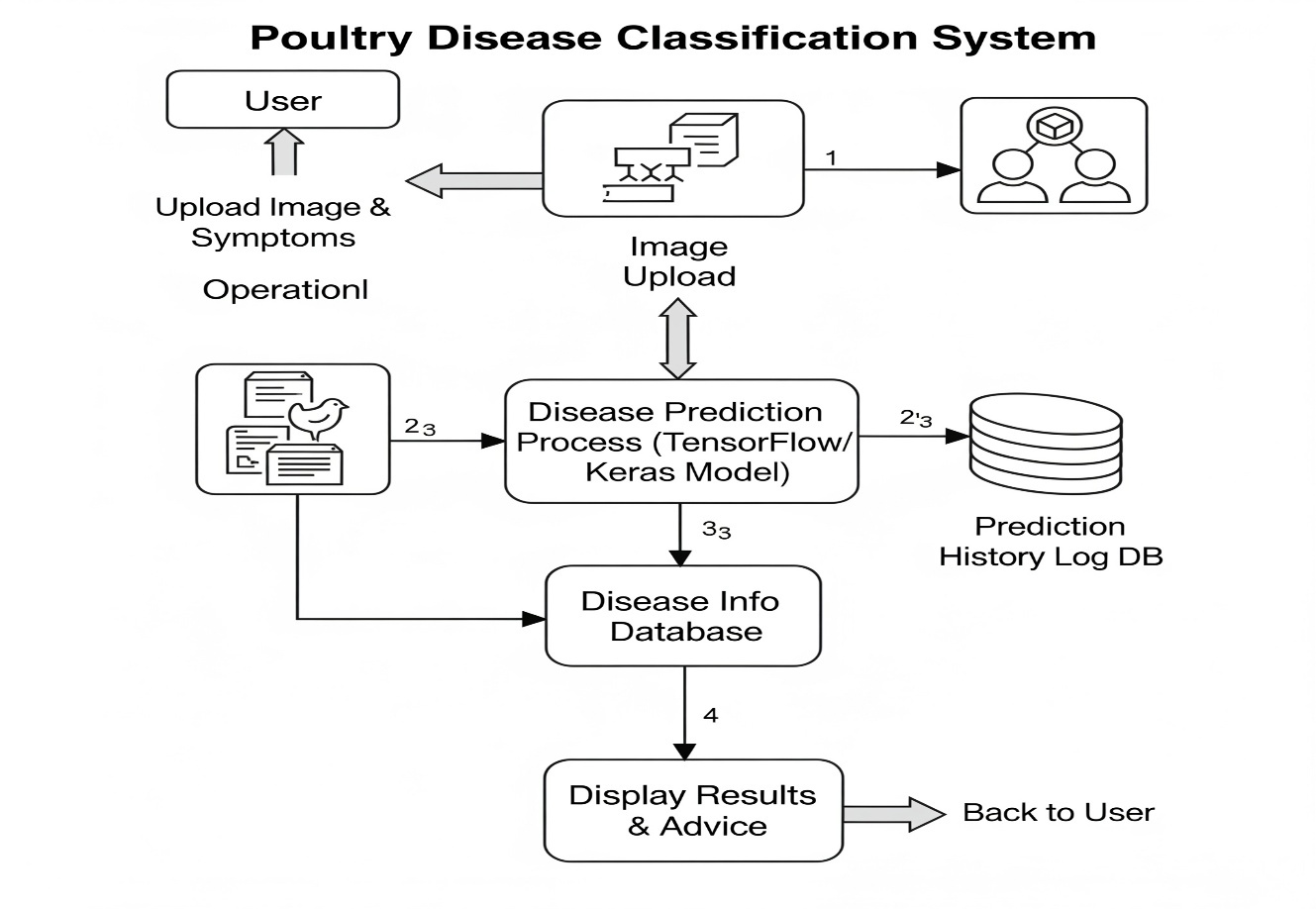
**3 Data Flow Diagram (DFD)**

|  |  |
| --- | --- |
| **Date** | **24 June 2025** |
| **Team ID** | **LTVIP2025TMID43223** |
| **Project Name** | **Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management** |
| **Maximum Marks** | **4 Marks** |

****

The Data Flow Diagram (DFD) represents how data moves within the poultry disease classification system. Users upload an image via the web app, which is processed by a TensorFlow/Keras AI model through a Flask backend. The system predicts the disease and retrieves related information from the disease database. The result, along with preventive measures, is displayed to the user. All predictions are logged in a history data store for future tracking. Additional features like About and Contact are also accessible through the interface**.**

**User Stories Table**

**Customer (Web User / Mobile User via Flask App)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| Customer | Image Upload | USN-1 | As a user, I can upload an image of a sick poultry bird for disease prediction. | Image uploads successfully to the Flask app. | High | Sprint-1 |
| Customer | Disease Prediction | USN-2 | As a user, I can receive the predicted disease and its confidence score after uploading an image. | Predicted disease and confidence score are displayed clearly. | High | Sprint-1 |
| Customer | Disease Information | USN-3 | As a user, I can view detailed information about the detected disease after prediction. | Disease details (symptoms, causes, effects) are shown. | High | Sprint-1 |
| Customer | Preventive Measures | USN-4 | As a user, I can view preventive care measures based on the detected disease. | A list of preventive actions specific to the disease is provided. | Medium | Sprint-2 |
| Customer | About Section | USN-5 | As a user, I can view information about the application and its AI model. | About page is accessible from the navigation bar. | Low | Sprint-2 |
| Customer | Contact Section | USN-6 | As a user, I can find contact details or feedback options for queries and assistance. | Contact info and feedback form are functional. | Medium | Sprint-2 |

**System Administrator (App Owner/ Developer)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| Administrator | Model Integration | USN-7 | As an admin, I can integrate the trained TensorFlow/Keras model into the Flask backend. | Model runs successfully via Flask routes and returns predictions. | High | Sprint-1 |
| Administrator | Deployment | USN-8 | As an admin, I can deploy the Flask app on Render or run it on a local server. | App is accessible on the selected hosting platform. | High | Sprint-1 |
| Administrator | Testing | USN-9 | As an admin, I can test image uploads and prediction outputs to ensure accurate functionality. | Image upload, prediction, and result display function correctly. | High | Sprint-1 |
| Administrator | Logs / Debugging | USN-10 | As an admin, I can view application logs in VS Code or terminal for debugging. | Debug logs are accessible and errors are traceable. | Medium | Sprint-1 |