
Monika Mitreva and Liyana Asenova

# Test Cases

Faculty of mathematics and informatics

**Course:** Software quality assurance

**Students:** Monika Krasimirova Mitreva, fn: 62522

Liyana Boykova Asenova, fn: 62570

# Table of contents

# 1. REGISTRATION

## 1.1. Test Information

### 1.1.1. Test type

✓ Functional Test ✗ Performance Test

### 1.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 1.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 1.2. Test Summary

### 1.2.1. Results

Total number of test cases: 3
Total number of test cases passed: 3
Total number of test cases failed: 0
Total number of bugs found: 0

## 1.3. Test Cases

### 1.3.1. Registration with correct data

Special Instructions
*NONE*

| Test Case ID | RegistrationWithValidUserInformation | |
|---|---|---|
| Description | Tests the login page | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged out. | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that there is a registration button. | There is a registration button |
| 2 | Open the registration tab. | The registration tab is loaded. |

| | | |
|---|---|---|
| 3 | Assert that the registration page is displayed. | The registration page is displayed correctly. |
| 4 | Assert that the full name field is present. | The full name field is present. |
| 5 | Enter the full name. | |
| 6 | Assert that the e-mail field is present. | The e-mail field is present. |
| 7 | Enter the e-mail. | |
| 8 | Assert the password field is present. | The password field is present. |
| 9 | Assert that the password will be masked. | The password will be masked. |
| 10 | Enter the password. | |
| 11 | Assert the re-password field is present. | The re-enter password field is present. |
| 12 | Assert that the re-enter password will be masked. | The re-enter password will be masked. |
| 13 | Re-enter the password. | |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The command echo is used several times throughout this test case, primarily for debugging and to make the test's procedure clear to other users and readers. Since this is a good practice, the command store is also used to save the registration information in variables, such as the password and email. Almost every element that the test "clicks" or "types in" in order to assert its existence uses the command assert element. Let's imagine the test case attempted to enter a value in the email box even though it wasn't there on the registration page when the test was performed. In an effort to locate the missing email field, the test would pause there. This would result in a lot of time being wasted. In order to ensure that the test is proceeding as intended and that we are not experiencing too many timeouts, the test team thought that it would be advantageous to make frequent calls to the assert element command. As a result, if a component that is essential to our test, such as the email field in the registration form, is absent, we stop the test since we are unable to evaluate the functionality of the registration process in its absence.

### 1.3.2. Registration with invalid password

Special Instructions

*NONE*

| Test Case ID | RegistrationWithInvalidUserPassword | |
|---|---|---|
| Description | Tests the registration functionality when given invalid password. | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged out. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that there is a registration button. | There is a registration button. |

| | | |
|---|---|---|
| 2 | Open the registration tab. | The registration tab is opened. |
| 3 | Assert that the registration page is displayed. | The registration tab is displayed correctly. |
| 4 | Assert that the full name field is present. | The full name field is present. |
| 5 | Enter the full name. | |
| 6 | Assert that the e-mail field is present. | The e-mail field is present. |
| 7 | Enter the e-mail. | |
| 8 | Assert the password field is present. | The password field is present. |
| 9 | Assert that the password will be masked. | The password will be masked. |
| 10 | Enter invalid symbols for the password(!). | |
| 11 | Assert that the re-enter password field is present. | The re-enter password field is present. |
| 12 | Assert that the re-enter password will be masked. | The re-enter password will be masked. |
| 13 | Re-enter the invalid symbols for password(!). | |
| 14 | Assert that an error alert is present. | An error alert is present. |
| 15 | Assert that the displayed error matches invalid password error. | The displayed error matches the invalid password error. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The command store is also used in this test to define variables for the registration information. The error message for an invalid password is, however, also stored in a variable using store. Selenium looks for an error message that needs to be displayed when the test tries to register with an invalid symbol password, and if one is, the test records the displayed error message in a variable with the command store text. The error message that should be displayed and the actual error message are then compared using the command assert. The test is deemed successful if this statement is true; otherwise, it is deemed unsuccessful.

### 1.3.3. Registration with invalid email

Special Instructions

*NONE*

| Test Case ID | RegistrationWithInvalidUserEmail | |
|---|---|---|
| Description | Tests the registration functionality when given invalid email address. | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged out. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that there is a registration button. | There is a registration button. |

| | | |
|---|---|---|
| 2 | Open the registration tab. | The registration tab is opened. |
| 3 | Assert that the registration page is displayed. | The registration tab is displayed correctly. |
| 4 | Assert that the full name field is present. | The full name field is present. |
| 5 | Enter the full name. | |
| 6 | Assert that the e-mail field is present. | The e-mail field is present. |
| 7 | Enter the invalid e-mail(test). | |
| 8 | Assert the password field is present. | The password field is present. |
| 9 | Assert that the password will be masked. | The password will be masked. |
| 10 | Enter the password. | |
| 11 | Assert that the re-enter password field is present. | The re-enter password field is present. |
| 12 | Assert that the re-enter password will be masked. | The re-enter password will be masked. |
| 13 | Re-enter the password. | |
| 14 | Assert that an error alert is present. | An error alert is present. |
| 15 | Assert that the displayed error matches invalid password error. | The displayed error matches the invalid password error. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

This test case utilizes the commands store, store text, and assert for the identical reasons as the test case regarding enrolling with an invalid password. This time, however, store text records the message indicating an invalid email address, whereas assert contrasts the displayed message with the one that should have been displayed (which was also recorded in a variable with the store command).

## 1.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| | None | None |

# 2. LOGIN

## 2.1. Test Information

### 2.1.1. Test type

✓ Functional Test ✗ Performance Test

### 2.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 2.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 2.2. Test Summary

### 2.2.1. Results

Total number of test cases: 1
Total number of test cases passed: 1
Total number of test cases failed: 0
Total number of bugs found: 0

## 2.3. Test Cases

### 2.3.1. Login

Special Instructions
*NONE*

| Test Case ID | SignIn | |
|---|---|---|
| Description | Tests the login page | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged out. | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that there is a login button. | There is a login button. |
| 2 | Open the login tab. | The login tab is loaded. |

| | | |
|---|---|---|
| 3 | Assert that the login tab is displayed. | The login tab is displayed correctly. |
| 4 | Assert that the email field is present. | The full name field is present. |
| 5 | Enter the email. | |
| 6 | Assert that the password field is present. | The password field is present. |
| 7 | Assert that the password will be masked. | The password will be masked. |
| 8 | Enter the password. | |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

With the exception of echo, which is used to debug and clarify the workflow and store, which defines variables and enhances the test's readability and usefulness, not many extra commands are used in this test due to its relative simplicity.

## 2.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| | None | None |

# 3. PROFILE SETTINGS

## 3.1. Test Information

### 3.1.1. Test type

✓ Functional Test ✗ Performance Test

### 3.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 3.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 3.2. Test Summary

### 3.2.1. Results

Total number of test cases: 4
Total number of test cases passed: 3
Total number of test cases failed: 0
Total number of bugs found: 1

## 3.3. Test Cases

### 3.3.1. Change the profile's full name

Special Instructions
*NONE*

| Test Case ID | ChangeFullNameInProfile | |
|---|---|---|
| Description | Tests the functionality of being able to change your profile's full name setting. | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged into an existing account. | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that the profile tab is present. | The profile tab is present. |
| 2 | Open the profile's settings. | The Settings.aspx is loaded. |

| | | |
|---|---|---|
| 3 | Assert that the profile's settings page is displayed. | The profile's settings page is displayed correctly. |
| 4 | Assert that the profile's data tab is present. | The profile's data tab is present. |
| 5 | Open the profile's data tab. | The Data.aspx page is loaded. |
| 6 | Compare the current page url and the profile's data page url. | The current page url matches the profile's data page url and we are in the profile's data page. |
| 7 | Assert that the full name field is present in the profile's data page. | The full name field is present. |
| 8 | Enter a new full name. | |
| 9 | Save the changes. | The changes that the test made are saved in the profile. |
| 10 | Re-open the profile's data settings. | The Data.aspx page is loaded correctly. |
| 11 | Compare the full name that was written by the test and the full name that is currently displayed. | The full name that was written by the test matches the full name that is currently displayed. |
| **Test verdict** | Passed | |

**Comments** (*commands used in addition to those generated during recording*):

Other commands used in the test include echo for debugging, store for storing the name and url for the "My profile's data." In order to compare the current test page's url with the one that was previously given in the test - the one that resembles the url to the "My profile's data" - the command execute script is also utilized. In this approach, we can correctly establish whether the opened page by the test is actually the page that we want to be displayed and tested by comparing the URLs of the two pages. We believe there are numerous ways to determine whether the correct page is displayed, and we have included a variety of them in each of our test cases so that we are not relying just on one approach to accomplish this.

### 3.3.2. Change the profile's phone number

Special Instructions

*NONE*

| **Test Case ID** | ChangePhoneNumberInProfile | |
|---|---|---|
| **Description** | Tests the functionality of being able to change your profile's phone number setting. | |
| **Applicable for** | Chrome | |
| **Initial Conditions** | The user should be logged into an existing account. | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that the profile tab is present. | The profile tab is present. |

| | | |
|---|---|---|
| 2 | Open the profile's settings. | The Settings.aspx is loaded. |
| 3 | Assert that the profile's settings page is displayed. | The profile's settings page is displayed correctly. |
| 4 | Assert that the profile's data tab is present. | The profile's data tab is present. |
| 5 | Open the profile's data tab. | The Data.aspx page is loaded. |
| 6 | Compare the current page url and the profile's data page url. | The current page url matches the profile's data page url and we are in the profile's data page. |
| 7 | Assert that the phone number field is present in the profile's data page. | The phone number field is present. |
| 8 | Enter a new phone number. | |
| 9 | Save the changes. | The changes that the test made are saved in the profile. |
| 10 | Re-open the profile's data settings. | The Data.aspx page is loaded correctly. |
| 11 | Compare the phone number that was written by the test and the phone number that is currently displayed. | The phone number that was written by the test matches the phone number that is currently displayed. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

This test case uses the same commands as the test case for altering the entire name in the profile settings, such as store and execute script. However, since the value of the displayed phone number is written in an attribute rather than an element, this test case also makes use of the command store attribute to store it in the profile's data page.

### 3.3.3. Change the profile's gender setting

Special Instructions

*NONE*

| Test Case ID | ChangeGenderInProfile | |
|---|---|---|
| Description | Tests the functionality of being able to change your profile's gender setting. | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged into an existing account. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the profile tab is present. | The profile tab is present. |
| 2 | Open the profile's settings. | The Settings.aspx is loaded. |

| | | |
|---|---|---|
| 3 | Assert that the profile's settings page is displayed. | The profile's settings page is displayed correctly. |
| 4 | Assert that the profile's data tab is present. | The profile's data tab is present. |
| 5 | Open the profile's data tab. | The Data.aspx page is loaded. |
| 6 | Compare the current page url and the profile's data page url. | The current page url matches the profile's data page url and we are in the profile's data page. |
| 7 | Assert that the gender field is present in the profile's data page. | The phone number field is present. |
| 8 | Check which gender option is selected. | |
| 9 | Select the gender option that is not selected. | The changes that the test made are saved in the profile. |
| 10 | Re-open the profile's data settings. | The non-selected gender option is now selected. |
| 11 | Check if the correct gender is now selected. | The correct gender is now selected. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

Several extra commands are utilized for this test scenario. One of them is execute script, which returns a boolean variable based on whether the radio button with the value "male" from the first gender choice is selected. Then, before making the change, the if/end command examines the boolean variable to see if the first choice (male) has already been picked. If the answer is "yes," the test enters the if clause, chooses the alternative choice (female), saves the modifications, then views the profile again to see if the second option (female) was actually saved in the profile's data (again with JavaScript code and the command assert). The test enters a second if statement if the first choice—male—has not already been picked. This time, the test first selects the male option before checking to see if it has been stored and displayed in the profile's data settings.

### 3.3.4. Change the profile's city setting

Special Instructions

*NONE*

| Test Case ID | ChangeCityInProfile | |
|---|---|---|
| Description | Tests the functionality of being able to change your profile's city setting. | |
| Applicable for | Chrome | |
| Initial Conditions | The user should be logged into an existing account. | |
| Test Step ID | Test Step Description | Expected Result |

| | | |
|---|---|---|
| 1 | Assert that the profile tab is present. | The profile tab is present. |
| 2 | Open the profile's settings. | The Settings.aspx is loaded. |
| 3 | Assert that the profile's settings page is displayed. | The profile's settings page is displayed correctly. |
| 4 | Assert that the profile's data tab is present. | The profile's data tab is present. |
| 5 | Open the profile's data tab. | The Data.aspx page is loaded. |
| 6 | Assert that the profile's data tab is displayed. | The profile's data tab is displayed correctly. |
| 7 | Change the value of the city field's drop-down menu. | The value of the city is set to the new one. |
| 8 | Save the changes. | The changes that were made are now saved. |
| 9 | Open the main profile's page. | The User.aspx page is loaded correctly. |
| 10 | Check if the correct city is displayed. | The new city that was set is now displayed in the profile's main page. |
| **Test verdict** | Failed | |

**Comments** (*commands used in addition to those generated during recording*):

In this test instance, variables that serve as the "value" of the select command are defined using the command store. The drop-down menu is managed by the select command. We may quickly change the choice (city) that is to be selected by modifying the variable's data by putting it in a variable called "value." Echo is utilized as normal to debug and explain to the readers the test case workflow.

## 3.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| ChangeCityInProfile | Not all city options that can be selected by the user are displayed as the profile's location on the main page of the account. Only a few city options are actually displayed on the main page of the account. | This bug is found in user.aspx page. |

# 4. SECTIONS

## 4.1. Test Information

### 4.1.1. Test type

✓ Functional Test ✗ Performance Test

### 4.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 4.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 4.2. Test Summary

### 4.2.1. Results

Total number of test cases: 5
Total number of test cases passed: 1
Total number of test cases failed: 4
Total number of bugs found: 4

## 4.3. Test Cases

### 4.3.1. Vouchers in the beauty section

Special Instructions
*NONE*

| Test Case ID | BeautySection | |
|---|---|---|
| Description | Tests the functionality of being able to display vouchers in the beauty category. | |
| Applicable for | Chrome | |
| Initial Conditions | None | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that the beauty category is present. | The beauty category is present. |
| 2 | Open the beauty category's tab. | The krasota-i-zdrave.aspx is loaded. |

| | | |
|---|---|---|
| 3 | Assert that the beauty category is displayed. | The beauty category is loaded correctly. |
| 4 | Assert that the beauty category has present subcategories. | The beauty category has one or more subcategories. |
| 5 | Loop through each subcategory. | Each subcategory is loaded correctly. |
| 6 | Open the first voucher from each subcategory. | The first voucher's page is loaded properly. |
| 7 | Verify that the name of the subcategory of the voucher from the voucher's own page matches the name of the subcategory of the current iteration of the loop. | The subcategory of the voucher matches the subcategory of the current iteration of the loop. |
| Test verdict | Failed | |

Comments (*commands used in addition to those generated during recording*):

This test case makes use of several extra commands. The store xpath count command is the first extra command. This command informs us of how many elements match a specific xpath expression. We can find out how many subcategories there are in the beauty category by using this command. This is helpful since it indicates how many iterations will be required as we loop through each subcategory. Execute script is another command that is employed. This command returns a "loopvariable," which is required for the iterations, using JavaScript code. Also utilized is the do/repeat if command. We can loop through each child category of the parent category using this command. The command store text contains the name of the subcategory that the current loop iteration falls under. This procedure also records the name of the selected voucher's subclass. Then, to see if those two variables are equivalent, we run the command verify. With each repetition, the command execute script increases the value of the loopvariable by 1 as well. The command assert element is used to confirm that the test actually loads the sites that it is supposed to. In some of the earlier test cases, we used it to compare the page URLs to ensure that we were on the right page, but the test team chose to combine several other ways for additional variety. In this test case, the assert element present looks for a header element that contains the keyword "Beauty" on the loaded page to verify that we have in fact accessed the "beauty" category.

### 4.3.2. Vouchers in the food section

Special Instructions

*NONE*

| | |
|---|---|
| Test Case ID | FoodSection |
| Description | Tests the functionality of being able to display vouchers in the food category. |
| Applicable for | Chrome |
| Initial Conditions | None |

| Test Step ID | Test Step Description | Expected Result |
|---|---|---|
| 1 | Assert that the food category is present. | The food category is present. |
| 2 | Open the food category's tab. | The restoranti.aspx is loaded. |
| 3 | Assert that the food category is displayed. | The food category is loaded correctly. |
| 4 | Assert that the food category has present subcategories. | The food category has one or more subcategories. |
| 5 | Loop through each subcategory. | Each subcategory is loaded correctly. |
| 6 | Open the first voucher from each subcategory. | The first voucher's page is loaded properly. |
| 7 | Verify that the name of the subcategory of the voucher from the voucher's own page matches the name of the subcategory of the current iteration of the loop. | The subcategory of the voucher matches the subcategory of the current iteration of the loop. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The commands employed in this test case are comparable to those employed in the test case for the attractiveness category. However, this category differs in that we chose to determine whether a certain subcategory (that of the present iteration) has any vouchers available. There must always be at least one voucher available for each subcategory in the system we are testing in order for that subcategory to be presented (it would be hidden). The number of vouchers per subcategory is acquired by a store xpath count command, and an if/end command determines whether that number is at least one ($vouchers > 1). If so, we try to access the voucher's page and compare the subcategory that is shown with the subcategory of the most recent iteration (we want both to match). As we previously stated, this update to the test is mostly superfluous for this particular system, but we still wanted to demonstrate how it could be done. Since all of the categories have the same functionality, the remainder of the exam is quite identical to the remaining categories (except for the travel one, which has a different interface).

### 4.3.3. Vouchers in the travel section

Special Instructions

*NONE*

| Test Case ID | TravelSection | |
|---|---|---|
| Description | Tests the functionality of being able to display vouchers in the travel category. | |
| Applicable for | Chrome | |
| Initial Conditions | None | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the travel category is present. | The travel category is present. |

| | | |
|---|---|---|
| 2 | Open the travel category's tab. | The travel.aspx is loaded. |
| 3 | Verify that the travel category is displayed. | The travel category is loaded correctly. |
| 4 | Loop through each type of place of stay (object) of the vacation. | Each place of stay subcategory is loaded correctly. |
| 5 | Select the rest of the characteristics of the vacation. | The rest of the characteristics of the vacation are selected based on the given variables. |
| 6 | Open the first voucher (if available). | The page of the first voucher is displayed. |
| 7 | Verify that the name of the voucher includes the name of the type of place of stay. | The voucher's name includes the type of the selected place of stay on the current iteration. |
| Test verdict | Failed | |

Comments (*commands used in addition to those generated during recording*):

Similar to the commands used in the remaining "sectors" test cases, the extra commands are also utilized in those circumstances. They include the do/repeat if command for iteratively looping over various options, the store xpath count for keeping track of how many options and subcategories there are, etc. But in this case, the JavaScript execute script command is used to show whether a given string contains another string. In this particular situation, that is very helpful. It is impossible to compare the subcategory of the iteration to the subcategory of the voucher's page as we did in the previous category test cases since this category has numerous subclasses with various interpretations (because there are many subcategories to choose from, not only one). Since the test cycles through the various hotels (for example), we can record the name of the voucher in a variable and then check to see if it contains the word "hotel" to confirm that the system has presented the relevant types of vouchers.

### 4.3.4. Vouchers in the vehicle section

Special Instructions

*NONE*

| Test Case ID | VehicleSection | |
|---|---|---|
| Description | Tests the functionality of being able to display vouchers in the vehicle category. | |
| Applicable for | Chrome | |
| Initial Conditions | None | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the vehicle category is present. | The vehicle category is present. |
| 2 | Open the vehicle category's tab. | The vehicle.aspx is loaded. |

| | | |
|---|---|---|
| 3 | Verify that the vehicle category is displayed. | The vehicle category is loaded correctly. |
| 4 | Loop through each type of vehicle service. | Each vehicle subcategory is loaded correctly. |
| 5 | Assert that the vehicle subcategory is present. | The vehicle subcategory is present. |
| 6 | Open the first voucher (if available). | The page of the first voucher is displayed. |
| 7 | Verify that the name of the voucher contains the name of the vehicle service subcategory. | The voucher's name contains the type of the selected vehicle service (subcategory) on the current iteration. |
| Test verdict | Failed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Before attempting to open a category's content, the assert element present, store text, and echo are frequently used to confirm if the category is there. The number of subcategories is saved using the command store xpath count. The next loop is started from 2 rather than 1 because "all subcategories" is counted as well. The command execute script is used to accomplish this. Each of the subcategories is iterated through using the variable with the value 2. For simpler debugging and follow-up, we assert the existence of the current subcategory on each iteration, record its name, and echo it. Then, after asserting the existence of a voucher, it is opened, its category is compared to that of the current iteration using store text, and the voucher is verified. The loop will keep going until its value equals the number of subcategories plus one.

### 4.3.5. Sort vouchers by location

Special Instructions

*NONE*

| Test Case ID | CouponsSortingByLocation | |
|---|---|---|
| Description | Tests the functionality of being able to correctly display vouchers for a specific location (neighborhood). | |
| Applicable for | Chrome | |
| Initial Conditions | None | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Loop through each location. | Each location filter shows vouchers that are from the current location. |
| 2 | Assert that the current location is present. | The current location is present. |
| 3 | Open the current location. | The current location's vouchers are displayed. |

| | | |
|---|---|---|
| 4 | Open the first voucher (if available). | The page of the first voucher is displayed. |
| 5 | Assert that the voucher has an address specified. | The voucher has a specified address. |
| 6 | Verify that the address of the voucher contains the name of the current location. | The voucher's address contains the name of the location of the current iteration. |
| **Test verdict** | Failed | |

**Comments** (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Together with store text and echo, assert element presence is used twice: once to verify that a location exists before attempting to filter the content by it, and once to see if the element has an address supplied. There are two uses of the command store xpath count. The first application is to save the locations' number. The second use is to record the quantity of coupons. Using the loopvariable, we cycle over each location repeatedly. We count the number of vouchers on each iteration, and if any, the first one is opened. The voucher's information is then scrutinized to see if an address was provided. If there is, JavaScript is used to execute the command execute script to see if the voucher's address contains the name of the loop's current place. The test is deemed failed if the verification is unsuccessful.

## 4.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| BeautySection | The title of the current subcategory in the tabs chain does not match the name of the subcategory, in which the voucher is found. | Subcategory name: "Други" voucher's shown category: "За децата" |
| TravelSection | The title of the category in the tabs chain after it is opened does not match the name of the category in the homepage. | Category in homepage: "421 Почивки в България" category in the subpage: "Почивки България" |
| VehicleSection | The title of the category in the tabs chain after it is opened does not match the name of the category in the homepage. | Category in homepage: "Годишен техн. преглед" category in the subpage: "Годишен преглед" |
| CouponsSortingByLocation | The addresses of some of the vouchers found through the locations filter does not match the actual location that they are shown in. | This bug is found in knyazhevo.aspx |

# 5. BLOG

## 5.1. Test Information

### 5.1.1. Test type

✓ Functional Test ✗ Performance Test

### 5.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 5.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 5.2. Test Summary

### 5.2.1. Results

Total number of test cases: 3
Total number of test cases passed: 2
Total number of test cases failed: 1
Total number of bugs found: 1

## 5.3. Test Cases

### 5.3.1. Like an article in the blog

Special Instructions
*NONE*

| Test Case ID | LikeArticleInTheBlog | |
|---|---|---|
| Description | Tests the functionality of being able to like an article and the correct number of likes after doing it. | |
| Applicable for | Chrome | |
| Initial Conditions | User must be logged in existing account before trying to like an article. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the Blog tab is present. | The Blog tab is present. |

| | | |
|---|---|---|
| 2 | Open the Blog tab. | The blog.aspx is loaded. |
| 3 | Assert that the Blog tab has at least one article. | The Blog tab has one or more articles. |
| 4 | Open the first article from the Blog. | The first article's page is loaded. |
| 5 | Verify that the article with the correct url is displayed. | The url of the displayed page matches the url of the article. |
| 6 | Store the amount of likes before pressing the like button. | The amount is stored in the variable priorLikeCount. |
| 7 | Press the like button. | Like button reacts and modifies count of likes after the click. |
| 8 | Compare the likes count before and after pressing the like button. | The difference in the count of likes before and after pressing the like button should be 1. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Three times, the assert element present is used with echo: once to verify that a Blog tab already exists before attempting to enter it; once to verify that the Blog tab contains at least one article; and once to verify that the article has a Like button at the top. The command execute script utilizes JavaScript to compare the url of the current page to the stored url after using the command store to save the article's url. By doing this, we guarantee that the article page will appear. The like count is saved twice using the command store text: once before pressing the like button and once again after. The command assert and command execute script are then used to confirm that the difference in likes between pressing the like button and before is exactly.

### 5.3.2. Dislike an article in the blog

Special Instructions

*NONE*

| Test Case ID | DisikeArticleInTheBlog | |
|---|---|---|
| Description | Tests the functionality of being able to dislike an article and the correct number of likes after doing it. | |
| Applicable for | Chrome | |
| Initial Conditions | User must be logged in existing account before trying to dislike an article and it should be liked by the user before. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the Blog tab is present. | The Blog tab is present. |
| 2 | Open the Blog tab. | The blog.aspx is loaded. |

| | | |
|---|---|---|
| 3 | Assert that the Blog tab has at least one article. | The Blog tab has one or more articles. |
| 4 | Open the first article from the Blog. | The first article's page is loaded. |
| 5 | Verify that the article with the correct url is displayed. | The url of the displayed page matches the url of the article. |
| 6 | Assert that the dislike button is present. | The dislike button is present. |
| 7 | Store the amount of likes before pressing the dislike button. | The amount is stored in the variable priorLikeCount. |
| 8 | Press the dislike button. | Dislike button reacts and modifies the count of likes after the click. |
| 9 | Compare the likes count before and after pressing the dislike button. | The difference in the count of likes before and after pressing the like button should be -1. |
| Test verdict | Failed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Three times, the assert element present is used with echo: once to verify that a Blog tab already exists before attempting to enter it; once to verify that the Blog tab contains at least one article; and once to verify that the article has a hate button on top of it. The command execute script utilizes JavaScript to compare the url of the current page to the stored url after using the command store to save the article's url. By doing this, we guarantee that the article page will appear. Before pushing the dislike button and again after pressing it, the command store text is used to save the number of likes twice. The command execute script and the command assert are then used to confirm that the difference in likes between pushing the dislike button and previously is exactly -1.

### 5.3.3. Write a comment to an article in Blog

Special Instructions

*NONE*

| Test Case ID | WriteCommentinTheBlog | |
|---|---|---|
| Description | Tests the functionality of being able to write a comment in an article in Blog tab. | |
| Applicable for | Chrome | |
| Initial Conditions | User must be logged in existing account before trying to write a comment. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the Blog tab is present. | The Blog tab is present. |
| 2 | Open the Blog tab. | The blog.aspx is loaded. |
| 3 | Open the first article from the Blog. | The first article's page is loaded. |

| | | |
|---|---|---|
| 4 | Verify that the article with the correct url is displayed. | The url of the displayed page matches the url of the article. |
| 5 | Assert that the comment button is present. | The comment button is present. |
| 6 | Add a comment to the article. | The comment is added successfully to the comments section of the article. |
| 7 | Assert that our comment is present in the article. | The comment is present and at the bottom of the comments section of the article. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Together with echo, assert element present is used three times: first, to ensure that a Blog tab already exists before attempting to add one; second, to check that the article has a "add comment" button; and third, to ensure that the comment has successfully been uploaded and has appeared in the comments section. The command store is used twice: once to save the text of the remark that will be uploaded later and once to save the article's url, which will then be compared to the current page's url using the command execute script. By doing this, we guarantee that the article page will appear. The value of our "comment" variable in the article's comment field is parsed using the command type. The remark is then uploaded, and the assertion is processed to ensure that it shows in the comment area after successfully verifying that there is a "add comment" button in the article.

## 5.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| DisikeArticleInTheBlog | Dislike button does not show up due to the fact that the page does not remember your likes correctly. if you like an article and re-open the website it is not accounted. | This bug is found in article.aspx. |

# 6. FAVOURITE LOCATIONS

## 6.1. Test Information

### 6.1.1. Test type

✓ Functional Test ✗ Performance Test

### 6.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 6.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 6.2. Test Summary

### 6.2.1. Results

Total number of test cases: 2
Total number of test cases passed: 1
Total number of test cases failed: 1
Total number of bugs found: 0

## 6.3. Test Cases

### 6.3.1. Add a location to favorites

Special Instructions
*NONE*

| Test Case ID | AddFavouriteLocation | |
|---|---|---|
| Description | Tests the functionality of being able to add a location to favorites. | |
| Applicable for | Chrome | |
| Initial Conditions | User must be logged in existing account before trying to add a location to favorites. | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that the Locations tab is present. | The Locations tab is present. |
| 2 | Open the Locations tab. | The locations.aspx is loaded. |

| | | |
|---|---|---|
| 3 | Verify that the locations page is displayed. | The header of the locations page is displayed ("Търговски обекти в София"). |
| 4 | Assert that the Locations tab has at least one location. | The Location tab has one or more articles. |
| 5 | Open the first location from the Locations tab. | The first location's page is loaded. |
| 6 | Verify that the location page is displayed. | The element for the location page's id should be present. |
| 7 | Store the second part of the url of the location. | The variable locUrl has the correct value. |
| 8 | Open my profile. | Profile section is displayed. |
| 9 | Open the favorite locations tab in my profile. | Favorite locations tab is displayed. |
| 10 | Assert a location with the same url as locUrl is present and open it. | Location with the same url is present and is displayed. |
| 11 | Names of the locations from the Locations tab and the favorite locations tab are compared. | Both location names should be the same. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Six times the assert element is present, together with echo and store text: In order to enter the Locations tab, we must first confirm that it exists. Then, we must confirm that we are in the Locations tab, confirm that there is at least one location there, confirm that the location page is displayed, confirm that the location has been added to favorites, and confirm that the location from favorites has a name. The command execute script is used twice: once to get the url of the page for the currently opened location, and once to store just the substring containing the location's name and section in a variable called locUrl. Then we access the location with the stored name in My Profile -> Favorite locations and compare both URLs from the locations page and from the favorite locations page.

### 6.3.2. Remove a location to favorites

Special Instructions

*NONE*

| | |
|---|---|
| Test Case ID | DeleteFavouriteLocation |
| Description | Tests the functionality of being able to remove a location to favorites. |
| Applicable for | Chrome |

| Initial Conditions | User must be logged in existing account and in possession of at least one favorite location before trying to delete it. | |
|---|---|---|
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the Profile tab is present. | The Profile tab is present. |
| 2 | Open the My profile tab. | The My profile tab is present. |
| 3 | Assert that the Favorite locations tab is present. | The Favorite locations tab is present. |
| 4 | Open the favorite locations tab. | The Favorite locations tab is displayed. |
| 5 | Store the name of the location to remove. | The variable "removedLoc" contains the name of the removed location. |
| 6 | Click the button for removing the location. | The prompt for location removal is displayed. |
| 7 | Click confirm on the confirmation prompt. | The location should disappear from the favorite locations list. |
| 8 | Open my profile. | Profile section is displayed. |
| 9 | Open the favorite locations tab in my profile. | Favorite locations tab is displayed. |
| 10 | Verify that the location with the same name is not present. | Location with the same name is not present. |
| Test verdict | Failed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Along with echo and store text, the assert element is used three times: once to verify the existence of the "Profile" drop-down menu before attempting to open it, once to confirm the existence of the "My profile" tab before attempting to enter it, and once to verify the existence of the "Favorite locations" tab. The automated test is interrupted on the pop up step after selecting the "remove location from favorites" button because Selenium IDE is unable to recognize or respond to the pop-up prompt for confirmation.

## 6.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| DeleteFavouriteLocation | Due to the fact that Selenium can only interact with one window, while the confirmation prompt is shown in a pop up, the automated test fails with a timeout error. | places.aspx |

# 7. SEARCH BAR

## 7.1. Test Information

### 7.1.1. Test type

✓ Functional Test ✗ Performance Test

### 7.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 7.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 7.2. Test Summary

### 7.2.1. Results

Total number of test cases: 1
Total number of test cases passed: 1
Total number of test cases failed: 0
Total number of bugs found: 0

## 7.3. Test Cases

### 7.3.1. Search for a voucher through the search bar

Special Instructions
*NONE*

| Test Case ID | CouponsSearchBar | |
|---|---|---|
| Description | Tests the functionality of the search bar and vouchers suggestions based on input. | |
| Applicable for | Chrome | |
| Initial Conditions | None | |
| **Test Step ID** | **Test Step Description** | **Expected Result** |
| 1 | Assert that the Search bar is present. | The Search bar tab is present. |

| | | |
|---|---|---|
| 2 | Type the stored string for the search. | The search bar should suggest string matches if there are any. |
| 3 | If there are any voucher suggestions based on the search, open the first one. | First voucher's information is displayed. |
| 4 | Verify that the voucher we opened contains the search string in its title. | The name of the voucher should contain the string we used as a keyword for searching. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: The assert element present is only used once to confirm the presence of a search bar. The string that will be used as a search keyword is then saved using store. The string for the search is entered using the command type. Using the command store xpath count, which stores the count of proposed vouchers in the variable "vouchers," it is possible to count the number of suggestions that pop up after inputting a word. The command store text is used in the If/end block to save the voucher's description and verify if it contains our search term. The command execute script, which uses JavaScript for the comparison, then does the comparison. The command verify then determines whether the comparison's result is true or false.

## 7.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| | None | None |

# 8. SHOPPING CART

## 8.1. Test Information

### 8.1.1. Test type

✓ Functional Test ✗ Performance Test

### 8.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 8.1.3. Test Personnel

Name: Monika Mitreva Date: 15.01.2023 Time/h: 0.5
Name: Liyana Asenova Date: 15.01.2023 Time/h: 0.5

## 8.2. Test Summary

### 8.2.1. Results

Total number of test cases: 2
Total number of test cases passed: 1
Total number of test cases failed: 1
Total number of bugs found: 0

## 8.3. Test Cases

### 8.3.1. Add a voucher to the shopping cart

Special Instructions
*NONE*

| | |
|---|---|
| Test Case ID | AddCouponInTheShoppingCart |
| Description | Tests the functionality of being able to add a voucher to the shopping cart and if the calculated total price is equivalent to the sum of the prices of the added vouchers. |
| Applicable for | Chrome |
| Initial Conditions | User must be logged in existing account before trying to add a voucher to his shopping cart. |

| Test Step ID | Test Step Description | Expected Result |
|---|---|---|
| 1 | Assert that the Vouchers tab is present. | The Voucher bar tab is present. |
| 2 | If there are any vouchers, begin the loop for opening each of them. | The vouchers.aspx is loaded. |
| 3 | Open the voucher on the current iteration. | The voucher's page is displayed. |
| 4 | Store the price of the voucher. | Variable "tempPrice" has the correct value. |
| 5 | Increment the total price and the counter of the iterations completed. | The values of the variables "loopvariable" and "totalPrice" are correct. |
| 6 | Verify that the location page is displayed. | The element for the location page's id should be present. |
| 7 | Go to step 3 if there are any vouchers left for opening. | Voucher is opened, otherwise the test proceeds with the next step. |
| 8 | Assert that the Shipping cart is present. | The Shopping cart is present. |
| 9 | Open the Shipping cart tab. | The Shopping cart page is displayed. |
| 10 | Assert a location with the same url as locUrl is present and open it. | Location with the same url is present and is displayed. |
| 11 | Store the value of the website calculated price. | The variable "sitePrice" has the correct value. |
| 12 | Compare the website total price and the one accumulated through the test. | The comparison should return the value true. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Three times the assert element is used, together with the echo and store text: first, check to see if the Vouchers tab is there, then, confirm the presence of the Shopping cart button, and last, ensure the Shopping cart tab is visible. The command execute script is used seven times: first, to declare the loop variable for iterating over the vouchers in the page, used with the store xpath count command that returns the number of vouchers in the page second, to declare the variable totalPrice that will store the total price of the vouchers in the cart calculated by the test; third, to save the voucher's price in the variable tempPrice, used with the command store text for obtaining the text; fourth, to declare the variable totalPrice; fifth: used with the command store text to obtain the price calculated from the website, format it without the non-numeric characters, and save it in the variable sitePrice; sixth: to increment the loop variable at the conclusion of each loop; seventh: to confirm that the website calculated price and test calculated price are equivalent.

### 8.3.2. Remove a voucher from the shopping cart

Special Instructions

*NONE*

| Test Case ID | RemoveCouponFromTheShoppingCart | |
|---|---|---|
| Description | Tests the functionality of being able to remove a voucher from the shopping cart. | |
| Applicable for | Chrome | |
| Initial Conditions | User must be logged in existing account before trying to remove a voucher from his shopping cart. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the Shopping cart button is present. | The Shopping cart button is present. |
| 2 | Open the Shopping cart page. | The Shopping cart page is displayed. |
| 3 | Store the count of the vouchers to remove from the shopping cart. | The value of the variable "itemsToDelete" is correct. |
| 4 | Assert there is a Delete button present. | The Delete button is present. |
| 5 | Remove a voucher from the shopping cart pressing the delete button and confirming in the confirmation prompt. | The shopping cart should appear without the deleted voucher. |
| Test verdict | Failed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: In addition to echo, the assert element present is used twice: once to verify that the "Shopping cart" button is present before attempting to open it, and once to verify that the "Delete voucher" button is present before attempting to remove a voucher from the cart. The amount of vouchers on the Shopping cart page can be found with the command store xpath count. Next, we establish a variable for iterating through each voucher using the command execute script. The automatic test is stopped when a confirmation pop-up appears because each coupon would then be deleted by clicking the delete button.

## 8.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| RemoveCouponFromTheShoppingCart | Due to the fact that selenium can only interact with one window, while the confirmation prompt is shown in a pop up, the automated test fails with a timeout error. | cart.aspx |

# 9. VIGNETTE

## 9.1. Test Information

### 9.1.1. Test type

✓ Functional Test  ✗ Performance Test

### 9.1.2. System Under Test

System name: Grabo
Version: 1.0

Short description of the system: Grabo is a website that offers discount vouchers for various goods and services.

### 9.1.3. Test Personnel

Name: Monika Mitreva  Date: 15.01.2023  Time/h: 0.5
Name: Liyana Asenova  Date: 15.01.2023  Time/h: 0.5

## 9.2. Test Summary

### 9.2.1. Results

Total number of test cases: 1
Total number of test cases passed: 1
Total number of test cases failed: 0
Total number of bugs found: 0

## 9.3. Test Cases

### 9.3.1. Check the vignette of a vehicle

Special Instructions
*NONE*

| Test Case ID | VignetteInspecting | |
|---|---|---|
| Description | Tests the functionality of being able to search for a vehicle and check for its vignette validity. | |
| Applicable for | Chrome | |
| Initial Conditions | User must be logged in existing account before trying to execute the test. | |
| Test Step ID | Test Step Description | Expected Result |
| 1 | Assert that the Vignette tab is present. | The Vignette tab is present. |

| | | |
|---|---|---|
| 2 | Assert that the Vignette page is loaded. | Vignette page is loaded. |
| 3 | Store the registration plate number of a vehicle that will be used for the test. | The variable "plateNumber" has a correct value. |
| 4 | Store the country the vehicle's registration that will be used for the test. | The variable "country" has a correct value. |
| 5 | Insert the data in the fields for registration and country. | The input should be present in the text fields of the search section. |
| 6 | Assert that the Check button is present. | The Check button is present. |
| 7 | Click the Check button and verify that the result for the vehicle vignette search is valid. | Correct vehicle details and vignette information displayed. |
| Test verdict | Passed | |

Comments (*commands used in addition to those generated during recording*):

The following extra commands are applicable to this test case: Three times, the assert element present is used: once to confirm that the Vignette tab exists, once to confirm that we are in the Vignette tab, and once to confirm that the Check button is present. The strings for the license plate and country of registration are then twice saved in the store. The text from the variable "plateNumber" is input using the command type into the field for the license plate number. The country of the car can be chosen using the drop-down menu by using the command "choose." Finally, the command assert verifies that the data provided for the vignette is accurate while the command verify element present makes sure that the results of the vignette search are displayed.

## 9.4. Traceability matrix

| Test Case ID | Bug Description | Note |
|---|---|---|
| | None | None |