# Chapter 6

# Testing

## Introduction

Testing is the process carried out on software to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications. Testing is advantageous in several ways. Firstly, the defects found help in the process of making the software reliable. Secondly, even if the defects found are not corrected, testing gives an idea as to how reliable the software is. Thirdly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

## 6.1 Testing Plan

The testing technique that is going to be used in the project is black box testing. In black box testing the expected inputs to the system are applied and only the outputs are checked.

The testing sub-process includes the following activities in a phase dependent manner:

a) Create Test Plans.
b) Create Test Specifications.
c) Review Test Plans and Test Specifications.
d) Conduct tests according to the Test Specifications, and log the defects.
e) Fix defects, if any.
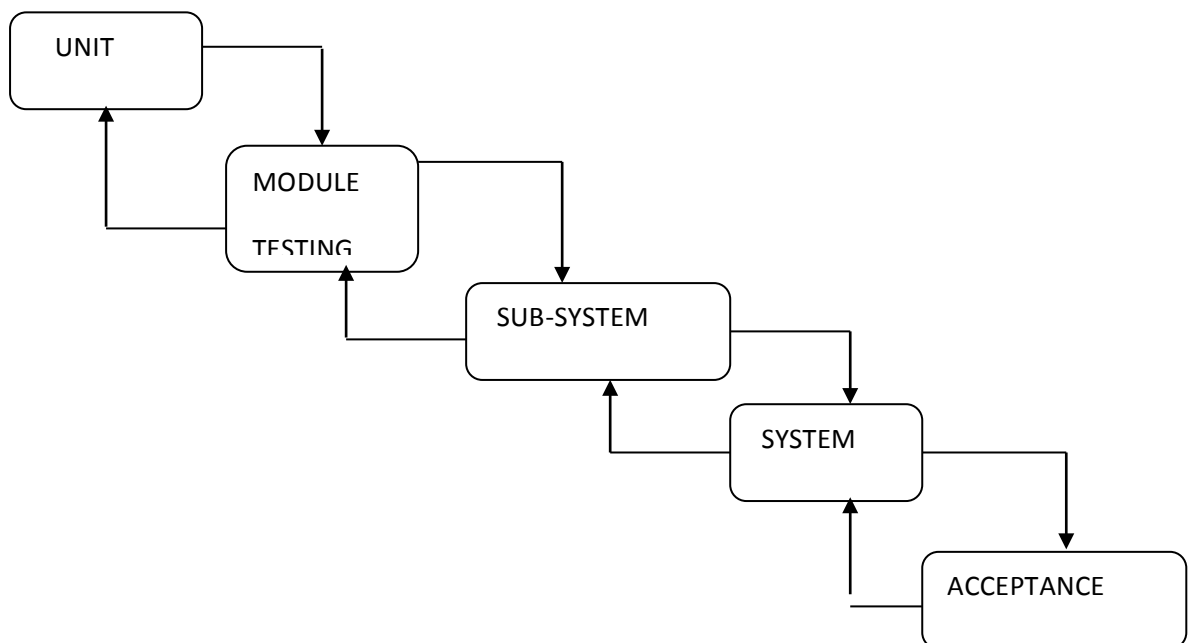f) When defects are fixed continue from activity.



Figure 6.1 Testing Plan

## 6.2 Testing Strategy

The development process repeats this testing sub-process a number of times for the following phases.

a) Unit Testing.
b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

We have performed both Unit Testing and System Testing to detect and fix errors. A brief description of both is given below.

## 6.2.1 Unit Testing

**Objective**

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

**Input:** Unit Test Specifications

**Testing Process**

- Checking for availability of Code Walk-thru reports which have documented the existence of and conformance to coding standards.
- Review of Unit Test Specifications

Verify the Unit Test Specifications conform to the program specifications. Verify that all boundary and null data conditions are included.

## 6.3 TESTING METHODS

**Black-box and White-box Testing**

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the structural system coverage in white box testing.

For this purpose tools or compilers that can insert test probes into the programs can be used.

**Code Coverage**

The way to make sure that you have got all the control flow covered is to cover all the paths in the program during the testing (via white-box testing). This implies that both branches are exercised for an 'if' statement, all branches are exercised for a case statement, the loop is taken once or multiple times as well as ignored for a while statement, and all components of complicated logical expressions are exercised. This is called Path Testing.

Branch Testing reports whether entire Boolean expression tested in control structures evaluated to both true and false.

Additionally it includes coverage of switch statement cases, exception handlers and interrupts handlers. Path testing includes branch testing as it considers all possible combination of individual branch conditions.

A simpler version is Statement Testing which determines if each statement in the program has been executed at least once. The coverage via Path Testing includes the coverage via Statement Testing. Since Path Testing is extremely comprehensive it is costly, hence a viable minimum should be measuring Statement Testing coverage.

**Performance Testing**

Performance testing is design to test the runtime performance of the system within the context of the system. This test is performed at module level as well as at system level. Individual modules developed by Developers are tested for required performance.

## 6.4 Test Cases

A test case has a component that describes an input, action or event and an expected response, to determine if a feature of an application is working correctly.

## 6.4.1 Save User Profile

Table 6.1: Save User Profile Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Profile tab | None | User profile screen shown | **Pass** |
| 3 | Tap on Save | None | Error: "Empty Fields" | **Pass** |
| 4 | Tap on Save | Empty DOB | Error: "Select DOB" | **Pass** |
| 5 | Tap on Save | Empty Name | Error: "Enter Name" | **Pass** |
| 6 | Tap on Save | Empty Height | Error: "Enter Height" | **Pass** |
| 7 | Tap on Save | Empty Weight | Error: "Enter Weight" | **Pass** |
| 8 | Tap on Save | Invalid Height | Error: "Enter Proper Height" | **Pass** |
| 9 | Tap on Save | Invalid Weight | Error: "Enter Proper Weight" | **Pass** |
| 10 | Tap on Save | Proper Details | Data Saved Successfully | **Pass** |
| 11 | Tap on Cancel | Proper Details | Error: "Enter Details to Proceed" | **Pass** |

## 6.4.2 Edit User Profile

Table 6.2: Edit User Profile Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Profile tab | None | User profile screen shown | **Pass** |
| 3 | Tap on Save | None | Error: "Empty Fields" | **Pass** |
| 4 | Tap on Save | Empty DOB | Error: "Select DOB" | **Pass** |

| 5 | Tap on Save | Empty Name | Error: "Enter Name" | **Pass** |
|---|---|---|---|---|
| 6 | Tap on Save | Empty Height | Error: "Enter Height" | **Pass** |
| 7 | Tap on Save | Empty Weight | Error: "Enter Weight" | **Pass** |
| 8 | Tap on Save | Invalid Height | Error: "Enter Proper Height" | **Pass** |
| 9 | Tap on Save | Invalid Weight | Error: "Enter Proper Weight" | **Pass** |
| 10 | Tap on Save | Proper Details | Data Saved Successfully | **Pass** |
| 11 | Tap on Cancel | Proper Details | Changes Discarded | **Pass** |

## 6.4.3 Connect BLE Peripheral

Table 6.3: Connect BLE Peripheral Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Connection Manager tab | None | Connection Manager screen shown | **Pass** |
| 3 | Tap on Scan button. | None | Scanning Started<br><br>Device Found | **Pass**<br><br>**Pass** |
| 4 | Tap On Device Name | None | Device Connected | **Pass** |

## 6.4.4 Disconnect BLE Peripheral

Table 6.4: Disconnect BLE Peripheral Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Connection Manager tab | None | Connection Manager screen shown | **Pass** |

| 3 | Tap On Connected Device Name | None | Confirmation Asked | **Pass** |
| 4 | Tap On Yes | None | Device Disconnected | **Pass** |

## 6.4.5 Save Workout Goal

Table 6.5: Save Workout Goal Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout Goal tab | None | Workout Goal screen shown | **Pass** |
| 3 | Tap on Set | None | Error: "Empty Fields" | **Pass** |
| 4 | Tap on Set | Invalid Duration | Error: "Enter Proper Duration" | **Pass** |
| 5 | Tap on Set | Invalid Distance | Error: "Enter Proper Distance" | **Pass** |
| 6 | Tap on Set | Invalid Calories | Error: "Enter Proper Calories" | **Pass** |
| 7 | Tap on Set | Proper Details | Goal Saved Successfully | **Pass** |

## 6.4.6 Enable Workout Goal

Table 6.6: Enable Workout Goal Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout Goal tab | None | Workout Goal screen shown | **Pass** |
| 3 | Tap on Enable WO Goal | None | Error: "Select Goal Type" | **Pass** |

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 4 | Select Goal Type | Goal Value | Goal Selected | **Pass** |
| 5 | Tap on Enable WO Goal | None | Workout goal enabled | **Pass** |

## 6.4.7 Disable Workout Goal

Table 6.7: Disable Workout Goal Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout Goal tab | None | Workout Goal screen shown | **Pass** |
| 3 | Tap on Disable WO Goal | None | Workout goal disabled | **Pass** |

## 6.4.8 Play Music

Table 6.8: Play Music Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on Music Button | None | Music Panel Appears | **Pass** |
| 2 | Tap on Play button | None | Error: "Empty Playlist" | **Pass** |
| 3 | Tap on Add button | None. | Music Picker Displayed | **Pass** |
| 4 | Tap on Tracks | Tracks to be added | Tracks selected | **Pass** |
| 5 | Tap on Done | None | Tracks added to playlist | **Pass** |
| 6 | Tap on Play button | None | Music Played | **Pass** |
| 7 | Tap on Pause button | None | Music Paused | **Pass** |
| 8 | Tap on Next button | None | Next track played | **Pass** |

| 9 | Tap on Previous button | None | Previous track played | **Pass** |
|---|---|---|---|---|

## 6.4.9 Delete Today's Workout Report

Table 6.9: Delete Today's Workout Report Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on View Report in Dashboard | None | Today's workout list displayed | **Pass** |
| 2 | Tap on workout report | None. | Workout details displayed | **Pass** |
| 3 | Tap on Delete button | None | Workout details deleted | **Pass** |

## 6.4.10 Delete Random Workout Report

Table 6.10: Delete Random Workout Report Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout reports tab | None | All Workouts displayed | **Pass** |
| 3 | Tap on workout report | None | Workout details displayed | **Pass** |
| 4 | Tap on Delete button | None | Workout details deleted | **Pass** |

## 6.4.11 Share Workout Report

Table 6.11: Share Workout Report Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout reports tab | None | All Workouts displayed | **Pass** |
| 3 | Tap on workout report | None | Workout details displayed | **Pass** |
| 4 | Tap on Share button | None | Share menu appears | **Pass** |
| 5 | Tap on Facebook button | None | Facebook Dialog appears | **Pass** |
| 6 | Tap on Post button | None | Status updated on Facebook | **Pass** |

## 6.4.12 Share Workout Statistics

Table 6.12: Share Workout Statistics Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Statistics tab | None | All Workout Statistics displayed | **Pass** |
| 3 | Tap on Share button | None | Share menu appears | **Pass** |
| 4 | Tap on Twitter button | None | Twitter Dialog appears | **Pass** |
| 5 | Tap on Tweet button | None | Tweet sent | **Pass** |

## 6.4.13 Plot HR

Table 6.13: Plot HR Test Case

| No. | Steps | Input | Expected Results | Status |
|-----|-------|-------|------------------|--------|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Heart Rate Monitor tab | None | Heart Rate Monitor displayed | **Pass** |
| 3 | Tap on From | From Date | Date picker appears | **Pass** |
| 4 | Tap on To | To Date | Date picker appears | **Pass** |
| 5 | Tap on Done | None | Plot Appears | **Pass** |
| 6 | Zoom in Plot | None | Units Changed | **Pass** |

## 6.4.14 Add New Reminder

Table 6.14: Add New Reminder Test Case

| No. | Steps | Input | Expected Results | Status |
|-----|-------|-------|------------------|--------|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout Goals tab | None | Workout Goals screen shown | **Pass** |
| 3 | Tap on Workout Reminders tab | None | Workout Reminders screen shown | **Pass** |
| 4 | Tap on Add button | None | New Reminder screen appears | **Pass** |
| 5 | Tap on Save | Empty Title | Error: "Enter Title" | **Pass** |
| 6 | Tap on Save | Empty Alert | Error: "Select Alert" | **Pass** |
| 7 | Tap on Save | Empty Repeat | Error: "Select Repeat" | **Pass** |
| 8 | Tap on Save | Proper data | Reminder Saved Successfully | **Pass** |

## 6.4.15 Edit Reminder

Table 6.15: Edit Reminder Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout Goals tab | None | Workout Goals screen shown | **Pass** |
| 3 | Tap on Workout Reminders tab | None | Workout Reminders screen shown | **Pass** |
| 4 | Tap on Reminder & edit button | None | Edit Reminder screen appears | **Pass** |
| 5 | Change Time | Start Time | Start Time changed | **Pass** |
| 6 | Tap on Save | Proper data | Reminder Updated Successfully | **Pass** |

## 6.4.16 Delete Reminder

Table 6.16: Delete Reminder Test Case

| No. | Steps | Input | Expected Results | Status |
|---|---|---|---|---|
| 1 | Tap on menu button | None | Menu shown | **Pass** |
| 2 | Tap on Workout Goals tab | None | Workout Goals screen shown | **Pass** |
| 3 | Tap on Workout Reminders tab | None | Workout Reminders screen shown | **Pass** |
| 4 | Tap on Reminder | None | Reminder Selected | **Pass** |
| 5 | Tap on Delete button | None | Conformation asked | **Pass** |
| 6 | Tap on Yes | None | Reminder Deleted Successfully | **Pass** |