Technical Analysis

|  |  |
| --- | --- |
| Ticket | #48 |
| Description | Technical Analysis: Introduce Assembly Plant information to the vehicle |

Scope of the Analysis

The scope of this analysis was to analyse the possibility of including the Assembly Plant information to a vehicle.

Change Description

## Initial Analysis

A car is produced by a single plant. All cars should have the assembly plant information (NON-NULL). A location can represent either a compound, plant or other types of locations. It is important to make the distinction between location types. The locations table already has a field for this (loc\_type), which specifies if the location is a plant or a compound.

## Database Changes

Although there are several strategies to implement this, the easiest one may be to create a field Assembly\_Plant of type VARCHAR(45) in the Vehicles table, which references the location id within the vehicles table.

Below is what the change would look like:

|  |  |  |  |
| --- | --- | --- | --- |
| Assembly\_Plant  | VARCHAR(45) | FK : Locations.ID | Foreign Key to Locations Table |

More information on the Assembly Plant can be looked at via the FK.

## Code Changes

Alongside changes to the database, we need to update our code to allow for this new information to be accessed.

On an initial stage, a simple addition of a String to the Vehicle, Vehicle Table and Vehicle Response classes. This will allow us to process the information on where a vehicle was produced via an FK to the locations table.

On a later stage, if the business desires, a Plant class with detailed information on a plant response would be useful to obtain the information about the plants. It would also be useful to be able to filter the plant locations from all the locations table as this would allow suppliers and external users of the API the availability of a full detailed list of every plant in the system. To prevent overengineering, joint queries can also be used to perform searches instead of adding further tables.