**TEST SCRIPTS**

***Version History Table***

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| v1.0 | 25.10.2023 | - |
| v1.1 | 2.12.2023 | * The second test step of all test cases is revised. * Test Case 3, the fifth test step is revised. |
| v.1.2 | 16.12.2023 | New test scripts are added for the new test cases. |

**Use Case: Make a Reservation**

**1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Name** | **Test Case 1: Validate Diner can make a reservation for a single person** |  |  |
| **Use Case Tested:** | **UC1: Make Reservation** |  |  |
| **Test Description:** | This test script verifies that the system allows a diner to make a reservation for a single person. |  |  |
| **Pre-conditions** | - The reservation system is operational. - The diner has a stable internet connection. |  |  |
| **Post-conditions** | - The diner has successfully completed a reservation for a single person. - The reservation information, including the restaurant location, date, time, and the number of guests (1), is stored in the restaurant's reservation system. - A unique reservation ID is generated and associated with the diner's reservation. - The diner receives confirmation, sent to their email or phone number, with the reservation details. - The reservation system is updated with the new reservation, making the reserved table or dining area unavailable to other diners during the specified date and time. |  |  |
| **Notes:** | - The test script uses the following test data: - Restaurant: [Restaurant name] - Diner name: [Diner name] - Diner email: [Diner email] - Date: [Date] - Time: [Time] - Number of guests: 1 |  |  |
| **Result** | Passed |  |  |

|  |  |
| --- | --- |
| **TEST STEP** | **EXPECTED TEST RESULTS** |
| 1. Navigate to the ReserveWell website. | The ReserveWell website homepage is displayed. |
| 2. Scroll down to see the restaurants. | The restaurants are displayed. |
| 3. Select the desired restaurant from the list. | The selected restaurant is displayed. |
| 4. Enter the reservation date and time. | The reservation date and time are entered successfully. |
| 5. Select the number of guests (1). | The number of guests is selected successfully. |
| 6. Enter the diner's name, email address, and phone number. | The diner's information is entered successfully. |
| 7. Click on the "Submit" button. | The reservation confirmation page is displayed. |
| 8. Verify that the reservation details are correct. | The reservation details are displayed correctly. |
| 9. Click on the "Confirm Reservation" button. | A confirmation email is sent to the diner's email address. |

|  |  |
| --- | --- |
| **Test Data Field** | **Test Data Value** |
| Restaurant name | Restaurant 1 |
| Diner name | John Doe |
| Diner email | johndoe@email.com |
| Date | 2023-11-24 |
| Time | 18:00 |
| Number of guests | 1 |

The test log can be found in Test Logs document.

**2)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Name** | **Test Case 2: Validate Diner can make a reservation for multiple people** |  |  |
| **Use Case Tested:** | **UC1: Make Reservation** |  |  |
| **Test Description:** | This test script verifies that the system allows a diner to make a reservation for multiple people. |  |  |
| **Pre-conditions** | - The reservation system is operational. - The diner has a stable internet connection. |  |  |
| **Post-conditions** | - The diner has successfully completed a reservation for multiple people. - The reservation information, including the restaurant location, date, time, and the number of guests, is stored in the restaurant's reservation system. - A unique reservation ID is generated and associated with the diner's reservation. - The diner receives a confirmation, sent to their email or phone number, with the reservation details. - The reservation system is updated with the new reservation, making the reserved table or dining area unavailable to other diners during the specified date and time. |  |  |
| **Notes:** | - The test script uses the following test data: - Restaurant: [Restaurant name] - Diner name: [Diner name] - Diner email: [Diner email] - Date: [Date] - Time: [Time] - Number of guests: 4 |  |  |
| **Result** | Passed |  |  |

|  |  |
| --- | --- |
| **TEST STEP** | **EXPECTED TEST RESULTS** |
| 1. Navigate to the ReserveWell website. | The ReserveWell website homepage is displayed. |
| 2. Scroll down to see the restaurants. | The restaurants are displayed. |
| 3. Select the desired restaurant from the list. | The selected restaurant is displayed. |
| 4. Enter the reservation date and time. | The reservation date and time are entered successfully. |
| 5. Select the number of guests (4). | The number of guests is selected successfully. |
| 6. Enter the diner's name, email address, and phone number. | The diner's information is entered successfully. |
| 7. Click on the "Submit" button. | The reservation confirmation page is displayed. |
| 8. Verify that the reservation details are correct. | The reservation details are displayed correctly. |
| 9. Click on the "Confirm Reservation" button. | A confirmation email is sent to the diner's email address. |

|  |  |
| --- | --- |
| **Test Data Field** | **Test Data Value** |
| Restaurant name | Restaurant 2 |
| Diner name | John Doe |
| Diner email | johndoe@email.com |
| Date | 2023-11-25 |
| Time | 20:00 |
| Number of guests | 4 |

The test log can be found in Test Logs document.

**3)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Name** | **Test Case 3: Diner can make a reservation for a restaurant with a group size exceeding the maximum** |  |  |
| **Use Case Tested:** | **UC1: Make Reservation** |  |  |
| **Test Description:** | This test script ensure that the system allows a diner to make a reservation for a restaurant with a group size exceeding the maximum. |  |  |
| **Pre-conditions** | - The reservation system is operational. - The diner has a stable internet connection. - The diner has selected a restaurant with a maximum group size. - The diner has entered the number of guests (exceeding the maximum group size). |  |  |
| **Post-conditions** | - The diner has successfully completed a reservation for a restaurant with a group size exceeding the maximum, with additional charges. - The reservation information, including the restaurant location, date, time, the number of guests stored in the restaurant's reservation system. |  |  |
| **Notes:** | - The test script uses the following test data: - Restaurant: [Restaurant name] - Number of guests: 88 |  |  |
| **Result** | Passed |  |  |

|  |  |
| --- | --- |
| **TEST STEP** | **EXPECTED TEST RESULTS** |
| 1. Navigate to the ReserveWell website. | The ReserveWell website homepage is displayed. |
| 2. Scroll down to see the restaurants. | The restaurants are displayed. |
| 3. Select the desired restaurant from the list. | The selected restaurant is displayed. |
| 4. Enter the reservation date and time. | The reservation date and time are entered successfully. |
| 5. Select the number of guests (88). | The number of guests cannot be selected, the maximum number to select is 15. |

|  |  |
| --- | --- |
| **Test Data Field** | **Test Data Value** |
| Restaurant name | Restaurant 3 |
| Date | 2023-11-25 |
| Time | 16:00 |
| Number of guests | 15 |

The test log can be found in Test Logs document.

**4)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Name** | **Test Case 4:** **System can handle incorrect diner information** |  |  |
| **Use Case Tested:** | **UC1: Make Reservation** |  |  |
| **Test Description:** | This test script ensures that the system can handle incorrect diner information and prompt the diner to correct it. |  |  |
| **Pre-conditions** | - The diner enters incorrect information during the reservation process. - The reservation system detects the incorrect information. |  |  |
| **Post-conditions** | - The system prompts the diner to correct the incorrect information. - The diner can successfully update their information and complete the reservation process. |  |  |
| **Notes:** | - The test script uses the following test data: - Restaurant: [Restaurant name] - Diner name: [Diner name] - Diner email: [Diner email] - Date: [Date] - Time: [Time] - Number of guests: 4 |  |  |
| **Result** | Passed |  |  |

|  |  |
| --- | --- |
| **TEST STEP** | **EXPECTED TEST RESULTS** |
| 1. Navigate to the ReserveWell website. | The ReserveWell website homepage is displayed. |
| 2. Scroll down to see the restaurants. | The restaurants are displayed. |
| 3. Select the desired restaurant from the list. | The selected restaurant is displayed. |
| 4. Enter the reservation date and time. | The reservation date and time are entered successfully. |
| 5. Select the number of guests (4). | The number of guests is selected successfully. |
| 6. Enter the diner's name, **wrong email address with no “@”,** and phone number. | The diner's information is not accepted. |

|  |  |
| --- | --- |
| **Test Data Field** | **Test Data Value** |
| Restaurant name | Restaurant 4 |
| Diner name | John Doe |
| Diner email | johndoe.email.com |
| Date | 2023-11-25 |
| Time | 14:00 |
| Number of guests | 4 |

The test log can be found in Test Logs document.

**Use Case 1: Register as a Diner**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Test Case 1 (TC-01-01) | Test Case 2 (TC-01-02) | Test Case 3 (TC-01-03) |
| Test Name | Diner Registration Successful | Required Fields Validation | Invalid Email Address |
| Use Case Tested | UC1: Register as a Diner | UC1: Register as a Diner | UC1: Register as a Diner |
| Test Description | Verify a diner can complete the registration process. | Verify the system validates all required fields in the registration form. | Verify the system detects and rejects invalid email addresses. |
| Pre-conditions | - Reservation system operational. - Stable internet connection. | - Reservation system operational. - Registration started. | - Reservation system operational. - Registration started. |
| Post-conditions | - New diner account created. | - Missing fields highlighted. - Error message displayed. | - Error message displayed indicating invalid email format. |
| Notes | - Test data: Username, password, phone number, email address. | - Attempt registration with various missing required fields. | - Attempt registration with various invalid email formats (e.g., missing "@", invalid domain). |
| Result | Passed | Passed | Passed |

**Test Case 1 (TC-01-01)**

|  |  |
| --- | --- |
| Test Step | Expected Test Result |
| 1. Navigate to the ReserveWell website. | ReserveWell homepage displays correctly. |
| 2. Click the "Register" button. | Registration form appears. |
| 3. Enter valid username, password, phone number, and email address. | All fields populated correctly. |
| 4. Click the "Register" button again. | Registration process completes, new account created. |

**Test Case 2 (TC-01-02)**

|  |  |
| --- | --- |
| Test Step | Expected Test Result |
| 1. Leave one or more required fields blank (e.g., name, email). | - Corresponding fields highlighted in red or with clear visual cues (e.g., asterisk, border). - No error message displayed at this stage. |
| 2. Click the "Register" button. | - Error message displayed near each missing field, clearly indicating what information is required. - Error message wording is clear and concise, guiding the user to correct the issue. |
| 3. Attempt registration with various combinations of missing required fields. | - The system behaves consistently, highlighting missing fields and displaying appropriate error messages for each case. - Error messages are consistent with the missing fields and provide helpful guidance for correction. |

**Test Case 3 (TC-01-03)**

|  |  |
| --- | --- |
| Test Step | Expected Test Result |
| 1. Enter valid name, surname, password, and phone number. | - Corresponding fields populated correctly. - No error message displayed for email field at this stage. |
| 2. Enter an invalid email address (e.g., missing "@", invalid domain). | - Error message displayed near the email field, clearly indicating the address is invalid. - Error message wording is specific and informative, explaining the format issues (e.g., missing "@", invalid domain). |
| 3. Click the "Register" button. | - Registration fails, and the error message remains displayed. - User cannot proceed with registration until the email address is corrected. |
| 4. Attempt registration with various invalid email formats (e.g., missing "@", no period, invalid domain extension). | - The system consistently detects and rejects all invalid email formats. - Error messages remain accurate and consistent for different invalid formats. |

**Use Case 7: Register as a Restaurant**

**Test Case 1 (TC-07-01)**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Test Case 1 (TC-07-01) | Test Case 2 (TC-07-02) | Test Case 3 (TC-07-03) |
| Test Name | Successful Restaurant Registration | Required Fields Validation | Invalid Restaurant Information |
| Use Case Tested | UC2: Register as a Restaurant | UC2: Register as a Restaurant | UC2: Register as a Restaurant |
| Test Description | Verify a restaurant can complete the registration process and receive a confirmation email. | Verify the system validates all required fields in the registration form. | Verify the system detects and rejects invalid restaurant information. |
| Pre-conditions | - Reservation system operational. - Restaurant has valid business information (e.g., name, address, contact details). - Stable internet connection. | - Reservation system operational. - Restaurant starts registration process. - Stable internet connection. | - Reservation system operational. - Restaurant starts registration process. - Stable internet connection. |
| Post-conditions | - New restaurant account created. - Confirmation email received with login details. | - Missing fields highlighted. - Error message displayed for each missing field. | - Error message displayed indicating invalid information (e.g., incorrect address, invalid license number). |
| Notes | - Test data: Restaurant name, address, contact details, license number (if applicable). | - Attempt registration with various missing required fields (e.g., address, contact number). | - Attempt registration with various invalid information formats (e.g., incorrect address format, invalid license number). |
| Result | Passed | Passed | Passed |

|  |  |
| --- | --- |
| Test Step | Expected Test Result |
| 1. Navigate to the ReserveWell website. | ReserveWell homepage displays correctly. |
| 2. Click the "Register" button. | Registration form appears. |
| 3. Select "Restaurant" as account type. | Restaurant registration fields displayed. |
| 4. Enter valid restaurant name, address, contact details, and license number (if applicable). | All fields populated correctly. |
| 5. Click the "Register" button again. | Registration process completes, new restaurant account created. |
| 6. Verify confirmation email received with login details. | Confirmation email received within specified timeframe. |

**Similar Test Scripts for TC-07-02 and TC-07-03:**

Follow the same format as TC-07-01, modifying the test steps and expected results to reflect the specific requirements of each test case. For example, in TC-07-02, you would test various scenarios where required fields are missing, and in TC-07-03, you would test various scenarios where invalid restaurant information is entered.