**Project Planning Phase**

**Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)**

|  |  |
| --- | --- |
| Date | 18 October 2022 |
| Team ID | PNT2022TMID21704 |
| Project Name | Project - Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation |
| Maximum Marks | 8 Marks |

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

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 | Dataset | USN-1 | Downloaded the dataset. | 2 | High | Sashrutha.M |
| Sprint-1 |  | USN-2 | Image preprocessing | 1 | High | Ramya.R |
| Sprint-1 |  | USN-3 | Import the ImageDataGenerator library | 2 | Low | Shantha Kumar.S |
| Sprint-1 |  | USN-4 | Configure ImageDataGenerator class | 2 | Medium | Swetha.B |
| Sprint-1 |  | USN-5 | Apply ImageDataGenerator functionality to trainingset and testingset. | 1 | High | Sashrutha.M |
| Sprint -2 |  | USN-6 | Import the libraries | 1 | Medium | Shantha Kumar.S |
|  |  | USN-7 | Initialise the model | 2 | Low | Ramya.R |
|  |  | USN-9 | Add the CNN layers | 2 | High | Swetha.B |
|  |  | USN-10 | Add Dense layers | 2 | Low | Shantha Kumar.S |
|  |  | USN-11 | Configure the learning process | 1 | Low | Ramya.R |
|  |  | USN-12 | Train the model | 3 | High | Sashrutha.M |
|  |  | USN-13 | Save the model | 2 | Medium | Shantha Kumar.S |
|  |  | USN-14 | Test the model | 3 | High | Sashrutha.M |
| Sprint-3 |  | USN-15 | Create HTML files | 2 | Medium | Swetha.B |
|  |  | USN-16 | Build the python Code | 2 | High | Sashrutha.M |
|  |  | USN-17 | Run the App | 2 | Medium | Ramya.R |
| Sprint-4 |  | USN-18 | Create the IBM cloud account | 2 | Medium | Swetha.B |
|  |  | USN-19 | Train the model on IBM Watson | 4 | High | Sashrutha.M |

**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 8 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 16 | 06 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 6 | 14 Nov 2022 |
| Sprint-4 | 20 | 3 Days | 13 Nov 2022 | 15 Nov 2022 | 6 | 16 Nov 2022 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Velocity:**

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)



**Burndown Chart:**

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile[software development](https://www.visual-paradigm.com/scrum/what-is-agile-software-development/) methodologies such as [Scrum](https://www.visual-paradigm.com/scrum/scrum-in-3-minutes/). However, burn down charts can be applied to any project containing measurable progress over time.

