CarBox Database

For our project we decided to use a document-oriented database, more precisely MongoDB. This is because, besides offering good persistence and being great at handling big volumes of data, it's a tool we also had expertise in, thanks to other classes such as Complementos de Base de Dados.

Having in mind our entity relationship diagram (available in the Information Perspective area of our Project Specification document), the decided collections and structures for our database are the following:

Collection: User

This collection saves all data related to a user, including its associated vehicles.

JSON:


```
{
    "collection": "User",
    "fields": {
        "Email": {
            "type": "String",
            "description": "User's email address, used as a unique identifier."
        },
        "Password": {
            "type": "String",
            "description": "Hashed password for the user's account security."
        },
        "Name": {
            "type": "String",
            "description": "User's name."
        },
        "Phone": {
            "type": "Int",
```

```
            "description": "User's contact phone number."
        },
        "Admin": {
            "type": "Boolean",
            "description": "Flag to indicate if the user has admin privileges in the app."
        },
        "CarsList": {
            "type": "Array",
            "description": "List of cars associated with the user.",
            "items": {
                "type": "Object",
                "properties": {
                    "ECU_ID": {
                        "type": "String",
                        "description": "Unique identifier of the car's ECU."
                    },
                    "Car_Name": {
                        "type": "String",
                        "description": "Name given by user to the car, equals its license plate by default."
                    }
                }
            }
        }
    }
}
```

Collection: Car_Info

This collection stores, for each car, all its default and static information, that is always available even when the car is turned off.

JSON:


```
{
   "collection": "Car_Info",
   "fields": {
      "ECU_ID": {
         "type": "String",
         "description": "Unique identifier for the car's ECU (Eletronics Control Unit)."
      },
      "Last_Revision": {
         "type": "Date",
         "description": "Date of the last maintenance revision of the car."
      },
      "Brand": {
         "type": "String",
         "description": "Car manufacturer brand."
      },
      "L_Plate": {
         "type": "String",
         "description": "License plate of the car."
      },
      "Model": {
         "type": "String",
         "description": "Model name of the car."
```

```
},
"Year": {
    "type": "Int",
    "description": "Manufacture year of the car."
},
"Tires": {
    "type": "String",
    "description": "Type of tires used by the car."
},
"Motor": {
    "type": "String",
    "description": "Type of motor in the car."
},
"Tank": {
    "type": "String",
    "description": "Type of fuel tank."
},
"Max_Speed": {
    "type": "Float",
    "description": "Maximum speed the car can achieve."
},
"Horse_Power": {
    "type": "Int",
    "description": "Car's horsepower rating."
},
"Autonomy": {
    "type": "Float",
```

```
            "description": "Estimated autonomy range of the car in kilometers."
        }
    }
}
```


Collection: Car_Live_Info

This collection stores dynamic information that is retrieved in real-time while the car is being driven, allowing it to be saved and processed by our system to provide useful feedback to the user.

JSON:


```
{
    "collection": "Car_Live_Info",
    "fields": {
        "Car_ID": {
            "type": "String",
            "description": "Reference to the unique Car ECU ID."
        },
        "TimeStamp": {
            "type": "Date_Time",
            "description": "Timestamp of when the live data was recorded."
        },
        "Trip_ID": {
            "type": "String",
            "description": "ID of the current trip being recorded."
        },
        "Oil_Level": {
            "type": "Float",
            "description": "Current oil level in the car."
```

```
},
"Battery_Charge": {
    "type": "Float",
    "description": "Current charge level of the battery."
},
"Car_Status": {
    "type": "Boolean",
    "description": "Status indicating if the car is operational (true) or not (false)."
},
"Speed": {
    "type": "Float",
    "description": "Current speed of the car."
},
"Rotations_PM": {
    "type": "Int",
    "description": "Engine rotations per minute."
},
"Gas_Level": {
    "type": "Float",
    "description": "Current level of gas in the tank."
},
"Location": {
    "type": "String",
    "description": "Current location coordinates of the car."
},
"Motor_Temperature": {
    "type": "Float",
```

```
            "description": "Temperature of the motor in Celsius."
        },
        "ABS": {
            "type": "Boolean",
            "description": "Indicates if the Anti-lock Braking System is active. False means it is not."
        },
        "Torque": {
            "type": "Float",
            "description": "Current torque applied by the motor."
        },
        "Tire_Pressure": {
            "type": "Array",
            "description": "Array representing tire pressures for each tire.",
            "items": {
                "type": "Float"
            }
        },
        "Errors": {
            "type": "Array",
            "description": "List of error codes or messages currently active in the car.",
            "items": {
                "type": "String"
            }
        }
    }
}
```

Collection: Trip_Info

This collection saves information about each Trip having in consideration the data retrieved in real time and stored in “Car_Live_Info”. Each Trip is only saved when the car is turned off, making our system aggregate through all the “Car_Live_Info” documents related to each trip.

JSON:


```
{
    "collection": "Trip_Info",
    "fields": {
        "Trip_ID": {
            "type": "String",
            "description": "Unique identifier for the trip."
        },
        "Car_ID": {
            "type": "String",
            "description": "Reference to the unique Car ECU ID."
        },
        "Trip_Start": {
            "type": "Date_Time",
            "description": "Start time of the trip."
        },
        "Trip_End": {
            "type": "Date_Time",
            "description": "End time of the trip."
        },
        "Trip_Speeds": {
            "type": "Array",
```

```
        "description": "List of speeds recorded during the trip.",
        "items": {
            "type": "Float"
        }
    },
    "Trip_RPM": {
        "type": "Array",
        "description": "List of RPM values recorded during the trip.",
        "items": {
            "type": "Int"
        }
    },
    "Trip_MotorTemp": {
        "type": "Array",
        "description": "List of motor temperatures recorded during the trip.",
        "items": {
            "type": "Float"
        }
    },
    "Trip_Torque": {
        "type": "Array",
        "description": "List of torque values recorded during the trip.",
        "items": {
            "type": "Float"
        }
    }
}
```

```
}
```