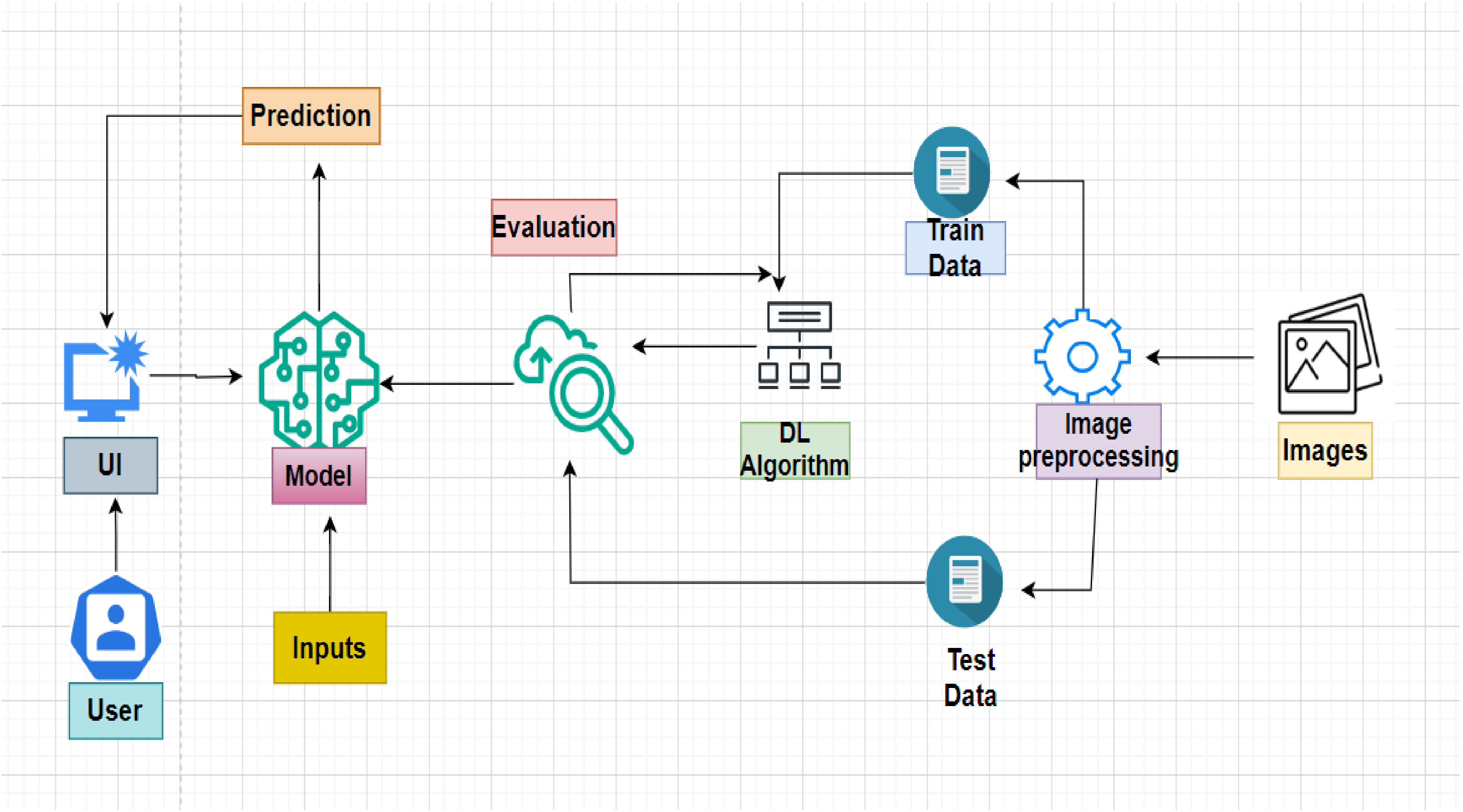
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	12-11-2023
Team ID	**Team-591719**
Project Name	Project -Image Caption Generation
Maximum Marks	4 Marks

Technical Architecture:

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

The **Image Captioning project** leverages deep learning technologies, employing Convolutional Neural Networks (**CNN**s) for image feature extraction and Long Short-Term Memory (**LSTM**) networks for sequential language generation. The user interacts through a user-friendly web interface built with HTML, CSS, and JavaScript, while the application's logic is implemented in Python. Data storage involves both traditional databases, like MySQL, and cloud-based solutions, such as IBM Cloudant. The overall architecture ensures scalability, security, and performance, making it capable of generating creative and relevant captions for uploaded images.


Prediction
Evaluation
Train Data
UI
Model
DL Algorithm
Image preprocessing
Images
Test Data
User
Inputs

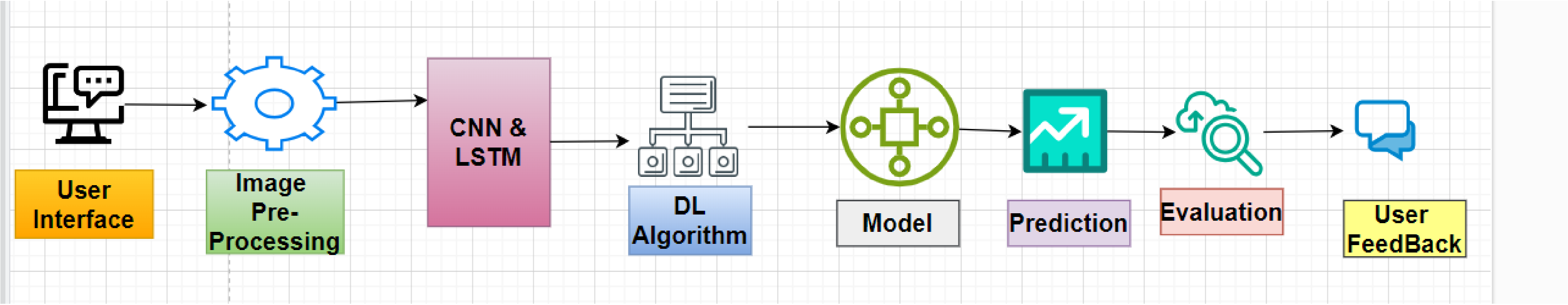


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How users interact with the image captioning application (UI/UX design)	Web UI, Mobile App, HTML, CSS, JavaScript / React Js
2.	Image Preprocessing	Preprocesses images before feeding to the model	OpenCV, PIL
3.	CNN & LSTM Model	Deep learning model for image captioning	TensorFlow, PyTorch
4.	Image Caption Generation Logic	Logic for generating captions based on the input images	Python, Natural Language Processing libraries

5.	Database	Storage for data related to images and generated captions	MySQL, NoSQL (e.g., MongoDB)
6.	Cloud Database	Cloud-based database service for scalability	IBM Cloudant, AWS DynamoDB
7.	File Storage	Storage for image files	Cloud Object Storage, Local Filesystem
8.	External API-1	Integration with external services for additional data	IBM Watson Visual Recognition API, etc.
9.	External API-2	Integration with external services for additional data	Language Translation API, etc.
10.	User Feedback	Collects feedback from users on generated captions	Web UI, Mobile App
11.	Infrastructure (Server / Cloud)	Deployment and hosting of the image caption generation application	Local Server: Local machine, Cloud Server: IBM Cloud, AWS, Azure, Kubernetes
12	Authentication	User authentication and authorization	OAuth, JWT, Firebase Authentication
13	CDN	Content Delivery Network for faster image loading	Cloudflare, Akamai, Amazon CloudFront

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Utilization of open-source frameworks for the image captioning model	PyTorch, TensorFlow, Flask, Django
2.	Security Implementations	Implementation of security measures for user data protection, e.g., encryption, secure APIs	SSL/TLS, OAuth, JWT, IAM Controls, Hashing
3.	Scalable Architecture	Design considerations ensuring scalability to handle a growing number of users and images	Docker, Kubernetes, Load Balancers
4.	Availability	Strategies ensuring high availability of the image captioning service, e.g., server redundancy	Load Balancers, Redundancy, Failover
5.	Performance	Design considerations for optimal performance in caption generation, handling multiple requests	Caching, CDNs, Efficient Algorithms