**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 13 October 2022 |
| Team ID | PNT2022TMID21704 |
| Project Name | Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

Graphical user interface, application, Word

Description automatically generated

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | A web application using Flask. | HTML, CSS, JavaScript |
|  | Image augmentation | Creating training examples from the existing ones. | Python |
|  | Building the model | Building the model by adding various layers and training the model | IBM Watson Cloud storage,IBM cloud Pak Data |
|  | Testing the model | Testing the model | IBM Watson Machine Learning |
|  | Database | Database contains the ECG signals of various arrythemia patients | IBM cloud storage |
|  | Cloud Database | Database Service on Cloud | IBM DB2. |
|  | File Storage | File storage requirements | IBM Block Storage |
|  | Anaconda Navigator | The web application is deployed and run on the local host with the help of anaconda navigator. | Anaconda Navigator |
|  | Tensorflow | For numerical computation that makes machine learning and developing neural networks | Tensorflow library, Jupyter Notebook |
|  | Keras | To make high level neural network API | Keras library, Jupyter Notebook |
|  | Machine Learning Model | To classify the type of arrythemia with Images uploaded by the users. | Image Classification Model |
|  | Infrastructure (Server ) | Application Deployment on Local System | Local |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Flask, Jupyter Notebook | Flask, Jupyter Notebook |
|  | Security Implementations | use of API key | IAM Controls,wml control(API Key) |
|  | Scalable Architecture | Micro-services | API gateway |
|  | Availability | Based on availability zones | IBM Watson studio |
|  | Performance | Using CDNs. | IBM Content Delivery Network. |