**Smart Fleet System**

**-POC-**

**Functional Design Document**

**Document Control**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Date** | **Doc. Version** | **Description (if CR is applicable, give CR id)** |
| Redouane BELBAHRI | 11/10/2018 | V1.0.0 | Full Draft |

Reviewed by

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reviewer** | **Date** | **Doc. Version** | **Review Type** | **Review report link** |
|  |  |  |  |  |
|  | dd/mm/yyyy |  |  |  |

Approved by

|  |  |  |  |
| --- | --- | --- | --- |
| **Approvers** | **Date** | **Doc. Version** | **Signature / Reference** |
|  | dd/mm/yyyy |  |  |
|  | dd/mm/yyyy |  |  |

Referential Document and Repository

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **File Name** | **File Location** | |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| Other | N/A | | File references have been included within each chapter. |

**Table of Contents**

[1. Purpose 5](#_Toc529792948)

[1.1 Intended Audience 5](#_Toc529792949)

[1.2 Scope of Work 5](#_Toc529792950)

[1.3 Business Case for the Product 5](#_Toc529792951)

[1.4 Acronyms 6](#_Toc529792952)

[2. PROJECT DEFINITION AND REQUIREMENTS 6](#_Toc529792953)

[2.1 Product Perspective 6](#_Toc529792954)

[2.2 Data Requirements 7](#_Toc529792955)

[2.3 Product Functions 7](#_Toc529792956)

[2.4 Functional Requirements 8](#_Toc529792957)

[2.5 Data Import Requirements 9](#_Toc529792958)

[3. Quality of Service Requirements 9](#_Toc529792959)

[3.1 Data Storage 9](#_Toc529792960)

[3.2 Data De-identification 9](#_Toc529792961)

[3.3 Data Validation 9](#_Toc529792962)

[3.4 Data and User Access 10](#_Toc529792963)

[3.5 Auditing 10](#_Toc529792964)

[3.6 Usability 10](#_Toc529792965)

[3.7 Reliability 10](#_Toc529792966)

[3.8 Performance 10](#_Toc529792967)

[3.9 Supportability 11](#_Toc529792968)

[3.10 System Availability 11](#_Toc529792969)

[4. Reporting Requirements 11](#_Toc529792970)

[5. Integration Requirements 11](#_Toc529792971)

[5.1 Fuel Card Integrations 11](#_Toc529792972)

[5.2 GPS Integrations 11](#_Toc529792973)

[6. User Characteristics 11](#_Toc529792974)

[6.1 General Constraints 12](#_Toc529792975)

[6.2 Assumptions and Dependencies 12](#_Toc529792976)

[7. Specific Requirements 12](#_Toc529792977)

[7.1 User Requirements 12](#_Toc529792978)

[7.2 System Requirements 12](#_Toc529792979)

[7.3 Setup and Software Requirements 12](#_Toc529792980)

[7.4 Interface Requirements 12](#_Toc529792981)

[7.5 Use Case Diagram 12](#_Toc529792982)

[8. Appendices 13](#_Toc529792983)

[9. Glossary 13](#_Toc529792984)

[10. References 13](#_Toc529792985)

[11. Index 13](#_Toc529792986)

[12. Project Execution:  13](#_Toc529792987)

[13. Open Issues 13](#_Toc529792988)

# Purpose

This document provides the specification and an overview of the intended system and a description of the functions that will be implemented in the Smart Fleet system.

The purpose of the Smart Fleet Management System is to provide companies with a unified solution to fleet management, vehicle monitoring. The system consists of interfaces specific to types of user that will provide simple access to the features useful to them.

## Intended Audience

This documentation is geared towards Integration Solution Architects, Technical Designers, and Informatica conversion and interface developers. Integration Solution Architects will gain a deeper knowledge of the technology being used to extract and load data from one system to the next. With this knowledge, the ISAs will be prepared to ask better questions of the business process teams to gain additional insight to improve the quality of data transfer as well as the quality of the SID documentation. Technical designers will use this documentation to understand when to utilize various extract and load strategies, what types of data conversion database objects need to be created, and how conversions and interfaces differ as business processes as well as units of code. The code developers will use this as a guideline for standards, conventions, and best practices as well as a first resource for answering questions relevant to development.

## Scope of Work

## Business Case for the Product

The motivation to develop this project “Smart Fleet Management System” is driven by two reasons.

1. The first and the foremost reason is my strong interest to learn the popular frameworks used in the industry to develop web applications, such as Spring MVC, Spring Security, Spring REST, and Hibernate. I also wanted to hone my mobile development skills by building an Android application. The reason to stick to Java and Android is their wide usage and acceptance across the globe.
2. The second reason is to create an application/ API which can be used in multiple ways. It is a common scenario where a business manager would like to track the location of all the vehicles owned by the firm. This application could be used by many businesses like delivery trucks, taxi services and public transport. Also this API can be consumed by developers who want to develop their own applications, such as an application to track all the vehicles belonging to a group of friends who are on a trip

## Acronyms

|  |  |
| --- | --- |
| **Term / Acronym** | **Description** |
| Fleet Manager | The Fleet Manager will be the company employed by the company to keep track of all the vehicles and their use. This user will use the Fleet Management System to track vehicles and plan vehicle maintenance. |
| Fleet Management | Fleet Management refers to the applications, tools, technologies and practices that help businesses maintain optimal use of their work vehicles from a central platform |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# PROJECT DEFINITION AND REQUIREMENTS

## Product Perspective

Why have you chosen to develop this product? What need does it serve? Who are the primary stakeholders, who is developing the project, and who will benefit from the finished product?

## Data Requirements

Describe the data requirements by producing a logical data model, which consists of entity relationship diagrams, entity definitions, and attribute definitions. This is called the application data model. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database.

## Product Functions

### Vehicle Management

Vehicles are at the heart of Smart Fleet. A "vehicle" represents any asset or unit (moving or otherwise) you want to manage in Smart Fleet. Examples include:

* Power units including tractors, trucks, cars, vans and busses
* Trailers
* Construction equipment
* Boats and vessels
* Aircrafts such as jets, airplanes, cargo planes
* Stationary engines and generators
* Overhead cranes, both indoor and outdoor
* Material handling equipment including forklifts, pallet trucks and piggy packers

We should be able to track a whole bunch of information about each Vehicle in Smart Fleet, including:

* Basic details such as year, make and model, VIN, license plate, color, etc.
* Vehicle specs such as dimensions, weight and performance information
* Engine and transmission details such as engine brand, cylinders, transmission type and gear information
* Wheel and tire details such as track width, wheel diameter, axle and PSI information
* Fluid details like fuel tank and oil capacity
* Custom data via Custom Fields
* Purchase information such as price, date, odometer at time of purchase, warranty info and more
* Loan and/or lease details such as monthly payment amount, start/end date, amount financed and interest rate
* Fuel transactions (we call them Fuel Entries in Smart Fleet)
* Maintenance and service records (we call them Service Entries in Smart Fleet)
* Preventative Maintenance (PM) schedule (we call them Service Reminders in Smart Fleet)
* General repairs that do not fit in the Preventative Maintenance (PM) category (we call them Issues in Smart Fleet)
* Preventative maintenance, general repairs and service records that require more robust planning and tracking (we call them Work Orders in Smart Fleet)
* Meter (e.g. odometer) readings history
* Important renewal dates such as registrations, DOT inspections, emission tests (we call them Renewal Reminders in Smart Fleet and the renewal "types" are completely customizable)
* Unlimited photos and documents. Ah, paperless is bliss ;)
* User comments and conversations

### Contacts & Users Management

### Vendor Management

A Vendor is any entity that your organization does business with. This might include vehicle or equipment service providers, purchasing and loans, retail suppliers, fuel providers and much more. This entity can be a third-party provider or an in-house group or cost center.

Smart Fleet gives you the ability to link a Vendor with a Service Entry, Fuel Entry and other records. This makes it easy to see exactly how much you are spending with each Vendor, and to generate Reports accordingly.

## Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Id** | **Function** | **Specification** | **Requirement (M)andatory or (D)esirable** |
| BR\_SF\_001 |  | The admin should be able to login. |  |
|  |  | The admin should be able to perform Vehicle registration. |  |
|  | Contact Management | Assign vehicles to drivers and operators. |  |
|  | Data Import | Smart Fleet supports the ability for users to import data from a Comma Separated Value (CSV) file. CSV files can be created from spreadsheet programs by accessing the "Download as" or "Save as" option and choosing this file type. |  |
|  | Vehicle Management | integration to VIN decode service providers to ensure accurate and consistent vehicle attributes. |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Data Import Requirements

Smart Fleet should offers Data Import for the following Record Types:

* Fuel Entries
* Meter Readings
* Parts
* Issues
* Contacts
* Contact Renewal Reminders
* Groups
* Service Entries
* Service Reminders
* Service Tasks
* Vehicles
* Vehicle Assignments
* Vehicle Renewal Reminders
* Vendors

# Quality of Service Requirements

## Data Storage

<List what should or should not be stored for historical purposes, in what format and how it is going to be used in the future and when. Do not rely on current defaults as they may change.>

## Data De-identification

<List what should or should not be de-identified. Do not rely on current defaults as they may change.>

## Data Validation

<List any Data Validation requirements, rules, checks, etc., as applicable for the project.>

## Data and User Access

<Identify the user hierarchy, who should be allowed to see and update what type of data. >

## Auditing

< How will the user access to the data and the data being modified or used for reporting be tracked, logged and reported.>

## Usability

< This section should include all of those requirements that affect usability. For example,

•specify the required training time for a normal users and a power user to become productive at particular operation. For example: The User Interfaces will be designed for usage with minimal or no user training.

•specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like. For example: The user should be able to reach the widget details after only 3 mouse clicks.

•specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards>

## Reliability

< Requirements for reliability of the system should be specified here. Some suggestions follow:

•Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

•Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system’s output.

•Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point (bugs/function-point).

•Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.>

## Performance

< The system’s performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

•response time for a transaction (average, maximum)

•throughput, for example, transactions per second

•capacity, for example, the number of customers or transactions the system can accommodate

•degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

•resource utilization, such as memory, disk, communications, etc.>

## Supportability

< This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.>

## System Availability

<Here are some examples:

The reports and software interface created as part of this project will be online / operational during normal business hours.

The maximum allowable downtime will be 3 working days outside the XXXX periods.

During the XXXX periods the maximum allowable downtime will be 8 hours.

System maintenance will be scheduled on a monthly basis and performed from Friday 5:00 p.m. through Monday at 8:00 am. Maintenance at other times can be performed with a minimum of two days advance notice to the affected parties>.

# Reporting Requirements

# Integration Requirements

Automate your data entry by integrating Smart Fleet with your fuel cards, telematics systems, and other third-party applications. Save time and effort, and never forget to enter a fuel transaction, odometer reading or other important data again!

## Fuel Card Integrations

With Fuel Card Integrations, you can automatically download fuel transactions and receive alerts for invalid odometer readings.

## GPS Integrations

Smart Fleet can integrates with several GPS and Telematics providers to sync odometer readings, import and manage Faults / Diagnostic Trouble Codes (DTC) alerts, and see a vehicle's current location.

Integration setup options include Native Integrations (those which are built in to Smart Fleet), and API Integrations (which you may establish yourself).

# User Characteristics

Who do you expect to use your finished product, and why? What is their technical background, their training or education, their motivation to use it? What obstacles might they encounter, and what specialized skills will they need?

## General Constraints

Did you work under any constraints such as platform or development environment? Did you have to make your product compatible with any existing software or other products currently in use?

## Assumptions and Dependencies

# Specific Requirements

This section of the document lists specific requirements for name of project. Requirements are divided into the following sections:

User requirements. These are requirements written from the point of view of end users, usually expressed in narrative form.

System requirements. These are detailed specifications describing the functions the system must be capable of doing.

Interface requirements. These are requirements about the user interface, which may be expressed as a list, as a narrative, or as images of screen mock-ups.

## User Requirements

List user requirements here.

## System Requirements

## Setup and Software Requirements

The following are the tools, frameworks and software’s I have used for the development of this project:

1. Spring
2. Angular
3. MySQL database
4. Etc…

## Interface Requirements

List interface requirements here; or include screen mockups. If you use mockups, be sure to explain major features or functions with narrative to avoid confusion or omission of desired features.

## Use Case Diagram

# Appendices

If you wish to append any documents, do so here. You may wish to include some or all of the following:

* Personas and scenarios developed for this project
* Transcripts of user interviews, observations, or focus groups
* Copies of communications which contain user requirements
* Original project proposals or other historical documents
* Lists of similar projects or products, with notes about how they differ from yours
* A list of requirements which were "wish-listed" or marked unfeasible at present
* Original screen mockups, if they are relevant

# Glossary

Include a glossary of definitions, acronyms, and abbreviations that might be unfamiliar to some readers, especially technical terms that may not be understood by end-users or domain-specific terms that might not be familiar to developers.

# References

List references and source documents, if any, in this section.

# Index

If your document is very large, consider compiling an index to help readers find specific items.

# Project Execution:

# Open Issues

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Issue ID** | **Issue** | **Raised By** | **Raised On** | **Solution/ Decision** | **Resolved By** | **Resolved On** | **Status** |
|  |  |  |  |  |  |  |  |