Chapter 7

Evaluation & Testing

7.1 Introduction

In the last chapter, IPS-IRD system implementation stage was discussed. The system implementation was based on the software requirements, on sub module scenarios and on sequence. Evaluation of the implemented system will be illustrated, in this chapter. Comparing to software requirements with the system functionalities, testing was done to find whether the implemented system addresses the problem identified at the Problem Domain stage discussed in Chapter 2.

The objective of carrying out a software evaluation is to find out whether the requirements requested by stake holders and the process planned by the developer met up to the expected requirements and the quality of the software with maintainability.

7.2 Assessment

IPS-IRD system is a web-based system that can be accessed through a web browser. Transmission media may be a local intranet system using web browser. If the system was installed in a web server, access can be done through internet. But if the program was installed in a departmental local intranet sever, it permits to access only to local machines for security restrictions. An advantage of using web based features was, personalized computers were not wanted for each user. to use them. This feature was very important factor to cost effectiveness of the system. All modifications to the system can be done at a single point in a web-based system. It was an other advantage of implementing IPS-IRD system as a web-based system, because there will be likely 300 computers available in the intranet to be managed.

7.3 System performance

According to the identification of software verification and validation principles, the system performance will be illustrated on following topics.

7.3.1 Efficiency

IPS-IRD provides a solution to improve the efficiency on activities of information handling process while reducing process time and security. After the implementation of the prototyped software, it was demonstrated to the users and the management of the department and they were extremely happy about the system which leaved manual interventions compared to current processes. Especially the management proposed to expand the system to the personal tax branches established in regionally when they can manage the financial requirements.

7.3.2 Usability

Usability of the implemented system to be understood on standardized approach learning system functionalities using user friendly GUIs and the expected output of the functions throughout the system data within a short period. To increase the usability of the user interaction with the system, it is highly simplified and understandable GUIs were introduced, because the understandability is the fundamental of the usability. Since the expected output was returned on the inputs, the system users make their works easier.

7.3.3. Learnability

When demonstrating the prototype to the users and management, the application process of the software was learned well by the users. So, the capability of the software product was successful. There were no major concerns over the application.

7.3.4. Operability

Extra skills or knowledge on software were not required by the system users to operate the IPS-IRD system, because the operational methods used on system development was very common and standardized for software usage. As an example, users have to operate the system using the navigation method which is pretty common in most of the software.

7.3.5 Attractiveness

There were four colours only used to design GUIs, for the expected clean with the simple express not leading confusion when using the system. But few users expressed their idea about the input information interface as an example, that there may be some confusion when choosing external or internal information input path. They suggested that separate interfaces must be implemented for two categories of information.

7.3.6 Installability

Since, IRD already use licensed version of Windows operation system, any issue on installablility was not created with WAMP which comprises three open source software namely Apache, MySQL and PHP. For the security purpose the security wall will be installed by the management of the Department.

7.4 Software testing

To validate the project, testing must be done because the errors in program can only be identified through testing. Executing a program with testing methods, it can be determined that the required result will meet. A software system will defer to a manual system by failures. It will not depends only on requirement fulfilling. As an example, if the requirement fulfilled, consisteney can be failed. Software testing must be done to evoke this type of failures and errors and the system functionalities done in actual manner must be ensured.

7.4.1 Fundamental testing processes

7.4.1.1 System testing

System testing involves integrating of groups of components to a system or sub-system. A separate testing team carries out system testing. It will be concerned with the user requirements.

7.4.1.2 Component testing

Component testing must be a responsibility of the component developer. Individual components will be tested in this process. Tests will be differed on developer's experience. This process is called as functionality testing also.

7.4.2 Main approaches for functionality testing

This testing method is to verify if the product meets the requirements and if the works were done in correct way. The requirements mentioned in the SRS can be tested on two ways.

- Black Box testing approach
- White Box testing approach

7.4.2.1 White box testing approach

In this testing method, test data will be derived from structure and flow of the software. Testing plans will be made on the details of the software implementation, such as programming language, logic, and styles.

7.4.2.2 Black box testing approach

In this testing method, test data will be derived from the specified functional requirements on SRS. To fulfill the testing activities identified using inputs and outputs will be used. And it visualizes functionality of the software requirements. If the system works correctly, predicted output must be released by the system.
The Black Box testing approach has been selected for testing the IPS-IRD system, the best way to use scenario-based testing developing test cases.

Black Box testing must be followed by built test cases on scenarios such as,

1. Use-cases
2. Activity diagrams
3. Sequence diagrams

Out of the given list, activity diagrams are best suited for the building of test cases as it clearly show interface between user and the system.

7.4.2.3. List of Test Case Scenario

01. Log-in to the system
02. Delete User
03. Edit User
04. Change password
05. Input Information

06. Input Files
07. Upload Action Taken
08. Distribute Information
09. View new Information
10. View past information records
11. Call Explanation

7.5 Test Plan

Example test case was planned on activity diagram on user log-in.

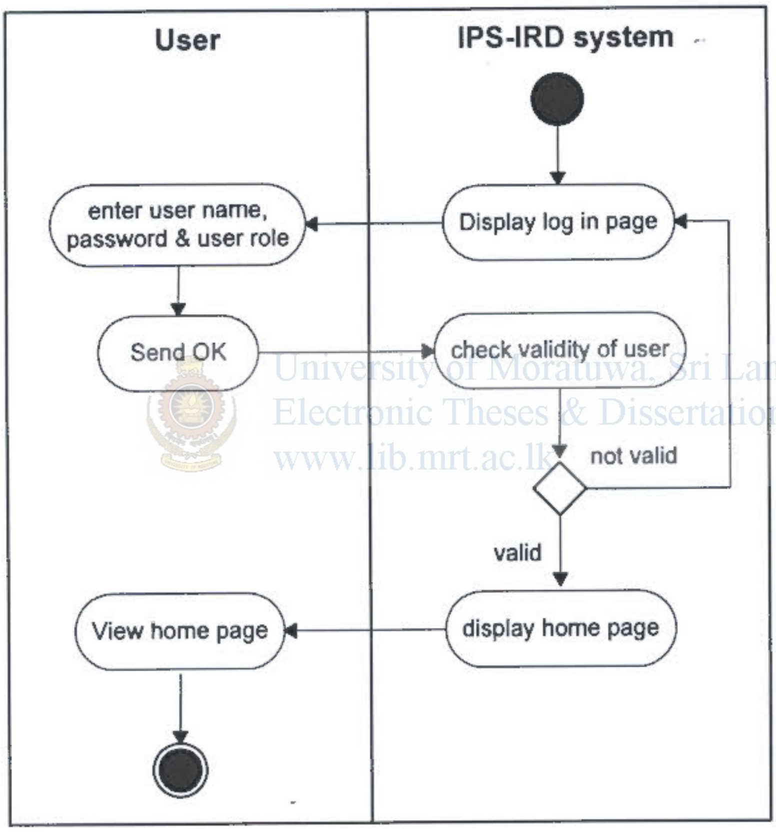

Figure 7.1: Activity diagram for log-in process

The activity diagram illustrates the communication path in between the user and the system.

7.5.1 Test Cases

Test Case – 01

Test Component - User Log-in to system

Test case description

	Input Activity(s)	Expected Output	Status
(!)	Enter Valid Username and Valid Password	Display home page	Success
(2)	Enter Incorrect Username and Correct Password	Error message should be displayed as "Invalid user name. Enter again"	Success
(3)	Enter Correct Username and Incorrect Password	Error message should be displayed as "Enter correct password"	Success
(4)	Enter Incorrect Username and Incorrect Password	Error message should be displayed as "Enter username & password correctly"	Success
(5)	Enter Invalid Username and Valid Password	Error message should be displayed as "Not a valid user"	Success
(6)	Enter Valid Username and Invalid Password	Error message should be displayed as "Enter valid password"	Success
(7)	Enter Invalid Username & Invalid Password	Error message should be displayed as "user does not exists"	Success

Table 7.1 : Test case description for log-in

Tested data & result for test case -01

Item No	Sample Data	Data specification	Path Result	System result
(1)	U.Name : 3020 Password: abt30	Text, Numeric	Data passed	Valid user identified
(2)	U.Name : 302 Password: abt30	Text, Numeric	Data passed	Invalid username identified
(3)	U.Name : 3020 Password: @@	Text, Numeric	Data passed	User identified and identified the password is incorrect
(4)	U.Name : 032 Password: ##	Text, Numeric	Data passed	Identified as incorrect username & incorrect password
(5)	U.Name : ABCD Password: abt30	Text, Numeric	Data passed	Identified not valid user
(6)	U.Name : 3020 Password: blank	Text, Numeric	Data passed	User identified and identified invalid password
(7)	U.Name : ABCD Password: blank	Text, Numeric	Data passed	Did not accepted the user

Table 7.2 : Tested data & results of test case 01

Other test cases are included in *Appendix J.*

7.6 Summery

This chapter described about the evaluation of the project and how the system tested. The next chapter will be focused on conclusion and further enhancements.