot_id": "uuid",
  "activity_type": "planting",
  "activity_date": "2026-01-26",
  "cladodes_planted": 400,
  "workers_count": 6,
  "hours_worked": 8,
  "reported_by": "+27123456789",
  "report_method": "whatsapp",
  "notes": "Rows look good, spacing a bit tight"
}
```

4. WhatsApp Webhook (Inbound)

```
POST /api/webhooks/whatsapp
```

Receives Twilio webhook, processes with Claude, creates activity/observation.

Flow:

1. Receive message
2. Store in `whatsapp_messages` table
3. Send to Claude API for extraction
4. Create `activities` and/or `field_observations` records
5. Trigger alerts if needed
6. Return 200 OK to Twilio

5. AI Processing

```
POST /api/ai/process-message
POST /api/ai/analyze-image
```

POST /api/ai/process-message:

```
json
```

```
{
  "message": "Planted 400 in Plot 2A, 6 workers, spacing tight",
  "from": "+27123456789",
  "timestamp": "2026-01-26T10:30:00Z"
}
```

Response:

```
json

{
  "success": true,
  "extracted_data": {
    "activity_type": "planting",
    "plot_id": "2A",
    "cladodes_planted": 400,
    "workers": 6,
    "issues": [{
      "type": "spacing_error",
      "severity": "medium"
    }]
  },
  "confidence": 0.92,
  "activity_id": "uuid"
}
```

6. Alerts

```
GET /api/alerts?status=active
POST /api/alerts/:id/acknowledge
POST /api/alerts/:id/resolve
```

7. Reports

```
GET /api/reports/weekly?start_date=2026-01-20
GET /api/reports/export?format=pdf
```

PART 6: IMPLEMENTATION ROADMAP

Phase 1: Foundation (Weeks 1-2)

Deliverables:

- [ ] Supabase project setup + schema deployment
- [ ] Next.js dashboard scaffolding with TFA brand theme
- [ ] Basic API routes (CRUD for plots, activities)
- [ ] WhatsApp webhook → database flow (no AI yet)
- [ ] Simple map view with static plot boundaries

Success Criteria:

- Can manually create plots and log activities via API
- Dashboard displays basic metrics from database
- WhatsApp messages stored (even if not processed)

Phase 2: Intelligence (Weeks 3-4)

Deliverables:

- [ ] Claude API integration for message processing
- [ ] Automated activity extraction from WhatsApp
- [ ] Alert rule engine (basic triggers)
- [ ] Weather API integration
- [ ] Photo upload to Supabase Storage

Success Criteria:

- WhatsApp message "Planted 400 in Plot 2A" → Auto-creates activity record
- Dashboard shows real-time alerts
- Weather data displayed on map

Phase 3: Field App (Weeks 5-6)

Deliverables:

- [ ] PWA with offline support
- [ ] Voice note capture + upload
- [ ] Photo capture with GPS tagging
- [ ] Task checklist UI
- [ ] Push notifications

Success Criteria:

- Workers can log work via app without internet
- Data syncs when online
- Supervisors receive push alerts

Phase 4: Polish (Weeks 7-8)

Deliverables:

- [ ] Computer vision for plant health (OpenCV)
- [ ] Predictive analytics (completion forecasts)
- [ ] Weekly PDF report generation
- [ ] Performance optimization
- [ ] User training materials

Success Criteria:

- CV model scores plant health with >70% accuracy
- ExCo receives automated weekly reports
- Dashboard loads <2 seconds

PART 7: METRICS & KPIs

Product Metrics

Metric	Target	Measurement
Data Capture Rate	>95% of field activities logged within 24h	Daily count: logged activities / planned activities
AI Accuracy	>85% correct extraction from messages	Manual validation sample (20 msgs/week)
Dashboard Usage	5+ ExCo sessions per week	Google Analytics events
Field App DAU	80% of workers use daily	Active users / total field staff
Alert Response Time	<4 hours average acknowledgment	Time from alert creation to acknowledgment

Operational Impact Metrics

Metric	Baseline (Pre-FOS)	Target (Post-FOS)
Reporting Lag	3 days (manual WhatsApp review)	15 minutes (real-time dashboard)
ExCo Decision Time	5 days (wait for weekly meeting)	Same-day (data-driven alerts)
Cost Variance	Unknown until post-completion audit	<5% (real-time tracking)
Planting Accuracy	±15% density variance	<5% (GPS validation)

PART 8: DESIGN SPECIFICATIONS

Dashboard Screens (Detailed)

Screen 1: Operations Overview

[TFA Logo] STEELPOORT OPERATIONS [User: Nick] [⚙]

🎯 KEY METRICS — Week 4, 2026

[Card: Area Planted] [Card: Planting Rate]
1.2 / 13 ha 850 / 1,200 per day
9.2% ——— 100% 71% ——— 100%
↓ 40% vs target ⚠ ↓ 29% below target ⚠

[Card: Plant Density] [Card: Survival Rate]
11,800 / 12,000 per ha 94% / 95% target
98% ——— 100% 99% ——— 100%
↓ 1.7% variance ✅ ↓ 1% below target ✅

🚨 ACTIVE ALERTS (2)

[🔴 HIGH] Planting Behind Schedule
Current rate 29% below target for 3 days
Recommend: Add 2 workers or extend hours
[Acknowledge] [View Details]

[🟡 MEDIUM] Row Spacing Variance in Plot 3B
Detected 248cm avg (target: 250cm)
Action: Supervisor inspection required
[Acknowledge] [View Details]

📍 FIELD MAP [🗺 Satellite]

[Interactive Mapbox map showing:]
• 13 ha boundary (white outline)
• Plots color-coded:
✅ Green (completed)
🟨 Yellow (in-progress)
⬜ Gray (pending)
• Activity markers (last 48h)

• Weather overlay (current conditions)
• Click plot → Show details sidebar

WEEKLY TRENDS [Jan 20-26]

[Line chart: Daily planting rate]
Target: ---- 1,200 plants/day
Actual: ———— (fluctuating 600-950)

[Bar chart: Labor productivity]
Plants per worker per day (by day of week)

Screen 2: Plot Detail View

← Back to Overview PLOT 2A DETAILS

📋 PLOT INFO

Code: 2A Status: In Progress 🟨
Area: 0.5 ha Progress: 80% complete
Started: Jan 20, 2026 Target: Jan 28, 2026
Density: 11,800/ha Budget: R89,700

📈 PERFORMANCE

Planted: 5,900 / 6,000 cladodes
Survival: 94% (5,546 alive)
Cost: R71,760 / R89,700 (80% of budget) ✅
Days Behind Schedule: 0 ✅

🔍 RECENT ACTIVITIES (Last 7 Days)

Jan 26 | Planting | 400 cladodes | 6 workers | ✅
Jan 25 | Planting | 850 cladodes | 6 workers | ✅
Jan 24 | Inspection | Quality check | Terence | ⚠️ (spacing)
Jan 23 | Planting | 950 cladodes | 7 workers | ✅
[View All Activities]

📷 PHOTOS (12) [View Gallery]

[Thumbnail 1] [Thumbnail 2] [Thumbnail 3] [Thumbnail 4]
Jan 26 Jan 26 Jan 25 Jan 24
Planting Row spacing Team photo Quality issue

🗺️ MAP VIEW

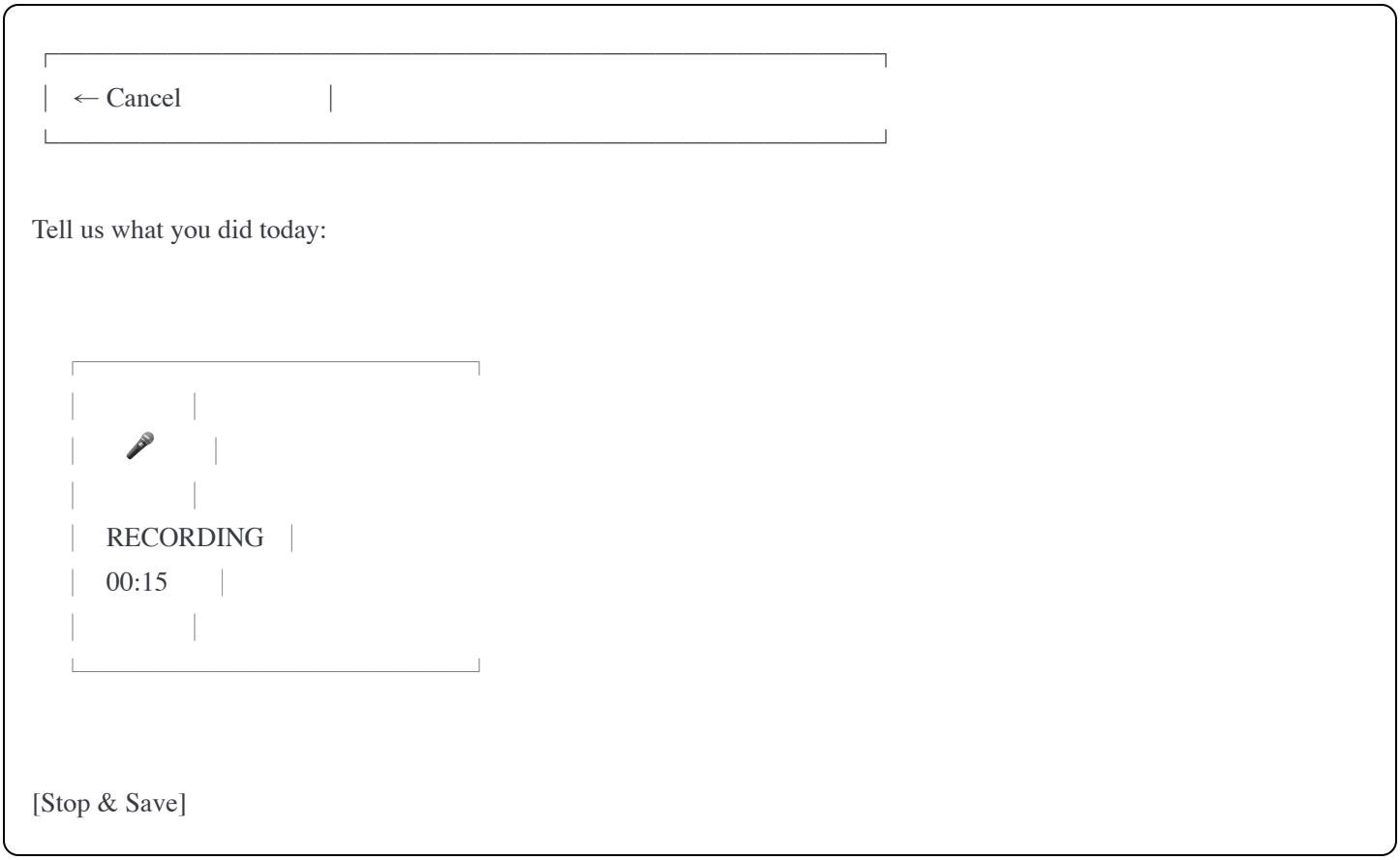
```
│ [Zoomed map showing Plot 2A boundary with activity markers] │
└──────────────────────────────────────────────────────────────
```

Mobile App Screens

Screen 1: Home (Check-In)

```
┌───────────────────────────────────────┐
│ TFA STEELPOORT │
│ Good Morning, Ansi! 🌅 │
└───────────────────────────────────────┘

Today: Monday, Jan 26

Your Goal: Plant 400 cladodes
Progress: 147 / 400 (37%)
[████████░░░░░░░░░░░░]

┌─────────────────────────────────────┐
│ │
│ 🎤 RECORD UPDATE │
│ │
│ Tap to tell us what │
│ you did today │
│ │
└─────────────────────────────────────┘

┌─────────────────────────────────────┐
│ 📷 TAKE PHOTOS │
└─────────────────────────────────────┘

┌─────────────────────────────────────┐
│ ✅ MY TASKS (3) │
│ 2 completed │
└─────────────────────────────────────┘

[Offline Mode: 2 items queued]
```

Screen 2: Voice Recording



PART 9: SUCCESS CRITERIA & RISKS

Definition of Success (MVP)

Must Have:

1. ✅ 100% of field activities logged within 24 hours
2. ✅ ExCo can view real-time KPIs on dashboard
3. ✅ WhatsApp messages auto-processed by Claude AI
4. ✅ Map shows plot boundaries and completion status
5. ✅ Alerts generated for critical issues (>8 hours delay)

Should Have: 6. ✅ Workers can use mobile app (even if also using WhatsApp) 7. ✅ Photos auto-tagged with GPS and activity type 8. ✅ Weekly report auto-generated

Could Have: 9. Computer vision plant health scoring 10. Predictive completion forecasts 11. Voice command interface for dashboard

Risk Matrix

Risk	Probability	Impact	Mitigation
Low worker adoption (prefer WhatsApp)	High	Medium	Make WhatsApp primary input, app is optional
Poor connectivity (rural Limpopo)	High	High	Offline-first PWA, queue data locally
AI extraction errors (Claude misinterprets)	Medium	Medium	Human review queue for low-confidence (<0.7) extractions
Cost overrun (AI API costs)	Low	Low	Set Claude API budget alerts, cache common queries
Data quality issues (duplicate entries)	Medium	Medium	Deduplication logic, activity ID linking
Scope creep (ExCo requests features)	High	High	Strict MVP scope, Phase 2 backlog for new features

---

PART 10: APPENDIX

A. TFA Brand Color Palette (Tailwind)

```javascript
// tailwind.config.js
module.exports = {
  theme: {
    extend: {
      colors: {
        'tfa-green': '#2B7035',
        'tfa-green-dark': '#1F5227',
        'tfa-blue': '#025373',
        'tfa-blue-dark': '#094C6A',
        'tfa-teal': '#01E3C2',
        'tfa-gold': '#A37A51',
        'tfa-charcoal': '#0F1419',
        'tfa-slate': '#1A2332',
      }
    }
  }
}
```

B. Sample Claude Prompt (Message Processing)

```
You are an AI assistant for TerraFerm Africa's farm operations system.

Extract structured data from this field message:

MESSAGE: "Hi Nick, planted 400 cladodes in Plot 2A today with 6 workers.
Rows look good but spacing a bit tight. Weather was hot."

Extract:
1. Activity type (planting, clearing, inspection, etc.)
2. Quantities (number of cladodes, area, etc.)
3. Location (plot ID)
4. Labor (worker count)
5. Issues (problems, concerns)
6. Resource needs (water, equipment, etc.)
7. Date (infer if not explicit)

Return JSON only, no explanation:
{
  "activity_type": "...",
  "plot_id": "...",
  "cladodes_planted": ...,
  "workers": ...,
  "issues": [...],
  "resources_needed": [...],
  "sentiment": "...",
  "confidence": 0.0-1.0
}
```

C. Key User Stories

As an ExCo member, I want to:

- See real-time planting progress so I can adjust strategy
- Receive alerts when operations are off-track
- Export weekly reports for board meetings
- Compare planned vs actual costs

As a field supervisor, I want to:

- Log daily activities quickly (voice or text)

- Attach photos to activities for documentation
- See my team's productivity metrics
- Get notified of quality issues

As a field worker, I want to:

- Check in each morning with one tap
- Report problems via voice note (no typing)
- See my daily progress toward goal
- Work offline when connectivity is poor

Document Status: MVP PRD v1.0 — Ready for Implementation
Next Steps: Create project structure, deploy database schema, build API
Target Launch: Phase 1 complete by Feb 15, 2026 (3 weeks)

"*You have no boss, your boss is data*" — Elon Musk