Intelligent Transportation System: Optimal Routing and a Novel OBD Data-Logging System

Project/Thesis ID. 2023: 111

Session: BS Spring 2020

Project Supervisor: Engr. Farheen Qazi

Submitted By:

Rohail Rasheed, Aarish Ahmed Khan,

Alyan Asif Ali, Hussain Shams

Department of Software Engineering

Sir Syed University of Engineering and Technology

Certification

This is to certify that **Rohail Rasheed** having **2020S -SE-049**, **Aarish Ahmed** having **2020S -SE-030**, **Alyan Asif** having **2020S -SE-045**, and **Hussain Shams** having **2020S -SE-011** have successfully completed the final project **Intelligent Transportation System: Optimal Routing and a Novel OBD Data-Logging System** at the **Sir Syed University of Engineering and Technology**, to fulfill the partial requirement of the degree **Bachelor of Science in Software Engineering.**

External Examiner

[Name of Examiner]

[Designation]

Project Supervisor

Engr. Farheen Qazi

Assistant Professor

Chairman

Department of Software Engineering, Sir Syed University of Engineering and Technology

Intelligent Transportation System: Optimal Routing and a Novel OBD Data-Logging System

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	**Description of SDG**	**SDG No**	**Description of SDG**
SDG 1	No Poverty	SDG 9 ✓	Industry, Innovation, and Infrastructure
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11 ✓	Sustainable Cities and Communities
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17 ✓	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	✓
4	Familiarity of issues	Involve infrequently encountered issues	✓
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	✓
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	✓
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	✓
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	✓
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	✓
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

The escalating demand for vehicles in response to population growth has led to an increase in the number of cars on the road, posing risks associated with unexpected failures. In response to this challenge, the automotive industry has embraced a concept inspired by the Aircraft Industry, introducing an "Automated Self-Diagnostic Mechanism." This innovative system employs predictive analytics to detect potential vehicle failures and prognostics, mitigating the hazards of unforeseen roadside breakdowns.

The approach of our proposed project involves retrieving data from the Vehicle On-Board Diagnostics (OBD) through a microprocessor. This data is then processed using Machine Learning (ML) and Deep Learning (DL) algorithms for training and testing, with cloud services utilized for seamless record-keeping. The resulting model becomes adept at predicting both expected and unexpected vehicle failures based on historical OBD data. It categorizes these issues into "GO ITEMS," faults that allow for delayed maintenance, and "NO GO ITEMS," faults requiring immediate attention. Furthermore, the system facilitates the scheduling of maintenance appointments with approved vendors. Users and drivers can conveniently visualize these diagnostic parameters through Power Bi Desktop and Mobile Applications, gaining insights through graphical charts for informed decision-making. This transformative approach enhances vehicle safety, minimizes unexpected breakdowns, and streamlines the maintenance process for a more secure and efficient driving experience.

Keywords: On Board Diagnosis (OBD), Electronic Control Unit (ECU), Machine Learning, Artificial Intelligence, Vehicle Faults, Faults Diagnosis, Fault Prognosis, Maintenance, Vendors, GO items, NO GO items, automotive, vehicles, transportation.

Undertaking

I certify that the project **Intelligent Transportation System: Optimal Routing and a Novel OBD Data-Logging System** is our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/ referred.

Rohail Rasheed

2020S-SE-049

Aarish Ahmed

2020S-SE-030

Alyan Asif

2020S-SE-045

Hussain Shams

2020S-SE-011

Acknowledgement

We truly acknowledge the cooperation and help make by **Engr. Farheen Qazi, Assistant Professor** of **Sir Syed University of Engineering and Technology.** He has been a constant source of guidance throughout the course of this project.

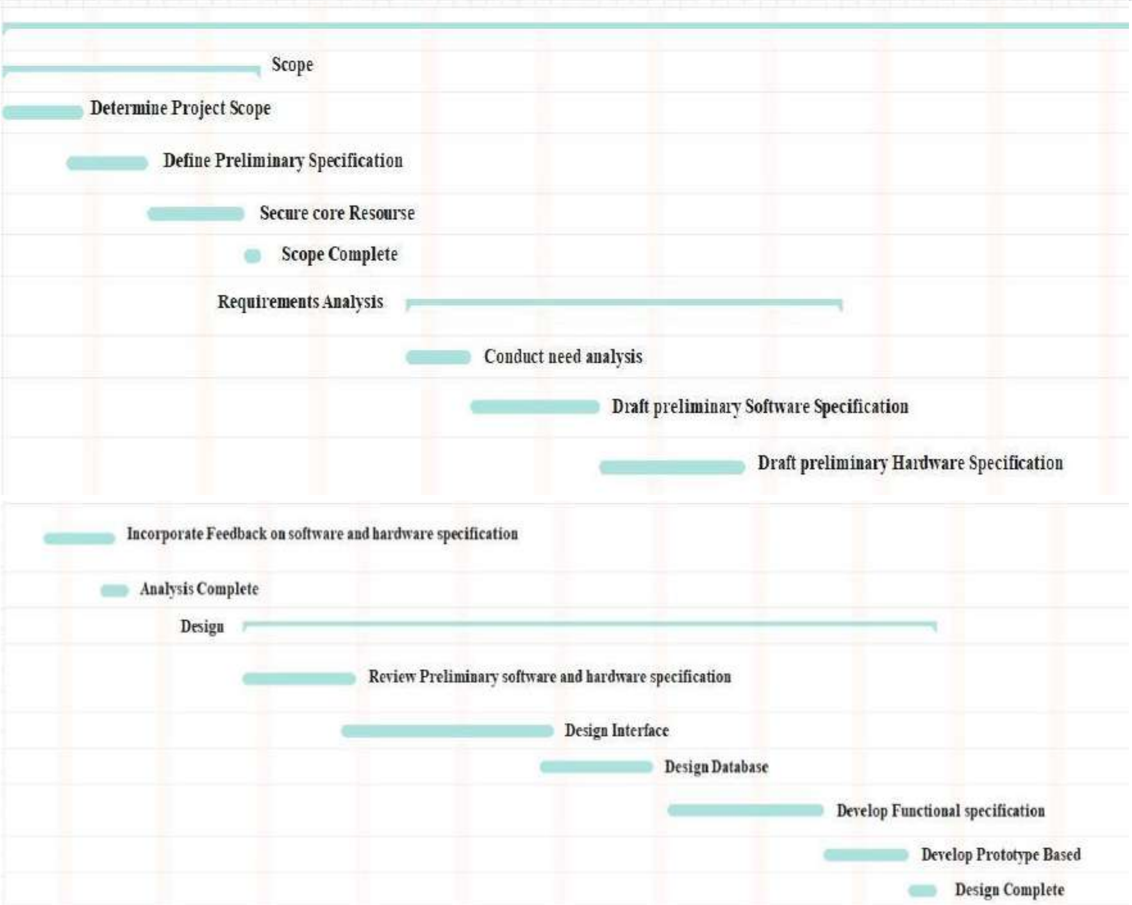
We are also thankful to our friends and families whose silent support led us to complete our project.

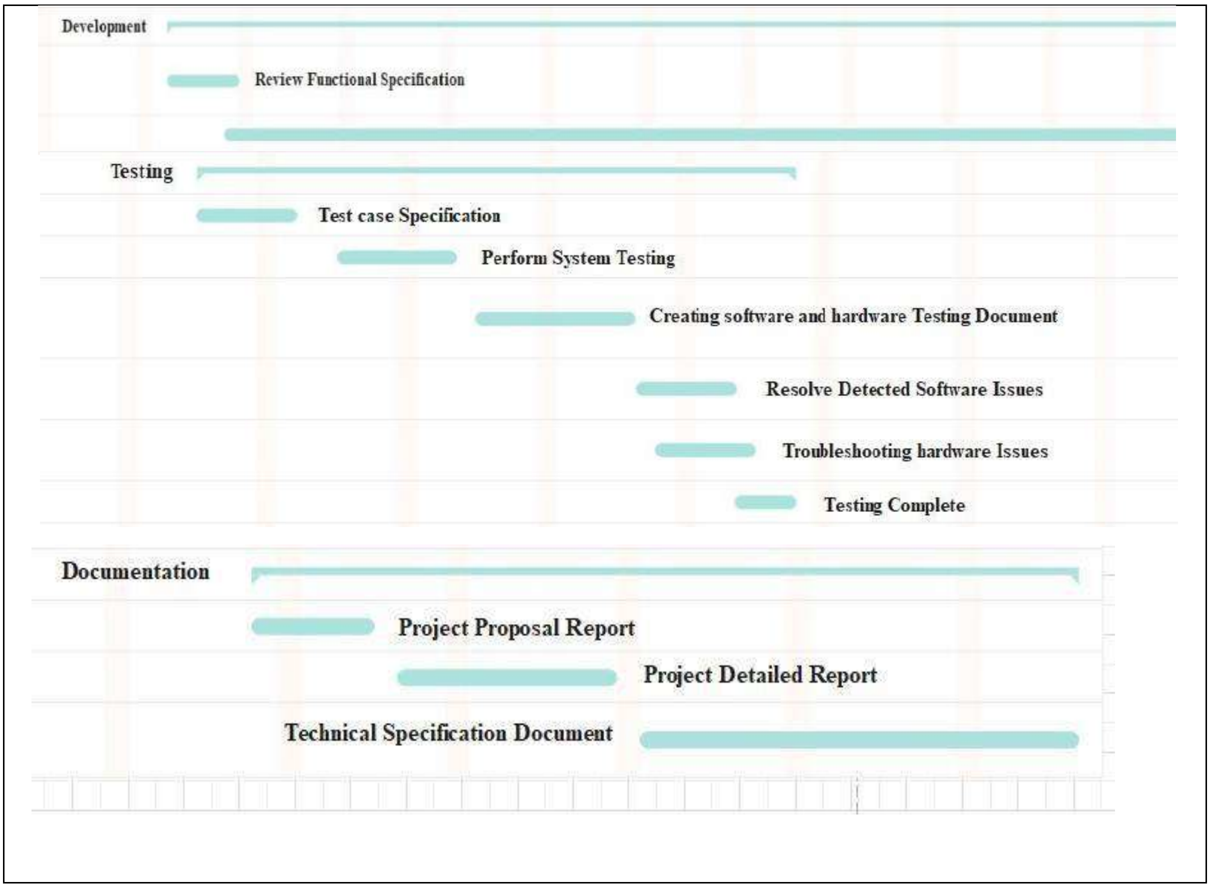
Table of Contents

List of Tables

ID	Task Name	Start	Finish	Duration	Responsibility
1	⊟ **FYP Development**	2022-11-20	2023-11-21	315.0 d.	
2	⊟ **Scope**	2022-11-20	2022-12-05	14.0 d.	
3	Determine Project Scope	2022-11-20	2022-11-24	5.0 d.	Rohail Rasheed
4	Define Preliminary Specification	2022-11-24	2022-11-28	4.0 d.	Alyan Asif
5	Secure core Resourse	2022-11-29	2022-12-04	5.0 d.	Aarish Ahmed
6	Scope Complete	2022-12-05	2022-12-05	1.0 d.	Hussain Shams
7	⊟ **Analysis/Software and Hardware Requirements**	2022-12-15	2023-01-10	23.0 d.	
8	Conduct need analysis	2022-12-15	2022-12-18	3.0 d.	Hussain Shams
9	Draft preliminary Software Specification	2022-12-19	2022-12-26	7.0 d.	Aarish Ahmed
10	Draft preliminary Hardware Specification	2022-12-27	2023-01-04	8.0 d.	Rohail Rasheed
11	Incorporate Feedback on software and hardware specification	2023-01-05	2023-01-09	4.0 d.	Alyan Asif
12	Analysis Complete	2023-01-09	2023-01-10	2.0 d.	Aarish Ahmed
13	⊟ **Design**	2023-01-19	2023-03-08	42.0 d.	
14	Review Preliminary software and hardware specification	2023-01-19	2023-01-26	7.0 d.	Rohail Rasheed
15	Design Interface	2023-01-26	2023-02-09	13.0 d.	Alyan Asif
16	Design Database	2023-02-09	2023-02-16	7.0 d.	Rohail Rasheed
17	Develop Functional specification	2023-02-18	2023-02-28	10.0 d.	Hussain Shams
18	Develop Prototype Based	2023-03-01	2023-03-06	5.0 d.	Alyan Asif
19	Design Complete	2023-03-07	2023-03-08	2.0 d.	Aarish Ahmed
20	⊟ **Development**	2023-03-12	2023-08-22	141.0 d.	
21	Review Functional Specification	2023-03-12	2023-03-16	5.0 d.	Aarish Ahmed,Hussain Shams
22	Develop Code	2023-03-16	2023-07-05	96.0 d.	All Teams Members
23	Perform Initial Testing	2023-07-10	2023-08-06	24.0 d.	Alyan Asif,Rohail Rasheed
24	Development Complete	2023-08-13	2023-08-22	9.0 d.	All Teams Members

25	⊟ **Testing**	2023-09-05	2023-10-04	26.0 d.	
26	Test case Specification	2023-09-05	2023-09-09	4.0 d.	Hussain Shams
27	Perform System Testing	2023-09-12	2023-09-17	5.0 d.	Alyan Asif
28	Creating software and hardware Testing Document	2023-09-19	2023-09-26	7.0 d.	Aarish Ahmed
29	Resolve Detected Software Issues	2023-09-27	2023-10-01	4.0 d.	Rohail Rasheed
30	Troubleshooting hardware Issues	2023-09-28	2023-10-02	4.0 d.	Rohail Rasheed
31	Testing Complete	2023-10-02	2023-10-04	3.0 d.	All Teams Members
32	⊟ **Documentation**	2023-10-19	2023-11-21	29.0 d.	
33	Project Proposal Report	2023-10-19	2023-10-23	4.0 d.	Aarish Ahmed
34	Project Detailed Report	2023-10-25	2023-11-02	8.0 d.	Alyan Asif
35	Technical Specification Document	2023-11-04	2023-11-21	16.0 d.	Rohail Rasheed




Development
Review Functional Specification
Testing
Test case Specification
Perform System Testing
Creating software and hardware Testing Document
Resolve Detected Software Issues
Troubleshooting hardware Issues
Testing Complete
Documentation
Project Proposal Report
Project Detailed Report
Technical Specification Document

List of Figures

Figure 1

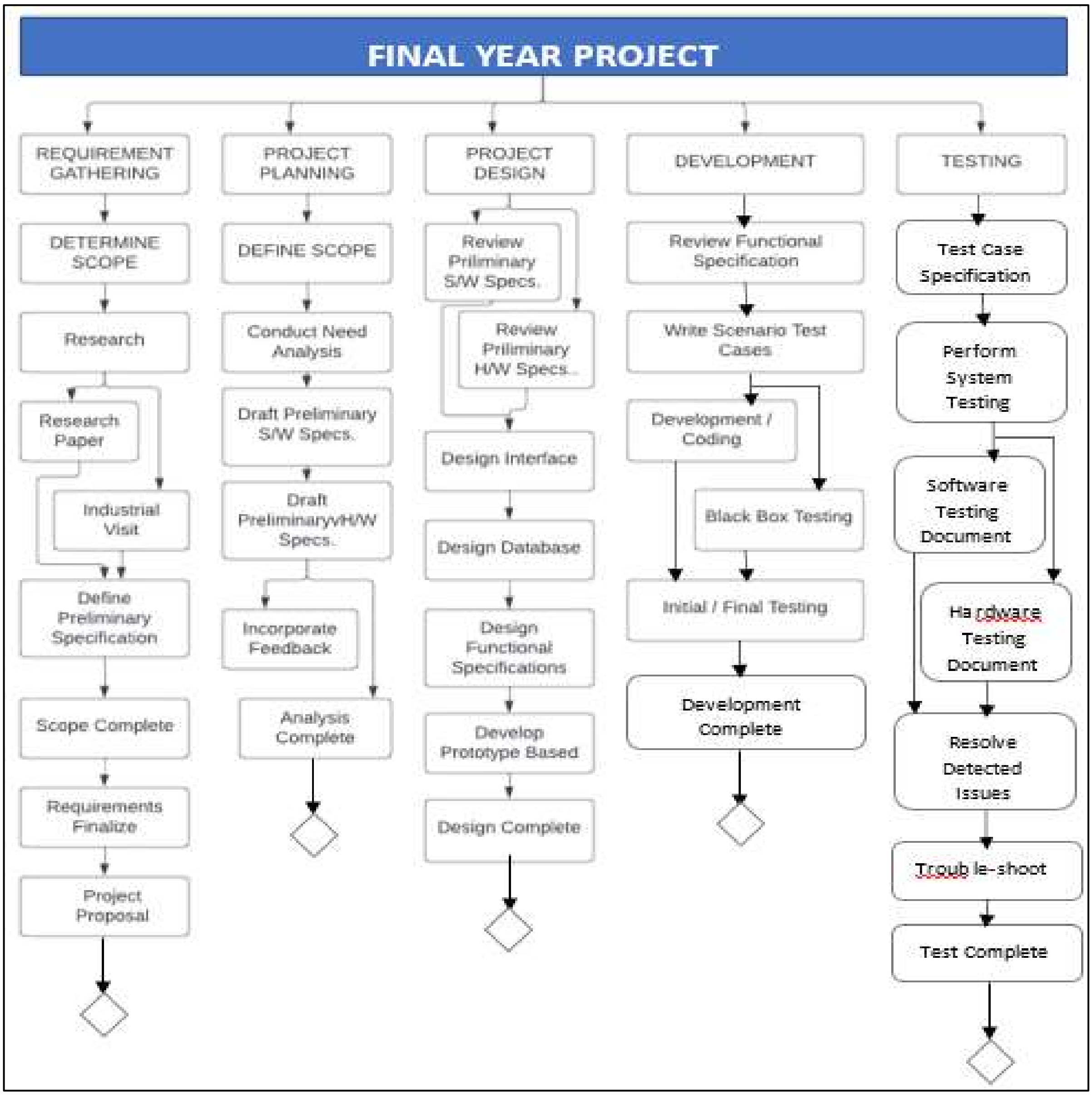


Figure 2

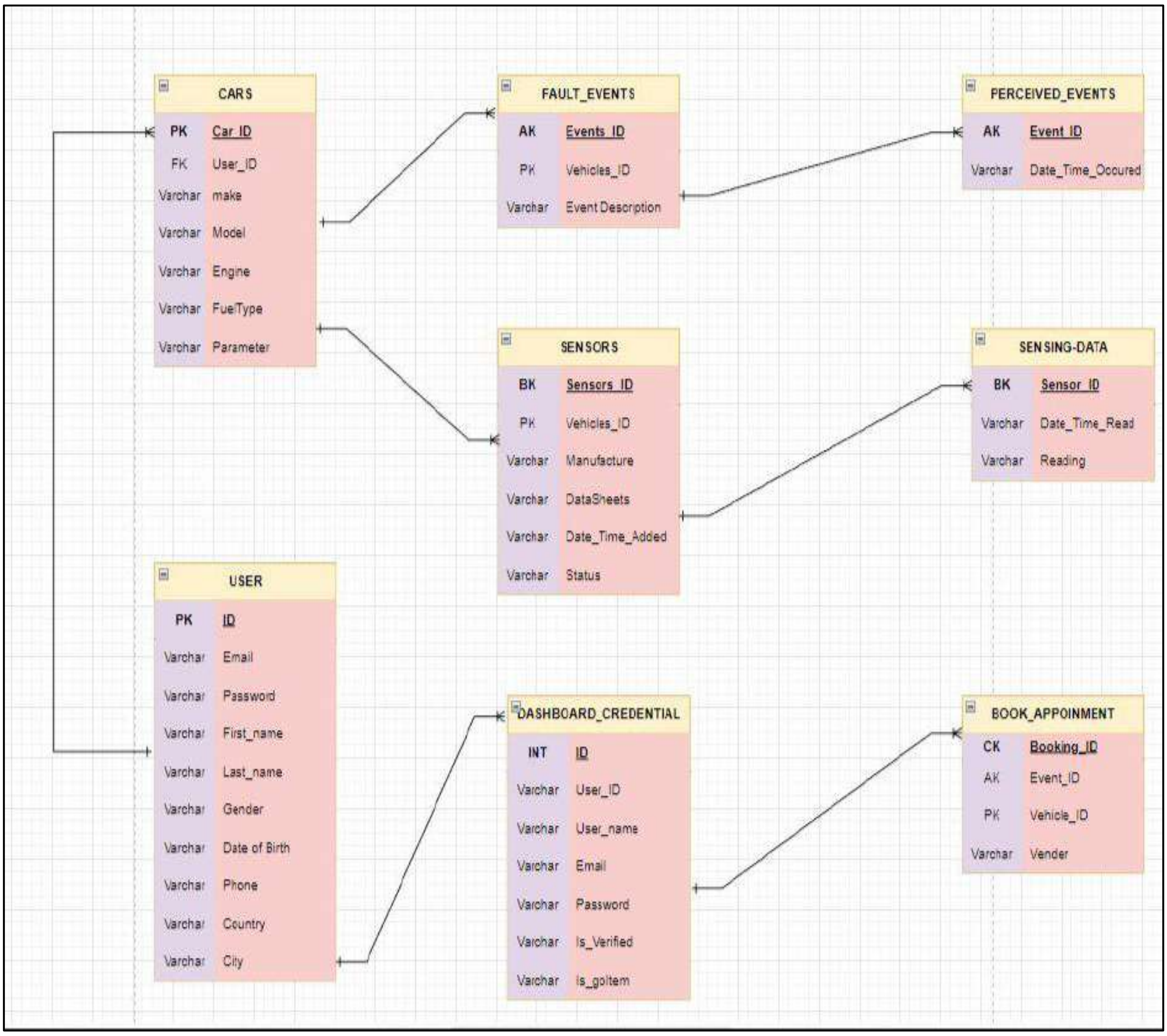

CARS
PK Car_ID
FK User_ID
Varchar make
Varchar Model
Varchar Engine
Varchar FuelType
Varchar Parameter
FAULT_EVENTS
AK Events_ID
PK Vehicles_ID
Varchar Event Description
PERCEIVED_EVENTS
AK Event_ID
Varchar Date_Time_Occured
SENSORS
BK Sensors_ID
PK Vehicles_ID
Varchar Manufacture
Varchar DataSheets
Varchar Date_Time_Added
Varchar Status
SENSING-DATA
BK Sensor_ID
Varchar Date_Time_Read
Varchar Reading
USER
PK ID
Varchar Email
Varchar Password
Varchar First_name
Varchar Last_name
Varchar Gender
Varchar Date of Birth
Varchar Phone
Varchar Country
Varchar City
DASHBOARD_CREDENTIAL
INT ID
Varchar User_ID
Varchar User_name
Varchar Email
Varchar Password
Varchar Is_Verified
Varchar Is_goItem
BOOK_APPOINMENT
CK Booking_ID
AK Event_ID
PK Vehicle_ID
Varchar Vender

Figure 3

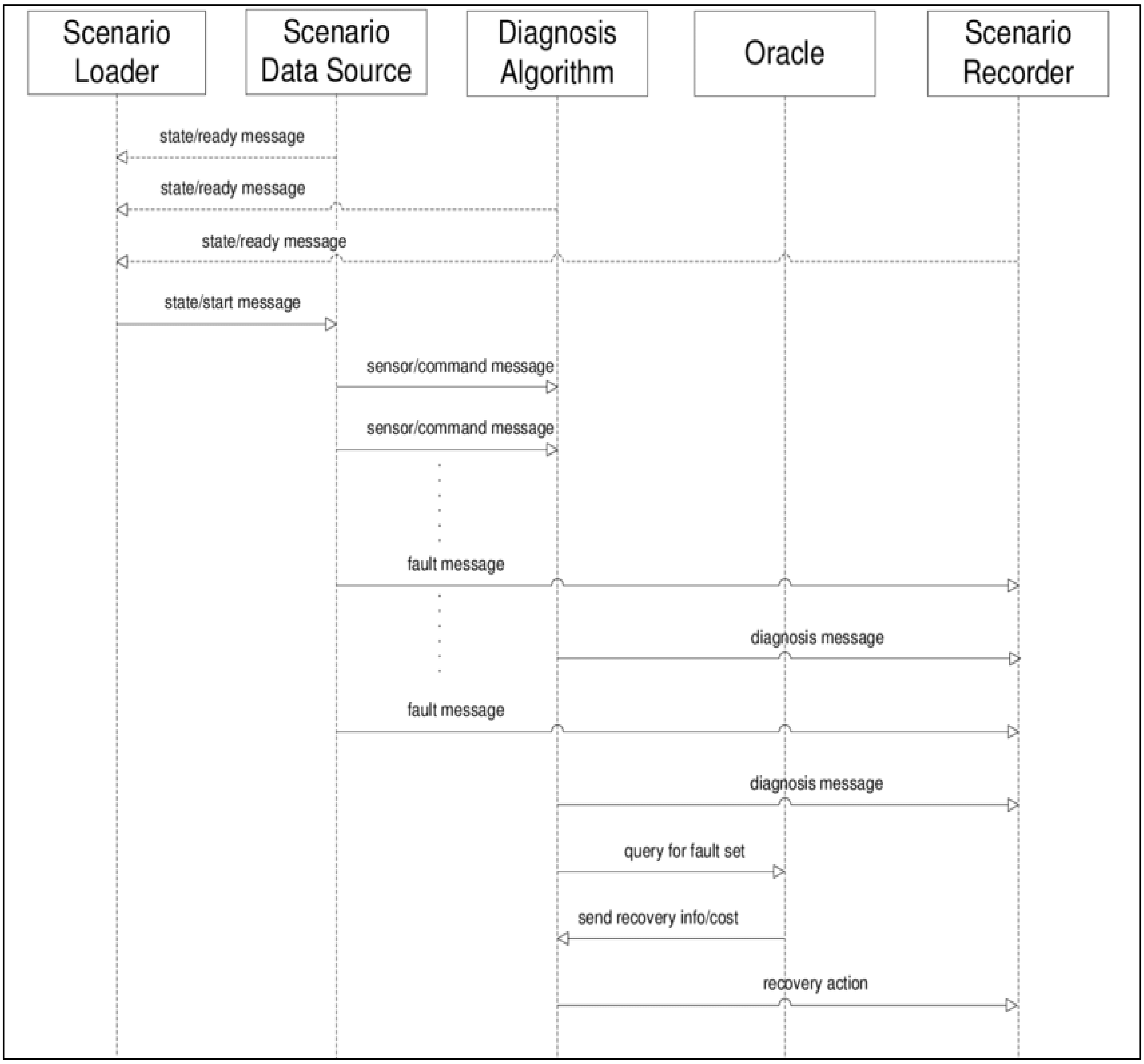
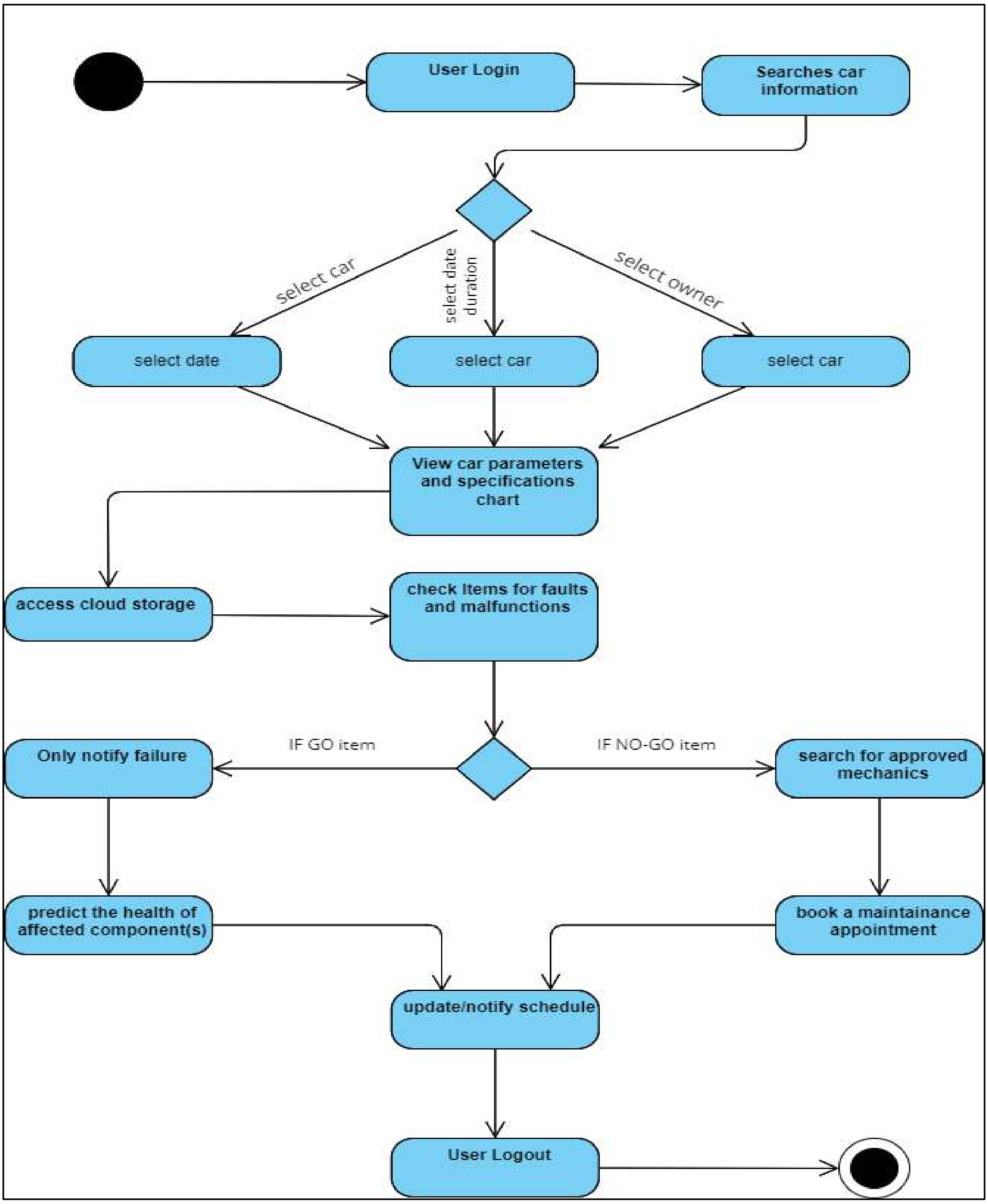

Scenario Loader
Scenario Data Source
Diagnosis Algorithm
Oracle
Scenario Recorder
state/ready message
state/ready message
state/ready message
state/start message
sensor/command message
sensor/command message
fault message
diagnosis message
fault message
diagnosis message
query for fault set
send recovery info/cost
recovery action

Figure 4


User Login
Searches car information
select car
select date duration
select owner
select date
select car
select car
View car parameters and specifications chart
access cloud storage
check Items for faults and malfunctions
IF GO item
IF NO-GO item
Only notify failure
search for approved mechanics
predict the health of affected component(s)
book a maintainance appointment
update/notify schedule
User Logout

List of Acronyms

OBD: On-Board Diagnostics
DTC: Diagnostic Trouble Code
ECU: Electronic Control Unit
ML: Machine Learning
DL: Deep Learning
API: Application Programming Interface
UI: User Interface
OS: Operating System
IoT: Internet of Things
GPS: Global Positioning System
HTTPS: Hypertext Transfer Protocol Secure
SDK: Software Development Kit
SVM: Support Vector Matrix
KNN: K Nearest Neighbor
ANN: Artificial Neural Network
MCC: Matrix Coefficient Correlation

List of Equations

Equation 1: $\sum_{i=1}^{n}\sqrt{}(X_i \ - \ Y_i)$

Equation 2: $\frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \cdot\cdot \ \cdot\cdot \ \beta_n X_n)}}$

Equation 3: MicrocontrollerRead (OBD_Port)

Equation 4: Original Dataset + Transformation

Chapter 1

1.1 Introduction

In a world where the automotive landscape is continually expanding to meet the demands of a growing population, the surge in the number of vehicles, particularly cars, has introduced a host of challenges. This proliferation has given rise to various unforeseen and potentially hazardous factors on our roads, stemming from sudden malfunctions and failures in vehicles. The complex consequences include not only accidents and injuries but also the broader spectrum of faults that compromise road safety. In response to this pressing need for preventive measures, a concept borrowed from the aircraft industry has been seamlessly integrated into the automotive industry, unveiling the innovative "Automated Self-Diagnostic Mechanism." This groundbreaking system is designed to proactively engage with prognostics, foreseeing and mitigating potential vehicle malfunctions that might unfold in the future (before original occurrence) and classify them into ‘GO ITEMS’ and ‘NO-GO ITEMS’. Additionally, the system is capable of booking maintenance appointments automatically to prevent the risk of over-maintenance and under-maintenance.

1.2 Statement of the problem

The steady rise in the number of vehicles, driven by the increasing demands of a growing population, has given rise to a concerning array of challenges on our roads. The surge in car ownership has resulted in unforeseen and often hazardous incidents stemming from sudden malfunctions, leading to accidents, injuries, and a multitude of faults. Traditional approaches to vehicle maintenance are reactive, addressing issues only after they've occurred, thereby failing to mitigate the risks associated with unexpected breakdowns. This critical gap in proactive vehicle health management necessitates an innovative solution.

To tackle this problem, we introduce the "Automated Self-Diagnostic Mechanism," drawing inspiration from the aviation industry's proactive maintenance practices. However, the integration of this advanced system into the automotive sector presents its own set of challenges. The need to anticipate and prevent future vehicle malfunctions requires a sophisticated interplay of data collection from On-Board Diagnostics (OBD), utilization of Machine Learning (ML) and Deep Learning (DL) Algorithms for predictive analytics, and seamless scheduling of maintenance appointments. Additionally, ensuring a user-friendly interface for drivers to understand, engage with, and act upon these predictions adds another layer of complexity. This project aims to bridge these gaps and revolutionize the traditional paradigm of vehicle maintenance by providing an intelligent, anticipatory, and user-centric solution for enhanced road safety and vehicle reliability.

1.3 Goals/Aims & Objectives

1.3.1 **Develop an Efficient Self-Diagnostic System:** Design and implement an automated self-diagnostic mechanism that seamlessly fetches and analyzes real-time data from the Vehicle On-Board Diagnostics (OBD) through the Raspberry Pi microprocessor. The system will utilize advanced Machine

Learning (ML) and Deep Learning (DL) algorithms to identify patterns and trends in the OBD data, enabling the accurate prediction of potential vehicle malfunctions. The goal is to create a robust and efficient system that empowers vehicle owners and fleet managers with proactive insights into their vehicles' health, facilitating timely maintenance and reducing the risk of unexpected breakdowns.

1.3.2 **Categorize and Prioritize Maintenance Tasks:** The developed system will categorize predicted faults into "GO ITEMS" and "NO-GO ITEMS" based on their urgency and criticality. "GO ITEMS" represent faults that can be temporarily delayed for maintenance without immediate consequences, while "NO-GO ITEMS" indicate critical faults that require immediate attention and cannot be postponed. By providing this categorization, the system will enable users to prioritize maintenance tasks effectively, optimizing vehicle performance and minimizing potential safety risks. Additionally, the system will facilitate seamless appointment scheduling with approved vendors for "NO-GO ITEMS," ensuring timely resolution of critical faults and enhancing road safety.

1.4 Motivation

The project's motivation lies in addressing the critical challenges associated with reactive vehicle maintenance and enhancing the overall safety and reliability of vehicles on the road. Traditional maintenance practices often lead to unexpected breakdowns and costly repairs, impacting vehicle owners' convenience and financial burden. By developing an "Automated Self-Diagnostic Mechanism" utilizing the data from Vehicle On-Board Diagnostics (OBD) through a Raspberry Pi microprocessor and employing advanced Machine Learning (ML) and Deep Learning (DL) algorithms, the system aims to proactively predict potential vehicle malfunctions. This proactive approach will empower vehicle owners and fleet managers to stay ahead and prioritize maintenance tasks efficiently.

Furthermore, the project's motivation stems from the need to improve road safety and prevent accidents caused by unexpected vehicle failures. By promptly identifying and addressing critical faults through the user-friendly Desktop and Mobile Application, the project strives to reduce the likelihood of safety-critical incidents. Moreover, the system's ability to schedule appointments with approved vendors for "NO-GO ITEMS" ensures that urgent maintenance needs are swiftly attended to, minimizing safety risks and potential accidents on the roads. Ultimately, this project aims to bring a paradigm shift in vehicle maintenance, optimizing performance, reducing downtime, and fostering a safer and more reliable driving experience for vehicle owners and drivers alike.

1.5 Assumption and Dependencies

Assumptions and dependencies for the "Automated Self-Diagnostic Mechanism" project are factors or conditions that are considered as true or necessary for the successful implementation of the project. These assumptions and dependencies provide a foundation for planning and executing the project. Here are some potential assumptions and dependencies for the project scope described:

Assumptions:

- **Availability of OBD-Equipped Vehicles:** The project assumes that a significant number of vehicles have OBD systems installed and that accessing the OBD data is feasible for a substantial portion of the target user base.
- **Access to Relevant OBD Data:** The project assumes that the necessary OBD data required for predicting vehicle failures, such as diagnostic trouble codes and vehicle parameters, is available and accessible through the OBD system. This assumes that the OBD system is functioning properly and transmitting accurate data.
- **Adequate Representation of Failure Patterns*:*** The project assumes that the historical OBD data collected from vehicles provides an adequate representation of failure patterns and trends. It assumes that the collected data is diverse enough to train ML/DL algorithms effectively.
- **Availability of Approved Vendors:** The project assumes that there are approved vendors or service centers available to address vehicle maintenance and repairs when immediate attention is required. It assumes that the necessary infrastructure is in place to schedule appointments with these vendors.
- **User Adoption and Engagement:** The project assumes that users will be willing to adopt and actively engage with the self-diagnostic mechanism. It assumes that users will find value in the system's predictions, categorizations, and appointment scheduling features, leading to sustained usage and feedback.

Dependencies:

- **Vehicle Compatibility and Cooperation*:*** The project depends on the cooperation of vehicle manufacturers to ensure that their vehicles are compatible with the self-diagnostic mechanism. This includes maintaining compatibility with various OBD systems, protocols, and communication standards across different vehicle models and manufacturers.
- **OBD Data Accuracy and Reliability*:*** The project depends on the accuracy and reliability of the OBD data collected from vehicles. The data quality affects the performance and effectiveness of the ML/DL algorithms used for predicting failures. The project depends on vehicles providing consistent and reliable data through their OBD systems.

- **Availability of ML/DL Algorithms and Libraries**: The project depends on the availability of suitable ML/DL algorithms and libraries for training and testing purposes. It relies on the availability of robust and well-established algorithms and tools to build accurate prediction models based on the OBD data.
- **Data Privacy and Security Compliance:** The project depends on compliance with data privacy and security regulations to protect user data and ensure secure data handling practices. It depends on the availability of resources and frameworks that facilitate secure data storage, transmission, and access control.
- **Vendor Integration Capabilities***:* The project depends on the integration capabilities and cooperation of approved vendors or service centers for scheduling appointments. It relies on vendors providing the necessary APIs, integration protocols, or communication channels to seamlessly connect with the self-diagnostic mechanism.
- **User Device Compatibility:** The project depends on the compatibility of the developed applications with various user devices, including desktop computers, smartphones, and tablets. It relies on the availability of compatible operating systems, software libraries, and development frameworks to ensure a consistent user experience across devices.

1.6 Methods

- **#1:** The project involves the method of fetching data from the Vehicle On-Board Diagnostics (OBD) using a Raspberry Pi microprocessor. The Raspberry Pi acts as an intermediary, collecting real-time data from the vehicle's OBD system. This method ensures a continuous flow of information for analysis.
- **#2:** Data collected from the OBD is dispatched to Machine Learning (ML) and Deep Learning (DL) Algorithms for Training and Testing purposes. This method involves sending the collected data to algorithms designed to learn and analyze patterns. It includes both training the model to make predictions and testing its accuracy.
- **#3:** The algorithms categorize the predicted vehicle failures into "GO ITEMS" (can be delayed for maintenance) and "NO-GO ITEMS" (cannot be delayed). The ML/DL algorithms use predefined criteria to categorize potential faults based on their urgency. This method assists in prioritizing maintenance actions.
- **#4:** If a fault is categorized as a "NO-GO ITEM," the system facilitates the scheduling of maintenance appointments with approved vendors. This method ensures that immediate attention is given to critical faults by automatically coordinating maintenance appointments with trusted service providers.
- **#5:** Users/drivers experience the functionalities and visualizations on Desktop and Mobile Applications (and potentially a web application). This involves the design and implementation of user interfaces for applications, allowing users to interact with and interpret the diagnostic insights provided by the system.
- **#6:** Cloud services are used to maintain records of the collected data. This method ensures that data is securely stored, easily accessible, and can be scaled as needed, leveraging the benefits of cloud computing.

- **#7:** The project incorporates continuous monitoring and a feedback loop, including user feedback, supervisor input, and committee reviews. This method involves regularly evaluating the system's performance, incorporating feedback, and making necessary adjustments to improve its capabilities and accuracy over time.

1.7 Report Overview

The "Automated Self-Diagnostic Mechanism" project represents a pioneering effort in the domain of vehicle maintenance and safety, strategically addressing the escalating risks associated with the growing number of vehicles on the road. The report succinctly encapsulates the project's raison d'être, illuminating the imperative for a proactive solution to counter the challenges posed by unexpected breakdowns, malfunctions, and potential accidents. With a keen focus on anticipating and categorizing potential vehicle failures, the project aims to redefine the paradigms of automotive maintenance.

The project's methods are meticulously detailed, starting with the sophisticated process of collecting real-time data from Vehicle On-Board Diagnostics (OBD) through the utilization of the Raspberry Pi microprocessor. This data undergoes a transformative journey, being dispatched to Machine Learning (ML) and Deep Learning (DL) Algorithms for Training and Testing, ultimately leading to the categorization of faults into "GO ITEMS" and "NO-GO ITEMS." User experience is emphasized, emphasizing the creation of user-friendly interfaces for Desktop and Mobile Applications, ensuring that drivers can easily interpret and act upon diagnostic insights.

Crucial to the project's narrative are the identified assumptions and dependencies. The report brings attention to the critical assumption of reliable OBD data and the dependency on approved vendors for scheduling maintenance appointments. The success of the ML/DL Algorithms is an assumption central to the project's predictive capabilities, and the stability of cloud services is a vital dependency for seamless record-keeping.

However, the report underscores the overarching objectives of the project, emphasizing its commitment to continuous monitoring and improvement. By incorporating user feedback, supervisor input, and committee reviews, the project aims not only to meet but exceed expectations, creating a dynamic self-diagnostic system that adapts and evolves over time. The "Automated Self-Diagnostic Mechanism" emerges as a transformative initiative poised to elevate road safety and redefine the standards of vehicle maintenance in an ever-evolving technological landscape.

Chapter 2

2.1 Project Planning and Management

2.1.1 SWOT Analysis:

SWOT examination is a helpful instrument for assessing the qualities, shortcomings, potential open doors, and dangers related with a task. The "Automated Self-Diagnostic Mechanism" project's SWOT analysis is as follows:

✓ **Strengths:**

- **Improved Maintenance Planning:** The undertaking will empower drivers to expect and anticipate potential vehicle disappointments, taking into consideration proactive support and diminishing the gamble of unforeseen breakdowns.
- **Cost Savings:** By identifying and addressing maintenance needs in advance, the project can potentially save costs associated with major repairs and unplanned downtime.
- **Enhanced Safety:** Early detection and resolution of critical issues through the self-diagnostic mechanism can enhance vehicle safety and reduce the risk of accidents.
- **User-Friendly Interface:** The development of user-friendly desktop and mobile applications will enhance the user experience and make it easier for drivers to access and interpret the diagnostic information.

✓ **Weaknesses:**

- **Reliance on OBD Data:** The accuracy and completeness of the predictions heavily depend on the quality of the data collected from the OBD system. If the OBD system itself is faulty or provides incomplete information, it may lead to inaccurate predictions.

- **Limited Predictive Accuracy:** Although ML/DL algorithms can learn patterns from historical data, there may be limitations to the accuracy of the predictions. Factors such as unforeseen mechanical issues or changes in vehicle usage patterns may affect the reliability of the predictions.

✓ **Opportunities:**

- **Competitive Advantage:** The project can provide a unique selling point for vehicle manufacturers or service providers by offering an automated self-diagnostic mechanism that enhances customer experience and vehicle reliability.

- **Expanded Service Offerings:** The project can open up opportunities for service providers to offer additional services such as appointment scheduling with approved vendors, creating new revenue streams.

- **Data Analytics Insights:** The collected OBD data can be utilized for broader data analytics purposes, such as identifying trends, optimizing maintenance schedules, or improving vehicle design and performance.

✓ **Thread*s:***

- **Data Privacy and Security:** Collecting and storing OBD data raises concerns about data privacy and security. Adequate measures must be in place to protect sensitive information from unauthorized access or misuse.

- **Regulatory Compliance:** The project must comply with relevant regulations and standards related to data privacy, security, and automotive diagnostics, ensuring that all legal requirements are met.

- **Technical Challenges:** Developing accurate ML/DL algorithms and integrating them into a user-friendly application may pose technical challenges, such as algorithm training, real-time data processing, and efficient visualization.

- **User Adoption:** The success of the project relies on drivers' willingness to adopt and use the self-diagnostic mechanism. Resistance to change or lack of awareness among users could pose a challenge to the project's acceptance and adoption. We can get a better understanding of the project's potential by considering its strengths, weaknesses, opportunities, and threats. They can also make well-informed choices about how to deal with challenges and take advantage of opportunities.

2.2 Project Overview

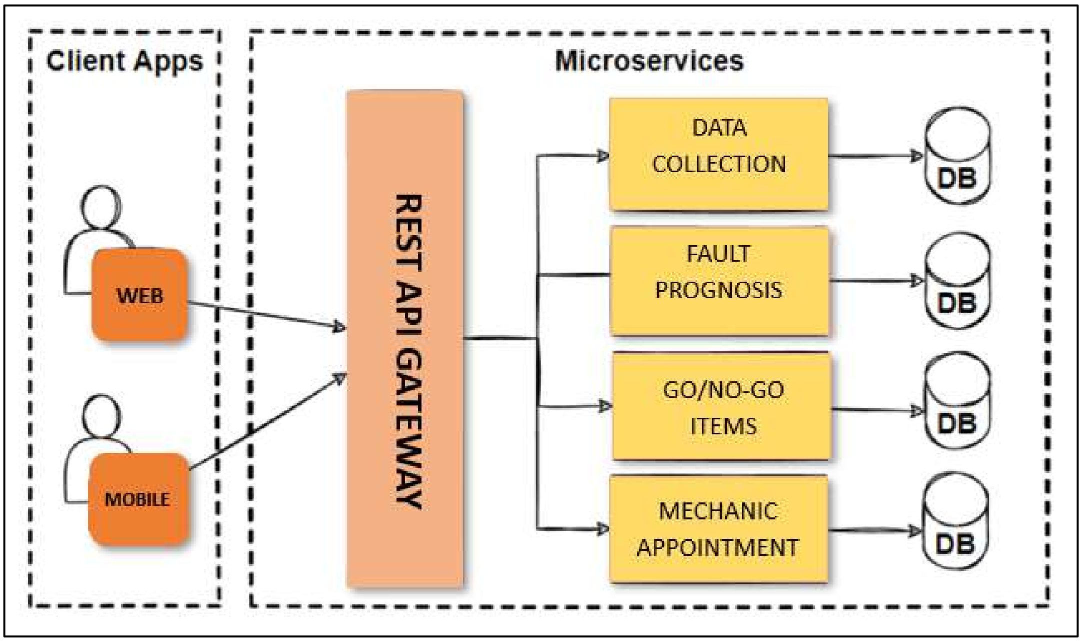
2.2.1 Architectural Design:

"Microservice Architecture Design"

- **Modularity and Scalability:** The Microservices architecture allows the system to be divided into smaller, independent services, each responsible for specific functionalities. This modularity enables easy scalability, as each service can be scaled independently based on its demand and usage.
- **Decoupled Components:** In the context of the self-diagnostic system, different functionalities like data collection, ML/DL algorithms, fault categorization, appointment scheduling, and user interfaces can be implemented as separate microservices. This decoupling ensures that changes or updates in one service do not impact others, enhancing maintainability.
- **Flexibility and Agility:** The project scope involves continuous iterations, feedback, and improvements. The Microservices architecture allows individual services to be developed, tested, and deployed independently, facilitating a more agile development process.
- **Technology Diversity:** Different services within the self-diagnostic system may require different technologies or programming languages. With the Microservices architecture, each service can be developed using the most suitable technology stack, promoting flexibility in technology choices.
- **Fault Tolerance and Resilience:** In a Microservices architecture, the failure of one service does not necessarily bring down the entire system. Services can be designed to handle failures gracefully, leading to a more robust and resilient overall system.

- **Easy Deployment and Updates:** Since each microservice is independent, deploying updates or changes to specific services is straightforward. This reduces downtime and makes it easier to roll back changes if needed.
- **Scalability for User Traffic:** As the system gains popularity and user traffic increases, specific microservices that require more resources, like appointment scheduling or fault prediction, can be scaled up to handle the load effectively.
- **Parallel Development:** The Microservices architecture allows multiple fyp development teams to work concurrently on different services, speeding up the overall development process.

Microservice Architecture Diagram:



2.2.2 Process Flow (Functional Requirement):

Data Collection and Integration:

- The system shall fetch real-time data from the Vehicle On-Board Diagnostics (OBD) through the Raspberry Pi microprocessor.
- The system shall integrate and store the OBD data for analysis and future reference.

ML/DL Algorithm Integration:

- The system shall incorporate Machine Learning (ML) and Deep Learning (DL) algorithms to analyze OBD data and predict potential vehicle malfunctions.
- The ML/DL models shall be trained and updated regularly with new data to improve prediction accuracy.

Fault Categorization:

- The system shall categorize predicted faults into two categories: "GO ITEMS" (can be delayed for maintenance) and "NO-GO ITEMS" (cannot be delayed for maintenance).
- Each category shall have a clear and distinct representation in the user interface.

Appointment Scheduling:

- For "NO-GO ITEMS," the system shall schedule appointments with approved vehicle vendors or mechanics to address critical faults promptly.
- The system shall send timely notifications to the vehicle owner or user about the scheduled appointments.

Real-Time Visualization:

- The system shall provide real-time visualizations of vehicle health status, including performance metrics and potential faults.
- Users shall be able to access visual representations of fault predictions through a user-friendly Desktop and Mobile Application.

User Authentication and Access Control:

- The system shall implement secure user authentication to ensure only authorized users can access vehicle data and diagnostic insights.
- Different user roles (e.g., vehicle owners, fleet managers) shall have appropriate access rights and permissions.

Web, Mobile, and Power Bi Applications:

- The system shall support three different user interfaces: Web Application, Mobile Application, and Power Bi Application.
- Each application shall provide a consistent and intuitive user experience.

User Feedback Mechanism:

- The system shall include a feedback mechanism that allows users to provide feedback on the diagnostic insights and user interface.
- User feedback shall be considered for continuous improvement and refinement of the system.

Data Privacy and Security:

- The system shall implement robust data privacy measures to ensure the confidentiality and integrity of the OBD data and user information.

- User data shall be encrypted, and access controls shall be enforced to prevent unauthorized access.

Automated Data Verification:

- The system shall verify the authenticity and validity of the OBD data received from the vehicle to ensure accurate predictions and prevent data tampering.

Chapter 3

3.1 Mathematical Equations used in Algorithms

3.1.1 Mathematical Equation

1. Euclidean Distance (KNN) = $\sum_{i=1}^{n}\sqrt{(X_i - Y_i)}$

Equation 1: Equation of Distance

2. P(Fault) = $\frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 \cdots \beta_n X_n)}}$

Equation 2: Fault Prediction Model (Logistic Regression)

3. Data = MicrocontrollerRead (OBD_Port)

Equation 3: Data Fetching through Microcontroller

4. Augmented Data = Original Dataset + Transformation

Equation 4: Dataset Augmentation and enlarge to avoid underfitting conditions.

Chapter 4

4.1 Proposed Solution/Results & Discussion

4.1.1 Web/Android Module:

"Web Module"

- **Frontend Components:** The frontend of the Web Module consists of various components representing the visual elements and functionalities of the application. These components will be designed using React.js, a popular JavaScript library known for its component-based architecture. Each component will handle specific aspects of the user interface, such as displaying real-time vehicle health status, showing predicted faults, and providing options for scheduling maintenance appointments.
- **User Authentication and Authorization:** The Web Module will include user authentication and authorization mechanisms to ensure secure access to vehicle data and diagnostic insights. Users will need to log in with their credentials, and the system will grant appropriate access based on user roles and permissions.
- **API Gateway:** To communicate with the backend microservices responsible for data processing and predictions, the Web Module will interact with an API Gateway. The API Gateway serves as an entry point for all incoming requests from the frontend and directs them to the appropriate microservices.
- **Real-Time Updates:** Using WebSocket or other real-time communication techniques, the Web Module will receive continuous updates on vehicle health and diagnostic insights, enabling users to monitor their vehicle's status in real time.

"Android Module"

- **User Interface (UI) Components:** The Android app's user interface will be built using native Android UI components. These components will be designed to offer a seamless and user-friendly experience on various Android devices, including smartphones and tablets.
- **Data Fetching and Caching:** The Android app will fetch data from the backend microservices through API calls. To ensure a smooth user experience, the app may employ data caching techniques to store and display certain information offline
- **Background Service:** The Android app can include a background service responsible for periodically fetching real-time data updates from the backend. This service will ensure that users receive timely information even when the app is running in the background.

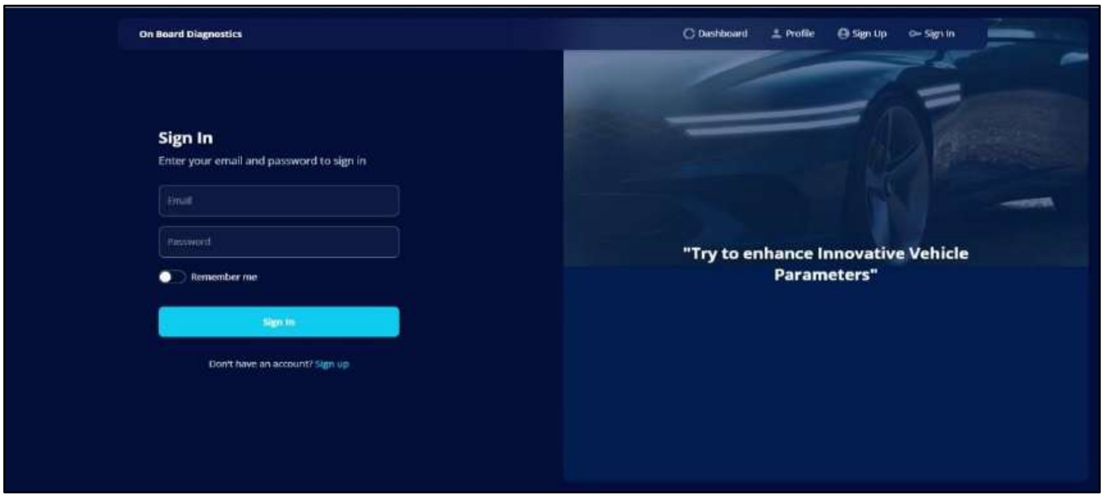
- **Local Storage:** The Android app may utilize local storage capabilities to store certain user preferences, login credentials, or other relevant data to provide a more personalized experience.
- **User Notifications:** The Android app can leverage the notification system to send users timely alerts and reminders about scheduled maintenance appointments or critical faults that require immediate attention.

“Overall System Integration”

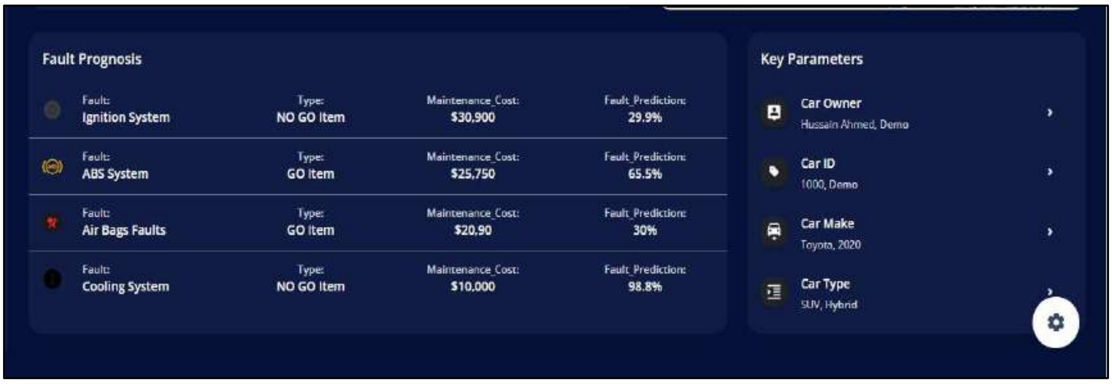
Both the Web Module and Android Module will interact with the backend microservices, which form the core of the "Automated Self-Diagnostic Mechanism" system. The backend microservices handle data processing, ML/DL predictions, fault categorization, and appointment scheduling. The communication between the frontend modules and the backend microservices will be facilitated by API calls through the API Gateway, ensuring a cohesive and efficient system. The combination of the Web Module and Android Module provides users with versatile access to the self-diagnostic system, allowing them to monitor their vehicle's health and make informed decisions on maintenance from both desktop and mobile environments. This architecture ensures a consistent user experience across different platforms while leveraging the strengths of web and mobile technologies.

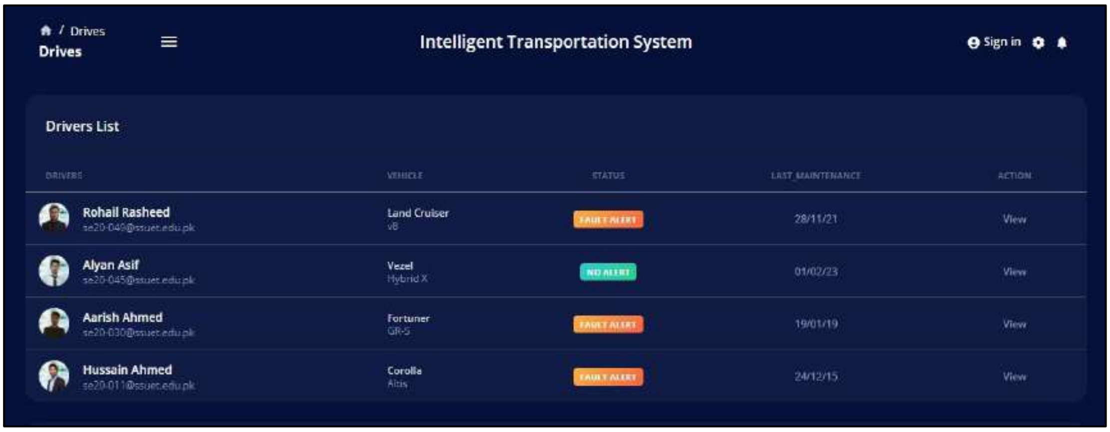
4.1.2 Functions (UX based methodology):

“Web Application UI”

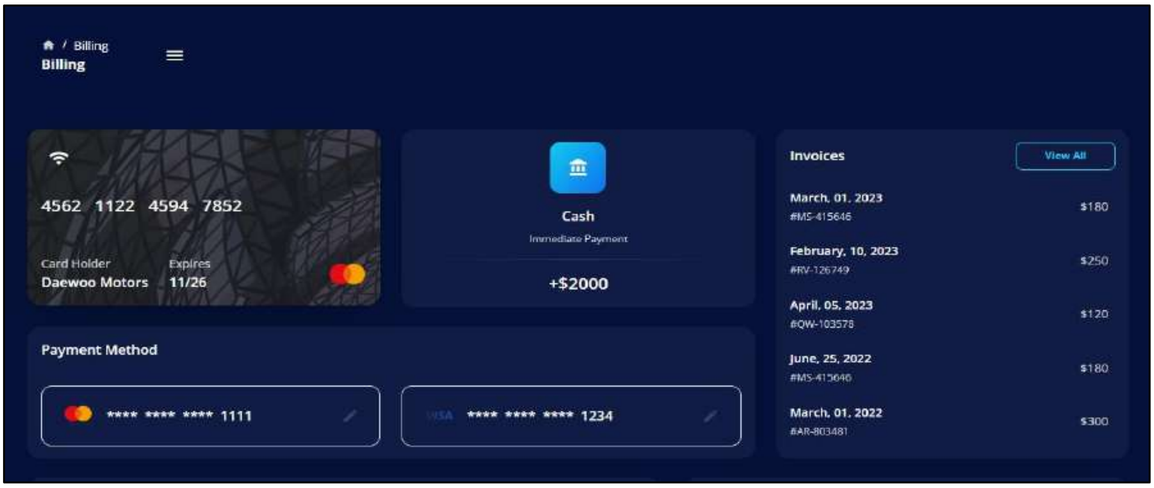


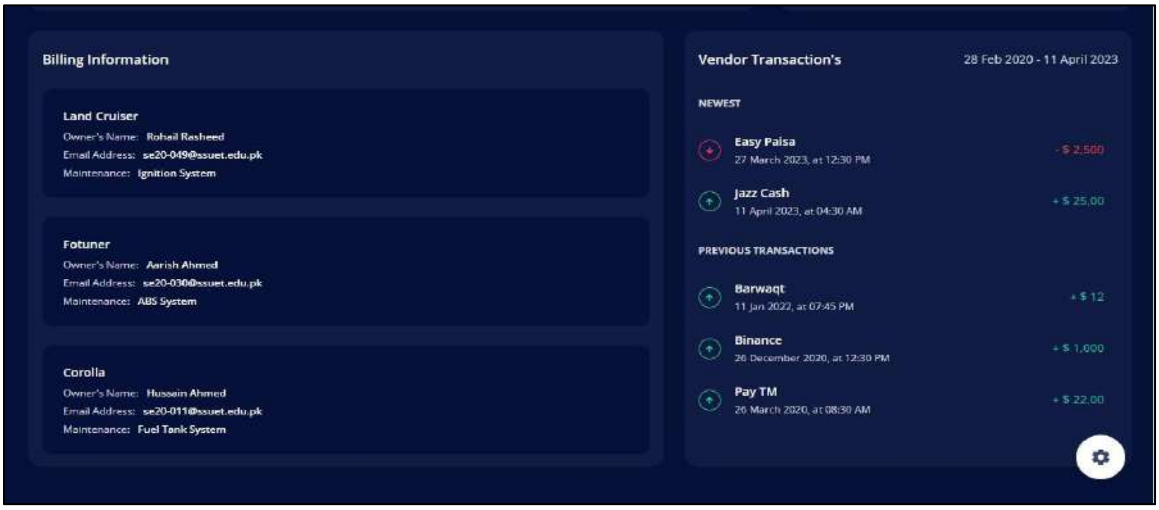

On Board Diagnostics
Dashboard
Drives
Billing
Virtual Reality
Sign In
Dashboard
Dashboard
Intelligent Transportation System
Sign In
AVERAGE SPEED
67.33 mph
+55% since yesterday
AVERAGE RPM
44.5 RPM
+3% since last week
AVG FUEL CONSUMPTION
26 litre
-2% since last quarter
AVG VOLT GENERATION
60 Volt
+5% than last month
Engine Performance
4% more in 2023
Fault in Ignition System
Trace ignition wiring while checking for grounds, shorts, and open circuits and resolve issue.


Fault Prognosis
Fault: Ignition System
Type: NO GO Item
Maintenance_Cost: $30,900
Fault_Prediction: 29.9%
Fault: ABS System
Type: GO Item
Maintenance_Cost: $25,750
Fault_Prediction: 65.5%
Fault: Air Bags Faults
Type: GO Item
Maintenance_Cost: $20.90
Fault_Prediction: 30%
Fault: Cooling System
Type: NO GO Item
Maintenance_Cost: $10,000
Fault_Prediction: 98.8%
Key Parameters
Car Owner
Hussain Ahmed, Demo
Car ID
1000, Demo
Car Make
Toyota, 2020
Car Type
SUV, Hybrid


Drives
Drives
Intelligent Transportation System
Sign In
Drivers List
Rohail Rasheed
Land Cruiser
28/11/21
View
Alyan Asif
Vezel
Hybrid X
NO ALERT
01/02/23
View
Aarish Ahmed
Fortuner
19/01/19
View
Hussain Ahmed
Corolla
24/12/15
View

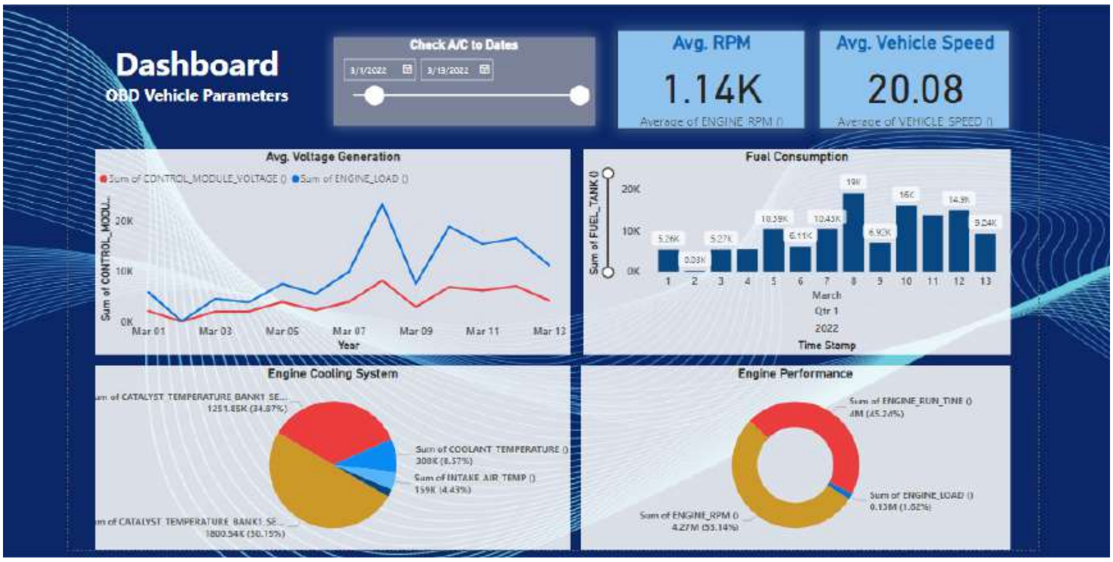

Approved Vendors List
Daewoo Motors
$2,500
Available
60%
Pak Wheels
$5,000
Available
100%
Jan Japan Motors
$3,400
Not Available
30%
Moin Motors Workshop
$1,400
Not Available
0%
SAP Car Detailing
$14,000
Not Available
80%
Toyota Workshop
$2,300
Not Available
100%


Billing
Billing
4562 1122 4594 7852
Card Holder
Daewoo Motors
Expires
11/26
Cash
Immediate Payment
+$2000
Invoices
View All
March, 01, 2023
#MS-415646
$180
February, 10, 2023
#RV-126749
$250
April, 05, 2023
#QW-103578
$120
June, 25, 2022
#MS-415646
$180
March, 01, 2022
#AR-803481
$300
Payment Method
**** **** **** 1111
**** **** **** 1234

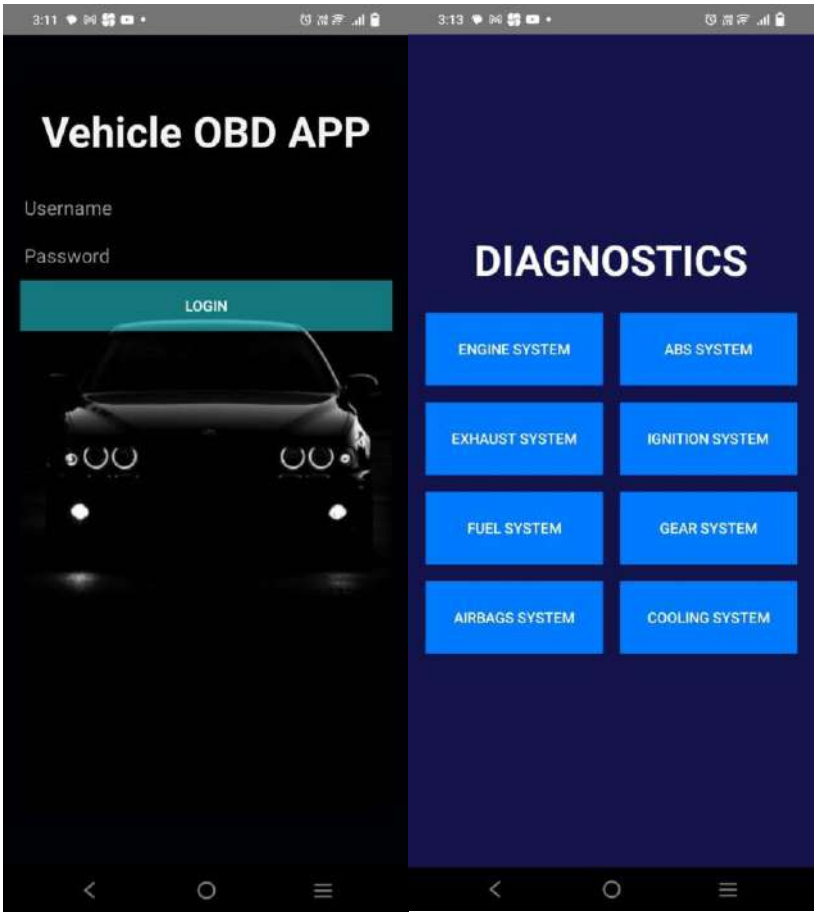

Billing Information
Land Cruiser
Owner's Name: Rohail Rasheed
Email Address: se20-049@ssuet.edu.pk
Maintenance: Ignition System
Fotuner
Owner's Name: Aarish Ahmed
Email Address: se20-030@ssuet.edu.pk
Maintenance: ABS System
Corolla
Owner's Name: Hussain Ahmed
Email Address: se20-011@ssuet.edu.pk
Maintenance: Fuel Tank System
Vendor Transaction's
28 Feb 2020 - 11 April 2023
NEWEST
Easy Paisa
27 March 2023, at 12:30 PM
- $ 2,500
Jazz Cash
11 April 2023, at 04:30 AM
+ $ 25,00
PREVIOUS TRANSACTIONS
Barwaqt
11 Jan 2022, at 07:45 PM
+ $ 12
Binance
26 December 2020, at 12:30 PM
+ $ 1,000
Pay TM
26 March 2020, at 08:30 AM
+ $ 22.00


Virtual Reality
Virtual Reality
Intelligent Transportation System
Sign In
Hussain Shams
COROLLA
Approved Vendors
3
Predicted Cost

"Power Bi Dashboard"



"Mobile App UI" (in progress)



4.1.3 Database module (Justification), Type, DDL (Data Definition Table):

- **Justification for Google Firebase Database Module:**

The Google Firebase database module is an ideal choice for the "Automated Self-Diagnostic Mechanism" project due to its real-time synchronization capabilities and seamless integration with our Web Module and Android Module. With a focus on predicting vehicle malfunctions and offering maintenance insights, Firebase's real-time updates enable users to receive instant diagnostic information, enhancing their ability to make timely and informed decisions. Its scalability and cloud-based nature ensure efficient handling of a growing user base and large volumes of vehicle data, while the serverless architecture eliminates the need for server management, allowing our team to concentrate on developing essential functionalities. Moreover, Firebase's robust security features and user authentication mechanisms guarantee the protection of sensitive vehicle data, ensuring a secure and reliable self-diagnostic system.

- **Type of Google Firebase Database Module:**

The Google Firebase database module falls under the category of NoSQL databases, specifically a document-oriented database. It stores data in JSON-like documents, making it flexible and adaptable to the dynamic structure of vehicle data in the "Automated Self-Diagnostic Mechanism" project. As vehicle data may vary in structure depending on the make and model, Firebase's document-oriented approach allows us to store and access diverse data types efficiently. Furthermore, Firebase's NoSQL nature enables easy horizontal scalability, making it suitable for handling the high volume of data expected in our application.

- **Data Definition (DDL) Table for Google Firebase Database Module:**

Google Firebase is a type of database known as NoSQL, which means it doesn't follow the traditional way of creating tables with a Data Definition

Language (DDL) like you would in a typical relational database. Instead, Firebase organizes data into "collections," which are similar to tables, and "documents," which are like rows in a table. It uses a JSON-like structure to store information.

So, in our project, we'll have a **"VehicleData"** collection that holds individual documents for each vehicle. These documents will have different fields like **"VehicleID", "OBDData," "DiagnosticInsights", "MaintenanceSchedule"** and **"AppointmentStatus".** The best part is that Firebase allows us to be flexible with our data and store vehicle information with varying attributes without being bound by a fixed table structure. This is perfect for our dynamic and ever-evolving "Automated Self-Diagnostic Mechanism" project, where vehicle data and diagnostic insights might change over time. With Firebase, we can easily manage and access the data efficiently, adapting to the needs of the project as it progresses.

4.1.4 Administration Module:

- **User Management:** The administration module allows administrators to manage user accounts, including creating new accounts, granting appropriate access rights, and revoking access if necessary. Administrators can assign different roles to users, such as vehicle owners, mechanics, or technicians, each with specific permissions for data access and system functionalities.
- **System Settings:** Administrators can configure system settings and preferences through the administration module. This includes defining parameters for fault categorization, maintenance scheduling rules, notification preferences, and other system-wide configurations.
- **Data Monitoring and Reporting:** The administration module provides tools for administrators to monitor the overall health of the self-diagnostic system. This includes real-time data analytics, system performance monitoring, and generating reports on fault prediction accuracy and maintenance trends.
- **Security and Authentication Management:** As administrators hold significant control over the system, the administration module includes functionalities to manage security measures, such as user authentication mechanisms, encryption settings, and access controls.
- **Data Integrity and Backup:** Administrators can ensure the integrity of the data stored in the Google Firebase database by implementing data

backup and recovery mechanisms. Regular data backups are crucial to protect against data loss or system failures.

- **System Maintenance and Updates:** The administration module allows administrators to manage system maintenance tasks and updates. This includes deploying software updates, managing server resources, and ensuring the system operates smoothly and efficiently.
- **Troubleshooting and Support:** The administration module may offer diagnostic tools and support features to help administrators troubleshoot issues within the self-diagnostic system. This can include detailed logs, error tracking, and user support functionalities.

User Interface and Accessibility:

The administration module's user interface is designed with an intuitive and user-friendly dashboard, allowing administrators to access and manage various functionalities efficiently. It will be accessible via secure login, ensuring that only authorized personnel can access and utilize the administration tools. The user interface will also provide relevant data visualizations and reports to help administrators make informed decisions regarding system performance and maintenance.

4.2 Data Dictionary:

Data Dictionary for Authentication

	DATABASE ELEMENT	DATA TYPE	DATA ELEMENT REQ
1	User_name	String	Yes
2	User_email	String	Yes
3	password	String	Yes

Data Dictionary for Application

	DATABASE ELEMENT	DATA TYPE	DATA ELEMENT REQ
1	VehicleData	String	Yes
2	DriverData	String	Yes
3	VendorData	String	Yes
4	OBD_Data	Numeric	Yes
5	DiagnosticInsights	String	Yes
6	AppointmentStatus	String	Yes
7	MaintenanceSchedule	String	Yes
8	VendorBilling	Integer	Yes

9	Fault_Prognosis	String	Yes

Vehicle Data Collection:

- **VehicleID:** Unique identifier for each vehicle in the system.
- **OBDData:** JSON object containing data fetched from the Vehicle On-Board Diagnostics (OBD), including engine parameters, sensor readings, and error codes.
- **DiagnosticInsights:** JSON object containing diagnostic insights generated by ML/DL algorithms, categorizing potential faults as "GO ITEMS" or "NO-GO ITEMS."
- **MaintenanceSchedule:** JSON object containing information about scheduled maintenance appointments, including date, time, and service details.
- **AppointmentStatus:** Status of the maintenance appointment, such as "Pending," "Scheduled," "Completed," or "Cancelled."

User Data Collection:

- **UserID:** Unique identifier for each user in the system.
- **FirstName:** First name of the user.
- **LastName:** Last name of the user.
- **Email:** Email address of the user, used for authentication.
- **Password:** Encrypted password associated with the user account.
- **UserRole:** Role assigned to the user (e.g., "Vehicle Owner," "Fleet Manager," "Mechanic").

Fault Category Collection:

- **CategoryID:** Unique identifier for each fault category.
- **CategoryName:** Name of the fault category, such as "Engine Fault," "Brake System Fault," etc.

Notification Data Collection:

- **NotificationID:** Unique identifier for each notification.
- **UserID:** ID of the user to whom the notification is sent.
- **Message:** Content of the notification message.
- **Timestamp:** Timestamp of when the notification was generated.

Chapter 5

5.1 Summary and Future work

1. **What is the thesis about?**

The thesis revolves around the development and implementation of the "Automated Self-Diagnostic Mechanism" in the context of vehicle maintenance. It tackles the escalating risks and challenges associated with the increasing number of vehicles on the roads, proposing a proactive solution inspired by the aircraft industry's predictive maintenance practices. The core focus is on leveraging Machine Learning (ML) and Deep Learning (DL) Algorithms to anticipate and categorize potential vehicle failures, thereby enhancing road safety and optimizing maintenance processes.

2. **What is the purpose of the project/thesis?**

The overarching purpose of the project/thesis is to redefine conventional approaches to vehicle maintenance. It seeks to introduce a transformative system capable of predicting and categorizing potential vehicle failures, distinguishing between issues that can be delayed ("GO ITEMS") and those requiring immediate attention ("NO-GO ITEMS"). The ultimate goal is to enhance road safety, minimize unexpected breakdowns, and streamline the maintenance process for a more secure and efficient driving experience.

3. **What were the methods used to research the information?**

The project employs a multifaceted approach, starting with the collection of real-time data from Vehicle On-Board Diagnostics (OBD) through the Raspberry Pi microprocessor. This data is then dispatched to advanced ML and DL Algorithms for Training and Testing purposes. The system's capabilities are honed through continuous monitoring and feedback loops, incorporating user interactions and insights. The methods also include the integration of cloud services for record-keeping and seamless user experiences through Desktop and Mobile Applications.

4. **What are the results, conclusions, and recommendations that the thesis presents?**

The results of the thesis manifest in the development of a robust "Automated Self-Diagnostic Mechanism" capable of predicting and categorizing potential vehicle failures. The ML and DL Algorithms exhibit promising accuracy in differentiating between "GO ITEMS" and "NO-GO ITEMS," providing a foundation for proactive vehicle maintenance. Conclusions drawn highlight the transformative potential of the system in enhancing road safety and optimizing maintenance practices.

The project advocates for a sustained commitment to continuous monitoring and improvement, specifically refining the "Automated Self-Diagnostic Mechanism" for greater precision. The focus remains on integrating user feedback for ongoing enhancements. Future work involves fine-tuning Machine Learning (ML) and Deep Learning (DL) Algorithms to ensure more precise predictions, advancing anticipatory vehicle maintenance. The next phase targets expanding data coverage to a broader array of OBD parameters, providing nuanced insights. Additionally, integrating Vehicle-to-Vehicle (V2V) communication for transmitting prognostic data fosters a networked ecosystem, enhancing collective awareness for safer driving. Collaborations with vendors are pivotal for efficient issue resolution, while optimizing user interfaces ensures broader accessibility, promoting widespread adoption in the automotive technology landscape.

Chapter 6

7.1 Conclusion & Recommendation

In conclusion, this research embarked on a transformative journey to address the pressing issue of vehicular malfunctions and enhance predictive maintenance through the "Automated Self-Diagnostic Mechanism." The research question focused on how to proactively anticipate and categorize potential vehicle failures, reshaping traditional approaches to automotive safety and maintenance. The methodology involved data collection from Vehicle On-Board Diagnostics (OBD) through the Raspberry Pi microprocessor, employing advanced Machine Learning (ML) and Deep Learning (DL) Algorithms. These algorithms, after rigorous training and testing, demonstrated a remarkable ability to predict and categorize faults into "GO ITEMS" and "NO-GO ITEMS," establishing a foundation for a more secure driving experience.

One of the significant accomplishments lies in the project's holistic approach, integrating cloud services for seamless record-keeping and providing users with immersive experiences through user-friendly Desktop and Mobile Applications. The system not only predicts faults but also schedules timely maintenance appointments with approved vendors, ensuring a swift response to critical issues.

In a broader context, the importance of this research transcends academic boundaries. By fostering collaborations with vendors and optimizing user interfaces for accessibility, the "Automated Self-Diagnostic Mechanism" envisions a real-world impact, streamlining vehicle maintenance processes and contributing to overall road safety. As we conclude, the research invites contemplation on the potential of this system to transform the automotive landscape and, in doing so, prompts a broader question about the future of predictive maintenance: How can anticipatory diagnostics redefine our relationship with vehicular safety and reliability? This question, left for contemplation, underscores the significance of ongoing research and development in this dynamic and impactful field.

References

[1] http://www.obdii.com

[2] Fang, Y., Min, H., Wang, W., Xu, Z., & Zhao, X. (2020). A Fault Detection and Diagnosis System for Autonomous Vehicles Based on Hybrid Approaches. IEEE Sensors Journal, 1– 1. doi:10.1109/jsen.2020.2987841

[3] Z. Gao, C. Cecati and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part II: Fault Diagnosis With Knowledge-Based and Hybrid/Active Approaches," in IEEE Transactions on Industrial Electronics, vol. 62, no. 6, pp. 3768-3774, June 2015.

[4] S. Yin, S. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," IEEE Trans. Ind. Electron., vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[5] N. Meskin and K. Khorasani, Fault detection and isolation: multi-vehicle unmanned systems. New York, USA: Springer, 2011.

[6] J V Moniaga et al 2018 J. Phys, "Diagnostics vehicle's condition using OBD-ii and raspberry pi technology: study literature". 2nd International Conference on Computing and Applied Informatics 2017. IOP Conf. Series: Journal of Physics: Conf. Series 978 (2018) 012011

[7] vneeth, S., Prithvil, K. P., Sri Hari, N. R., Thushar, R., & Rajeswari, M. (2020). "OnBoard Diagnostics and Driver Profiling". 2020 5th International Conference on Computing, Communication and Security (ICCCS).doi:10.1109/icccs49678.2020.9277449

[8] Rimpas, Dimitrios, Andreas Papadakis, and Maria Samarakou. 2019. "OBD-II Sensor Diagnostics for Monitoring Vehicle Operation and Consumption." in Energy Reports. Elsevier Ltd.

[9] Shafi, Uferah, Asad Safi, Ahmad Raza Shahid, Sheikh Ziauddin, and Muhammad Qaiser Saleem. 2018. "Vehicle Remote Health Monitoring and Prognostic Maintenance System." Journal of Advanced Transportation 2018. doi: 10.1155/2018/8061514.

[10] Sim, Alex Xandra Albert, and Benhard Sitohang. 2014. "OBD-II Standard Car Engine Diagnostic Software Development." 2014 International Conference on Data and Software Engineering, ICODSE 2014. doi: 10.1109/ICODSE.2014.7062704.

[11] Vasavi, S., K. Aswarth, T. Sai Durga Pavan, and A. Anu Gokhale. 2021. "Predictive Analytics as a Service for Vehicle Health Monitoring Using Edge Computing and AK-NN Algorithm." Materials Today: Proceedings 46(xxxx):8645–54. doi: 10.1016/j.matpr.2021.03.658.

[12] M. M. Oluwaseyi, A. M. Sunday. 2020. "Specifications and Analysis of Digitized Diagnostics of Automobiles: A Case Study of on Board Diagnostic (OBD II)" International Journal of Engineering Research and Technology (IJERT) Volume 09, Issue 01 (January 2020) (2278-0181). doi: 10.17577/IJERTV9IS010045

[13] B. Bánhelyi, T. Szabó. 2020. "Data Mining and Analysis for Data From Vehicles Based on the OBDII Standard" 11th International Conference on Applied Informatics Eger, Hungary, January 29–31, 2020, published at http://ceur-ws.org

[14] Cai, X., Lei, C., Peng, B., Tang, X., & Gao, Z. (2020). “Road Traffic Safety Risk Estimation Method Based on Vehicle Onboard Diagnostic Data” Journal of Advanced Transportation, 2020, 1–13. doi:10.1155/2020/3024101

[15] Geivanidis et al. 2020. “Next generation of vehicle diagnostics based on advanced onboard monitoring and cloud-based diagnostics” 8th Transport Research Arena TRA 2020, April 27-30, 2020, Helsinki, Finland

[16] Zheng, X., Lu, S., Yang, L., Yan, M., Xu, G., Wu, X., … Wu, Y. (2020). “Real-world fuel consumption of light-duty passenger vehicles using on-board diagnostic cimplementation of hybrid artificial intelligence model[J]. Eng Comput 36(1):1–11.

[17] Ferrari R, Mancini-Terracciano C, Voena C et al (2019) MR-based artificial intelligence model to assess response to therapy in locally advanced rectal cancer[J]. Eur J Radiol 118:1–9

[18] Fu T, Wang C (2018) A hybrid wind speed forecasting method and wind energy resource analysis based on a swarm intelligence optimization algorithm and an artificial intelligence model[J]. Sustainability 10(11):3913

[19] Sun X, Young J, Liu JH et al (2018) Prediction of pork loin quality using online computer vision system and artificial intelligence model [J]. Meat Sci 140:72–77

[20] Kin, S.-C. Chen, Y.-T. Shih, and S.-H. Chen, “A study on remote on-line diagnostic system for vehicles by integrating the technology of OBBD, GPS, and 3G,” World Academy of Science, Engineering and Technology, vol. 56, 2009.

[21] B. Lee, “OBD-II(exhaust),” Kyung Young sa, 2005

[22] On-Board Diagnostics, http://en.wikipedia.org/wiki/OBD-II#OBD-II.

[23] A. I. Santini, OBD-II: Function, Monitors and Diagnostic Techniques, Cengage Learning, 2010

[24] SO 14230-4, Road Vehicles-Diagnostic Systems, https://law.resource.org/pub/us/cfr/ibr/004/iso.14230-4.2000.pdf

[25] Jaehyu Bae, Design and Development of Automobile Fault Diagnosis System Using AI Machine Learning, in: Next-generation convergence information service technology society, Vol. 2, 2018

[26] Tafreshi R., Ahmadi H., Sassani F., et al. Informative wavelet algorithm in diesel engine diagnosis Proceedings of the IEEE internatinal symposium on intelligent Control, IEEE (2002), pp. 361-366

[27] Wu Jian-da, Liu Chiu-Hong Investigation of engine fault diagnosis using discrete wavelet transform and neural network Expert Syst Appl, 35 (3) (2008), pp. 1200-1213

[28] Wu Jian-da, Liu Chiu-Hong An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network Expert Syst Appl, 36 (3) (2009), pp. 4278-4286

[29] Prokhorov D.V. Toyota prius HEV neurocontrol and diagnostics Neural Netw, 21 (2) (2008), pp. 458-465

[30] Shafi U, Safi A, Shahid AR, Ziauddin S, Saleem MQ (2018) Vehicle remote health monitoring and prognostic maintenance system. J Adv Trans 201:1–10

[31] Gupta N, Khosravy M, Gupta S, Dey N, Crespo RG (2020) Lightweight artificial intelligence technology for health diagnosis of agriculture vehicles: parallel evolving artificial neural networks by genetic algorithm. Int J Parallel Program 1–26

[32] Liu et al., (2022) "Fault mode detection of a hybrid electric vehicle by using support vector machine" 2022 The 3rd International Conference on Power Engineering (ICPE 2022), Science and Engineering Institute, China
[33] Chapter, December 09–11, Sanya, Hainan Province, China https://doi.org/10.1016/j.egyr.2023.04.328
[34] J. Buddhakulsomsiri and A. Zakarian, "Sequential pattern mining algorithm for automotive warranty data," Computers & Industrial Engineering, vol. 57, no. 1, pp. 137–147, 2009.
[35] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, "Data mining in manufacturing: a review based on the kind of knowledge," Journal of Intelligent Manufacturing, vol. 20, no. 5, pp. 501–521, 2009.
[36] S. Byttner, T. Rögnvaldsson, and M. Svensson, "Consensus self-organized models for fault detection (COSMO)," Engineering Applications of Artificial Intelligence, vol. 24, no. 5, pp. 833–839, 2011.
[37] R. Prytz, S. Nowaczyk, and S. Byttner, "Towards relation discovery for diagnostics," in Proceedings of the 1st International Workshop on Data Mining for Service and Maintenance, KDD4Service 2011—Held in Conjunction with SIGKDD'11, pp. 23–27, USA, August 2011
[38] A. Alzghoul, M. Löfstrand, and B. Backe, "Data stream forecasting for system fault prediction," Computers & Industrial Engineering, vol. 62, no. 4, pp. 972–978, 2012.
[39] K. Napierala and J. Stefanowski, "BRACID: a comprehensive approach to learning rules from imbalanced data," Journal of Intelligent Information Systems, vol. 39, no. 2, pp. 335– 373, 2012.
[40] R. Prytz, S. Nowaczyk, T. Rögnvaldsson, and S. Byttner, "Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data," Engineering Applications of Artificial Intelligence, vol. 41, pp. 139–150, 2015.
[41] Rajakumar, M. P., Ramya, J., & Maheswari, B. U. (2021). Health monitoring and fault prediction using a lightweight deep convolutional neural network optimized by Levy flight optimization algorithm. Neural Computing and Applications. doi:10.1007/s00521-021- 05892-0
[42] Khorsravinia, K., Hassan, M. K., Rahman, R. Z. A., & Al-Haddad, S. A. R. (2017). Integrated OBD-II and mobile application for electric vehicle (EV) monitoring system. 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS). doi:10.1109/i2cacis.2017.8239058
[43] Bánhelyi, B. and Szabó, T. (2020) 'Data mining and analysis for data from vehicles based on the OBDII standard', CEUR Workshop Proceedings, 2650, pp. 30–37.
[44] Khorsravinia, K. et al. (2017) 'Integrated OBD-II and mobile application for electric vehicle (EV) monitoring system', Proceedings - 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems, I2CACIS 2017, 2017-Decem(October), pp. 202–206. doi: 10.1109/I2CACIS.2017.8239058.
[45] Kowalik, B. (2018) 'Introduction to car failure detection system based on diagnostic interface', 2018 International Interdisciplinary PhD Workshop, IIPhDW 2018. IEEE, pp. 4–7. doi: 10.1109/IIPHDW.2018.8388233.

[46] Liu, H., Song, X., & Zhang, F. (2021). Fault diagnosis of new energy vehicles based on improved machine learning. Soft Computing, 25(18), 12091–12106. doi:10.1007/s00500- 021-05860-9.

Annexure

Annexure (if any) should be placed at the end of the project report.

Annexure 1: (the table shows the study of previous researches for fault prediction and prognosis).

Annexure I

Table: the table shows the study of previous researches for fault prediction and prognosis

	Ignition System	Fuel System	Exhaust Recirculation System	Engine Cooling System	Engine Oil Service Life (Temperature)	Engine Oil Service Life (Temperature)	Engine Load	Engine RPM	Vehicle Speed
D. Rimpas et al. (2019)		✓	✓	✓					
Shafi et al. (2018)	✓	✓	✓	✓					
Vasavi et al. (2021)					✓	✓			
Wei et al. (2020)					✓	✓			
S. Geivanidis et al. (2020)			✓						
S. Navneeth et al. (2020)	✓	✓							
M. Oluwaseyi et al. (2020)	✓	✓	✓	✓	✓	✓		✓	✓
B. Bánhelyi et al. (2020)		✓	✓				✓	✓	✓
X. Zheng et al. (2020)		✓	✓					✓	✓
A. Kulakov et al. (2020)		✓							
L. Wang et al. (2021)		✓	✓						
B. Suwandi et al. (2019)			✓						