WINDSHIELD

**1. INTRODUCTION**

* 1. **PURPOSE**

The purpose of this document is to build an online system to manage vehicles to ease the vehicle management.

* 1. **INTENDED AUDIENCE AND READING SUGGESTIONS**

This project is a prototype for the vehicle management. This will be implemented by the developers of OPUS TECHNOLOGY LIMITED. This project is useful for vehicle management of any organization.

* 1. **PROJECT SCOPE**

The purpose of WINDSHEILD, vehicle management system is to ease vehicle management and to create a convenient and easy-to-use application for any organization, trying to maintain a vehicle management system. The system is based on a relational database with its car management and allocation functions. We will have a database server to manage Cars, Drivers, Maintenance etc. Above all, we hope to provide a comfortable user experience along with the best pricing available.

## 2. OVERALL DESCRIPTION

**2.1 PRODUCT PERSPECTIVE**

A distributed vehicle management system stores the following information.

* **Car Management:**

It includes car information such as Car Type, Model Number, Brand, Registration Number etc.

* **Driver Management:**

It includes driver allocation information.

* **Maintenance:**

It includes maintenance details, such as maintenance type, vendors, requisition information, purchase information etc.

* **Fuel:**

It includes vehicle fuel consumption information.

* **Operation/Requisition:**

It includes vehicle requisition information.

* **BRTA:**

It includes BRTA Dealings.

**2.2 PRODUCT FEATURES**

The major features of Windshield are shown below

**2.2 DATABASE DESIGN**

Car Management

**Car Information**

-Car Type

-Model No

-Brand

-Registration No

-Mileage

-Seat Capacity

-License No

-Color

-Engine Capacity

-Fuel Type

-Source Type

-Tax Token

-E-Token

-Chassis No

-Vehicle Status

**Rental Management**

-Car No

-Payment Mode

-Agreement Duration

-Owner Name

-Owner Address

-Owner NID

-Owner Phone No

-Chassis No

-Vehicle Status

**Spare parts Information**

-Parts Name

-Brand Name

-Model Name

**Rental Expense Head**

-ID

-Head Name

**Rental Commercial Master**

-ID

-Name

-Agreement No

**Rental Commercial Details**

-

-

-

-Organization

-Post

-Resource

**Driver Allocation**

-Allocation ID

-Resource ID

-Car ID

-Start Time

-End Time

Driver Management

Maintenance Management

**Maintenance Type**

-ID

-Type Name

**Maintenance Work**

-ID

-Master ID

-Component

**Maintenance Work Details**

-ID

-Subject

-Date

-Car No

-Requisition No

-Maintenance Type ID

**Vendor**

-Vendor ID

-Vendor Name

-Vendor Address

-Vendor Type

**Maintenance Requisition Master**

-ID

-Parent ID

-Work Name

-Work Type

**Purchase Process Master**

-ID

-Requisition ID

-Requisition Master ID

-Purchase Date

-Vendor ID

**Maintenance Requisition Details**

-ID

-Requisition Master ID

-Component ID

-Approximate Price

-Approximate Quantity

-Unit

-Specification

**Purchase Process Details**

-ID

-Requisition Details ID

-Master ID

-Price/Cost

-Quantity

-Specification

**2.3 USER CLASS and CHARACTERISTICS**

Users of WINDSHIELD, a Vehicle Management System should be able to retrieve car information, driver information, maintenance information and other necessary information from the database. The system will support two types of user privileges, User, and Admin. Users will have access to User functions, and the Admin will have access to both User and vehicle management functions.

The User should be able to do the following functions:

* Make a car requisition  
  • One-way  
  • Round-Trip  
  • Flexible Date/time  
  • Confirmation
* Cancel an existing reservation

The Admin should have following management functionalities:

* USER FUNCTIONS.  
  •   
  •   
  •   
  •   
  •
* ADMINISTRATIVE  
  • Add/Delete a vehicle  
  • Add a new airport  
  • Update vehicle.  
  •  
  •

**2.5 DESIGN and IMPLEMENTATION CONSTRAINTS**

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. Implementing the database at least using a centralized database management system.

**3. SYSTEM FEATURES**

* **DESCRIPTION and PRIORITY**

WINDSHIELD, Vehicle Management System maintains information of vehicles, allocated drivers, maintenance etc. Of course, this project has a high priority because it will be very easy to manage vehicles if the system is automated.

* **STIMULUS/RESPONSE SEQUENCES**
  + Search for available vehicles
  + Displays a detailed list of available vehicles and make a “Requisition”.
  + Cancel an existing Reservation/Requisition.
* **FUNCTIONAL REQUIREMENTS**

Other system features include:

**CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

## 4. EXTERNAL INTERFACE REQUIREMENTS

**4.1 USER INTERFACES**

* Front-end software: C#
* Back-end software: SQL

**4.2 HARDWARE INTERFACES**

* Windows.
* A browser which supports CGI, HTML & Javascript.

**4.3 SOFTWARE INTERFACES**

Following are the software used for the WINDSHEILD; Vehicle Management System.

Operating system: We have chosen Windows operating system for its best support and user-friendliness.

Database: To save Vehicle management Information we have chosen SQL database.

Platform: To implement the project we have chosen C# language for its more interactive support.

**4.4 COMMUNICATION INTERFACES**

WINDSHEILD should support all types of web browsers.

**5.** **NONFUNCTIONAL REQUIREMENTS**

**5.1 PERFORMANCE REQUIREMENTS**

The steps involved to perform the implementation of WINDSHEILD, Vehicle Management System database are as listed below

**5.1.1. E-R DIAGRAM**

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* **ENTITIES:**Which specify distinct real-world items in an application.
* **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
* **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

User

Requisition

**5.1.2. NORMALIZATION:**

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

**5.2 SAFETY REQUIREMENTS**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

**5.3 SECURITY REQUIREMENTS**

Security systems need database storage just like any other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

**5.4 SOFTWARE QUALITY ATTRIBUTES**

* **AVAILABILITY:** The Vehicle should be available on the specified date and specified time as.
* **CORRECTNESS:** The vehicle should start from correct start terminal with correct officials and should reach the correct destination.
* **MAINTAINABILITY:** The admin or in chargers should maintain correct schedules of vehicles.
* **USABILITY:** The Vehicle schedules should satisfy a maximum number of User needs.