Spring 2015

Software Test Specification (STS)

Version 1.0

CMIS 495

Andrew Bernier, Jason Foley, Brett Fry, John Livingston, Christopher Overby, Ina-Marie Sanabria, Charles Schultz

**University of Maryland University College**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 1/21/2015 | 1.0 | STS 1.0 | Group D |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Background 1

1.3 Scope 1

1.4 Definitions, Acronyms, and Abbreviations 1

1.5 Test items 2

2. References 2

3. Test Environment 2

3.1 Hardware 2

3.2 Software 2

3.3 Communications 2

3.4 Tools 3

3.5 Data 3

4. Architectural Context Diagram Mappings 3

4.1 User Interface 3

4.2 Services 4

4.3 Domain Objects 5

5. Traceability Matrix 6

6. Test Case Specifications and Approach 7

6.1 User Interface Test Cases 8

6.1.1 Specification ID: WEBSTORE-UI-01 8

6.1.2 Specification ID: WEBSTORE-UI-02 9

6.1.3 Specification ID: WEBSTORE-UI-03 10

6.1.4 Specification ID: WEBSTORE-UI-04 11

6.1.5 Specification ID: WEBSTORE-UI-05 12

6.1.6 Specification ID: WEBSTORE-UI-06 13

6.1.7 Specification ID: WEBSTORE-UI-07 14

6.1.8 Specification ID: WEBSTORE-UI-08 15

6.1.9 Specification ID: WEBSTORE-UI-09 16

6.1.10 Specification ID: WEBSTORE-UI-10 17

6.1.11 Specification ID: WEBSTORE-UI-11 18

6.1.12 Specification ID: WEBSTORE-UI-12 19

6.2 Service Test Cases 20

6.2.1 Specification ID: WEBSTORE-DS-01 20

6.2.2 Specification ID: WEBSTORE-DS-02 21

6.2.3 Specification ID: WEBSTORE-DS-03 22

6.3 Domain Object Test Cases 23

6.3.1 Specification ID: WEBSTORE-DO-01 23

6.3.2 Specification ID: WEBSTORE-DO-02 24

6.3.3 Specification ID: WEBSTORE-DO-03 24

6.3.4 Specification ID: WEBSTORE-DO-04 25

6.3.5 Specification ID: WEBSTORE-DO-05 26

7. Training Needs 26

8. Risks and Contingencies 27

# 1. Introduction

## 1.1 Purpose

The objective of this document is to provide a testing framework for the three primary levels of the WEBSTORE. This framework describes both black-box (component requirements) and white-box (algorithmic) tests that can be used to validate both requirements and design.

## 1.2 Background

The WEBSTORE is used to assist the management/employees of a brick-and-mortar store with customer transactions, taxes, and inventory management. Additionally it will ease the overhead of managing different sets of customers. By developing software to automate many of these repetitive tasks, higher efficiencies can be achieved and the organization can increase revenue by developing an online presence. This document and subsequent testing framework are used to assist in the development of a quality software product that provides this service.

## 1.3 Scope

This testing documentation is scoped to the test cases specific for the three modules of the architectural context diagram:

1. User Interface – Section 4.1
2. Domain Services – Section 4.2
3. Domain Objects – Section 4.3

Other services in the system, such as data persistence and database configuration, are out of scope for this document.

## 1.4 Definitions, Acronyms, and Abbreviations

For the scope of this document, the following definitions and lexicon should be noted.

* *concurrent process:*  process that is always running.
* *design entity:* An element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced
* *domain object (DO):* Something that holds information for a specific function.
* *Java Persistence API (JPA)*: Is a Java programming language application programming interface specification that describes the management of relational data in applications using Java Standard and Enterprise Editions.
* *Java Virtual Machine (JVM)*: An abstract computing machine.
* *module:* A well-defined component of a software system or part of a system that provides a set of services to other modules.
* *domain service (DS)* : Something that performs business logic.
* *System on Chip (SOC)*: An integrated circuit that integrates all components of a computer or other electronic system into a single device.
* *user interface (UI)*: The graphical element that lets users interface with the system.
* *workflow*: The series of activities that are necessary to complete a task

## 1.5 Test items

All major items that make up the WEBSTORE will be tested during the system test. The versions to be tested will be placed in the appropriate libraries by the configuration administrator. The administrator will also control changes to the versions under test and notify the test group when new versions are available.

The following documents will provide the basis for defining correct operation:

1. Software Requirements Specification (SRS) for WEBSTORE
2. Software Design Document (SDD) for WEBSTORE

# 2. References

References for the software test specification include:

1. IEEE std. 829-1998 – IEEE Standard for Software Test Documentation
2. Software Requirements Specification (SRS) for WEBSTORE
3. Software Design Document (SDD) for WEBSTORE

# 3. Test Environment

The testing environment for the WEBSTORE requires minimal setup. Since this software has limited requirements and interaction with other components, a simple environment for testers should be relatively easy to accommodate.

### 3.1 Hardware

Any modern computer system should work. Since resource requirements are minimal the application could run on a computer, SoC, or ARM device with RAM (128MB or more useable) and physical storage (at least 500MB post-operating system install) that can accommodate the WEBSTORE software and stored data. The computer should also include a keyboard, display (not required), and mouse for optimal results testing/viewing the software during testing.

### 3.2 Software

The WEBSTORE is platform independent; therefore it can run on any system that can run Java (JVM) (Linux, Solaris, Unix, or Windows operating systems). Since Java is the foundation of the application/software it is expected that the computer(s) provided for testing are running an operating system that supports Java (JVM). In addition, the WEBSTORE should be installed on the server computer and verified for operability by accessing the WEBSTORE on a client computer.

### 3.3 Communications

As specified in section 3.1.4 of the SRS, external communications requirements are not applicable (N/A).

### 3.4 Tools

Testing of the WEBSTORE requires a usable computing environment (defined in sections 1.5.1 and 1.5.2). Since the API and domain object layers of the software are not designed to be public, their access is limited to testers. As a result, the primary method of testing is the WEBSTORE user interface itself and white-box testing done by the developers. To test data entry into the database across the three modules, a MySQL tool, such as MySQL Admin or other equivalent RDBMS management tool can be used. The tester should have full administrative rights to the testing database in order to access both the database and fields in which the domain objects are stored.

### 3.5 Data

Testers of the WEBSTORE should be prepared with the following data in order to complete the use cases in section 4 of the STS:

1. Site Navigation
   1. Check all links and buttons
   2. Ensure all functions work and transition is user friendly
2. Test database connectivity
   1. Ensure database is running
   2. Ensure Site can access database
3. Shopping Cart
   1. Test ability to add to cart
   2. Test ability to remove from cart
   3. Test ability to update cart
   4. Test ability to clear contents of cart
4. Checkout
   1. Test names (first name, middle name (middle initial), last name)
   2. Test addresses (street, city, state (region), country, and zipcode)
   3. Test phone numbers (valid and invalid. Due to difference in country specific formats will accept any numerical value as long as no letters are encountered)
   4. Test credit card information (valid and invalid formats)
5. WEBSTORE Data
   1. Ensure data is stored in proper database tables

The set of data listed above will accommodate the entirety of test cases across the user interface, service, and domain object cases listed in this document.

# 4. Architectural Context Diagram Mappings

## 4.1 User Interface

This module is for users of the WEBSTORE. They are able to use the system via this interface exclusively. Access to this user interface does not require any external interface, other than via the keyboard and mouse, and is a single-user set of views.



Figure 4.1.1 Architectural Context Diagram – User Interface

## 4.2 Services

This module is used to house the business logic for processes that the WEBSTORE needs to conduct where domain objects are involved, such as looking up items/products, customers, and transactions. The service module provides output to the user interface, and input from the user interface is used to manage domain objects where appropriate. Persistence of the domain objects to the database is also managed by the service component.



Figure 4.2.1 Architectural Context Diagram – Services

## 4.3 Domain Objects

This module covers all entities that hold data about a particular function in the WEBSTORE system. These objects are referred to as domain objects.



Figure 4.3.1 Architectural Context Diagram – Domain Objects

# 5. Traceability Matrix

The traceability matrix provides a linkage between the initial software requirements (SRS) and software design (SDD) to the testing (STS) use cases defined below. Additionally, the matrix will be used to validate successful completion (pass/fail) for each test.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Description or Use Case** | **Software Req.** | **Test Case** | **Pass/Fail** |
| User Interface | Home Page |  | 6.1.1 |  |
| User Interface | Database Access |  | 6.1.2 |  |
| User Interface | Site Navigation |  | 6.1.3 |  |
| User Interface | Shopping Cart Page |  | 6.1.4 |  |
| User Interface | Shopping Cart (Add) |  | 6.1.5 |  |
| User Interface | Shopping Cart (Update) |  | 6.1.6 |  |
| User Interface | Shopping Cart (Remove) |  | 6.1.7 |  |
| User Interface | Shopping Cart (Clear) |  | 6.1.8 |  |
| User Interface | Checkout |  | 6.1.9 |  |
| User Interface | Checkout (Process valid information) |  | 6.1.10 |  |
| User Interface | Checkout (Handle invalid information) |  | 6.1.10 |  |
| User Interface | Checkout (Cancel) |  | 6.1.11 |  |
| Services | Cart |  | 6.2.1 |  |
| Services | addItem |  | 6.2.1 |  |
| Services | CartItems() |  | 6.2.2 |  |
| Services | getProduct |  | 6.2.2 |  |
| Services | removeProduct |  | 6.2.2 |  |
| Services | incrementQuantity |  | 6.2.2 |  |
| Services | decrementQuatity |  | 6.2.2 |  |
| Services | Customer |  | 6.2.3 |  |
| Services | getCustomer |  | 6.2.3 |  |
| Domain Objects | Validate Customer (name) |  | 6.3.1 |  |
| Domain Objects | Validate Customer (Credit Card) |  | 6.3.2 |  |
| Domain Objects | Invalidate Customer (name) |  | 6.3.3 |  |
| Domain Objects | Invalidate Customer (Credit Card) |  | 6.3.4 |  |
| Domain Objects | Validate Order |  | 6.3.5 |  |

# 6. Test Case Specifications and Approach

Test cases are provided for the three core modules of the WEBSTORE, covered in section 4 of this document. These tests constitute the completion of steps to validate inventory and algorithmic design (white-box), as well as requirements-bound (black-box) processes. Overall, the test case specifications are a derivative of both the SRS and SDD documents for the provided for the WEBSTORE engineering process.

The test personnel will use the system documentation to prepare all test design, case, and procedure specifications. This approach will verify the accuracy and comprehensiveness of the information in the documentation in those areas covered by the tests, as well as ensuring that the tests represent the production use of the system.

In order to ensure privacy, all test data extracted from production files will have privacy-sensitive fields changed. The pass/fail criteria is that the desired outcome must be met /satisfied without any bugs or unhandled errors.

Note that the address used for the web store throughout these test cases is [www.xstore.com](http://www.xstore.com). This is a placeholder; the tester should substitute the correct address of the web store in its current environment.

## 6.1 User Interface Test Cases

These cases are designed for interactive testing of the WEBSTORE through the user interface (UI). The data to be used for testing will be entering various values to include values of different lengths and testing to make sure each field handles invalid or empty input, as well as validating the UI elements (such as buttons) function as expected.

### 6.1.1 Specification ID: WEBSTORE-UI-01

**Objective**: The objective of this test case is to provide a black-box test for the Home Page portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user opening the Web Store

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

None

### 6.1.2 Specification ID: WEBSTORE-UI-02

**Objective**: The objective of this test case is to provide a black-box test for the database connectivity portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a tester opening the test.jsp to verify database is populated and working.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)/test.jsp
4. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. The test.jsp page should open without any issues and output contents of the category table in the database.
4. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

None

### 6.1.3 Specification ID: WEBSTORE-UI-03

**Objective**: The objective of this test case is to provide a black-box test for the shopping portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user navigating the site and browsing different categories of products.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Click on each of the categories listed on the home page.
5. Click all links on the Home page
6. Click on the checkout button
7. Click on view cart button
8. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. All categories should open without issues
5. All links should be functional and result in no errors or crashes
6. Takes user to checkout process
7. Takes user to cart with correct items, if any, listed
8. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

None

### 6.1.4 Specification ID: WEBSTORE-UI-04

**Objective**: The objective of this test case is to provide a black-box test for the Cart portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user opening the cart page.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Click on “view cart” button
5. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Takes user to cart with correct items if any listed
5. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

None

### 6.1.5 Specification ID: WEBSTORE-UI-05

**Objective**: The objective of this test case is to provide a black-box test for the Cart portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user adding items in their shopping cart.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Click on each of the movies category
5. Select movie title “Die Hard”
6. Click the “add to cart” button
7. Navigate back to the Home page
8. Click on the books category
9. Select “Oracle/PL Programming”
10. Click “add to cart”
11. Click “view cart”
12. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Category “Movies” should open without issues
5. Item should be added to cart and showing at top of page
6. Category “Books” should open without issues
7. Web Store Home page should open without issues and cart should show item
8. Item should be added to cart
9. Cart should display with items added.
10. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, and WEBSTORE-UI-04 must be completed first in order to initiate a workflow.6.1.5 Specification ID: WEBSTORE-UI-05

### 6.1.6 Specification ID: WEBSTORE-UI-06

**Objective**: The objective of this test case is to provide a black-box test for the Cart portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user removing items in their shopping cart.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Click on the books category
5. Select “Oracle/PL Programming”
6. Click “add to cart”
7. Click on “view cart”
8. Click “remove” button next to Oracle/PL Programming
9. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Category “Books” should open without issues
5. Web Store Home page should open without issues and cart should show item
6. Item should be added to cart
7. Cart should display with items added.
8. Item should be removed from the cart
9. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, and WEBSTORE-UI-05 must be completed first in order to initiate a workflow.6.1.6 Specification ID: WEBSTORE-UI-06

### 6.1.7 Specification ID: WEBSTORE-UI-07

**Objective**: The objective of this test case is to provide a black-box test for the Cart portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user updating items in their shopping cart.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Click on the books category
5. Select “Oracle/PL Programming”
6. Click “add to cart”
7. Click on “view cart”
8. Click inside “quantity” input box and enter a valid number (e.g. “2”, “10”)
9. Click “update” button
10. Click inside “quantity” input box and enter an invalid number (e.g. “-5”, “abc”)
11. Click “update” button
12. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Category “Books” should open without issues
5. Web Store Home page should open without issues and cart should show item
6. Item should be added to cart
7. Cart should display with items added.
8. Input box will allow user input
9. Item quantity should be updated in cart
10. Input box should allow user input
11. Item quantity will display error message and revert to previous quantity
12. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, and WEBSTORE-UI-05 must be completed first in order to initiate a workflow.6.1.7 Specification ID: WEBSTORE-UI-07

### 6.1.8 Specification ID: WEBSTORE-UI-08

**Objective**: The objective of this test case is to provide a black-box test for the Cart portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user clearing all items in their shopping cart.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Click on the books category
5. Select “Oracle/PL Programming”
6. Click “add to cart”
7. Click on “view cart”
8. Click on “Clear cart” button
9. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Category “Books” should open without issues
5. Item should be added to cart
6. Cart should display with items added.
7. Cart should clear all items and empty the cart.
8. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, and WEBSTORE-UI-05 must be completed first in order to initiate a workflow.6.1.8 Specification ID: WEBSTORE-UI-08

### 6.1.9 Specification ID: WEBSTORE-UI-09

**Objective**: The objective of this test case is to provide a black-box test for the Checkout portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user opening adding items and opening Checkout page.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Navigate to categories and add items to cart
5. Click on “view Checkout” button
6. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Items should be added to cart
5. Takes user to Checkout page with correct items if any listed
6. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, and WEBSTORE-UI-05 must be completed first in order to initiate a workflow.6.1.9 Specification ID: WEBSTORE-UI-09

### 6.1.10 Specification ID: WEBSTORE-UI-10

**Objective**: The objective of this test case is to provide a black-box test for the Checkout portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user inputting valid information during checkout.

**Input Specifications**

1. Follow steps 1-5 of WEBSTORE-UI-10 test case
2. Click on First Name data entry box
3. Enter “John”
4. Click on Last Name data entry box
5. Enter “Doe”
6. Click on Street data entry box
7. Enter “CMR 469 Box 1368”
8. Click on State data entry box
9. Enter “APO”
10. Click on State/Region data entry box
11. Enter “AE”
12. Click on Country data entry box
13. Enter “USA”
14. Click on Zip Code data Entry Box
15. Enter “09227”
16. Click on Phone Number data entry box
17. Enter “314-483-4606”
18. Click on Credit Card Number data entry box
19. Enter “1111 222233 33444”
20. Click “Pay”

**Output Specifications**

1. Program should execute as anticipated.
2. Cursor appears in First Name data entry box
3. “John” appears in First Name data entry box and remains throughout the test
4. Cursor appears in Last Name data entry box
5. “Doe” appears in Last Name data entry box and remains throughout the test
6. Cursor appears in Street data entry box
7. “CMR 469 Box 1368 Street data entry box and remains until changed or test complete
8. Cursor appears in City data entry box
9. “APO” appears in the City data entry box and remains until changed or test complete
10. Cursor appears in State/Region data entry box
11. “AE” appears in the State/Region data entry box and remains until changed or test complete
12. Cursor appears in Country data entry box
13. “USA” appears in the Country data entry box and remains until changed or test complete
14. Cursor appears in Zip Code data entry box
15. “09227” appears in Zip Code data entry box and remains until changed or test complete
16. Cursor appears in Phone Number data entry box
17. “314-483-4606” appears in Phone Number data entry box and remains throughout the test
18. Cursor appears in Credit Card Number data entry box
19. “1111 222233 33444” appears in Credit Card Number data entry box and remains until changed or test complete
20. Page will display showing user information and confirmation number

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, WEBSTORE-UI-05, and WEBSTORE-UI-09 and must be completed first in order to initiate a workflow.6.1.10 Specification ID: WEBSTORE-UI-10

### 6.1.11 Specification ID: WEBSTORE-UI-11

**Objective**: The objective of this test case is to provide a black-box test for the Checkout portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user inputting invalid information during checkout.

**Input Specifications**

1. Follow steps 1-5 of WEBSTORE-UI-10 test case
2. Click on First Name data entry box
3. Enter “”
4. Click on Last Name data entry box
5. Enter “Doe”
6. Click on Street data entry box
7. Enter “CMR 469 Box 1368
8. Click on State data entry box
9. Enter “APO”
10. Click on State/Region data entry box
11. Enter “AE”
12. Click on Country data entry box
13. Enter “-1”
14. Click on Zip Code data Entry Box
15. Enter “09227”
16. Click on Phone Number data entry box
17. Enter “314-483-4606”
18. Click on Credit Card Number data entry box
19. Enter “1111 222233 33444”
20. Click “Pay”

**Output Specifications**

1. Program should execute as anticipated.
2. Cursor appears in First Name data entry box
3. “ ” appears in First Name data entry box and remains throughout the test
4. Cursor appears in Last Name data entry box
5. “Doe” appears in Last Name data entry box and remains throughout the test
6. Cursor appears in Street data entry box
7. “CMR 469 Box 1368 Street data entry box and remains until changed or test complete
8. Cursor appears in City data entry box
9. “APO” appears in the City data entry box and remains until changed or test complete
10. Cursor appears in State/Region data entry box
11. “AE” appears in the State/Region data entry box and remains until changed or test complete
12. Cursor appears in Country data entry box
13. “-1” appears in the Country data entry box and remains until changed or test complete
14. Cursor appears in Zip Code data entry box
15. “09227” appears in Zip Code data entry box and remains until changed or test complete
16. Cursor appears in Phone Number data entry box
17. “314-483-4606” appears in Phone Number data entry box and remains throughout the test
18. Cursor appears in Credit Card Number data entry box
19. “1111 222233 33444” appears in Credit Card Number data entry box and remains until changed or test complete
20. Page will display validation errors for First Name and Country data

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, WEBSTORE-UI-05, and WEBSTORE-UI-09 and must be completed first in order to initiate a workflow.6.1.11 Specification ID: WEBSTORE-UI-11

### 6.1.12 Specification ID: WEBSTORE-UI-12

**Objective**: The objective of this test case is to provide a black-box test for the Checkout portion of the user interface as required by section 3.1.1 of the SRS.

**Test Items**: This test will emulate a user cancelling order on Checkout page.

**Input Specifications**

1. Locate browser icon located on the Desktop or navigation menu.
2. Click on the program to launch the web browser
3. Navigate to [www.xstore.com](http://www.xstore.com)
4. Navigate to categories and add items to cart
5. Click on view “Checkout” button
6. Click “Cancel” button
7. Close web browser

**Output Specifications**

1. Program should be on the desktop or navigation menu
2. Program should run/execute successfully without any bugs or crashes
3. Web Store Home page should open without issues
4. Items should be added to cart
5. Takes user to Checkout page with correct items if any listed
6. Web Store Home page should display without checking out.
7. Web browser should close successfully

**Environmental Needs**

1. Computer or ARM device
2. Operating system configured to include WEB STORE.
3. Java and Glassfish installed.
4. Sufficient disk space to store increasing amounts of data.
5. Monitor
6. Keyboard
7. Mouse

**Special Procedural Requirements**

WEB STORE services will require appropriate system permissions so it may be launched.

**Inter-case Dependencies**

WEBSTORE-UI-01, WEBSTORE-UI-03, WEBSTORE-UI-04, WEBSTORE-UI-05, and WEBSTORE-UI-09 and must be completed first in order to initiate a workflow.6.1.12 Specification ID: WEBSTORE-UI-12

## 6.2 Service Test Cases

These cases are designed to test the application-programming interface (API) of the WEBSTORE, and can be performed either by a user testing team or through automated testing.

### 6.2.1 Specification ID: WEBSTORE-DS-01

**Objective**: The objective of this test case is to provide a white-box (process) test for the add to cart algorithm to add an item to cart through the service.

**Test Items:** Cart (Object), addItem, “Add” button (See SDD: , sec 6.1.5 pg. 14)

**Input Specifications**

1. Navigate to product you like
2. Click “add” button

**Output Specifications**

1. Product page will display and user will be able to add item to cart
2. addItem service will show a new domain object being created.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

More than one item may want to be added so user might have to make multiple clicks

**Inter-case Dependencies**

None

### 6.2.2 Specification ID: WEBSTORE-DS-02

**Objective**: The objective of this test case is to provide a white-box (process) test for the update/add to/remove from cartItems algorithm to add/remove an item to/from the cartItems through the service.

**Test Items:** CartItems (Object), getProduct, removeProduct, incrementQuantity, decrementQuantity “update”, “add”, and “remove” buttons

**Input Specifications**

1. Navigate to products you like
2. Click “add” button next to a few products
3. Navigate to cart
4. Click “remove” next to one of the products
5. Click in quantity input box and increase quantity
6. Click in quantity input box and decrease quantity

**Output Specifications**

1. Product page will display and user will be able to add item to cart
2. addItem service will show a new domain object being created.
3. Cart page will display, showing items in cart using getProduct
4. removeProduct service will delete the domain object that was created
5. decrementQuanity reduces items available in stock by the number you wish to order
6. incrementQuanity adds items back to inventory by the number reduced

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

More than one item may want to be added so user might have to make multiple clicks

**Inter-case Dependencies**

None

### 6.2.3 Specification ID: WEBSTORE-DS-03

**Objective**: The objective of this test case is to provide a white-box (process) test for the customer information algorithm to assign a new customer and transaction through the service.

**Test Items:** Customer(name, address, phone, credit card), Customer checkout form, “Process Order” (See SDD: , sec 6.1.11 pg. 21-22)

**Input Specifications**

1. Verify the Customer checkout form is shown after navigating to checkout page.
2. Enter customer’s first name.
3. Enter customer’s middle name
4. Enter customer’s last name.
5. Enter customer’s street address.
6. Enter customer’s city.
7. Enter customer’s state/region.
8. Enter customer’s country
9. Enter customer’s zip code.
10. Enter customer’s phone number.
11. Enter customer’s credit card number.
12. Click on “Checkout” button.
13. If successful, the Guest Information form will automatically close and the system will return to the Guest Reservation form.

**Output Specifications**

1. Customer form will automatically close and the and display confirmation
2. Customer service will show a new domain object being created.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

WEBSTORE must be at a point to accept guest information. At this point, system expects valid input.

**Inter-case Dependencies**

The Cart test case (WEBSTORE-DS-01) must be completed prior to the Customer test because the Customer form can only be accessed after the user selects items and clicks “Checkout” on the navigation bar.

## 6.3 Domain Object Test Cases

These cases are designed to test the data encapsulation and storage portion of the WEBSTORE, and can be performed either by a user testing team or through automated testing.

### 6.3.1 Specification ID: WEBSTORE-DO-01

**Objective**: The objective of this test case is to provide a white-box (process) test for the Customer domain object where the guest name is changed.

**Test Items:** The items to be tested in this test case are the data validation function of the “get” and “set” member functions of the guest object. This case tests valid data entry.

**Input Specifications**

1. The user of the WEBSTORE enters a valid customer name (e.g. John Doe) of string data type.

**Output Specifications**

1. The successful creation of a Customer object
2. Entry into an event log indicating successful update.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

None

**Inter-case Dependencies**

WEBSTORE-DS-03

### 6.3.2 Specification ID: WEBSTORE-DO-02

**Objective**: The objective of this test case is to provide a white-box (process) test for the Customer domain object where the guest credit card information is changed.

**Test Items:** The items to be tested in this test case are the data validation function of the “get” and “set” member functions of the guest object. This case tests valid data entry.

**Input Specifications**

1. The user of the WEBSTORE enters a valid (up to 16 digit) credit card number with or without spaces of type integer.

**Output Specifications**

1. The successful creation of a guest object
2. Entry into an event log indicating success.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

None

**Inter-case Dependencies**

WEBSTORE-DS-03

### 6.3.3 Specification ID: WEBSTORE-DO-03

**Objective**: The objective of this test case is to provide a white-box (process) test for the Customer domain object where the guest name is incorrectly changed. Validate that improper data was not accepted.

**Test Items:** The items to be tested in this test case are the data validation function of the “get” and “set” member functions of the guest object. This case tests invalid data entry.

**Input Specifications**

1. The user of the WEBSTORE enters a user name that includes a value of 0-9 or special character.

**Output Specifications**

1. No update to the Customer entity.
2. Validation shows error of data entry.
3. Event log indicating failure.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

None

**Inter-case Dependencies**

WEBSTORE-DS-03

### 6.3.4 Specification ID: WEBSTORE-DO-04

**Objective**: The objective of this test case is to provide a white-box (process) test for the Customer domain object where the guest name is incorrectly changed and validate that improper data was not accepted.

**Test Items:** The items to be tested in this test case are the data validation function of the “get” and “set” member functions of the guest object. This case tests invalid data entry.

**Input Specifications**

1. The user of the WEBSTORE enters a credit card number that includes spaces or any character other than 0-9.

**Output Specifications**

1. No update to the Guest entity.
2. Validation shows error of data entry.
3. Event log indicating failure.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

None

**Inter-case Dependencies**

WEBSTORE-DS-03

### 6.3.5 Specification ID: WEBSTORE-DO-05

**Objective**: The objective of this test case is to provide a white-box (process) test for the Orders domain object to contain valid information.

**Test Items:** The items to be tested are the getCustomerOrderCollection and setCustomerOrderCollection member functions of the OrderedProduct object.

**Input Specifications**

1. The user of the WEBSTORE attempts to checkout
2. The customer information is entered in the appropriate fields of the customer information form.
3. The user clicks “Checkout.”

**Output Specifications**

1. The user will see a checkout page
2. User input will be accepted
3. Page will process without errors and setCustomerOrder will create a new database object.

**Environmental Needs**

1. Computer or ARM device
2. Sufficient disk space to store increasing amounts of data.
3. Could use external server to store data for recovery purposes
4. Monitor
5. Keyboard
6. Mouse
7. Operating system configured with WEBSTORE installed.

**Special Procedural Requirements**

None

**Inter-case Dependencies**

WEBSTORE-DS-03

# 7. Training Needs

The Technical personnel must be trained to do the data entry transactions and install software on system. The Development team and this document will be the basis of this training.

# 8. Risks and Contingencies

If hardware problems impact system availability during the day, then the test group will schedule their activities during the evening.

The first production runs of the WEBSTORE must be checked out in detail before the put into use and distributed, and any checks in error must be corrected manually.