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

Date	16 October 2022
Team ID	PNT2022TMID01315
Project Name	Classification of Arrhythmia By Using Deep Learning with 2-D ECG Spectral Image Representation
Maximum Marks	4 Marks

Technical Architecture:

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

PREDICTION

UI

TRAIN DATA

DL ALGORITHM

IMAGE DATA

MODEL

Evaluation

DATA PREPROCESSING

INPUTS

TEST DATA
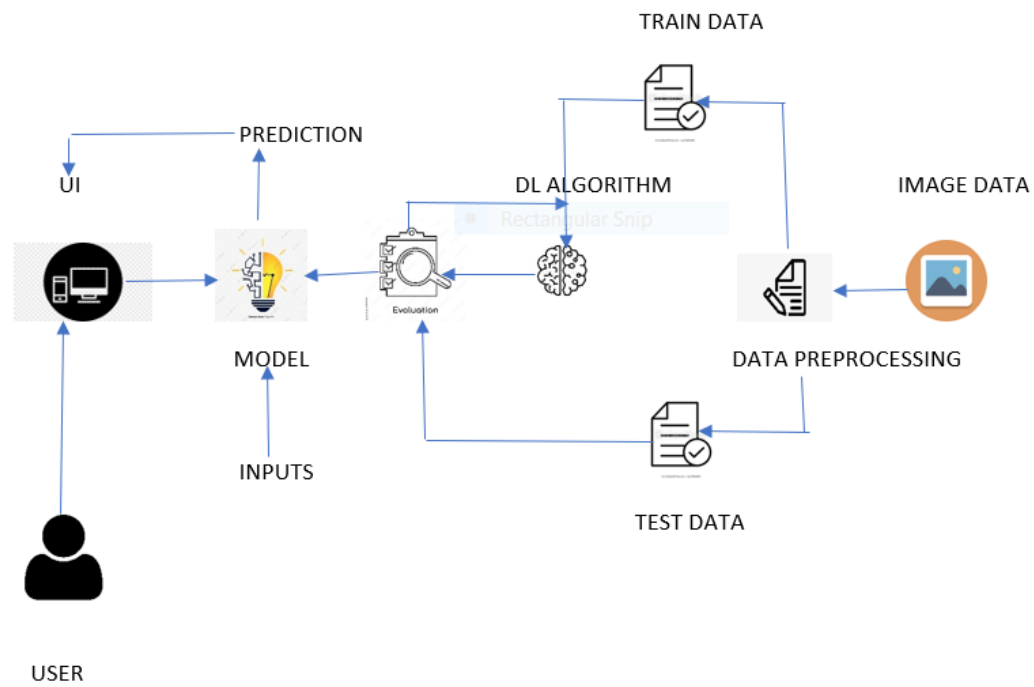
USER

Table-1 : Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application to upload image	HTML, CSS, Java Script, python
2.	Model analysis	After that Prediction is showcased on the UI	CNN Model
3.	Data collection	Create the Dataset	Python, Keras, numpy
4.	Data Preprocessing-1	Import the Library	Python, Keras, numpy
5.	Data Preprocessing-2	Configure the class	Python, Keras, numpy
6.	Data Preprocessing-3	Apply Image Data Generator functionality to Trainset and Testset	Python, Keras, numpy
7.	Model Building-1	Import the model building libraries and initialise the model	Python, Keras, numpy
8.	Model Building-2	Adding layers and configure	Python, Keras, numpy
9.	Model Building-3	Training and testing the model	Python, Keras, numpy
10.	Application Building	Building python code	HTML, Python
11.	Train the model on IBM	CNN development	IBM Watson.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open source software is by which the source code or base code is usually available for modification	Flask(python)
2.	Security Implementations	The WAF is able to protect against attacks like cross site forgery, cross site scripting and SQL injection	e. g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Though many users used, doesn't affect the performance	CNN Model
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	HTML, CSS, Javascript, Python
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	HTML, CSS, Javascript, Python