**CHAPTER 7**

**SYSTEM TESTING**

The testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**7.1 Types of Testing**

**7.1.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**7.1.2 Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**7.1.3 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**7.1.4 Performance Testing**

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**7.2 Testing Strategies**

Features to be tested

1. Module which enables user to register.
2. Module which enables user to login.
   1. **Test Cases**

**Table 7.3.1**: Module which enables user to register.

|  |  |
| --- | --- |
| SlNo. of test case: | 1 |
| Name of test: | Register Test |
| Item/Feature being tested | Whether system is able to register new user |
| Sample Input : | Email and Password |
| Expected output : | Detecting whether user credential is added to cloud DB |
| Actual output : | User credential is added to the cloud DB |
| Remarks : | Module is working properly |

**Table 7.3.2 :**Module which enables user to login

|  |  |
| --- | --- |
| SlNo .of test case: | 2 |
| Name of test: | Login Test |
| Item/Feature being tested: | Whether the module allows user to Login |
| Sample Input : | Registered Username and password |
| Expected output : | Allow User to Login with the credentials registered |
| Actual output : | Login Successful, enters Application. |
| Remarks : | Module is working properly |

**Table 7.3.3 :**Module which enables student to search books

|  |  |
| --- | --- |
| SlNo. of test case: | 3 |
| Name of test: | Book Search Test |
| Item/Feature being tested | Whether system is able to search books |
| Sample Input : | Book Title/Author/Edition |
| Expected output : | Detecting whether book credentials are Valid |
| Actual output : | Book credential is added to the cloud DB |
| Remarks : | Module is working properly |

**Table 7.3.4 :**Module which enables faculty to upload material.

|  |  |
| --- | --- |
| SlNo. of test case: | 4 |
| Name of test: | Material update Test. |
| Item/Feature being tested | Whether system is able to update material. |
| Sample Input : | Any pdf file/document. |
| Expected output : | Detects whether material is uploaded |
| Actual output : | material is added to the system |
| Remarks : | Module is working properly |

**Tablse 7.3.5 :**Module which enables admin to update.

|  |  |
| --- | --- |
| SlNo. of test case: | 5 |
| Name of test: | Student/staff information update Test. |
| Item/Feature being tested | Whether system is able to update student information. |
| Sample Input : | Details of new batch . |
| Expected output : | Detects whether details is uploaded |
| Actual output : | Updated details. |
| Remarks : | Module is working properly. |

**7.4 IntegrationTesting**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.