**Backbase QA Test Exercise**

**Author: Steven Price**

1. **Test approach**

Generally I would prefer to take a proactive approach for generating test scenarios, however in this example development has already began so a reactive approach needs to be taken. In this instance I will be taking a methodical approach by testing the basic CRUD functionality, followed by some edge cases that I have experienced previously. There are a few edge cases defined below that I feel should generate errors when creating data, but I have noticed that the website application allows these edge cases to be created, meaning that it is not possible to find data once the test is completed. I will mention these cases in the test case.

I have selected to use a simple step/outcome test for the test steps as this is where most of my experience has occurred. I do have some experience of creating BDD test scenarios however I feel it will demonstrate my logic better.

1. **Test cases - CRUD tests**

**Test 1: Create a new record**

Basic manual test to test the CREATE function. I have picked static data for this to make the retrieval of the easier. This only checks the basic creation process. The READ test will expand the full test range.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page <http://computer-database.herokuapp.com/computers> | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Take note of the value of how many computers have been found | Value noted. |
| 3 | Click the "Add a computer" button | Brower navigates to the "Add a computer page".  Fields to input a new computer are displayed. |
| 4 | Populate the field with "uniqueTestCaseOne" | Computer name inputted |
| 5 | Populate the field "Introduced date" with "2011-05-09" | Introduced date added |
| 6 | Populate the field "Discontinued date" with "2014-05-30" | Discontinued date added |
| 7 | Select "Lincon Laboratory" from the Company drop down. | Company selected |
| 8 | Click the "Create this computer" button | No error message is displayed to the user. Web page navigates to Home page. |
| 9 | Confirm the information banner text states "Done! Computer uniqueTestCaseOne has been created" | Banner displayed stating a record has been created successfully. |
| 10 | Confirm the number of computers displayed in the information banner | Number of computers has increased by 1. |
| 11 | Populate the "Filter by computer name" text box with “uniqueTestCaseOne” and click the "Filter by name button". Confirm the text of the information banner | One result is returned to the screen from the search. The information banner states "One computer found" |
| 12 | Confirm the following fields are populated with the correct data: Computer name: uniqueTestCaseOne Introduced date: "2011-05-09" Discontinued date: "2014-05-30" Company: Lincon Laboratory | Data returned matches the expected values |

**Test 2: Read a record**

Although there is some aspect of reading a record in the CREATE test, this test will go one step further and check the data in the Edit page. I have selected not to specify data in this test, and am relying on the tester to take note of the data under test. This also gives a change to test the pagination function of the system.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page <http://computer-database.herokuapp.com/computers> | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Using the pagination navigation buttons, navigate the application and locate a computer that has values for Introduced date, Discontinued date and Company name. Take note of all data fields. | Computer located that has data for each field and the values are noted, |
| 3 | Click on the name of the computer | Brower navigates to the "Edit computer page" |
| 4 | Confirm the computer name matches the value noted in step 2 | Computer name matches noted data |
| 5 | Confirm the "Introduced date" matches the value noted in step 2 | The value noted in step 2 is in the format DD-MMM-YYYY and the value on the Edit page will be in the format YYYY-MM-DD. Both values should convert and match |

**Test 3: Update**

Simple update record test. As per the CREATE test case, I have selected here to used fixed data for this manual test case.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page <http://computer-database.herokuapp.com/computers> | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Take note of the value of how many computers have been found | Value noted. |
| 3 | Using the pagination navigation buttons, navigate the application and locate a computer that has values for Introduced date, Discontinued date and Company. Take note of the computer name | Computer entry located. Computer name noted. |
| 4 | Click on the name of the computer | Brower navigates to the "Edit computer page" |
| 5 | Edit the Computer name to "UniqueUpdateName" | Computer name updated. |
| 6 | Populate the field "Introduced date" with "1999-12-31" | Introduced date updated |
| 7 | Populate the field "Discontinued date" with "2018-10-31" | Discontinued date updated |
| 8 | Select "IBM" from the Company drop down. | Company name updated. |
| 9 | Click the "Save this computer" button | Brower navigates back to the list of computers. |
| 10 | Confirm the number of computers displayed in the information banner | As no new computers have been added, the value must match the value obtained in step 2 |
| 11 | Confirm the information banner states "Done! Computer UniqueUpdateName has been updated" | Banner displayed stating a record has been updated successfully. |
| 12 | Populate the "Filter by computer name" text box with the value noted in step 2 and click the "Filter by name button". Confirm the text of the information banner | As this computer has had its name updated, it is no longer stored in the database. A message stating "Nothing to display" is shown and no matches are available. |
| 13 | Populate the "Filter by computer name" text box with the value UniqueUpdateName and click the "Filter by name button" | One result is returned to the screen from the search. The information banner states "One computer found". The only record displayed is for the computer record UniqueUpdateName |
| 14 | Confirm the following information is displayed: Computer: UniqueUpdateName Introduced: 31 Dec 1999 Discontinued: 31 Oct 2018 Company: IBM | Data returned matches the expected values |
| 15 | Click on the name of the computer | Edit computer page is displayed |
| 16 | Confirm the following data: Computer name: UniqueUpdateName Introduced date: 1999-12-31 Discontinued date: 2018-10-31 Company: IBM | Data returned matches the expected values |

**Test 4: Delete a record**

As stated previously, we will be using data currently in the system rather than creating new data, which gives us a chance to check pagination of the system, rather than creating a test case just for that.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page <http://computer-database.herokuapp.com/computers> | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Take note of the value of how many computers have been found | Value noted. |
| 3 | Using the pagination navigation buttons, navigate the application and locate a computer that has values for Introduced date, Discontinued date and Company. Take note of the computer name | Computer entry located. Computer name noted. |
| 4 | Click on the name of the computer | Brower navigates to the "Edit computer page" |
| 5 | Click the "Delete this computer" button | Brower returns to the computer listing page. |
| 6 | Confirm the amount of computers found | Value is one less than the value noted in step 2 |
| 7 | Confirm the information banner states "Done! Computer has been deleted" | Information banner text is correct |
| 8 | Populate the "Filter by computer name" text box with the value noted in step 2 and click the "Filter by name button" | As this computer has been deleted, it is no longer stored in the database. A message stating "Nothing to display" is shown and no matches are available. |

**3: Edge cases – Add a computer**

**Test 1: Name validation – No name supplied**

Here is the first test to check the name validation on the create computer page. Here will be checking that you cannot create a computer with a blank name, or a computer that has a space as a name.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page http://computer-database.herokuapp.com/computers | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Click the "Add a computer" button | Brower navigates to the "Add a computer page".  Fields to input a new computer are displayed. |
| 3 | Leave the name field blank and click the "Create this computer" button | Error message is displayed to the user stating that the name field is a required field. |
| 4 | Enter a space in the name field and click the "Create this computer" button | Error message remains on screen stating the name field is required |

**Test 2: Single character name**

I noticed that there is no validation when inserting a single character name. The computer appears to be created, however, since the computer count on the home screen increases, but this computer cannot be located with the filter search.

I have written the test here in the expectation it should be able to input a single character name. The test could also be written so that it would check for an error message instead.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page http://computer-database.herokuapp.com/computers | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Click the "Add a computer" button | Brower navigates to the "Add a computer page".  Fields to input a new computer are displayed. |
| 3 | Input "z" in the computer name field | Data inputted |
| 4 | Click the "Create this computer" button | No error message is displayed to the user. Web page navigates to Home page. |
| 5 | Confirm the information banner text states "Done! Computer "name added in step 3 has been created" | Banner displayed stating a record has been created successfully. |
| 6 | Populate the "Filter by computer name" text box with the value "z" and click the "Filter by name button". | Results displayed will be computer names that contain the letter "z". One entry should have the name "z" which was created earlier. |

There are other tests I would run for name validation, such as incorrect characters and maximum length name strings. Having tested these in the application I noticed there were no restrictions on these, so I have not written a test for this without knowing what the limits would be.

Looking at the long name, I can see that the system allows this, and the home screen does accommodate this, however a scroll bar is necessary.

**Test 3: Date validation**

Here are some tests related to testing the date.

|  |  |  |
| --- | --- | --- |
| **Step** | **Test step** | **Expected outcome** |
| 1 | Open a browser and navigate to the page http://computer-database.herokuapp.com/computers | Web page opens the title "Play sample application — Computer database" is displayed |
| 2 | Click the "Add a computer" button | Brower navigates to the "Add a computer page".  Fields to input a new computer are displayed |
| 3 | Input "date test" in the computer name field | Data inputted |
| 4 | Populate the field "Introduced date" with "02-04-2019" | Introduced date updated |
| 5 | Click the "Create this computer" button | The "introduced date" field is highlighted as being in an errored state |
| 6 | Clear the "introduced field. Populate the field "Discontinued date" with "06-08-2008" | Data populated |
| 7 | Click the "Create this computer" button | Error state is removed from "introduced date" field. The "Discontinued date" field is highlighted as being in an errored state |
| 8 | Clear the "Discontinued field. Populate the field "Introduced date" with "06/08/2008" | Data populated |
| 9 | Click the "Create this computer" button | Error state is removed from "Discontinued date" field. The "Introduced date" field is highlighted as being in an errored state |
| 10 | Clear the "Introduced date" field. Populate the field "Discontinued date" with "06/08/2008" | Data populated |
| 11 | Click the "Create this computer" button | Error state is removed from "Introduced date" field. The "Discontinued date" field is highlighted as being in an errored state |

1. **Edit page.**

I would duplicate all of the edge cases above and run them against the edit page to ensure the page has the same validation as the create page. The tests are basically identical, just an extra step of selecting an existing record rather than creating a new computer.

1. **Test automation.**

My background in test automation has mainly been with C#, therefore I have selected Java to create the regression scripts with, as this is closest to C#.

I have deviated from the manual tests in that the automation scripts starts off by creating a randomly generated name. This is to avoid duplication in the database, which could give incorrect results further in the automation tests.

I have used the Page Object model to create the regression scripts. Looking at the web page, I could have incorporated the Add and Edit page into the same page, as the functionality and the IDs of the fields are the same, but I have kept them separated here to show how the page object model would expand with more pages.

I have based the scripts on the CRUD functionality, and in this case I will be using the same data for each test, so I will store the data of the computer in an object when it is created. Then I will read that data back and check that the conversion of the date between the pages is operating as expected. The same computer will then be edited, and checked that the old name record has been removed. Finally the record is deleted. This covers all of the basic functionality of the website and the back end connection.

Normally for a read/edit test I would normally query the database directly for the data, but since we don’t have that access here, I will stick to the method of creating the data on the fly.

I can appreciate that the tests I have written are not stand alone, and the regression pack has to be run in order. This would not be the case if the data to set up the test could be read directly form the database for the read/edit/delete record.

I have also created a two utility classes –one for date functions and one for a random string. These would probably expand as the project grew.

I have also included one additional test in the regression pack which is based on the name validation field. I have included this one extra test as from experience I have seen page validation fail due to a javascript error, resulting in a defect going into live. Therefore I have included one test to try to prevent such an issue occurring.