KHAZANAH NASIONAL BERHAD / K-YOUTH NURTURE 2026

CAPSTONE PROJECT CASE STUDY REPORT

DRIVE BUDDY: SMARTPHONE SENSOR BASED VEHICLE DIAGNOSTIC AND ASSISTANCE APPLICATION

Prepared by:

Name: Muhammad Hadif Bin Nazrujehan

Student Id: 2024978083

Programme: Bachelor of Computer Science (Hons.) Netcentric Computing

University: Universiti Teknologi MARA

Project Supervisor: Assoc. Prof. Ts. Dr. Nor Shahniza Kamal Bashah

K-Youth Trainer: Mr Mohd Shafiq Yussof

Submission Date: 7 September 2026

Executive Summary

Vehicle ownership presents a significant challenge for everyday motorists who lack the technical expertise to interpret dashboard warning indicators or understand how their driving habits impact vehicle wear. Conventional automotive telematics and diagnostic systems rely on specialized On-Board Diagnostics (OBD-II) hardware, imposing financial and technical barriers.

Drive Buddy is a mobile-native application built using the Flutter framework and Google Firebase to provide an accessible, hardware-free vehicle diagnostic and assistance solution. The application utilizes the embedded inertial measurement unit (IMU) within standard smartphones—specifically the accelerometer, gyroscope, and GPS—to capture real-time motion data. By computing total vector acceleration magnitudes and angular velocity, the system accurately detects unsafe behaviors such as harsh braking, sudden acceleration, and aggressive cornering. Software-level filtering algorithms were engineered to mitigate common mobile sensor inaccuracies, such as stationary GPS drift.

To address the usability gap in understanding vehicle symptoms, Drive Buddy features a generative AI chatbot powered by Google Gemini via a Retrieval-Augmented Generation (RAG) architecture. The chatbot dynamically retrieves the user's specific vehicle telemetry, recorded mileage, and logged behavioral events from Cloud Firestore to generate context-aware, plain-language troubleshooting advice.

Rigorous functionality testing on physical devices confirmed accurate kinematic event classification, effective noise suppression, and real-time cloud data synchronization. This project demonstrates how leveraging existing mobile computing power and generative AI can democratize vehicle telematics, reduce preventative maintenance costs, and promote safer driving habits for the general public.

Table of Contents

1. Project Context

Modern automobiles are equipped with complex electronic control units (ECUs), yet ordinary drivers remain largely disconnected from the operational health of their vehicles. Driving habits heavily influence the fatigue rate of mechanical components. Repetitive harsh braking, abrupt acceleration, and sharp cornering rapidly accelerate the wear of critical parts such as brake pads, suspension bushings, and tires. Because this deterioration occurs gradually, drivers often fail to associate their daily driving patterns with premature and costly servicing expenses.

This project was developed as a Capstone Project for the K-Youth Programme, addressing the real-world challenge faced by Malaysian private car owners who lack access to professional diagnostic advice. By utilizing the computational power and sensors already present in modern smartphones, Drive Buddy was conceptualized as a zero-additional-cost mobile companion. The system operates as an automated telematics recorder for preventative maintenance and an AI-driven virtual mechanic that interprets vehicle warnings in accessible language.

2. Problem Identification and Re-Definition

2.1. Initial Assumptions vs. Actual Discovered Issues

Initially, the problem was assumed to be a simple lack of access to scanning tools; drivers experienced breakdowns because they did not own diagnostic hardware. The early concept considered integrating low-cost Bluetooth OBD-II scanners or training neural networks to listen to engine acoustics via the phone's microphone.

Further analysis into automotive usability and mobile hardware limitations revealed deeper operational realities:

Hardware Friction: Mandating users to purchase, configure, and connect external OBD-II dongles creates a significant barrier to entry for non-technical drivers and the general public.

Dashboard Comprehension Gap: Drivers frequently ignore warning indicators (e.g., check engine, power steering) because the technical language in vehicle handbooks is intimidating and difficult to parse during urgent situations.

Rigid Maintenance Schedules: Traditional maintenance relies strictly on odometer milestones (e.g., every 10,000 km). This ignores behavioral variables; an aggressive driver will degrade brake pads long before the scheduled service interval, posing a safety hazard.

Computational Bottlenecks: AI acoustic engine characterization requires massive labeled sound datasets and heavy cloud processing, making it highly impractical and battery-intensive for a consumer mobile application.

2.2. Re-Defined Problem Statement

How can drivers monitor driving behaviors—such as aggressive acceleration and harsh braking that influence vehicle wear—and interpret dashboard warning indicators in real-time, using only inbuilt smartphone sensors, without requiring external diagnostic devices or technical expertise?

3. Analysis and Insight

To understand the root cause of deferred maintenance and evaluate potential solutions, an extensive review of existing driver behavior and technical tools was conducted.

A "5 Whys" analysis revealed that drivers do not inspect components early because there is no accessible mechanism logging their driving severity, and existing tools (like OBD-II scanners) output overly technical Diagnostic Trouble Codes (DTCs) that are incomprehensible to laypersons.

3.1. Gap Analysis of Existing Technical Solutions

System / Tool	Operational Method	Identified Gap / Limitation
Torque Pro / FIXD	Reads real-time ECU fault codes via OBD-II Bluetooth adapters.	Requires the purchase of external hardware; provides no driving behavior tracking or guidance.
The AI Mechanic	Uses neural networks trained on car sound patterns to detect anomalies.	Requires high processing power, advanced audio datasets, and is highly susceptible to environmental

		noise.
DriveSafe Prototype	Uses smartphone sensors to detect risky driving behavior with threshold logic.	Lacks any chatbot integration or post-trip predictive maintenance recommendations.
Rule-Based Chatbots	Provides troubleshooting via predefined keyword matching.	Operates independently of real-time sensor data; cannot inject dynamic vehicle context into answers.

The analysis concluded that a significant gap exists for a hybrid system: one that tracks kinematics via the smartphone while offering context-aware conversational AI support, entirely free of third-party hardware.

4. Creative Exploration

4.1. Comparison of Explored Approaches

During the planning phase, three distinct technological approaches were evaluated to solve the defined problem.

Evaluation Criteria	**Approach A: OBD-II Bluetooth Dongle**	**Approach B: Acoustic Engine Neural Network**	**Approach C (Chosen): Smartphone Sensor Fusion + RAG AI**
Hardware Cost	Moderate to High (Requires dongle purchase)	Free (Uses internal microphone)	Free (Uses internal IMU/GPS sensors)

User Setup Friction	High (Locating port, Bluetooth pairing)	Low (Recording audio)	Very Low (App download and immediate use)
Technical Reliability	Very High (Direct vehicle data)	Low (Interference from road/wind noise)	High (Physics-based vector calculations)
Processing Demand	Low (Data passthrough)	Very High (Continuous audio classification)	Balanced (Local math, cloud AI processing)

Rationale for Selection: Approach C was selected because it fulfills the core requirement of maximum accessibility. It eliminates financial barriers by relying strictly on the smartphone's built-in accelerometer, gyroscope, and GPS. Furthermore, coupling this with a Generative AI chatbot provides the necessary translation layer to turn raw telemetry into actionable, plain-language advice for the user.

5. Problem-Solving and Decision-Making

5.1. Agile Development Phases

The project was executed using the Agile methodology to ensure continuous testing and iterative improvement of sensor tracking and AI response capabilities.

Phase	**Key Professional Activities Executed**
Planning	Defined functional requirements (trip tracking, hazard detection) and non-functional requirements (real-time cloud sync, minimal delay).

Design	Developed system architecture, Entity Relationship Diagrams (ERD), flowcharts, and high-contrast user interface wireframes optimized for driving.
Development	Configured Flutter environment, integrated Firestore NoSQL collections, and engineered local algorithms to process sensor data into actionable driving events.
Testing	Conducted unit testing on sensor limits, integration testing for module interaction, and validation using simulated and real-world driving data.

5.2. Sensor Kinematics and Orientation Independence

A critical technical challenge was variable phone orientation. Drivers place phones in cup holders, mounts, or pockets at various angles. Relying on a single axis (e.g., the Y-axis) for acceleration would yield false data.

To solve this, the application calculates the total Euclidean vector magnitude of the G-forces across all three orthogonal axes:

$$M = \sqrt{(x^2 + y^2 + z^2)}$$

Harsh Braking & Acceleration: The system flags an event when the total acceleration vector magnitude exceeds 4.0~m/s^2.

Sharp Cornering: The system monitors the gyroscope's angular velocity, triggering a flag when the rotation rate exceeds 2.5~rad/s.

5.3. Mitigating GPS Signal Noise

During field testing, stationary vehicles (e.g., stopped at traffic lights) accumulated false mileage due to GPS signal multipath errors ("drift"). A two-tier software noise filter was implemented to ensure data integrity:

Speed Gate: Any GPS speed reading below 3.0~km/h is forcibly registered as 0.0~km/h.

Accuracy Gate: GPS coordinate fixes with an accuracy radius greater than 20 meters are immediately discarded.

5.4. Structuring the Database Architecture

To ensure real-time synchronization and secure data segregation, a NoSQL structure was designed using Google Cloud Firestore.

Cloud Firestore Data Collections

Collection	Stored Document Data	Purpose
Users	fullName, email, phone, UID	Manages secure authentication profiles.
Cars	plateNumber, model, transmission, currentMileage, ownerId	Stores vehicle specifications and links them to the user via foreign keys.
Trips	durationInSeconds, distanceInMeters, routePath, harshBrakingCount, sharpTurnCount	Records high-frequency telemetry data and safety event counters per driving session.

6. Use of Digital Tools or AI

Multiple digital platforms were utilized to architect, build, and deploy the solution.

- **Flutter & Dart:** Utilized as the primary front-end SDK to build a responsive, single-codebase application capable of running background sensor services on mobile devices.
- **Google Cloud Firestore & Firebase Auth:** Employed as the backend-as-a-service (BaaS) for persistent cloud storage, ensuring real-time data synchronization across user devices.
- **Google Gemini (Generative AI) API:** Integrated as the core logic engine for the conversational diagnostic assistant.

AI tools were used as support tools, while all final decisions and analysis were completed by the student.

To prevent the AI from generating generic or incorrect advice ("hallucinations"), a **Retrieval-Augmented Generation (RAG)** architecture was developed. Before the user's question is sent to the Gemini AI, the system automatically fetches the user's specific vehicle data (e.g., car model, current mileage, accumulated harsh braking count) from Firestore. This data is injected into the system prompt, forcing the AI to base its troubleshooting advice strictly on the factual, real-time condition of the user's vehicle.

7. Outcome and Impact

The project successfully delivered a fully functional mobile application featuring ten distinct user interfaces, including a Car Selection Dashboard, a Real-Time Driving Session HUD, a dynamic Trip Logbook, and the AI Chatbot Interface.

7.1. Functionality Testing Validation

The system underwent rigorous physical testing to validate data integrity and sensor logic.

Core Functionality Test Results

Module Tested	**Test Procedure**	**Actual Observed Result**	**Verdict**
GPS Noise Filtering	User sat in a stationary car with the app active.	Speed remained at 0.0~km/h despite minor GPS fluctuations; no false mileage added.	**PASS**
Harsh Event Detection	User performed a sudden brake force > 4.0~m/s^2.	System detected vector magnitude of 4.2~m/s^2; "Harsh Brake" counter incremented.	**PASS**
Cornering Detection	User performed a sharp turn > 2.5~rad/s.	Rotation rate exceeded 2.6~rad/s; "Sharp Turn" successfully logged to trip history.	**PASS**
Cloud Synchronization	User finished a trip and checked the log on a second device.	Trip data (Distance, Events) appeared on the second device within 2 seconds.	**PASS**
AI RAG & Safety Guardrails	User asked about a brake light, then asked for dangerous speeding advice.	AI retrieved exact car model for brake advice; AI refused to provide speeding instructions.	**PASS**

7.2. Project Impact

By combining automated behavioral logging with an intelligent virtual mechanic, Drive Buddy bridges the knowledge gap between technical automotive systems and ordinary drivers. The system proactively alerts users to component wear before failures occur, directly contributing to lower unexpected repair costs and safer driving environments.

8. Skills Demonstrated

This project required the application of diverse technical and analytical competencies.

8.1. Analytical Problem-Solving:

Identified physical limitations in smartphone hardware (e.g., variable phone placement and GPS drift) and developed programmatic mathematical solutions (3D vector magnitude and speed gates) to ensure data accuracy.

8.2. Cloud Architecture & Database Design:

Structured a scalable NoSQL database schema using Cloud Firestore to securely isolate user data while enabling real-time telemetry updates.

8.3. AI Integration & Prompt Engineering:

Implemented Retrieval-Augmented Generation (RAG) to safely constrain Large Language Models (LLMs), ensuring the AI generated factual, context-specific responses rather than harmful or generic output.

8.4. UI/UX Design for Safety:

Designed modular, high-contrast user interfaces with minimal text to reduce driver distraction during active vehicle operation.

9. Professional Reflection

Executing this project highlighted the critical difference between theoretical software design and real-world deployment. Algorithms that performed perfectly in controlled environments required significant recalibration when subjected to the physical vibrations and signal disruptions of a moving vehicle.

9.1. Identified Limitations

9.1.1. Internet Dependency

Because the application relies on Cloud Firestore and the Gemini API, core features (such as saving trips or asking the chatbot questions) become unavailable in areas with poor cellular coverage, such as rural highways or underground parking.

9.1.2. High Battery Consumption

Continuous, high-frequency polling of the GPS, accelerometer, and gyroscope drains smartphone batteries rapidly, making the app unsuitable for multi-hour road trips without a dedicated power source.

9.1.3. Signal Instability

While stationary drift was mitigated, physical obstructions like long tunnels can still cause signal loss, temporarily impacting route tracing.

9.2. Recommendations for Future Iterations

To elevate the system to a commercial grade, future development should focus on:

9.2.1. Offline Local-First Architecture

Integrating a local database (such as Hive or SQLite) to store trip data during network outages, with an automated background sync manager to push data to the cloud once connectivity is restored.

9.2.2. Voice-Activated AI

Implementing on-device Speech-to-Text (STT) and Text-to-Speech (TTS) to allow users to interact with the diagnostic chatbot completely hands-free while driving.

9.2.3. Gamification

Transforming the behavioral thresholds into a safety scoring system with leaderboards, utilizing behavioral psychology to incentivize safer driving habits across a community.

9.2.4. Hybrid OBD-II Support

Adding optional Bluetooth integration to pull precise Engine RPM and coolant temperatures for users who desire professional-level ECU diagnostics alongside smartphone tracking.

References

Ben Brahim, M., & Mahjoub, M. A. (2022). A smartphone-based driver behavior monitoring system using machine learning techniques. *Procedia Computer Science*, 198, 311-318.

Kwon, D., & Lee, H. (2021). An analysis of GPS-based odometer accuracy using smartphone sensors. International Journal of Transportation Science and Technology, 10(4), 390–399.

Lokman, M., & Salleh, M. (2022). Driving behavior classification using smartphone sensors and deep learning. Indonesian Journal of Electrical Engineering and Computer Science, 27(3), 1420–1428.

Pawlak, A., Mrugalska, B., & Wieleba, W. (2022). Does the driver understand the warning? Comprehension of vehicle dashboard symbols in modern cars. Advances in Science and Technology Research Journal, 16(5), 88–97.

Zhou, Y., Hu, Q., Song, J., & Qin, W. (2022). The AI mechanic: Acoustic vehicle characterization using neural networks. Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3553–3557.