Fish Ledger App: Complete Development Roadmap

Executive Summary

Project Goal: Create a voice-activated digital ledger for fish sellers in Ghana that automatically records sales through natural conversation monitoring.

Timeline: 12 months from research to public launch
Target Users: Tabletop fish sellers in Ghana markets
Success Metric: 100+ active daily users with 70%+ transaction accuracy by Month 12

PHASE 0: Pre-Development Foundation (Weeks 1-4)

Week 1-2: Market Research & Validation

Objectives
- Validate the problem and solution with real sellers
- Understand daily seller workflows and pain points
- Identify partnership opportunities

Activities
- [ ] Visit 5 different fish markets in Ghana (Accra, Kumasi, Tema)
- [ ] Conduct 30+ seller interviews (15-20 minutes each)
- [ ] Shadow 5 sellers for full market days
- [ ] Document typical daily transaction volumes
- [ ] Identify technology access (smartphone ownership, data plans)

Key Questions to Answer
- How many sales do sellers make per day?
- How do they currently track (if at all)?
- What's their biggest business challenge?
- Would they pay for this solution? How much?
- What languages do they primarily use?

Deliverables
- Market research report with key findings
- User persona profiles (3-5 typical seller types)
- Problem validation document
- Competitive analysis (existing solutions)

Budget: $500-1000 (travel, small compensation for interviews)

Week 3-4: Audio Data Collection

Objectives
- Collect authentic fish market transaction recordings
- Build initial training dataset for AI models
- Understand acoustic environment challenges

Activities
- [ ] Obtain permission from 10 sellers to record transactions
- [ ] Set up recording equipment (smartphones with external mics)
- [ ] Record 200+ complete transactions across different:
  - Times of day (morning rush, midday, evening)
  - Weather conditions (affecting background noise)
  - Market locations (indoor, outdoor, busy, quiet)
- [ ] Manually annotate 100 transactions with:
  - Speaker labels (seller/customer)
  - Transaction phases
  - Fish types mentioned
  - Prices stated
  - Languages used

Recording Protocol
```
For each transaction record:
- Audio file (WAV format, 16kHz)
- Metadata (time, location, market conditions)
- Manual transcript
```

- Fish type and price (ground truth)
- Audio quality rating (1-5)
- Background noise level (low/medium/high)
```

#### **Deliverables**
- 200+ annotated transaction recordings
- Audio quality analysis report
- Common phrase catalog (English + Twi)
- Noise profile analysis

**Budget**: $300-500 (recording equipment, seller compensation)

---

## PHASE 1: MVP Development (Months 2-4)

### **Month 2: Core Audio Processing**

#### **Week 1-2: Development Environment Setup**

**Activities**
- [ ] Set up development environment
  - Install Android Studio / Flutter SDK
  - Set up version control (GitHub/GitLab)
  - Configure CI/CD pipeline
  - Set up testing devices (2-3 Android phones)

- [ ] Create project architecture
  ```
  fish-ledger/
  ├── app/   # Flutter mobile app
  ├── ml_models/   # ML training scripts
  ├── backend/   # Cloud functions (Firebase)
  ├── docs/   # Documentation
  └── tests/   # Test suites
  ```
  ```

- [ ] Set up development team roles (if applicable)
  - Lead developer (you)
  - ML engineer (you or partner)
  - UI/UX designer (freelance or partner)

Deliverables
- Working development environment
- Project repository with initial structure
- Team collaboration tools set up

Week 3-4: Basic Audio Capture

Technical Implementation
```kotlin
// Core audio capture functionality
class AudioCaptureEngine {
    - Continuous audio recording (16kHz, mono)
    - Circular buffer (30 seconds)
    - Voice Activity Detection (VAD)
    - Basic noise filtering
    - Audio file management
}
```

Activities
- [ ] Implement continuous audio recording service
- [ ] Add Voice Activity Detection (using WebRTC VAD)
- [ ] Create circular buffer for audio storage
- [ ] Implement basic noise gate
- [ ] Test battery consumption (target: <10% per hour)
- [ ] Test audio quality in various conditions

Testing Criteria
- Successfully records clear audio in quiet environment

- VAD correctly identifies speech vs silence (>90% accuracy)
- Battery usage acceptable for all-day operation
- Audio files properly managed (no storage overflow)

Deliverables
- Working audio capture module
- Battery consumption report
- Audio quality test results

Month 3: Speech Recognition & Pattern Matching

Week 1-2: Speech-to-Text Integration

Technical Implementation
```python
class SpeechRecognitionEngine {
    - Google Speech-to-Text API integration
    - Offline fallback (basic model)
    - Text preprocessing and cleaning
    - Language detection (English/Twi)
}
```

Activities
- [ ] Integrate Google Cloud Speech-to-Text API
- [ ] Implement offline speech recognition (Vosk/Whisper Lite)
- [ ] Create hybrid online/offline strategy
- [ ] Test recognition accuracy with collected audio samples
- [ ] Optimize for Ghanaian English and Pidgin

Testing Criteria
- >80% word accuracy on clear fish market audio
- <3 second latency for transcription
- Graceful offline mode degradation
- Proper handling of code-switching

Deliverables
- Working speech-to-text module
- Accuracy benchmark report
- Cost analysis for API usage

Week 3-4: Transaction Pattern Detection

Technical Implementation
```python
class TransactionDetector {
    - Keyword pattern matching
    - Price extraction (numbers + currency)
    - Fish type identification
    - Transaction state machine
    - Confidence scoring
}
```

Activities
- [ ] Build pattern matching engine for:
  - Price inquiries ("How much", "Sɛn na ɛyɛ")
  - Price responses (number + "cedis")
  - Fish names (tilapia, tuna, mackerel, etc.)
  - Payment confirmations
- [ ] Implement simple state machine
- [ ] Create confidence scoring algorithm
- [ ] Test with annotated transaction samples

Testing Criteria
- Correctly identifies price mentions (>85% accuracy)
- Recognizes top 5 fish types (>80% accuracy)
- Detects complete transactions (>70% accuracy)
- Minimal false positives (<10%)

Deliverables
- Transaction detection module
- Pattern matching test results
- False positive/negative analysis

Month 4: Database & Basic UI

Week 1-2: Local Database Implementation

Technical Implementation
```sql
-- Core database schema
- Transactions table
- Daily summaries table
- Audio logs table
- User settings table
```

Activities
- [ ] Set up SQLite database
- [ ] Create database models and DAOs
- [ ] Implement CRUD operations
- [ ] Add data validation
- [ ] Create backup/restore functionality
- [ ] Test data integrity and performance

Deliverables
- Working local database
- Data model documentation
- Database performance benchmarks

Week 3-4: Minimum Viable UI

UI Components
```
Screens:
1. Home/Dashboard (today's sales summary)
2. Transaction List (scrollable history)
3. Settings (start/stop listening, language)
4. Setup/Onboarding (first-time user guide)
```

Activities
- [ ] Design simple, accessible UI (low-literacy friendly)
- [ ] Implement home dashboard with:
  - Today's total sales
  - Number of transactions
  - Start/stop listening button
- [ ] Create transaction list view
- [ ] Build simple settings screen
- [ ] Design onboarding flow
- [ ] Test UI with 5 potential users

Testing Criteria
- UI understandable without instructions
- Large, clear buttons and text
- Works well in bright sunlight
- Fast loading (<2 seconds)

Deliverables
- Working mobile app UI
- User testing feedback report
- UI/UX documentation

PHASE 2: Alpha Testing & Refinement (Months 5-6)

Month 5: Integration & Alpha Testing

Week 1: System Integration

Activities
- [ ] Connect all modules (audio → STT → detection → database → UI)
- [ ] Implement end-to-end transaction flow
- [ ] Add error handling and recovery
- [ ] Create logging and debugging tools
- [ ] Performance optimization
- [ ] Battery optimization tweaks

Integration Testing
- Test complete flow: conversation → detected transaction → saved to database → displayed in UI
- Stress testing (100+ transactions in one day)
- Edge case handling
- Memory leak detection

Week 2-4: Alpha Testing with Real Sellers

Alpha Test Program
```
Participants: 5 fish sellers (friendly early adopters)
Duration: 3 weeks
Location: 2-3 different markets
Methodology: Daily usage with weekly check-ins
```

Activities
- [ ] Recruit 5 alpha testers
- [ ] Provide testing phones (or install on their phones)
- [ ] Conduct onboarding training
- [ ] Monitor usage daily via remote logging
- [ ] Weekly in-person check-ins
- [ ] Collect feedback through:
  - Voice recordings (what they like/dislike)
  - Transaction accuracy validation

  - Feature request discussions
  - Problem reports

Success Metrics for Alpha
- All 5 testers use app for full 3 weeks
- >60% transaction detection accuracy
- <5 critical bugs reported
- >50% of testers willing to continue using

Deliverables
- Alpha testing report
- Bug list with priorities
- Feature request backlog
- User testimonial videos (if positive)

Month 6: Iteration Based on Alpha Feedback

Week 1-2: Critical Bug Fixes

Activities
- [ ] Fix all critical bugs from alpha testing
- [ ] Improve transaction accuracy based on real usage data
- [ ] Optimize patterns that caused false positives
- [ ] Enhance UI based on user confusion points
- [ ] Improve battery optimization

Week 3-4: Feature Enhancements

High Priority Features (based on expected feedback)
- [ ] Manual transaction entry (for missed sales)
- [ ] Transaction editing/deletion
- [ ] End-of-day summary voice report
- [ ] Weekly sales comparison

- [ ] Simple profit tracking (buy price vs sell price)

Testing
- Regression testing (ensure fixes didn't break existing features)
- User acceptance testing with alpha testers
- Performance validation

Deliverables
- Improved app version (v0.5)
- Updated documentation
- Release notes

PHASE 3: Beta Launch & Language Expansion (Months 7-8)

Month 7: Beta Launch Preparation

Week 1-2: Twi Language Support

Activities
- [ ] Collect 100+ Twi transaction recordings
- [ ] Train Twi-specific speech model
- [ ] Add Twi pattern recognition
- [ ] Test Twi/English code-switching
- [ ] Update UI with Twi translations

Week 3-4: Beta Program Setup

Beta Test Program
```
Participants: 30 fish sellers across 5 markets
Duration: 8 weeks
Selection: Mix of alpha testers + new users
Support: Dedicated WhatsApp group for support
```

```

**Activities**
- [ ] Recruit 30 beta testers
- [ ] Create beta tester agreement
- [ ] Set up support infrastructure:
  - WhatsApp support group
  - Phone support line (2 hours daily)
  - Remote monitoring dashboard
- [ ] Prepare training materials:
  - Video tutorials (English + Twi)
  - Printed quick-start guide
  - FAQ document

**Beta Success Metrics**
- 25+ active users after 8 weeks (>80% retention)
- >70% transaction accuracy
- <10 critical bugs
- Average 4/5 stars user satisfaction

---

### **Month 8: Beta Testing & Monitoring**

#### **Ongoing Activities**
- [ ] Daily monitoring of app performance
- [ ] Weekly feedback collection via WhatsApp
- [ ] Bi-weekly in-person check-ins at markets
- [ ] Continuous bug fixing
- [ ] Performance optimization based on usage data
- [ ] Collect success stories and testimonials

#### **Data Collection**
- Transaction accuracy rates by market/seller
- Most common false positives/negatives
- Battery usage across different phone models
- Feature usage statistics
```

- User satisfaction scores

Deliverables
- Beta testing report
- Updated feature roadmap
- User case studies (3-5 detailed stories)
- Performance benchmark report

PHASE 4: Advanced Features & Scaling (Months 9-10)

Month 9: Advanced Intelligence Features

Week 1-2: Sales Analytics

Features
- [ ] Weekly/monthly sales trends
- [ ] Best-selling fish identification
- [ ] Peak hours analysis
- [ ] Sales predictions
- [ ] Comparative analytics (this week vs last week)

Week 3-4: Business Intelligence

Features
- [ ] Inventory recommendations
- [ ] Pricing insights (compare with market averages)
- [ ] Customer pattern recognition (repeat customers)
- [ ] Seasonal trend alerts
- [ ] Profit margin calculator

Month 10: Additional Language Support

Languages to Add
- [ ] Ga (coastal regions)
- [ ] Ewe (Volta region)
- [ ] Fante (Central region)

Activities for Each Language
- Collect 50+ transaction recordings
- Train language-specific models
- Add UI translations
- Test with native speakers
- Document common phrases

PHASE 5: Pre-Launch & Marketing (Month 11)

Week 1-2: Production Readiness

Technical Preparation
- [ ] Security audit and penetration testing
- [ ] Privacy compliance review (GDPR, local laws)
- [ ] Performance optimization for scale
- [ ] Set up production infrastructure:
  - Firebase production environment
  - Cloud backup systems
  - Monitoring and alerting
  - Customer support ticketing system

Legal & Compliance
- [ ] Terms of service
- [ ] Privacy policy
- [ ] User data protection measures
- [ ] Business registration (if needed)

Week 3-4: Marketing & Launch Preparation

Marketing Materials
- [ ] Create promotional videos (Twi + English)
- [ ] Design posters for markets
- [ ] Build simple website/landing page
- [ ] Social media presence (Facebook, WhatsApp)
- [ ] Press releases for local media

Partnerships
- [ ] Approach market associations
- [ ] Connect with microfinance institutions
- [ ] Partner with mobile money providers (MTN, AirtelTigo)
- [ ] Reach out to NGOs supporting small businesses

Launch Strategy
```
Soft Launch: 3 markets with existing beta testers
Public Launch: Expand to 20+ markets over 4 weeks
Launch Event: Market demonstrations and free training
```

PHASE 6: Public Launch (Month 12)

Week 1: Soft Launch

Activities
- [ ] Launch in 3 beta test markets
- [ ] Intensive on-ground support (daily presence)
- [ ] Monitor closely for issues
- [ ] Quick iteration on feedback
- [ ] Gather launch testimonials

Week 2-3: Expanded Rollout

Activities
- [ ] Expand to 10 additional markets
- [ ] Conduct market demonstrations
- [ ] Train early adopters to help others
- [ ] Media outreach (radio, TV, newspapers)
- [ ] Social media campaign

Scaling Support
- Hire 2-3 field support staff
- Set up regional training hubs
- Create seller ambassador program
- 24/7 WhatsApp support

Week 4: Full Launch & Celebration

Activities
- [ ] Make app publicly available (Google Play Store)
- [ ] Launch celebration events in key markets
- [ ] Share impact stories and metrics
- [ ] Gather user testimonials and videos
- [ ] Plan for continued growth

Post-Launch: Continuous Improvement (Ongoing)

Monthly Activities
- Release app updates (bug fixes, features)
- Collect user feedback and feature requests
- Monitor transaction accuracy and performance
- Expand to new markets and regions
- Build partnerships for financial inclusion

Quarterly Goals
- Q1 Post-Launch: 500 active users
- Q2: 2,000 active users, add 2 new languages
- Q3: 5,000 active users, launch in new cities
- Q4: 10,000 active users, explore expansion to other product types

---

Resource Requirements

Development Team
- 1 Full-time developer (you + possibly 1 partner)
- 1 Part-time ML engineer (can be same person or consultant)
- 1 Part-time UI/UX designer (freelance, months 3-4)
- 2-3 Field support staff (months 11-12+)

Technology Costs
- Google Cloud credits: $100-200/month
- Firebase: $50-100/month
- Domain & hosting: $20/month
- Testing devices: $500-1000 one-time
- Recording equipment: $300 one-time

Operational Costs
- Market research: $1,000
- Beta tester compensation: $500
- Marketing materials: $1,000
- Launch events: $2,000
- Field staff salaries: $1,500/month (3 people)

Total Estimated Budget: $15,000-25,000 for Year 1

Risk Management

Technical Risks

- **Risk**: Poor audio quality in noisy markets
  Mitigation: Advanced noise cancellation, multiple testing iterations

- **Risk**: Battery drain issues
  Mitigation: Extensive optimization, smart listening modes

- **Risk**: Low recognition accuracy
  Mitigation: Continuous model training, confidence scoring, manual override

Market Risks
- **Risk**: Low smartphone penetration
  Mitigation: Start with markets with higher smartphone usage, consider feature phone version later

- **Risk**: User reluctance to adopt technology
  Mitigation: Free usage, intensive training, visible quick wins

- **Risk**: Competition from existing solutions
  Mitigation: Focus on authentic voice recognition advantage, build strong community

Business Risks
- **Risk**: Difficulty monetizing
  Mitigation: Multiple revenue models (freemium, partnerships, data insights)

- **Risk**: Scaling challenges
  Mitigation: Gradual expansion, ambassador program, automated support

---

Success Criteria by Phase

Phase 1 (Month 4): Working MVP with >60% accuracy
Phase 2 (Month 6): 5 happy alpha users, <5 critical bugs
Phase 3 (Month 8): 25+ active beta users, >70% accuracy
Phase 4 (Month 10): Advanced features working, 3+ languages
Phase 5 (Month 11): Production ready, partnerships secured
Phase 6 (Month 12): Public launch, 100+ active users

Year 1 End Goal: 500-1000 active daily users, 75%+ transaction accuracy, sustainable growth model

---

This roadmap embodies Luther's principles: starting with listening to real people, building something accessible and practical, testing continuously with actual users, and gradually expanding based on real needs rather than assumptions.