**TESTING**

**1 Introduction**

Testing is the process of detecting errors. Testing performs a very special role for quantity assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

**Psychology of Testing**

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phaseis to detect the errors that may be present in the program. Hence one should notstart testing with the internal to show that a program doesn’t work. Testing is the process of executing a program with the intent of finding errors.

**8.2 Testing Objectives**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Starting formally, we can say, testing is a process of executing a program with the intent of finding error.

* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has a high probability of finding error, if it exists.
* The tests are inadequate to detect possibly present errors.
* The software more or less confirms to quality and reliable standards.

**System Testing**

Software testing is a critical element of software quantity assurance and represents the ultimate reviews of specifications, design and coding. The testing phase involves the testing of system using various test data. Preparation of test data plays a vital role in the system testing.After preparation the test data, the system under study is tested.

Those test data, errors were found and corrected following testing steps and corrections are recorded for future references. Thus a series testing is performed on the system before it is ready for implementation.

**8.3 Levels of testing**

* Unit testing
* Integrated testing
* Validation testing
* Output testing
* User Acceptance testing

**Unit Testing**

Unity testing focuses verification effort on the smallest unit of software unit of software i.e. the module. Using detailed design and the process specification testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of as a module. There are basic modules. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrate, to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors.

**Integrated Testing**

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions. Inthis project integrating all the modules, I have checked whether the integration effects working of any of the services by giving different combinations of inputs.

**Validation Testing**

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software validation testing began. Here we test the system in a manner that can be reasonable expected by the customer; the system was tested against system requirement specification.

**Output Testing**

After performing validation test the next phase is output test of the system, since no system could be useful if it does not produce the desired output in desired format. By considering the format of the report/output, output/report is generated or displayed and is tested. Here output format is considered in two ways: one is the screen and other is on printed form.

**User Acceptance Testing**

Acceptance test is performed with realistic data of the client to demonstrate that the software is working satisfactory. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized. Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once.

The testing phase is an important part of the software developed. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

**Registration Component.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition**  **To be Tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If any field in the form is empty. | Value of form fields. | Alert the user to enter all the fields and then proceed. | SUCCESSFUL |
| 2. | If first name, last name contains other than Character values. | custfname, custllname | Alert the user to enter only Characters and return to same page. | SUCCESSFUL |
| 3. | If user tries to enter date greater than the current date. | Dob | Alert the user to enter correct date. | SUCCESSFUL |
| 4. | If the username Already Exist. | username | Alert the user to enter other Username. | SUCCESSFUL |
| 5. | If the the Email Id is not in the specified format | email | Alert the user to enter proper Email ID | SUCCESSFUL |
| 6 | If password and confirm password does not match | c\_password | Alert the user to enter correct password | SUCCESSFUL |
| 7. | If Submit Button is clicked. | Value of form fields | Validate all the entries.  Execute the INSERT query.  Display the message “Customer record inserted successfully”. | SUCCESSFUL |

**Login Component.**

* **Authenticate :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition**  **To be Tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If any field in the form is empty. | Value of form fields. | Alert the user to enter all the fields and then proceed. | SUCCESSFUL |
| 2. | If the Email ID and Password does not match. | email, c\_password | Alert user that “Email ID and Password Not match” And stay in same page. | SUCCESSFUL |
| 3. | If the Email ID and Password Matched. | email, C\_password | Extract it from the DB Table.  Display the Home Page to User. | SUCCESSFUL |

**Forgot password :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition**  **To be Tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If field in the form is empty. | Email | Alert the user to enter the fields and then proceed. | SUCCESSFUL |
| 2. | If the Email id does not match. | Email | Alert user that “Email id Not match” And stay in same page. | SUCCESSFUL |
| 3. | If the Email id  Matched. | Email | Extract it from the DB Table.  Send a mail to user and Alert the user “Mail Has sent”. | SUCCESSFUL |

**Add Administrator:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Serial No.** | | **Condition**  **To be Tested** | **Test Data** | **Expected Output** | **Remarks** |
| **1** | | If any field in the form is empty | Value of form fields. | Alert the user to enter all the fields and then proceed. | SUCCESSFUL |
| **2** | | If Admin name contains other than character values | Name | Alert the user to enter only character values | SUCCESSFUL |
| 3 | | If login id less than 6 characters | Login\_id | Alert the user to enter login id more than 6 characters | SUCCESSFUL |
| 4 | | If password less than 8 characters | a\_password | Alert the user to enter password more than 8 characters | SUCCESSFUL |
| 5 | | If password and confirm password doesn’t matches | A\_password | Alert the user enter correct password | SUCCESSFUL |
| 6 | If the Email is not matching the original Format. | | Email\_id | Alerts the user to enter email in correct format | SUCCESSFUL |
| 7 | If contact no field contains only numeric values | | Contact\_no | Alerts the user to enter numeric values | SUCCESSFUL |

**Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition**  **To be Tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1. | If any field in the form is empty. | Value of form fields. | Alert the user to enter all the fields and then proceed. | SUCCESSFUL |
| 2. | If first name, last name contains other than Character values. | custfname, custllname | Alert the user to enter only Characters and return to same page. | SUCCESSFUL |
| 3. | If the user enters date greater than current date | Dob | Alert the user to enter correct date | SUCCESSFUL |
| 4. | If the Email ID Already Exist. | Email | Alert the user to enter other Email ID. | SUCCESSFUL |
| 5. | If the Email is not matching the original Format. | Email | Alert the user to enter proper Email ID. | SUCCESSFUL |
| 6 | If password and confirm password doesn’t matches | C\_password | Alert the user enter correct password | SUCCESSFUL |
| 7. | If Submit Button is clicked. | Value of form fields | Validate all the entries.  Execute the INSERT query.  Display the message “Your Account Has created”. | SUCCESSFUL |

* + **Add category**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Serial No.** | | **Condition**  **To be Tested** | **Test Data** | | **Expected Output** | **Remarks** | |
| 1 | | If any field in the form is empty. | Value of form fields. | | Alert the user to enter all the fields and then proceed. | SUCCESSFUL | |
| 2 | | If category name fIeld is empty . | Cat\_name | | Alert the user to enter the fields by displaying “Category Name should not be empty” | SUCCESSFUL | |
| 3 | If category description fIeld is empty . | | Cat\_des | Alert the user to enter the fields by displaying “Category description should not be empty” | | | SUCCESSFUL |

**Add Sub category**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Serial No.** | | **Condition**  **To be Tested** | **Test Data** | | **Expected Output** | **Remarks** | |
| 1 | | If any field in the form is empty. | Value of form fields. | | Alert the user to enter all the fields and then proceed. | SUCCESSFUL | |
| 2 | | If main category is empty | Cat\_name | | Alert the user to enter the fields by displaying “Category Name should not be empty” | SUCCESSFUL | |
| 3 | | If sub category is empty | subcategory | | Alert the user to enter the fields by displaying “Sub Category Name should not be empty” | SUCCESSFUL | |
| 4 | If sub category description is empty | | Description | Alert the user to enter the fields by displaying “Sub Category description should not be empty” | | | SUCCESSFUL |

**Add products:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Serial No.** | | **Condition**  **To be Tested** | **Test Data** | | | **Expected Output** | **Remarks** | |
| 1 | | If any field in the form is empty. | Value of form fields. | | | Alert the user to enter all the fields and then proceed. | SUCCESSFUL | |
| 2 | | If product name is emoty . | Prodname | | | Alert the user to enter the fields by displaying “Product Name should not be empty” | SUCCESSFUL | |
| 3 | | If category name fIeld is empty . | Cat\_id | | | Alert the user to enter the fields by displaying “Category Name should not be empty” | SUCCESSFUL | |
| 4 | | If sub category is empty | Subcat\_id | | | Alert the user to enter the fields by displaying “Sub Category Name should not be empty” | SUCCESSFUL | |
| 5 | | If Shop name is empty | Shop\_id | | | Alert the user to enter the fields by displaying “Shop owner Name should not be empty” | SUCCESSFUL | |
| 6 | | If quantity contains only numeric values | qty | | | Alert user to enter only numeric values | SUCCESSFUL | |
| 7 | | If price contains only numeric values | Price | | | Alert user to enter only numeric values | SUCCESSFUL | |
| 8 | If price contains only numeric values | | Discount | | Alert user to enter only numeric values | | | SUCCESSFUL |
| 9 | If warranty is empty | | Warranty | Alert the user to enter the fields by displaying “Warranty should not be empty” | | | | SUCCESSFUL |
| 10 | If stock status is empty | | Stockstatus | Alert the user to enter the fields by displaying “Select stock status” | | | | SUCCESSFUL |
| 11 | If Delivered In is empty | | Deliveredin | Alert the user to enter the fields by displaying “delivered In should not be empty” | | | | SUCCESSFUL |
| 12 | If Status is empty | | Status | Alert the user to enter the fields by displaying “Select Status” | | | | SUCCESSFUL |

**Add shopowner**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Serial No.** | | **Condition**  **To be Tested** | **Test Data** | | **Expected Output** | **Remarks** | |
| 1 | | If any field in the form is empty. | Value of form fields. | | Alert the user to enter all the fields and then proceed. | SUCCESSFUL | |
| 2 | | If company name contains only character values | Compname | | Alerts the user to enter only character values | SUCCESSFUL | |
| 3 | | If address field contains empty record | Address | | Alerts the user to enter empty field by displaying address should not be empty | SUCCESSFUL | |
| 4 | | If country contains empty value | country | | Alert the user to enter country by displaying “Select country” | SUCCESSFUL | |
| 5 | | If state contains empty value | State | | Alert the user to enter country by displaying “Select State” | SUCCESSFUL | |
| 6 | | If contact number contains numeric values | Contactno | | Alerts the user to enter only numeric values | SUCCESSFUL | |
| 7 | | If the Email is not matching the original Format. | login\_id | | Alert the user to enter proper Email ID. | SUCCESSFUL | |
| 8 | If password contains less than 8 characters | | S\_password | Alerts the user to enter password greater than 8 characters | | | SUCCESSFUL |
| 9 | If confirm password doesn’t match with password | | S\_password | Alerts the user to enter correct passwords | | | SUCCESSFUL |

**Add shipping details**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Serial No.** | **Condition**  **To be Tested** | **Test Data** | | **Expected Output** | **Remarks** | |
| 1 | If any field in the form is empty. | Value of form fields. | | Alert the user to enter all the fields and then proceed. | SUCCESSFUL | |
| 2 | If address field contains null value | Address | | Alerts the user to enter value by displaying “Address should not be null” | SUCCESSFUL | |
| 3 | If address field contains null value | Country | | Alerts the user to enter value by displaying “Select country” | SUCCESSFUL | |
| 4 | If State field contains null value | state | | Alerts the user to enter value by displaying “Select State” | SUCCESSFUL | |
| 5 | If pincode contains only numeric value and value equal to 6 | Pincode | | Alerts the user to enter only numeric and 6 letter values | SUCCESSFUL | |
| 6 | If contact number contains only numeric value and value less than or equal to 12 | | Contactno | Alerts the user to enter only numeric and less than 12 letter values | | SUCCESSFUL |

**Payment details**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No.** | **Condition**  **To be Tested** | **Test Data** | **Expected Output** | **Remarks** |
| 1 | If card type contains null value | Cardtype | Alerts the user to enter value by displaying “Select card type” | SUCCESSFUL |
| 2 | If card number contains numeric value and 16 digits | Card number | Alerts user to enter 16 digit numeric value | SUCCESSFUL |
| 3 | If CVV number 3 digit numeric value | CVV number | Alerts user to enter 3 digit numeric value | SUCCESSFUL |
| 4 | If expiry date contains year more than current year | Expirydate | Alerts the user to enter correct expiry date | SUCCESSFUL |