

# SENG8040

## Testing Methodologies

---

### Assignment #4: Selenium / Automation Testing – Due November 20<sup>th</sup>, 2019 @ 8:00AM – In Class

This assignment should be done **individually**. Do your own work and **do not share** your work with others. Sharing work is an academic offense and is subject to penalty. Be aware that source code and documents are automatically checked by eConestoga against every other student's work in the course. Academic Offenses will be reported to the College Registrar.

**This assignment will be demonstrated in class.** Any assignment not demonstrated in class will receive a 20% penalty.

In this assignment we will move from unit testing, and integration testing, into system testing. To this end, you will construct a prototype of a browser-based Web application.

#### Part 1 [20 Marks]:

You have been tasked with creating software that will determine the insurance rate of an individual. Your manager explains the business requirements as:

*If the driver has 3 or more accidents on record, refuse to provide them with insurance. Otherwise, provide them with insurance.*

*If the driver has no driving experience, apply a base rate of \$3000 annually. If a driver has 1 to 9 years of experience, apply a base rate of \$2000 annually. Otherwise, apply a base rate of \$1000 annually.*

*If a driver is 30 years old or older and have at least two years of driving experience, apply a rate reduction of 22% of the base rate.*

Create a decision table based on this set of business requirements.

Create a second decision table, removing items that don't matter with "don't cares".

Create a final decision table by removing redundancies.

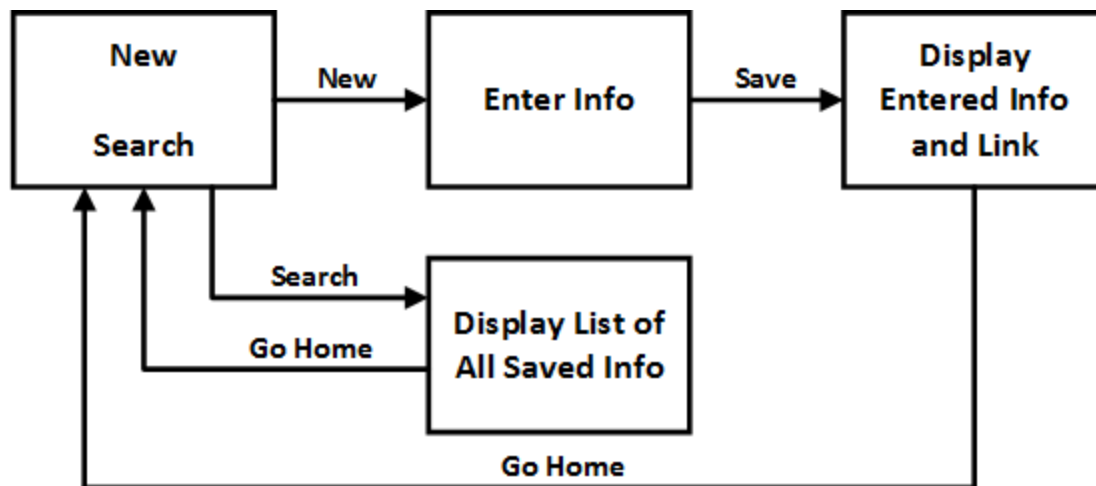
Finally, provide a list of test cases with specific input and expected output, based on the results of your reduced decision table.

## Part 2 [55 Marks]:

### Task 1: HTML/CSS/JavaScript Web Application [12 Marks]

The application allows you to enter new information, or search for saved information. The application's basic idea is to permit people to place their used vehicles up for sale by submitting their contact information and vehicle details into a standard HTML form on a Web page. Some validation of the input is necessary, which must be done using JavaScript or HTML5 (Note: You may find it easier to use Selenium to test for an error message created in JavaScript, rather than testing for HTML5 validation. Though the latter can be done). Once the information is validated, the contact information and vehicle make, model and year will be saved. After saving, the saved information will be displayed, along with the vehicle's make, model and year as a hyperlink; clicking on the hyperlink will cause the application to redirect the user to the correct JD Power and Associates page that contains information about that specific vehicle.

If the search option is selected from the home page, a list of all the previously entered cars is displayed.



The implementation details of the application are left to you; however you may not use any web application frameworks (i.e. Dreamweaver, Wordpress, etc.) to build the page/site. You must implement everything by hand-coding the HTML, CSS, and JavaScript code.

The prototype is to meet the minimal requirements of:

1. Validating the form's input fields
2. Saving the data to an HTML5 local data store on the client (or any other method of saving)
3. Displaying the previously-entered contact and vehicle information, including a link to the correct JD Power and Associates page for that specific vehicle
4. Displaying a list of previously saved data

Your application must require the following **mandatory** input fields:

- Seller name
- Address
- City
- Phone number (Acceptable formats: 123-123-1234, or (123)123-1234)
- Email address
- Vehicle make, model and year (e.g. 2012 BMW 328i, or 1987 Chrysler LeBaron)

As part of its display, your application should construct a URL to the JD Power site. These pages are always in the format:

`http://www.jdpower.com/cars/<Make>/<Model>/<year>`

For example:

<http://www.jdpower.com/cars/Ford/Mustang/2012> - be sure to try the interesting car names!

The make, model and year are saved by the application, in addition to the seller's contact information.

## **Task 2. System Tests Using Selenium with the NUnit Test Framework [40 Marks]**

Create at least five system test cases (using the Selenium Web Driver and NUnit) to test your application.

As ideas for your system tests, you may create tests to check:

- Valid contact information along with the correct make, model and year of vehicle are entered and the JD Power page for that vehicle type is displayed
- Either an error or some other behaviour is detected
- The application properly detects an invalid 10-digit telephone number
- The application requires all fields be mandatory

There is no need to test your application with anything other than Firefox as your Web client.

You will run all of your tests through the NUnit GUI console.

You will also need to host your web application on a web server such as Apache, IIS or

[XAMPP...](#)

**For this assignment you must use a form of version control.** You must use Git through Git Bash or GitHub as in Assignment 01. Provide a screenshot of at least three commits you have made. **[3 marks]**

The format for submitting the assignment is as follows:

### **1. Demonstrate your program in class.**

### **2. MS Word document containing (in this order):**

- a. Assignment Title Page with your name, student ID, "Assignment #4" in the title and date
- b. Assignment Rubric left blank (found on eConestoga)
- c. Decision tables using Excel and derived test cases
- d. Screenshots of your working application (all pages), with form fields populated
- e. Source code for all C# system tests
- f. Your HTML, CSS and / or JavaScript source code for your application
- g. A screenshot of your NUnit GUI console that shows all your tests passing and your commits from the Version Control System (Git).

### **3. Source Code (zipped):** A single compressed (.zip format) archive file containing your source code for your website (along with any instructions on how to deploy it), and test case code.

Do not include the MS Word document in the .zip file - keep them separate (-10% penalty if otherwise)