```
//-- ***************************************
//--     Project:         HND Data Project
//--     Module:          PageVehicleCreate
//--     Author:          Paul McKillop
//--     Date created:    14 December 2018
//-- ***************************************

using System;
using System.Collections.Generic;
using System.Windows;
using System.Windows.Controls;
using RentalWPF.Models;
using RentalWPF.Data;


namespace RentalWPF
{
    /// <summary>
    /// Interaction logic for PageVehicleCreate.xaml
    /// </summary>
    public partial class PageVehicleCreate : Page
    {
        //-- Module wide variables to track current values
        #region Module data variables
        private List<Manufacturer> manufacturers = new List<Manufacturer>();
        private List<Model> models = new List<Model>();
        private int currentManufacturerId = 0;
        private int currentModelId = 0;
        private string currentVehicleType = string.Empty;
        private int currentVehicleTypeID = 1;
        private DateTime selectedDate = DateTime.Today;
        private string formattedDate = string.Empty;
        private int maxLadenWeight = 0;
        private string currentSelectedGearbox = string.Empty;
        #endregion

        //-- Initialise the page
        public PageVehicleCreate()
        {
            InitializeComponent();
            manufacturers = ManufacturerDB.ManufacturersGetAll("RentalAzure");
        }

        //-- on Page load
        private void Page_Loaded(object sender, RoutedEventArgs e)
        {
            //-- Page level events

        }

        #region Manufacturers and Models drill-down
        //-- Get values to the manufacturers combo
        private void ManufacturersCombo_Loaded(object sender, RoutedEventArgs e)
        {
            var combo = sender as ComboBox;
            combo.ItemsSource = manufacturers;
            combo.SelectedItem = 0;
        }

        //-- get current ID for Manufacturer and load Models combo based on choice
        private void ManufacturersCombo_SelectionChanged(object sender,
        SelectionChangedEventArgs e)
        {
            //-- Getting the index from the Combo
            bool parseOk = Int32.TryParse(ManufacturersCombo.SelectedValue.ToString
            (), out currentManufacturerId);
            //-- use index selected to filter models by manufacturer
            //-- populate the ModelsCombo
            models = ModelDB.ModelsGetByManufacturerID("RentalAzure",
            currentManufacturerId);
            this.ModelsCombo.ItemsSource = models;
```

```
        this.ModelsCombo.SelectedIndex = 0;
        bool parseModel = Int32.TryParse(ModelsCombo.SelectedValue.ToString(),
        out currentModelId);
    }

    //-- Harvest current Model_Id
    private void ModelsCombo_SelectionChanged(object sender,
    SelectionChangedEventArgs e)
    {
        bool parseOk = Int32.TryParse(ModelsCombo.SelectedValue.ToString(), out
        currentModelId);
        //-- DEBUG CHECK: MessageBox.Show(currentModelId.ToString());
    }

    #endregion

    #region Vehicle Type management

    //-- Set up values for vehicle types
    private List<string> VehicleTypes()
    {
        List<string> types = new List<string>
        {
            "Commercial",
            "Domestic"
        };
        return types;
    }

    //-- Load the Vehicle Types
    private void VehicleTypesCombo_Loaded(object sender, RoutedEventArgs e)
    {
        var combo = sender as ComboBox;
        combo.ItemsSource = VehicleTypes();
        combo.SelectedIndex = 0;
    }

    //-- Record type of vehicle chosen
    private void VehicleTypesCombo_SelectionChanged(object sender,
    SelectionChangedEventArgs e)
    {
        var selectedComboItem = sender as ComboBox;
        currentVehicleType = selectedComboItem.SelectedItem as string;
        if (currentVehicleType == "Commercial")
        {
            //-- hide Gearboxes and show Laden Weigh
            GearboxPanel.Visibility = Visibility.Collapsed;
            LadenWeightPanel.Visibility = Visibility.Visible;
            currentVehicleTypeID = 1;
        }
        else
        {
            //-- Hide LadenWeight and show Gearboxes
            GearboxPanel.Visibility = Visibility.Visible;
            LadenWeightPanel.Visibility = Visibility.Collapsed;
            currentVehicleTypeID = 2;
        }
    }
    #endregion


    #region Gearbox type management
    //-- Set up values for Gearbox types
    private List<string> Gearboxes()
    {
        List<string> gearboxes = new List<string>
        {
            "Manual",
            "Automatic"
        };
        return gearboxes;
```

```
        }

        //-- Load the geabox types
        private void GearboxesCombo_Loaded(object sender, RoutedEventArgs e)
        {
            var combo = sender as ComboBox;
            combo.ItemsSource = Gearboxes();
            combo.SelectedIndex = 0;
        }

        //-- set current gearbox type
        private void GearboxesCombo_SelectionChanged(object sender,
        SelectionChangedEventArgs e)
        {
            var selectedComboItem = sender as ComboBox;
            currentSelectedGearbox = selectedComboItem.SelectedItem as string;
        }

        #endregion

        //-- Get the selected date
        private void FleetDatePicker_SelectedDateChanged(object sender,
        SelectionChangedEventArgs e)
        {
            selectedDate = FleetDatePicker.DisplayDate;
            //-- work with string values of the selected date
            formattedDate = FleetDatePicker.DisplayDate.ToShortDateString();
            // -- DEBUG MessageBox.Show(formattedDate);
        }

        #region Vehicle data harvesting
        //-- get commercial data
        private VehicleCommercial HarvestCommercial()
        {
            bool parseOk = Int32.TryParse(LadenWeightTextBox.Text, out maxLadenWeight
            );

            var commercial = new VehicleCommercial
            {
                VehicleType_Id = currentVehicleTypeID,
                Model_Id = currentModelId,
                Registration = RegistrationTextBox.Text.Trim(),
                DateOnFleet = FleetDatePicker.SelectedDate.Value.Date,
                MaxLadenWeight = maxLadenWeight
            };

            return commercial;
        }

        //-- Get domestic vehicle data
        private VehicleDomestic HarvestDomestic()
        {
            var domestic = new VehicleDomestic
            {
                VehicleType_Id = currentVehicleTypeID,
                Model_Id = currentModelId,
                Registration = RegistrationTextBox.Text.Trim(),
                DateOnFleet = FleetDatePicker.SelectedDate.Value.Date,
                Gearbox = currentSelectedGearbox
            };

            return domestic;
        }
        #endregion

        //-- add the current vehicle data to the database.
        private void AddVehicleButton_Click(object sender, RoutedEventArgs e)
        {
            if (currentVehicleTypeID == 1)
            {
                var vehicleCommercial = HarvestCommercial();
```

```
                    VehicleDB.VehicleInsertCommercial(vehicleCommercial);
                    MessageBox.Show("Commercial Vehicle " + vehicleCommercial.
                    Registration + " inserted into database");
                }
                else
                {
                    var vehicleDomestic = HarvestDomestic();
                    VehicleDB.VehicleInsertDomestic(vehicleDomestic);
                    MessageBox.Show("Domestic Vehicle " + vehicleDomestic.Registration +
                    " inserted into database");
                }

                GoToVehicleHome();
            }


            //-- Clear form data
            private void ClearFormButton_Click(object sender, RoutedEventArgs e)
            {
                ClearForm();
            }

            private void ClearForm()
            {
                LadenWeightTextBox.Text = "";
                GearboxesCombo.SelectedValue = null;
                FleetDatePicker.Text = "";
                RegistrationTextBox.Text = "";
                VehicleTypesCombo.SelectedValue = null;
                ModelsCombo.SelectedValue = null;
                ManufacturersCombo.SelectedValue = null;


            }

            private void VehicleHomeButton_Click(object sender, RoutedEventArgs e)
            {
                GoToVehicleHome();
            }

            private void GoToVehicleHome()
            {
                var vehicleHome = new PageVehicle();
                this.NavigationService.Navigate(vehicleHome);
            }
        }
    }
```