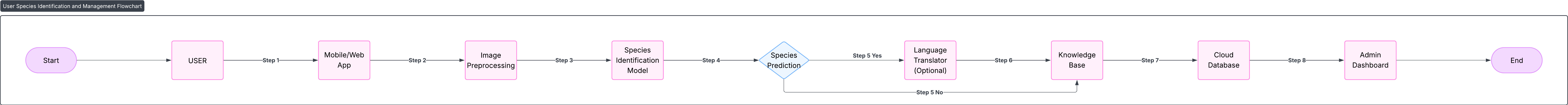
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 31 January 3035 |
| Team ID |  |
| Project Name | Enchanted wings: |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

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

****

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

| S.No | Component | Description | Technology / Example Implementation |
| --- | --- | --- | --- |
| 1 | User Interface | How user interacts with application (web/mobile app) | HTML, CSS, JavaScript, React.js, Flutter |
| 2 | Application Logic-1 | Handles user requests, API calls, and business logic | Python (Flask/Django), Node.js |
| 3 | Application Logic-2 | Image processing and species identification | TensorFlow, Keras, PyTorch, ResNet50V2 CNN[7](https://huggingface.co/KameliaZaman/Butterfly-Classification-Using-CNN/blob/a235b72ff37fe1987519960fec646b116d72ec9d/README.md)[8](https://huggingface.co/KameliaZaman/Butterfly-Classification-Using-CNN/blob/8d0a1b78791cd8d82d5ad46f4d20cdea30752176/README.md) |
| 4 | Application Logic-3 | Chatbot for butterfly info and user queries | IBM Watson Assistant, Gradio[7](https://huggingface.co/KameliaZaman/Butterfly-Classification-Using-CNN/blob/a235b72ff37fe1987519960fec646b116d72ec9d/README.md)[8](https://huggingface.co/KameliaZaman/Butterfly-Classification-Using-CNN/blob/8d0a1b78791cd8d82d5ad46f4d20cdea30752176/README.md) |
| 5 | Database | Stores user data, images metadata, and results | MySQL, MongoDB, SQLite |
| 6 | Cloud Database | Cloud-hosted database for scalability and reliability | IBM DB2, IBM Cloudant, Firebase |
| 7 | File Storage | Stores uploaded butterfly images | IBM Block Storage, AWS S3, Local Filesystem |
| 8 | External API-1 | Weather info for butterfly activity prediction | IBM Weather API, OpenWeatherMap |
| 9 | External API-2 | User verification or location services | Google Maps API, Aadhar API (if required) |
| 10 | Machine Learning Model | Recognizes butterfly species from images | CNN (ResNet50V2, MobileNet, KNN Classifier)[1](https://itnext.io/butterflix-349a27ab3733)[7](https://huggingface.co/KameliaZaman/Butterfly-Classification-Using-CNN/blob/a235b72ff37fe1987519960fec646b116d72ec9d/README.md)[8](https://huggingface.co/KameliaZaman/Butterfly-Classification-Using-CNN/blob/8d0a1b78791cd8d82d5ad46f4d20cdea30752176/README.md) |
| 11 | Infrastructure | Deployment platform (local/cloud, containerized) | Cloud Foundry, Kubernetes, Docker, Local Server |

## Table-2: Application Characteristics

| S.No | Characteristics | Description | Technology / Implementation Examples |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Frameworks and libraries used | TensorFlow, Keras, Flask, React.js, Gradio, PyTorch |
| 2 | Security Implementations | Authentication, encryption, access control, firewalls | JWT Auth, HTTPS/SSL, SHA-256, IAM, OWASP practices |
| 3 | Scalable Architecture | Microservices or 3-tier, load balancing, horizontal scaling | Microservices, Docker, Kubernetes, Cloud Load Balancer |
| 4 | Availability | Measures for high uptime (redundancy, failover, distributed servers) | Cloud DB, Multi-zone deployment, Load Balancer |
| 5 | Performance | Caching, CDN, optimized queries, concurrency, response time | Redis Cache, CDN, Efficient ML inference, Async APIs |

Notes:

[**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)