AgroVision Pro - Functional Requirements Specification (Lite)

Core System Requirements & Specifications

Document Information

- **Version**: 2.0 (Lite)
- **Date**: 2024
- **Document Type**: Functional Requirements Specification (Streamlined)
- **Target Audience**: Developers, project stakeholders
- **Project Type**: Masters Thesis Research Project
- **Scope**: Core video analysis and crop health detection functionality

---

1. System Overview

1.1 System Architecture

AgroVision Pro is a web-based agricultural intelligence platform with three main components:

1. **React Frontend**: User interface for video upload and results visualization
2. **Spring Boot Backend**: RESTful API for business logic and data management
3. **Python AI Service**: Computer vision processing for crop analysis

1.2 Core Features

Primary Scope:

- Video upload and processing
- AI-powered crop disease detection
- User authentication and management
- Analysis result visualization
- Basic reporting capabilities

Out of Scope (for this version):

- Payment processing
- Weather API integration
- Energy optimization
- Sustainability reporting
- Advanced analytics

1.3 System Dependencies

Essential Dependencies:

- PostgreSQL Database

- Python AI Service (FastAPI)
- File Storage System (Local/AWS S3)

---

2. Core Functional Requirements

2.1 User Management

2.1.1 User Registration

Function: Allow new users to create accounts with email verification.

Input Requirements:

- Email (valid format, unique)
- Password (min 8 chars, secure)
- First Name, Last Name
- Organization Type (FARMER, RESEARCHER, CONSULTANT)

Processing Rules:

1. Validate email format and uniqueness
2. Validate password strength
3. Generate email verification token (24-hour expiry)
4. Hash password using BCrypt
5. Create user record with INACTIVE status
6. Send verification email

Output:

- User ID and success/error message
- Verification status

2.1.2 User Authentication

Function: Authenticate users and provide session management.

Input Requirements:

- Email and password
- Remember me option

Processing Rules:

1. Validate credentials
2. Check account status (must be active)
3. Generate JWT tokens
4. Update last login timestamp

Output:

- Access token and refresh token

- User profile information
- Token expiry information

2.2 Video Processing

2.2.1 Video Upload

Function: Handle video file uploads with validation and storage.

Input Requirements:

- Video file (MP4, MOV, AVI formats)
- Crop type (TOMATO, POTATO, WHEAT, etc.)
- Optional: location, description, GPS coordinates
- Recording timestamp

Processing Rules:

1. Validate file format and size (max 500MB)
2. Extract video metadata (duration, resolution)
3. Store file securely
4. Create database record with UPLOADED status
5. Queue for AI processing

Output:

- Video ID and upload confirmation
- File metadata
- Processing status

2.2.2 Video Processing Status

Function: Provide real-time status updates for video processing.

Input: Video ID

Processing Rules:

1. Verify user ownership
2. Retrieve current processing status
3. Calculate progress percentage
4. Estimate remaining time

Output:

- Current status (UPLOADED, PROCESSING, COMPLETED, FAILED)
- Progress percentage (0-100)
- Estimated completion time
- Error details (if failed)

2.3 AI Analysis

2.3.1 Disease Detection Analysis

Function: Detect and classify crop diseases with severity assessment.

Processing Logic:

1. Extract frames from video (2-3 FPS)
2. Preprocess images (resize, normalize)
3. Run disease detection AI model
4. Aggregate results across frames
5. Calculate confidence scores
6. Generate severity assessment

Output Requirements:

- List of detected diseases with confidence scores
- Overall severity level (0-100%)
- Primary disease classification
- Affected area percentage
- Severity categories (MILD, MODERATE, SEVERE, CRITICAL)

2.3.2 Crop Counting Analysis

Function: Count fruits, vegetables, or other countable crop elements.

Processing Logic:

1. Apply object detection model
2. Filter by confidence threshold (>0.5)
3. Track objects across frames to avoid double-counting
4. Classify maturity levels
5. Calculate density metrics

Output Requirements:

- Total count with confidence interval
- Maturity distribution (unripe, partially ripe, fully ripe, overripe)
- Density per square meter
- Harvest readiness assessment

2.3.3 Growth Stage Analysis

Function: Analyze crop growth stage and maturity level.

Processing Logic:

1. Extract plant structure features
2. Analyze color distribution for ripeness
3. Measure size distributions
4. Compare against growth stage models
5. Calculate days to harvest estimation

Output Requirements:

- Growth stage classification (SEEDLING, VEGETATIVE, FLOWERING, FRUITING, MATURE)
- Maturity percentage (0-100%)
- Days to harvest estimation
- Harvest window recommendation
- Uniformity index

2.4 Recommendation Engine

2.4.1 Treatment Recommendations

Function: Generate treatment recommendations based on analysis results.

Input Requirements:

- Analysis results (diseases, severity)
- User preferences (organic/conventional)
- Local conditions

Processing Rules:

1. Query treatment database for applicable interventions
2. Filter by user preferences
3. Calculate treatment urgency based on severity
4. Estimate costs and effectiveness
5. Rank by cost-benefit ratio
6. Check for treatment conflicts

Output Requirements:

- Prioritized treatment list
- Urgency level (IMMEDIATE, URGENT, MODERATE, MONITOR)
- Estimated costs and potential savings
- Application timing recommendations
- Safety warnings and precautions

2.5 Data Management

2.5.1 Core Data Models

Videos Table:

```sql
CREATE TABLE videos (
    id UUID PRIMARY KEY,
    user_id BIGINT NOT NULL,
    filename VARCHAR(255) NOT NULL,
    file_path VARCHAR(500) NOT NULL,
    file_size BIGINT NOT NULL,
    duration INTEGER,
    crop_type VARCHAR(50),
    location VARCHAR(100),
    recorded_at TIMESTAMP,
    uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(20) DEFAULT 'UPLOADED',
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

Analyses Table:

```
CREATE TABLE analyses (
    id UUID PRIMARY KEY,
    video_id UUID NOT NULL,
    analysis_type VARCHAR(50) NOT NULL,
    status VARCHAR(20) DEFAULT 'PROCESSING',
    results JSONB,
    confidence_score DECIMAL(5,4),
    processed_at TIMESTAMP,
    FOREIGN KEY (video_id) REFERENCES videos(id)
);
```

2.5.2 Historical Data Access

Function: Provide access to historical analysis data for trend analysis.

Requirements:

- Retrieve analyses by time period
- Group data by intervals (daily, weekly, monthly)
- Calculate trend metrics
- Identify patterns and anomalies
- Generate basic visualizations

2.6 Reporting

2.6.1 Analysis Report Generation

Function: Generate comprehensive reports from analysis results.

Input Requirements:

- Analysis ID
- Report type (SUMMARY, DETAILED)
- Format preference (PDF, HTML)

Processing Rules:

1. Retrieve analysis results
2. Apply report template

3. Calculate summary statistics
4. Generate visualizations
5. Include recommendations
6. Format according to user preferences

Output Requirements:

- Downloadable report file
- Summary statistics
- Visualizations (charts, images)
- Treatment recommendations
- Cost analysis

---

3. Use Cases

3.1 Primary Use Case: Farmer Disease Detection

Actor: Commercial Farmer **Goal**: Detect crop diseases early to prevent yield losses

Main Flow:

1. User logs into system
2. Uploads video of affected crop area
3. Selects crop type and adds metadata
4. System processes video (5 minutes max)
5. User receives notification of completed analysis
6. Reviews disease detection results
7. Reviews treatment recommendations
8. Implements recommended treatment
9. Uploads follow-up video to track progress

Success Criteria:

- Disease detected with >85% accuracy
- Processing completed within 5 minutes
- Treatment recommendations provided
- User successfully implements treatment

3.2 Secondary Use Case: Researcher Data Analysis

Actor: Agricultural Researcher **Goal**: Analyze crop data for research purposes

Main Flow:

1. Researcher uploads multiple videos
2. System processes all videos
3. Researcher accesses analytics dashboard
4. Exports data for statistical analysis
5. Generates comparative reports
6. Downloads publication-ready charts

Success Criteria:

- Consistent processing methodology
- Complete data export with metadata
- Statistical significance calculated
- Results suitable for research publication

---

4. Business Rules

4.1 Authentication Rules

- Password minimum 8 characters with complexity requirements
- Maximum 5 failed login attempts before temporary lockout
- JWT token expiry: 8 hours (standard), 30 days (remember me)
- Email verification required for account activation

4.2 Video Processing Rules

- Supported formats: MP4, MOV, AVI
- Maximum file size: 500MB
- Minimum duration: 10 seconds
- Maximum duration: 30 minutes
- Minimum resolution: 480p

4.3 Analysis Confidence Rules

- Minimum confidence threshold: 0.5 (50%)
- High confidence threshold: 0.85 (85%)
- Low confidence results flagged with warnings
- Recommendations only shown for confident detections

4.4 Treatment Priority Rules

- Critical diseases: Immediate treatment required
- High severity + high confidence: Urgent treatment
- Spreading diseases: Higher priority regardless of severity
- Economic viability: Minimum 200% ROI for recommendations

---

5. Performance Requirements

5.1 Response Times

- User authentication: < 2 seconds
- Video upload initiation: < 5 seconds
- Video processing status: < 1 second
- Analysis results retrieval: < 3 seconds
- Report generation: < 30 seconds

5.2 Processing Capacity

- Concurrent video uploads: 20 users
- Concurrent AI analyses: 5 videos
- Video processing time: Maximum 5 minutes
- System availability: 99% uptime

5.3 Scalability

- Target concurrent users: 100
- Video storage: 5GB per user
- Database growth: 5GB per month

---

6. Security Requirements

6.1 Authentication Security

- JWT tokens with secure signing
- Password hashing using BCrypt (12 rounds)
- HTTPS for all communications
- Session timeout after inactivity

6.2 Data Security

- Encrypted storage for sensitive data
- Input validation and sanitization
- SQL injection prevention
- File upload security (virus scanning)

6.3 Access Control

- User can only access own data
- Role-based permissions (FARMER, RESEARCHER, CONSULTANT)
- API rate limiting
- Audit logging for sensitive operations

---

7. Integration Requirements

7.1 AI Service Integration

API Contract:

```
{
  "videoId": "string",
  "videoPath": "string",
  "cropType": "string",
  "analysisTypes": ["DISEASE_DETECTION", "COUNTING", "MATURITY"],
  "parameters": {}
}
```

Response:

```
{
  "analysisId": "string",
  "status": "SUCCESS|FAILED|PROCESSING",
  "processingTimeMs": "number",
  "results": {},
  "warnings": ["string"],
  "errorMessage": "string"
}
```

7.2 Error Handling

- Retry mechanism for AI service failures (max 3 attempts)
- Graceful degradation for non-critical features
- User-friendly error messages
- Detailed logging for debugging

---

8. Implementation Guidelines

8.1 Development Approach

- Follow specification for core functionality
- Allow flexibility for UI/UX enhancements
- Implement comprehensive error handling
- Maintain backward compatibility

8.2 Testing Requirements

- Unit tests for all business logic
- Integration tests for API endpoints
- End-to-end tests for critical user flows
- Performance testing for video processing

8.3 Documentation

- API documentation with OpenAPI/Swagger
- Code comments for complex logic
- User guides for key features
- Deployment and setup instructions

---

9. Success Metrics

9.1 Technical Metrics

- Video processing success rate: >95%
- Disease detection accuracy: >85%
- System response time: ❤️ seconds average
- User satisfaction score: >4.0/5.0

9.2 Business Metrics

- User adoption rate
- Video upload frequency
- Treatment recommendation acceptance rate
- User retention rate