

# CHAPTER 4

## IMPLEMENTATION & EVALUATION

---

### 4.1 Introduction

Testing is vital of success in any software. Testing is also carried in two phases. First phase is during the software engineering during the module creation. Second phase is after the completion of software. This is system testing which verifies that the whole set of programs hanged together. This chapter describes the project result or total outcome after developing the emergency ambulance system. We also describes unit testing which is a level of software testing where individual units/ components of software are tested. In this chapter, we also introduce white box testing method, test scenario and test cases. We also analysis the result by entering various input. This part shows all possible better outcomes through necessary test cases of our system. This part must proof that we have built our proposed system properly.

#### 4.1.1 Testing

Testing plays a vital role in the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Once program code has been developed, testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals, that is conducted tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

#### 4.1.2 Objectives of Testing

- 1) Testing is a process of executing a program with the intent of finding the error.
- 2) A good test case is one that has a high probability of finding on unpredictable error.
- 3) A successful test is one that provides solution for unpredictable error.

The Minimum aim of testing process is to identify all defects existing in software product. Software product testing accomplishes a variety of things, but most importantly it measures the

quality of the software that is developed. This view presupposes that there are defects in the software waiting to be discovered and this view is rarely disproved or even disputed.

### **4.1.3 Testing Plan**

Specifications of the product would be related to:

- i) Functions of the system.
- ii) Response criteria
- iii) Volume constraints (no. of users)
- iv) Stability criteria (24 hour)
- v) Database responses (flushing, cleaning)
- vi) Network criteria (network traffic)
- vii) Compatibility (Environment & Browsers)
- viii) User Interface / Friendliness criteria
- ix) Modularity (ability to easily interface)
- x) Security

### **4.1.4 Testing Strategy**

- i) As each module is developed it is tested and if found faultless is integrated in main module.
- ii) If the module is not perfect it is built again.

Each test plan item should have the following specific characteristics:

- i) It should be uniquely identifiable.
- ii) It should be unambiguous.
- iii) It should have well-defined test-data (test parameters)
- iv) It should have well-defined pass/fail criteria for each sub-item and overall-criteria for the pass/fail of the entire test itself.
- v) It should be easy to record.

- vi) It should be easy to demonstrate repeatedly
- vii) To prepare test plans.
- viii) To specify conditions for user acceptance testing.
- ix) To prepare test data for transaction path testing.
- x) To plan user training.

#### **4.1.5 Testing Modology**

To be truly robust, distributed applications require more than simple functional testing before release into production. At least one and preferably all of the following types of testing before releasing application to customers should be performed.

- a) Performance Testing
- b) Load Testing
- c) Stress Testing
- d) Endurance Testing

#### **4.1.6 Testing procedure**

The testing part forms an important aspect of any System and is vital for success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Philosophy behind testing the system is to find errors & rectify it.

The system test change is transitional one, as it represents the period during which control of the newly developed system passes from the hands of the development team to final users. It is therefore a critical point as it is the last opportunity to check the system before it is being used. The testing stage seeks to ensure following aspects of system from user point of view:

- a) Completeness
- b) Correctness
- c) Reliability

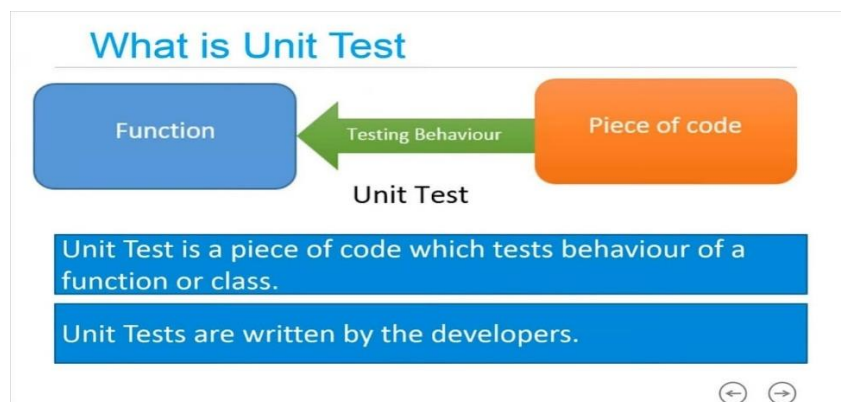
Thus, a testing plan is necessary, as it will aid to maximize the effectiveness of discovering error by early & controlled production of test plans & test specification.

## 4.2 Result Analysis

This section shows various test results just to insure that the system is working properly.

### 4.2.1 Unit Testing

Unit testing is a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. A unit test is a programmer-written test for a single piece of functionality in an application. Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. It is performed by using the white box testing method. Preparing the unit test tastes document which is complete with every possible test case, is an important task in Unit Testing activity. It gives an assurance of defect-free Unit at the end of Unit Testing stage. Below are some useful tips:



**Figure 4.1:** Unit Testing

- i. **Input values:** Write test cases for each of the identified inputs (positive & negative) accepted by the Unit.
- ii. **Expected Functionality:** To cover all functionality that is expected to be in the Unit.
- iii. **Output values:** Write test cases, which will produce all types of output values that are expected from the module / unit.
- iv. **Path coverage:** If the Unit has conditional processing those results in various paths, then write test cases to cover each of these paths.

- v. **Abnormal terminations:** Behavior of the Unit in case of abnormal termination should be tested.
- vi. **Error messages:** Check error messages / warnings. These should be short, precise and self-explanatory. They should be properly phrased and free of grammatical mistakes.
- vii. **Screen Layout:** Web page or screen layout must be tested against the requirements. Ensure that pages and screens are consistent and as per requirements.
- viii. If you are testing database application, it is important to make sure that transactions are properly designed and no way inconsistent data gets saved in the database.

### 4.2.2 White Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers. It is one of two parts of the "Box Testing" approach to software testing. Its counterpart, Black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

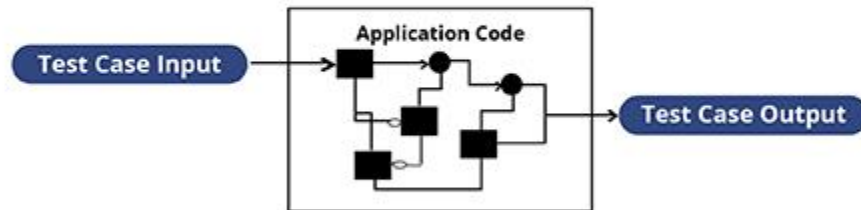
White box testing involves the testing of the software code for the following:

- i. Internal security holes
- ii. Broken or poorly structured paths in the coding processes
- iii. The flow of specific inputs through the code
- iv. Expected output
- v. The functionality of conditional loops
- vi. Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of white box testing is to verify a working flow for an application. It involves testing

a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

## WHITE BOX TESTING APPROACH



**Figure 4.2:** White box testing

### 4.2.3 Test Scenario

A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility. Test scenario consists of a detailed test procedure. A test scenario has many test cases associated with it. Before executing the test scenario, we need to think of test cases for each scenario.

Table 4.1 shows some possible test scenario.

SL No.	Test Scenario ID	Test Scenario Description
1	TS1.1	Username & Password Match
2	TS1.2	Invalid Username & Password

**Table 4.3:** Example of test scenario

### **4.3 Test Cases**

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. A test case could simply be a question that you ask of the program. The point of running the test is to gain information, for example whether the program will pass or fail the test. Test cases are the cornerstone of quality assurance where they are developed to verify the quality and behavior of a product.

### **4.4 Application Outcome**

This system is for online buy & sell system. Therefore, its main application is,

- i. Online buying & selling.
- ii. Provide property information
- iii. View property details.
- iv. User can see the property status page.
- v. Approve or cancel request by admin
- vi. View user details

### **4.5 Conclusions**

The system is built such a way that it will be useful and userfriendly to general user. Result analysis give a brief how we execute our overall system through testing, observing, analyzing. This is the proper way to know whether the system is working properly or not. It also means the system provides mainly what kinds of facilities and if it is better and reliable option for people to get online registration through our system. The application of the system also belongs in this part that clear we have created our expected online emergency ambulance system to support people with saving their time and energy [27].