**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 13 November 2023 |
| Team ID | Team-592189 |
| Project Name | Lip Reading using Deep Learning |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

Lip-reading, also known as visual speech recognition (VSR), is the ability to understand spoken language by observing the movements of the lips, tongue, and jaw. Lip-reading is a complex task that requires the ability to recognize and interpret subtle visual cues.

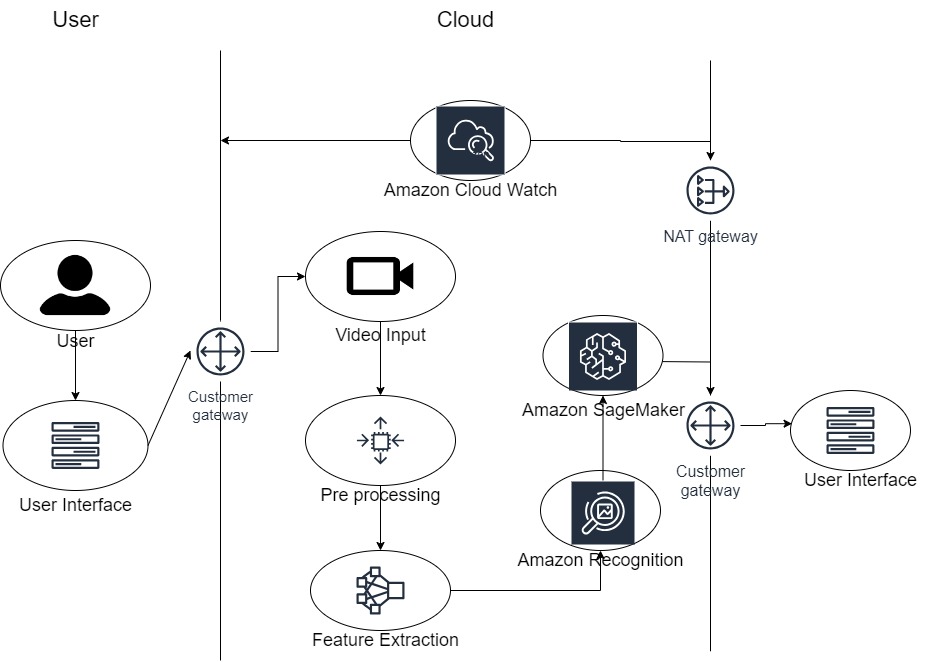
In recent years, there has been significant progress in the development of lip-reading systems using deep learning techniques. These systems typically involve two main components: a feature extractor and a sequence model.

The feature extractor is responsible for extracting relevant visual cues from the input video sequence. This may involve identifying and tracking facial landmarks, extracting lip shape features, or analyzing optical flow patterns.

The sequence model is responsible for interpreting the extracted visual features and predicting the corresponding spoken words or phrases. This may involve using recurrent neural networks (RNNs), long short-term memory (LSTM) networks, or other deep learning architectures.

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

**Reference:** <https://drive.google.com/drive/my-drive>



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)

**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 / Node 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 | AWS Cloud |
| 4. | Application Logic-3 | Logic for a process in the application | AWS Cloud Watch |
| 5. | Database | Data Type, Configurations etc. | Dynamo DB, Amazon Aurora, .etc.. |
| 6. | Cloud Database | Database Service on Cloud | Dynamo DB, Amazon Aurora, .etc.. |
| 7. | File Storage | File storage requirements | Dynamo DB, RDS, Amazon S3 ..etc |
| 8. | External API-1 | Purpose of External API used in the application | Security of User data VPN security. |
| 9. | Machine Learning Model | Purpose of Machine Learning Model | Object Recognition Model, etc. |
| 10. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud Local Server Configuration:  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, AWS Controls, OWASP etc. |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Technology used |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 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://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>

[https://aws.amazon.com/architecture/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=\*all&awsf.methodology=\*all&awsf.tech-category=\*all&awsf.industries=\*all&awsf.business-category=\*all](https://aws.amazon.com/architecture/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=*all&awsf.methodology=*all&awsf.tech-category=*all&awsf.industries=*all&awsf.business-category=*all)

<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>