**Project Design Phase-II**

**Data Flow Diagram & User Stories**

| Date | 27th june 2025 |
| --- | --- |
| Team ID | LTVIP2025TMID42332 |
| Project Name | Enchanted Wings: Marvels of Butterfly Species |
| Maximum Marks | 4 Marks |

## ****User Stories Table****

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| Researcher / Web User | Image Upload | USN-1 | As a user, I can upload butterfly images for prediction | I can browse and upload a valid image file | High | Sprint-1 |
| Researcher / Web User | Image Classification | USN-2 | As a user, I can receive predicted species and a confidence score | The species name and score are shown after upload | High | Sprint-1 |
| Researcher / Web User | UI Interaction | USN-3 | As a user, I see a visually pleasing result layout with image preview and prediction | The interface displays result, image, and butterfly fact | Medium | Sprint-2 |
| Researcher / Web User | Offline Use | USN-4 | As a user, I can use the system without internet access | The system works locally with all features active | High | Sprint-2 |
| Admin | Model Management | USN-5 | As an admin, I can load a trained model (MobileNet, ResNet) into the system | Model loads without crashing the app | Medium | Sprint-2 |
| Admin | View Prediction Logs | USN-6 | As an admin, I can view past user predictions with timestamp and accuracy | History displays stored records of predictions | Medium | Sprint-2 |
| Educator / Enthusiast | Educational Output | USN-7 | As a user, I can see butterfly fun facts and classification info alongside results | An interesting butterfly fact appears after prediction | Low | Sprint-2 |
| Developer | Dataset Handling | USN-8 | As a developer, I can collect and load butterfly images into training/testing folders | Images are organized into class-wise folders | High | Sprint-1 |
| Developer | Data Cleaning | USN-9 | As a developer, I can handle missing or corrupt data before training | No missing values exist before model training | High | Sprint-1 |
| Developer | Model Training | USN-10 | As a developer, I can build and train a CNN model with butterfly images | Training completes with good validation accuracy | High | Sprint-2 |

## ****Data Flow Diagrams****

## ****Level 0 DFD (Context Diagram)****

A high-level view showing external interaction between the user and system.

+-------------------+

| Butterfly |

| Classifier |

+--------+----------+

^

Upload Image | Predicted Species

+-------------+ | +----------------+

| User |<-----+------->| Web App |

+-------------+ +----------------+

### 🔷 ****Level 1 DFD (Detailed Functional Flow)****

[User]

└──► (1.0 Register/Login)

└──► [Authentication System]

└──► (2.0 Upload Butterfly Image)

└──► [Image Preprocessor]

└──► Validates Format, Resizes, etc.

└──► (3.0 Prediction Request)

└──► [CNN Model Engine]

└──► Loads MobileNetV2 / ResNet

└──► (4.0 View Output)

└──► [Result Viewer]

└──► Displays Species + Confidence

└──► (5.0 View History)

└──► [Database: MongoDB/SQLite]

└──► Fetch past predictions

**Entities Involved:**

* **Web App (Flask/Streamlit)**: Takes input, routes it to backend.
* **Model Engine**: Trained deep learning model loaded via TensorFlow/Keras.
* **Database**: Logs predictions, timestamps, species name, accuracy.
* **User Interface**: Displays predictions, images, and butterfly facts.