# **REQUIREMENT ANALYSIS**

# **1 . Customer Journey Map**

A Customer Journey Map helps visualize the experience of a user interacting with a product or service over time. It helps understand the user's needs, pain points, emotions, and goals at each stage of interaction.

## **User: Rural Poultry Farmer**

Scenario: Farmer trying to identify and treat poultry disease using the AI tool

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stage | User Action | Touchpoints | Pain Points | User Emotion |
| Awareness | Farmer sees poster/ad about AI poultry disease detector | Flyers, Social Media, Word of Mouth | Not sure if it will work; new to technology | Curious, Cautious |
| Consideration | Farmer decides to try the tool after a chicken falls sick | Mobile phone, Website | Uncertain how to use; worries about data use | Hopeful, Hesitant |
| Interaction | Farmer uploads photo of sick chicken | Web interface | Slow internet; unsure about photo quality | Anxious, Interested |
| Diagnosis | System displays disease prediction and tips | AI Model, Screen Output | Doesn’t fully understand medical terms | Relieved, Slightly Confused |
| Action | Farmer takes suggested action or contacts local help | Local Vet, Self-treatment Guide | No vet nearby; unsure of exact medicine | Determined, Worried |
| Feedback | Farmer sees improvement or shares result | App Feedback Form, Peer Sharing | No one to verify result | Satisfied or Unsure |

**2 . Solution Requirements**

|  |  |
| --- | --- |
| Date | 16th June 2025 |
| Team ID | LTVIP2025TMID34447 |
| Project Name | Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management |
| Maximum Marks | 4 Marks |

# **Functional Requirements:**

The following are the functional requirements of our poultry disease detection web application.

|  |  |  |
| --- | --- | --- |
| FR No. | Functional Requirement (Epic) | Sub-Requirement (Story / Sub-Task) |
| FR-1 | User Registration | • Registration through Form • Registration through Gmail • Registration through LinkedIn |
| FR-2 | User Confirmation | • Confirmation via Email • Confirmation via OTP |
| FR-3 | Image Upload | • Upload poultry image (JPG/PNG) • Show image preview before prediction |
| FR-4 | Disease Prediction | • Predict disease using AI model (VGG16) • Display result with accuracy/confidence |
| FR-5 | Result History (Optional) | • Show last 5 predictions • Option to download prediction result/report |

# **Non-Functional Requirements:**

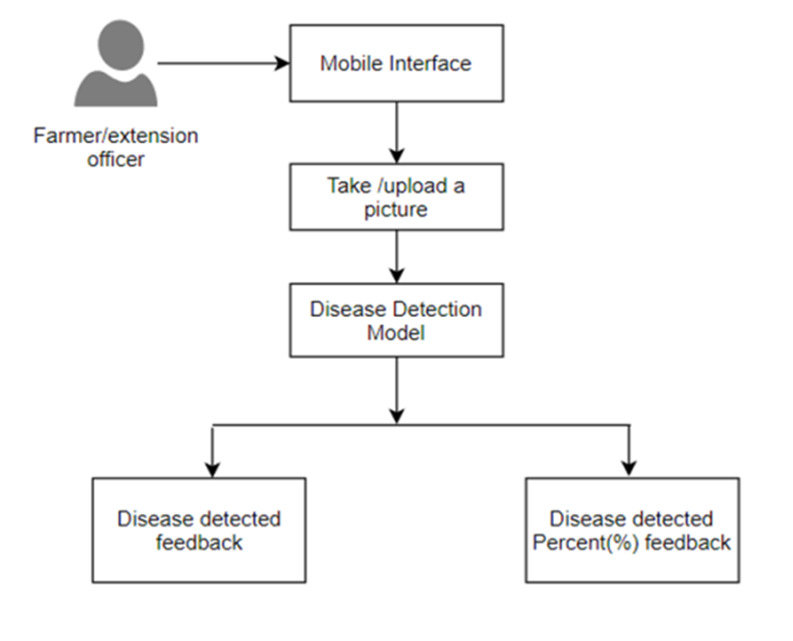
These requirements ensure system quality, performance, and usability.

|  |  |  |
| --- | --- | --- |
| NFR No. | Non-Functional Requirement | Description |
| NFR-1 | Usability | Easy-to-use interface for farmers with clear buttons and steps |
| NFR-2 | Security | HTTPS encryption, secure login system |
| NFR-3 | Reliability | System provides consistent and stable results |
| NFR-4 | Performance | Prediction results load within 2–3 seconds |
| NFR-5 | Availability | Web application available 24x7 with minimal downtime |
| NFR-6 | Scalability | System can be expanded to support more users and detect more diseases |

# **3 .Data Flow Diagrams (DFD):**

|  |  |
| --- | --- |
| Date | 16th June 2025 |
| Team ID | LTVIP2025TMID34447 |
| Project Name | Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management |
| Maximum Marks | 4 Marks |

A Data Flow Diagram (DFD) is a simple graphical way to show how information flows through a system. It shows how data enters, where it goes, how it's processed, and where it's stored.  
  
DFD Level 0 (Context Diagram):  
User → uploads image → System → returns predicted disease

DFD Level 1 (Detailed Flow):  
- User uploads poultry image  
- Web Application validates image and sends it to AI Model  
- AI Model processes image and predicts disease  
- Result shown on Web Interface  
- (Optional) Result stored in database for user history

# **User Stories:**

The table below lists the user stories that describe how different users will interact with the system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User Type | Functional Requirement (Epic) | User Story No. | User Story / Task | Acceptance Criteria | Priority / Release |
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register by entering my email, password, and confirm password. | I can access my account / dashboard | High / Sprint-1 |
|  | Registration | USN-2 | As a user, I receive a confirmation email after registering. | I can receive email & click confirm | High / Sprint-1 |
|  | Registration | USN-3 | As a user, I can register using Facebook. | I can access the dashboard with Facebook login | Low / Sprint-2 |
|  | Registration | USN-4 | As a user, I can register using Gmail. | I can access the dashboard with Gmail login | Medium / Sprint-1 |
|  | Login | USN-5 | As a user, I can log in using my email and password. | I am redirected to dashboard after successful login | High / Sprint-1 |

# **4. Technical Architecture:**

|  |  |
| --- | --- |
| Date | 16th June 2025 |
| Team ID | LTVIP2025TMID34447 |
| Project Name | Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management |
| Maximum Marks | 4 Marks |

The project uses a simple 3-tier architecture:  
Frontend (User Interface) → Backend (Application Logic + AI Model) → Data Storage  
  
Architecture Flow:  
- User uploads image via browser (HTML/Flask)  
- Image is processed by AI model (VGG16 via TensorFlow)  
- Result is returned to user and optionally stored in database

## **Table-1: Components & Technologies**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology Used |
| 1 | User Interface | How users interact (via browser) | HTML, CSS, JavaScript, Flask Templates |
| 2 | Application Logic-1 | Web app routing and backend logic | Python, Flask |
| 3 | Application Logic-2 | Image preprocessing for model input | OpenCV, Pillow (PIL) |
| 4 | Application Logic-3 | AI Model Prediction using VGG16 | TensorFlow, Keras |
| 5 | Database | Store user details, history | SQLite / MySQL |
| 6 | Cloud Database | Optional – Cloud-based database storage | Firebase / Google Cloud SQL (optional) |
| 7 | File Storage | Temporary storage of uploaded images | Local File System |
| 8 | External API-1 | Optional – Add weather info (future scope) | OpenWeather API (optional) |
| 9 | External API-2 | Not used | – |
| 10 | Machine Learning Model | Classify poultry disease from images | VGG16 with Transfer Learning |
| 11 | Infrastructure | Hosted on local system or cloud (future scope) | Localhost / Google Cloud / Render / AWS |

## **Table-2: Application Characteristics**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology Used |
| 1 | Open-Source Frameworks | Used free and open-source libraries | Flask, TensorFlow, OpenCV |
| 2 | Security Implementations | Basic data validation, secure file upload, HTTPS (future scope) | Flask Validation, SHA256 (optional) |
| 3 | Scalable Architecture | Based on modular 3-tier design (frontend, backend, model) | Flask API + ML microservice |
| 4 | Availability | Can be hosted on any cloud with 24/7 uptime using hosting providers | Google Cloud, AWS (optional) |
| 5 | Performance | Light model, low-latency image processing, suitable for real-time use | Model size optimized, Cacheable |