**Test case 1: User Interface**

|  |  |  |
| --- | --- | --- |
| **Steps** | **Test Action** | **Results** |
| Step 1 | Enter www.vtuproject2015.com/vB to check the voting ballot | Voting Ballot loaded successfully |
| Step 2 | Enter www.vtuproject2015.com/vB to check the voting ballot | Voting Server loaded successfully |
| Step 3 | Under Voting Ballot, click on Register and fill out the form | User able to register and also to upload the ID proof document |
| Step 4 | Under Voting Ballot, click on login and enter | User received one time password in an email and able to login |
| Step 5 | Under Voting Ballot, after login, click on Elections | User able to view all the scheduled elections |
| Step 6 | Under Voting Ballot, after login, click on vote now | User able to view the voting page |
| Step 7 | Under Voting Ballot, after login, click on vote now and click vote | Unable to vote, invalid/missing key |
| Step 8 | Under Voting Ballot, after login, click on vote now and click vote | User able to vote securly |
| Step 9 | Under Voting Server, after login, click on pending requests | Admin is able to take actions against pending requests |
| Step 10 | Under Voting Server, after login, click on participants | Admin is able to add or remove participants |
| Step 11 | Under Voting Server, after login, click on elections | Admin is able to create and edit the elections |
| Step 12 | Under Voting Server, after login, click on results | Admin is also able to view the results |
| Step 13 | Click on various links on the project to see if the links are broken or working | All links appears to be working fine |
| Step 14 | Click on various features on the front end interface. | All features are tested and works fine |

Table 8.1: Test cases for user interface

**8.2.2 User Input**

In User Interface the data entry is done by casting a vote and accessing various pages in the browser

**8.2.3 Error Handling**

In this system, all the errors that occurred while running the application are handled. The common errors were reading a tuple with an attribute set to null and database connection getting lost.

For Testing Top-Down design is used, a decomposition process which focuses as the flow of control, at latter strategies concern itself with code production. The first step is to study the overall aspects of the tasks at hand and break it into a number of independent modules. The second step is to break one of these modules further into independent sub modules. One of the important features is that each level the details at lower levels are hidden. So unit testing was performed first and then system testing.

**8.3 Integration Testing**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors with an interface. This testing is done with simple data and developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance.

**Steps to perform integration testing:**

Step 1: Create a Test Plan   
Step 2: Create Test Cases and Test Data   
Step 3: Once the components have been integrated execute the test cases   
Step 4: Fix the bugs if any and re test the code   
Step 5: Repeat the test cycle until the components have been successfully integrated

|  |  |
| --- | --- |
| **Name of the Test** | **Integration testing** |
| Test Plan | To check whether the system works properly when all the modules are integrated. |
| Test Data | Casting a vote |

Table 8.2: Test cases for integration testing

**8.4 System testing**

Ultimately, software is included with other system components and the set of system validation and integration tests are performed. System testing is a series of different tests whose main aim is to fully exercise the computer-based system. Although each test has a different role all work should verify that all system elements are properly integrated and formed allocated functions.

|  |  |
| --- | --- |
| **Name of the Test** | **System Testing** |
| Item being tested | Over all functioning of GUI with all functions properly linked. |
| Sample Input | Casting a vote |
| Expected Output | All the modules like login, execution, etc |
| Actual Output | Application reacts to user inputs in expected manner. |
| Remarks | Successful |

Table 8.3: Test cases for Input-Output

**8.5 Validation Testing**

At the culmination of black box testing, software is completely assembled is as a package. Interfacing errors have been uncovered and the correct and final series of tests, i.e., validation tests begins. Validation test is defined with a simple definition that validation succeeds when the software function in a manner that can be reasonably accepted by the customer.

**8.6 Output Testing**

After performing validation testing, the next step is output testing of the proposed system. Since the system cannot be useful if it does not produce the required output. Asking the user about the format in which the system is required tests the output displayed or generated by the system is required tests the output displayed or generated by the system under consideration. The output format is considered in two ways, one is on screen format and the other is printed format. The output format on the screen is found to be corrected as the format was designated in the system has according to the user needs. As for the hard copy the output comes according to the specification requested by the user. The output testing does not result in any correction in the system.

**8.7 Test Data and Output**

Taking various kind soft data plays a vital role in system testing. After preparing the test data system under study is tested using the test data. While testing, errors are again uncovered and corrected by using the above steps and corrections are also noted for future use.

**8.8 User Acceptance Testing**

User acceptance testing of the system is the key factor for the success of the system. A system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.