**Project Design Phase-II**

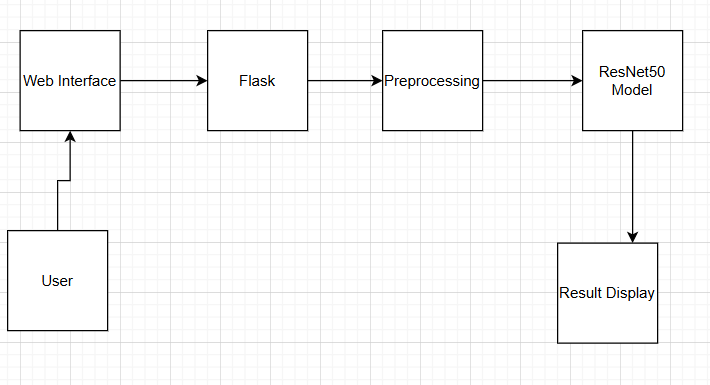
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID40145 |
| Project Name | Transfer Learning-Based Classification of Poultry Diseases for Enhanced Health Management. |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example:**

****

Guidelines:

Include all processes:

* Image upload, preprocessing, prediction, result display

Show infrastructural boundaries:

* Frontend (Local), Backend & Model (Cloud / Streamlit Cloud / Heroku)

Indicate external interfaces:

* Optional APIs for alerts (e.g., Twilio) or translation (Google API)

Highlight data storage components:

* Image and result storage (Local or Cloud File Storage)

Indicate ML model interface:

* ResNet50 model (via TensorFlow/Keras) integrated through Flask API

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for uploading poultry images and viewing predictions. | HTML, CSS, Flask. |
|  | Application Logic-1 | Image preprocessing before prediction | Python ,OpenCV |
|  | Application Logic-2 | Model loading and inference logic | TensorFlow, Keras (ResNet50) |
|  | Application Logic-3 | Display disease name | Python, Flask |
|  | Database (optional) | Storing uploaded image metadata and results (optional) | SQLite / MySQL |
|  | Cloud Database(optional) | Optional remote storage and backup | Firebase / AWS S3 |
|  | File Storage(optional) | Store uploaded poultry images locally or in cloud | Local filesystem / Google Cloud Storage |
|  | External API-1(optional) | Send notifications or alerts to farmers | Twilio API / SendGrid |
|  | External API-2(optional) | Translate or read out results in regional languages | Google Translate API / Text-to-Speech API |
|  | Machine Learning Model(optional) | Classifies poultry disease using pre-trained model | ResNet50 via Keras / TensorFlow |
|  | Infrastructure (Server / Cloud) (optional) | Hosting application and AI model | Streamlit Cloud / Heroku / AWS EC2 |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Uses widely available open-source tools and libraries | TensorFlow, Keras, Flask,OpenCV |
|  | Security Implementations | Input validation for image uploads, HTTPS enabled deployment, file type checks | SHA-256, Flask-Login, HTTPS, reCAPTCHA (optional) |
|  | Scalable Architecture | Can be containerized and deployed on cloud, supports scaling for increased traffic | Docker, Streamlit Cloud / AWS Elastic Beanstalk |
|  | Availability | Hosted on a cloud platform to ensure 24/7 access and low downtime | Streamlit Cloud / Heroku |
|  | Performance | Optimized model for fast inference; average prediction under 3–5 seconds; model cached on server | Flask, TensorFlow, Model Caching (joblib/pickle) |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)