Solution Requirements Document

Project: PoultryDetect - AI-Powered Poultry Disease Detection System
Location: Ongole, Andhra Pradesh
Date: June 2025
Team ID: LTVIP2025TMID42969
Team Members: M. Karthik Reddy, P. Srinivasa Kalyan

1. Functional Requirements

1.1 Core Functionality Requirements

FR-001: Image Upload System

- **Priority:** High
- **Description:** Users must be able to upload poultry images for analysis
- **Requirements:**
  - Support for JPEG, JPG, PNG image formats
  - Maximum file size limit of 10MB
  - File validation and sanitization
  - Progress indicator during upload
  - Error handling for invalid files
- **Acceptance Criteria:**
  - File upload success rate > 95%
  - Upload completion within 30 seconds
  - Clear error messages for rejected files

FR-002: AI Disease Prediction

- **Priority:** High
- **Description:** System must classify uploaded images into disease categories
- **Requirements:**
  - Integration with pre-trained CNN model (healthy_vs_rotten.h5)
  - Support for 4 disease classifications:
    - Coccidiosis
    - Healthy
    - Salmonella
    - Newcastle Disease

    - Image preprocessing to 224x224 pixel format
    - Confidence score calculation and display
- **Acceptance Criteria:**
    - Prediction accuracy > 85%
    - Processing time < 10 seconds
    - Confidence score displayed as percentage

FR-003: Result Display System

- **Priority:** High
- **Description:** Present prediction results in user-friendly format
- **Requirements:**
    - Clear disease name display
    - Confidence percentage visualization
    - Original uploaded image display
    - Treatment recommendations
    - Management suggestions
- **Acceptance Criteria:**
    - Results displayed immediately after processing
    - Information presented in simple, non-technical language
    - Visual clarity for users with basic literacy

FR-004: Educational Content System

- **Priority:** Medium
- **Description:** Provide comprehensive disease information and research access
- **Requirements:**
    - Disease information cards with symptoms, treatment, management
    - Research links to Google Scholar
    - Educational journey timeline
    - Visual disease identification guides
    - Prevention and management best practices
- **Acceptance Criteria:**
    - Complete information for all 4 disease types
    - External research links functional and current
    - Content accessible without technical knowledge

1.2 User Interface Requirements

FR-005: Web Interface Navigation

- **Priority:** High
- **Description:** Intuitive navigation system for all user types
- **Requirements:**
  - Four main navigation sections: Home, About, Contact, Discover
  - Responsive design for mobile and desktop
  - Consistent visual design across pages
  - Clear call-to-action buttons
  - Accessibility features for users with disabilities
- **Acceptance Criteria:**
  - Navigation functional on all device types
  - Page load times < 3 seconds
  - Intuitive user flow with minimal learning curve

FR-006: Visual Design System

- **Priority:** Medium
- **Description:** Appealing and professional visual presentation
- **Requirements:**
  - Tailwind CSS framework implementation
  - Farm/agricultural theme with background imagery
  - Glass morphism design elements
  - Consistent color scheme (green primary)
  - Animation elements (Lottie, CSS animations)
- **Acceptance Criteria:**
  - Professional appearance suitable for agricultural context
  - Visual hierarchy guides user attention effectively
  - Animations enhance rather than distract from functionality

1.3 Content Management Requirements

FR-007: Static Content Delivery

- **Priority:** Medium
- **Description:** Efficient delivery of static assets

- **Requirements:**
  - Image storage in static/uploads directory
  - CSS and JavaScript asset optimization
  - CDN integration for external libraries
  - Automated cleanup of temporary files
- **Acceptance Criteria:**
  - Fast asset loading across different connection speeds
  - Reliable access to external CDN resources
  - No storage overflow from temporary files

2. Non-Functional Requirements

2.1 Performance Requirements

NFR-001: Response Time

- **Requirement:** System must provide fast response times for all operations
- **Specifications:**
  - Page load time: < 3 seconds
  - Image upload: < 30 seconds for 5MB files
  - Prediction processing: < 10 seconds
  - Navigation response: < 1 second
- **Measurement:** Response time monitoring and user experience testing

NFR-002: Throughput

- **Requirement:** Support concurrent user operations
- **Specifications:**
  - Handle 10 concurrent image uploads
  - Process 50 predictions per hour
  - Serve 100 page requests per minute
- **Measurement:** Load testing and performance monitoring

NFR-003: Resource Utilization

- **Requirement:** Efficient use of system resources
- **Specifications:**
  - Memory usage < 1GB during peak operation
  - CPU utilization < 80% under normal load

    - Storage growth < 100MB per day
- **Measurement:** System monitoring and resource tracking

2.2 Reliability Requirements

NFR-004: Availability

- **Requirement:** System should be available for users when needed
- **Specifications:**
    - Uptime target: 99% during development/testing
    - Graceful degradation during high load
    - Error recovery within 30 seconds
- **Measurement:** Uptime monitoring and error rate tracking

NFR-005: Error Handling

- **Requirement:** Robust error handling and user feedback
- **Specifications:**
    - No system crashes from user input
    - Clear error messages for all failure scenarios
    - Automatic recovery from transient errors
    - Fallback options for failed operations
- **Measurement:** Error rate monitoring and user feedback

2.3 Security Requirements

NFR-006: Input Validation

- **Requirement:** Secure handling of user inputs and uploads
- **Specifications:**
    - File type validation for image uploads
    - File size limits enforcement
    - Filename sanitization using secure_filename()
    - Path traversal attack prevention
- **Measurement:** Security testing and vulnerability assessment

NFR-007: Data Privacy

- **Requirement:** Protection of user data and privacy
- **Specifications:**
    - No permanent storage of uploaded images

    - Automatic cleanup of temporary files
    - No collection of personal information
    - No tracking cookies or user identification
- **Measurement:** Privacy audit and data flow verification

2.4 Usability Requirements

NFR-008: User Experience

- **Requirement:** Easy-to-use interface for non-technical users
- **Specifications:**
    - Intuitive navigation requiring no training
    - Clear visual hierarchy and information presentation
    - Consistent interaction patterns
    - Mobile-friendly responsive design
- **Measurement:** User testing and feedback collection

NFR-009: Accessibility

- **Requirement:** Accessible to users with varying abilities
- **Specifications:**
    - Keyboard navigation support
    - Screen reader compatibility
    - High contrast color options
    - Large text options for readability
- **Measurement:** Accessibility testing and compliance verification

2.5 Compatibility Requirements

NFR-010: Browser Compatibility

- **Requirement:** Function across modern web browsers
- **Specifications:**
    - Chrome, Firefox, Safari, Edge support
    - Mobile browser compatibility
    - JavaScript-enabled browsers required
    - HTML5 and CSS3 feature support
- **Measurement:** Cross-browser testing and compatibility verification

NFR-011: Device Compatibility

- **Requirement:** Responsive design for various devices
- **Specifications:**
  - Desktop computers (1920x1080+)
  - Tablets (768px+ width)
  - Mobile phones (320px+ width)
  - Touch and mouse input support
- **Measurement:** Device testing and responsive design verification

3. Technical Requirements

3.1 System Architecture Requirements

TR-001: Backend Framework

- **Requirement:** Flask-based web application architecture
- **Specifications:**
  - Python 3.8+ runtime environment
  - Flask framework for web server
  - Werkzeug for file handling utilities
  - Modular application structure
- **Dependencies:** Python, Flask, Werkzeug

TR-002: Machine Learning Integration

- **Requirement:** AI model integration for disease prediction
- **Specifications:**
  - Keras/TensorFlow model loading and inference
  - NumPy for numerical computations
  - Image preprocessing pipeline
  - Model versioning capability
- **Dependencies:** TensorFlow, Keras, NumPy, PIL

TR-003: Frontend Technology Stack

- **Requirement:** Modern web frontend implementation
- **Specifications:**
  - HTML5 semantic markup
  - CSS3 with Tailwind framework
  - Vanilla JavaScript for interactions

    - Jinja2 templating engine
- **Dependencies:** Tailwind CSS CDN, Lottie animations

3.2 Data Requirements

TR-004: File Storage System

- **Requirement:** Temporary file storage for image processing
- **Specifications:**
    - Local filesystem storage in static/uploads
    - Automatic cleanup of processed files
    - File organization by processing session
    - Storage quota management
- **Capacity:** 1GB temporary storage allocation

TR-005: Model Data Requirements

- **Requirement:** ML model and associated data files
- **Specifications:**
    - Pre-trained model file (healthy_vs_rotten.h5)
    - Model metadata and configuration
    - Class label definitions
    - Model performance metrics
- **Size:** ~100MB model file storage

3.3 Integration Requirements

TR-006: External Service Integration

- **Requirement:** Integration with external services and APIs
- **Specifications:**
    - CDN integration for CSS/JS libraries
    - Google Scholar search integration
    - Lottie animation service
    - External font and icon libraries
- **Dependencies:** Internet connectivity for CDN resources

TR-007: API Design

- **Requirement:** Internal API structure for future extensibility

- **Specifications:**
  - RESTful endpoint design principles
  - JSON response format standardization
  - Error response standardization
  - Version control for API changes
- **Future:** Mobile app integration capability

4. Constraints and Assumptions

4.1 Technical Constraints

TC-001: Development Timeline

- **Constraint:** 3-day development timeline (June 24-26, 2025)
- **Impact:** Limited scope and feature complexity
- **Mitigation:** Focus on core functionality, defer advanced features

TC-002: Resource Limitations

- **Constraint:** Two-developer team with limited hardware
- **Impact:** Simplified architecture and minimal infrastructure
- **Mitigation:** Use efficient frameworks and cloud-ready design

TC-003: Model Limitations

- **Constraint:** Pre-trained model with fixed accuracy and capabilities
- **Impact:** Cannot modify model architecture or training
- **Mitigation:** Optimize preprocessing and result presentation

4.2 Business Constraints

BC-001: Target Audience

- **Constraint:** Primary users have limited technical expertise
- **Impact:** Interface must be extremely simple and intuitive
- **Mitigation:** User-centered design and extensive testing

BC-002: Geographic Context

- **Constraint:** Focus on Ongole, Andhra Pradesh agricultural context
- **Impact:** Content and examples must be locally relevant
- **Mitigation:** Use local terminology and farming practices

4.3 Assumptions

AS-001: User Environment

- **Assumption:** Users have access to smartphones or computers with cameras
- **Validation:** Target demographic analysis
- **Risk:** Limited device access may reduce adoption

AS-002: Internet Connectivity

- **Assumption:** Basic internet connectivity available for web access
- **Validation:** Regional connectivity studies
- **Risk:** Poor connectivity may affect user experience

AS-003: Model Accuracy

- **Assumption:** Pre-trained model provides sufficient accuracy for user needs
- **Validation:** Testing with known disease samples
- **Risk:** Poor predictions may damage user trust