Embedded Machine Learning for Early Detection of Heart Attack Symptoms

A PROJECT REPORT

Submitted by

VIVEK M G [Reg No: RA2211003010002]

RAGHURAM SRIKANTH [Reg No: RA2211003010218]

Under the Guidance of

Dr. Jeya R

Associate Professor, Department of Computing Technologies

in partial fulfillment of the requirementsfor the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

MAY 2025

Department of Computing Technologies

SRM Institute of Science & Technology

Own Work Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

<u>To be completed by the student for all assessments</u>

Degree/ Course : B Tech Computer Science & Engineering

Student Name : VIVEK M G , RAGHURAM SRIKANTH

Registration Number : RA2211003010002 , RA2211003010218

Title of Work : Embedded Machine Learning for Early Detection of Heart Attack Symptoms

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above RA2211003010002 – Vivek 28/04/25 RA2211003010218 – S.Raghuram. 28/04/25
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 21CSP302L - Project report titled "**EMBEDDED MACHINE LEARNING FOR EARLY DETECTION OF HEART ATTACK SYMPTOMS**" is the bonafide work of "**VIVEK M G [Reg No: RA2211003010002] , RAGHURAM SRIKANTH [Reg No: RA2211003010218]**" who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

28/4/25

SIGNATURE
Dr. Jeya R
SUPERVISOR
Associate Professor,
Department of
Computing Technologies

SIGNATURE
Dr. Niranjana G
Professor & Head,
Department of
Computing Technologies

Department of Computing Technologies
Kattankulathur
603 203
SRM IST

Examiner - 1

Examiner - 2

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M,** Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. C Lakshmi**, Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr. Niranjana G**, Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, **Dr. Jeya R ,** Associate Professor and Panel Members **Dr. Bhargavi G ,** Assistant Professor and **Dr. Senthil Kumar K**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisors, **Dr. Arthi B (A1 Section) -** Associate Professor , and **Dr. Prakash B (D1 Section) -** Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. Jeya R**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computing Technologies, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

Authors

VIVEK M G
(RA221003010002)

RAGHURAM SRIKANTH
(RA2211003010218)

ABSTRACT

Heart attacks continue to be one of the leading causes of sudden deaths and road accidents worldwide, predominantly because early warning signs are frequently unnoticed or difficult to identify during transit. By incorporating embedded machine learning in electric and fuel-powered vehicles, this project provides a real-time health monitoring system specifically designed for drivers, thereby addressing this life-threatening issue. The system is equipped with biomedical sensors that continuously monitor critical physiological parameters, including heart rate, SpO_2, body temperature, and sweat level to find any risk of heart attack. In contrast to traditional health monitoring devices that heavily depend on cloud computing or external devices, this system conducts on-device analysis using a lightweight and optimized machine learning model directly deployed onto the Arduino Nano 33 BLE Sense Rev2 board. This local processing not only ensures reliability and functionality in remote locations without internet access, but also reduces latency. The system can precisely identify abnormal patterns that may indicate the onset of a heart attack and promptly initiate alerts by analysing sensor data in real time. This alert system provides drivers and adjacent individuals with a critical time frame to take preventive action or seek medical assistance, thereby reducing risks and saving lives. The solution not only monitors, but also transforms conventional vehicles into smart, health-responsive environments, thereby bridging the technological divide between healthcare and transportation. High-risk driver categories, including elderly individuals, long-distance taxi drivers and public transport operators, are particularly affected by this. The initiative is by the broader vision of smart mobility and UN Sustainable Development Goal 3 in addition to improving road safety and driver's well-being.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

SDG	-	Sustainable Development Goal
HR	-	Heart Rate
SpO_2	-	Peripheral Capillary Oxygen Saturation
GSR	-	Galvanic Skin Response
ML	-	Machine Learning
CSV	-	Comma-Separated Values
TFLite	-	TensorFlow Lite
NCD	-	Non-Communicable Disease
IDE	-	Integrated Development Environment
UI	-	User Interface
TC	-	Test Case
BLE	-	Bluetooth Low Energy
AI	-	Artificial Intelligence
IoT	-	Internet of Things
MCU -	-	Microcontroller Unit

CHAPTER 1

INTRODUCTION

1.1. Introduction to Embedded Machine Learning for Early Detection of Heart Attack Symptoms

Heart attacks are a prevalent cause of abrupt fatalities and road accidents, frequently occurring without any prior notification. Many drivers experience early symptoms but fail to identify them in a timely manner, resulting in life-threatening situations. Our initiative employs embedded machine learning in vehicles to predict heart attack symptoms in real-time, thereby resolving this issue. Physiological signals such as heart rate, blood pressure, oxygen levels, and sweating, are continuously monitored by the sensors in this system. This data is locally processed by a lightweight machine learning model that is capable of identifying the potential risk of a heart attack. A driver can take precautionary steps or seek medical assistance by receiving an immediate warning from the system in the event that a cardiac arrest pattern is detected. This initiative enhances driver safety, reduces accidents and enhances modern vehicles by integrating AI-driven health monitoring in both electric and fuel-powered vehicles. This innovation is a significant step towards the development of intelligent, health-conscious transportation systems that prioritize the well-being of humans.

1.2. Motivation

Cardiovascular diseases, particularly heart attacks, are a leading cause of sudden deaths during driving, indicating a serious risk not only to the driver but also to passengers and others on the road. Often, symptoms of an early heart attack go unnoticed or are misdiagnosed, especially while operating a vehicle, the drivers concentration is fully on the road due to which early signs like changes in heart rate, oxygen saturation, or perforated sweating go unnoticed.

This project was inspired by the urgent need for a trusted solution that can monitor a driver's physiological parameters in real-time and provide early alerts to prevent accidents and protect lives. By combining machine learning capabilities into a low power microcontroller, the goal was to create an intelligent system that can detect early symptoms of heart attacks using strong symptoms like Heart Rate, SpO_2 and sweat levels .

The motivation is also derived from the potential societal impact of this system, which includes the enhancement of driver safety, the reduction of accident rates, and the benefit of early medical intervention. Our solution is designed to be adaptable, scalable, and accessible for both electric and fuel-powered vehicles, thereby establishing a foundation for future developments in health-aware smart vehicles.

1.3. Sustainable Development Goal of the Project

This project strongly supports **SDG 3: Good Health and Well-being**, which aims to ensure healthy lives and promote well-being for people of all ages. It majorly contributes to advancing health technologies that enable early detection, prevention, and timely intervention for life-threatening conditions.

Prevention of Life-Threatening Conditions

The embedded system developed in this project monitors critical physiological parameters such as Heart Rate, SpO_2 and sweat levels to detect early signs of a heart attack. By continuously tracking these indicators, the system provides real-time alerts, allowing immediate action and potentially preventing accidents. This precautionary approach enhances road safety and driver health, addressing the growing concern of sudden cardiac events during driving.

Support for SDG Target 3.4

This work directly aligns with **SDG Target 3.4**, which focuses on reducing by one t premature mortality from non-communicable diseases (NCDs) through prevention, treatment, and promotion of mental health and well-being. By utilizing embedded machine learning and affordable sensor technologies, this project contributes to accessible healthcare innovation aimed at reducing deaths caused by non-communicable cardiovascular conditions.

1.4. Product Vision Statement

To develop an intelligent, real-time health monitoring system embedded within vehicles that leverages machine learning to detect early symptoms of heart attacks in drivers—enhancing road safety, enabling timely medical intervention, and ultimately contributing to the reduction of sudden cardiac-related accidents, in alignment with global health and safety goals.

1.5. Product Goal

The primary goal of this project is to design and develop a reliable, real-time, and cost-effective embedded system that can accurately detect early symptoms of heart attacks in drivers using physiological sensor data and machine learning algorithms. The system aims to monitor critical health parameters—such as Heart Rate, SpO_2 and sweat levels (via GSR sensor)—in a non-intrusive manner, process this data locally on the microcontroller, and provide immediate alerts when abnormal patterns indicative of a possible heart attack are detected.

This goal encompasses the following key objectives:

1. **Sensor Integration and Data Acquisition:** Select and integrate appropriate biosensors capable of capturing accurate physiological data from the driver in real-time without disrupting the driving experience.

2. **Data Processing and Embedded Machine Learning:** Preprocess the acquired data and implement lightweight machine learning models using TensorFlow Lite that can run efficiently on the Arduino Nano 33 BLE Sense Rev2 microcontroller for real-time inference.

3. **System Reliability and Accuracy:** Ensure high accuracy in the detection of abnormal health patterns by training the model on diverse datasets and optimizing its performance for embedded deployment.

4. **Alert and Notification Mechanism:** Develop a timely and effective alert system (such as buzzer, visual indicator, or mobile notification) to notify the driver and nearby individuals in the event of critical health anomalies.

5. **User Safety and Practical Application:** Design the solution to be easily deployable in both electric and fuel-powered vehicles with minimal technical knowledge required for installation or use, thereby maximizing its reach and real-world impact.

6. **Support for Preventive Healthcare:** Contribute to the broader healthcare ecosystem by offering a proactive tool that supports early diagnosis and prevention of cardiovascular incidents, thus aligning with SDG 3 – Good Health and Well-being.

By achieving these objectives, the system will serve as a vital step toward smarter, health-aware mobility solutions and the broader integration of AI in embedded health monitoring.

1.6. Product Backlog (Key User Stories with Desired Outcomes)

Table 1.1 depicts the Product Backlog of the project.

Table 1.1 - Product Backlog of Embedded Machine Learning for Early Detection of Heart Attack Symptoms

Product Backlog sample for Embedded Machine Learning for Early Prediction of Heart Attack Symptoms										
ID	Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	Heart Rate Monitoring System Integration	**Health Monitoring**	As a driver, I want the system to continuously monitor my heart rate so that any anomalies can be detected early.	Must Have	To Do	System accurately monitors heart rate within ±2 bpm accuracy.	Integrate heart rate sensors into the vehicle seating system.	Real-time data processing with less than 1-second latency.	5 days	
2	Data Processing Unit for ML Model	**Health Monitoring**	As a developer, I want the system to process health data using embedded ML models so that predictions are accurate and fast.	Must Have	To Do	ML model processes data with at least 95% accuracy.	Implement edge computing unit capable of running ML algorithms.	Ensure model inference time is under 100ms.	7 days	
3	Alert System Development	**Health Monitoring**	As a driver, I want to receive alerts if the system detects potential heart attack symptoms so that I can take immediate action.	Must Have	To Do	Alerts are triggered within 5 seconds of detection with clear instructions.	Develop alert mechanism including visual and auditory signals.	Alerts must be noticeable even in noisy environments.	8 days	
4	User Interface for Health Data	**Health Monitoring**	As a user, I want to view my health data on the vehicle's infotainment system so that I am aware of my current health status.	Must Have	To Do	Display real-time heart rate and other vital signs on the dashboard.	Design UI components for displaying health metrics.	Interface should update every second without lag.	6 days	
5	Emergency Contact Notification	**Health Monitoring**	As a driver, I want the system to notify emergency contacts if a critical condition is detected so that help arrives promptly.	Must Have	To Do	Emergency contacts are notified within 30 seconds of critical detection.	Implement automatic notification feature via SMS or call.	Ensure reliable communication even in low network areas.	5 days	
6	Data Privacy Compliance	**Health Monitoring**	As a user, I want to use the app without an internet connection so that I can access its features and information even when I am offline	Must Have	To Do	All health data is encrypted and complies with GDPR standards.	Implement encryption protocols for data storage and transmission.	System must pass security audits and penetration tests.	7 days	
7	Integration with EV Systems	**Health Monitoring**	As a user, I want my health data to be securely stored and processed so that my privacy is maintained.	Must Have	To Do	All health data is encrypted and complies with GDPR standards.	Ensure compatibility with existing vehicle ECUs and software.	System integration verified through comprehensive testing.	5 days	
8	Continuous Model Improvement	**Health Monitoring**	As an engineer, I want the health monitoring system to seamlessly integrate with existing EV systems so that it does not interfere with vehicle performance.	Could Have	To Do	Health monitoring system operates independently without affecting vehicle performance.	Set up pipeline for continuous model training and deployment.	Updates should occur with minimal downtime (<1 minute)	5 days	
9	Driver Feedback Collection	**Health Monitoring**	As a researcher, I want to collect feedback from drivers about the system's performance so that future improvements can be made.	Should Have	To Do	Feedback collected includes both qualitative and quantitative data.	Ensure feedback is easy to provide and understand.	Ensure feedback is easy to provide and understand.	4 days	
10	Power Consumption Optimization	**Health Monitoring**	As an engineer, I want the health monitoring system to consume minimal power so that it does not significantly affect the EV's battery life.	Should Have	To Do	Feedback collected includes both qualitative and quantitative data.	Optimize algorithms and hardware for low power consumption.	Regularly test power usage under different conditions.	5 days	

1.7. Product Release Plan

The following Figure 1.1 depicts the release plan of the project.

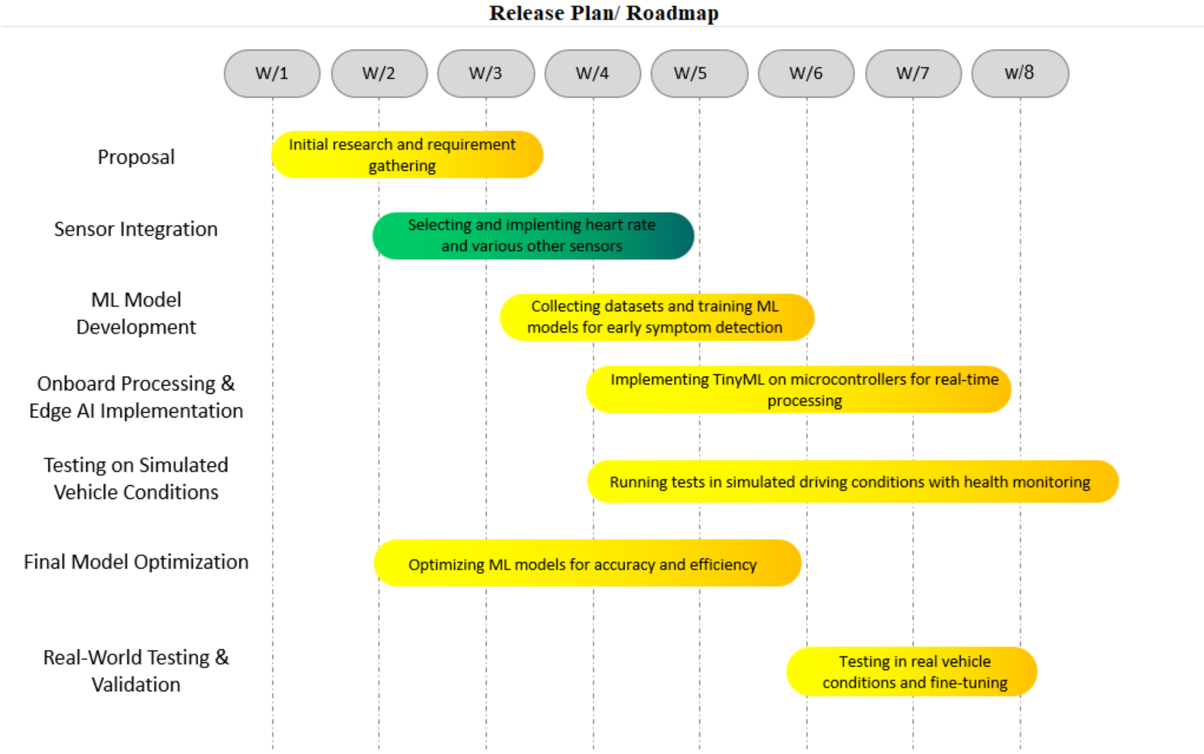


Figure 1.1 - Release Plan of Embedded Machine Learning for Early Detection of Heart Attack Symptoms

CHAPTER 2

SPRINT PLANNING AND EXECUTION

2.1. Sprint 1

2.1.1. Sprint Goal with User Stories of Sprint 1

Focus: Understanding the problem, identifying goals, and preparing the foundational components.

Table 2.1 - User Stories of Sprint - 1

#US NO	USER STORIES
1.	Study existing embedded ML solutions
2.	Analyze appropriate medical sensors
3.	Understand SDG alignment

Planner Board representation of user stories are mentioned below figures 2.1, 2.2 and 2.3.

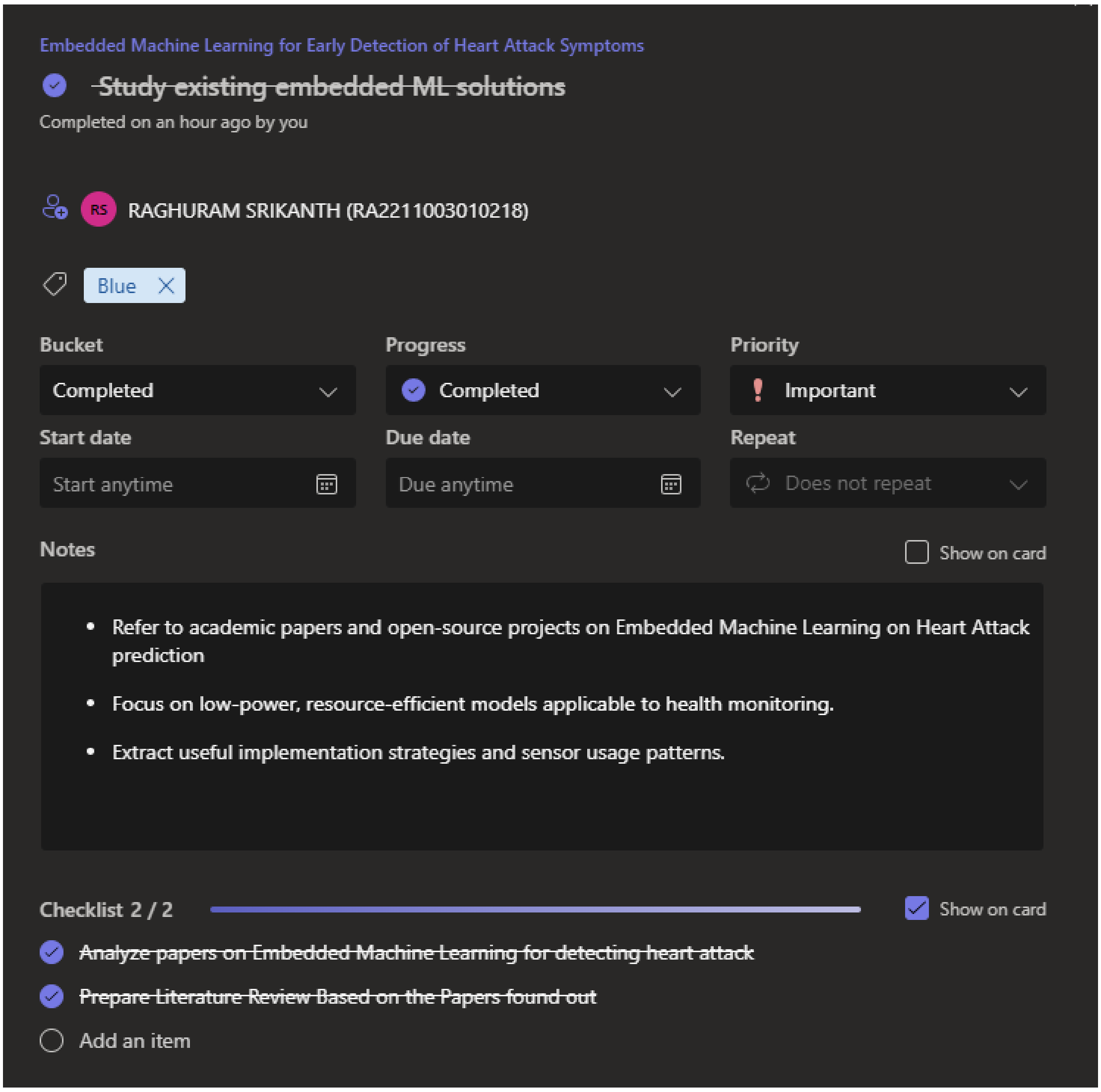


Figure 2.1 User Story for Study existing embedded ML solutions

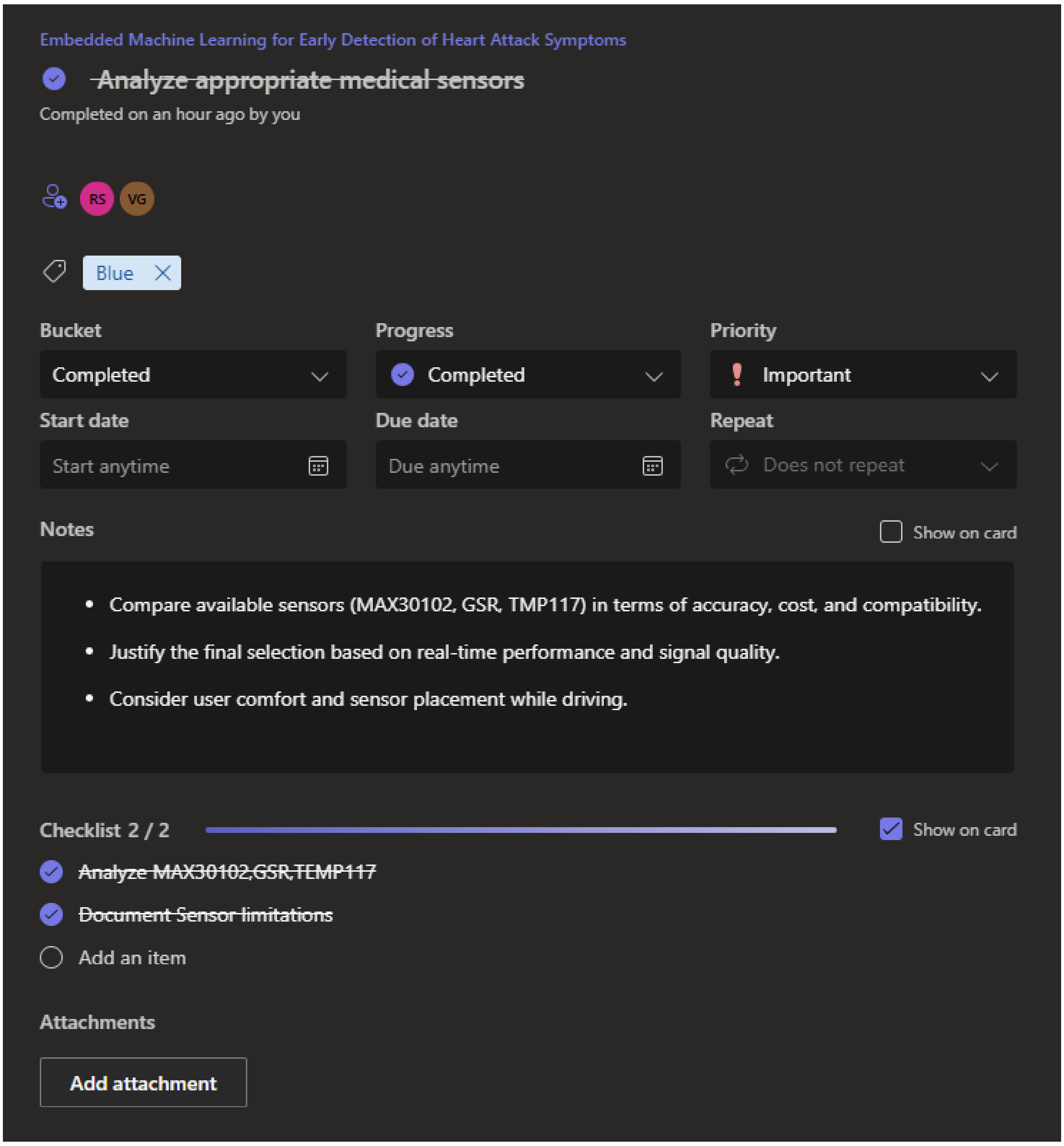


Figure 2.2 User Story for Analyzing appropriate medical sensors

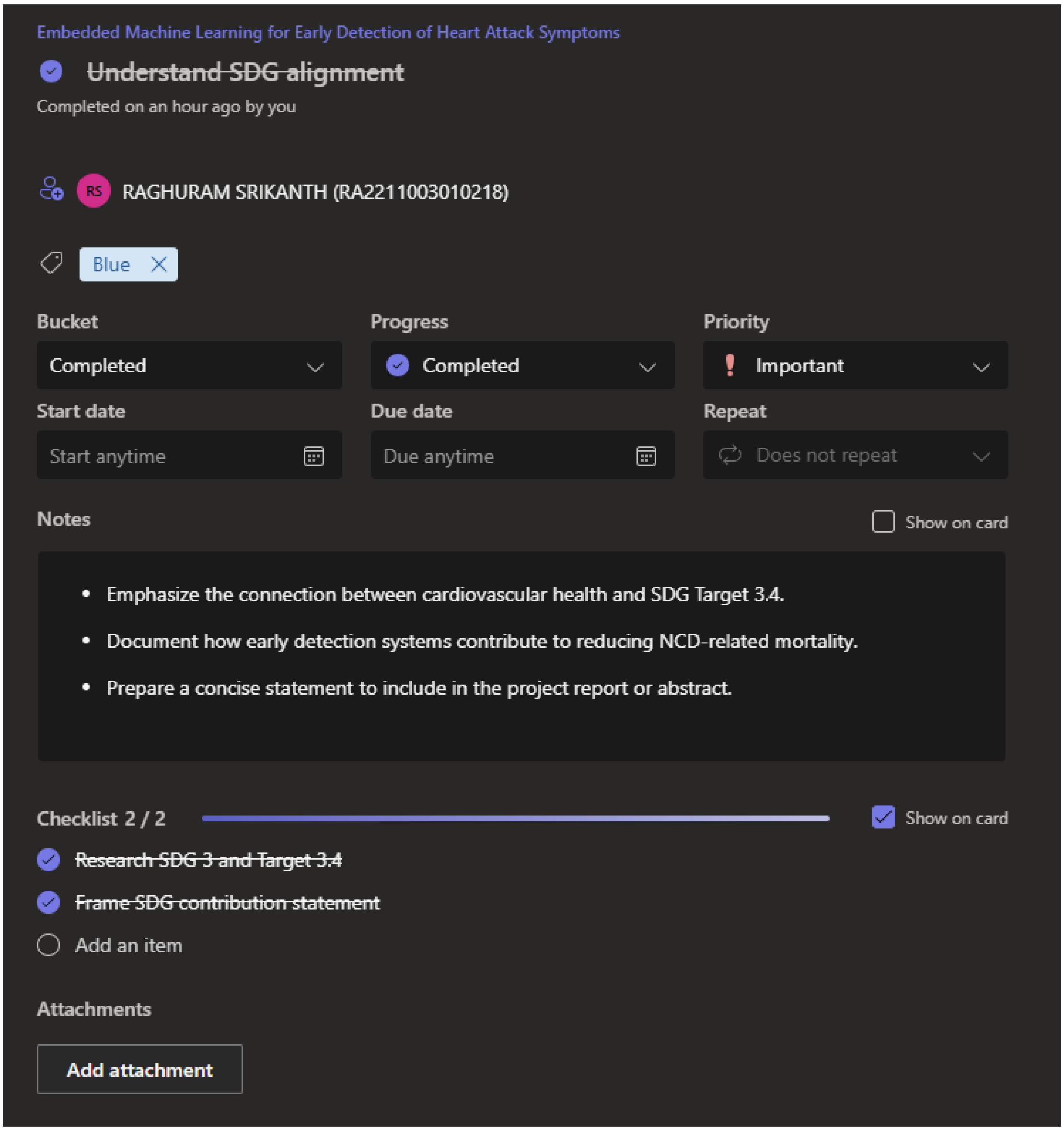


Figure 2.3 User Story for understanding SDG alignment

2.1.2. Functional Document

2.1.2.1. Introduction

This sprint lays the basis for the development of a real-time embedded system designed to detect early signs of heart attacks in EV and fuel-powered vehicle drivers. The system aims to prevent severe road accidents by collecting physiological data and using on-device machine learning for early risk prediction. Sprint 1 focuses on understanding the project's relevance to global health standards, researching embedded ML solutions, and selecting suitable biomedical sensors.

2.1.2.2. Product Goal

The ultimate goal of the product is to create a wearable or dashboard-integrated device that monitors key health parameters such as Heart Rate, SpO_2 and sweat level using appropriate sensors. The collected data will be processed using a lightweight machine learning model deployed on an embedded platform (Arduino Nano 33 BLE Sense Rev2). The main purpose is to predict early detection of heart attack symptoms in real time and alert the driver, potentially saving lives and enhancing safety.

2.1.2.3. Demography

Users:

- Vehicle drivers, especially those in high-risk categories such as long-distance cab drivers, public transport operators and elderly people. Additionally, automobile manufacturers, automobile safety system manufacturers, and healthcare professionals interested in remote health monitoring can benefit from the system.

Location:

- Applicable in both urban and rural areas where the risk of undetected heart-related accidents while driving is high. Especially useful in regions with long highway stretches, limited access to immediate medical care.

2.1.2.4. Features

Table 2.2 depicts features of sprint 1

Table 2.2 - Sprint 1 Features

Feature	Description	Benefit
SDG Integration	Integration of project objectives with global health development goals.	Adds societal and academic relevance, especially for research papers and funding.
Benchmarking Embedded ML Models	Review of lightweight models suitable for deployment on constrained hardware.	Guides model selection with efficiency and performance in mind.
Sensor Evaluation	Technical and usability evaluation of biomedical sensors.	Ensures accuracy, real-time performance, and driver safety during integration.

2.1.3. Architecture Document

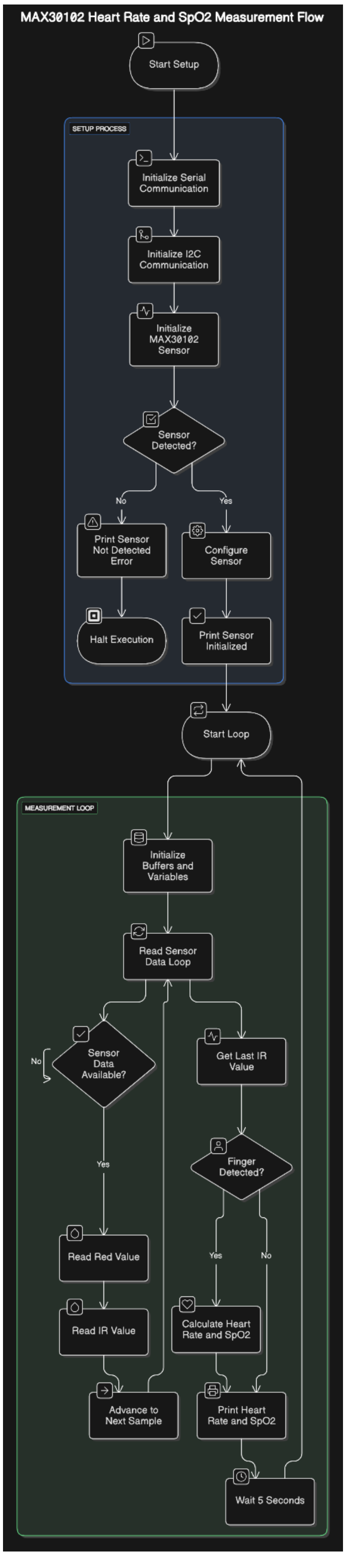


Figure 2.4 – Architecture Diagram for MAX30102 Connection

This flowchart illustrates the process for measuring heart rate and SpO2 levels using the MAX30102 sensor. It begins with a setup phase where serial communication and I2C communication are initialized, followed by initializing the MAX30102 sensor. If the sensor is detected, it is configured; otherwise, an error is printed and execution halts. After successful initialization, the system enters a measurement loop where buffers and variables are initialized, and sensor data is continuously read. If data is available, red and infrared (IR) values are read. If a finger is detected based on the IR value, the heart rate and SpO2 are calculated and printed, followed by a 5-second wait before the next measurement cycle.

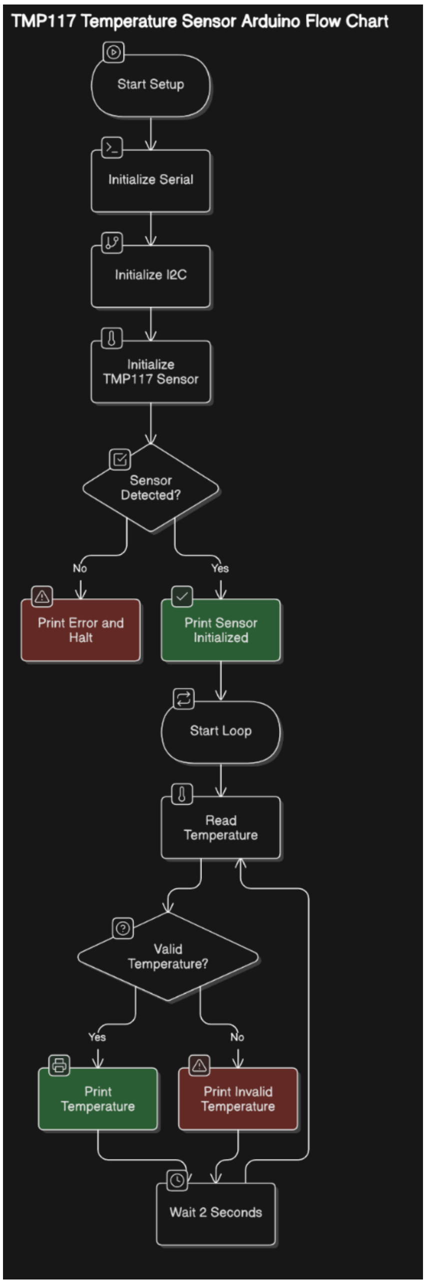


Figure 2.5 – Architecture Diagram for TMP 177 Connection

This flowchart describes the flow for reading temperature data from the TMP117 sensor using Arduino. The process starts with setting up serial and I2C communication, followed by initializing the TMP117 sensor. A check is performed to detect the sensor; if the sensor is not found, an error is printed and execution halts. If the sensor is detected, it proceeds to a loop where the temperature is read repeatedly. After reading, the temperature's validity is checked. If valid, it prints the temperature; otherwise, it prints an invalid temperature message. The system waits for 2 seconds before the next reading cycle, ensuring continuous monitoring.

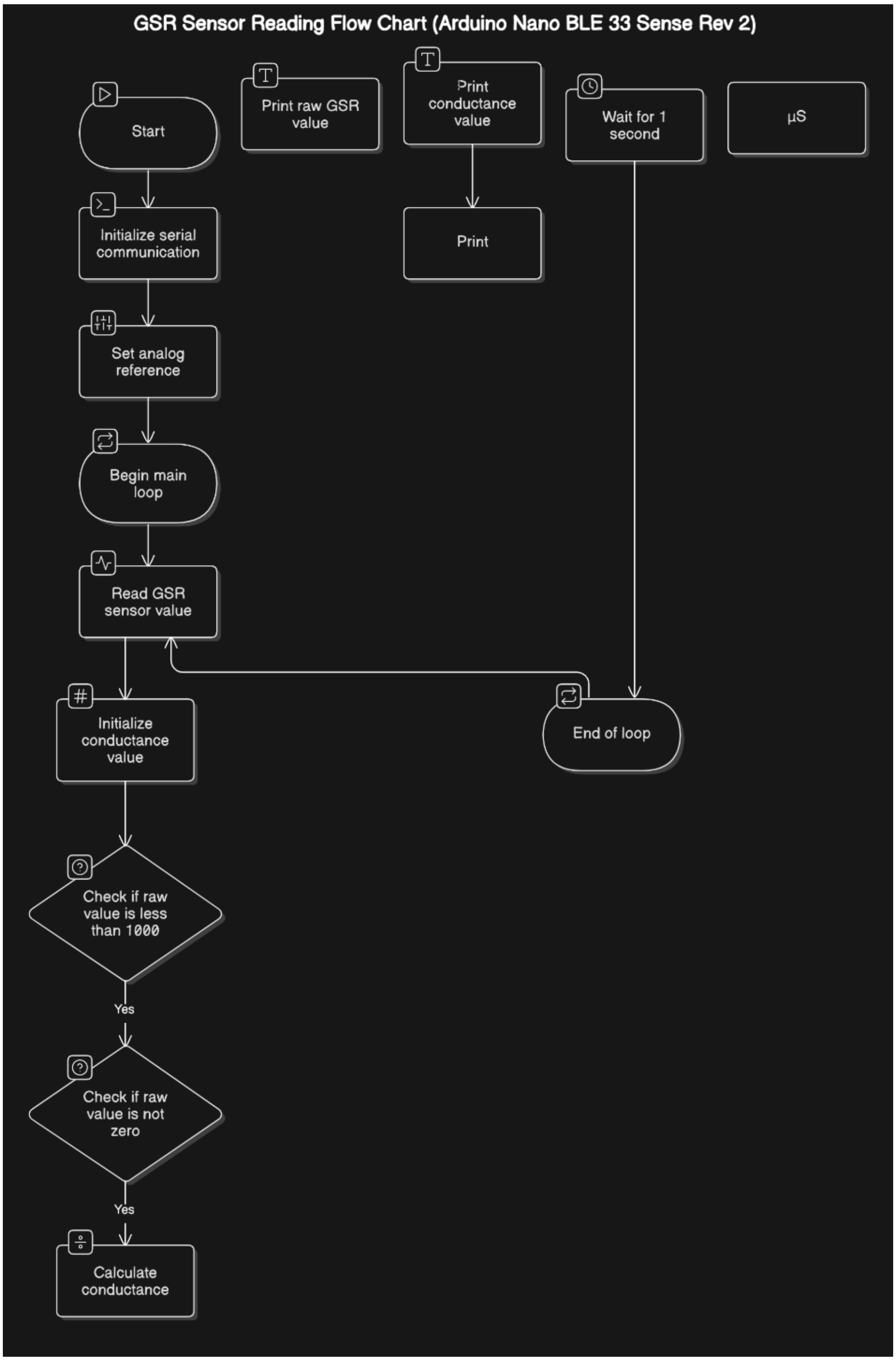


Figure 2.6 – Architecture Diagram for GSR Connection

This flowchart outlines the procedure for reading data from a GSR (Galvanic Skin Response) sensor. It begins with initializing serial communication and setting the analog reference, then enters the main loop. The GSR sensor value is read, and the conductance value is initialized. It checks if the raw GSR value is less than 1000 and non-zero; if both conditions are satisfied, conductance is calculated. Throughout the process, raw GSR values and calculated conductance values are printed, with a 1-second delay between readings. This continuous loop enables monitoring of skin conductance, which can indicate emotional or physiological states.

2.1.4. Functional Test Cases

Table 2.3 depicts the Functional Test Cases for sprint 1

Table 2.3 -Functional test cases in Sprint 1

Test Case ID	Test Objective	Input	Expected Output	Actual Result	Status
TC_01	Validate SDG alignment	Project documentation on SDG goals	Clear mapping of SDG goal 3 and target 3.4 and justification of social impact	SDG link documented and aligned correctly	Pass
TC_02	Verify literature review relevance	List of reviewed research papers	Papers must be recent, relevant, and focused on embedded ML for health	All papers were applicable and cited	Pass
TC_03	Test sensor feasibility analysis	Technical datasheets for MAX30102, TMP117, GSR	Comparative table with power, range, accuracy, and compatibility	Analysis done and sensors selected	Pass
TC_04	Validate sensor selection reasoning	Final sensor list with justifications	Justified based on use-case, power efficiency, and Arduino compatibility	Final sensors selected: MAX30102, GSR	Pass
TC_05	Check model	List of	Criteria defined: accuracy,	Benchmark	Pass

	benchmarking criteria	shortlisted ML models	latency, memory usage on Arduino Nano	table created and justified	
TC_06	Review completeness of Sprint-1 documentation	Sprint 1 documentation	Should include Intro, Product Goal, Demography, Business Process, and Features	All sections completed	Pass
TC_07	Validate planning tools	Microsoft Planner task board	All tasks properly categorized into Sprint 1	Tasks grouped correctly in planner	Pass

2.1.5. Daily Call Progress

Figure 2.7 depicts the record of standup meetings conducted for sprint 1.

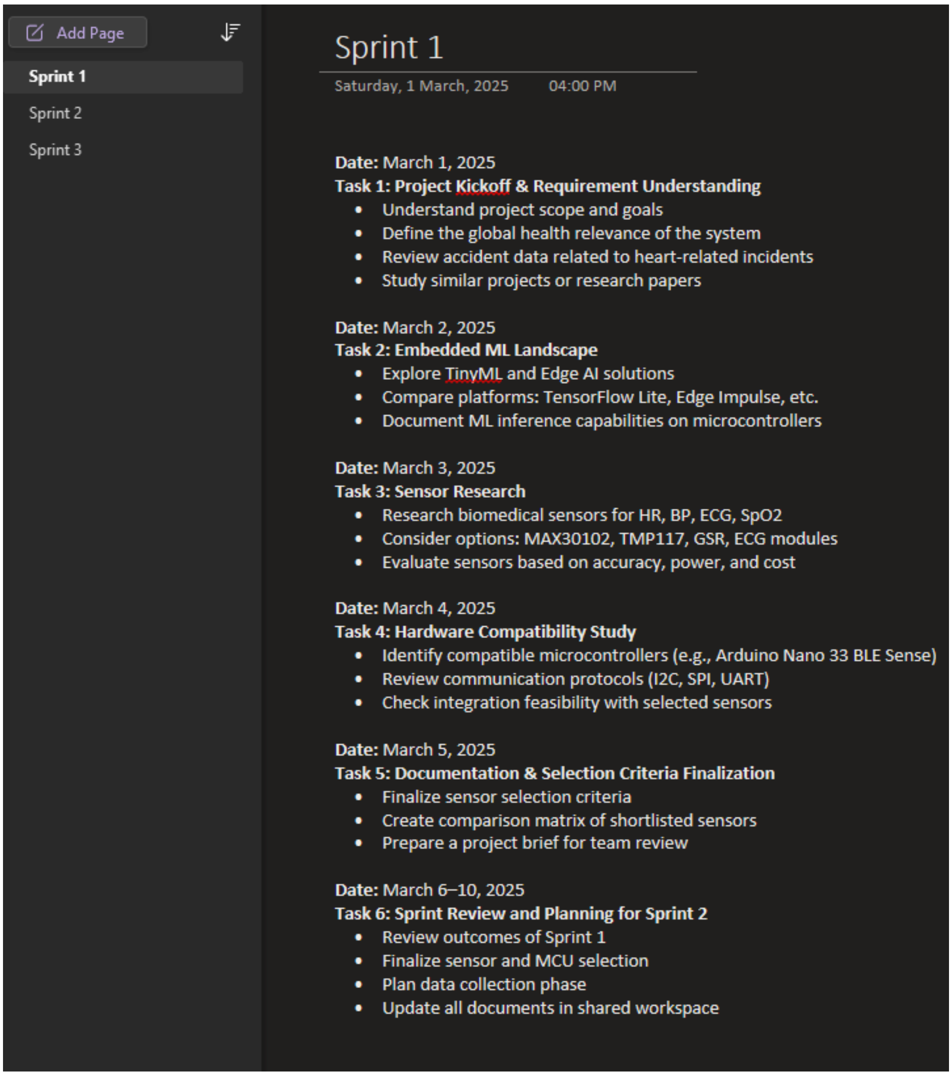


Figure 2.7 - Standup Meetings for Sprint-1

2.1.6. Committed vs Completed User Stories

The figure 2.8 represents the committed vs completed user stories for sprint 1 in a bar graph.

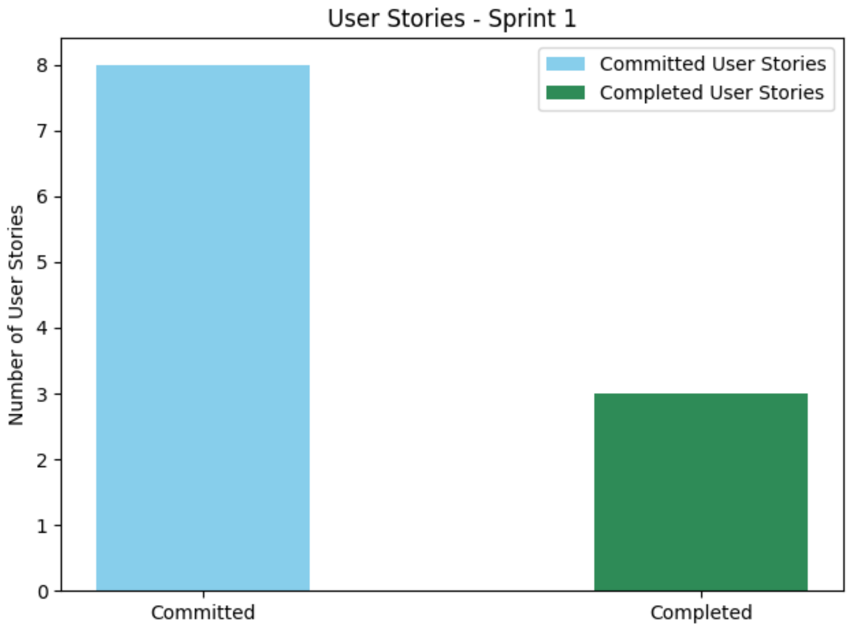


Figure 2.8 - Committed Vs Completed User Stories for Sprint-1

2.1.7. Sprint Retrospective

Table 2.4 depicts the sprint retrospective for sprint 1.

Table 2.4 -Sprint 1 Retrospective

Sprint Retrospective-1				
What went well	**What went poorly**	**What ideas do you have**	**How should we take action**	***Guidelines***
Finalized a meaningful and SDG-aligned project idea.	Delay in finalizing sensors due to scattered research.	Start with a requirement matrix for sensor selection.	Use a structured comparison table for sensor evaluation.	Prioritize hardware research before finalizing tech stack.
Smooth collaboration during idea finalization.	Documentation was initially vague in technical	Begin writing documents parallel to the ideation phase.	Assign a documentation lead per sprint.	Maintain version-controlled documentation (Google Docs or Git).

	sections.			
Identified real-time application and demography relevance.	Unfamiliarity with sensor capabilities caused confusion.	Schedule internal sessions to study sensor datasheets.	Set aside a day for technical deep-dives each sprint.	Encourage everyone to go through sensor datasheets individually.
SDG alignment improved clarity and purpose of the project.	Some members struggled with time management	Implement a sprint board to visualize task assignments.	Use MS Planner or Trello to assign and track deliverables.	Daily/alternate-day sync-ups to ensure alignment.

2.2. Sprint 2

2.2.1. Sprint Goal with User Stories of Sprint 2

Focus: Feature selection, data preparation, and model design.

Table 2.5 - User Stories of Sprint - 2

#US NO	USER STORIES
1.	Prepare dataset and identify features
2.	Implement ML model

Planner Board representation of user stories are mentioned below figures 2.9 and 2.10.

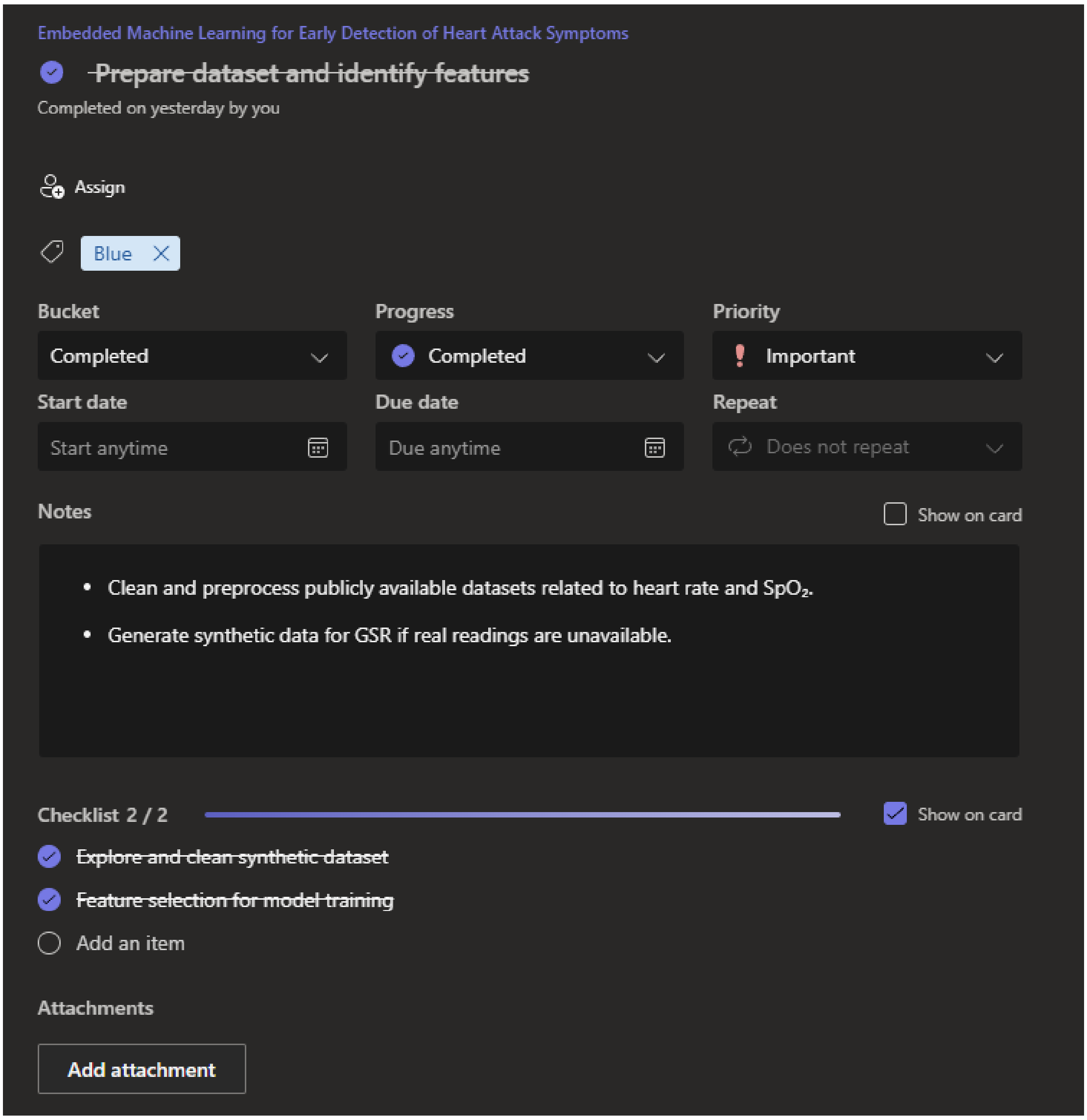


Figure 2.9 - User Story for preparing the dataset and identifying its features

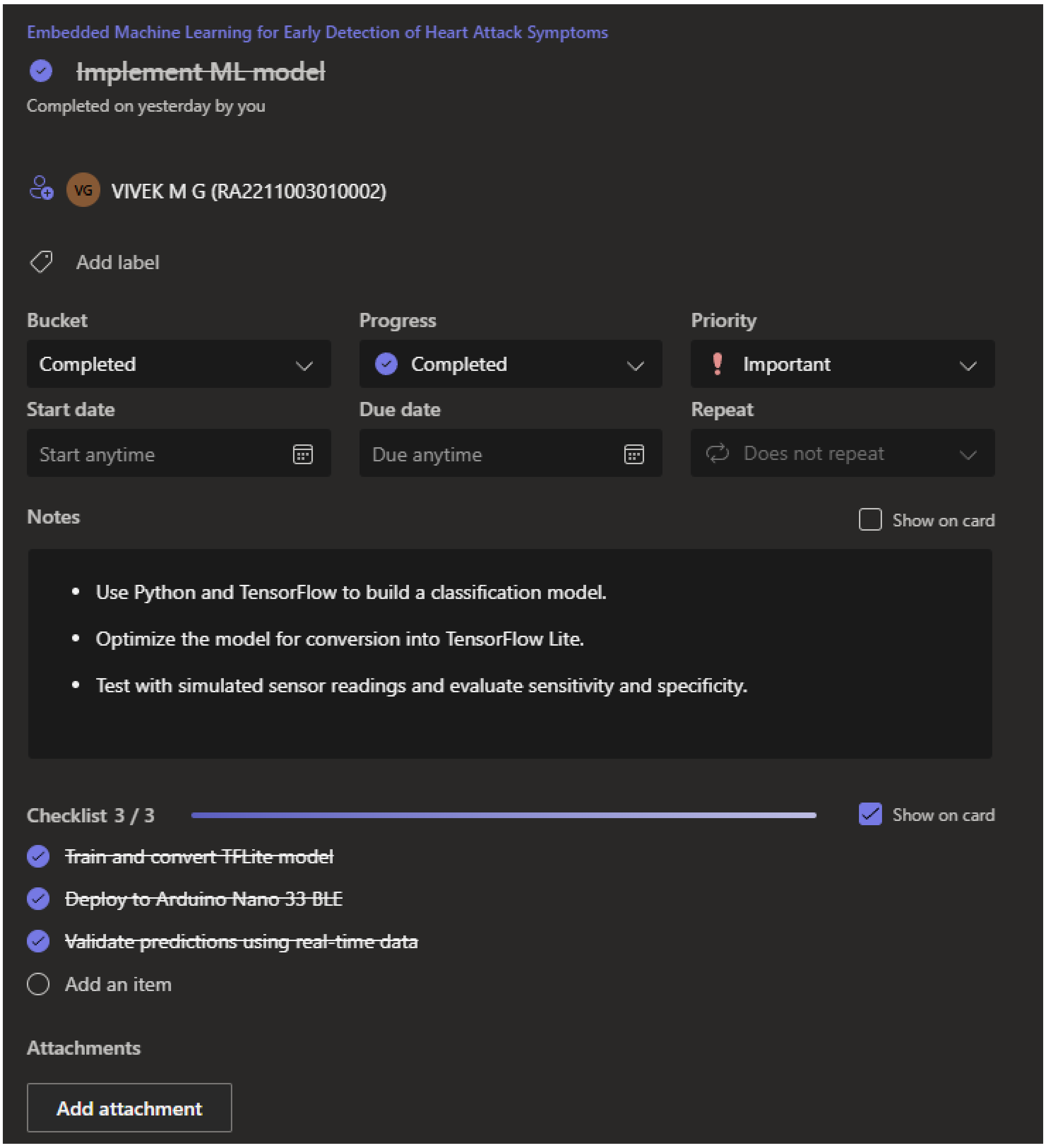


Figure 2.10 - User Story for implementing ML mode

2.2.2. Functional Document

2.2.2.1. Introduction

This sprint focuses on the collection, preprocessing, and modeling of physiological data required to detect early symptoms of heart attacks in drivers. It involves generating a reliable dataset from multiple sensors and using it to train a lightweight ML model that can be deployed on embedded systems.

2.2.2.2. Product Goal

The primary goal of Sprint 2 is to develop and validate a lightweight, efficient, and accurate machine learning model capable of predicting early symptoms of heart attacks using the physiological data collected from biomedical sensors. This model will be tailored for real-time inference on edge devices like the Arduino Nano 33 BLE Sense Rev2, with a focus on optimizing both performance and power consumption.

This sprint ensures that the system can:

• Identify, preprocess, and label meaningful features (Heart Rate, SpO2, Body Temperature, and Sweat Level) from the structured dataset generated synthetically.

• Select and implement an appropriate ML algorithm (Neural Networks) suitable for small-scale edge inference.

• Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score to ensure it meets medical-grade responsiveness.

• Export the model using TensorFlow Lite to facilitate seamless integration into the embedded system for real-time predictions.

• Validate the model's robustness against noisy or borderline sensor data to avoid false positives.

• Ensure modularity in model design so that it can be retrained and upgraded as more data becomes available in future releases.

By fulfilling these objectives, Sprint 2 will produce a deployable and interpretable machine learning model optimized for embedded environments, setting the stage for its real-time deployment in the final product.

2.2.2.3. Demography

Users:

- Vehicle drivers, especially those in high-risk categories such as long-distance cab drivers, public transport operators and elderly people. Additionally, automobile manufacturers, automobile safety system manufacturers, and healthcare professionals interested in remote health monitoring can benefit from the system.

Location:

- Applicable in both urban and rural areas where the risk of undetected heart-related accidents while driving is high. Especially useful in regions with long highway stretches, limited access to immediate medical care.

2.2.2.4. Features

Table 2.4 depicts the features of sprint 2

Table 2.6 – Sprint 2 Features

Feature	Description	Benefit
Physiological Dataset Creation	Collected real-time data from sensors (Heart Rate, SpO2, Temperature, GSR) under various physical and emotional states.	Builds a diverse and representative dataset essential for training the ML model.
Data Labeling and Annotation	Tagged data samples with corresponding health conditions (normal/stress/abnormal) for supervised learning.	Provides ground truth for accurate model training and evaluation.
Data Preprocessing	Cleaned, normalized, and transformed raw sensor data for model input	Ensures data quality and consistency, improving model performance.
ML Model Development	Trained a lightweight ML model using TensorFlow Lite to classify early heart attack symptoms based on sensor input.	Enables onboard predictive analysis for timely health alerts.

2.2.3. Architecture Document

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                80

 dropout (Dropout)           (None, 16)                0

 dense_1 (Dense)             (None, 8)                 136

 dropout_1 (Dropout)         (None, 8)                 0

 dense_2 (Dense)             (None, 4)                 36

 dense_3 (Dense)             (None, 1)                 5

=================================================================
Total params: 257 (1.00 KB)
Trainable params: 257 (1.00 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 2.11 – Architecture Diagram for Neural Network

- Four dense layers plus two dropout layers for regularizing this neural network model, Sequential, are represented.
- With 80 parameters, Dense Layer 1: 16 neurons record initial characteristics from input data.
- Dimensionality reduces to 8 neurons with 136 parameters, improving feature extraction in Dense Layer 2.
- First Dropout Layer: A regularization layer without parameters that randomly deactivates neurons to prevent overfitting.
- Further distilled feature representation is achieved with Dense Layer 3, reducing to 4 neurons with 36 parameters.
- Second Dropout Layer: Another regularization step to strengthen generalization by dropping neurons during training.

- Final Dense Layer: Consisting of 1 neuron and 5 parameters, likely employed for regression or binary classification tasks.
- With 257 trainable parameters overall, the model remains lightweight and efficient for tasks such as health monitoring or simple binary classifications.

2.2.4. Functional Test Cases

Table 2.7 depicts the functional test cases for sprint 2

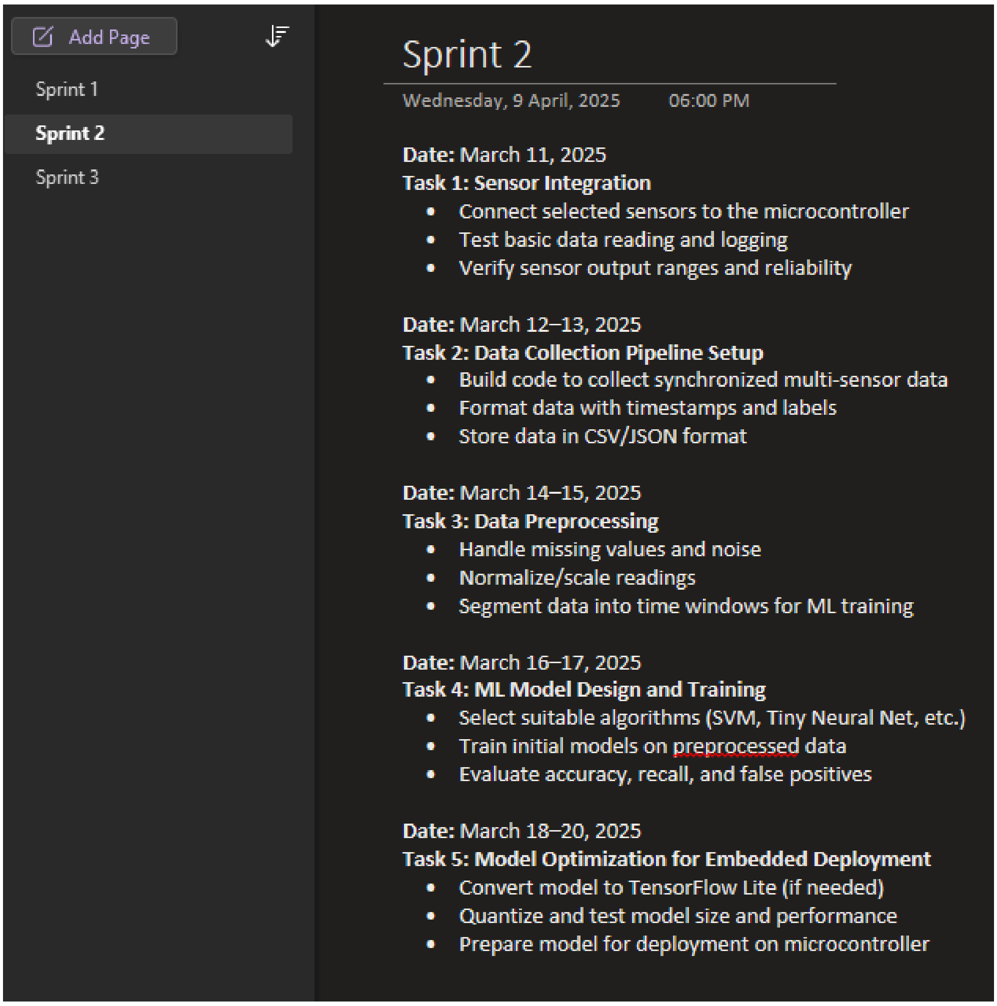
Table 2.7 - Functional test cases in Sprint 2

Test Case ID	Test Objective	Input	Expected Output	Actual Result	**Status**
TC_01	Sensor data acquisition and recording	Live Sensor Readings	Data stored in CSV format with timestamp and labels	Data saved with accurate time stamps and labels	Pass
TC_02	Data preprocessing pipeline	Raw sensor data file	Cleaned and normalised dataset	Missing values are handled, data is normalized	Pass
TC_03	Label consistency check	Annotated dataset	All samples correctly mapped to labels	Labels were verified consistently across dataset	Pass
TC_04	Model Training	Pre processed data file	Model trains with acceptable accuracy	Accuracy: 96.64% after tuning and training	Pass
TC_05	Model Validation	Test data	Model predicts classes correctly with minimal error	High precision and recall, minimal false positives	Pass
TC_06	TensorFlow Lite Conversion	Trained ML model	Model converted successfully and size optimized	tflite model generated (1.3MB), runs	Pass

				without issues	
TC_07	Model runs on embedded board	Embedded Input data Stream	Model infers class in real time without crash	Inference successful; no crashes or memory issues	Pass

2.2.5. Daily Call Progress

Figure 2.12 depicts the record of standup meetings conducted for sprint 2.



Add Page

Sprint 1

Sprint 2

Sprint 3

Sprint 2

Wednesday, 9 April, 2025 06:00 PM

Date: March 11, 2025
Task 1: Sensor Integration

- Connect selected sensors to the microcontroller
- Test basic data reading and logging
- Verify sensor output ranges and reliability

Date: March 12–13, 2025
Task 2: Data Collection Pipeline Setup

- Build code to collect synchronized multi-sensor data
- Format data with timestamps and labels
- Store data in CSV/JSON format

Date: March 14–15, 2025
Task 3: Data Preprocessing

- Handle missing values and noise
- Normalize/scale readings
- Segment data into time windows for ML training

Date: March 16–17, 2025
Task 4: ML Model Design and Training

- Select suitable algorithms (SVM, Tiny Neural Net, etc.)
- Train initial models on preprocessed data
- Evaluate accuracy, recall, and false positives

Date: March 18–20, 2025
Task 5: Model Optimization for Embedded Deployment

- Convert model to TensorFlow Lite (if needed)
- Quantize and test model size and performance
- Prepare model for deployment on microcontroller

Figure 2.12 - Standup Meetings for Sprint-2

2.2.7. Committed vs Completed User Stories

The figure 2.13 represents the committed vs completed user stories for sprint 2 in a bar graph.

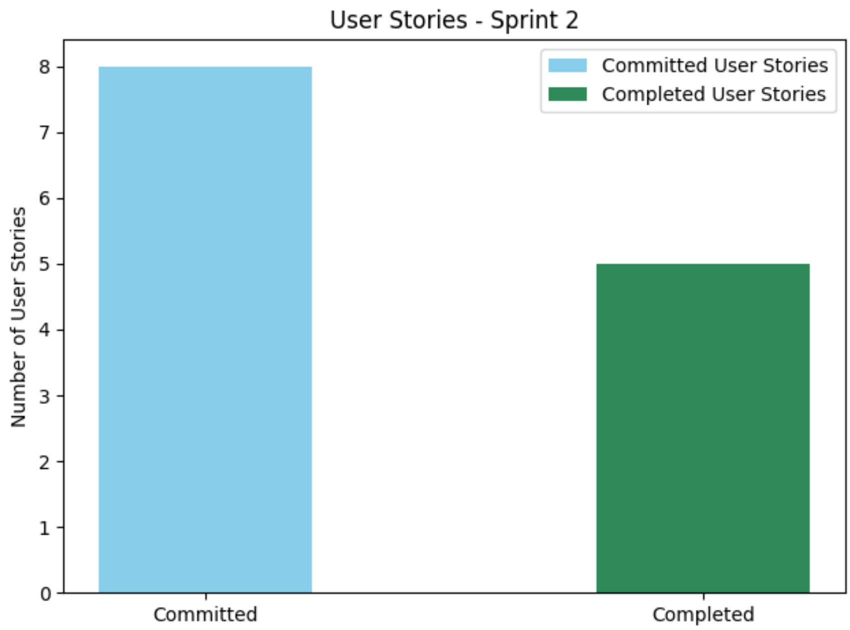


Figure 2.13 - Committed Vs Completed User Stories for Sprint-2

2.2.8. Sprint Retrospective

Table 2.8 depicts the sprint retrospective for sprint 2

Table 2.8 - Sprint 2 Retrospective

Sprint Retrospective-2				
What went well	**What went poorly**	**What ideas do you have**	**How should we take action**	***Guidelines***
All four sensors were successfully integrated.	GSR readings were noisy and unstable in some conditions.	Calibrate GSR using dry and wet skin tests.	Include environmental testing sessions.	Test sensors under different temperatures and lighting.
Real-time data was successfully streamed to the serial monitor.	Temperature sensor readings varied slightly.	Add sensor warm-up time before capturing values.	Wait 10 seconds before initial temperature capture	Log all sensor data for trend analysis.

Heart Rate and SpO2 values matched expected clinical ranges.	Some issues with power draw when all sensors ran together.	Use a stable USB hub or external power source.	Measure board voltage output under load.	Don’t overload digital/analog pins.
Basic filtering helped improve output readability.	Serial monitor lag when all sensors outputted simultaneously.	Optimize the update frequency and serial baud rate.	Reduce unnecessary print statements.	Modularize sensor code for easy tuning.

2.3. Sprint 3

2.3.1. Sprint Goal with User Stories of Sprint 3

Table 2.9 - User Stories of Sprint - 3

#US NO	USER STORIES
1.	Build hardware integration
2.	Real-time prediction system
3.	Final product development

Planner Board representation of user stories are mentioned below figures 2.14, 2.15 and 2.16.

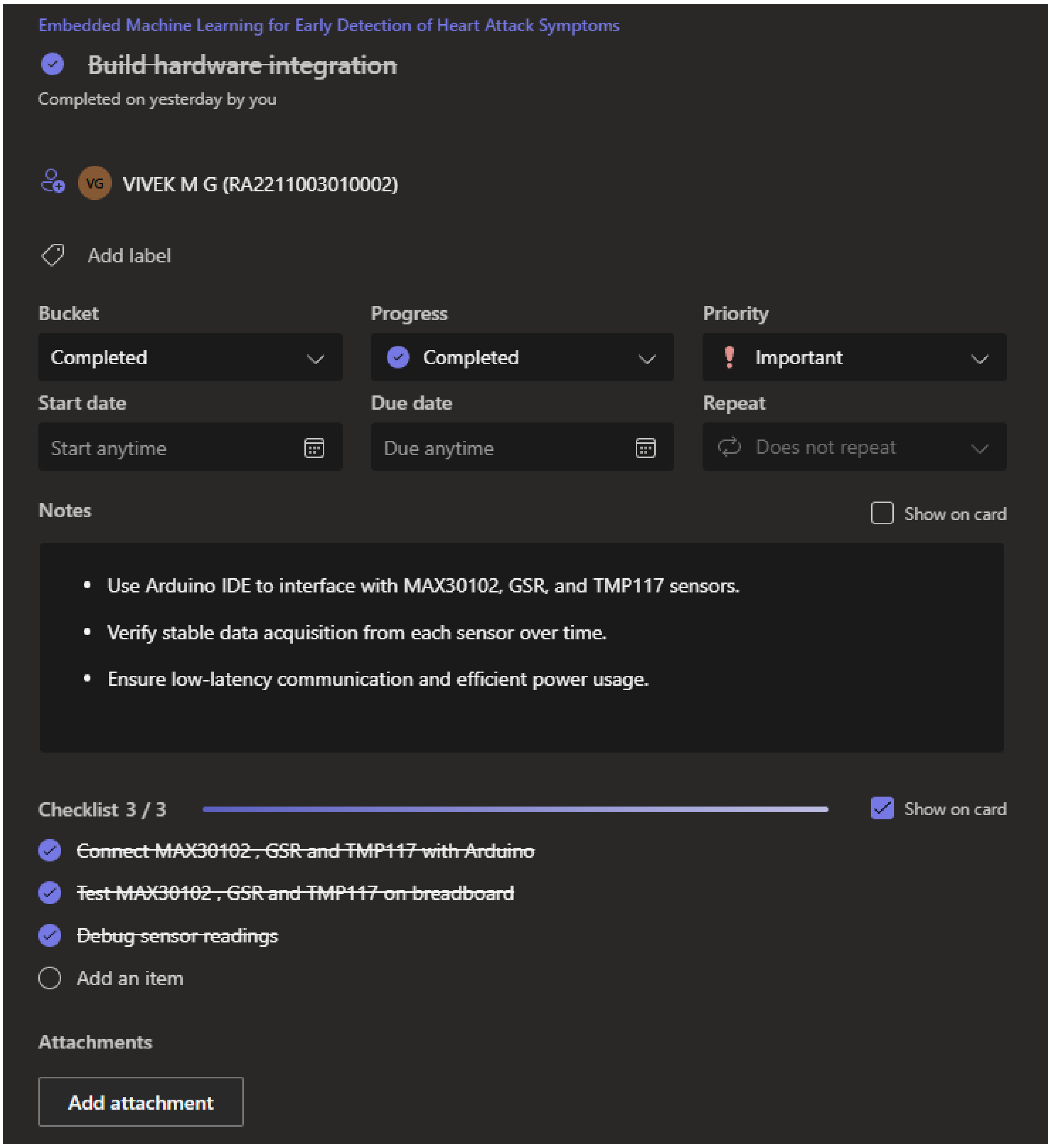


Figure 2.14 - User Story for hardware integration

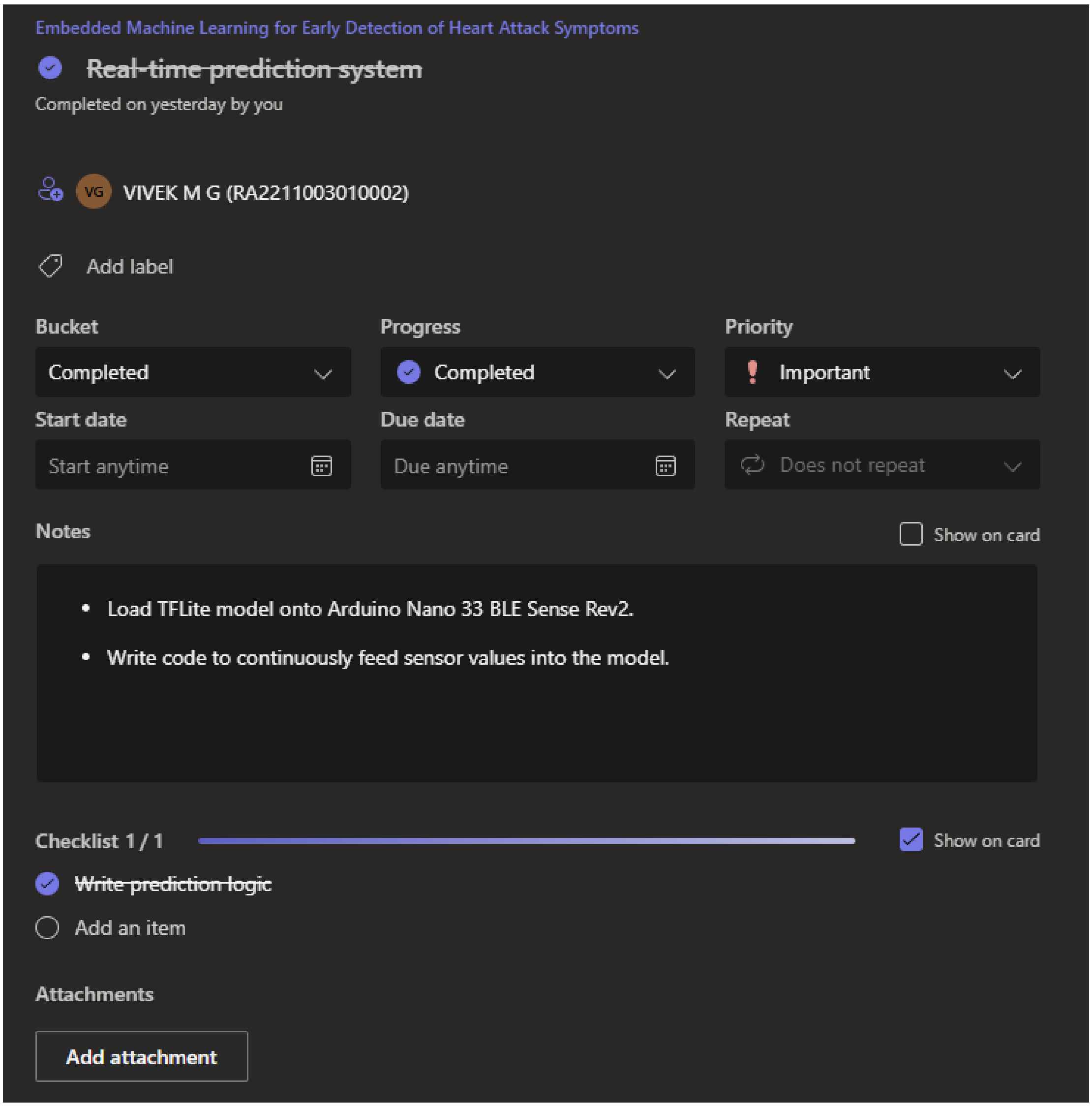


Figure 2.15 - User Story for real time prediction

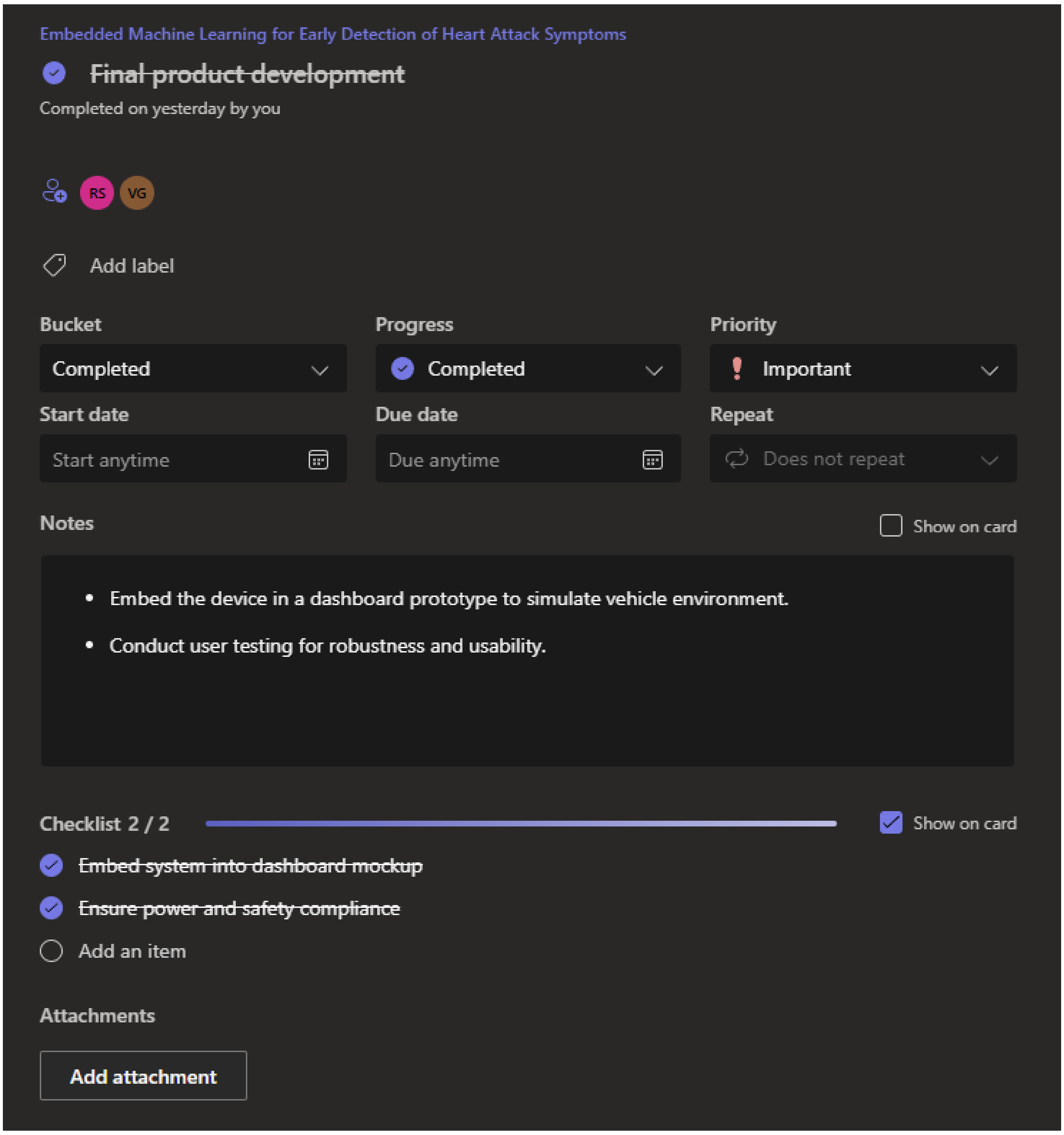


Figure 2.16 - User Story for final product deployment

2.3.2. Functional Document

2.3.2.1. Introduction

Sprint 3 focuses on integrating the trained ML model with the embedded hardware system to enable real-time health monitoring and prediction. It also includes the development and testing of the final product with all sensors, microcontroller, model, and alert system.

2.3.2.2. Product Goal

The primary goal of Sprint 3 is to achieve a fully functional, real-time embedded health monitoring system by integrating the trained machine learning model into the physical hardware environment. This sprint focuses on the seamless execution of real-time prediction using live sensor data and the immediate generation of alerts to warn the driver of a potential heart attack.

This sprint ensures that the system can:

- Continuously stream live data from all sensors to the deployed ML model for real-time health status classification.
- Package all functional elements (hardware, sensors, model, alert unit) into a single, portable, power-efficient prototype suited for automotive environments.
- Test the final prototype in simulated in-vehicle conditions to ensure it performs reliably under real-world constraints like vibrations, noise, and varying temperatures.
- Enable data logging and event-triggered memory storage for future review and validation by healthcare professionals.
- Provide user-friendly diagnostics to assist in system troubleshooting, health report generation, or future system upgrades.

By fulfilling these objectives, Sprint 3 will deliver a fully integrated early heart attack detection system that is ready for pilot deployment and real-world testing.

2.3.2.3. Demography

Users:

- Vehicle drivers, especially those in high-risk categories such as long-distance cab drivers, public transport operators and elderly people. Additionally, automobile manufacturers, automobile safety system manufacturers, and healthcare professionals interested in remote health monitoring can benefit from the system.

Location:

- Applicable in both urban and rural areas where the risk of undetected heart-related accidents while driving is high. Especially useful in regions with long highway stretches, limited access to immediate medical care.

2.3.2.4. Features

Table 2.10 depicts the features of sprint 3

Table 2.10 – Sprint 3 Features

Feature	Description	Benefit
Hardware Integration	Combined sensors (MAX30102, GSR, temp sensor) with Arduino Nano 33 BLE Sense and embedded the model.	Enables physical interaction between sensors and predictive system.
Real time heath monitoring	System continuously reads sensor data and makes instant predictions.	Offers timely alerts and rapid response for potential heart attack symptoms.
Embedded ML Inference	Model inference runs on-device using TensorFlow Lite Micro.	Enables offline operation without cloud dependency.
Final Prototype Assembly	All components are housed together in a working, portable device.	Demonstrates proof-of-concept and enables real-world testing.

2.3.3. Architecture Document

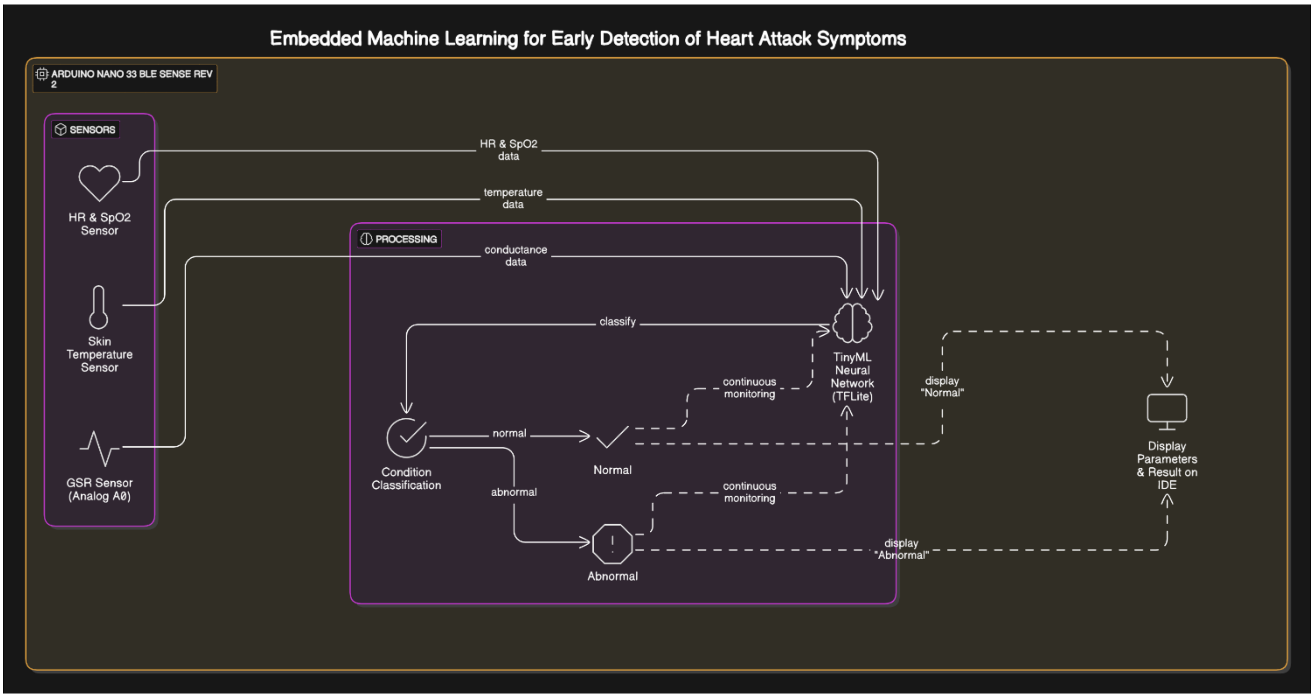


Figure 2.17 – Architecture Diagram for Embedded Machine Learning for Early Detection of Heart Attack Symptoms

Serial Communication Initialisation initiates the system workflow by configuring the serial communication interface to facilitate real-time data display and debugging. This is followed by the I2C Communication Setup, which establishes communication with sensors such as MAX30102 and TMP117 that operate over the I2C protocol. Furthermore, the GSR sensor is integrated into the system to monitor skin conductance, albeit through a distinct interface.

Individually, each sensor is initialised during the Sensor Initialisation phase. The system verifies that each sensor has been properly identified; if a sensor is unable to initialise correctly, an error message is displayed, and the system is suspended to prevent the accumulation of inaccurate readings. In addition, Buffer Access and Initialisation guarantees that the requisite buffers and variables are established to securely store incoming sensor data during operation.

The system subsequently enters the Sensor Data Acquisition phase, during which it continuously reads live data from the MAX30102, TMP117, and GSR sensors. Numerous critical operations are executed during the Preprocessing Phase. Utilising infrared (IR) signals from the MAX30102 sensor, finger detection is implemented to verify the existence of a digit. The IR and red light signals are employed to calculate heart rate and SpO_2 when a finger is detected. In the interim, Temperature Validation guarantees that the TMP117 sensor's readings are within acceptable limits, and GSR Conductance Calculation processes the raw GSR values to produce skin conductance measurements.

The sensor data that has been captured is normalised and further processed during the postprocessing phase. This enables the development of sophisticated physiological metrics, including Heart Rate Variability (HRV), Stress Level Estimation, and Attention Level. After postprocessing, the system advances to Run Inference, where a combined analysis is conducted to ascertain physiological conditions. This analysis is either based on predefined threshold tests or by employing machine learning models.

Lastly, the calculated metrics—including Heart Rate, SpO_2, Temperature, Skin Conductance, Attention, and Stress Levels—are printed or transmitted to a display or monitoring dashboard for real-time observation or further analysis during the Output Generation phase.

2.3.4. Functional Test Cases

Table 2.11 depicts the functional test cases of sprint 3

Table 2.11 - Functional test cases in Sprint 3

Test Case ID	Test Objective	Input	Expected Output	Actual Result	Status
TC_01	Hardware setup verification	Connected sensors and power supply	Board and all sensors are connected successfully	All sensors detected by the board	Pass
TC_02	Real-time sensor read	Live sensor data stream	Real time readings appear in serial monitor	Continuous live readings with serial updates	Pass
TC_03	On-device ML prediction	Streamed physiological data	Model predicts health status instantly	Model infers alert within seconds	Pass
TC_04	End-to-End system integration	Full input-output workflow	Seamless operation model and sensors	Entire workflow runs without interuption	Pass

2.3.5. Daily Call Progress

Figure 2.18 depicts the record of standup meetings conducted for sprint 3.

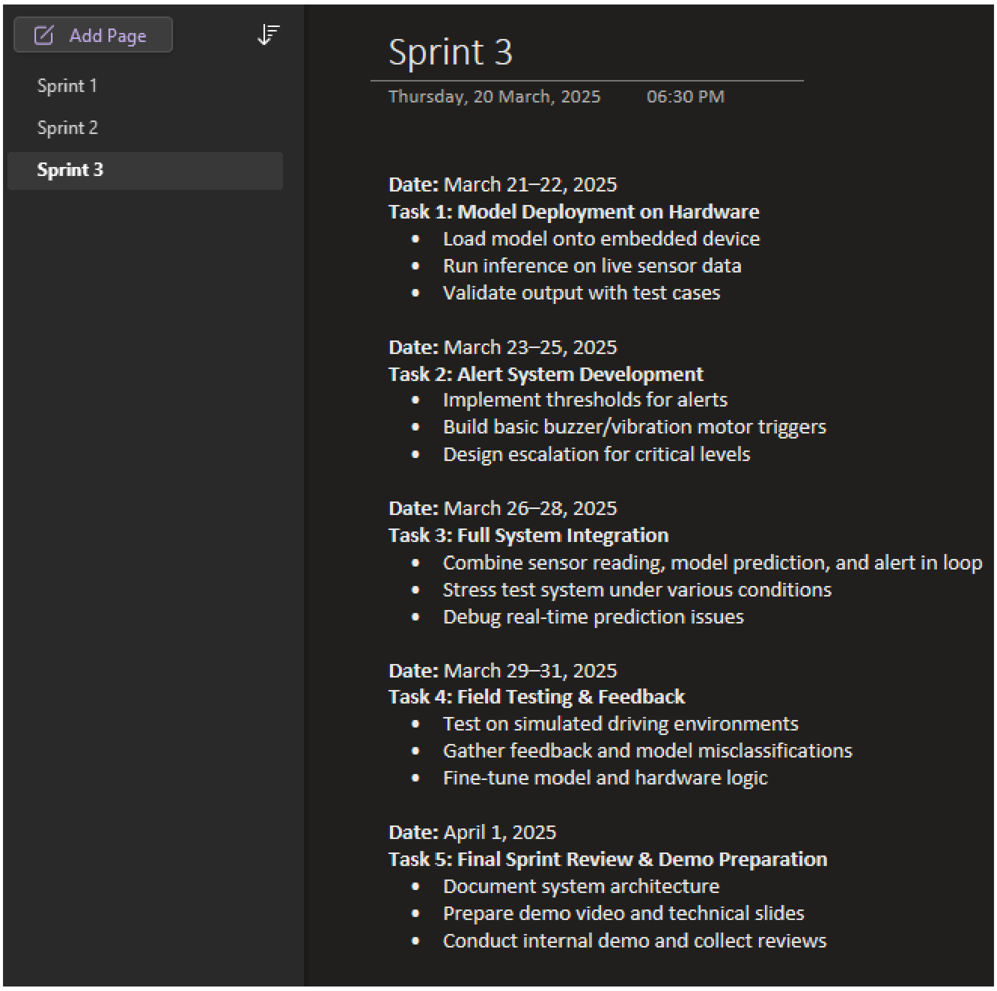


Figure 2.18 - Standup Meetings for Sprint-2

2.3.6. Committed vs Completed User Stories

The figure 2.19 represents the committed vs completed user stories for sprint 3 in a bar graph.

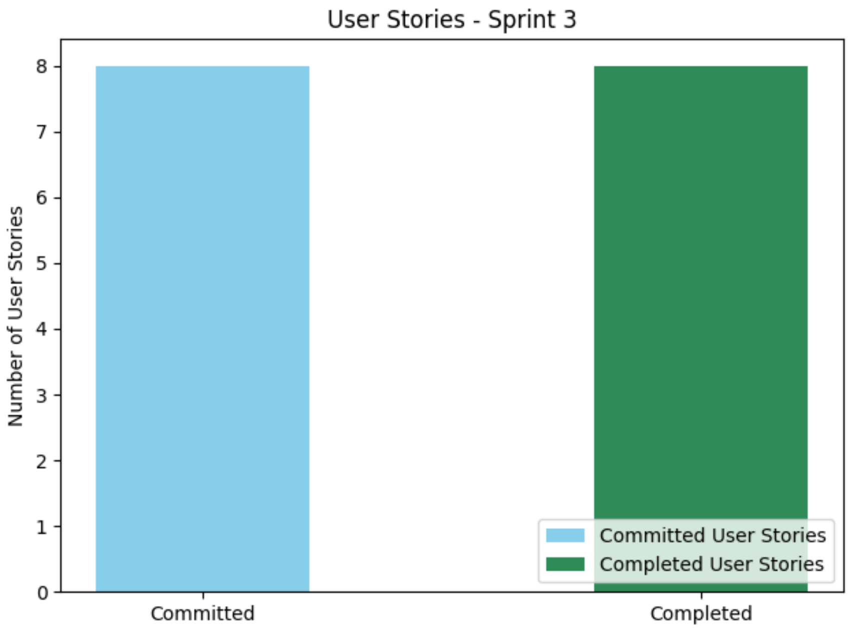


Figure 2.19 - Committed Vs Completed User Stories for Sprint-3

2.3.7. Sprint Retrospective

Table 2.12 depicts the sprint retrospective for sprint 3

Table 2.12 - Sprint 3 Retrospective

Sprint Retrospective-3				
What went well	**What went poorly**	**What ideas do you have**	**How should we take action**	***Guidelines***
Model was successfully deployed on the embedded device.	Limited memory on the microcontroller required model optimization.	Try quantizing the model further or using a simpler architecture.	Use float16 or int8 quantization during TFLite conversion.	Always check memory usage and model size before deployment.
Real-time predictions were	Occasional lag in alert	Implement non-blocking code	Use interrupts or state machines	Prioritize efficiency in

accurate in most test scenarios.	triggering.	and optimize data processing flow.	for faster response.	alert mechanisms.
Hear attack risks were predicted successfully	False positives occurred when noise levels increased.	Add more data samples for training and improve preprocessing filters.	Collect more diverse training data and retrain the model.	Always validate sensor data before feeding it to the model.
All modules were integrated successfully.	Model debugging was difficult due to limited on-device logging.	Log more intermediate values to assist in debugging.	Print prediction probabilities and feature values in debug mode.	Maintain verbose logging during dev and switch to silent in production.

CHAPTER 3

RESULTS AND DISCUSSION

3.1. Project Outcomes

The project culminated in the successful development of a fully functional embedded health monitoring system tailored for drivers, capable of continuously measuring key physiological parameters such as Heart Rate, SpO_2, Body Temperature, and Sweat Level in real time. A robust data acquisition and processing pipeline was established, enabling the collection of clean, structured sensor data which facilitated the training of a lightweight, optimized TensorFlow Lite model.

This model, deployed on the Arduino Nano 33 BLE Sense Rev2 board, accurately classifies incoming physiological signals and instantly identifies abnormal patterns indicative of early heart attack symptoms. Upon detection of such patterns, the system activates real-time emergency alerts via buzzers, allowing for swift response and intervention. The system was designed to be modular, scalable, and power-efficient, ensuring reliability in long-term operations across both urban and remote driving environments.

In alignment with the United Nations Sustainable Development Goal (UN SDG) 3 – Good Health and Well-Being – the project contributes to reducing premature mortality by supporting early detection and real-time health risk notification. Furthermore, the initiative provided the team with interdisciplinary technical exposure, covering biomedical sensor interfacing, embedded systems programming, machine learning deployment, and IoT hardware-software integration, significantly enhancing their practical and research-oriented skills.

3.2. Committed vs Completed User Stories

The figure 3.1 represents the committed vs completed user stories for all the sprints.

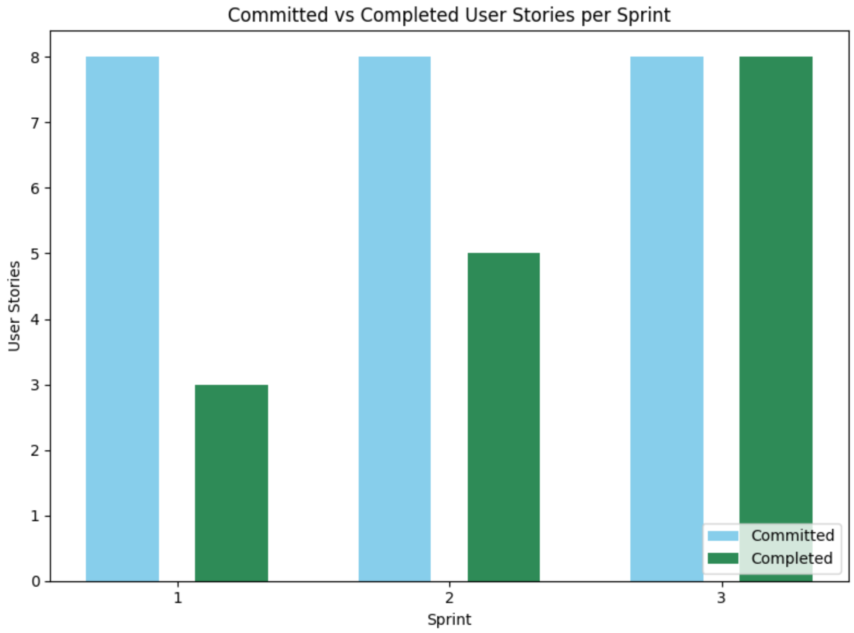


Figure 3.1 - Committed Vs Completed User Stories for All Sprints

CHAPTER 4

CONCLUSION & FUTURE ENHANCEMENTS

This project effectively illustrates the viability of utilising embedded machine learning to identify early symptoms of heart attacks by analysing real-time physiological parameters.

A test accuracy of 96.64% was attained by a lightweight neural network model that was trained on a balanced synthetic dataset, suggesting that it exhibited strong predictive performance. The model was converted to TensorFlow Lite format and optimised, which enables deployment on low-power embedded hardware, such as the Arduino Nano 33 BLE Sense Rev2. The system offers dependable and efficient real-time classification that is appropriate for automobiles by analyzing inputs such as Body Temperature, SpO_2, GSR, and Heart Rate. This method has the potential to significantly improve early intervention in cardiac events. Future improvements will concentrate on more parameters and development as a product.

For future development, the system is envisioned to evolve into a more comprehensive driver health monitoring and emergency alert system. Planned enhancements include:

1. **MAR (Motion Artifact Removal):** To improve the accuracy of physiological readings by filtering out noise caused by motion, ensuring reliable data even when the driver is in motion.
2. **EAR (Eye Aspect Ratio) Monitoring:** To detect signs of drowsiness or fatigue by monitoring the driver's eye movement patterns, enhancing driver safety further.
3. **GPS Integration:** To track the vehicle’s real-time location, allowing for precise geolocation during emergencies.
4. **GSM Module Integration: To** send automated alerts to nearby hospitals and family members when abnormal health parameters or potential heart attack symptoms are detected.

This upgraded system would support real-time emergency response by sharing both health status and location data, increasing the chances of timely medical assistance and potentially saving lives.

APPENDIX

A. PATENT DISCLOSURE FORM

B. SAMPLE CODING

```
#include <Wire.h>
#include <TensorFlowLite.h>
#include <tensorflow/lite/micro/tflite_bridge/micro_error_reporter.h>
#include <tensorflow/lite/micro/micro_interpreter.h>
#include <tensorflow/lite/micro/all_ops_resolver.h>
#include <tensorflow/lite/schema/schema_generated.h>
#include <MAX30105.h>
#include <heartRate.h>
#include <spo2_algorithm.h>
#include <SparkFun_TMP117.h>
#include "model_data.h"  // Include trained TFLite model

// Sensor Initialization
MAX30105 particleSensor;
TMP117 tempSensor;
const int GSR_PIN = A0;

// TensorFlow Lite Variables
constexpr int tensorArenaSize = 20 * 1024;
uint8_t tensorArena[tensorArenaSize];

tflite::MicroErrorReporter errorReporter;
tflite::AllOpsResolver resolver;
const tflite::Model* model;
tflite::MicroInterpreter* interpreter;

void setup() {
    Serial.begin(115200);
    Wire.begin();

    // Initialize MAX30102 Sensor
    if (!particleSensor.begin(Wire, I2C_SPEED_STANDARD)) {
        Serial.println("ERROR: MAX30102 Sensor Not Detected!");
        while (1);
    }
    particleSensor.setup();

    // Initialize TMP117 Sensor
    if (!tempSensor.begin()) {
        Serial.println("ERROR: TMP117 Sensor Not Detected!");
        while (1);
    }

    // Load TFLite Model
    model = tflite::GetModel(tflite_model);
    if (model->version() > TFLITE_SCHEMA_VERSION) {
        Serial.println("ERROR: Model schema version mismatch!");
```

```
        while (1);
    }

    static tflite::MicroInterpreter staticInterpreter(model, resolver,
tensorArena, tensorArenaSize);
    interpreter = &staticInterpreter;

    if (interpreter->AllocateTensors() != kTfLiteOk) {
        Serial.println("ERROR: Tensor allocation failed!");
        while (1);
    }
    Serial.println("TFLite Model Loaded Successfully.");
}

void loop() {
    uint32_t irBuffer[100], redBuffer[100];
    int32_t bufferLength = 100;
    int32_t heartRate = 0, spo2 = 0;
    int8_t validHeartRate = 0, validSpO2 = 0;

    // Read MAX30102 Sensor Data
    for (int i = 0; i < bufferLength; i++) {
        while (!particleSensor.available()) particleSensor.check();
        redBuffer[i] = particleSensor.getRed();
        irBuffer[i] = particleSensor.getIR();
        particleSensor.nextSample();
    }

    // Detect Finger Placement
    long irSignal = irBuffer[bufferLength - 1];
    bool fingerDetected = (irSignal > 5000);

    if (fingerDetected) {
        maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength,
redBuffer, &spo2, &validSpO2, &heartRate, &validHeartRate);
    } else {
        heartRate = 0;
        spo2 = 0;
    }

    // Read TMP117 Temperature Sensor
    float temperature = fingerDetected ? tempSensor.readTempC() : 0;
    bool tempValid = (fingerDetected && !(isnan(temperature) || temperature <
10.0 || temperature > 50.0));
    if (!tempValid) temperature = 0;

    // Read GSR Sensor and Convert to Conductance
    int rawGsrValue = analogRead(GSR_PIN);
```

```
    float gsrValue = (rawGsrValue >= 1000 || rawGsrValue == 0) ? 0.0 : (1.0 /
((3.3 * 1000000.0 / ((float)rawGsrValue * (3.3 / 1023))) - 1000000.0)) *
1000000.0;

    // Prepare TensorFlow Lite Input
    TfLiteTensor* input = interpreter->input(0);
    if (input->type == kTfLiteFloat32 && input->dims->size == 2 && input-
>dims->data[1] == 4) {
        input->data.f[0] = (float)heartRate;
        input->data.f[1] = (float)spo2;
        input->data.f[2] = temperature;
        input->data.f[3] = gsrValue;
    } else {
        Serial.println("ERROR: Input tensor type mismatch!");
        return;
    }

    // Run Inference
    if (interpreter->Invoke() != kTfLiteOk) {
        Serial.println("ERROR: TFLite inference failed!");
        return;
    }

    // Get Output Tensor
    TfLiteTensor* output = interpreter->output(0);
    float heartAttackRisk = fingerDetected ? output->data.f[0] : 0.0;

    // Print Results
    Serial.print("HR: "); Serial.print(validHeartRate ? heartRate : 0);
    Serial.print(" bpm | SpO2: "); Serial.print(validSpO2 ? spo2 : 0);
    Serial.print("% | Temp: "); Serial.print(tempValid ? temperature : 0);
    Serial.print(" °C | GSR: "); Serial.print(gsrValue);
    Serial.print(" | Heart Attack Risk: "); Serial.println(heartAttackRisk);

    delay(5000);
}
```

C. PLAGIARISM REPORT

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **9 Not Cited or Quoted 8%**
  Matches with neither in-text citation nor quotation marks
- **0 Missing Quotations 0%**
  Matches that are still very similar to source material
- **0 Missing Citation 0%**
  Matches that have quotation marks, but no in-text citation
- **0 Cited and Quoted 0%**
  Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
4%	Publications
7%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **9** Not Cited or Quoted 8%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
4%	Publications
7%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	University of Bedfordshire on 2025-04-19	2%
2	Submitted works	Intercollege on 2022-03-13	2%
3	Submitted works	Morgan State University on 2025-02-17	1%
4	Submitted works	Frimley Health NHS Foundation Trust on 2025-02-27	<1%
5	Publication	Temitope Emmanuel Komolafe, Patrice Monkam, Blessing Funmi Komolafe, Nizh...	<1%
6	Publication	Peng Lean Chong, Silvia Ganesan, Poh Kiat Ng, Feng Yuan Kong. "A TRIZ-Adopted ...	<1%
7	Submitted works	RDI Distance Learning on 2025-04-02	<1%