# **Software Requirements Specification**
for
Mileage Calculator Application

Prepared by Ajmal M S

Independent Project
Business Analysis

20 September, 2025

Contents

Revision History

Revision	**Date**	**Author(s)**	**Description**
1.0	20.09.2025	Ajmal M S	Version A

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to define the business requirements for the Mileage Calculator for Indian Vehicles Application. This application aims to provide vehicle owners with an easy-to-use tool to accurately estimate the amount of fuel needed and the corresponding cost for a given trip, based on the selected vehicle model, travel distance, and local fuel prices.

Normally, only high-end vehicles like a motorbike (which is used among most of the Indian population and tourists for convenience) contain an option in the TFT/digital information cluster to check and track mileage. This mileage is also an average mileage, so the user cannot get a real understanding of the mileage. The application will streamline journey planning by allowing users to select specific vehicle types and models, input current fuel rates, and receive precise, actionable information to optimize fuel usage and cost management.

This document serves as a reference for stakeholders—including product owners, business analysts, developers, and testers—to ensure a shared understanding of the project objectives, scope, and deliverables, and to guide the design, development, and implementation of the solution.

1.2 Document Conventions

To ensure consistency and clarity throughout this document, the following conventions are used:

- Terminology and Acronyms: Terms that are either specific to the project or technical in nature are defined in the Glossary section (Appendix A) and are capitalized upon first use.
- Text Formatting:
  - Bold text highlights section titles, critical requirements, and emphasis points.
  - Italic text denotes notes, remarks, or examples.
- Requirement Identification:
  - Each requirement is assigned a unique identifier composed of a prefix (e.g., BR for Business Requirement, FS for Functional Specification) followed by a sequential number to enable easy tracking.
- Tables and Lists:
  - Requirements, constraints, and features are presented in tabular or bulleted format for readability.
- Diagrams and Models:

  - Any referenced process flows, use case diagrams, or architectural models follow standard UML notation and are included in the Appendices (Appendix B).
- Version Control:
  - All document versions and major revisions are tracked in the Revision History section to maintain traceability.
- ** References:
  - External documents, standards, or regulatory guidelines referred to during requirements elicitation are cited in the References section.

1.3 Intended Audience and Reading Suggestions

This Business Requirement Document (BRD) is intended for multiple stakeholders involved in the development, management, and usage of the Mileage Fuel Cost Calculator Application, including:

- Project Sponsors and Product Owners: To understand the high-level business needs, objectives, and scope of the application.
- Business Analysts and Functional Analysts: To gather detailed requirements, analyze business processes, and ensure alignment with stakeholder needs.
- Development Team and Technical Leads: To translate the business requirements into functional and technical specifications for design and implementation.
- Quality Assurance/Testers: To define acceptance criteria and validate that the solution meets business requirements.
- End Users: To gain insight into the application features and usability expectations.
- Support and Maintenance Teams: To understand operational and user support requirements post-deployment.

Reading Suggestions:

- Readers unfamiliar with business requirement documents should start with Sections 1 and 2 for project overview and scope.
- Technical teams should focus on Sections 3 to 5 for detailed interface requirements, system features, and non-functional aspects.
- Appendices provide supporting information such as glossary, analysis models, and pending decisions for a comprehensive understanding.

This structure ensures that each stakeholder accesses relevant information efficiently, facilitating collaboration throughout the project life-cycle.

1.4 Product Scope

The Mileage Fuel Cost Calculator Application project focuses on delivering a user-friendly solution for vehicle owners to estimate fuel consumption and associated costs based on specific vehicle models, distances to be traveled, and current local fuel prices.

In Scope:

- Development of a web-based or command-line interface enabling users to select vehicle make and model.
- Input fields for distance and fuel price to facilitate accurate cost calculations.
- Implementation of a calculation engine that uses vehicle mileage data to compute fuel requirements and total cost.
- Display of results with options for users to review and export summaries.
- Documentation and training materials, including business requirements and functional specifications.
- Agile-based project management with progress tracking using Jira, including user story creation, sprint planning, and status reporting.
- Process flow modeling using Lucidchart or draw.io to visualize user interactions and system workflows.

Out of Scope:

- Real-time fuel price integration from external APIs (planned for future releases).
- Mobile application development (limited to web or desktop interfaces).
- Comprehensive vehicle maintenance tracking or diagnostics beyond fuel calculation.

This scope ensures focused development and delivery of a valuable MVP while setting a foundation for incremental enhancements aligned with user feedback and evolving requirements.

1.5 References

The following documents, websites, and tools were referenced during the preparation of this Business Requirements Document:

- Atlassian Documentation and User Guides
  `https://www.atlassian.com/software/jira`
- Lucidchart Documentation
  `https://www.lucidchart.com/pages/how-to-make-a-flowchart`
- Business Requirements Document Templates and Examples," Asana Blog, 2025
  `https://asana.com/resources/business-requirements-document-template`
- How to Write a Business Requirements Document," ProjectManager.com, 2025
  `https://www.projectmanager.com/blog/business-requirements-document`
- Internal discussions and feedback from potential users and stakeholders during the requirements elicitation phase.

Chapter 2

Overall Description

2.1 Product Perspective

The Mileage & Fuel Cost Calculator Application will serve as a standalone tool designed to assist individual vehicle owners and commuters in estimating their fuel consumption and related expenses more accurately based on vehicle-specific data and current fuel prices.

This product presents a user-centric approach by offering specific vehicle models and year selections, enabling tailored mileage calculations instead of generic assumptions. It will function primarily as a web-based or command-line interface, easily accessible without requiring integration with larger enterprise systems at this stage.

From a high-level architectural standpoint, the application operates as a black-box system where inputs (vehicle data, distance, fuel price) produce outputs (fuel quantity, cost) without exposing internal implementation details to the end user.

While the initial release focuses on core fuel calculation and cost estimation capabilities, the product scope anticipates future enhancements such as real-time fuel price integrations, multi-vehicle support, and advanced reporting features.

The product will rely on standardized technologies such as Python for processing, supported by documentation, diagrams (created using Lucidchart), and Agile-managed development practices using Jira and Confluence for task and story tracking.

This perspective reflects a business-driven solution optimized for simplicity, accuracy, and usability, targeting individual users rather than integration into enterprise IT landscapes or extended fleet management systems.

2.2 Product Functions

The Mileage & Fuel Cost Calculator Application will provide the following key functions to meet the user's needs and business requirements:

- Vehicle Selection: Users can select the make, model, and year of their vehicle from a predefined list containing associated mileage data. This ensures accurate fuel consumption estimates.
- Input Distance: Users can enter the distance they plan to travel in kilometers. The input will be validated to accept only positive numeric values.
- Input Fuel Price: Users will input the local price of fuel per liter to calculate the estimated cost for the journey accurately.
- Fuel Calculation: The system will calculate the volume of fuel required using the formula:

$$\textbf{Fuel Needed (Liters)} = \textbf{Distance (km) / Vehicle Mileage (km/liter)}$$

- Cost Estimation: The app will multiply the calculated fuel needed by the fuel price to estimate the total cost of fuel for the journey:

$$\textbf{Total Cost}=\textbf{Fuel Needed} \times \textbf{Fuel Price}$$

- Display Results: The calculated fuel quantity and estimated cost will be displayed clearly to the user with two decimal places.
- Export Calculation: Users will have the option to export the trip summary (vehicle details, inputs, calculation results) as a CSV or PDF file for record-keeping.
- User Interface: The interface will be responsive and designed for ease of use, with appropriate input controls and helpful error messages.
- Data Validation and Security: Input validations will prevent incorrect or malicious data. Basic security measures will ensure safe handling of user input and prevent injection attacks, preparing the system for future web deployment.

2.3 User Classes and Characteristics

The Mileage & Fuel Cost Calculator Application is intended to serve a diverse group of users with varying degrees of technical proficiency and usage patterns. The following user classes have been identified:

- Individual Vehicle Owners: Primary users who own a single vehicle (car or bike) and seek to estimate their fuel consumption and cost for daily commuting or occasional travel. Typically, these users have basic to moderate technical skills and prefer a simple, intuitive interface.
- Fleet Managers or Small Business Owners: Users responsible for managing multiple vehicles within a small fleet who require periodic fuel cost tracking for budget management. These users are more data-driven and may benefit from export and reporting features for record-keeping.
- Environmental Enthusiasts: Users interested in monitoring their vehicle's fuel efficiency to reduce environmental impact. They may use the application regularly to optimize driving patterns and cost.

Key characteristics across user classes include:

- Need for accuracy in calculation for budget and planning purposes.
- Preference for a clear and concise interface and outputs.
- Demand for some level of reporting or export capability.
- Varying levels of comfort with technology; therefore, the application must maintain user-friendliness and accessibility.

The system is designed primarily for individual users; however, scalable features like multi-vehicle management and detailed reporting accommodate broader user needs and future growth.

2.4 Operating Environment

The Mileage & Fuel Cost Calculator Application will operate primarily as a lightweight, cross-platform tool accessible through a web browser or command-line interface. It is designed to function efficiently in the following environment constraints

- Hardware Requirements:
  The application requires minimal hardware resources and is compatible with standard consumer-grade laptops or desktops with at least 4 GB of RAM and modern multi-core processors.
- Operating Systems Supported:
  The application is platform-independent and supports Windows 10/11, macOS (latest versions), and popular Linux distributions.
- Software Requirements:
  Python 3.x runtime environment for the backend calculation engine. Modern web browsers (Google Chrome, Mozilla Firefox, Microsoft Edge) for web-based UI. Compatibility with Microsoft Office 365 suite for export and reporting features.
- Network Requirements:
  Internet connectivity is required for users wishing to access online fuel price updates or cloud-based documentation and collaboration tools like Jira and Confluence.
- Development and Deployment:
  The project will be developed following Agile methodologies, managed via Jira project boards, and documentation maintained in Confluence. Lucidchart or draw.io will be utilized for process and architectural diagrams.
- Security Considerations:
  Basic input validation and secure coding practices will be implemented to mitigate common vulnerabilities such as injection attacks or cross-site scripting, especially if extended to web deployment.

This operating environment ensures the application remains accessible to a broad user base while leveraging modern and scalable tools to facilitate development, collaboration, and maintenance. This operating environment ensures the application remains accessible to a broad user base while leveraging modern and scalable tools to facilitate development, collaboration, and maintenance.

2.5 Design and Implementation Constraints

The design and implementation of the Mileage & Fuel Cost Calculator Application will be subject to the following constraints:

- Technical Constraints:
  The application will be developed using Python as the core programming language for calculations, with compatibility targeted for cross-platform environments. User interface design will be constrained by the choice of either a command-line interface or a lightweight web interface for prototype purposes.
- Resource Constraints:
  Development efforts will be limited by available time and budget, necessitating a minimum viable product (MVP) approach focusing on core functionality before any advanced features, such as real-time fuel price integration or multi-vehicle management.
- Operating Constraints:
  The software must run reliably on major operating systems: Windows, macOS, and Linux, with minimal hardware requirements to accommodate a broad user base.
- Security Constraints:
  While basic validation and protection against common input vulnerabilities will be implemented, comprehensive security frameworks and compliance certifications will not be part of the initial version.
- Usability Constraints:
  The application interface will prioritize simplicity but may lack advanced interactive features or accessibility compliance in the first iteration.

- Documentation and Tooling Constraints:
  Project tracking and documentation will use Jira and Confluence on their free or trial tiers, which may restrict advanced customization or user numbers.

These constraints shape project decisions, ensuring delivery feasibility while setting expectations for future enhancements and scalability.

2.6 User Documentation

Comprehensive user documentation will be provided to assist end users in effectively operating the Mileage & Fuel Cost Calculator Application. The documentation will include:

- User Guide:
  The guide will include screenshots and step-by-step instructions to facilitate ease of use.
- Installation and Setup Guide:
  Guidance will also be provided for initial configuration.
- Frequently Asked Questions (FAQ):
  A section addressing common user questions and troubleshooting tips to reduce support requests.
- Quick Reference Sheet:
  A concise summary of key functions and shortcuts to aid frequent users in quick operation.
- Support Information:
  Contact details and guidance on how users can seek assistance, report issues, or provide feedback.

2.7 Assumptions and Dependencies

Assumptions:

- Users possess basic computer literacy and can interact effectively with either the command-line or web interface.
- Fuel prices will be manually entered by users, as integration with real-time fuel pricing APIs is not within this project's current scope.
- Vehicle mileage data provided is representative based on manufacturer specifications; actual mileage may vary due to external factors.
- All necessary software tools for development and documentation (Python, Lucidchart, Jira, Confluence) will be accessible and operational throughout the project lifecycle.
- Stakeholders will respond promptly to feedback and requirement clarification requests to ensure project progress.

Dependencies:

- The project depends on development and documentation systems such as Jira (for issue tracking), Confluence (for documentation), and Lucidchart/draw.io (for process and architecture diagrams).
- Accuracy of outputs is dependent on precise user inputs, including vehicle details and fuel prices.
- The deployment environment must support the required software components, including Python runtime and modern browsers.
- Effective communication and collaboration among project stakeholders and teams are essential for timely delivery and requirement validation.

Chapter 3

External Interface Requirements

3.1 User Interfaces

The user interface for the Mileage & Fuel Cost Calculator Application will be designed to offer a seamless and intuitive experience for users of all technical abilities. The primary interface will be a responsive web-based application accessible via modern browsers (Chrome, Firefox, Edge, Safari). The interface will include:

- Dropdown menus for vehicle model and year selection.
- Numeric input fields for travel distance and fuel price.
- Display panels to show calculated fuel quantity and estimated cost.
- Export buttons for generating CSV or PDF reports.
- Error validation and help tooltips for user guidance.

For development and prototype demonstration, a command-line version will be provided to illustrate core logic and calculation accuracy.

3.2 Hardware Interfaces

The application will operate on standard consumer hardware, including laptops, desktops, and tablets, with minimum specifications:

- CPU: At least a dual-core modern processor
- RAM: Minimum 4GB
- Compatible operating systems: Windows 10+, macOS 10.14+, Linux
- Input devices: Keyboard and mouse or touchscreen for data entry
- No specialized hardware interfaces are required

3.3 Software Interfaces

The Mileage App interfaces with the following software components:

- Python Runtime Environment: For the calculation engine in the backend.
- Web Browser: For accessing the web-based UI version.
- MS Office 365 Applications: For exporting calculations and reports into Excel or Word formats.
- Atlassian Jira and Confluence: For project management, requirements tracking, and documentation.
- Lucidchart or draw.io: For visualizing flow diagrams and system architecture.

No external database or API integration is incorporated in the initial release.

3.4 Communications Interfaces

The application's communication will occur primarily between the user's device and local processing components. As the app does not currently have remote servers or cloud integration, no network communication protocols are initially required.

Future versions may include internet-based services for retrieving live fuel prices or cloud storage for user preferences, which would then implement secure HTTP/HTTPS protocols.

Project communication among stakeholders will be facilitated through:

- Microsoft Teams: For meetings and instant messaging.
- Outlook: For email notifications and formal communications.
- Jira: For task status updates and release notes.
- Confluence: For collaborative documentation and change logs.

Chapter 4

System Features

4.1 Vehicle Selection & Management

- The system shall provide users with an interface to select their vehicle make, model, and year from a predefined list containing accurate mileage data.
- The vehicle database will be expandable to allow adding new models in future updates.
- Users shall be able to manage their preferred vehicles for quicker access in repeated use cases.

4.2 Distance and Fuel Price Input

- Users shall be able to enter the distance they intend to travel in kilometers through a numeric input field.
- Input validation will ensure positive numeric values only.
- Fuel price per liter will be entered manually by the user to accommodate location-specific variations.

4.3 Fuel Calculation & Cost Estimation

- The application shall calculate the required volume of fuel based on the formula:
- Fuel Needed = Distance / Vehicle Mileage
- The total cost will be estimated by multiplying the fuel required by the input fuel price.
- Calculations shall be accurate up to two decimal places.

4.4 Result Display and Export Options

- The calculated fuel quantity and estimated cost will be displayed clearly on the interface.
- Users shall have the ability to export the trip summary, including vehicle details and calculations, in CSV or PDF formats for record-keeping or sharing.
- Proper error messages and input prompts shall guide users for any incorrect or missing input values.

Chapter 5

Other Nonfunctional Requirements

5.1 Performance Requirements

- The system shall provide calculation results within 2 seconds after user input submission under normal operating conditions.
- The user interface shall respond seamlessly to user interactions, with page loading times not exceeding 3 seconds on standard broadband connections.
- The application shall handle up to 100 concurrent users in the initial deployment environment without performance degradation.

5.2 Safety Requirements

- he application shall ensure that input data is validated to prevent processing errors and system crashes.
- User inputs will be sanitized to protect against injection attacks and malformed data that could compromise system stability.
- Error messages will be informative yet secure, avoiding exposure of system internals.

5.3 Security Requirements

- The system shall implement input validation to prevent code injection or scripting attacks.
- Sensitive user data, if stored or transmitted, shall be encrypted according to industry standards.
- Access to application source code and project documentation will be restricted to authorized personnel through secured collaboration platforms like Jira and Confluence.

5.4 Software Quality Attributes

- Reliability: The application shall maintain a 99.9% uptime without critical failures during standard use.
- Usability: The system shall be intuitive and accessible, catering to users with varying technical skills.
- Maintainability: Code and documentation shall be modular and well-commented to support easy updates.

- Portability: The application shall run on major operating systems, including Windows, macOS, and Linux.
- Scalability: The design shall accommodate future feature additions such as multi-vehicle support and real-time fuel pricing.

5.5 Business Rules

- Only valid, positive numeric inputs will be accepted for distance, fuel price, and vehicle mileage.
- Vehicle mileage data will be sourced from reliable manufacturer specifications and updated periodically.
- Users shall be responsible for entering current fuel prices accurately.
- Exported reports must include a timestamp and user details for tracking purposes.

Chapter 6

Other Requirements

This section captures additional requirements not specifically covered in prior sections but necessary for a complete understanding of project scope and execution.

- Legal and Regulatory Requirements:
  The application will comply with applicable data privacy regulations and user consent guidelines for any stored or processed data.
- Data Storage and Retention:
  Any user data retained (such as vehicle preferences or trip history) will be stored securely for a period defined by user settings and compliance requirements.
- Training and Support:
  Basic user training materials will be developed to aid onboarding. Support will be provided via email and integrated within the application help sections.
- Backup and Recovery:
  Data backup and recovery procedures will be considered in future versions as user data retention grows.

Appendices

Appendix A

Glossary

This appendix defines key terms, acronyms, and abbreviations used throughout the document to ensure a common understanding among all stakeholders.

Term	Definition
TFT	Thin Film Transistor
BRD	Business Requirement Document
FSD	Functional Specification Document
MVP	Minimum Viable Product
API	Application Programming Interface
UI	User Interface
CSV	Comma-Separated Values
PDF	Portable Document Format
Jira	Agile project management and issue tracking tool
Confluence	Documentation and collaboration platform

Table A.1: Term / Abbreviation and Definition

Appendix B

Analysis Models

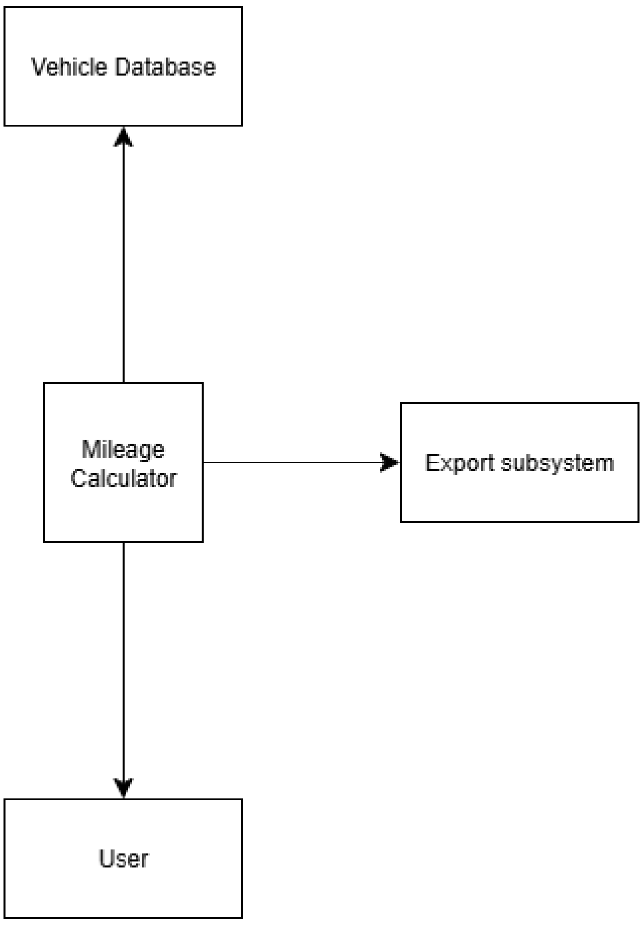

Figure B.1: System Context Diagram

Appendix C

To Be Determined List

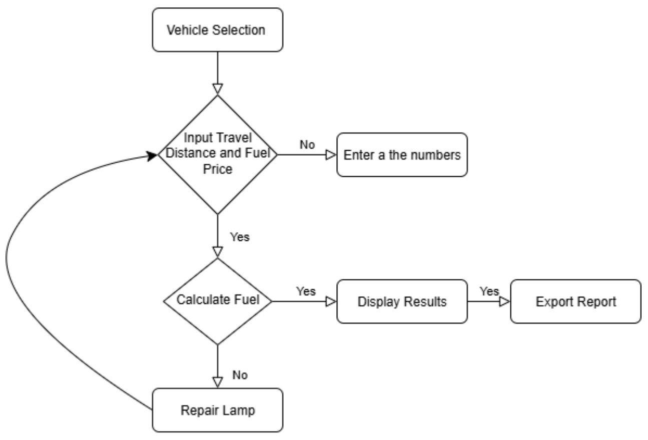

Figure C.1: Business Process