**6. SYSTEM TESTING**

Testing Software is a critical process which includes many activities, elements of software excellence assertion and represents the ultimate review of specification, design and coding, Software Testing presents a wide nature of an interesting variance for the software developers.

**Testing Objectives**

1. Testing is a series of steps which includes executing a program with various inputs and intent of finding an error from the inputs and making the developer to make corrections on error finding.
2. A good Software test case is one that has a possibility of finding an undiscovered error in the designed program.
3. A successful Software Testing is one that exposes an unknown or undiscovered error.
4. These above objectives imply a dramatic change in view port.
5. Software Testing is a series of steps but it cannot show the absence of defects and errors but it can only show various errors that are found software or program.

**6.1. TESTING METHODOLOGIES**

Any Software product can be tested in one of two ways

* **White Box Testing**

White Box Testing is also called as Open or Glass box testing. In White Box Testing, by finding the specified program or function that a software product or a software program has been designed or developed to perform or execute the test can be implemented and conducted for the demonstrates each program or function in a fully operated at the same time finding for errors in each program. It is a glass box or open test case design method that uses the wide control on structure of the procedural program and design to find and drive the test cases. The starting path testing activities is a white box testing.

* **Black Box Testing**

In Black Box testing by understanding and knowing the various program internal operation of a application or product or program, Black Box Testing can be conducted to guarantee that all gears mesh of the internal activities of the product or program or application can be tested. The process provides a internal operation to check the performance and specifications of all the internal mechanism which have been passably exercised. Black Box Testing fundamentally focuses on the functional activities and requirements of the software.

The steps involved in black box test case design are:

* Graph based testing methods
* Equivalence partitioning
* Boundary value analysis
* Comparison testing
* Graph matrices

**6.2. IMPLEMENTATION AND TESTING**

Implementation is one of the most important tasks in this is the phase in which one has to be cautions because all the efforts undertaken during the system will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Software Testing Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Software Testing Strategies

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

### Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

### Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

### Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**Performance Testing**

In software engineering, performance testing is processed to check the workload, usage of system, memory, processing, network and other system functionalities. It can also serve to investigate measure the program structure and its process activities inside the system, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

**Security Testing**

Attempts to verify the protection and security mechanisms built into the system for protecting the data, program and other integrations related to system.

**Unit Testing**

In software testing, Unit testing mainly focuses on verification effort on the smallest unit of program or software design that is also called a module. In unit testing the procedural or functional design provides a detailed description as a guide, focal the control paths are tested to uncover errors occurred in the designed software within the boundaries of the module. The unit testing of software is normally white box or open testing oriented and the series of steps can be conducted in corresponding or parallel for multiple modules or functions.

**Integration Testing**

Integration testing is another Testing for systematic technique and product module integrating which constructs the program structure and makes the data flow between the modules, while conducting Integration Testing it requires to uncover errors associated with various interfaces. The main objective is to take unit tested methods and activities to build a program structure that have been dictated by design.

**Top-Down Integration**

The next Testing process is top down integrations is an sequence approach for construction and testing of a program structure. In a Software or product or application various modules are integrated with each other by moving downward through the systematic control hierarchy between the modules, beginning with the main control or home control or index program. Various activities or modules connected to the main program are included in the structure of the system or either in the breath first or depth first manner.

**Bottom-up Integration**

The next testing method as the name suggests, which begins in construction and testing with various atomic modules of the product i.e., modules or functions at the lowest level. Because the all the functions or modules are having integration between bottom up manner in which the processing is required for the modules having connection to a given level is always available and the need for remnant is eliminated.

**Validation Testing**

The Validation Testing is integration testing for software which is completely assembled as a package. The Validation testing is the next stage in Testing Activities, which can be defined as successful testing process for the software functions in the manner reasonably expected by the customer. The validation Testing is mainly performed at the end approach of the user needs in testing the information inputed to the product and information contained in those sections are to validated through various testing approaches.The sensible prospect is defined in the software development with a requirement specification, and a document that gives de tailed information of all uservisible attributes of the software methodologies.

**6.3. TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Name** | **Test Case Description** | **Test Steps** | | | **Test Case Status** |
| **Step** | **Expected** | **Actual** |
| Logical layer server | Start the logical layer server | If the server is not started | Updated details will not be viewed here | Server is started | Fail |
| Registration | New user has to enter all details to register | If the user is not registered | User can’t perform further operation | Registration process is completed. | Fail |
| Upload file | User has to upload file into server | If we not upload any file at server | No file will be there at server | File is uploaded and it is splitted into chunks based on their size & stored at server | Fail |
| View file info | At logical layer can view the file | If files are not uploaded | We can’t view the file | We can view all the file information. | Fail |
| Down Load File | Select a file in cloud & download it | If we can’t download any file | Nothing will happens | File is downloaded and saved in D-drive | Fail |

**Table 6.3 (a):** FailureSystem Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Name** | **Test Case Description** | **Test Steps** | | | **Test Case Status** |
| **Step** | **Expected** | **Actual** |
| Logical layer server | Start the logical layer server | If the server is not started | Updated details will not be viewed here | Server is started | Pass |
| Registration | New user has to enter all details to register | If the user is not registered | User can’t perform further operation | Registration process is completed. | Pass |
| Upload file | User has to upload file into server | If we not upload any file at server | No file will be there at server | File is uploaded and it is splitted into chunks based on their size & stored at server | Pass |
| View file info | At logical layer can view the file | If files are not uploaded | We can’t view the file | We can view all the file information. | Pass |
| Down Load File | Select a file in cloud & download it | If we can’t download any file | Nothing will happens | File is downloaded and saved in D-drive | Pass |

**Table 6.3 (b):** Pass System Test Case