Chapter 7

Evaluation and testing

7.1. Introduction

In the last chapter, the implementation of each sub system, module and unit of the proposed system was discussed. In this chapter, the proposed system functionalities are compared and tested against software requirements specifications to find out whether the proposed system meets the requirements identified.

7.2 Evaluation of the proposed system

Software evaluation can be defined as the assessment of software product characteristics against the software requirements specifications. During a Software product Evaluation process, a comparative assessment is done, in order to determine how well the software product meets the identified needs of that product . Therefore, the Objective of carrying out this evaluation is to find out whether the software product meets identified requirements of the Compulsory Import Inspection Scheme (CIIS) of Sri Lanka Standards Institution (SLSI).

At first the software was demonstrated and explained to all users of the system at a session held at SLSI training room and encouraged them to use the software hosted in the free server. For the evaluation purpose, an evaluation form was given to the users of the system for their feedback after their trials. Refer appendix N – Software Evaluation. By studying the comments received from the users, it was able to evaluate the software. These comments were received from Director Quality Assurance, Assistant Directors, Perusing officers, Chief Clerks, Receive Issue clerk and Importers. By this evaluation, it was possible to judge about the system, in respect of these Functionality, Understandability, Learnability, Operability, Attractiveness, and Efficiency.

7.3 Software testing

Software application testing is an umbrella term that covers several types of tests. The types of tests are identified according to whether they employ static or dynamic techniques and whether the test is automated or manual. Static testing means, that the code being tested is not executed. Dynamic testing on the other hand involves execution of the code. Automated testing means the computer conducts the test while manual means that people do [13] it.

The objective of testing software is to find and fix any error prior to its acceptance by the client, the SLSI. Testing software begins in the early stages of the design life cycle as there can be design errors, which may be very expensive to fix, as well as implementation errors.

7.4 Stages of testing

7.4.1 Unit testing

Each module is tested alone in an attempt to discover any errors in is code [13]

7.4.2 Integration testing

The process of bringing together all of the modules that a program comprises for testing purposes [13]

7.4.3 System testing

The bringing together of all of the programs that a system comprises for testing purposes [13]

Unit testing was carried out soon after the development of each unit. Once all the units were developed and tested, the integration testing was carried out successfully (e.g. Nomination and processing stage display). After finishing the integration of programs the system testing was carried out in order to ensure that the software meets all specifications of the proposed system.

7.5 Testing strategies

Testing approaches are traditionally categorized as

- Black box - generating tests without the knowledge of the system's implementation, or source code [8]
- White box - generating tests derived in some way by analysis of the source code [8]

To carryout the Black Box testing the following analysis and design diagrams can be used to generate test cases.

- Use-Case
- Use-Case Description
- Activity Diagram
- Sequence Diagram

7.5.1 List of test cases

01. Log-in
02. Add Items
03. Edit Items
04. Delete Items
05. Add Staff
06. Assign staff roles
07. Add Custom Officer
08. Add Importer
09. Add Manufacturer
10. Add Exporter
11. Add New Consignment
12. Submit Notifications
13.Nomination for perusal
14.Entry perusing
15.AD Recommendation
16. D(QA) Recommendation
17. Calculate charges
18. Issue Notification
19.Change Importer Password
20 Change Custom Officer Password
21 Importer Feedback
22 Check Notifications

7.5.2 Test cases

Test case – Log-in

Test Case ID	1
Tested Component	Log-in
Tested Area	Functionality
Purpose	User can log in with UserId and Password and Role
Prerequisites	User already exist
Tested by	105
Tested Date	2009-10-01
Test System	Windows XP, IntelP(IV) , Internet Explorer

Test Data	UserId = {correct, incorrect, empty}
	Password = {correct, incorrect, empty}
	Role={correct, incorrect}

Test Case Description	

No.	**Test Case**	**Sample Data**	**Expected output**	**Results**
1	UserId: correct Password :correct Role: correct	UserId: 105, password: 107 Role: Director	Display home page	Pass
2	UserId: correct , Password: correct Role: incorrect	UserId: 105, password: 107 Role: Clerk	Error message "Incorrect UserId,Password or Role." should be displayed	Pass
3	UserId :correct , Password :incorrect Role: correct	UserId: 105, password: abc123 Role: Director	Error message "Incorrect UserId,Password or Role." should be displayed	Pass
4	UserId: correct, Password: incorrect Role: incorrect	UserId: 105, password: abc123 Role: importer	Error message "Incorrect UserId,Password or Role." should be displayed	Pass
5	UserId: correct Password: empty Role: correct.	UserId: 105 password: Role: Assistant Director	Message "Please input your password" should be displayed	Pass

6	UserId: correct Password: empty Role: incorrect.	UserId: 105 password: Role: Chief Clerk	Message ”Please input your password” should be displayed	Pass
7	UserId: incorrect Password: correct Role: correct.	UserId: 200, password: 107 Role: Assistant Director	Error message “Incorrect UserId,Password or Role.” should be displayed	Pass
8	UserId: incorrect Password: correct Role: incorrect.	UserId: 200, password: 107 Role: Counter Clerk	Error message “Incorrect UserId,Password or Role.” should be displayed	Pass
9	UserId: incorrect Password: incorrect Role: any	UserId: 5, password: abc123 Role: Director	Error message “Incorrect UserId,Password or Role.” should be displayed	Pass
10	UserId: incorrect Password: empty Role: any	UserId: 5 password: Role: Counter Clerk	Message ”Please input your password” should be displayed	Pass
11	UserId: empty Password: correct Role: correct	UserId: password: 107 Role: Director	Message ”Please input your UserId” should be displayed	Pass
12	UserId: empty Password: correct Role: incorrect	UserId: password: 107 Role: Assistant Director	Message ”Please input your UserId” should be displayed	Pass
13	UserId: empty Password: empty Role: any	UserId: password: Role: Director	Message ”Please input your UserId and Password” should be displayed	Pass

Table 7.1 - Test case – Log-in

(Please refer appendix L for other test cases)

7.6 Summary

In this chapter, the Objectives and strategies of software evaluation, diagrams that can be helpful in generating test cases and a list of test cases including test data & results are discussed.

The next chapter will be focused on the conclusion of the software development.