**TESTING**

The most important phase in developing any software is testing. Before the implementation of the package, testing has been carried out thoroughly to illuminate any bug, which may be present. The project has to be submitted for testing phase. This illustrates the correctness of the project. Means, whether it satisfying the user requirements and is it holds the requirement specifications as mentioned. It enables us to correct, if any error occur.

**5.1 SCOPE**

Software testing is a broader topic that is often referred to as Verification and Validation. Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensures that has been built in traceable to customer requirements. In other words Verification means “Are we building the product right?” Validation means “Are we building the right product?” The definition of Verification & Validation encompasses many of the activities that we have referred to as Software Quality Assurance (SQA). Validation encompasses a wide array of SQA activities that includes formal technical review, quality, and configuration audits, performance monitoring, simulation, feasibility study, documentation review, database review, algorithm analysis, development testing and installation testing.

The following are the objectives in testing

1. Testing is the process of executing the program with the intent of finding an error.
2. A good testing is one that has a high probability of finding an as yet undiscovered error.

To achieve these, it includes various testing methods. Some of them of which, our project is undergone were as follows.

**5.2 FUNCTIONAL TESTING**

Functional testing is a one where each and every functionality of the project is checked for its working. We have checked for the functionality of each and every function.

Each and every function of the server, client and device side functions are tested individually for all the possibilities of inputs for verifications of their proper working and functionality. Ex: Portinit.java and Remotectrl.java are to program files in the system side. They are individually compiled and executed for functionality testing.

**5.3 BLACK BOX TESTING**

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non- functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

The higher the level, and hence the bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested.

When black-box testing is performed in one project then the results obtained clearly specify the type of error occurrence whenever input are not given properly they are explained as below.

*Java file not present:* If java files containing java code is not present then the corresponding task cannot be done. Hence in order to perform any task corresponding java files should be present containing the code for its working.

**5.4 ALPHA TESTING**

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

For Alpha testing in our project, the better way was to conduct the practical of all sort on our project.

We checked for the bugs that can erupt in various situations and which provided insight into environmental and utilization conditions that can impact the software. We looked at code coverage test results to help devise test cases and input or configuration sets that will increase the code coverage over vital functions. We tried out with various input test cases, which generated bugs and were overcome to provide a reliable and bug free software for the general user. It was all the matter of intuition and to how far you can analyze the non-desirable errors in your bugs.

**5.5 BETA TESTING**

Beta testing comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the company. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

In our project we introduced our project to various friends and our guide for beta testing. They tried the project with various inputs and working from the remote area. We left the script running the whole day which enabled the users to have an opportunity to exercise the software and find errors. They provided valuable suggestions which we incorporated for better functionality of our software.

**5.6 UNIT TESTING**

In Unit testing every model was tested independent of the other verify that it is working properly. Unit testing focus verification efforts on the smallest unit of the software design in the model. To check, whether each model in the software works properly. So that, it gives desired outputs to the given inputs. All the validation and conditions are tested in the model level. In this project, each of the modules, server, system,module and sub-modules are unit tested and the bugs were identified and rectified.

**5.7 SYSTEM TESTING**

System testing involves putting all the modules together and checking the entire software. It is useful in checking whether with the given input. The desired output is got as a result.

System testing will be largely functional in nature. The focus is an invalid and valid cases, boundary values and special cases. After this phase of testing, the package was tested in the integration testing.

**5.8 INTEGRATION TESTING**

Integration testing is done to verify if the package has a whole, after the integration of all modules is working properly. This phase of testing is mainly concerned with finding out if variables and data are sending correctly from one module to another.

The aim of integration test is to put together the individually unit tested modules and build a program structure, while at the same time are conducted to uncover errors associated with the interfacing of the various modules. In order to conduct the above test, the active program is compiled. The Java file is executed with different inputs. This package has been tested for number of inputs (different focus). Also the package was tested for its different functions. It was found that the package prefers its function to meet the requirements.