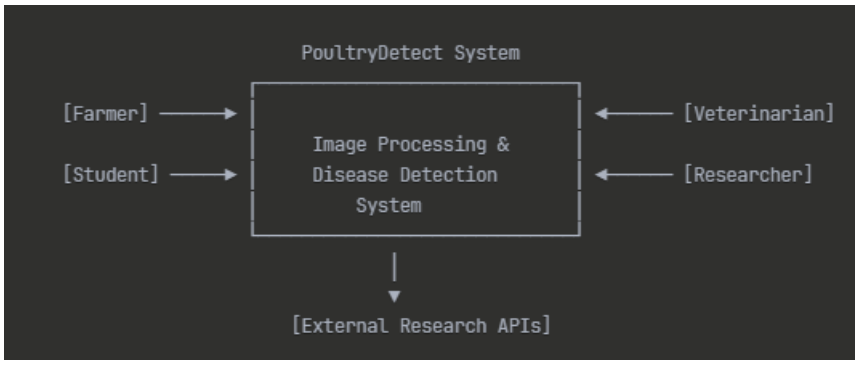
Data Flow Diagrams and User Stories

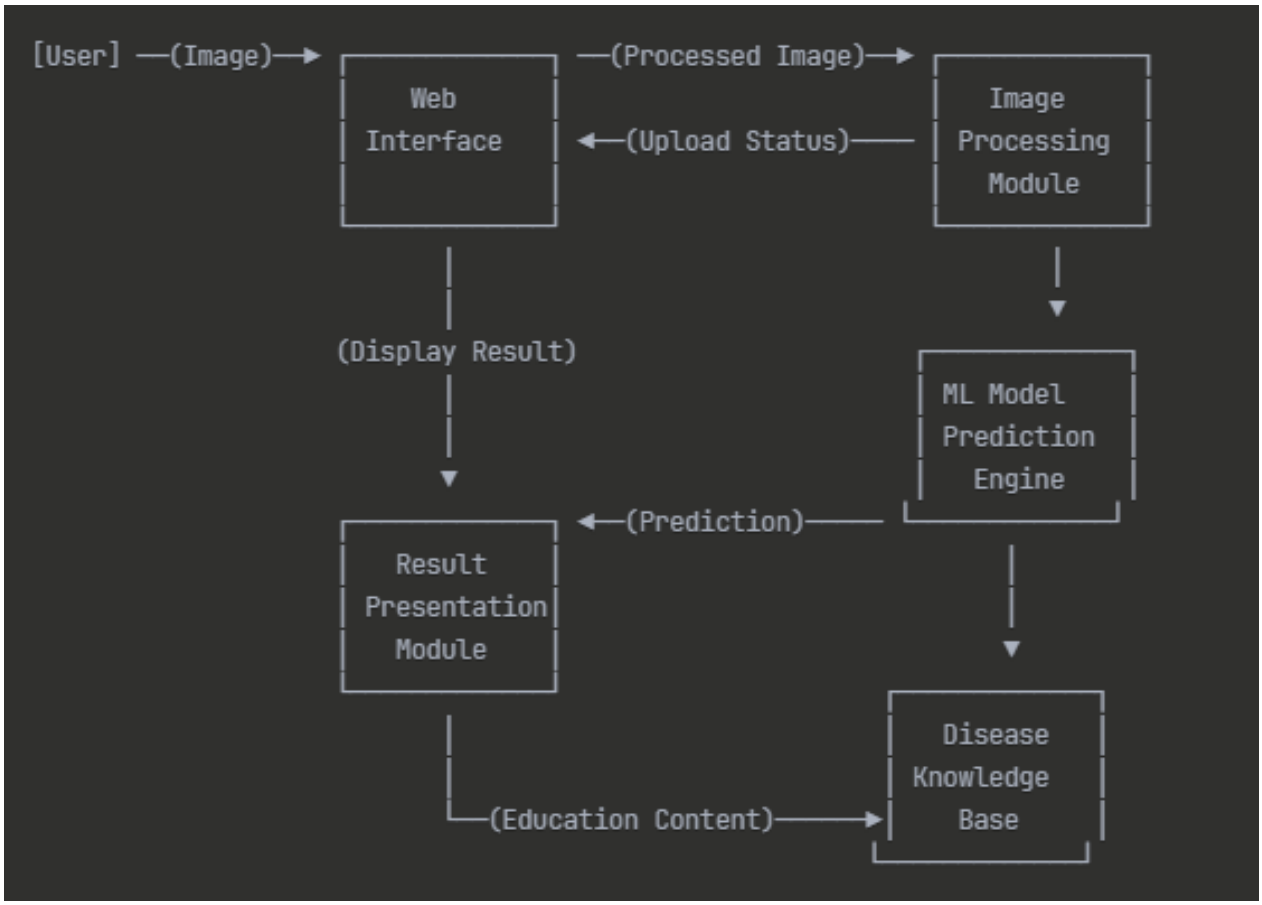
Project: PoultryDetect - AI-Powered Poultry Disease Detection System
Location: Ongole, Andhra Pradesh
Date: June 2025
Team ID: LTVIP2025TMID42969
Team Members: M. Karthik Reddy, P. Srinivasa Kalyan

1. Data Flow Diagrams

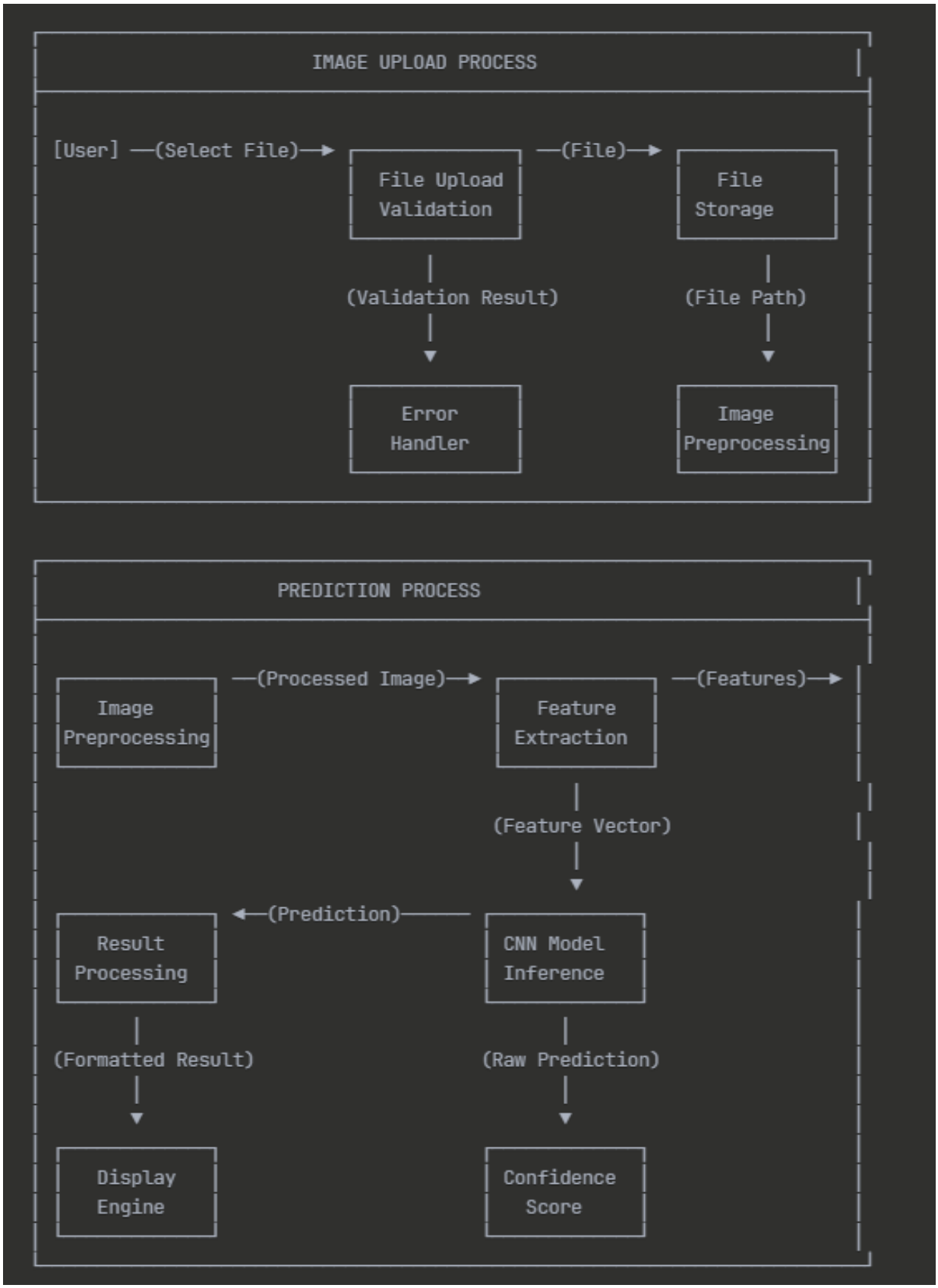
1.1 Level 0 DFD (Context Diagram)



1.2 Level 1 DFD (System Overview)



1.3 Level 2 DFD (Detailed Process Flow)



1.4 Data Flow Specifications

Process 1: File Upload Validation

- **Input:** User selected image file
- **Processing:**
    - Check file type (JPG, PNG, JPEG)
    - Validate file size (< 10MB)
    - Sanitize filename using secure_filename()
- **Output:** Valid file path or error message

Process 2: Image Preprocessing

- **Input:** Raw image file
- **Processing:**
    - Resize to 224x224 pixels
    - Normalize pixel values (0-1 range)
    - Convert to numpy array
- **Output:** Preprocessed image array

Process 3: ML Model Inference

- **Input:** Preprocessed image array
- **Processing:**
    - Load pre-trained CNN model
    - Run forward pass prediction
    - Extract confidence scores
- **Output:** Disease classification and confidence

Process 4: Result Presentation

- **Input:** Raw prediction results
- **Processing:**
    - Map prediction to disease name
    - Format confidence percentage
    - Generate treatment recommendations
- **Output:** User-friendly result display

2. User Stories

2.1 Epic 1: Disease Detection

Story 1.1: Basic Image Upload

As a farmer

I want to upload a photo of my sick poultry

So that I can get a quick disease diagnosis

Acceptance Criteria:

- User can select image files from device
- System accepts common image formats (JPG, PNG)
- File size validation prevents oversized uploads
- Clear error messages for invalid files
- Upload progress indication for slow connections

Story 1.2: AI Disease Prediction

As a poultry owner

I want to receive an accurate disease prediction

So that I can take appropriate treatment action

Acceptance Criteria:

- Prediction completed within 10 seconds
- Result shows disease name clearly
- Confidence score displayed as percentage
- Original uploaded image shown with result
- Prediction covers 4 disease categories: Coccidiosis, Healthy, Salmonella, Newcastle Disease

Story 1.3: Treatment Guidance

As a farmer with limited veterinary access

I want to receive basic treatment suggestions

So that I can start appropriate care immediately

Acceptance Criteria:

- Treatment recommendations displayed with prediction
- Management tips provided for each disease
- Clear disclaimer about professional consultation
- Simple language avoiding medical jargon

- Links to detailed research for more information

2.2 Epic 2: Educational Resources

Story 2.1: Disease Information Learning

As a poultry farmer

I want to learn about common poultry diseases

So that I can better prevent and manage health issues

Acceptance Criteria:

- Dedicated training/research page available
- Information cards for each disease type
- Symptoms, treatment, and management tips
- Visual icons and clear formatting
- Links to scientific research sources

Story 2.2: Research Access

As a veterinary student

I want to access research papers and studies

So that I can deepen my knowledge of poultry diseases

Acceptance Criteria:

- Direct links to Google Scholar searches
- Disease-specific research queries
- Current and relevant study access
- Educational journey timeline
- Step-by-step learning process

Story 2.3: Professional Development

As a veterinarian

I want to use the tool for preliminary screening

So that I can optimize my consultation time

**Acceptance

2.3 Epic 3: User Experience

Story 3.1: Intuitive Navigation

As a first-time user

I want to easily navigate the application

So that I can use it without technical training

Acceptance Criteria:

- Clear navigation menu with 4 main sections
- Consistent design across all pages
- Mobile-responsive interface
- Visual cues and animations for guidance
- Breadcrumb navigation for complex flows

Story 3.2: Multilingual Support (Future)

As a Telugu-speaking farmer

I want to use the application in my native language

So that I can better understand the results

Acceptance Criteria:

- Language selection option
- Complete Telugu translation
- Cultural adaptation of content
- Local disease terminology
- Regional treatment preferences

Story 3.3: Offline Capability (Future)

As a farmer in areas with poor connectivity

I want to use basic features offline

So that I can still get disease identification

Acceptance Criteria:

- Offline model caching
- Local image processing
- Sync when connection available
- Reduced feature set for offline mode
- Clear online/offline status indication

2.4 Epic 4: System Performance

Story 4.1: Fast Response Times

As a user with urgent disease concerns

I want quick prediction results

So that I can take immediate action

Acceptance Criteria:

- Image upload completes in < 5 seconds
- Prediction processing in < 10 seconds
- Page load times under 3 seconds
- Responsive interface during processing
- Progress indicators for long operations

Story 4.2: Error Handling

As a user who may make mistakes

I want clear error messages and recovery options

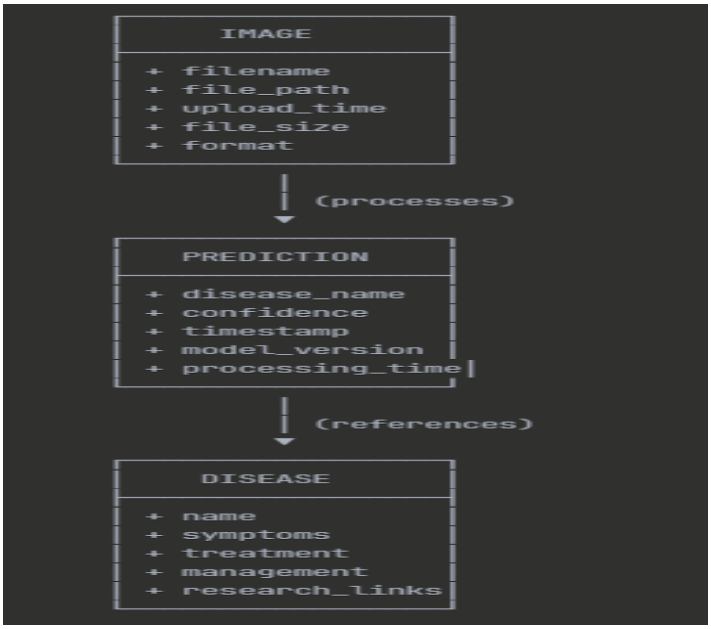
So that I can successfully complete my task

Acceptance Criteria:

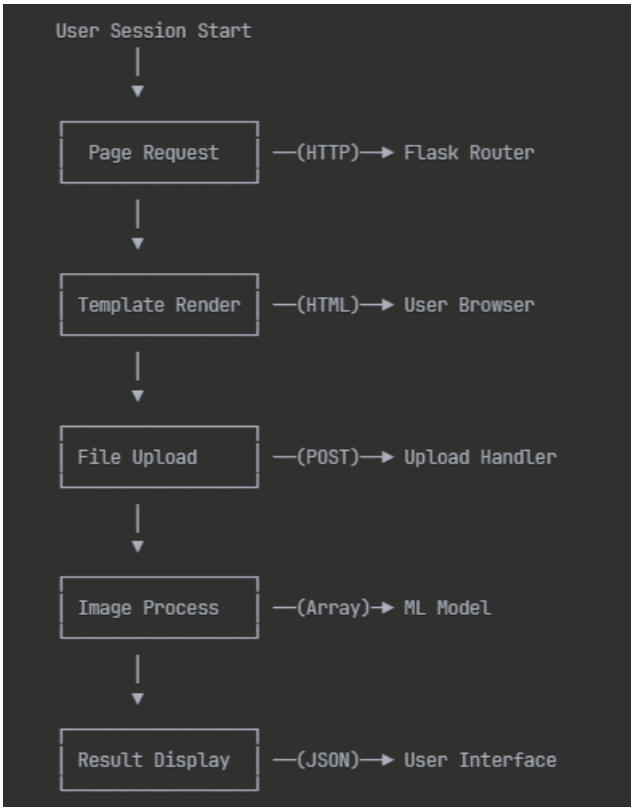
- Graceful handling of invalid files
- Clear error messages in simple language
- Suggested corrective actions
- Fallback options for failed processes
- No system crashes from user errors

3. Data Entity Relationships

3.1 Core Data Entities



3.2 Session Data Flow



4. Technical User Stories

4.1 Developer Stories

Story D1: Model Integration

As a developer

I want to easily integrate new ML models

So that I can improve prediction accuracy

Acceptance Criteria:

- Modular model loading system
- Version control for models
- A/B testing capability
- Performance monitoring
- Rollback mechanisms

Story D2: API Development

As a developer

I want to create RESTful APIs

So that mobile apps can integrate with the system

Acceptance Criteria:

- RESTful endpoint design
- JSON response format
- Authentication mechanism
- Rate limiting implementation
- API documentation

4.2 Administrator Stories

Story A1: System Monitoring

As a system administrator

I want to monitor application performance

So that I can ensure optimal user experience

Acceptance Criteria:

- Real-time performance metrics
- Error rate monitoring
- User activity tracking
- Resource utilization alerts
- Automated scaling triggers

Story A2: Content Management

As a content administrator

I want to update disease information

So that users get current medical knowledge

Acceptance Criteria:

- Easy content editing interface
- Version control for content changes
- Review and approval workflow
- Scheduled content updates
- Multi-language content support

5. Data Security & Privacy

5.1 Privacy User Stories

Story P1: Data Protection

As a privacy-conscious user

I want my uploaded images to be handled securely

So that my farm information remains confidential

Acceptance Criteria:

- No permanent storage of uploaded images
- Automatic file deletion after processing
- No personal data collection
- Secure file transmission
- Clear privacy policy

Story P2: Anonymous Usage

As a user concerned about privacy

I want to use the system without creating accounts

So that my identity remains anonymous

Acceptance Criteria:

- No user registration required
- Stateless session management
- No tracking cookies
- No personal information requests
- Anonymous usage analytics only