# Project Planning Phase

## Project Planning Template

| | |
|---|---|
| Team ID | LTVIP2023TMID07965 |
| Project Name | Classification Of Arrhythmia By Using Deep Learning With 2-D ECG Spectral Image Representation |

**Product Backlog, Sprint Schedule, and Estimation**

Use the below template to create product backlog and sprint schedule

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Download The Dataset | USN-1 | We can download the Dataset contains Six classes | 1 | Low | Kavya.G<br>Vasantha.L |
| Sprint-1 | Import The ImageDataGenerator Library | USN-2 | We can import ImageDataGenerator | 1 | Low | Kavya.G<br>Gayatri |
| Sprint-1 | Configure ImageDataGenerator class | USN-3 | We can configure the ImageDataGenerator class | 1 | Low | Gayatri<br>Kavya.G |
| Sprint-2 | Apply the ImageDataGenerator | USN-4 | We can apply ImageDataGenerator to train dataset | 2 | Medium | Vetriselvan.PL<br>Bharathidasan.R |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| | functionality to Trainset and Dataset | | | | | Vasantha.L |
| Sprint-1 | Import Libraries | USN-5 | We can import required Libraries | 1 | Low | Vasantha.L |
| Sprint-1 | Initialize the Model | USN-6 | Initializing the Image recognition model | 2 | Medium | Kavya.G<br>Gayatri<br>Vasantha.L |
| Sprint-4 | Adding CNN layer | USN-7 | We can add Convolutional Neural Network(CNN) used for image/object recognition and classification | 4 | High | Kavya.G<br>Gayatri |
| Sprint-4 | Adding Dense Layer | USN-8 | We can add Dense Layer in which each neuron receives input from all the neurons of previous layer | 4 | High | Kavya.G<br>Gayatri<br>Vasantha.L |
| Sprint-4 | Configure The Learning Process | USN-9 | We can configure The Learning process which is a method, mathematical logic or algorithm that improves the network's performance and/or training time. | 4 | High | Kavya.G<br>Gayatri<br>Vasantha.L |
| Sprint-3 | Train the Model | USN-10 | We can train our model with our image dataset. fit_generator functions used to train a deep learning neural network | 3 | High | Vasantha.L<br>Kavya.G |
| Sprint-1 | Save the Model | USN-11 | We can save The model with .h5 extension | 2 | Medium | Gayatri<br>Vasantha.L |
| Sprint-2 | Test the model | USN-12 | We can Test the model through Loaded necessary libraries, the saved model | 2 | Medium | Kavya.G<br>Gayatri |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-3 | Create Html files | USN-13 | We use HTML to create the front end part of the web page. | 3 | High | Kavya.G<br>Gayatri<br>Vasantha.L |
| Sprint-4 | Build Python code | USN-14 | We build the flask file ‘app.py’ which is a web framework written in python for server-side scripting. | 4 | High | Kavya.G<br>Gayatri<br>Vasantha.L |
| Sprint-1 | Run the App | USN-15 | We can run the App | 2 | Medium | Vasantha.L<br>Kavya.G |
| Sprint-2 | Register IBM Cloud | USN-16 | We can register IBM Cloud | 2 | Medium | Vasantha.L<br>Gayatri |
| Sprint-3 | Train the model on IBM | USN-17 | We can Train Out model on IBM | 3 | High | Kavya.G<br>Gayatri |

**Velocity:**
To calculate the team's **average velocity (AV)** per iteration unit

$$Av = \frac{Velocity}{Sprint\ duration}$$

Where,

**Average Velocity** - Story points per day
**Sprint duration** - Number of days (Duration) for Sprints
**Velocity** - Points per Sprint

$$Av = \frac{20}{5} = 5$$

Average Velocity is **4** points per Sprint

**Burndown Chart:**
A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

## Burndown Chart