

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

| | |
|---------------|---|
| Date | 14 October 2022 |
| Team ID | PNT2022TMID00195 |
| Project Name | Project – A Gesture-based Tool for Sterile Browsing of Radiology Images |
| Maximum Marks | 4 Marks |

Technical Architecture:

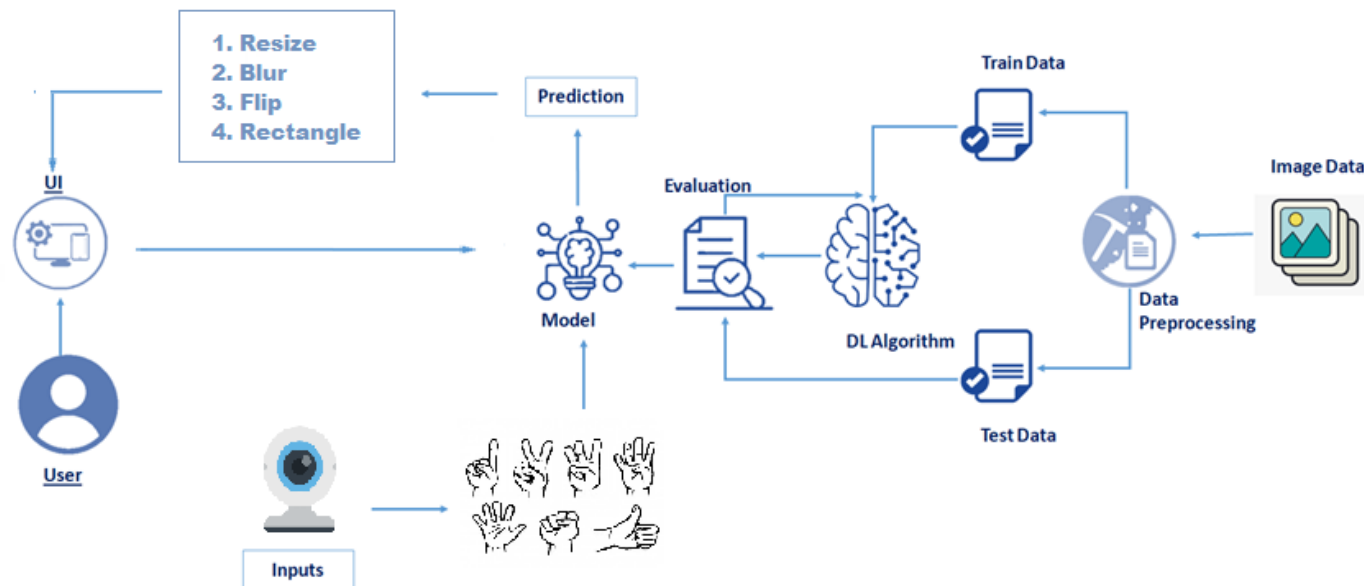


Table-1 : Components & Technologies:

| S.No | Component | Description | Technology |
|------|---------------------------------|---|--|
| 1. | User Interface | How user interacts with application e.g. Web UI | HTML, CSS, JavaScript |
| 2. | Application Logic-1 | Logic for a process in the application | Python |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson STT service |
| 4. | Application Logic-3 | Logic for a process in the application | IBM Watson Assistant |
| 5. | Database | Data Type, Configurations etc. | MySQL |
| 6. | Cloud Database | Database Service on Cloud | IBM DB2 |
| 7. | File Storage | File storage requirements | Local Filesystem |
| 8. | External API | Purpose of External API used in the application | Aadhar API |
| 9. | Machine Learning Model | Purpose of Machine Learning Model | Deep Neural Networks |
| 10. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud | Local Server Configuration: Local System Cloud Server Configuration: IBM Watson |

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|--|----------------------------------|
| 1. | Open-Source Frameworks | List the open-source frameworks used | Flask, Scikit learn, Tensor flow |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | Encryptions, Decryptions |

| S.No | Characteristics | Description | Technology |
|------|-----------------------|---|---|
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | MySQL – As it can store huge amount of data |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | IBM Watson – Can easily be accessed |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Flask – Handle multiple requests |