Project Design Phase-II
Technology Stack (Architecture & Stack)

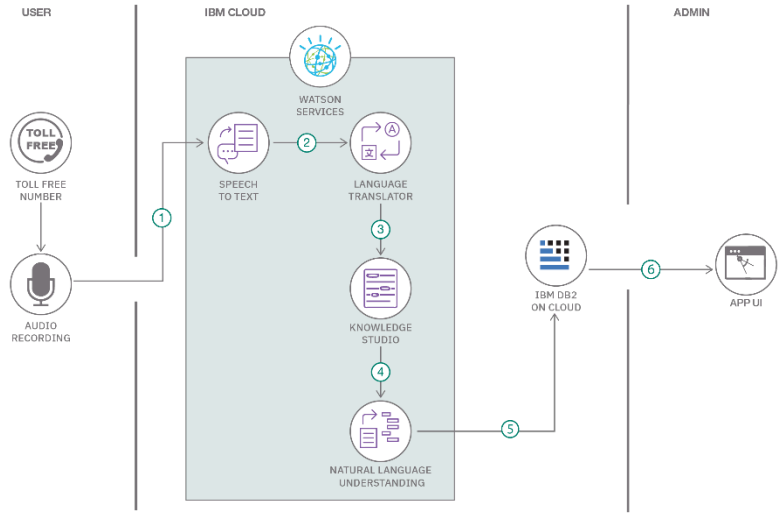
Date	16 July 2025
Team ID	LTVIP2025TMID32540
Project Name	Transfer learning based classification of poultry diseases for enhanced Health management
Maximum Marks	4 Marks

Technical Architecture:

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

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Interactive dashboard for customers, customer care executives, and admin to access features	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Frontend logic: routing, state management, form handling, chart display	React Router, Redux / Context API, Recharts / Chart.js, Axios
3.	Application Logic-2	Backend API handling user login, dashboard data, ticket creation	Node.js, Express.js, JWT for authentication
4.	Application Logic-3	Machine Learning API service to classify poultry disease from image	Python, FastAPI or Flask, connects to ML model
5.	Database	Stores structured data: users, tickets, prediction logs	PostgreSQL
6.	Cloud Database	Cloud-hosted version of PostgreSQL for scalability and availability	AWS RDS, Google Cloud SQL, Azure PostgreSQL
7.	File Storage	Stores uploaded poultry images for prediction	SendGrid, Mailgun, or SMTP
8.	External API-1	Sends email notifications for prediction results and ticket updates	Tawk.to, Firebase Cloud Messaging, OneSignal
9.	External API-2	Chat integration or push notifications for customer support	TensorFlow / Keras, or PyTorch, using MobileNet / ResNet
10.	Machine Learning Model	Transfer learning model for classifying poultry diseases (trained offline, deployed via API)	AWS EC2 / GCP Compute Engine, Docker, GitHub Actions, NGINX
11.	Infrastructure (Server / Cloud)	Hosts UI, backend, model APIs, and database (CI/CD + Monitoring)	AWS EC2 / GCP Compute Engine, Docker, GitHub Actions, NGINX

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List of frameworks used to develop the application frontend, backend, and ML model	React.js, Express.js, Node.js, FastAPI / Flask, TensorFlow, PyTorch, PostgreSQL, Docker
2.	Security Implementations	Security features and access control mechanisms across frontend, backend, and deployment	- JWT (authentication) - HTTPS / TLS encryption - SHA-256 password hashing - IAM Roles (AWS/GCP) - CORS - OWASP Top 10 guidelines
3.	Scalable Architecture	Designed using modular components and services that can scale independently	- Microservices-based (ML API, Web API, Frontend) - 3-Tier Architecture - Dockerized Containers - Kubernetes (optional)
4.	Availability	Redundancy and uptime measures to ensure high availability of services	- Load Balancer (AWS ELB, NGINX) - Distributed servers - Auto-scaling Groups - Cloud Database (e.g. AWS RDS Multi-AZ)
5.	Performance	Application optimized for speed and efficient handling of requests	- Redis (caching) - CDN (CloudFront / Cloudflare) - Asynchronous processing (e.g., background model inference) - Optimized ML model (e.g., MobileNet) - Request rate: Designed for 50–100 req/sec

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d