

Archway Project

SOFTWARE TEST SPECIFICATION

Version 3.0

**Direct 6 Software Solutions - CSC 191 Senior Project
Sacramento State - Department of Computer Science
College of Engineering and Computer Science
Advisor Dr. Bolan Jiang**

Table of Contents

1 INTRODUCTION.....	4
1.1 Purpose	4
1.2 Scope.....	4
1.3 Definitions, Acronyms and Abbreviations	5
1.4 References.....	5
1.5 Overview of Contents of Document.....	6
2. TEST PLAN DESCRIPTION.....	7
2.1 Product Summary.....	7
2.2 Responsibilities	10
2.3 Schedule	10
3. TEST DESIGN SPECIFICATION	11
3.1 Testing Approach.....	11
3.2 Feature or Combination of Features not to be Tested.....	11
3.3 Environmental Needs	11
3.4 Suspension / Resumption Criteria	12
3.5 Risks and Contingencies.....	12
4. TEST SPECIFICATION.....	13
4.1 Test Procedures.	13
4.2 Test Procedure Conventions.....	13
4.3 Test Data.	13
4.4 Use Case Testing.....	14
4.4.1 Use Case: Take Survey.....	14
4.4.2 Use Case: Login.....	15
4.4.3 Use Case: Log out.....	17
4.4.4 Use Case: Generate Reports.....	18
4.4.5 Use Case: Print Reports.....	19
4.4.6 Use Case: Create Custom Reports.....	20
4.4.7 Use Case: Create Data.....	21
4.4.8 Use Case: Update Data.....	23
4.4.9 Use Case: Remove Entry.....	25
4.4.10 Use Case: Reset Table.....	26
4.4.11 Use Case: Remove User.....	27
4.4.12 Use Case: Reset Password.....	28
4.4.13 Use Case: Update User.....	29
4.4.14 Use Case: Create User.....	31

5. SYSTEM TEST / REQUIREMENTS TRACEABILITY	33
5.1 Design Component Traceability Matrix.....	33
6. APPROVAL SIGNATURE PAGE	34
Appendix A	35

1. INTRODUCTION

This is the System Test Specification (STS) document for the Archway project sponsored by Steven Archer.

This project is being undertaken by the Direct 6 Software Solutions development team. The team is comprised of undergraduate students majoring in Computer Science at California State University, Sacramento. The team members are enrolled in a two-semester senior project course required of all undergraduate majors. Successful delivery of the desired software product will fulfill the senior project requirement for the student team members.

PROJECT SPONSOR:

Name: Mr. Steven Archer

Title: Planning and Development

Organization name: Inter-tribal Council of California

Contact information: Cell: (916) 320-0233

Email: archers@cwnet.com

DIRECT 6 SOFTWARE SOLUTIONS DEVELOPMENT TEAM

Jared Arbaugh	Cell: (916) 771 – 3501	Email: arbaughj@gmail.com
Dave Diel	Cell: (916) 402 – 8640	Email: diel@surewest.net
Ben Mackin	Cell: (408) 646 – 3521	Email: bmackin@mac.com
Khae Saetern	Cell: (916) 208 – 1964	Email: khaes1421@hotmail.com
Benjamin Tovar	Cell: (209) 712 – 1417	Email: tovarb@gaia.ecs.csus.edu
Kou Vang	Cell: (916) 233 – 5051	Email: vangkou@gmail.com

1.1 Purpose

The purpose of the STS are: 1) to describe the plan for testing the Archway project, and 2) to specify the test cases and test procedures necessary to demonstrate that the Archway project satisfies the requirements as specified in the project's System Requirements Specification document.

1.2 Scope

This document contains a list, and brief description of the use cases to be tested and the software components associated with each test case. The software testing plan also provides a schedule for the testing, and also provides the assignment of team members to their respective testing tasks. The process for documenting and resolving software errors and/or anomalies that are found during the testing is also specified. The test specification includes a list of the features to be tested for each of the use cases, along with test procedures, and steps necessary to execute each of the test cases.

1.3 Definitions, Acronyms, and Abbreviations

This subsection serves as a glossary for the document. All technical terms used as well as all acronyms and abbreviations used are arranged in alphabetical order. The purpose of this subsection is to provide the reader with quick access to the technical references used throughout the document.

1.3.1. Definitions:

Archway – The name of the software program being developed for ITCC. (Formally known as the ITCC Database Project)

MySQL – An open source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. Because it is open source, anyone can download MySQL and tailor it to their needs in accordance with the general public license. MySQL is noted mainly for its speed, reliability, and flexibility.

Software feature - A distinguishing characteristic associated with a use case (e.g. its functionality, performance, ease of use, performance, etc.).

Software problem report - A document reporting on any event that occurs during the testing process which requires investigation (see appendix A for a copy of the Software Problem Report form).

System test report - A document summarizing testing activities and results. It also contains an evaluation of the degree to which the software product satisfies to the system requirements for each of the use cases.

Test case specification - A specification of inputs, expected results, and a set of execution steps associated with the testing of a feature (or features) associated with a use case.

Test log - A chronological record of relevant details about the execution of tests.

1.3.2. Acronyms:

DBMS – Database Management System
ITCC – Inter Tribal Council of California
HTML – Hypertext Markup Language
PC – Personal Computer
PHP – PHP Hypertext Preprocessor
STS – System Test Specification
STR – System Test Report

1.4 References:

Document was prepared using the following references below.

- Direct 6 Software Solutions Software Requirements Specification document

- Direct 6 Software Solutions Software Design Specification document
- Buckley, R. 2004. Guide to Preparing the System Test Specification Document. 1998. IEEE Standard for Software Test Documentation - IEEE STD 829-1998. New York: IEEE Press, 1998.
- Hower, Rick. 1996-2005. Software QA and Testing Resource Center © <http://www.softwareqatest.com/index.html>.

1.5 Overview of Contents of Document

Section 2: Test Plan Description

This section provides a summary of the Use Cases and the plan for carrying out the system test phase of the team's software development process. More specifically, this section contains a brief description of each Use Cases to be tested, the team member (or members) assigned to test each Use Case, the testing schedule, and the risk management plan.

Section 3: Test Design Specification

This section describes the details of the test approach, lists the features that are and are not to be tested, lists the environmental needs, and details the pass/fail and suspension/resumption criteria.

Section 4: Test Specification

This section contains subsections for each of the use cases to be tested. Each use case subsection specifies the features to be tested, the procedures necessary to run the test cases, the items being tested.

Section 5: Requirements Traceability

This section provides for a cross referencing of each test case to its software requirement specification (or specifications) and also to its design component (or components).

Section 6: Approvals

This section contains the list of the key signatories necessary to sign-off on the STS, thereby agreeing to the scope and content of the test plan and test cases specified within the document. Approval constitutes a guarantee that the development team has produced a test specification sufficient for validating the software to be delivered to the sponsor.

Appendix A: Software Problem Report

This section contains a copy of the Software Problem Report.

2. TEST PLAN DESCRIPTION

The intent of system test is as follows

- To specify the activities required to prepare for and conduct the system test.
- To identify the tasks that must be performed, the team members assigned to each of the tasks, and the schedule to be followed in performing the tasks.
- To identify the sources of information used to prepare the plan.
- To identify the test tools and environmental needs for conducting the tests.

2.1 Product Summary:

The software being developed can be divided into two main sections.

- The first section is an office database. It will keep information in the areas of Reservations and Rancherias. This database is password protected.
- The second section is an online survey. Our sponsor will be able to create multiple surveys. Each survey created will generate reports to satisfy the sponsor's data collection needs. Surveys are password protected.

Both sections use php and html for the front end, and mysql as a database. The project is being tested and developed on gaia, for ease of development and debugging, 24 hour access, and to make sure that anything we do does not interfere with ITCC's day to day operations.

The following table maps use cases to features and components.

Use Case	Features	Components
Take Survey	Radio Button survey Submit Button	Access page take_survey.php Access Database Tables surveyCatagory surveyQuestions surveyAnswers surveyResponse
Login	Textbox name Textbox password Submit Button Clear Button	Access page login.php Access Database Tables uLogin
Logout	Link	Access Pages header.php
Generate Reports	Survey Submit Radio Database Submit Radio	Access Pages generate_reports.php

	Search Submit Button Text Box Search Multiple Checkboxes	Access Database Tables All Tables used for this
Print Reports	Submit Button Identifier Textbox	Access Pages generate_reports.php
Create Custom Reports	Submit Button Radio button byCounty Radio button byName Checkboxes for all Tables in database.	Access Pages create_custom_reports.php Access Database Tables All Tables except for survey tables and uLogin used
Create Data	Radio Buttons for Choice Depending on user needs. Input text boxes depending on previous choices Submit Button	Access Pages create_data.php Access Database Tables All Tables except for survey tables and uLogin used
Use Case	Features	Components
Update Data	Radio buttons for user needs choice Dynamically created input boxes depending on user choice above. Submit Button	Access Pages update_data.php Access Database Tables All Tables except for survey tables and uLogin used
Remove Entry	Radio Buttons for Choice Depending on user needs. Dynamically created Checkboxes for removal of each data depending on choice above Submit Button	Access Pages remove_entry.php Access Database Tables All Tables except for survey tables and uLogin used

Reset Tables	Dropdown menu to select a table Submit Button	Access Pages remove_entry.php Access Database Tables All tables
Delete User	Dropdown Box with dynamic content Remove Submit Button	Access Pages remove_user.php
Reset Password	Dropdown Box with dynamic content for choice Password Textbox Update Submit Button	Access Pages reset_password.php
Use Case	Features	Components
Update User	Dropdown Box with dynamic content for choice Dropdown Box for level Choice Update Submit Button	Access Pages update_user.php
Create User	Dropdown box for level Two text input boxes for name and password Submit Button	Access Pages Create.users.php

Table 2.1 Use Case Features and Components Resources

2.2. Responsibilities:

The following table contains a listing of each Use Case, the team member who will be testing that Use Case, and finally the team member who developed the Use Case.

Use Case	Team Member(s) Tester	Team Member(s) Develop
Take Survey	Ben Mackin	Kou Vang
Login	Kou Vang	Ben Mackin
Logout	Ben Tovar	Ben Mackin
Generate Reports	Dave Diel	Khae Saturn
Print Reports	Jared Arbaugh	Khae Saturn
Create Custom Reports	Jared Arbaugh	Dave Diel
Create Data	Ben Mackin	Jared Arbaugh
Update Data	Ben Mackin	Ben Tovar
Remove Entry	Ben Tovar	Khae Saturn
Reset Tables	Ben Tovar	Kou Vang
Remove User	Khae Saturn	Dave Diel
Reset Password	Dave Diel	Ben Tovar
Update User	Ben Mackin	Jared Arbaugh
Create User	Khae Saturn	Dave Diel

Table 2.2 Programmers and testers responsibilities

All testing is to be preformed on gaia at the completion of the base line coding. All test cases will be followed explicitly, following any prerequisites that are required to sufficiently test that specific test case (see the Intercase Dependencies subsection of the test cases). No team member tests the Use Case they implement.

2.3 Schedule:

This subsection contains the testing schedule for Archway Project. It specifies the date on which testing will begin and the date that testing will be completed. At the end of testing on the sponsor's server the software will be ready for delivery.

Type of test	Date Start	Complete	Repair Start	Repair Complete
Single Pages	April 2 nd	April 9 th	April 9 th	April 16 th
Whole Program	April 16 th	April 23 rd	April 23 rd	April 30 th
Sponsor Server	April 24	April 30 th	April 24 th	May 5 th

Table 2.3 Tests Schedule

3. TEST DESIGN SPECIFICATION

3.1 Testing Approach:

The following documentation will be used to prepare the test design, case and procedure specifications. The tests are intended to verify the accuracy and completeness of the information in the documentation in the areas covered by the tests.

The testing approach will account for the following:

- Interface Testing
- Security Testing
- Recovering Testing
- Performance Testing
- Regression Testing
- Constraints

The test will be performed by entering test data into the web pages, and observing the displayed output. Validations will require that the database be queried to ensure that the proper data was created, retrieved, updated or deleted.

3.2 Feature or Combination of Features Not To Be Tested:

The following list of features will not be tested:

- Ability to cope with volume, load, and hardware faults. The tests case will not include testing the systems abilities to deal with multiple users and any hardware issues that may arise.
- Time related bugs like a session timeout. Once the session times out, the variables are lost, therefore the user must start over.
- Unanticipated error conditions. Although most errors are unanticipated, tests cases will not test for error conditions that may arise outside the DBMS.
- User interface inconsistency, where the users resolution or PC settings may cause some difference in appearance to the webpage.
- User-friendliness, although the system is designed to be easy to use, the computer experience of the end users can not be measured.

3.3 Environmental Needs:

The testing environment for the system will be done using Direct 6 Software Solution's account on gaia. The following necessities will be required in order to support the testing process.

Software

- Windows Server – The platform in which Archway will be installed on.
- PHP – The language used for developing the interface and user interaction between the user and the DBMS.
- MySQL – The database platform used for storing, creating, updating, and deleting the data.
- Unix – The platform Archway is being developed on.

Hardware

- Hard Disk Space – The disk size is sufficient enough that it can be used to store various amounts of data.
- Processor – The processor can handle multiple users' interactions.
- Ram – The memory will support multiple user interactions.

Communication

- Internet Browser – The application that will be used in order to access Archway interface.

3.4 Suspension / Resumption Criteria:

All tests will be run to completion once the test has begun. If the test is interrupted, then the testing must start over, and the problem will be recorded. If an error occurs that makes continuing with a test impossible then the cause of the error shall be reported, examined, and repaired as quickly as possible. Once the error is fixed, the test during which the error occurred will be restarted, and the test will be run in its entirety. In addition, if the affected component is used elsewhere, all related cases will need to be rerun.

3.5 Risks and Contingencies:

The following list provides some examples.

- If hardware problems occur on the sponsor's system which affect the systems availability, Direct 6 Software Solution will recommend that the sponsor upgrade their current hardware.
- If access to gaia for testing purposes is not possible, or not available according to the testing schedule, the team will use an alternate server on the CSUS system.
- If access to the entire CSUS network is not possible or not available, then testing will proceed on group member machine.
- If a team member failed to complete his/her assigned use case, then the team needs to get together and develop the uncompleted use case.

4. TEST SPECIFICATION

4.1 Test Procedures:

The testing procedures are listed for each use case. The tester will follow every step in the following test cases. If no errors occurred, then the tester will need to validate that the outcome is correct. If errors occur, a Software Problem Report will be written.

4.2 Test Procedure Conventions:

Each test case has given specific instructions describing the steps which need to be taken to complete the test cases. All test cases will start at the home page.

4.3 Test Data:

All test data is contained within the test cases below. The following are test cases for each use case described in this document.

4.4.1 Use Case: Take Survey:

This test case will emulate a user taking a survey.

STEPS:

1. Starting at the home page, click on the 'Survey' menu option.

At this point, the survey screen will appear, prompting the user to answer the questions.

4.4.1.1 Test Items

The following program files that are needed to perform this test:

- login.php
- menu.php
- main.php
- take_survey.php
- completed_survey.php

4.4.1.2 Input Specifications

STEPS:

1. Select answers to the various questions.
2. On the single answer questions, always choose the second choice.
3. On multiple answer questions, always choose two choices.
4. Click on submit.
5. Verify that a completion webpage comes up letting the user know they are done.
6. Verify that in the database, the answers chosen were the answers recorded.

4.4.1.3 Output Specifications

If everything worked, then the page displayed will state that the survey was successfully completed, and answers were recorded to the system.

4.4.1.4 Intercase Dependencies

The login test case must be completed prior to the take survey use case, as only logged in users can take surveys.

4.4.2 Use Case: Login:

4.4.2.1 Test Case #1

This use case will emulate a user logging into the system.

STEPS:

1. Load the Archway website.

4.4.2.1.1 Test Items

The program files that are needed to perform this test:

- login.php
- main.php
- menu.php

4.4.2.1.2 Input Specifications

STEPS:

1. At the login screen, enter:

Username: archer

Password: 1234

2. Click on the Log In button.

The main page will be loaded. Displayed on the left frame is a menu system for navigating through all of the components of the system. In the center frame will be the content for the selected page they are on, at first being only a welcome screen.

4.4.2.1.3 Output Specifications

None.

4.4.2.1.4 Intercase Dependencies

None

4.4.2.2 Test Case #2

This use case will emulate an attempt to enter an incorrect password when logging in.

STEPS:

1. Load the Archway website.

4.4.2.2.1 Test Items

The program files that are needed to perform this test:

- login.php
- main.php
- menu.php

4.4.2.2.2 Input Specifications

STEPS:

1. At the login screen, enter:

Username: admin

Password: WRONG

2. Click on the Log In button.

4.4.2.2.3 Output Specifications

The user will be returned to the login screen, with a new message stating that login failed because the username or password was incorrect.

4.4.2.2.4 Intercase Dependencies

None

4.4.3 Use Case: Log out:

This test case will emulate a user logging out of the system.

STEPS:

None

4.4.3.1 Test Items

The following program files that are needed to perform this test:

- header.php
- menu.php
- main.php

4.4.3.2 Input Specifications

STEPS:

1. From the current page, click on the Log Out link, which is located in the menu on the left frame.

You should be logged out of the system and taken back to the login page.

4.4.3.3 Output Specifications

If the logout was successful, then the page displayed will be the Login page, but with an added message stating that the user was successfully logged out.

4.4.3.4 Intercase Dependencies

The login test case must be completed prior to the log out use case, as only logged in users can log out of the system.

4.4.4 Use Case: Generate Reports:

This test case will emulate a user generating a report.

STEPS:

1. Starting at the home page, click on the 'Reports' menu item.
2. Now click on the 'Generate Reports' submenu item.

The generate reports page is now displayed, and awaits menu entry.

4.4.4.1 Test Items

The following program files that are needed to perform this test:

- login.php
- menu.php
- generate_reports.php
- generated_report.php

4.4.4.2 Input Specifications

STEPS:

1. Select the 'rancheria1' rancheria from the pull down menu.
2. Click on Submit.
3. Verify that all of the information is displayed

4.4.4.3 Output Specifications

If everything worked, then the page displayed will display all of the information in the database for rancheria1.

4.4.4.4 Intercase Dependencies

The login test case must be completed prior to the generate reports use case, as only logged in users can generate reports.

4.4.5 Use Case: Print Reports:

This test case will emulate a user printing a generated report.

STEPS:

1. After a report is generated, the user clicks on the 'Print Report' button.

Now the print dialog is displayed and ready to be filled out.

4.4.5.1 Test Items

The following program files that are needed to perform this test:

- login.php
- menu.php
- generate_reports.php
- generated_report.php

4.4.5.2 Input Specifications

STEPS:

1. Enter in any options for multiple copies, or other print settings.

We will assume that printing multiple copies, or changing other settings will work, as we are just using the standard print function for Windows.

2. Click on 'Print'.

4.4.5.3 Output Specifications

If everything worked, then a page will be printed out of the printer with the report information on it.

4.4.5.4 Inter-case Dependencies

The generate reports test case must be completed prior to the print reports use case, as in order to print a report, it must first exist.

4.4.6 Use Case: Create Custom Reports:

This test case will emulate a user creating a custom report.

STEPS:

1. Starting at the generated report page, click on the Reservation radio button
2. Then click on the Populate button to populate the dropdown box.
3. Choose Reservation and click on the following checkboxes:

- Waterways
- Roads

2. Click on Submit.

Now the custom report is displayed.

4.4.6.1 Test Items

The following program files that are needed to perform this test:

- login.php
- menu.php
- generate_reports.php
- generated_report.php
- custom_report.php

4.4.6.2 Input Specifications

None.

4.4.6.3 Output Specifications

If everything worked, then the page displayed will be a printer friendly report generated based on the Waterways and Roads for the chosen rancheria.

4.4.6.4 Intercase Dependencies

The generate reports test case must be completed prior to the custom reports use case, as in order to create a custom report, a base report must first exist.

4.4.7 Use Case: Create Data:

4.4.7.1 Test Case #1

This test case will emulate a user entering data in the system.

STEPS:

1. Starting at the home page, click on the 'Database' menu.
2. From here, click on the 'Add Data' submenu item.

Now the create data page is displayed and ready to be filled out.

4.4.7.1.1 Test Items

The following program files that are needed to perform this test:

- login.php
- menu.php
- create_data.php

4.4.7.1.2 Input Specifications

STEPS:

1. Select a table to edit

The user is then presented an appropriate form, with the correct number of fields based on the table selected.

2. Fill in all of the fields with appropriate data.
3. Click on Submit.

4.4.7.1.3 Output Specifications

The page displayed will state that the new information was successfully posted to the system.

4.4.7.1.4 Intercase Dependencies

The login test case must be completed prior to the create data use case, as in order to add information to the database a user must first be logged in with correct privileges.

4.4.7.2 Test Case #2

This test case will emulate a user attempting to enter not enough data into the system.

STEPS:

1. Starting at the home page, click on the 'Database' menu.
2. From here, click on the 'Add Data' submenu item.

Now the create data page is displayed and ready to be filled out.

4.4.7.1 Test Items

The following program files that are needed to perform this test:

- login.php
- menu.php
- create_data.php

4.4.7.2 Input Specifications

STEPS:

1. Select the 'Waterways' table to edit

The page is dynamically updated to include editing options for the Waterway table.

2. Fill in the first field with:

badWaterWay

3. Without entering any other data in, click on submit.

4.4.7.3 Output Specifications

The page will present an error message informing the user that they did not enter enough data, and to please try again.

4.4.7.4 Intercase Dependencies

The login test case must be completed prior to the create data use case, as in order to add information to the database a user must first be logged in with correct privileges.

4.4.8 Use Case: Update Data:

This test case will emulate a user who has administrator or super user privileges to update data in the database.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin

Password: admin

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Update Data' link.

The web page will now display a list of records in the database

4.4.8.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- update_data.php

4.4.8.2 Input Specifications

STEPS:

1. Select contact person for 'reservation 1'
2. Click on the 'Submit' button.

The web page will now display all the information for the contact person inside a textbox.

3. Go to the telephone textbox and enter '916-555-5555'.
4. Click on the 'Update Data' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.8.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.8.4 Intercase Dependencies

This test case requires that the add data test case be completed before this test can be performed.

4.4.9 Use Case: Remove Entry:

This test case will emulate a user who has administrator or super user privileges to delete data in the database.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin

Password: secret

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Delete Data' link.

The web page will now display a list of records in the database

4.4.9.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- delete_data.php

4.4.9.2 Input Specifications

STEPS:

1. Select contact person for 'reservation 1'
2. Click on the 'Delete Data' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.9.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.9.4 Intercase Dependencies

This test case requires that the add data test case be completed before this test can be perform.

4.4.10 Use Case: Reset Table:

This test case will emulate a user who has administrator or super user privileges to reset an individual table in the database.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin

Password: admin

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Reset Table' link.

The web page will now display a list of tables in the database

4.4.10.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- reset_table.php

4.4.10.2 Input Specifications

STEPS:

1. Select the table 'ccontactinfo'
2. Click on the 'Reset Table' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.10.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.10.4 Intercase Dependencies

This test case requires that the add data test case be completed before this test can be perform.

4.4.11 Use Case: Remove User:

This test case will emulate a user who has administrator privileges to delete a user from the database.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin

Password: secret

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Delete User' link.

The web page will now display a list of users in the database

4.4.11.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- delete_user.php

4.4.11.2 Input Specifications

STEPS:

1. Select 'user1' from the list
2. Click on the 'Delete User' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.11.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.11.4 Intercase Dependencies

This test case requires that the create user test case be completed before this test can be perform.

4.4.12 Use Case: Reset Password

This test case will emulate a user who has administrator privileges to reset a user's password.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin

Password: secret

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Reset User Password' link.

The web page will now display a list of users in the database

4.4.12.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- reset_user_password.php

4.4.12.2 Input Specifications

STEPS:

1. Select 'user1' from the list
2. Click on the 'Reset User Password' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.12.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.12.4 Intercase Dependencies

This test case requires that the create user test case be completed before this test can be perform.

4.4.13 Use Case: Update User:

This test case will emulate a user who has administrator privileges to change the user type.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin

Password: secret

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Update User Type' link.

The web page will now display a list of users in the database

4.4.13.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- update_user_type.php

4.4.13.2 Input Specifications

STEPS:

1. Select 'user1' from the list
2. Select the "Administrator" radio button
3. Click on the 'Update User Type' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.13.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.13.4 Intercase Dependencies

This test case requires that the create user test case be completed before this test can be performed.

4.4.14 Use Case: Create User:

This test case will emulate a user who has administrator privileges to create a new user.

STEPS:

1. Starting at the login page, enter the following username and password:

Username: admin
Password: secret

2. Click on the 'Login' button.

The web page will now display the administrator menu

3. Click on the 'Create User' link.

The web page will now display a form that contains textbox to capture all the required field for a user profile to be store in the database

4.4.14.1 Test Items

The following program files that are needed to perform this test:

- login.php
- verifyuser.php
- menu.php
- create_user.php

4.4.14.2 Input Specifications

STEPS:

1. Enter the following information

Username: user2
Password: secret
Level: 1

2. Click on the 'Create User' button.

The web page will display confirmation message that the operation was complete successfully. It will also display two options, back to main menu and make another update, for the next user action

4.4.14.3 Output Specifications

1. Click on the 'Back to Main Menu' link

The web page will now display menu for the administrator.

4.4.14.4 Intercase Dependencies

Not Applicable

5. SYSTEM TEST / REQUIREMENTS TRACEABILITY

5.1 System Test / Requirements Specification / Design Component Traceability Matrix:

STS	SRS	SDS	Use Case Names
4.4.1	3.1.1	5.1.1	Take Survey
4.4.2	3.1.2	5.1.2	Login
4.4.3	3.1.3	5.1.3	Logout
4.4.4	3.1.4	5.1.4	Generate Reports
4.4.5	3.1.5	5.1.5	Print Reports
4.4.6	3.1.6	5.1.6	Creating Custom Reports
4.4.7	3.1.7	5.1.7	Create Data
4.4.8	3.1.9	5.1.9	Update Data
4.4.9	3.1.10	5.1.10	Remove Entry
4.4.10	3.1.11	5.1.11	Reset Table
4.4.11	3.1.12	5.1.12	Remove User
4.4.12	3.1.13	5.1.13	Reset Password
4.4.13	3.1.14	5.1.14	Update User
4.4.14	3.1.15	5.1.15	Create User

Table 5.1 – Requirements Traceability Matrix

6. APPROVAL SIGNATURES

Affiliation	Name	Title	Signature	Date
CSUS	Dr. Bolan Jiang	Project Advisor	_____	_____
Direct 6	Ben Mackin	Project Manger	_____	_____
Direct 6	Dave Diel	Programmer	_____	_____
Direct 6	Kou Vang	Programmer	_____	_____
Direct 6	Ben Tovar	Programmer	_____	_____
Direct 6	Khae Satern	Programmer	_____	_____
Direct 6	Jared Arbaugh	Programmer	_____	_____

Appendix A - Software Problem Report Template

This section provides a sample template for reporting software problems that are discovered during the course of performing the test cases detailed in this document.

SOFTWARE PROBLEM REPORT Problem Report ID _____

PROGRAM _____ RELEASE _____ VERSION _____

REPORT TYPE SEVERITY ATTACHMENTS: Yes No
 Coding Error Documentation Fatal If yes, list attachments
 Design Error Hardware Serious _____
 Suggestion Query Minor _____

PROBLEM SUMMARY _____

CAN YOU REPRODUCE THE PROBLEM? (Y/N) ____

PROBLEM AND HOW TO REPRODUCE IT _____

SUGGESTED FIX (optional) _____

REPORTED BY _____ DATE __/__/__

Items Below Are For Use Only By the Development Team

FUNCTIONAL AREA _____ ASSIGNED TO _____

COMMENTS _____

STATUS: PRIORITY
 Open Closed High Medium Low

RESOLUTION: RESOLUTION VERSION NO: _____
 Pending Deferred Withdrawn by reporter
 Fixed As designed Need more info
 Irreproducible Can't be fixed Disagree with suggestion

RESOLVED BY _____ DATE __/__/__

RESOLUTION TESTED BY _____ DATE __/__/__

TREAT AS DEFERRED: Yes No