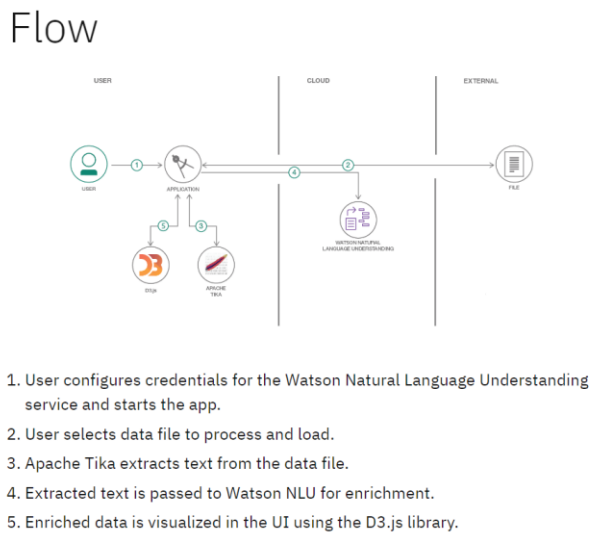
Project Design Phase-II
Data Flow Diagram & User Stories

Date	28 June 2025
Team ID	LTVIP2025TMID41808
Project Name	Grainpalette - a deep learning odyssey in rice type classification through transfer learning
Maximum Marks	4 Marks

Data Flow Diagrams:

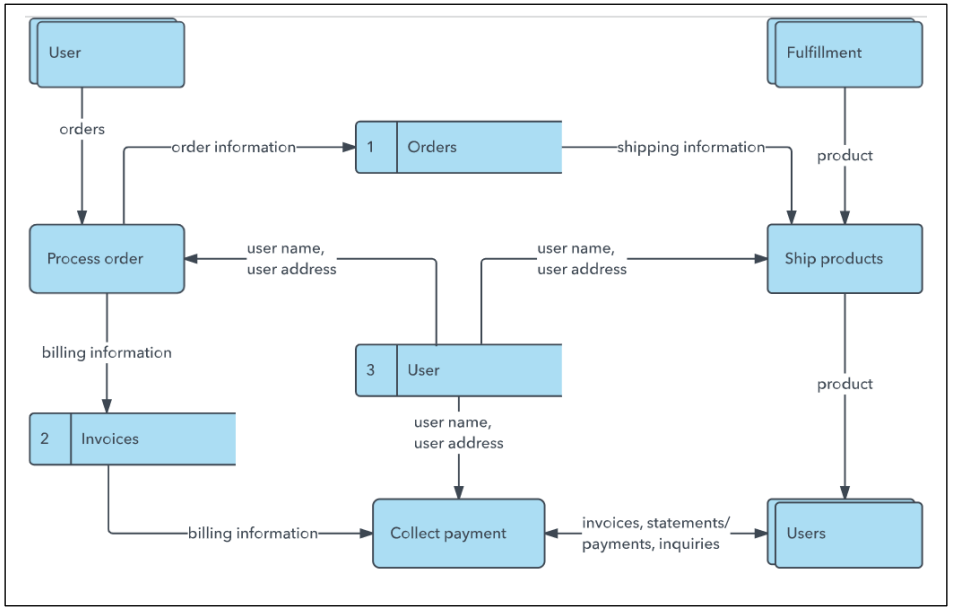
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)



1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

Example: DFD Level 0 (Industry Standard)



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Quality Inspector)	Image Upload	USN-1	As a user, I can upload a rice grain image to the system	I can upload an image successfully	High	Sprint-1
Quality Inspector	Classification	USN-2	As a user, I can get the rice grain type predicted by the model	I can view the predicted class after image is processed	High	Sprint-1
Quality Inspector	Confidence Score Display	USN-3	As a user, I can see a confidence score for each prediction	I can view prediction with percentage confidence value	Medium	Sprint
Quality Inspector	Report Generation	USN-4	As a user, I can download a report with prediction results	I can download a simple report with grain type, image, and confidence	Medium	Sprint-3
Admin (Dev User)	Dataset Management	USN-5	As an admin, I can upload and manage the rice grain dataset used for training	I can upload new dataset and trigger training if needed	Low	Sprint-4
Admin (Dev User)	Model Evaluation	USN-6	As an admin, I can view training/validation performance of the model	I can see accuracy, loss, and confusion matrix	Medium	Sprint-3

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	28 June 2025
Team ID	LTVIP2025TMID41808
Project Name	Grainpalette - a deep learning odyssey in rice type classification through transfer learning
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image Upload	Upload image through web interface (Streamlit/Notebook)
FR-2	Preprocessing	Resize and normalize image Apply data augmentation
FR-3	Classification	Predict rice grain type using trained DL model
FR-4	Prediction Output	Show predicted rice type and confidence score
FR-5	Report Generation	Download prediction report with class and score
FR-6	Dataset Management(Admin)	Upload new dataset Trigger model re-training

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The UI must be simple, clean, and user-friendly for non-technical users
NFR-2	**Security**	The system should validate image types and prevent any malicious file uploads
NFR-3	**Reliability**	The system should produce consistent results across repeated predictions
NFR-4	**Performance**	Model should provide predictions within 2–3 seconds per image
NFR-5	**Availability**	The system should be accessible online (via Notebook or Web) during working hours
NFR-6	**Scalability**	The architecture should allow future expansion for more grain types or crops

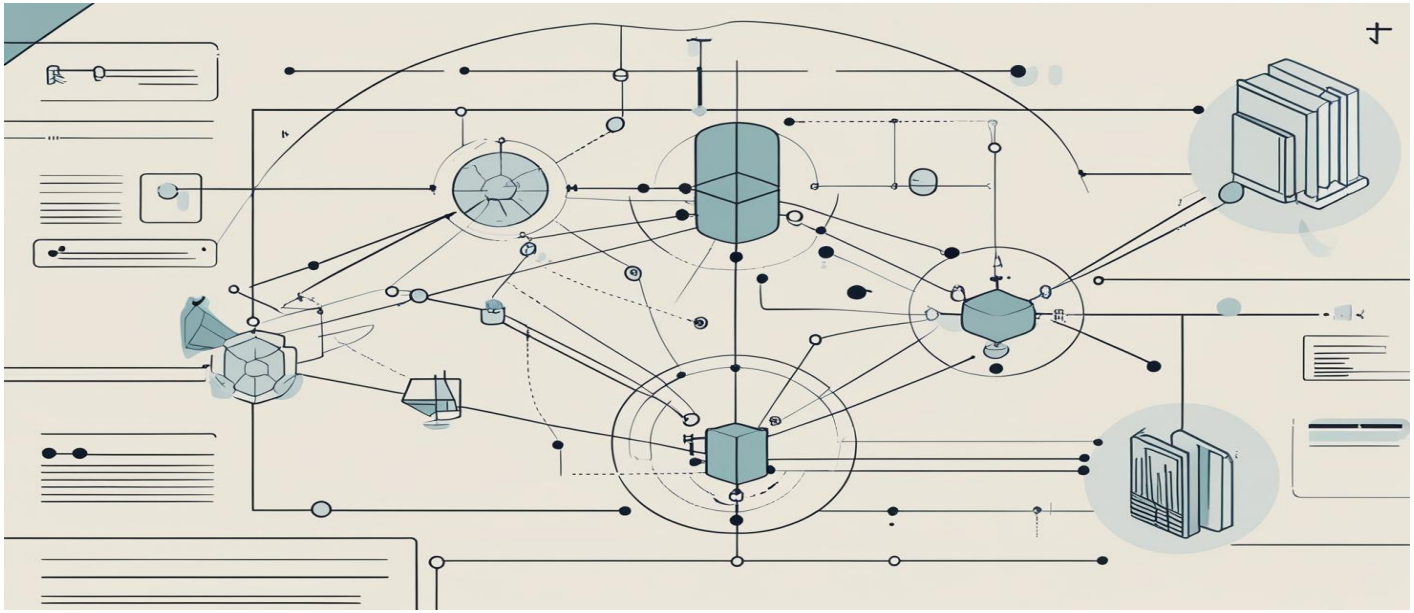
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	28 June 2025
Team ID	LTVIP2025TMID41808
Project Name	Grainpalette - a deep learning odyssey in rice type classification through transfer learning
Maximum Marks	4 Marks

Technical Architecture:

The technical architecture of **GrainPalette** is modular, scalable, and built on modern deep learning tools. It ensures smooth data flow, seamless user interaction, and efficient model performance using a lightweight deployment environment.

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



S.No	Component	Description	Technology
1	User Interface	Web interface to upload rice images and display predictions	Streamlit / Kaggle Notebook UI
2	Application Logic-1	Image preprocessing logic: resizing, normalization	Python (OpenCV, PIL)
3	Application Logic-2	Prediction logic using pre-trained model	Python (TensorFlow / Keras)
4	Application Logic-3	Report generation and formatting prediction output	Python (Pandas, Matplotlib)
5	Database	(Optional) Store prediction history or logs	CSV/JSON files (No DB used in current setup)
6	Cloud Database	Not used in this prototype phase	—
7	File Storage	Stores input images and downloadable result files	Local File System / Kaggle File Manager
8	External API-1	(Optional) Load dataset from external source	Kaggle Dataset API
9	External API-2	Not used	—
10	Machine Learning Model	Transfer Learning model for rice type classification	MobileNetV2 (TensorFlow / Keras)
11	Infrastructure (Server / Cloud)	Hosted on Kaggle Notebook (GPU) or Local Streamlit App	Kaggle Cloud Notebook / Localhost

Table-2: Application Characteristics:

S.No	Component	Description	Technology
1	User Interface	Web interface to upload rice images and display predictions	Streamlit / Kaggle Notebook UI
2	Application Logic-1	Image preprocessing logic: resizing, normalization	Python (OpenCV, PIL)
3	Application Logic-2	Prediction logic using pre-trained model	Python (TensorFlow / Keras)
4	Application Logic-3	Report generation and formatting of prediction output	Python (Pandas, Matplotlib)
5	Database	(Optional) Store prediction logs / inputs / results	CSV / JSON files (local file system)

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