

Requirements Specification

Data-Log-System - Regterschot Racing

Version: 1.0 **Date:** January 2026 **Project:** Telemetry Data Logging System for Regterschot Racing Team

1. Introduction

1.1 Purpose

This document specifies the functional and non-functional requirements for the Data-Log-System, a telemetry data logging and visualization platform for the Regterschot Racing Team.

1.2 Scope

The system captures real-time sensor data from a racing vehicle via CAN bus, stores it in a database, and provides web-based visualization for race engineers and analysts.

1.3 Stakeholders

- **Race Engineers:** Monitor real-time vehicle performance
 - **Data Analysts:** Analyze historical race data
 - **Pit Crew:** Track vehicle location and status
 - **Team Management:** Review race event summaries
-

2. FURPS+ Requirements

2.1 Functionality

ID	Requirement	Description
F-01	CAN Data Acquisition	System shall capture sensor data from CAN bus at 500 KBPS
F-02	Sensor Support	System shall support pressure, temperature, and wheel speed sensors
F-03	GPS Tracking	System shall record and display vehicle GPS coordinates
F-04	Data Storage	System shall persist all telemetry data to MySQL database

ID	Requirement	Description
F-05	User Authentication	System shall authenticate users via username/password login
F-06	Real-time Dashboard	System shall display live telemetry data on web dashboard
F-07	Event Management	System shall support starting and finishing race events
F-08	Data Visualization	System shall display sensor data as graphs and gauges
F-09	Tab Management	Users shall create custom analysis tabs with selected sensors
F-10	Historical Data	System shall allow querying of past race event data

2.2 Usability

ID	Requirement	Description
U-01	Web Interface	Dashboard shall be accessible via standard web browser
U-02	Responsive Design	Interface shall work on desktop and tablet devices
U-03	Login Feedback	System shall provide clear feedback on authentication status
U-04	Visual Clarity	Graphs shall use clear colors and labels for readability
U-05	Navigation	System shall provide intuitive navigation between views
U-06	Dark Mode	Dashboard shall support dark mode for pit environment

2.3 Reliability

ID	Requirement	Description
R-01	Data Persistence	No telemetry data shall be lost during normal operation
R-02	Connection Recovery	System shall reconnect automatically after network interruption

ID	Requirement	Description
R-03	Error Handling	System shall gracefully handle sensor disconnection
R-04	Service Restart	Raspberry Pi service shall auto-start on boot
R-05	Database Backup	Database shall support data export for backup purposes

2.4 Performance

ID	Requirement	Description
P-01	Data Interval	System shall aggregate and transmit data every 5 seconds
P-02	CAN Speed	CAN bus communication shall operate at 500 KBPS
P-03	Dashboard Latency	Dashboard shall update within 2 seconds of data receipt
P-04	Concurrent Users	System shall support at least 5 concurrent dashboard users
P-05	Data Retention	System shall retain telemetry data for at least 1 year

2.5 Supportability

ID	Requirement	Description
S-01	Modular Architecture	System shall use modular design for independent component updates
S-02	Configuration Files	System parameters shall be configurable without code changes
S-03	Logging	System shall log errors and events for debugging
S-04	Documentation	Code shall include inline documentation for maintenance
S-05	Standard Protocols	System shall use standard protocols (MQTT, HTTP, CAN)

2.6 + (Plus) Constraints

2.6.1 Design Constraints

ID	Constraint	Description
D-01	Hardware Platform	Data logger shall run on Raspberry Pi 4
D-02	Sensor Interface	Sensors shall connect via MCP2515 CAN controller
D-03	Frontend Framework	Dashboard shall be built with Angular 15+
D-04	Database	Data storage shall use MySQL database

2.6.2 Implementation Constraints

ID	Constraint	Description
I-01	Programming Languages	Backend: Python, C++; Frontend: TypeScript, PHP
I-02	Communication Protocol	MQTT for sensor-to-database communication
I-03	Operating System	Raspberry Pi shall run Raspberry Pi OS (Linux)

2.6.3 Interface Constraints

ID	Constraint	Description
IF-01	CAN Protocol	CAN frames shall follow standard CAN 2.0A format
IF-02	API Format	REST API shall return JSON responses
IF-03	Authentication	API shall use JWT tokens for authentication

3. MoSCoW Prioritization

3.1 Must Have (Critical for MVP)

ID	Requirement	Rationale
F-01	CAN Data Acquisition	Core functionality - no system without data capture
F-02	Sensor Support	Essential sensors for vehicle monitoring

ID	Requirement	Rationale
F-04	Data Storage	Data must be persisted for analysis
F-05	User Authentication	Security requirement for team data
F-06	Real-time Dashboard	Primary user interface for engineers
R-01	Data Persistence	Cannot lose race data
P-01	Data Interval	Acceptable refresh rate for monitoring

3.2 Should Have (Important but not critical)

ID	Requirement	Rationale
F-03	GPS Tracking	Valuable for track position analysis
F-07	Event Management	Organizes data by race session
F-08	Data Visualization	Enhances data interpretation
F-10	Historical Data	Enables post-race analysis
U-01	Web Interface	Accessibility from any device
R-02	Connection Recovery	Improves reliability in field conditions
S-01	Modular Architecture	Facilitates future development

3.3 Could Have (Desirable enhancements)

ID	Requirement	Rationale
F-09	Tab Management	Customization for different analysts
U-06	Dark Mode	Comfort feature for pit environment
P-04	Concurrent Users	Scalability for larger team
S-03	Logging	Debugging aid
U-02	Responsive Design	Mobile/tablet access

3.4 Won't Have (This iteration)

ID	Feature	Rationale
W-01	Predictive Analytics	Requires ML expertise, future enhancement

ID	Feature	Rationale
W-02	Mobile App	Web interface sufficient for MVP
W-03	Real-time Alerts	Notification system complexity
W-04	Video Integration	Out of scope for telemetry system
W-05	Voice Commands	Not essential for pit environment

4. Traceability Matrix

Requirement	Use Case	Component	Test
F-01	UC-02	Arduino, Raspberry Pi	Integration Test
F-02	UC-02	car_sensors.h	Unit Test
F-03	UC-04	gps.php, GPS module	Integration Test
F-04	UC-02	MySQL, rawDataSender	Unit Test
F-05	UC-01	AuthService, login.php	Unit Test
F-06	UC-03	Angular Dashboard	UI Test
F-07	UC-05	admin.php	Integration Test
F-08	UC-03	tabs.component	UI Test

5. Glossary

Term	Definition
CAN	Controller Area Network - vehicle communication protocol
MQTT	Message Queuing Telemetry Transport - IoT messaging protocol
JWT	JSON Web Token - authentication token format
Telemetry	Remote measurement and data transmission
PoC	Proof of Concept
MVP	Minimum Viable Product