

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

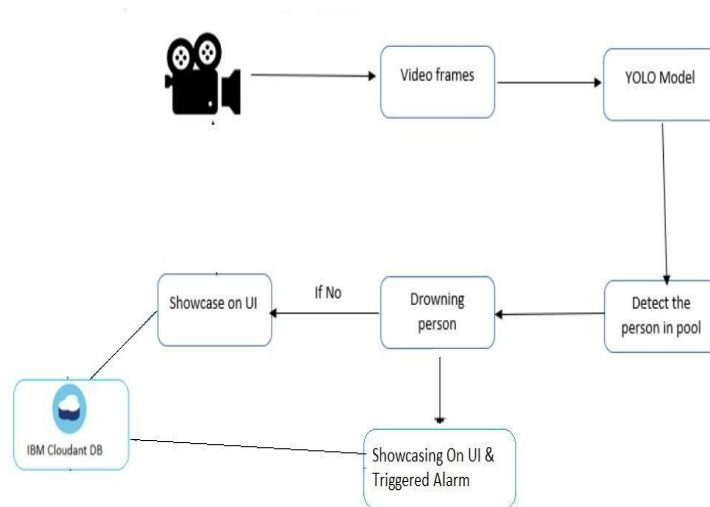
|               |  |
|---------------|--|
| Date          | 14 October 2022  |
| Team ID       | PNT2022TMID29283   |
| Project Name  | Project – VIRTUALEYE-LIFEGUARD FOR SWIMMING POOL TO DETECT ACTIVE DROWNING |
| Maximum Marks | 4 Marks  |

### 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:

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

**Table-1 : Components & Technologies:**

| S.No | Component                       | Description   | Technology   |
|------|---------------------------------|---|--|
| 1.   | User Interface                  | How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.                                     | HTML, CSS, JavaScript / Angular Js / React Js etc.             |
| 2.   | Application Logic-1             | Logic for a process in the application  | Java / 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, NoSQL, etc.   |
| 6.   | Cloud Database                  | Database Service on Cloud   | IBM DB2, IBM Cloudant etc.                                     |
| 7.   | File Storage                    | File storage requirements   | IBM Block Storage or Other Storage Service or Local Filesystem |
| 8.   | External API-1                  | Purpose of External API used in the application   | IBM Weather API, etc.  |
| 9.   | External API-2                  | Purpose of External API used in the application   | Aadhar API, etc.   |
| 10.  | Machine Learning Model          | Purpose of Machine Learning Model   | Object Recognition Model, etc.                                 |
| 11.  | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud<br>Local Server Configuration:<br>Cloud Server Configuration : | Local, Cloud Foundry, Kubernetes, etc.                         |

**Table-2: Application Characteristics:**

| S.No | Characteristics          | Description   | Technology  |
|------|--------------------------|---|---|
| 1.   | Open-Source Frameworks   | List the open-source frameworks used  | Technology of Opensource framework                  |
| 2.   | Security Implementations | List all the security / access controls implemented, use of firewalls etc.  | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
| 3.   | Scalable Architecture    | Justify the scalability of architecture (3 – tier, Micro-services)  | Technology used                                     |
| 4.   | Availability             | Justify the availability of application (e.g. use of load balancers, distributed servers etc.)                            | Technology used                                     |
| 5.   | Performance              | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Technology used                                     |

#### 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>