

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS SEG2202 SOFTWARE ENGINEERING**

**ACADEMIC SESSION: Feb 2025**

**COURSEWORK PROJECT DUE DATE: 21 Mar 2025**

**STUDENT NAME: Goh Shu Yi**

**STUDENT ID: 22057905**

**PROGRAMME: Bachelor of Software Engineering (BSE)**

**YEAR / SEMESTER: Year 2 Semester 2**

**PROJECT TITLE:** Smart Vehicle Performance System (SVPS)

**INSTRUCTIONS TO CANDIDATES**

* This is a project which consists of **individual** and **group** effort.
* The total mark for this project is **100%**.
* The project mark will contribute 30% of the Coursework component.

shuyi

**21/3/2025**

**Goh Shu Yi**

**IMPORTANT NOTICE**

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to all late submission of work. Project submitted after the deadline will be regarded as a non-submission and marked zero.

**Academic Honesty Acknowledgement**

“I ......................................... (student name) verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer undergraduate programme student handbook)* for any kind of copying or collaboration on any assignment.”

… .........................................................(Student’s signature / Date)

# Part A: Project Planning

## 1.0 INTRODUCTION

With the rise of road accident rates in recent years, it has led to the emergence of road safety as a growing issue. Statistics claim that Malaysia reported a significant growth in road accidents, increasing from 477,200 cases in 2010 to 545,590 cases in 2022, which is about a 14% increase from 2010 to 2022 (Siddharta, 2022).

In 2022, the Department of Statistics Malaysia (DOSM) highlighted that one of the leading causes of death in Malaysia is road accidents. With a population exceeding 32 million and a vast network of roads, regulating road safety in Malaysia has become a significant challenge (Azami et al., 2024).

The World Health Organization (WHO) further proves that road traffic accidents have become a global issue, with approximately 1.3 million people dying annually due to road accidents and human error accounting for 90% of these incidents (World Health Organization, 2023).

Besides, the environmental impact of transportation has become an increasing concern, as transportation and logistics contributes to 24% of global CO2 emissions. As this continues, the European Environment Agency predicts that by 2025, global logistics will cause 40% of worldwide carbon emissions unless effective action is taken (Carbon Care Asia, 2023). This rising trend is severely harming our ecosystem and action must be taken in response to it, especially given that there is potential improvement in fossil fuel-powered vehicles which rely on burning fuel that produce greenhouse gasses and other pollutants.

To address these challenges and enhance road safety, the Smart Vehicle Performance System (SVPS) was designed to improve fuel efficiency and significantly reduce carbon footprints. Not only does it improve fuel usage, it also provides route optimization to minimize time wastage and prevent unnecessary fuel consumption caused by traffic congestion or inefficient navigation.

The SVPS is a system that will continuously monitor key performance indicators to analyze driving patterns and suggest optimal routes or actions to maximize fuel efficiency. In this system, real-time data will be collected from the vehicle sensors. These data include information of the vehicle, such as tire pressure, engine performance, fuel consumption, and mileage. By monitoring the vehicle’s metrics, the system can predict maintenance needs like oil changes, tire replacement, brake inspections, and trigger an alert to notify the user of the needs of the vehicle.

Other than that, the system can also identify the user’s various driving behaviors like average speed, braking and accelerating patterns, distracted driving like using a mobile phone or interacting with the car screen interface while driving, and concerning behaviors like rapid lane changes or sharp turns.

Another feature the system has is the in-built navigation system on the screen interface of the car which can optimize routes for the user to save fuel. Each time real-time data from the vehicle is collected, they will be uploaded to the backend of the system, which is a cloud-based platform that uses advanced algorithms for data analysis. Through this analysis, the system generates insights regarding the user’s driving behavior, such as speeding dangerously in a low speed area, and will display alerts or personalized recommendations and warnings on the app to advise the user to slow down or take other actions. This helps the user to have a safer and more efficient driving experience.

Not only that, each user can connect more than one vehicle to their account, which allows them to access all their vehicle insights in one spot with ease.

### 

### 1.1 Relevant Work

Some similar works or systems that are being used commercially now are the On-Board Diagnostics II (OBD-II) and GPS tracking systems, which are widely used for monitoring vehicle conditions and route mapping.

1. OBD-II System
   1. A vehicle diagnostics system that monitors the vehicle’s condition and allows car technicians to access internal diagnostic information to improve the vehicle maintenance experience.
   2. Tracks the vehicle’s wear and tear trend to enable the user to identify parts of the vehicle that tend to wear out fast and detect potential issues early on. This proactive diagnosis approach helps to prevent vehicle problems before they occur, rather than deal with it only after it has occurred.
   3. Measure driving patterns, including speed, idling time and other factors to support a more comprehensive understanding of both the driver’s behavior and vehicle conditions (Team, 2023).
2. GPS Tracking System
   1. Modern GPS tracking systems utilize satellite-based technology to access the driver’s real time location data to provide accurate real-time tracking, route optimization and performance monitoring.
   2. By adjusting the routes based on the driver's location, the system offers a more efficient driving experience which helps to reduce fuel consumption.
   3. For example, modern systems for delivery businesses are capable of monitoring driving actions, such as braking, acceleration and speed, which allows the businesses to identify problem areas and train their employees to drive safer and in a more fuel efficient manner (Woods, 2023).

The SVPS integrates both OBD-II and GPS tracking technologies into one holistic system. This integrated system improves vehicle health monitoring and driver behavior analysis, encouraging safe driving habits and ensuring better vehicle safety for everyday consumers.

## 

## 2.0 PROJECT OBJECTIVES AND PROJECT SCOPE

### 2.1 Project Objectives

The SVPS aims to enhance driving efficiency, safety, and environmental sustainability by leveraging real-time data analytics and predictive insights.

The key project objectives include:

1. Fuel Efficiency and Lower Carbon Emissions
   1. Recommend efficient routes regarding driving patterns to minimize fuel usage and therefore reduce environmental impact and promote environmental sustainability.
2. Real-time notifications and alerts
   1. Display and send real-time notifications and alerts about vehicle maintenance needs, potential issues and unsafe driving behaviors, such as warning users of low fuel tank or low tire pressure, to help users stay informed and act promptly to address the maintenance requirements to ensure a safe journey.
3. Increase Vehicle Efficiency
   1. Monitor and analyze driving patterns to improve fuel efficiency.
   2. Analyze wear and tear patterns to identify problematic driving behaviors like braking abruptly, accelerating aggressively or taking sharp turns that contribute to poor vehicle health.
4. Route Optimization
   1. Suggest the most efficient routes to save time, fuel and enhance driving experience.
5. Predictive Maintenance
   1. Utilize algorithms with data from the vehicle sensors to predict maintenance needs, such as oil changes, tire replacements and brake inspections.
   2. Internal diagnostic system to allow technicians to diagnose vehicle problems more easily.
6. Enhance User Experience and Safety
   1. Provide a mobile application with intuitive and user-friendly design to display maintenance needs, route suggestions and other features.

### 2.2 Project Scope

1. Mobile and In-car Application Development
   1. Develop a mobile application and in-car display interface with features such as real-time vehicle monitoring, driving behavior analysis and insights, route suggestions and maintenance alerts.
   2. Implement vehicle sensors and cloud-based analytics to collect and analyze vehicle performance data in real time including fuel consumption, engine health, driving patterns and other aspects.
2. Cloud-Based Analytics and Data Processing
   1. Create a backend infrastructure for storing and analyzing sensor data.
   2. Predict maintenance necessities and analyze driving behavior using advanced algorithms to enhance vehicle efficiency.
3. Notifications and Alert System
   1. Implement a real-time notifications and alert system that is triggered when potential issues, such as vehicle maintenance alerts, critical vehicle issues, fuel warnings and unsafe driving behaviors occur.
4. Route Optimization
   1. Integrate navigation systems to suggest the most fuel-efficient routes by accessing real-time traffic data and driving patterns to avoid congested areas to minimize travel time.
5. Multi Vehicle Management
   1. Enable users to connect and manage multiple vehicles from a single account, allowing them to track each vehicle’s performance and health all on one platform.
6. Driving Recommendations
   1. Offer comprehensive insights into driving behavior to encourage safer and more efficient driving habits.

#### 2.2.1 Resources

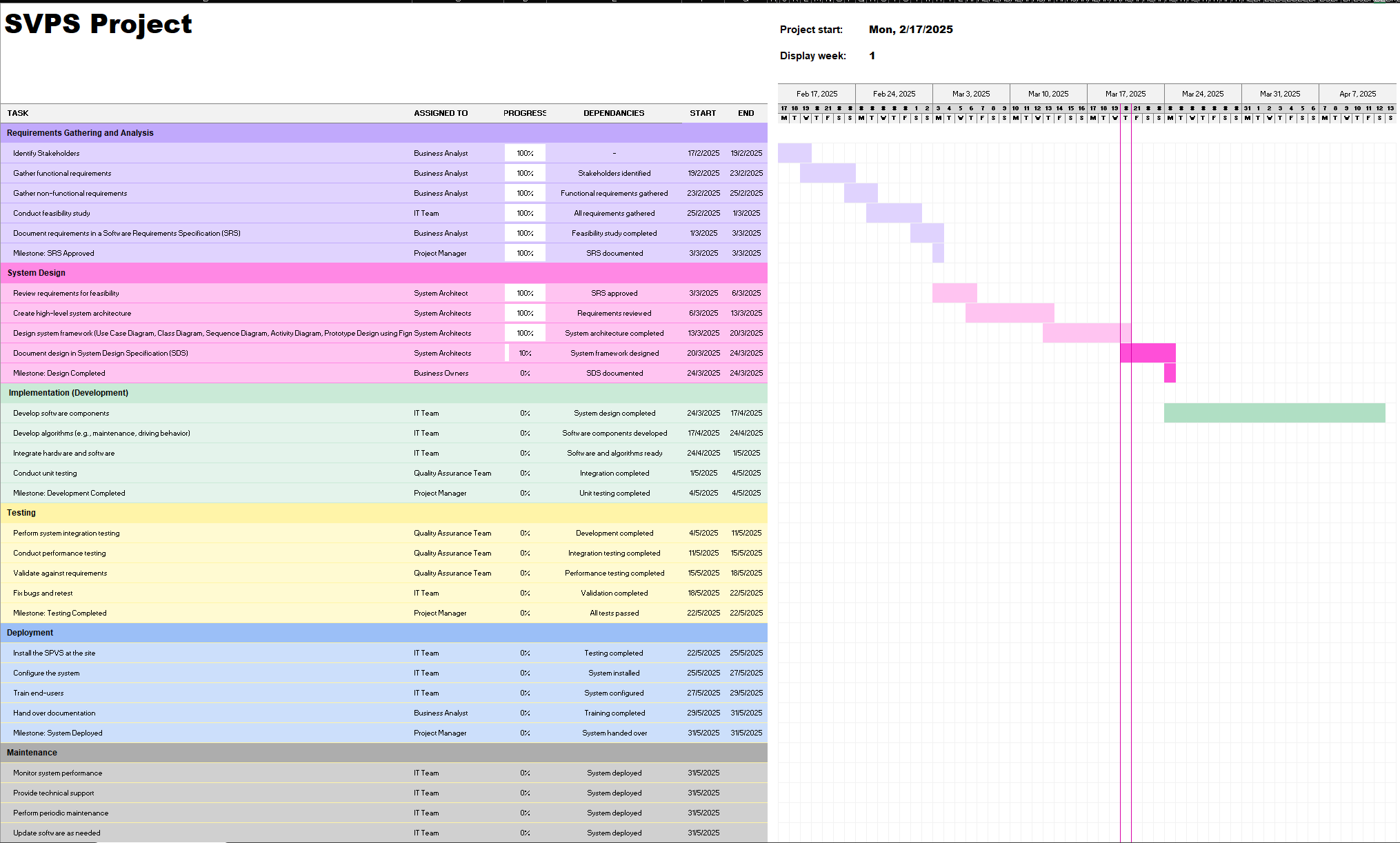
1. IoT sensors and vehicle diagnostic systems
2. Cloud infrastructure for sensor data storage and analysis
3. Advanced algorithms for data analysis
4. Development tools for mobile application programing
5. Navigation and mapping API

#### 2.2.2 Boundaries

The SVPS is designed for monitoring and improving vehicle performance, but it does not directly control vehicle operations, such as autonomous driving or automatic braking. Additionally, the system is primarily intended for passenger vehicles, with no immediate scope for heavy duty industrial vehicles. By establishing these clear project objectives and scope, the SVPS aims to transform how drivers manage vehicle efficiency and road safety. This system provides a comprehensive solution for modern vehicle performance management through real-time analytics, predictive maintenance, and smart navigation.

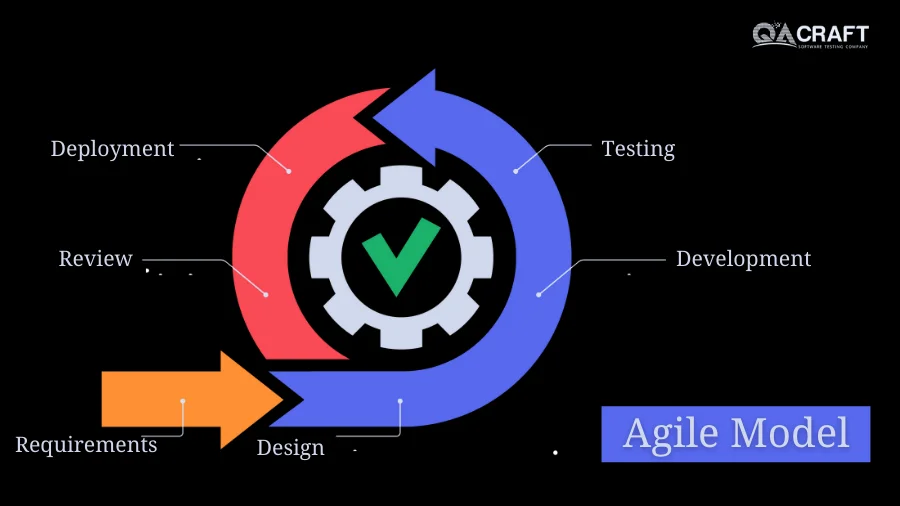
#### 2.2.3 Timeline

[Link to Gantt Chart](https://imailsunwayedu-my.sharepoint.com/:x:/g/personal/22070353_imail_sunway_edu_my/EWMayb7dPhxBmE8lKcMT2bgBqrB8p_4J6-eNZrJBVAaJQQ?e=l1L66x)



## 3.0 SOFTWARE PROCESS MODEL

Software Process Models are crucial for developing the SVPS. To develop this SPVS, the agile software process model has been applied, which is primarily designed for a project to adapt quickly to change requests (GeeksforGeeks, 2024).



According to Bhat (2023), the first step in the agile model is requirement gathering, where the development team will reach out to stakeholders, such as business owners and end users, to gather requirements. This is often done through meetings, interviews and surveys to understand the needs of the stakeholder, allowing the development team to evaluate and understand the technical and economical feasibility of the requirements to estimate development time, resources, potential cost and risks. Some examples of requirements would be real-time vehicle monitoring, driving behavior analysis and fuel efficiency tracking.

After the requirements are finalized, the team designs the requirements by creating diagrams, such as use case diagram, class diagram, sequence diagram and activity diagram, to illustrate how the features will function and integrate into the system. In our system, we also created prototypes of the mobile app and in-car display interfaces to ensure that it will provide an intuitive user experience.

Next, the team will start on the construction of the project in iterations to ensure a functional product is deployed after each sprint, which is a short, time-boxed period when a scrum team works to complete a set amount of work. In our system, we will be developing the system up until the requirement design step, where each sprint focuses on important features. Each sprint is followed by stakeholder reviews to refine and improve the system based on feedback.

Once the project development has been completed, the Quality Assurance team will conduct rigorous testing to ensure that the system meets the functional and performance expectations. Testing involves unit testing, where each module, like notification system and data collection, is tested individually to ensure that each individual part of the system can work properly on its own. Integration testing, which is used to identify issues that may arise when different parts of the system are combined and working together will also be conducted to ensure that there is seamless interaction between vehicle sensors, cloud processing and the application display. Lastly, system testing will be conducted to ensure that the system aligns with all the requirements and expectations of the stakeholders and it functions properly in any possible scenarios. For example, testing the system by simulating real-world usage to confirm that the system works under various common conditions that everyday users may encounter such as network delays or sensor errors.

After ensuring everything is prepared and the core functionalities have been developed and tested, the team will deploy the working project to the end users. In our case, deployment will include the mobile app launch via the app store and play store, vehicle sensor integration with collaborating with fleet operators, and the cloud platform setup for data processing and storage.

Finally the last step in the agile model is collecting overall user experience and feedback from the end users. Feedback could be both good and bad, such as glitches and bugs, certain features being useful to users, needing to scale up as new users grow, new feature requests and more. Listening to user feedback allows the software to continuously improve and evolve into a more refined version.

This process model facilitates rapid software development, where it builds the software by dividing the project into small and manageable parts, each controlling one working process of the software. This strategy allows for incremental development, where the system is able to continuously improve and efficiently address any issues that may arise during development. This enables stakeholders to provide feedback and new requirements more frequently as they can review and manage each iteration separately, ensuring that the system meets evolving requirements.

In addition, the agile model’s flexibility and adaptability make it an ideal choice for this project. By using the agile model, it is convenient to adapt to changes since the development is divided into small and iterative phases. This process allows for frequent releases and continuous refinement, in contrast to the Waterfall model, where the development follows a linear sequence and allows changes only after the system is fully completed. Furthermore, the agile model enables for rapid releases of functional components such as real-time data processing, cloud-based analytics, and mobile interfaces. This approach encourages the consistent collaboration between developers and stakeholders as the development team can rapidly incorporate feedback from the stakeholders and make adjustments along the way. The testing process is also more efficient in the agile model as each small phase is examined individually. Conversely, the Waterfall model tested the entire development system only after completion, which can be time-consuming and overwhelming.

Although the agile model is favored due to its flexibility and adaptability, it is not without its limitations. One of the largest limitations of the agile model is the potential lack of predictability because of the iterative nature of its development process, which makes it difficult to estimate the final completion time in the planning phase of the project. This leads to challenges in estimating the time, cost, and scope of the project development due to various uncertainties. Not just that, the agile model also requires a high level of user involvement in the testing and review stages. This is because the agile methodologies highlight the importance of continuous feedback and engagement from end-users. Therefore, if the end-users do not have the required time, expertise, or resources to be fully involved and engaged during testing, the lack of detailed and thorough feedback can result in poor software decisions or even slow down the development of the project.

Moreover, as the agile model focuses more on developing a working software, where they often prioritize speed and flexibility over extensive documentation, the lack of detailed documentation provides a big challenge for developers. Since each task is completed just in time for development, there may not be enough time to create detailed documentation about system requirements, design decisions, or changes made during development. Thus, the lack of detailed documentation creates difficulties for new team members to understand the project if it is handed over to a different team or when long-term maintenance is required, potentially leading to miscommunication (University of Minnesota, 2022).

# 

# Part B: Project Analysis and Design

## 4.0 REQUIREMENTS PLAN

### 4.1 Stakeholders and Their Roles

1. Project Sponsor
   1. Secure funding and ensure the project aligns with our company’s goal.
   2. Approve budgets and ensure the financial resources are allocated effectively.
2. Business Owners
   1. Define the business requirements for the system.
   2. Ensure compatibility and integrate the system into their existing operations/products.
   3. Approve requirements and ensure the project aligns with their organizational strategy.
3. Project Manager
   1. Manage the entire project, like planning, resource allocation and deployment.
   2. Oversee timelines and ensure deliverables are within the scope of the project.
4. Business Analyst
   1. Collect, analyze and document requirements from business owners.
   2. Coordinate communication between technical teams and clients.
5. IT Team
   1. Design, develop and maintain the system.
   2. Create user-friendly and intuitive designs for the system interface.
   3. Design the system according to the requirements stated by the clients.
   4. Develop algorithms to predict maintenance needs and driving behavior analysis.
   5. Ensure availability of real-time data processing.
6. Quality Assurance Team
   1. Test all features and functions.
   2. Ensure that all requirements are included.
   3. Certify that the system meets quality standards.
7. End User
   1. Users who will be using the system in their vehicle.
   2. Provide feedback on the user features, like the accuracy of real-time vehicle data, reliability of notifications and alerts, functionality of recommendations given to users for safer driving behavior, etc.
8. Regulatory and Compliance
   1. Ensure the system meets legal and safety regulations.
   2. Define the standards of data privacy and security regulations.
   3. Ensure compliance with transportation and safety regulations.
9. System Architects
   1. Review requirements for technical feasibility and scalability.
   2. Design framework and ensure functional and non functional requirements.

### 4.2 Requirements Process

1. Elication
   1. Collect requirements from direct discussions with key stakeholders through meetings, interviews or surveys to identify and understand user needs.
   2. Brainstorming sessions to refine raw requirements by discussing potential features, technology limitations and user expectations.
   3. Focus on understanding the expectations of the system.
2. Analysis
   1. Analyze collected requirements to analyze feasibility and scalability.
   2. Categorize requirements into functional and non-functional requirements.
   3. Categorize requirements by priority (Must-have, Nice-to-have, Optional/future features).
   4. Identify potential risks and issues of each feature.
   5. Create diagrams to better visualize the flow and interactions of the project, such as use case diagram, class diagram, sequence diagram and activity diagram.
   6. Ensure requested features are in alignment with project objective and scope.
   7. Give alternative solutions for unrealistic or inefficient requirements.
3. Validation
   1. Ensure the proposed system meets the stakeholder’s needs.
   2. Develop a prototype as a visual representation of the system, such as using Figma.
   3. Allow stakeholders to review the requirements to their satisfaction.
   4. Ensure all features and functions are in alignment to the stakeholder’s expectation.
   5. Documentation of all agreed requirements in a formal approval document.
4. Management
   1. Ensure all requirements remain accurate, updated and aligned with stakeholder needs while maintaining project feasibility.
   2. Handle any changes requested by the stakeholders.
   3. List all modifications, removal and additions of requirements in the documentation to ensure accountability.
   4. Ensure that modifications do not introduce inconsistencies or cause unintended issues in other system functionalities.
   5. Regular updates with the stakeholders, either through meetings, reports or demonstrations, to ensure the project is within expectations.

## 

## 5.0 SYSTEM FUNCTIONALITY

### 5.1 Functional Requirements

1. Vehicle Data Collection
   1. Data will be collected through vehicle sensors and uploaded to the cloud-based platform for analysis.
2. Predict And Track Maintenance Needs
   1. The notifications and alerts will display on the application and in-car display interface to inform the user of potential issues of the vehicle.
   2. Help users keep track of maintenance dates, like oil change or tire replacement based on the last changed date.
3. Identify Driving Behavior
   1. Use analysis algorithms in real time to calculate and determine problematic driving behavior and patterns that can be changed for safer and more efficient driving which leads to less wear and tear on the car over time.
   2. Provide tips and tricks to the user to improve fuel efficiency based on the user’s driving style, such as reducing rapid accelerations and brakings.
4. Optimize Routes
   1. Integrate GPS and navigation systems like Waze or Google Maps, and real-time traffic data to suggest optimized routes that will save the user’s time and increase fuel efficiency.
   2. Show route recommendation based on current road conditions.
5. Real-Time Notifications And Alerts
   1. Send alerts in real time to inform the user and suggest actions if there are problematic driving behaviors from the user.
   2. Send notifications for maintenance reminders such as oil change, tire change, low fuel, or even when the tire needs to be pumped.
   3. Send notifications when cars have warning lights, with explanations of the issue that is happening.
6. Vehicle Management
   1. Multiple cars can be added by entering the vehicle VIN (Vehicle Identification Number) and plate number.
   2. Added vehicles can be deleted from the account.
   3. Multi-vehicle support allows the user to monitor multiple vehicles at once.
7. User Authentication
   1. Users will be able to log in using their email or phone number with a password, or by using their Google or Facebook accounts.
   2. Users may reset their forgotten passwords via their emails.
   3. New users will need to register for an account in order to use the app.
   4. Users will need to be logged in to the app to access certain functions such as the vehicle and driving behavior insights.
8. Settings and Customization
   1. Allows users to enable or disable notifications from the application.
   2. Allows users to customize measurement units, such as miles to kilometers.
   3. Allows users to change between dark and light mode for the user interface.
9. Integration with OBD-II
   1. The SVPS will be integrated with the OBD-II system to collect vehicle diagnostics and performance data to improve efficiency by utilizing existing functionalities and codes where applicable.

### 

### 5.2 Non-Functional Requirements

1. Usability
   1. Intuitive design with a short learning curve for new users.
   2. Easy to follow tutorial to guide users through the system features.
   3. Accessible and user friendly interface.
   4. Security and privacy.
   5. Multi-factor authentication.
   6. Ensure that the system is in compliance with data security and privacy standards.
   7. Encrypt all user and vehicle data using advanced cryptographic methods to prevent unauthorized access.
2. Performance
   1. Provide real-time data processing with minimal latency.
   2. Send vehicle performance data from vehicle sensors near-instantaneously, ensuring fast and accurate analysis.
   3. Support high speed uploads and instant notification to keep users updated.
   4. Optimize data transmission speed for instantaneous processing and display.
3. UI/UX
   1. Consistent design and an aesthetic look to enhance user experience.
   2. High contrast visuals to ensure readability and accessibility for all users.
   3. Dark and light mode options for personal preferences.
4. Scalability
   1. Scalable system to allow for easy integration of new sensors and additional data streams without requiring major changes.
   2. Platform is able to handle an expanding user base, ensuring smooth operation over time.
5. Availability and uptime
   1. Maintain high availability and low downtime to ensure continuous operation.
   2. Stable system design to prevent crashes and ensure reliability.

## 

## 6.0 PROTOTYPE DESIGN

### 6.1 Mobile App Design:

Design Link:

<https://www.figma.com/design/4MyesULwWCKo2gpav50ndG/se?node-id=0-1&t=0nLBrXQs0axpABdq-1>

Prototype Link:

<https://www.figma.com/proto/4MyesULwWCKo2gpav50ndG/se?node-id=0-1&t=0nLBrXQs0axpABdq-1>

### 6.2 In-Car Display Design:

Design Link:

<https://www.figma.com/design/4MyesULwWCKo2gpav50ndG/se?node-id=1-2&t=0nLBrXQs0axpABdq-1>

# 

# Part C: UML Diagrams

## 7.0 USE CASE DIAGRAM

**A diagram of a network

AI-generated content may be incorrect.**

*7.0.1 Description for Use Case Diagram*

Actors:

* User
* Vehicle Sensors

Relationships:

* User is associated with use cases like login, register, get driving patterns advice, get routes suggestions, get maintenance advice, receive tyre pressure alert, receive speed alert, receive engine performance alert, receive fuel consumption alert, and manage vehicles, each representing a function that can be carried out by a user.
* Functions like getting insights and getting alerts are all included in the login function as users will need to login before they can perform these functions.
* The extension points from login to manage vehicles indicates that it is optional for users to manage their vehicles after successful login, such as adding or deleting vehicles from their account.
* An include relationship linked from manage vehicle to register function, as users will need to add a vehicle in the registering process to successfully create an account.
* All the functions except for login and register are included in getting info from the cloud database, which shows that these insights, alerts and vehicle information are stored and generated from the cloud database.
* Vehicle sensors will send all the vehicle information such as tyre pressure, fuel consumption, route, mileage and speed, driving patterns, engine performance and vehicle info, so an association relationship is needed between the vehicle sensors actor and these sending info use cases. The information will be sent, and it is included in update info to the cloud database, so that information can be stored in the cloud database and generate for the user.

Additional Explanation:

* The send vehicle info function associated with the vehicle sensors actor, detects the vehicle information such as Vehicle Identification Number (VIN) and sends the information to the cloud database for processing.

## 8.0 CLASS DIAGRAM

**A diagram of a computer flowchart

AI-generated content may be incorrect.**

## 

*8.0.1 Description for Class Diagram*

Relationships:

1. Generalization

Generalization is applied to represent the relationship between the classes stated below, as it allows an easier expansion process in the future. For example, expanding the insight class by adding new insights.

* Insight & MaintenanceNeed, RouteSuggestion, Driving Pattern
* Alert & SpeedAlert, FuelAlert, TyrePressureAlert, EnginePerformanceAlert
* SensorData & EnginePerformanceData, SpeedData, FuelData, TyrePressureData

2. Composition

Composition is applied to represent the relationship between the classes stated below, as they depend on each other and could not exist or process without each other. For example, a vehicle cannot exist in the system without a corresponding user.

* User & Vehicle
* Vehicle & SensorData

3. Association

Association is applied to represent the rest of the relationship between classes that are not mentioned above. They are connected and interact with each other, and they can exist independently. For example, User class and UserDBManager class is an association relationship as the User class uses the UserDBManager class to interact with the cloud database but does not depend on it for existence.

## 9.0 SEQUENCE DIAGRAM

1. Login to SPVS

A diagram of a project

AI-generated content may be incorrect.

2. Register

A diagram of a project

AI-generated content may be incorrect.

3. Get Insights: Driving Pattern, Route Suggestion, Maintenance Need

A blue and white diagram

AI-generated content may be incorrect.

4. Get Alerts

A white paper with blue squares

AI-generated content may be incorrect.

5. Sending Information

**A diagram of a process flow

AI-generated content may be incorrect.**

*9.0.1 Description for Sequence Diagram*

User-Interface:

* Front-end views where users can interact with, such as main view and login view
* Acts as an object to capture user input and display data

Controller:

* Acts as an endpoint between the front-end and back-end processes
* Different controllers used for specific tasks to avoid centralized logic
* For example, login controller and user controller, where login controller manage login process, and user controller manage user-related operations

Object:

* Represents an instance of a class (e.g. User object of User class)
* Acts as an intermediary between the Controller and DBManager Object
* Handles specific logic and data before interacting with the database through DBManagerObject

DBManager Object:

* Interacts with the cloud database
* Contains SQL queries and handles database operations
* Using different DBManager for different object ensures that the object has a specific interface for database interactions, which makes the system more organized, scalable, and easier to debug

## 10.0 ACTIVITY DIAGRAM

A diagram of a company structure

AI-generated content may be incorrect.

*10.0.1 Description for Activity Diagram*

The flow starts from the welcome screen of the application. Users can choose to login or sign up, each containing a slightly different process. For example, login will need users to choose the vehicle they want to monitor, while sign up will need users to add a vehicle to create their new account. Upon successful login or sign up process, users will be logged in to the profile page, which is also the main page of the application. From the profile page, users can access all of the other functions such as check driving pattern insights, trigger alerts, check maintenance need advice, check route suggestions, and manage vehicles. Decision node is not used here as the profile page provides direct access to all functions without any conditional logic, and users can freely choose any of the functions.

# References

Azami, M. F. A. M., Misro, M. Y., & Hamidun, R. (2024). Road crash dynamics in Malaysia:

Analysis of trends and patterns. *Heliyon*, 10(2). <https://doi.org/10.1016/j.heliyon.2024.e37457>

Bhat. (2023). *Agile process model: A deep dive into the SDLC agile model*. QACraft

<https://qacraft.com/agile-process-model-sdlc-agile-model/>

Carbon Care Asia. (2023). *Climate change and road traffic impact*.

<https://www.carboncare.org/en/climate-change>

GeeksforGeeks. (2024). *Agile development models Software engineering*. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering-agile-development-models/>

Siddharta, A. (2022). *Number of road accidents in Malaysia from 2012 to 2022.* Statista.

<https://www.statista.com/statistics/794875/number-of-road-accidents-malaysia/#:~:text=In%202022%2C%20the%20number%20of,highest%20number%20of%20road%20%20accidents>

Team, G. (2023). What is OBDII? History of on-board diagnostics. *Geotab*. <https://www.geotab.com/blog/obd-ii/>

University of Minnesota. (2022). *Agile Methodology: Advantages and Disadvantages.* <https://ccaps.umn.edu/story/agile-methodology-advantages-and-disadvantages>

World Health Organization. (2023). *Road traffic injuries*

<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>

Woods, C. (2023). *The role of GPS tracking in modern delivery businesses*. Locate2u. <https://www.locate2u.com/gps-tracking/the-role-of-gps-tracking-in-modern-delivery-businesses/#:~:text=These%20systems%20track%20the%20position%20and%20status,are%20available%20with%20vehicle%2Dmounted%20GPS%20tracking%20systems>