Software Requirements Specification

Software Requirements Specification

"OK GR"

Table of Contents

1. Introduction

What is “OK GR”

“OK GR” is your personal race engineer. Designed for teams in the GR Cup Series.

Purpose of “OK GR”

“OK GR” allows drivers and engineers to fully utilise their race data by working together with the user to understand specific issues, challenges and goals. It then acts as an interactive coach that listens to the user and provides personalised coaching insights based on the data.

Intended Use

“OK GR” is intended to be used as a tool to analyse and visualise race data, whilst also acting as a personalised coach to help drivers improve their driving/strategy and engineers improve tuning parameters and race strategy.

Product Scope

The software comprises an interactive UI, a back end business logic and server side data storage.

2. Overall Description

Problem Definition

- Amateur and small GR-Cup teams lack the ability to effectively coach/train drivers to improve their performance and make their team more competitive in the series.

User Needs

Race Drivers/ Team Principles

- Drivers do not have access to elaborate engineering insight.
- No access to specialised engineers to analyse data.
- No personal race coach to help improve driving and offer actionable insight.

Engineers

- Lack the coaching skills needed to translate technical knowledge into feedback for the driver (more technically orientated).
- Vast quantities of data that cannot be easily analysed causes engineers to be overloaded with information.
- Much of the data goes unused as engineers are busy working on the car and not analysing data.

Assumptions & Dependencies

1. **Lack of Coaches** - it is assumed that most teams lack the role of coach and or race engineer that can offer actionable insight for the driver.
2. **Sensor Data is Available** - sensors on the car make it straight forward to pull telemetry data from the car that can then be analysed.

Solution - Product Introduction

- Personal AI-powered coach that draws on your race data to offer tailored insight, custom teaching points and summaries/logs of data to help you reach the podium.

3. System Features & Requirements

Functions of “OK GR”

High Level Overview of Functionality

“OK GR” allows the user to:

- Upload race data as a CSV file.
- View visualisations of the data.
- Ask about:
    - How they can improve
    - Information about specific parts of the race
    - Ask for explanations of the data
    - Ask for race summaries
- Create a roadmap to improve their performance based on the uploaded data.
- Save/download a summary of the conversion.
- View summaries of the previous coaching sessions.

User Flow

1. The user opens the site and is prompted with a clean chat window.
2. User uploads CSV of race data.
3. GR does a first pass analysis of the data and creates relevant visualisations of the data and gives a brief summary of the race.
4. GR asks the user how they felt about the race
5. Depending on the response, GR will dive deeper into the data regarding the issue/point that the user has brought up and create actionable insights on the topic to help the driver improve.
6. Steps 4 & 5 repeat until the user has gained all needed insight.
7. User presses “finish coaching session” and GR generates a summary of the session along with actionable insights to help the driver improve.

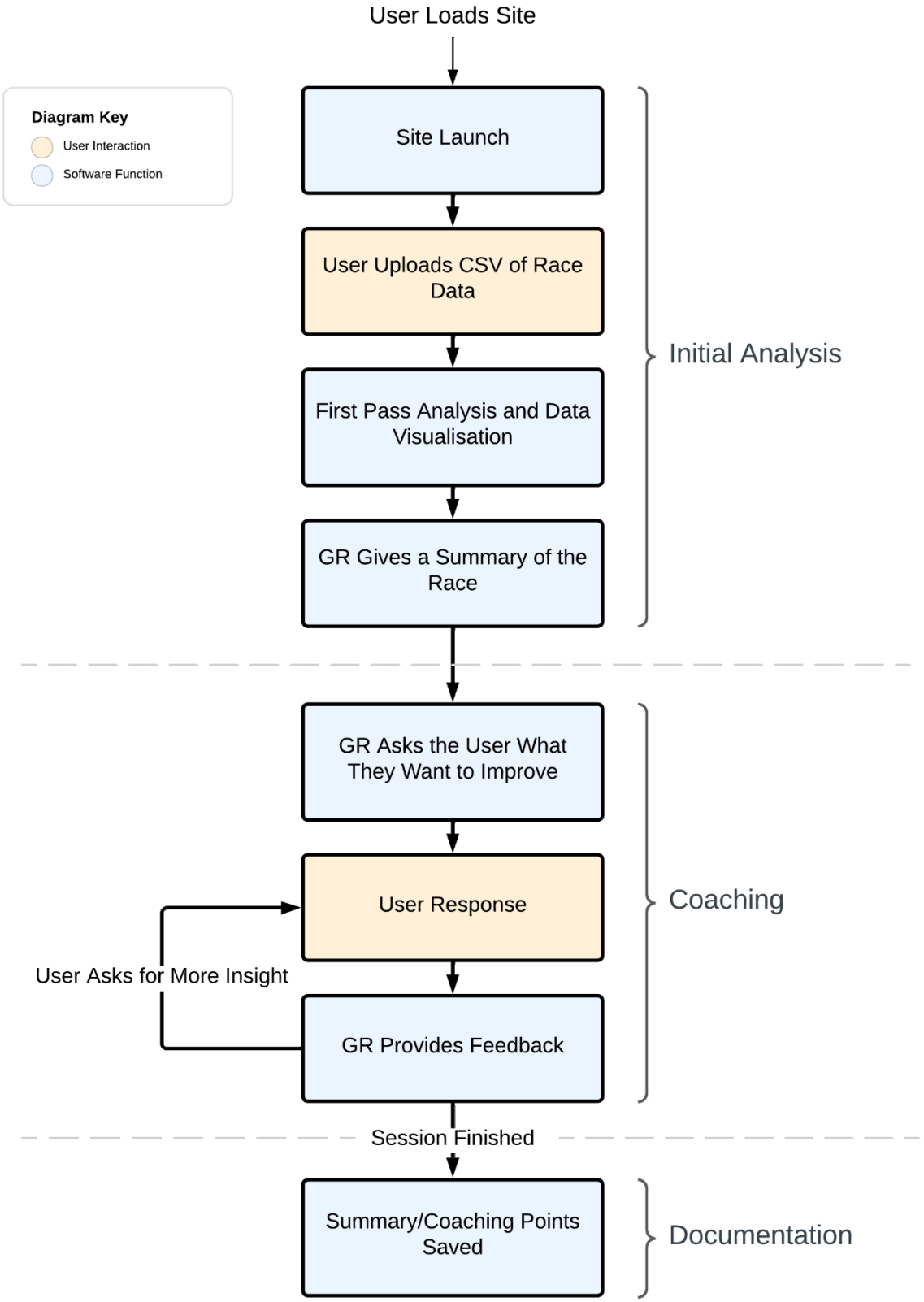

Figure 1. User flow diagram showing how a user interacts with the “OK GR” software.

Requirements

UI

1. **Chat Interface** - User can upload CSV data, ask questions via text or voice.
2. **Data Visualisation** - Area for visualised data to come up, option to maximise the view.
3. **Response of GR** - Once GR responds to a question, the response should show up in a clear section of the page.
4. **Summary Documents** - Section for saved summary documents and coaching documents.

Backend

Data Analysis

- GR takes in the CSV and provides a short summary of the race (positions, car behaviour, driving style etc.)

Data Visualisation

- First-pass visuals are created for the race and are presented along with the initial race summary.

Coaching Request

- User can make requests to GR via voice or text.

Coaching Response

- GR responds to the requests of the user via text and/or voice.

Race Summary Document

- GR creates a summary report of the race along with data visualisations.

Coaching Summary Document

- GR creates a coaching summary document for the user, with actionable insights.

Document Bank

- A tile in the UI is reserved for the summary documents that the user has created. Here they can view or download the documents.

4. Software Construction

UI - Front End

- Built with Streamlit

Data Analysis - Back End

- Built with Python, OpenAI API.

Data Storage - Server Side Logic

- Data stored in a Supabase database.

5. Software Subsystems Construction

The software consists of 3 subsystems as shown in Figure 1. Each of these subsystems creates a core functionality of the software and together they for the complete product.

Initial Analysis

Task of Subsystem

1. Take in race data as a CSV file.
2. Perform initial data analysis to create data visualisation of the session.
3. Write a summary of the race, drawing from the data. This includes a basic explanation of the data and highlights of key data points and trends.
4. Display data visuals and summary to the user via the UI.

AI Coaching

Task of Subsystem

The GR Agent uses a function calling algorithm to execute analysis on the user's data. To implement this, the Agent has access to tools which it can deploy to perform data analysis functions.

Tools available to GR Agent

The tools have been designed in order to:

1. **Maximally Leverage Available Data** - the tools are optimised to be most effective with the Toyota GR datasets that can be pulled from the cars during GR Cup events.

2. **Provide Maximal Insight** - the tools have been created to create the most effective data insight with respects to helping drivers improve. With this, the Agent aims to provide the most effective coaching possible to the user.

Research & Ideation

“OK GR” - Research & Ideation

The personal AI race engineer to help you reach the podium.

Toyota GR Hackathon

Table of Contents

Aim of Hackathon

Build software that helps people in the GR Cup ecosystem win races – especially **drivers and the people who support them** – using the official GR Cup data.

Primary Stakeholders

GR Cup Drivers (especially "Amature" class / gentleman drivers)

- Because there's an explicit "Driver Training/Insights" category, and the series is designed to be driver-centric rather than car-development-centric.

Driver Coaches & Race Engineers on GR Cup teams

- They are the ones who actually live inside the timing & telemetry all weekend, and would use tools for pre-event prediction, post-event analysis, and real-time analytics.

Toyota Gazoo Racing / TRD Performance & Data Engineers

- They run the series, curate the datasets, and care about tools that: Make GR Cup more competitive and professional. Help them understand performance, safety, series balance etc.

Problems Faced by GR-Cup Drivers

Profile:
Mix of pros and amateurs in identical GR86s in a very competitive spec series. They often have limited time, limited engineering support, and are not data scientists.

Driver Training / Insights – pain points

1. **"Where am I losing time, specifically?"**
   - Lap times + sector data exist, but:
   - Drivers struggle to see which corners matter most to overall lap time.
   - They often get spreadsheets or generic plots, not clear coaching cues.
   - Pain: "I see I'm 0.8s off, but I don't know if it's braking, mid-corner speed or exit."

2. **Converting complex telemetry into plain-language actions**
   - Telemetry includes speed, throttle, brake pressures, gears, etc.
   - Pain: It's hard for a non-technical driver to know: "On Lap 8 in T5, just brake 3m later and commit more apex speed."

3. **Learning new tracks quickly**
   - GR Cup visits multiple circuits; many Am drivers have limited or no prior experience.
   - Pain: Limited track walks / laps before qualifying. No easy way to use **historical GR Cup laps** to auto-generate "this is the reference style for each corner."

4. **Consistency & mental performance**
   - Drivers may put in one great lap but can't repeat it.
   - Pain: They don't have tools to see "You fade in laps 12-18 when in traffic" or "You brake earlier whenever someone is directly ahead."

5. **Racecraft instead of just hot-lapping**
   - Data is often focused on pure pace; but GR Cup racing is wheel-to-wheel.
   - Pain: Drivers can't easily see **how overtakes, defending, being stuck in traffic** affected their lap times and positions.

Problems Faced by Race Engineers/Coaches

Pre-event Prediction – pain points

1. **Planning realistic performance targets**
   - They have historical lap times, best 10-lap averages, class results, weather, etc.
   - Pain: A lot of manual work to estimate: “Given our driver and conditions, what’s a good target lap time / position / quali benchmark?”

2. **Setup / driving focus vs. time available**
   - Limited practice sessions, lots of possible improvements.
   - Pain: Hard to know whether to focus on driver confidence, brake bias, tire warm-up, etc. because they **don’t have a simple tool that ties historical data to likely gains**.

3. **Qualifying strategy**
   - When to send the car out, how many laps, how much space in traffic.
   - Pain: Manually anticipating track evolution, traffic density, and optimum lap timing.

Post-event Analysis – pain points

4. **Information overload after each session**
   - They get multiple CSVs per race (laps, sectors, results, weather, best 10 laps, telemetry)
   - Pain: It’s time-consuming to stitch that together into a **clear debrief story** for driver and team. Easy to miss patterns (e.g., “we always underperform after safety cars.”)

5. **Comparing across weekends and seasons**
   - Pain: It’s hard to benchmark “Did we actually improve this year?” across tracks and seasons because the data lives in separate files.

6. **Objective driver evaluation**
   - Teams might run multiple drivers; TRD might monitor many.
   - Pain: No simple, consistent metric that combines pace, consistency, racecraft, and incident rate using the existing timing/telemetry.

Real-time Analytics – pain points

7. **Spotting problems before it’s too late**
   - Telemetry includes trends like brake pressures, throttle application, pace relative to earlier laps.
   - Pain: During the race, pitwall is busy; they often only look at basic timing screens. They may miss early warning signs: brake fade, pace drop, driver fatigue, changing conditions.

8. **On-the-fly strategy decisions**

- Even in sprint races, things like yellow flags, safety cars, tire condition, and traffic matter.
- Pain: No smart system to say “If pace stays like this, you’ll catch P5 in 4 laps” or “Back off 2 tenths to save tires and still keep position.”

Problems Faced by Toyota Gazoo Racing / TRD Series Engineers

1. **Ensuring close, fair, driver-centric racing**
   - In a spec series, they want racing decided by driver skill and strategy, not randomness.
   - Pain: Hard to quantify whether BoP / rules are producing healthy racing (overtake opportunities, close gaps, etc.) using current tools.

2. **Understanding safety risks**
   - They have data about weather, sector times, incidents (implicitly via results & slowdowns).
   - Pain: Hard to identify “high-risk” corners or conditions where incidents spike, purely from existing spreadsheets.

3. **Showcasing GR Cup as a data-driven, modern series**
   - Hackathon tagline: “Unleash the Data. Engineer Victory.”
   - Pain:They want compelling demos / tools that make the series look cutting-edge and accessible to drivers and fans.

Problem

- Race engineers speak engineering slang
- Drivers are not as technical
- Mismatch in communication style means that the really important engineering insight does not land and is not as effective in helping the driver improve.
- Engineers don't have as good people skills to talk to the driver

Use Case

- Drivers want to analyse their race data and gain insight, in their own time, but do not have access to a race engineer to discuss.

Functionality

- AI race engineer that analyses race data, coaches the driver, supports them.
- Emotional support
- Summary of chat documents
- Improvement plan document - saved to a database/log/UI for reference of the chat

Product

- Systems that gather the technical data from race engineers and analyse the data.
- Clean web UI interface that starts blank.
- It has an LLM style chat window with a text enter or voice enter.
- The driver says "OK GR" and the program wakes up. It asks the driver what they need help with.
- The driver explains their need: help me analyse my race results, where can I improve, what was my worst lap, how can I change my driving style, is my braking good, should we adjust the car, etc.
- GR replies to the query by using the data to analyse the race and give feedback.
- The dashboard is updated with relevant data and graphs when the GR responds to questions.
- A summary (via transcription) and an improvement plan is created at the end of the discussion session which is provided to the driver as a document stored in the application. These can be downloaded, and are stored in the program so that it has a log of what has gone on.

Further Development

- This could be installed in Toyota GR vehicles as an AI system that helps you improve your driving on the track. You can speak to it in your car and the data is relayed to an app.

Judging Criteria

- **Application of the Datasets**: Does the Project apply the Datasets in an effective way, fitting with the categories? Does the Project showcase the datasets uniquely?
- **Design:** Is the user experience and design of the Project well thought out? Is there a balanced blend of frontend and backend in the software?
- **Quality of the Idea:** How creative and unique is the Project? Does the concept exist already? If so, how much does the Project improve on it?
- **Potential Impact:** How big of an impact could the project have on Toyota Gazoo Racing or the GR Cup?

Idea Finalisation

Driver Training/Insights

Problem Definition

- Amateur and small GR-Cup teams lack the ability to effectively coach/train drivers to improve their performance and make their team more competitive in the series.

Solution

- Personal AI coach that draws on your previous race data to offer tailored race insight, custom teaching points and summaries/logs of data to help you to reach the podium.