Seosamh Ó Cinnéide

Creating an event driven computer program using C# Assignment A

Task B

1. Create test data and expected results to test the Update, Add, Delete, Cancel and Search buttons on the frmCars form and the Run button on the frmSearch form.
2. Prepare a test plan and test the software. Compare the actual results to the expected results keeping a log for each test which identifies any discrepancies between actual and expected results and records any amendments to correct errors. Use debugging tools to help locate and remove errors.

Test Plan and Log					
Component: frmCars.cs		Date: 2015/08/06	Tester: Seosamh	Version: 1.0	Page: 1/1
Test#	**Purpose/Type of Test**	**Input**	**Expected Output**	**Actual Output**	**Pass?**
1	Validate Update button	Click Update button	Changes made to the Details are saved to the database	Changes made to the Details are saved to the database	✓
2	Validate Add button	Click Add button	Details for a new blank record are shown	Details for a new blank record are shown	✓
3	Validate Delete button	Click Delete Button	The selected record is deleted	The selected record is deleted	✓
4	Validate Cancel button	Click Cancel button	Any changes made since last pressing update are undone	Any changes made since last pressing update are undone	✓
5	Validate Search button	Click Search button	A new frmSearch Window is opened	A new frmSearch Window is opened	✓

Test Plan and Log					
Component: frmSearch.cs		Date: 2015/08/06	Tester: Seosamh	Version: 1.0	Page: 1/1
Test#	**Purpose/Type of Test**	**Input**	**Expected Output**	**Actual Output**	**Pass?**
1	Validate Run button	Click Run button	The Datagrid shows the results based on the inputted search parameters or the user is asked to correct the value they inputted	The Datagrid shows the results based on the inputted search parameters or the user is asked to correct the value they inputted	✓

3. Locate the EXE file and run the executable file to demonstrate the software.
   ✓

4. Produce technical documentation to describe the connection details and the purpose of the software.

hireDataSet

This is the storage container that temporarily stores the data within the application locally. On opening the form the dataset is populated from the database.

tblCarTableAdapter

This gets the data from the database and contains a connection to the database. It contains an object that connects to Hire.mdf to retireve and resolve the information back into the database.

tblCarBindingSource

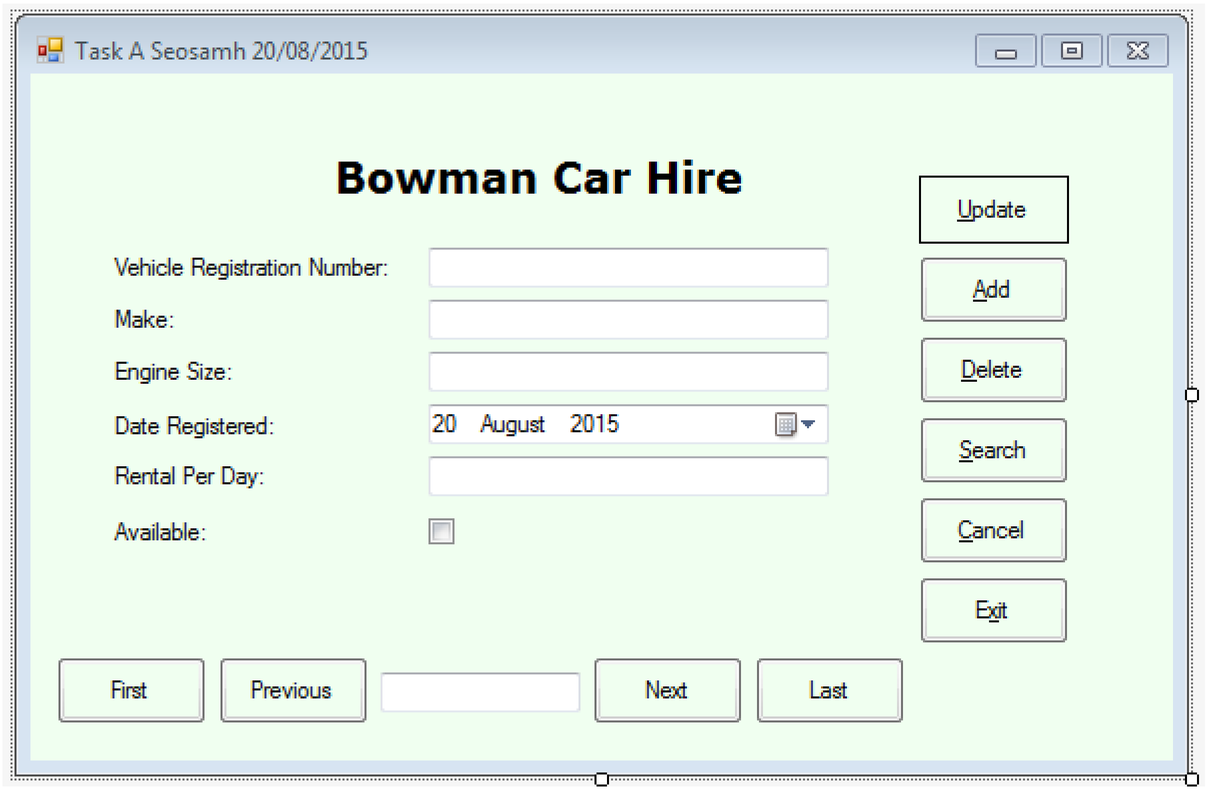
This is the bridge between the information in the dataset and the current row being displayed on the frmCars form. It keeps all the controls on the form bound to the row of data in the data set and it co-ordinates what row from the dataset should be currently displayed.

tableAdapterManager

The TableAdapterManager is a component that provides the functionality to save data in related data tables. The TableAdapterManager uses the foreign-key relationships that relate data tables to determine the correct order to send the Inserts, Updates, and Deletes from a dataset to the database without violating the foreign-key constraints (referential integrity) in the database.

5. Print a program listing and screen prints of the forms frmCars and frmSearch.

frmCars


Task A Seosamh 20/08/2015

Bowman Car Hire

Vehicle Registration Number:

Make:

Engine Size:

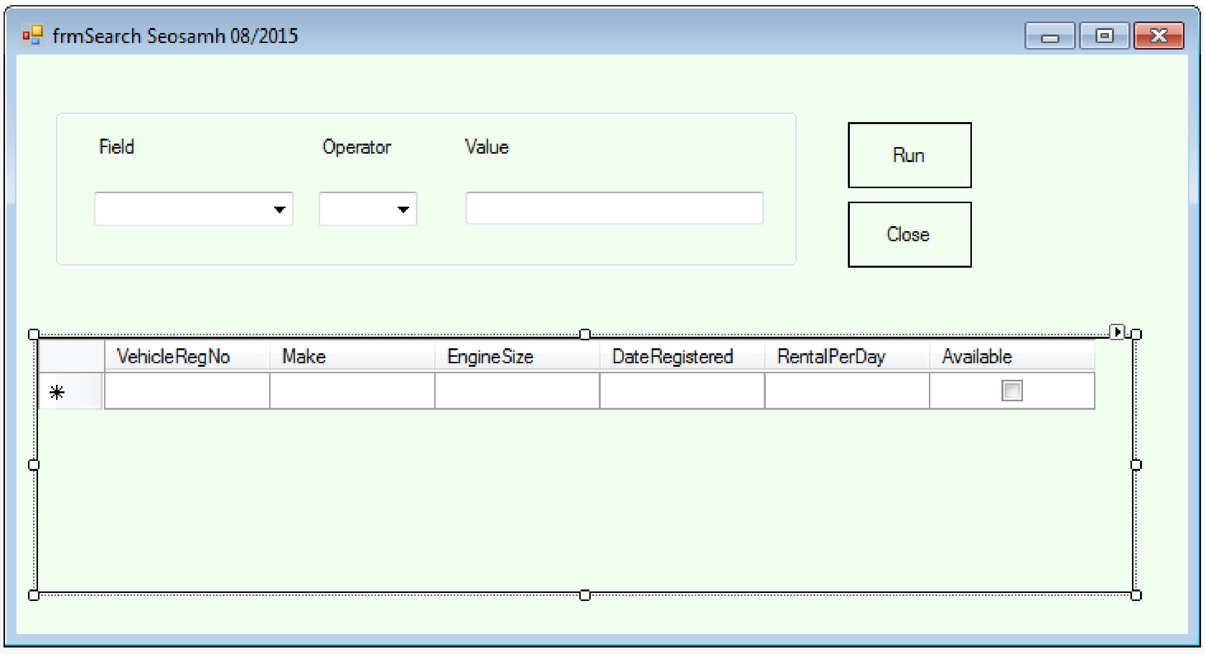
Date Registered: 20 August 2015

Rental Per Day:

Available:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EventDrivenAssignment
{
    public partial class frmCars : Form
    {
        public frmCars()
        {
            InitializeComponent();
        }
        private void frmCars_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'hireDataSet.tblCar' table. You can move, or remove it, as 
needed.
            this.tblCarTableAdapter.Fill(this.hireDataSet.tblCar);
            rentalPerDayTextBox.Text = "£" + rentalPerDayTextBox.Text;
            this.firstButton.Enabled = false;
            this.previousButton.Enabled = false;
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " + 
this.tblCarBindingSource.Count;
            this.toolTip1.SetToolTip(vehicleRegNoTextBox, "Enter the registration number of the vehicle");
            this.toolTip1.SetToolTip(makeTextBox, "Enter the make of the vehicle");
            this.toolTip1.SetToolTip(engineSizeTextBox, "Enter the engine size of the vehicle");
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void firstButton_Click(object sender, EventArgs e)
        {
            if (this.tblCarBindingSource.Position != 0)
            {
                this.tblCarBindingSource.MoveFirst();
                rentalPerDayTextBox.Text = "£" + rentalPerDayTextBox.Text;
            }
            this.nextButton.Enabled = true;
            this.lastButton.Enabled = true;
            this.firstButton.Enabled = false;
            this.previousButton.Enabled = false;
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }

        private void previousButton_Click(object sender, EventArgs e)
        {
            if (this.tblCarBindingSource.Position != 0)
            {
                this.tblCarBindingSource.MovePrevious();
                rentalPerDayTextBox.Text = "£" + rentalPerDayTextBox.Text;
            }
            else
            {
                this.firstButton.Enabled = false;
                this.previousButton.Enabled = false;
            }
            this.nextButton.Enabled = true;
            this.lastButton.Enabled = true;
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }

        private void nextButton_Click(object sender, EventArgs e)
        {
            if(this.tblCarBindingSource.Position != this.tblCarBindingSource.Count-1)
            {
                this.tblCarBindingSource.MoveNext();
                rentalPerDayTextBox.Text = "£" + rentalPerDayTextBox.Text;
            }
            else
            {
                this.nextButton.Enabled = false;
                this.lastButton.Enabled = false;
            }
            this.firstButton.Enabled = true;
            this.previousButton.Enabled = true;
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }

        private void lastButton_Click(object sender, EventArgs e)
        {
            if (this.tblCarBindingSource.Position != this.tblCarBindingSource.Count-1)
            {
                this.tblCarBindingSource.MoveLast();
                rentalPerDayTextBox.Text = "£" + rentalPerDayTextBox.Text;
            }
            this.firstButton.Enabled = true;
            this.previousButton.Enabled = true;
            this.nextButton.Enabled = false;
            this.lastButton.Enabled = false;
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }

        private void updateButton_Click(object sender, EventArgs e)
        {
            try
            {
                this.Validate();
                this.tblCarBindingSource.EndEdit();
                this.tblCarTableAdapter.Update(this.hireDataSet);
                MessageBox.Show("Update successful");
            }
            catch (System.Exception ex)
            {
                MessageBox.Show("Update failed" + ex);
            }
        }

        private void addButton_Click_1(object sender, EventArgs e)
        {
            this.tblCarBindingSource.AddNew();
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }

        private void deleteButton_Click_1(object sender, EventArgs e)
        {
            this.tblCarBindingSource.RemoveCurrent();
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }

        private void searchButton_Click_1(object sender, EventArgs e)
        {
            frmSearch form2 = new frmSearch();
            form2.Show();
        }

        private void cancelButton_Click_1(object sender, EventArgs e)
        {
            this.tblCarBindingSource.CancelEdit();
            this.positionTextBox.Text = this.tblCarBindingSource.Position + 1 + " of " +
this.tblCarBindingSource.Count;
        }
    }
}
```

frmSearch



	VehicleRegNo	Make	EngineSize	DateRegistered	RentalPerDay	Available
*						☐

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace EventDrivenAssignment
{
    public partial class frmSearch : Form
    {
        public frmSearch()
        {
            InitializeComponent();
        }

        private void tblCarBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.tblCarBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.hireDataSet);

        }

        private void frmSearch_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'hireDataSet.tblCar' table. You can move, or remove it, as
needed.
            this.tblCarTableAdapter.Fill(this.hireDataSet.tblCar);
            // TODO: This line of code loads data into the 'hireDataSet.tblCar' table. You can move, or remove it, as
needed.
            this.tblCarTableAdapter.Fill(this.hireDataSet.tblCar);

        }

        private void btnRun_Click(object sender, EventArgs e)
        {
            switch (cboField.Text)
            {
                case "Make":
                    switch(cboOperator.Text)
                    {
                        case "=":
                            this.tblCarTableAdapter.FillByEqualMake(this.hireDataSet.tblCar, valueTextBox.Text);
                            break;
                        case "<":
                            this.tblCarTableAdapter.FillByGreaterMake(this.hireDataSet.tblCar, valueTextBox.Text);
                            break;
                        case ">":
                            this.tblCarTableAdapter.FillByLessMake(this.hireDataSet.tblCar, valueTextBox.Text);
                            break;
                        case "<=":
                            this.tblCarTableAdapter.FillByGreaterEqualMake(this.hireDataSet.tblCar,
valueTextBox.Text);
                            break;
                        case ">=":
                            this.tblCarTableAdapter.FillByLessEqualMake(this.hireDataSet.tblCar, valueTextBox.Text);
                            break;
                    }
                    break;
                case "EngineSize":
                    switch(cboOperator.Text)
                    {
                        case "=":
                            this.tblCarTableAdapter.FillByEqualEngineSize(this.hireDataSet.tblCar,
valueTextBox.Text);
                            break;
                        case "<":
                            this.tblCarTableAdapter.FillByGreaterEngineSize(this.hireDataSet.tblCar,
valueTextBox.Text);
                            break;
                        case ">":
                            this.tblCarTableAdapter.FillByLessEngineSize(this.hireDataSet.tblCar, valueTextBox.Text);
                            break;
                        case "<=":
                            this.tblCarTableAdapter.FillByGreaterEqualEngineSize(this.hireDataSet.tblCar,
valueTextBox.Text);
                            break;
                        case ">=":
                            this.tblCarTableAdapter.FillByLessEqualEngineSize(this.hireDataSet.tblCar,
valueTextBox.Text);
                            break;
                    }
                    break;
                case "RentalPerDay":
                    Decimal dailyRent;
                    try
                    {
                        dailyRent = Convert.ToDecimal(valueTextBox.Text);
                    }
                    catch(FormatException)
                    {
                        MessageBox.Show("I didn't understand what you wrote for Value, try writting numbers without
any currency symbols");
                        break;
                    }
                    switch (cboOperator.Text)
                    {
                        case "=":
                            this.tblCarTableAdapter.FillByEqualRentalPerDay(this.hireDataSet.tblCar, dailyRent);
                            break;
                        case "<":
                            this.tblCarTableAdapter.FillByGreaterRentalPerDay(this.hireDataSet.tblCar, dailyRent);
                            break;
                        case ">":
                            this.tblCarTableAdapter.FillByLessRentalPerDay(this.hireDataSet.tblCar, dailyRent);
                            break;
                        case "<=":
                            this.tblCarTableAdapter.FillByGreaterEqualRentalPerDay(this.hireDataSet.tblCar,
dailyRent);

                            break;
                        case ">=":
                            this.tblCarTableAdapter.FillByLessEqualRentalPerDay(this.hireDataSet.tblCar, dailyRent );
                            break;
                    }
                    break;
                case "Available":
                    Boolean carAvailable;
                    try
                    {
                        carAvailable = Convert.ToBoolean(valueTextBox.Text);
                    }
                    catch (FormatException)
                    {
                        if (valueTextBox.Text.ToLower() == "yes" || valueTextBox.Text == "1")
                        {
                            carAvailable = true;
                        }
                        else if (valueTextBox.Text.ToLower() == "no" )
                        {
                            carAvailable = false;
                        }
                        else
                        {
                            MessageBox.Show("I didn't understand what you wrote for Value, try 'true' or 'false'");
                            break;
                        }
                    }
                    if (cboOperator.Text == "=")
                        this.tblCarTableAdapter.FillByEqualAvailable(this.hireDataSet.tblCar, carAvailable);

                    break;
            }

        }

        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void cboField_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (cboField.Text == "Available")
            {
                cboOperator.Text = "=";
                cboOperator.Enabled = false;
            }
            else
            {
                cboOperator.Enabled = true;
            }
        }

    }
}
```

6. You should check that the program produced meets the following requirement:

• The program must conform to the design specification: ✓
• The program uses the most appropriate controls and events: ✓
• Meaningful names are used for constants, variables, objects, forms and controls using a consistent naming convention: ✓