

## Exercise 3 Test Program and Document Procedures

This exercise requires you to document and execute a functional test on the logon dialog as well as briefly outline and document the requirements for load testing the logon screen.

You are also required to produce clear and concise technical documentation, end-used documentation and report on the results of usability testing the logon dialog with recommendations.

### Exercise 3 (a)

This first part of the exercise requires you to create a document that critically reviews and analyses the program against expected outcomes, identifying any discrepancies. This review and analysis is to be done under two headings; Functional Testing and Load Testing.

#### Functional Testing

You are required to create documentation for a black box (functional) test of the logon dialog. Include the following in your documentation:

- Test Plan. A completed test plan which includes test cases, test data and expected results. In your test cases include the following test cases where you test the:
  - Data entered in each of the input fields
  - Error messages in the event of an invalid entry
  - Navigation about the logon dialog – tab index
  - Links between forms and dialogs, both forward and backward
- Test Execution. A completed test log that records the results of each test case, compares the actual results with the expected results and comments on any discrepancies between the two.

## Load Testing

Load testing is a performance test which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously. Load testing usually identifies -

- The maximum operating capacity of an application
- Determine whether the current infrastructure is sufficient to run the application
- Sustainability of application with respect to peak user load
- Number of concurrent users that an application can support, and scalability to allow more users to access it.

You are required to create documentation for load testing the logon dialog. Include the following in your documentation:

- Test Environment. A brief outline of the test environment setup including the software, hardware and network configuration for the testing teams to execute the test cases.
- Test Tools. Briefly outline the load testing tools to be used. Load testing is very important and you will most probably decide to build your own software tools to automate the load test and to execute test cases.
- Test Plan. A completed test plan which will probably have only one test case. Include the test data the test case (large volumes of data in data files – just include a small sample set of say 5 – 10 cases) and expected results. Remember, it is a load test so you are testing:

- How many successful login attempts can happen in a second/minute/hour
- How many unsuccessful login attempts can happen in a second/minute/hour

Note any assumptions that you make, for example:

- Min/max number of concurrent users
- Acceptable response time measures in milliseconds, seconds
- Test Execution. You are not required to actually run the load test, however, document this as well as what you would expect the automated load test software suite to report; such as a comparison between the actual results with the expected results and any discrepancies between the two.

## Exercise 3 (b)

This second part of the exercise requires you to produce documentation that evaluates independent feedback on the program and make recommendations for improvement. Also produce documentation that will support future maintenance of the program (e.g. user manual, technical manual).

In detail, produce four reports; Technical Documentation, End-User Documentation, Results of Usability Testing and Recommendations based on Usability Testing.

- Produce clear and concise **Technical Documentation** for the logon dialog Application Programming Interface (API).
- Produce clear and concise **End-Used Documentation** that describes how the end-user uses the logon dialog.
- From an independent feedback perspective, produce **Usability Testing Documentation** that outlines the result of usability testing on the logon dialog.

Usability testing involves reviewing the user interface (logon dialog) with the people who will be using the application. This is to ensure that the design (layout and sequence, etc.) enables the application (logon dialog) functions to be executed as easily and intuitively as possible.

Usability testing asks: Is the software:

- Easily understood?
  - Easy to learn?
  - Easy to operate?
  - Attractive to the end-user?
  - Are the widgets used the most appropriate?
  - Is the tab order correct?
  - Is the end-user documentation easily accessible and helpful?
- Now, based on the results of the usability testing document, produce documentation making **Recommendations** on how the logon dialog could be improved.