

Guide to Preparing the SYSTEM TEST SPECIFICATION

© R. Buckley

CSc 190 Senior Project

Department of Computer Science - College of Engineering and Computer Science
California State University, Sacramento

Version 10.18.06

1. INTRODUCTION

This is the System Test Specification (STS) document for the *<name of the project>* project sponsored by *<name of sponsor>*.

This project is being undertaken by the *<name of team>* 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 (*if there is more than one person serving as sponsor, list each sponsor*):

Name

Title

Company *or* Organization name

Contact information (phone number and Email address)

<NAME OF TEAM> DEVELOPMENT TEAM

List of team member names

Team contact information (phone number and Email address)

The remainder of this section is intended to introduce the reader to the document and to serve as a high level, executive-like summary to the document.

1.1 Purpose

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

1.2 Scope

The plan contains a list and brief description of the use cases to be tested and the software components associated with each test case. The plan also provides a schedule for the testing and the assignment of team members to their respective testing tasks. The process for documenting 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, the

description each test case needed to fully test the use case, and the test procedures, or 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.

All references to the software technology used should be included. For example:

ASP. Active Server Pages (*would require a definition*)
HTML. Hypertext Markup Language (*would require a definition*)
XML. Extensible Markup Language (*would require a definition*)

NOTE. *You will need to indicate with the appropriate symbol whether the products you reference are “trademarked” (using ® or ™) or “copyrighted” (using ©).*

1.3.1. Definitions. The following are examples of what might be included in this section:

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

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.

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

1.3.2. Acronyms (*definition and/or explanations may need to be included for those acronyms that are not commonly used or understood*)

1.3.3. Abbreviations (*again, definition and/or explanations may need to be included for those acronyms that are not commonly used or understood*). The following are examples of what might be included in this section:

MGT. Management
LRN. Learning

STS. System Test Specification
STR. System Test Report

1.4 References. This subsection serves the same purpose as a bibliography. If any documents were used in the preparation of the document, the formal bibliographic information should be specified here. A reference to a specific bibliographic entry should be made at the location in the document where information from that specific source is used. Included here should be all documents provided by the sponsor.

Use the Modern Language Association's citation rules and conventions for acknowledging sources used in a preparing the document. One of the many Web links that contains MLA citation examples is:

http://www.library.cornell.edu/newhelp/res_strategy/citing/mla.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.

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.

The intent of system test plan is as follows:

- To specify the activities required to prepare for and conduct the system test.
- To clearly indicate 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. Brief description of the software product followed by the summary table below.

The following table contains a listing of the use cases, the system's features associated with each use case along with its files and database tables.

<i>USE CASE</i>	<i>FEATURES</i>	<i>COMPONENTS</i>
Use Case 1	Feature 1 Feature 2 Feature 3	<i>List of files and data base tables associated with this Use Case</i>
Use Case 2	Feature 1 Feature 2 Feature 3	<i>List of files and data base tables associated with this Use Case</i>
Etc.		

2.2 Responsibilities. The following table contains a listing each Use Case, the team member assigned to testing the Use Case, and whatever set-up is necessary for testing the Use Case.

<i>USE CASE</i>	<i>TEAM MEMBER</i>	<i>SET-UP</i>
Use Case 1	Team member 1	<i>(indicate the team member of members responsible for all set-up needed prior to the beginning of testing on each of the Use Cases)</i>
Use Case 2	Team member 2	
Use Case 3	Team member 3	
Use Case 4	Team member 4	
Use Case 5	Team member 5	
etc.		

NOTE. *Projects are expected to have at least as many major use cases as there are team members. In assigning the development work, each team member is assigned to implement – either individually or with another team member. The testing assignment should be made in such a way that team members DO NOT test the use cases that they implemented.*

2.3 Schedule. This subsection contains the testing schedule for each Use Case, specifying the date on which testing should begin and the date that testing should be expected to be completed. The schedule should include time for resolving all problems reported during the testing. In addition, if the software has been developed and tested in a development environment, a plan and schedule should be included for testing in the sponsor's operational environment. At the end of the testing cycle the software should be ready for delivery.

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.

3.1 Testing Approach. The following documentation was used to prepare the test design, case and procedure specifications. The tests are intended to verify the accuracy and comprehensiveness of the information in the documentation in those areas covered by the tests.

<The project's software requirements specification document>

<The project's software design specification document>

The testing approach should account for the following, if appropriate:

- Conversion testing.
- Job stream testing.
- Interface testing.
- Security testing.
- Recovery testing.
- Performance testing.
- Regression testing.
- Comprehensiveness.
- Constraints.

Typically, the tests will be performed by entering test data into the various web pages, and observing the output displayed. In most cases, validation 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. As indicated by the title, this section describes what features of the software are not being tested. The following list provides some examples:

- Ability to cope with volume, load, and hardware faults. The test cases will not include testing the systems ability to deal with multiple users and any hardware issues that may arise.
- Time related bugs like a session time out. Once the session times out the session variables are lost, therefore the user must start over.
- Unanticipated error conditions. Although most errors are unanticipated, the test cases will not test for error conditions that may arise outside of the DBMS.
- User interface inconsistency, where the user's resolution or PC settings may cause some differing in appearance of the web pages.

3.3 Environmental Needs. Specify both the necessary and desired prosperities of the test environment. This should include the physical characteristics of the facilities; the hardware, the communications and system software, the mode of use (stand-alone or distributed), and any other software or supplies needed to support the test. Identify and special test tools (or any other testing needs) that will be needed. For those environmental needs that are not available to the project team, indicate who is responsible and how and when these needs will be met.

Testing typically is done using the development environment set-up by the team. If this is the case, specify both the development environmental needs and the operational – sponsor owned – environmental needs.

3.4 Suspension / Resumption Criteria. The following is an example of such criteria: All tests must 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 logged. If an error occurs that makes continuing with a test impossible, 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 attempt will be made to run the test in its entirety. In addition, if the affected component is used elsewhere, all related test cases must be rerun (this is an example of regression testing).

3.5 Risks and contingencies. This subsection contains a list of the possible risks that are most likely to affect the testing schedule and the ability to deliver the software according to schedule. The following list provides some examples.

- If the testing schedule is significantly impacted by system failure, *<specify what action the team will take>*.
- If hardware problems impact the system availability, *<specify what action the team will take>*.
- If access to the operational environment for testing purposes is not possible or not available according to the testing schedule, *<specify what action the team will take>*.
- If *<list any other possible risks and the team's intended response>*.

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.

- 4.1 Test Procedures. The testing procedures (steps) will be listed for each test case. The operator will follow each step until complete. If no errors have occurred, then the test operator will need to validate that the outcome is correct by examining the database. If errors occur, a Software Problem Report (SPR) will be written (a sample SPR template is included in appendix A). Each SPR will then be analyzed and a determination made as to whether a software change is warranted. If so, the test cases will be re-run after the changes have been implemented and the software appropriately updated.
- 4.2 Test Procedure Conventions. Each test case will give specific instructions describing what steps need to be taken to complete the test case. All test cases will start at the home page. Also, provided will be the expected results. This subsection includes the software installation instructions.
- 4.3 Test Data. This subsection should specify all the test data that is needed to run the test cases associated with each use case. This specification should include the specific data that must be set into the database along with instructions on to how this is to be done. For example, the data base would need to be appropriately set in order to test for user authentication for a login use case. In addition, permissions would need to appropriately set in order to test for various categories of restricted and unrestricted user access.

If appropriate, a cross reference listing should be provided of those data sets associated with specific use case tests.

- 4.4 Use Case 1. *List the software features to be tested and include a brief description for each feature. Indicate the path of links from the home page to the page (or pages) being tested. If access to the page to be tested requires login, provide the instructions for login.*

4.4.1 Test Case 1.

EXAMPLE.

This test case will emulate a student entering a resume for the first time.

PROCEDURE.

STEPS:

1. Starting at the home page, click on the 'Student' link.
2. To create a new login, so click on the link to register.
3. Enter the following information:

First Name: John

Last Name: Doe

E-Mail Address: john_doe@lost.com

Password: 123
Confirm Password: 321

4. Click on submit.

The web page will not continue because the password and confirmed password don't match. At the top of the form it will state "Passwords do not match".

5. Enter '123' as the new Confirmed Password.
6. Click on submit.

Now the resume form is displayed and ready to be filled out.

4.4.1.1 Test Items

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

- StudentLogin.jsp
- Register.jsp
- CreateUser.java
- ResumeForm.html
- SubmitResume.java
- VerifyResumeInfo.html
- FinalizeResume.java
- ConfirmSubmission.html

4.4.1.2 Input Specifications

STEPS:

1. Enter the following data into the resume form:

Address: 123 Sesame St

City: Hollywood

State: CA

Zip: 12345

Phone: 123 456 7890

Residency Status: US Citizen

CSUS Major: Computer Science

Graduation Date: Fall 2003

G.P.A.: 3.40

Objective: Get a real job.

Experience Related to Major

Title: Web Developer

Company: Some WEB Co.

Dates: June 1996 – Current

Description of Duties: Maintaining and enhancing a web environment.

Skill/Relevant Coursework: Data Structures, Java, Operating Systems

2. Click on all of the help links to verify that they are functional.
3. Click on submit.
4. Verify that all of the information is displayed
5. Click on 'Continue'.

4.4.1.3 Output Specifications

If everything worked, then the page displayed will state that the resume was successfully posted to the system.

4.4.1.4 Intercase Dependencies

Indicate which test cases, if any, must be executed prior to this test case. Summarize the nature of these dependencies.

4.4.2 Test Case: Viewing the administrator's page

The administrator page displays all student names that have their resume active in the system. It also shows statistics on the searches that have been performed. To view the administrator's page:

STEPS:

1. Click on the administrator link on the home page.
2. At the login page, enter:

Username: cici

Password: blah

3. Click on the Sign-In button.

4.4.2.1 Test Items

The program files that are needed to perform this test:

- AdminLogin.jsp
- UserLogin.java
- InfoGenerator.java
- Administration.html

4.4.2.2 Input Specifications

Click the Sign-In button

The administrator's page will be loaded. Displayed on the left frame will be a list of all of the students that are in the system. The students will be organized by major. In the center frame will be the search statistics. One section will be the top 5 majors that have been searched, along with the number of times that the searches have occurred. And to the other section will be the top 10 skills that have been searched for, along with the number of times the searches have occurred.

Note the top skill and major along with their count. Next, verify that the database shows the same counts.

STEPS:

1. Open the database ACM, by < include instructions here>
2. Login in by typing in User name: Admin and Password: admin.
3. Click on the Tools tab at the top of the window
4. Enter into the Enter Xpath field: <instructions provided>
5. Click on the submit button.

Now verify that the count displayed matches the count that was noted from the Administrator page.

4.4.2.3 Output Specifications

Once the top skill and major has been verified, it can be assumed that all of the other statistics will be correct. This is because the same logic is used for all of them to come up with the numbers.

Note. The next Use Case to be tested would be numbered 4.4.3. If there were 5 test cases associated with this Use Case, they would be numbered 4.4.3.1 through 4.4.3.5.

5. SYSTEM TEST / 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). The appropriate section and its title in each document are provided.

5.1 System Test / Requirements Specification / Design Component Traceability Matrix

System Test	Requirement Specification	Design Component
4.3 Use Case 1	X.X Specification title	Y.Y Design component title(s)

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.

Each signatory should be identified by name, title and affiliation. A signature line should be provided along with a date line.

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

- ☐ Coding Error ☐ Documentation
☐ Design Error ☐ Hardware
☐ Suggestion ☐ Query

SEVERITY

- ☐ Fatal
☐ Serious
☐ Minor

ATTACHMENTS: ☐ Yes ☐ No
If yes, list attachments

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:

- ☐ Open ☐ Closed

PRIORITY

- ☐ High ☐ Medium ☐ Low

RESOLUTION:

- ☐ Pending ☐ Deferred ☐ Withdrawn by reporter
☐ Fixed ☐ As designed ☐ Need more info
☐ Irreproducible ☐ Can't be fixed ☐ Disagree with suggestion

RESOLUTION VERSION NO: _____

RESOLVED BY _____

DATE __/__/__

RESOLUTION TESTED BY _____

DATE __/__/__

TREAT AS DEFERRED: ☐ Yes ☐ No