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

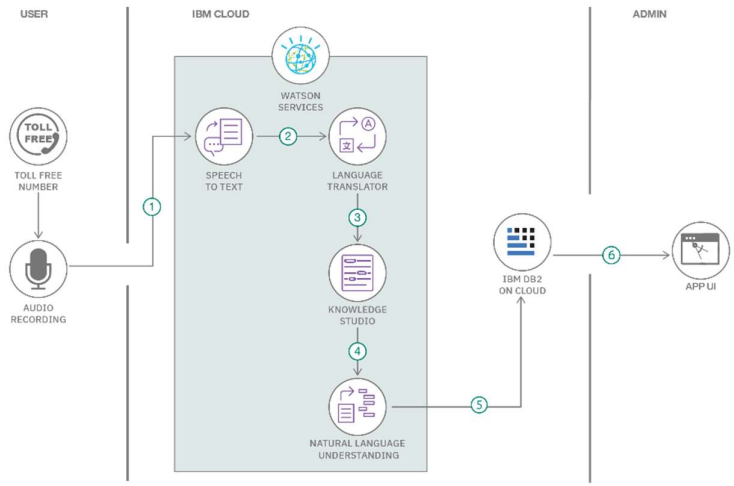
Date	03 October 2022
Team ID	PNT2022TMID06155
Project Name	Project-A Gesture-based Tool for Sterile Browsing of Radiology Images
Maximum Marks	4 Marks

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:

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

Technical Architecture:

-

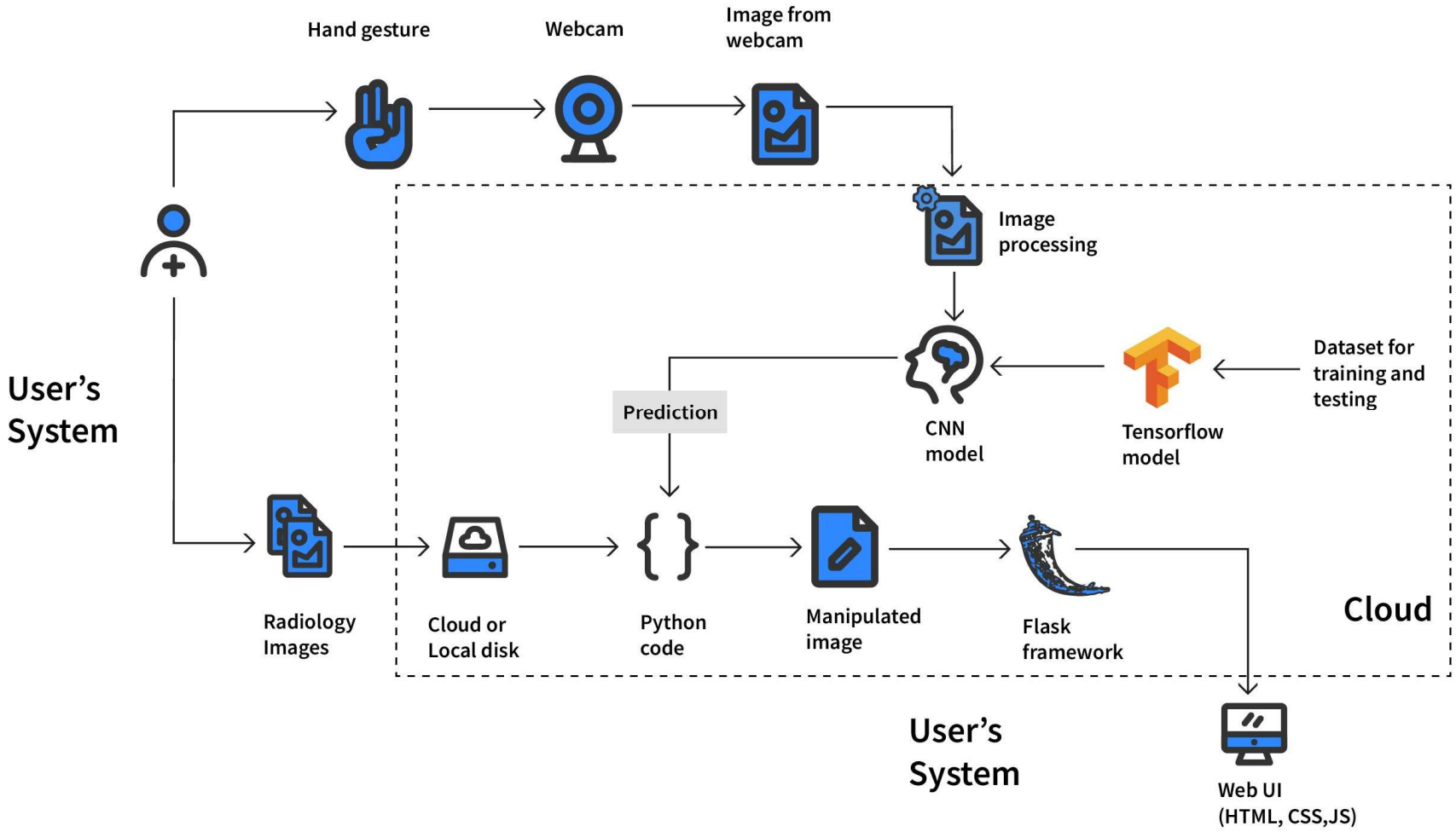


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User can upload radiology images, configure some settings and can give hand gesture input through webcam.	HTML, CSS, JavaScript
2.	Application Logic-1 Converting video into frames.	Using OpenCV the input video is converted into frames.	Python, OpenCV
3.	Application Logic-2 Image processing	The image is cropped as per the region of interest, background is eliminated by thresholding and converted into binary image	Python, OpenCV
4.	Application Logic-3 Gesture prediction	The processed gesture image is predicted as one finger, two finger, etc., using a CNN model	Tensorflow, Keras
5.	Database	We can use MySQL or JSON file to store the user settings and configurations.	MySQL, NoSQL, JSON file etc.
6.	Cloud Database	Clous database is used for model testing, serving app, storing credentials, etc.,	IBM DB2 or IBM Cloudant or other database etc.
7.	File Storage	File storage requirements.	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Machine Learning Model	To predict the given processed hand gesture image	Object Recognition Model, Image Classification Model, etc.
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Flask framework creates a local server Cloud Server Configuration : IBM cloud server configured to start from the web UI.	Local, IBM cloud, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask, OpenCV	Flask – a microweb framework used to integrate python with web UI, OpenCV – A computer vision library it gives some utility functions for accessing the webcam and image processing functions
2.	Scalable Architecture	We can use JS frameworks in the future when the app grows and with Flask web UI templates can be easily changed. We can add more gesture inputs for performing actions on the radiology image.	OpenCV, Modern javascript, Flask HTML templates
3.	Availability	This app can be available to everyone through the cloud. And this can be executed in the local server environment also.	IBM cloud, Local server from Flask framework.
4.	Performance	We can store the radiology images in the cache or session storage to increase the performance and to reduce the delay.	Browser session or local storage, Cookies and server cache.

References:

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

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

https://docs.opencv.org/4.x/d9/df8/tutorial_root.html

https://keras.io/api/

https://flask.palletsprojects.com/en/2.2.x/

https://developer.mozilla.org/en-US/docs/Web/JavaScript