

2. Requirements, Constraints, And Standards

2.1 REQUIREMENTS & CONSTRAINTS

Functional Requirements:

OBD-II Reader Compatibility:

The system must be able to read DTC codes from all OBD-II compliant vehicles.
(Constraint: OBD-II compatibility required for vehicles manufactured after 1996).

ESP32 Communication:

The ESP32 must establish a stable connection with the user's phone personal hotspot via Wi-Fi to transmit data to the cloud.

AI-Based Code Interpretation:

The system must interpret DTC codes using AI to gather insights from community forums, databases, and other online resources.

Real-Time Diagnostics:

The app must provide real-time analysis of car issues, displaying results as soon as codes are read from the vehicle.

User Notifications:

The app must send notifications to users when critical car issues are detected that require immediate attention.

Vehicle Health Monitoring:

The system should continuously monitor car performance metrics and alert the user to potential preventative maintenance opportunities.

Physical & Resource Requirements:

Hardware Portability:

The OBD-II reader and any other required side hardware must be small and portable enough to be carried on the go

Wi-Fi and Cloud Integration:

The system requires a stable Wi-Fi connection via the user's phone personal hotspot to send data to the cloud for processing.
(Constraint: Requires continuous hotspot availability during operation).

ESP32 Power Supply:

The ESP32 must be powered by the car's OBD-II port or an internal battery, eliminating the

need for external power sources.

Aesthetic Requirements

User-Friendly App Design:

The app's interface must be aesthetically pleasing with a clean, modern, and intuitive layout, incorporating easily recognizable icons and color schemes for different car statuses (e.g., green for good, red for urgent).

Intuitive App Design:

The app's interface must be easy for the user to understand and navigate.

User Experiential Requirements:

Easy Setup:

The Wi-Fi OBD dongle and ESP32 must be easy to set up, with clear instructions for pairing the device with the user's phone and connecting it to the cloud.

Fast Data Processing:

The system must send DTC data to the cloud and return results within a few moments

Seamless User Experience:

The user experience should be seamless, with the app automatically receiving results from the cloud once the ESP32 has sent diagnostic data.

Secure User Experience:

The user's data should be secure, which can be achieved through various cloud networking protocols.

Economic Requirements:

Cost-Effective Solution:

The Wi-Fi OBD dongle and ESP32 system must be affordable for DIY users and casual car

owners. (Constraint: Total hardware cost must not exceed \$100).

2.2 ENGINEERING STANDARDS

IEEE 802.11:

Ensures reliable Wi-Fi communication between the OBD dongle, ESP32, and the cloud.

IEEE 12207:

Guides the development and maintenance of FixIt's software and cloud infrastructure.

IEEE 11073:

Ensures secure data transmission between the ESP32 and the cloud.

SAE J1979:

Enables FixIt to reliably read and interpret DTC codes from OBD-II systems.

SAE J1850:

Facilitates seamless communication between the vehicle's OBD system and FixIt.

SAE J2534:

Ensures compatibility for future reprogramming or updates to vehicle control modules.

ISO 26262:

Ensures that FixIt's interaction with the OBD-II system doesn't interfere with vehicle safety.

ISO/IEC 27001:

Ensures the security of vehicle data stored and transmitted by FixIt.