Software Engineering Design I

Winter 2015

Assignment 4 – Software Testing Document

Group #1

PRMS

PROFESSIONAL RESOURCE MANAGEMENT SYSTEM

Contributing Members:

Abad H., Brandon C. and Rafael N.

Date Submitted: 3/17/2015

Approval

This document has been read and approved by the following team members responsible for its implementation:

Print Name	Signature	Comments

Print Name	Signature	Comments

Print Name	Signature	Comments

Document Status Sheet

Document Title	Software Testing Plan (Document)
Document ID	PRMS/Documents/Product/STP/1.0.0
Author(s)	A. Hameed, B. Crispino, R. Nazario
Version	1.0.0
Document Status	draft / internally approved / <u>conditionally approved</u> / approved

Version	Date	Author(s)	Comments
0.0.1	03-03-2015	A. Hameed	Document Creation
0.0.2	06-03-2015	A. Hameed	Added text to ch. 1.1 and 1.3
0.0.3	09-03-2015	A. Hameed	Added text to ch. 1.2 and 2
0.0.4	13-03-2015	A. Hameed	Added text to ch. 3.1, 3.2 and 3.3
0.0.5	14-03-2015	A. Hameed	Added text to ch. 4
0.0.6	14-03-2015	R. Nazario	Added text to ch. 4
0.0.7	15-03-2015	B.Crispino	Added to and edited text in ch.4
0.0.8	15-03-2015	B.Crispino	Edited text in ch.3
0.0.9	16-02-2015	A. Hameed	Added text to ch. 4
0.1.0	16-03-2105	R. Nazario, A. Hameed, B. Crispino	Internally accepted
1.0.0	16-03-2015	A. Hameed	Fixed document status

Contents

1   Introduction

1.1   System Overview

The software will be used to implement a client-server resource acquisition system, namely Professional Resource Management System (PRMS). This system allows users to access a database of library resources consisting of textbooks, encyclopedias, articles and technical documents, from which they can acquire resources relevant to their needs. It hides the complexity of a resource acquisition system making the software easy to use. Usability is further increased by offering a front-end GUI for users to log in to and access the system.

The environment identified for the PRMS system is similar to a library system. The software will allow the user to specify one or more of the following to search/filter through the database of resources: document name, document type, subject, publication year and author. The software also enables the user to sort acquired data by document type, subject, publication year and author for efficiency and a better user experience. Being a client-server system, all user requests at the client system will be sent to a server for acknowledgement and a response. System administrators will be the power users with access to the backend of the software.

1.2   Test Approach

The initial testing approach for the PRMS software will be to perform 'black-box testing'. This means that all initial tests performed will be in accordance to the software specifications (reference 1.3.2) and will primarily be input/output driven. The test cases written and performed will all reference the specifications and its various diagrams (use-case, class, sequence) without an in-depth look into the software logic.

Upon completion of the majority of the implementation, a 'glass-box/white-box testing' approach will be taken. This approach will be fundamentally logic-driven and path-oriented and will require use of the actual product code to finalize and perform test runs and scenarios.

The ultimate goal of software testing will be to ensure the PRMS software works exactly as is required and provides the user with a functional and efficient software to utilize for their resource acquisition needs.

1.3 Documents

1.3.1 Reference Documents

[SWE Design I Project Outline]	Naser, H. (2015). Team-project: client-server resource acquisition system [class handout]. Department of Engineering, Lakehead University, Thunder Bay, ON.
[SWE 3670 Textbook]	Schach, S. (2011). *Object-Oriented and Classical Software Engineering* (8th ed., p. 667). New York: McGraw-Hill Education.
[SWE 3050 Textbook]	Lethbridge, T., & Laganiere, R. (2005). *Object-Oriented Software Engineering* (2nd ed., p. 533). Maidenhead, Berkshire: McGraw-Hill Education.

1.3.2 Applicable Documents

[SW Requirements Document]	Hameed, A. Crispino, B., Nazario, R. (2015). PRMS [requirements document]. Department of Engineering, Lakehead University, Thunder Bay, ON.
[SW Specifications Document]	Hameed, A. Crispino, B., Nazario, R. (2015). PRMS [specifications document]. Department of Engineering, Lakehead University, Thunder Bay, ON.
[SW Design Document]	Hameed, A. Crispino, B., Nazario, R. (2015). PRMS [design document]. Department of Engineering, Lakehead University, Thunder Bay, ON.

2 Glossary

2.1 Definitions

Algorithm	A process or set of rules to be followed in calculations or other problem-solving operations.
Client	Program that is used by all the users and system admins.
PRMS Software	A software implementing a client-server resource acquisition system.
Resource	Material which can be readily drawn from a server and displayed to the client for effective use.
Server	A computer or computer program that manages access to a centralized resource or service in a network (i.e. PRMS database)
System Administrator	The system admin oversees the entire PRMS system and has the right to configure the system, to create and remove admins and resources as well as any other high level configuration.
User	Any individual or system administrator with access to the client machine using PRMS.

2.2 Abbreviations

CASE Tools	Computer-aided Software Engineering Tools
COCOMO	Constructive Cost Model
GUI	Graphical user Interface
IDE	Integrated Development Environment
IP	Internet Protocol
KLOC	Thousands (K) of Lines of Code
PRMS	Professional Resource Management System
SVN	Apache Subversion
TCP	Transmission Control Protocol
UML	Unified Modeling Language
XML	eXtensible Markup Language

3 Test Plan

3.1 Features Being Tested

The PRMS software will be thoroughly and fundamentally tested both from a functional and non-functional standpoint. In order to test its prime features, the test plan will split the software into the following four categories:

1. *Client-Server Architecture*
   This test category will ensure that all socket communication between the client machine(s) and server is fully functional, efficient and accurate with minimal disruptions (i.e. network lag, client's inability to support software, server disruption).

2. *Resource Acquisition and Database*
   This category will test the back-end application functionality. This primarily means any and all PRMS functions from a server's standpoint will be rigorously tested for effortless performance. The database connectivity, various algorithms for searching, sorting and filtering as well as all functions implementing the user's actions fall into this category.

3. *User Interface*
   This category will test the front-end application functionality. The user interface being the face of the software the users utilize and the media by which they communicate their request definitely needs to work with perfection. As the user will see nothing but the front-end portion of the software, all physical features of the user interface (GUI) from drop-down menus to text boxes to button clicks will be rigorously tested for the best performance and ultimate user experience.

4. *Complete Software*
   This category of the test is the final stage where end-to-end testing will be performed. This means, ensuring the PRMS software works in all manners as per the requirements, specifications and design documentation. Here, all functionality and interfaces will be tested overall finalizing the completeness of the product. The end-to-end testing will primarily ensure that a user is able to sit on a client machine and be able to use the software as they would without any conflict or confusion.

3.2 Features Not Being Tested

The following features as displayed in the use-case diagrams (specifications, sections: 4.1-4.2) of the PRMS software will be excluded from the test plan as they do not directly relate to the core functionality of this software, which is resource acquisition.

1. *Access Tutorial*
   The user is able to access a tutorial page where they may clarify any confusion or better learn to use the software.

2. *Modify Database*
   The system administrator is able to access the database and insert, update and/or delete resources as they may wish.

3. *Backup Database*
   The system administrator privileged to access the database may choose to create a backup for any reason they may see fit.

3.3 Testing Tools and Environment

With knowledge of various integrated development environment (IDE) platforms available on the market, Microsoft VisualStudio has been chosen to implement the PRMS system. Tied together source control management system such as TFS or SVN, this IDE provides the best interface to work together in a team environment.

A variety of testing tools are also available to ensure that all testing requirements are fulfilled. In reference to the PRMS system, when implementing black-box testing, a team member will simply run through the test cases to ensure all input/output functionality is working as per the specifications. Moving forward to glass-box testing, the following tools may be utilized:

1. *MSTest*
   A command line utility provided by Microsoft to execute unit tests.

4　Test Cases

4.1　Case 1: Test communication link (socket) between client/server

Purpose	Ensuring connection/communication between client/server establishes
Precondition	The client must know the Port_Number and IP_Address of server
Inputs	1. Server IP address 2. Server port number
Expected Outputs	1. Message confirmation: 'Connection Established!' 2. Login Page displayed
Pass/Fail Criteria	1. Server IP and port must be valid 2. Connection request must be made from valid client
Test Procedure	1. Run 'server program' on computer A 2. Type the server port number and press <enter> 3. Run 'client program' on computer B 4. Type the server IP address and port number and press <enter> 5. Wait for 'connection to server established' confirmation on B 6. Wait for Login Page to be displayed on B

4.2   Case 2: Test communication link over multiple clients

Purpose	Ensuring connection between all clients and server establishes
Precondition	Each client must know the Port_Number and IP_Address of server
Inputs	1. Server IP address 2. Server port number
Expected Outputs	1. Message confirmation: 'Connection Established!' 2. Login Page displayed on each client
Pass/Fail Criteria	1. Server IP and port must be valid 2. Connection request must be made from valid clients
Test Procedure	1. Run 'server program' on computer A 2. Type the server port number and press <enter> 3. Run 'client program' on each client computer 4. Type the server IP address and port number and press <enter> 5. Wait for 'connection to server established' confirmation 6. Wait for Login Page to be displayed

4.3   Case 3: Test creating a new user-account

Purpose	To create a new user-account to be able to log onto the system.
Precondition	The user knows the *IP_Address*, *Port_Number,* and has an e-mail in the Lakeheadu.ca domain
Inputs	1. Port_Number 2. IP_Address 3. Username required 4. Password preferred
Expected Outputs	1. Message confirmation "Connection Established!" 2. Login Page Displayed 3. Message confirmation "New User Successfully Registered!"
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username must finish with "@lakeheadu.ca"
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Create Account> 6. Wait for the message "New User Successfully Registered!" 7. The *Login Page* will be re-loaded.

4.4 Case 4: Test user logon to system

Purpose	To access the Server.
Precondition	The user knows the *IP_Address*, *Port_Number,* and is a registered user in the system.
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. Wait for the *Search Page* to be displayed

4.5   Case 5: Test Searching the Database

Purpose	To search database based on Name, Subject, Type, Author and Year
Precondition	User already has an account and knows how to use the search system
Inputs	1. Port_Number and IP_Address 2. Username and Password 3. Specific Document Name 4. Specific Document Subject 5. Specific Document Type 6. Specific Document Author 7. Specific Document Publication Year
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid 3. "Specific Document Name" must be valid 4. "Specific Document Subject" must be valid 5. "Specific Document Type" must be valid 6. "Specific Document Author" must be valid 7. "Specific Document Publication Year" must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Enter *Specific Document Name* in <Document Name>, select the required *Document Type, Subject, Publication Year and Author* from one of the options displayed and click on the <Search> button 9. Wait for the *Result Page* to be displayed

4.6   Case 6: Test New Search Request

Purpose	To start a new search after already searching once
Precondition	User has already searched once and is now looking for a new source
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password 5. Search operation
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed 6. Search Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Perform search described in the above case 9. Wait for the *Result Page* to be displayed 10. Click on the <New Search> button

4.7   Case 7: Test Sort by Document Name

Purpose	To sort the results by Document Name
Precondition	User has already performed the search in the system
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password 5. Search Operation 6. Select Document Name Option for Sort
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed 6. Result Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid 3. Search performed must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Perform search described in above case 9. Wait for the *Result Page* to be displayed 10. Select *Document Name* in the option menu displayed and click on the <Sort> button 11. Wait for the *Result Page* to be displayed

4.8   Case 8: Test Sort by Document Type

Purpose	To sort the results by Document Type
Precondition	User has already performed the search in the system
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password 5. Search Operation 6. Select Document Type Option for Sort
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed 6. Result Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid 3. Search performed must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Perform search described in the above case 9. Wait for the *Result Page* to be displayed 10. Select *Document Type* in the option menu displayed and click on the <Sort> button 11. Wait for the *Result Page* to be displayed

4.9   Case 9: Test Sort by Subject

Purpose	To sort the results by Document Type
Precondition	User has already performed the search in the system
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password 5. Search Operation 6. Select Subject Option for Sort
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed 6. Result Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid 3. Search performed must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Perform search described in the above case 9. Wait for the *Result Page* to be displayed 10. Select *Subject* in the option menu displayed and click on the <Sort> button 11. Wait for the *Result Page* to be displayed

4.10 Case 10: Test Sort by Publication Year

Purpose	To sort the results by Publication Year
Precondition	User has already performed the search in the system
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password 5. Search Operation 6. Select Publication Year Option for Sort
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed 6. Result Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid 3. Search performed must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Perform search described in the above case 9. Wait for the *Result Page* to be displayed 10. Select *Publication Year* in the option menu displayed and click on the <Sort> button 11. Wait for the *Result Page* to be displayed

4.11 Case 11: Test Sort by Author

Purpose	To sort the results by Author
Precondition	User has already performed the search in the system
Inputs	1. Port_Number 2. IP_Address 3. Username 4. Password 5. Search Operation 6. Select Author Option for Sort
Expected Outputs	1. Message Confirmation "Connection Established!" 2. Login Page Displayed 3. Message Confirmation "User Successfully Validated!" 4. Search Page Displayed 5. Result Page Displayed 6. Result Page Displayed
Pass/Fail Criteria	1. Port_Number and IP_Address must be valid 2. Username and Password must be valid 3. Search performed must be valid
Test Procedure	1. Run *Client.exe* on the computer 2. Type the server *Port_Number* and *IP_Address* and click on the <Request Connection> button 3. Wait for the message from the server "Connection Established!" 4. Wait for the loading of the *Login Page* 5. Type Username and Password and click on the button <Login> 6. Wait for the message "User Successfully Validated!" 7. The *Search Page* will be Displayed 8. Perform search described in the above case 9. Wait for the *Result Page* to be displayed 10. Select *Author* in the option menu displayed and click on the <Sort> button 11. Wait for the *Result Page* to be displayed

Appendix: Test Logs

A.1 Log for Test 1

Test Results	
Incident Reports	

A.2 Log for Test 2

Test Results	
Incident Reports	

A.3 Log for Test 3

Test Results	
Incident Reports	

A.4 Log for Test 4

Test Results	
Incident Reports	

A.5 Log for Test 5

Test Results	
Incident Reports	

A.6 Log for Test 6

Test Results	
Incident Reports	

A.7 Log for Test 7

Test Results	
Incident Reports	

A.8 Log for Test 8

Test Results	
Incident Reports	

A.9   Log for Test 9

Test Results	
Incident Reports	

A.10 Log for Test 10

Test Results	
Incident Reports	

A.11 Log for Test 11

Test Results	
Incident Reports	